UNIVERSITA' DEGLI STUDI DI PAVIA

FACOLTA' DI INGEGNERIA
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

DOTTORATO DI RICERCA IN BIOINGEGNERIA E BIOINFORMATICA
XXXII CICLO - 2019

DECISION SUPPORT THROUGH TEMPORAL ANALYTICS ON PATIENT-GENERATED DATA

PhD Thesis by
ELISA SALVI

Advisor:
Prof. Lucia Sacchi

PhD Program Chair:
Prof. Riccardo Bellazzi

Abstract (Italiano)

L'attività di ricerca descritta in questa tesi è stata svolta da ottobre 2016 a settembre 2019, all'interno del Laboratorio di Informatica biomedica Mario Stefanelli dell'Università di Pavia.

Il lavoro è stato motivato dalla recente diffusione di sensori indossabili e ambientali per il monitoraggio dello stile di vita e della salute dell'individuo. Al giorno d'oggi, tali sensori sono comunemente utilizzati sia da soggetti sani sia da popolazioni di pazienti cronici. In questa tesi, utilizzeremo il termine *patient-generated health data* (PGHD) per riferirci ai dati raccolti da tali sensori. I dati PGHD possono contribuire a fornire una visione più approfondita delle condizioni di salute del soggetto e possono facilitare la personalizzazione del percorso di cura per i pazienti cronici. Tuttavia, interpretare serie temporali di PGHD senza strumenti di supporto può creare difficoltà. Pertanto, sono necessari sistemi di supporto decisionale che permettano all'utente di eseguire analisi avanzate di tali dati; eppure tali sistemi sono rari. Ciò può essere dovuto alle difficoltà che insorgono nell'analisi di serie temporali di PGHD, e alla mancanza di linee guida per superarle.

In questa tesi presentiamo tre contributi principali. Il primo contributo è un framework concettuale per l'analisi di serie temporali di misurazioni raccolte da sensori indossabili e/o ambientali, volto a fornire supporto decisionale per la gestione di pazienti cronici. In particolare, combiniamo due approcci, l'astrazione temporale e le regole, per sintetizzare le serie temporali di PGHD e fornire supporto decisionale in funzione dei risultati ottenuti. Inoltre, in questo lavoro descriviamo come abbiamo applicato il framework proposto nella progettazione e sviluppo di due sistemi di supporto decisionale. Di questi due sistemi, uno è volto alla prevenzione delle cadute negli anziani e uno supporta la gestione di pazienti affetti da diabete di tipo 1. Entrambe le applicazioni sono state distribuite e testate in studi pilota che hanno coinvolto pazienti reali. I due sistemi descritti rappresentano gli altri due contributi principali di questo lavoro, poiché presentano funzionalità che li rendono innovativi rispetto agli strumenti noti in letteratura.

Riteniamo che la lettura di questa tesi possa supportare il ricercatore interessato a sviluppare la propria applicazione per l'analisi di serie temporali di PGHD, permettendogli di riutilizzare, almeno in parte, un design che abbiamo già testato nelle nostre due applicazioni.

Abstract (English)

The research activity described in this thesis was carried out from October 2016 to September 2019, within the Laboratory for Biomedical Informatics *Mario Stefanelli* of the University of Pavia, Italy.

This work was motivated by the emerging use of wearable and environmental sensors for monitoring the individual's health status and lifestyle, in his/her living environment. Nowadays, monitoring sensors are widely used, both by healthy subjects and chronic patient populations. We will refer to the data collected by such sensors as patient-generated health data (PGHD). PGHD may allow gaining deeper insight on the subject's health condition and may facilitate targeting care to the individual. However, interpreting time series of PGHD may be challenging and/or time consuming. Thus, decision support systems able to perform advanced analyzes to extract new knowledge from such data are needed; yet they are rare. This may be due to the challenges that rise in the analysis of longitudinal PGHD, and to the lack of guidelines to overcome them.

In this thesis we present three main contributions. The first contribution is a conceptual framework for analyzing time series of PGHD with the ultimate goal of supporting chronic patients care. In particular, we combine two approaches, namely temporal abstraction and rules, to summarize the collected time series of PGHD and exploit the result of the analysis for providing decision support. In addition, we describe how we applied the proposed framework to design and develop two clinical decision support systems. Of these two systems, one aims at preventing falls in the elderly and one supports the management of Type 1 Diabetes. Both applications were deployed and tested in real-world pilot studies. These two systems represent two other main contributions of this work, since they have functionalities that make them innovative compared to the known tools with similar purposes.

We believe that reading this thesis may help researchers develop their own application for analyzing PGHD, leveraging on the design we have proposed and tested in our applications.

Contents

1. Motivation and introduction

This work was motivated by the emerging use of sensors for continuous monitoring of health-related parameters, and by the potential offered by the huge amount of data that can be collected.

Commercial wearable sensors are increasingly popular, thanks to their ease of use and their increasingly affordable prices. In some cases, specific wearables are provided for free to specific patients' categories by the national healthcare service. In Italy, for example, patients affected by Type 1 Diabetes may be eligible to receive a known commercial blood glucose monitoring device through the national health care system [1]. Moreover, non-clinical wearable devices for continuous monitoring are becoming increasingly popular. For example, recently the use of activity trackers has intensified, both by healthy subjects and chronic patient populations [2–5]. Generally, such devices can be used for the long-term monitoring of the subject's physical activity, heart rate profile, and sleep. In addition to wearable sensors, nowadays it is possible to easily purchase environmental sensors, aimed at monitoring the quality of the environment in which the subject lives, and his/her activities inside the house.

When such kinds of data are used for clinical decision making, they can be referred to as *patient-generated health data* (PGHD). The term *patient-generated* indicates that these data are the result of remote monitoring, which takes place in the environment in which the patient carries out his/her usual activities of daily living. The term *health data* indicates that they may contribute to gaining deeper insight on the subject's health condition, and they may facilitate targeting care to the individual.

Given the widespread availability of continuous monitoring devices, both wearable and environmental, collecting large amounts of PGHD has become practical. However, interpreting the collected time series of measurements in clinical decision making may be challenging and/or time consuming. Thus, systems able to extract knowledge from PGHD data to provide clinical decision support are needed. Nevertheless, such systems are still rare. On one hand, this may be due to the fact that the analysis of large amounts of patient-generated continuous monitoring data is a very recent topic, as these data were not widely available until a few years ago. Most likely, a significant number of systems for the analysis of PGHD is currently under development and will be published shortly. On the other hand, the shortage of such systems may be due to the lack of guidelines for their design and development. In fact, several challenges rise in integrating PGHD from multiple sources and in performing temporal data analysis on the collected heterogenous data to support clinical decision making.

In this thesis we will discuss three main contributions which address the limited availability of decision support systems able to analyze continuous monitoring data collected by PGHD devices. First, we will present a conceptual framework that summarizes the steps needed to analyze time series of PGHD with the ultimate goal of providing clinical decision support for chronic patients care. The proposed framework stems from our need to apply temporal PGHD data analysis in two specific clinical domains, namely the management of patients affected by type 1 diabetes and the prevention of falls in the elderly. Although these two clinical domains are very different, they share most of the challenges that rise in exploiting PGHD for decision support, since they are independent of the considered clinical domain. Thus, we summarized the experience gained in formulating an approach to design and develop the two different decision support systems. We believe that the proposed framework may be valuable for other researchers who need to summarize time series of PGHD in order to provide clinical decision support. We will also present the two systems in which we have applied the described framework, since they represent two innovative contributions to the field of decision support systems for chronic care based on the interpretation of remote monitoring data.

The dissertation is organized as follows.

Chapter 2 introduces the readers to the background of this work. Since the focus of this work is designing and developing decision support systems (CDSSs), Section 2.1 explores the existing different definitions of CDSS, including the one we adopted. It also investigates what characteristics influence the users' willingness to use such systems and discusses their adoption in daily clinical practice.

Section 2.2 describes the clinical context for which the framework was built, namely the management of chronic diseases, whose prevalence has recently been increasing. First, the costs and challenges related to assisting chronic patients are outlined, to emphasize the need for decision support in this domain. Second, a short review on sensors that are often used to remotely monitor chronic patients is presented. The data collected using these sensors are referred to as patient-generated health data.

Section 2.3 depicts the state of the art on which we designed our framework, by providing a short review on the literature related to CDSSs aimed at supporting the chronic subject and his/her caregivers in managing the disease. The overview pays particular attention to the clinical domains to which the two CDSSs described in this thesis are targeted.

Section 2.4 introduces temporal data analysis. After discussing the difficulties posed by the analysis of PGHD, we focus on knowledge-based temporal abstraction (KBTA), a known technique for analyzing time series of data. Often clinicians need to search for the occurrence of specific trends, or more complex behaviors, in time series of health indicators. Behaviors of interest in the patient's data are commonly referred to as *patterns*. KBTA allows searching for user-defined patterns in temporal data. After introducing KBTA, we describe Java Time Series Abstractor (JTSA), a software tool that was recently developed within our laboratory to perform KBTA, and we provide an example of its application for analyzing time series of blood glucose measurements.

Chapter 3 describes the proposed framework for providing clinical decision support based on the analysis of longitudinal PGHD data through KBTA. First, we describe at a high level the components of the framework and their desired characteristics, independently of the implementation technologies. Finally, we outline the proposed architecture for CDSSs that perform KBTA exploiting JTSA.

Chapter 4 describes one use case in which the framework proposed in this thesis was applied. NONCADO is a CDSS that monitors the elderly living alone at home and alerts the subject's remote family of possible changes in his/her daily habits, that may correspond to a decline in his/her health status and, consequently, to an increase in his/her risk of falling. The CDSS analyzes data collected by a network of sensors, including both a Fitbit activity tracker recording data on the patient's sleep and physical activity, and environmental sensors, monitoring the quality of the environment the subject lives in, and his/her activities within the house. Possible changes in the individual's habits are detected by exploiting JTSA to apply KBTA on the collected data.

After illustrating the purpose and use-cases of the NONCADO system, in Section 4.1 we describe how the system architecture extends the framework described in Chapter 3. We then describe the patterns that the CDSS searches for in the collected data, and how they are used to assist the monitored subject and his/her remote family.

Section 4.2 describes how we evaluated the NONCADO system in a two-stage approach. First, we describe the preliminary test phase that was carried out on healthy volunteers to tune the parameters needed for applying KBTA. Finally. we report the results obtained in a 2-weeks pilot study involving 16 patients with history of falls treated at the Casa di Cura Privata del Policlinico Hospital, in Milan, Italy.

Chapter 5 describes another use case in which the framework was applied. AID-GM is a web application for supporting diabetic subjects and their diabetologists. The application collects both blood glucose measurements from a device for flash glucose monitoring and data on the subject's sleep and physical activity from a Fitbit activity tracker. AID-GM facilitates data sharing between the patient and his/her diabetologist and provides tools to perform temporal data analysis through KBTA, exploiting the Fitbit data to contextualize the patient's blood glucose profile. After illustrating the use-cases of the AID-GM system, Section 5.1 describes the architecture, based on the proposed framework. We then describe the patterns that the CDSS allows searching for in the collected data, and how the graphical interface presents them to the users to provide decision support.

Section 5.2 describes the system evaluation. First, we describe the preliminary test phase carried out on data willingly provided by a small group of volunteers, which we exploited to tune the parameters needed for KBTA. Finally. we report the outcome of a 6-months pilot study involving

30 diabetic patients receiving care at the Policlinico San Matteo Hospital, in Pavia, Italy.

Finally, **Chapter 6** summarizes and discusses the main contributions of the thesis.

2. Background

2.1. Clinical decision support systems

Definition and aims

The definition of a computerized *clinical decision support system* (CDSS) has been widely discussed in the literature, and several alternatives are available. After performing a review of the literature, we identified three main dimensions in the definition of a *CDSS*, i.e., the aim, the user, and the beneficiary of the system.

The system aim seems to be the most discussed aspect. Some definitions are generic and focus on establishing the main goal of CDSSs in a broad sense. For example, in 2013 Castillo et al. define a CDSS as a system designed to make patient care more efficient, without specifying any details on how that purpose should be achieved [6]. In a review article published in 2015, Khong et al. provided a more detailed definition, describing a CDSS as <<*heuristic-based information technology used to support evidence-based clinical practice at the point of care as the patient or population is being served.*>> [7]. Such definition is more specific, since it suggests that, to make care more efficient, a CDSS should support healthcare personnel in clinical decision making, based on the available medical evidence. It also emphasizes that the timing of action is fundamental, since the CDSS must be able to provide support exactly when the patient (or group of patients) is taken care of. There seems to be consensus on this definition since the late 1990s and early 2000s, when several authors introduce CDSSs as software tools that exploit a domain-specific knowledge-base to timely assist clinicians, or other healthcare personnel, in making clinical decisions [8–10].

In a narrower sense, the scientific community has identified two aims that are characteristic for CDSSs [11]; each CDSS can pursue one or both. The first aim is to help the user gather in one place all the information that is relevant for the clinical task at hand, which can be either making a clinical decision, or monitoring a health condition over time [6]. In general, multiple variables need to be taken into consideration in clinical decision making; in addition, some variables are not informative as they are, but must be processed and/or summarized to be informative in the decision task. In this context, a system capable of providing a summarized and integrated view of all the variables of interest for the decision task is often necessary, or at least useful. The method of integrating and visualizing the different variables depends on the decision problem at hand. Since the systems that pursue this aim are specific to the decision problem and are meant to guide the user to focus on the most relevant aspects for its solution, their graphical interfaces are often called *insight-driven dashboards* [6].

The second identified aim is to provide recommendations and/or reminders [11]. Compared to the systems that only visualize data, the systems pursuing this aim are defined as *active* CDSSs [6]. Depending on the application domain, the provided recommendations may support a wide range of tasks, that might be more or less complex [7]. For example, some systems support the clinician in making a diagnosis for a specific patient, by providing suggestions computed by matching the patient's data to recommendations formalized in validated clinical guidelines. Other systems allow early detection of drug-related adverse events, by monitoring a specific set of vital signs and comparing the measured values with known critical thresholds [9]. Other patient-oriented systems generate a reminder when the subject does not remember to take his/her therapy. Others have the purpose to run predictive models built on data, such as risk scores, and to compute the patient's risk of developing complications of interest for the considered domain [11].

Besides defining what aims are characteristic for CDSSs, several definitions mention the intended users. As it can been noticed in the previously reported definitions, generally the initial descriptions of *CDSS* explicitly mention healthcare personnel as the target users of the system. However, more recently the definition of CDSS has been extended in terms of target user; nowadays the term *CDSS* can also be used to refer to systems that are mainly, or exclusively, intended for patients [10, 12].

While the target user is a component that has changed over time, all the definitions available in the literature agree on who the final beneficiary of the CDSS should be, i.e., the patient. To emphasize the relevance of the beneficiary in depicting the concept of CDSS, we refer to the volume <<*Clinical Decision Support - The Road to Broad Adoption*>>, which in 2014 described a CDSS as <<*information and communication technologies to bring relevant knowledge to bear on the health care and well-being of a patient*>> [11].

To be compliant with the new trends of decision support, in the following of this thesis we will adopt such definition when presenting our work.

Adoption of clinical decision support systems in clinical practice: opportunities and challenges

The willingness of health care personnel and patients to use CDSSs has also been evolving over time [10]; the acceptance towards CDSSs has been increasing, although their use still involves criticalities [13].

One of the main criticalities is the users' computer literacy, that is often limited. According to the literature [6], the complexity of the system's graphical interface is one of the main reasons of non-adoption of CDSSs that are most reported by the users. It is common that users who are not comfortable with using technology do not accept the system. According to a report produced in the US in 2017 by the Agency for Healthcare Research & Quality, the health care personnel struggle to use computerized systems in general [14]. Among the different roles in health care, the nurse seems to be the one who has the worst relationship with technology, leading to a low level of adoption of CDSSs within such category. Providing statistics on the computer literacy of health care personnel is not straightforward, since the attitude towards technology highly depends on the considered country, besides varying significantly over time. For example, in a recent cross-sectional study on 554 health care professionals working in Ethiopian hospitals, only 18.7% showed high computer literacy [15]. In developed countries, digital skills seem more widespread among the health care personnel. For example, in a study conducted in Australia on 246 respondents, 80% of them reported using a computer on a weekly basis [16]. In a study performed in Austria in 2009, 1160 medical students filled-in a questionnaire about their use of computers. According to the results, 94% used a computer for their studies and used regularly at least a set of functionalities, including text editors and e-mail accounts.

Even users with high computer literacy are often refractory to the use of CDSSs. As regards systems that produce warnings and reminders, a commonly highlighted criticality is an excessive frequency of notifications received by the user from the system [6, 13, 17]. In particular, a high number of false positive warnings, or a high number of notifications sent with incorrect timing can undermine the user's trust in the system. In the literature, this phenomenon is known as *alert fatigue* [18].

Other systems that are not used willingly are those that require the user to invest a significant amount of time in entering input data [6, 19–21].

In addition, CDSSs that support the diagnostic process have limited adoption rates [6, 14, 22]. According to the review [6] published by Castillo et al., the main reason is that the majority of these systems have two weaknesses. First, they often implement a clinical knowledge base that is not wide, or detailed, enough to support the physician in the differential diagnosis process. Second, they often do not manage the temporal component of the course of the disease, e.g., to produce suggestions, they rarely consider in detail the temporal evolution of the patient's symptoms, that represents one of the key factors in the diagnosis process [23].

However, even systems that implement a suitable knowledge base when deployed can become critical to use over time, due to poor maintenance. Usually, during the development of the system, acquiring and implementing the knowledge necessary to provide support is time consuming. Keeping it up to date might be just as demanding, thus it is considered one of the major challenges in the development and distribution of CDSSs [22]. Consequently, another frequent reason for not adopting CDSSs is the user’s awareness or perception that the system is not up to date [14].

Another key factor affecting the adoption of the system is the adherence of the CDSS to the user's expectations [6, 13]. It is known that the acceptance rate of the system is higher when the CDSS provides the user with those functionalities that she/he considers necessary. Even providing more functionalities than necessary can be counterproductive to the adoption of the system. Without the necessary precautions, a gap may arise between the user's expectations and the ambitions of the CDSS developer. In fact, the developer is often a researcher, who aims to create a system with innovative features, which can represent a step forward compared to the state of the art of decision support systems. However, the user often needs simpler features and may not appreciate the proposed innovative features, if she/he does not understand and share their usefulness. As a result, those systems for which users have been actively involved in the development process generally have higher adoption rates [13]. Through an intense collaboration with the intended users, it is possible to develop an innovative system that is innovative, but also perceived as useful by the consumer. This is the concept of *participatory design* [24], in which representatives of the end users are active members of the group of the system designers and are involved in the entire development process. The literature provides several examples of adoption of the participatory design approach for developing CDSSs [25–29]. In some cases, the authors have shown that user satisfaction with the CDSS developed using the participatory design approach is higher than the user satisfaction with alternative systems developed with a traditional approach [30].

When CDSSs are willingly adopted, they can have a significant impact on the health care personnel' performances and/or on the patient's outcome [20, 31–33]. The application in which CDSSs seem to be the most successful is the prescription of drugs [34–36]. In particular, the available scientific evidence points out that in this context the use of CDSSs reduces the number of prescription errors, making cases of adverse events (e.g., drug-drug interactions, allergic reactions, and inappropriate prescriptions in pregnant women) less frequent. On the contrary, the most critical CDSSs are those dedicated to supporting the diagnostic process [20, 31–33]. The evidence concerning the effectiveness of these systems is still quite limited, and it is known that clinicians often do not trust computerized systems when performing the diagnostic task, which is considered one of the most complex in the clinical domain [37].

Given the potential benefit of the CDSSs that are willingly adopted, several authors have reviewed the literature to produce a list of general requirements that the systems must satisfy to be accepted by users, as well as effective [6, 38–40]. Most of the identified requirements address the issues discussed above. For example, those reviews highlight the need to deliver timely suggestions, to provide explanations on how the proposed suggestions have been computed, to avoid alert fatigue, to make the user perceive that she/he is saving time and/or making better decisions, and to avoid requiring the subject to enter too much input data manually. These requirements have been taken into account in the formalization of the proposed framework, aimed to provide decision support through temporal analytics on patient generated (health) data. As the framework is mainly dedicated to supporting chronic patients, the next section will introduce the clinical context of this work, i.e., chronic illness and its impact on the health care system.

2.2. Chronic diseases and patient-generated (health) data

2.2.1. Chronic diseases: costs and challenges

As seen for *CDSSs*, there is also no consensus in the literature on the definition of *chronic disease* [41]. The concept itself is fairly shared; according to all the definitions available in the literature, a chronic disease is a long-term health-related issue that requires continuous treatment or monitoring, possibly leading to limitations in the patient's activities of daily living. However, there is no consensus on the minimum duration that a disease should have to be considered chronic. According to the National Center for Health Statistics [42], an agency of the U.S. Federal Statistical System, such duration consists of three months [41]. According to the World Health Organization (WHO) and to the Centers for Disease Control and Prevention (CDC), such duration is higher, i.e., one year [43]. In addition, the WHO does not consider the diseases that are transmissible from patient to patient as chronic diseases. In this work, we will consider the definition provided by the WHO. Well known examples of chronic diseases are specific cardiovascular diseases (e.g., dilated cardiomyopathy), specific neurological diseases (e.g., dementia), cancer, diabetes, and a subset of respiratory diseases, including asthma [44].

Worldwide, the prevalence of chronic diseases is considerably high. In 2014, an overview on chronic illness in America was depicted in a report produced by the Agency for Healthcare Research and Quality (AHRQ), a government agency working within the United States Department of Health and Human Services [45, 46]. According to such report, in 2014, 60% of the American population had at least one chronic disease. The co-presence of multiple chronic diseases in the same subject was also frequent: the AHRQ reported that 40% of the population had at least two chronic diseases and 12% at least five. Even higher percentages have been recorded by health insurance agencies that have low-income consumers, such as Medicaid [47]. In general, as regards the affected population, in 2014 in America the prevalence of co-morbidity was more frequent in the elderly; the most common chronic disease was hypertension, followed by depression and diabetes.

According to the WHO, the chronic illness situation in Italy is not different from the overview that has been described so far [48, 49]. In Italy the high prevalence of chronic diseases is also due to population aging, since it is known that our nation shows the oldest population in Europe, having 22% of the inhabitants aged over 65 [49]. In 2014 the most frequent causes of death were chronic diseases, namely cardiovascular diseases, cancer, neurological diseases, and chronic respiratory diseases. Furthermore, for the same year the WHO reported also high prevalence of other chronic diseases, which did not lead the patients to death, but reduced their quality of life. In particular, 5% of the population had asthma, 6.5% diabetes, and more than 20% hypertension.

According to the estimates provided by the WHO, by 2020 the worldwide annual percentage of deaths due to chronic diseases will rise to 73%, and 60% of the reported diseases will be chronic. In addition to health issues due to old age, the most frequent health conditions will be cardiovascular diseases (e.g., stroke), cancer, obstructive pulmonary disease, and type 2 diabetes [44].

Given their prevalence, chronic diseases are often referred to as <<*the epidemic of our time*>> [50–52]. Obviously, this has a significant impact on the health-related costs sustained by national health care systems and by the individual. Due to the continuous need for care, a subject having chronic issues costs considerably more than a patient suffering from an acute problem in terms of medical expenses [53]. The WHO has identified three components in the cost of chronic illness. The *direct costs* are those related to the concrete cost of treatment and represent the easiest component to estimate. A second component is represented by the *indirect costs*, i.e., secondary costs, which may result from multiple circumstances, such as the patient's or his/her relatives' unproductiveness during the care process (e.g., unemployment or sick leave), the need to hire specialized health personnel for home assistance, and the patient's expenses to reach the health centers where she/he is treated. Given their heterogeneity, indirect costs are often difficult to estimate. However, the third component is the most challenging to measure. It is referred to as *intangible costs* and includes the secondary costs that are due to the psychological consequences of chronic illness and to reduced quality of life [53, 54].

To give an idea of the extent of the burden in America, we refer to a report by the CDC, based on data referring to 2016 [55]. According to such report, in terms of tangible costs, the annual health care expenditure consists of 3.3 trillion $, and 90% of this amount is due to the management of chronic diseases. To compare this data to the Italian situation, we consulted a report produced in 2018 by The National Observatory on Health in the Italian Regions (Osservatorio Nazionale sulla Salute nelle Regioni Italiane) [56]. According to the reported results, the impact of chronic illness on the total annual expenditure for health care appears to be slightly lower in Italy, although remaining considerably high, being equal to 80%.

In this context, technology can help to remotely monitor chronic patients throughout the course of the disease, and provide decision support accordingly, facilitating the containment of the costs of chronic illness [57–59]. The use of technology can be useful to health managers and policy makers, who need to thoroughly analyze the patient population for which they have to make clinical decisions, such as whether or not to activate a specific health care intervention or preventive program. In that context, CDSSs are aimed to process data from a large number of patients, by applying data mining or data analysis techniques, and to produce intuitive summaries (e.g., graphs), that can be easily navigated by the user, in a process that can be defined as *visual data exploration* [60]. Often, such systems do not provide suggestions, but allow users to apply their analytic and decision-making skills leveraging on the data overview provided by the system. As stated by Raghupathi et al. [60], such systems <<*turn information overload into opportunities*>>.

Systems dedicated to health managers are beyond the scope of this thesis, which will focus on systems aimed to support the chronic patient, his/her family, and his/her clinicians in managing the disease. However, the concept of converting information overload into opportunity is one of the bases of this work. Generally, few clinical parameters need to be monitored for chronic patients (e.g., blood glucose for diabetic subjects), but to gain insights on their trends the desired sampling frequency is high, and the duration of the monitoring period is long, potentially equal to the subject's expected life. While with the technologies of the past such daily monitoring was not easily practicable, nowadays it is possible to collect a set of health indicators around-the-clock, in a non-invasive manner (see Section 2.2.2) . A large availability of data consequently involves the need for tools capable of analyzing them. Thus, systems capable of processing time series of clinical parameters of interest can be useful to gain deeper insight into the patient's profile, to monitor the outcome of the care process, to provide personalized suggestions based on the subject's profile, and to identify early deterioration in the patient's health condition. This can facilitate both the patient and the doctor in managing the disease and it might help contain the costs of chronic illness [57]. As regards the users' perception, it is also known that chronic patients and their families appreciate computerized systems that help them day by day in the management of their pathology and of its consequences on their daily lives [61].

Computerized systems for supporting chronic care often need to collect a set of variables of interest to gain deeper insight on the chronic patient's health condition and how it evolves over time, between medical visits. In the next section we will present an overview of the monitoring data which may be used to that aim. The overview will dedicate particular attention to devices used for diabetic and elderly subjects, which are the patient populations to which the two applications described in this thesis will be addressed.

2.2.2. Patient generated (health) data: an opportunity for monitoring chronic patients

Recently, the use of the term *patient-generated health data* (PGDH) has spread to indicate data useful for understanding the state of health of a subject, and collected outside health care centers [62]. PGHD may include heterogeneous variables, being either clinical parameters that can be measured without medical examinations and other data useful for contextualizing clinical data, such as details on the subject's lifestyle (e.g., amount of physical activity performed during the day). PGHD include data that are self-reported by the patient via questionnaires, which can cover a wide range of health-related topics, depending on the purpose for which they are collected. For example, self-reported data might include weight, meal habits, perceived quality of life, symptoms, possible difficulties in carrying out daily activities, considerations regarding the impact of the illness on the emotional sphere, or off-the-shelf medications. However, nowadays self-reported information represents a very limited percentage of the PGHD volume. In fact, advances in technology have made it possible to collect several health indicators automatically and on a regular basis by using sensors, which can be wearable or not.

Recently, the use of wearable sensors has intensified thanks to technological innovation, which has made sensing components increasingly miniaturized, designed to be placed on wearable accessories, such as bracelets, belts or clothes [63, 64]. Nowadays, several health indicators can be continuously monitored using wearable sensors, including for example blood pressure, blood glucose, heart rate, electrocardiogram, body temperature, electrodermal activity blood oxygen saturation, respiratory activity, physical activity, consumption of energy, and sleep quality and quantity [65, 66]. Usually, the sensor is wireless: it performs the measurement and communicates the value to another device, using a low energy consumption communication protocol, such as Bluetooth. The other device might be a smartphone, a computer, or a system designed specifically for that sensor. The device often displays the collected values to the user, and sometimes analyze them, and/or stores them in a dedicated cloud repository. The technologies used by wearable sensors to perform the measurements, which vary significantly from one health indicator to the other, are beyond the scope of this thesis, which will focus more on how to retrieve the time series of the collected values and how to analyze them.

Some sensors are targeted on a specific category of patients. For example, sensors for measuring the subject's blood glucose (BG) values are used for diabetic patients. Usually, several types of these sensors are available for measuring the same health indicator, but in clinical practice some types are more used than others due to different reasons, including ease of use, durability or accuracy [65]. For example, several devices can be used to measure BG values, including non-invasive bracelets, subcutaneous sensors, and even contact lenses [67]. However, in daily practice subcutaneous sensors are the most widely used to remotely monitor the patient's BG profile around the clock.

Other sensors have wider domains of application. This is the case of activity trackers, i.e., wearable sensors originally developed for the general population to monitor the subject's physical activity, which are now often used for monitoring the patient's lifestyle in several chronic conditions [66]. The same consideration holds for sensors measuring blood pressure and heart rate. Long battery life, comfort, and ease of download of the measured values are factors that allow long-term use of these sensors for monitoring chronic patients between check-ups. More and more frequently, commercial sensors measuring such health indicators are used by the healthy population for prevention purposes. In a review published in 2019, two clinicians, Bhavnani and Sitapati, described a new paradigm of health care, which they defined *Virtual Care 2.0*, in which monitoring by wearable sensors plays a central role in preventing health of the individual [68]. In particular, they presented a set of imaginary examples of how these sensors may be used by the healthy individual for early detection of changes in his/her health indicators, possibly contributing to early diagnosis of diseases. For example, in one use case a healthy woman wears an activity tracker for a long time, constantly monitoring her heart rate (HR). When the monitoring device returns a HR profile that is clearly different from the one that she is used to observing, she worries and requests an appointment from a cardiologist. Further clinical investigations make it possible to diagnose a heart disease, for which she is treated promptly. While the previously described scenario was hypothetical, the new *Virtual Care 2.0* paradigm is also observable in real-world clinical cases reported in the literature, as the one published by Weichert in 2019 [69]. In this case report, a woman with history of transient ischemic attacks presented to the emergency room, although she was asymptomatic, because her activity tracker kept reporting high HR values while she was resting and not engaged in any physical activity. Following a battery of clinical tests, she was diagnosed with atrial fibrillation. Case reports like the described ones have raised interest in wearable devices and have made the analysis of their data a current research topic.

Wearable sensors are not the only technology available for remote patient monitoring. For managing chronic conditions, such as respiratory diseases, it is also important to monitor parameters related to the quality of the patient's living environment, such air pollution. It is known, for example, that a short-term increase in air pollution may cause acute health events in specific patient populations, including the elderly, children, and subjects affected by chronic diseases such as congestive heart failure, diabetes, and cardiovascular diseases [70]. Thus, interest in monitoring air quality has recently grown, especially in urban areas. Wireless sensors networks have been produced for measuring the most critical air quality parameters, including carbon monoxide, nitrogen dioxide, ground level ozone, sulfur dioxide, particulate matter, and lead [71]. The research and innovation Horizon 2020 program, by the European Commission, has funded several projects that include air monitoring through environmental sensors. Among these, one is the PULSE (Participatory Urban Living for Sustainable Environments) project [72]. The PULSE project collects air quality data to study the correlation between the subject's living environment and the onset of specific chronic diseases, i.e., asthma and type 2 diabetes.

It is not only the external environment that can affect the well-being of chronic patients, but also the quality of the living indoor environment, i.e., mainly the subject's home. For example, it is known that maintaining adequate ranges of temperature and humidity in the elderly's home is a fundamental factor for their well-being, since an inadequate environment can weaken the subject [73–75]. Digital sensors can measure environmental temperature almost continuously, providing up to one measurement per second. Another parameter to monitor for the well-being of the elderly is the luminosity of the environment in which they live, since it is known that poor lighting condition may increase the individual’s risk of falling [76–79]. Measurements of indoor luminosity can be collected by sensors based on photoresistors, which can provide values even more frequently than digital temperature sensors.

In addition, there are environmental sensors that do not measure parameters related to the quality of the subject’s living environment, but which are useful for monitoring the activities carried out by the individual inside his/her home. This is the case for example of motion detectors. Among these, the ones based on passive infrared (PIR) sensors are widely used. They detect movement by detecting infrared radiation emitted by or reflected from the objects in their field of view. Thus, they can return a binary value, assessing the presence or absence of movement at that instant. The frequency of measurements can be high, up to one value every second or couple of seconds. Generally, they are cheap, so it is possible to install several PIR sensors in each room, to map the movements of the subject inside the house. However, PIR sensors cannot distinguish *who* moves when there is more than one subject moving within their visual field, so it is not trivial to map movements within the house in case of presence of several people. In the hospital setting, this difficulty can be overcome by tracing the movements of patients with other technologies, which require the subject to wear a portable device (known as *tag*) that uniquely identifies him/her and detects his/her position within the building [80]. However, this solution is viable for use cases of limited duration, since it is not realistic to expect a chronic subject to wear the tag in the long term, unless it is integrated into a device that he/she would wear anyway, such as a smart watch. Alternatively, camera-based sensors represent another common modality for monitoring activities without using wearable sensors. However, the processing of data collected by the camera is not trivial and requires advanced image analysis techniques. Furthermore, installing a large number of cameras in the patient's home can be expensive, as well as intrusive to the subject's privacy [65].

In the following paragraphs we will provide more details on the two categories of wearable tools for collecting PGHD that have been used in this work, namely activity trackers and devices for blood glucose monitoring.

Activity trackers

Nowadays, activity trackers are among the most widely used wearable devices for general purpose monitoring, since they are cheap, and they can be comfortably worn by the individual around the clock. Several brands of activity trackers are available, including Fitbit [81], Polar [82], and Garmin [83]. Regardless of the brand, activity trackers usually record a standard set of parameters related to the user’s sleep and physical activity.

Usually, sleep detection is carried out using proprietary algorithms that combine movement and heart rate (HR) measurements. For each detected sleep occurrence, the information provided by the trackers usually include start time, end time, number of night-time arousals, and sleep efficiency, i.e., the ratio of the time spent sleeping to the time spent in bed. Generally, they also provide sleep staging, i.e., they distinguish deep sleep from light sleep. Their reliability in detecting and analyzing sleep has been tested in previous studies [84–86]. The reported results confirm the trackers' ability to detect overall sleep duration and some macroscopic phenomena like night stepping, while the accuracy in detecting the different sleep phases is lower. As a general conclusion, sleep data from activity tackers cannot be used to for diagnosing sleep disorders, or for performing fine sleep analysis. However, allowing the continuous monitoring over a high number of nights, they may be useful to detect changes in the subject's sleep habits[87]. While monitoring sleep is important for any individual, detecting changes in sleep habits is particularly important for chronic patients. For example, sleep anomalies may be suggestive of the onset or worsening of dementia[88]. Recent studies claim that sleep quantity and quality affect glycemic control in patients affected by type 2 diabetes [89]. It is also known that sleep disorders might also increase the risk of falling in the elderly [90–92].

As regards physical activity, for each detected workout (i.e., activity lasting more than a threshold duration, that may be brand-specific), activity trackers record start time, end time, and intensity. They also provide a summary of the overall daily activity, including the total number of steps and an estimate of the calories burned by the subject. However, this estimate varies significantly depending on the brand of the device. In addition, most activity trackers also monitor the subject's heart rate (HR) on a regular basis, most commonly minute by minute.

Usually, the data collected by the tracker can be visualized using a mobile application. Upon synchronization, the data is transferred from that application to a proprietary cloud. Once in the cloud, it can be retrieved by any third application that has been explicitly authorized by the user.

Blood Glucose Monitoring Systems

To optimize therapy for the individual, diabetologists need to monitor his/her BG values over time. Recently, new devices for continuous glucose monitoring (CGM) have been proposed, to reduce the need to perform blood glucose tests using the traditional glucometer, or smart glucometers, while providing deeper insight on the patient's BG trends, by monitoring the BG value around-the-clock [93]. Reasonably, deeper understanding of the patient's glycemic profile results in a better ability to maintain the BG value in a target range, thus reducing the number, or duration, of the episodes of hypoglycemia and hyperglycemia experienced by the patient [93, 94]. Several wearable devices for CGM are available on the market, including the Medtronic Minimed, and the Dexcom G4 Platinum CGM [95–97]. Usually, such devices exploit a subcutaneous sensor, which detects BG concentration in the interstitial subcutaneous fluid. The collected measurements are then displayed on a reader or on a smartphone application. While CGM devices return the BG value continuously, other devices apply flash glucose monitoring (FGM), which means they require a scanner to periodically collect the BG values from the sensor. Among the FGM devices, the Abbott's Freestyle Libre System (Figure 1) is one of the most known [98].

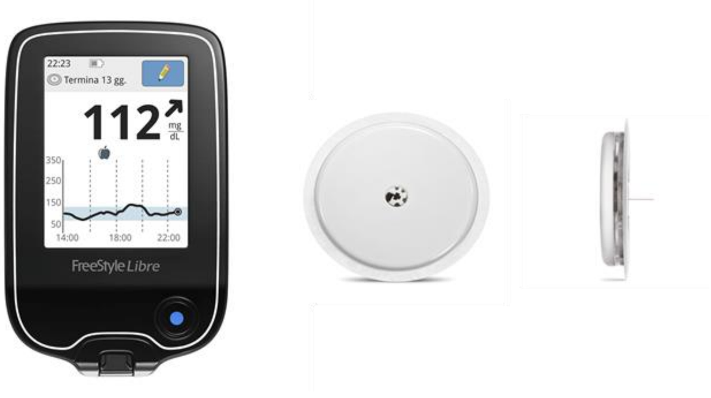


Figure 1: The Freestyle Libre system for flash glucose monitoring

As many FGM devices, the Freestyle Libre system integrates two components. The main component is the wearable sensor (on the right in Figure 1), which automatically performs one BG measurement per minute. The system also includes a reader (on the left in Figure 1), which the patient must use every 8 hours to scan the sensor and export the collected BG measurements. After the scan, the Freestyle Libre system provides the patient with the recently measured values and with a summary of their most recent trend, in the form of an arrow [99]. The slope of the arrow indicates whether the patient's BG level is rising or falling, and the extent of the variation. This information should help the subject in selecting the dose of insulin to be injected, if needed. By connecting the reader to a dedicated software provided by Abbott, the BG data can be uploaded to the Abbott system for further analysis. The proprietary software allows visualizing reports such as the daily glycemic profiles and the average glucose profile (AGP) over a pre-defined time interval (e.g., latest 2 weeks). Usually, these summaries are then discussed with the diabetologist during the patient's check-up.

It is proven that FGM and CGM systems improve the quality of life of diabetic patients, increasing the permanence of the subject's BG value in the target range [99]. Recently, the Freestyle Libre system has also been approved for use in pediatric patients, and it is currently being adopted by an increasing number of subjects [100].

2.3. Decision support systems for chronic patients

This section will provide a review of the literature related to the CDSSs aimed at supporting the chronic subject and his/her caregivers in managing the disease. The review will depict the state of the art on which we built the proposed framework for providing decision support through advanced analyses on temporal patient-generated health data (PGHD). As the previous section, after a general overview, the chapter will focus on systems for diabetic and elderly subjects. We identified four major application areas in the management of chronic illness that can benefit from the use of CDSSs, although they are not mutually exclusive.

The first major application area is facilitating the coordination between the different and heterogeneous actors involved in the management of the disease. As highlighted in a review published in 2017 by Koiij et al. [101], significant effort has been directed towards such aim. In particular, we can identify two main sub-groups of systems, covering two different use cases. On one hand, there are the systems that allow the exchange of information among doctors, potentially having different specialties, who need to interact to fully understand the patient's clinical situation. For example, to facilitate shared care in the diabetes context, in 2008 Smith et al. presented a system that allows the endocrinologist to exchange messages with the diabetologist, so that she/he can have deeper insight on the patient's health condition when discussing the treatment plan with the subject [102].

On the other hand, there are the CDSSs that facilitate communication between health care personnel and the patient, or communication between health care personnel and the patient's caregivers. Such systems often also include video conferencing features [103, 104]. Among these, some systems allow clinicians to deliver remote consultations for patients who live in areas from which it is not easy to reach the health care center. Other systems are designed for health care personnel to provide timely support, both practical and psychological, and educational material to the patient's families. For example, some systems allow caregivers of deaf patients to remotely consult with experts on sign language [104].

The second macro area is represented by decision support tools for diagnosis and characterization of the disease, and for treatment selection. As an example of tool to support the diagnosis of chronic respiratory diseases, Harber et al. proposed a system to guide the clinician in early recognition of the specific type of asthma that the patient is suffering from, to establish whether it may be related to the subject's work environment [105]. In particular, their system asks the clinician to fill in a questionnaire on the patient's symptoms, and then computes a set of suggestions according to the provided answers, by applying a set of rules extrapolated from a well-known clinical guideline.

While few systems focus on diagnosis, more systems focus on treatment selection and/or optimization. Many of the systems that support the selection of therapy for chronic patients also provide functionalities for monitoring and reviewing the treatment plan over time. For example, Ebrahiminia et al. designed a rule-based CDSS to ease the management of hypertension [106]. In particular, the system stores the measurements of the patient's blood pressure, entered by the clinician. If the values provided over time do not show the specific effect that the treatment should have obtained (e.g., decrease or increase in blood pressure), the system may suggest changing the dose of the prescribed medication, or prescribing a different drug. Ebrahiminia et al. designed the system to be flexible, so that it could be extended to other chronic conditions (e.g., type 2 diabetes) simply by adding new rules in the knowledge base of the tool. Often these systems do not only take into account the effect of the therapy on the measured outcome (e.g., blood pressure values, in the previous example), but also of possible changes in the patient's characteristics. For example, Gorman et al. proposed a CDSS for selecting the treatment for diabetic patients, that also advises the diabetologist to reconsider the treatment plan when a female patient reaches the menopause age, which requires special precautions.

Other tools may be used to facilitate patient involvement in the decision-making process, while easing the communication between the subject and the clinician. Decision aids are known to be especially useful for the patient population that is most affected by chronic diseases, namely the elderly, who may need support to understand the possible consequences and/or risks involved in the different therapeutic options [107].

The third major application area includes CDSSs for supporting the patient in the daily management of the disease, at home. For example, in 1994 a research group from the Massachusetts Institute of Technology (USA) described a set of projects meant to support chronic patients at home, within a wider project called *Guardian Angel: Patient-Centered Health Information Systems* [108]. In general, the *Guardian Angel* project aimed to helping the patient keep track of his/her medical records and/or measurements performed at home (e.g., blood glucose measurements), possibly receiving patient-specific suggestions based on their content. For example, one use-case of the *Guardian Angel* project is to support diabetic patients in adjusting the dose of insulin to be injected, according to their blood glucose (BG) measurements, collected using glucometers. More recently, systems for supporting the diabetic patient in adjusting the dose of insulin consider the patient's blood glucose (BG) profile, collected using continuous monitoring devices [99]. As anticipated in Section 2.2.2, FGM and CGM devices often provide the patient with an indication of the most recent trend in his/her BG profile, in the form of an arrow. According to this information, the patient should be able to autonomously compute the amount of insulin to inject, in case of increase of BG value, or the amount of carbohydrate to intake, in case of decrease. Recently, some CDSSs have been proposed to guide the patient in this decision task [99]. However, their accuracy is to be tested, since these systems often do not consider factors which may impact on the glycemic metabolism, such as the patient's physical activity, which can affect his/her BG value up to several hours after the end of the workout [97].

The fourth major application area includes CDSSs for remote monitoring of chronic patients and has developed quickly since the 2000s. The monitoring can be aimed at observing the patient's health status over the period of time between check-ups, to keep the clinician up to date on what happens to the subject outside the health care center. For example, in 2010 Ciccone et al. [109] proposed a system that ask patients affected by cardiovascular diseases to share clinical parameters (e.g., measured weight and blood pressure) with their clinicians on a regular basis, while waiting for a medical appointment. Similar systems are available for several chronic conditions, and tend to replace the paper-based diary, which used to represent the standard monitoring tool before the use of technology became widespread in the management of chronic diseases. Of course, the set of monitoring variables depends on the disease. For example, CDSSs for supporting patients affected by asthma usually ask the patient to enter symptoms and details on inhalations [110]. Some of them are also integrated with smart inhalers, which record details on inhalations automatically. Some applications provide additional functionalities, including reminders for medication and access to educational material. CDSSs that include smart diaries are also available for diabetic patients [103, 111]. Usually, such systems assist the diabetic subject in recording the daily BG readings collected using glucometers. Some systems require diabetologists or nurses to regularly check the values entered by the patients. Other tools are only meant to share the measurements and to store them, so they can be analyzed by the diabetologist during the following medical appointments.

Recently, much focus has been directed towards improving CDSSs for monitoring diabetic patients. The advances in this field are triggered both by the spread adoption of CGM and FGM devices, and by the need to complement the information provided by the patient's BG profile with information on the subject's lifestyle. It is well known that the BG profile over the day is influenced by multiple factors related to the patient's daily routine, which alter the BG metabolism and/or the body response to insulin [97]. Besides the consumption of meals and the therapy intake, such factors include, for example, physical activity and sleep, in terms of both quality and quantity. In fact, low sleep quality may cause hyperglycemic effects, up to several hours after the awakening. Physical activity may lead either to hypoglycemic episodes or to hyperglycemic episodes, up to 48 hours afterwards. Since their effects on the BG value do not run out immediately, it is fundamental to keep track of the patient's sleep and activity over time. Collecting data from diabetic patients wearing both a BG monitoring device and an activity tracker may help to better understand the relationship between BG values and HR, which is a debated research topic in the literature [112–114]. The need to complement the glycemic profile with information on the subject’s lifestyle has been recently discussed by Rodríguez-Rodríguez et al. [97]. In their work, the authors monitored a diabetic patient (male, 25 years old) for 2 weeks, using both the Abbott Freestyle Libre system and a Fitbit activity tracker, which collected information on the subject’s sleep and physical activity. The subject was also asked to report meals and insulin intake. The graph integrating all the collected data helped interpreting BG values in relation to the patient’s lifestyle factors, that are known to influence the subject’s status. Since the interest in integrating the BG profile with lifestyle information has been increasing and the use of wearable sensors has intensified, smart applications to facilitate data integration are needed. Several commercial platforms focus on supporting patients affected by Type 2 Diabetes [115–118] by tracking both BG measurements collected using glucometers and activity data collected by trackers. Such applications usually provide dashboards for the integrated visualization of the collected information, and sometimes allow the patient to share the data with the physician. Recent applications intended for Type 1 Diabetes collect and visualize CGM or FGM monitoring data [119]. Such applications usually provide daily reports, including the number and the duration of hypoglycemic and hyperglycemic episodes, possibly filtered by time of occurrence (e.g., nighttime or daytime). They also provide the same summaries as the CGM or FGM proprietary software, including the Average Glucose Profiles (AGPs). However, fewer applications integrate BG monitoring (either CGM or FGM) with activity and sleep tracking. One of the first solutions proposed in this area is Nightscout [120], an open source project developed by volunteers in 2014 to help the patient, or his/her family, set up a custom system for collecting and visualizing both BG monitoring data and activity tracking. However, this solution was not suitable to all patients, since it required the ability to build and maintain such a system, which includes a web site and a database. Other solutions [121, 122] are commercial integrated systems, which include a sensor for CGM and proprietary software to process the collected data. Such systems gather information on physical activity using proprietary insulin pumps [122], or commercial activity trackers [121]. To our knowledge, such systems provide an integrated visualization of data, but do not provide any tool to perform advanced analyses that combine data from both the sources. The same consideration holds for recent applications, such as the Diabetes:M mobile application [123], which is not developed by the same companies that produce the sensors, and aims at being compatible with multiple brands of sensors, insulin pumps, and wearable devices for activity tracking. Despite the growing interest in tools for the remote/continuous monitoring of diabetic patients, the literature lacks CDSSs capable of performing advanced analysis on time series of PGHD.

Another recent trend for the remote monitoring and support of chronic patients is the design of systems to make the subject's home "smart" [124]. Usually, these are independent systems, designed for the patient and his/her family, and are not integrated with hospital information systems. The beneficiaries are often the elderly [104, 125]. The majority of smart houses have been designed to support the management of the elderly suffering from neurological diseases (e.g., dementia) and living at home, or to support the elderly showing high risk of falling [104, 126]. For example, in [127] Khattak et al. presented a system for monitoring the activities of daily living of the individual affected by dementia. The proposed CDSSs gathers data from several sensors, including cameras, activity trackers, and localization systems. The CDSS then applies a set of rules to the collected data to identify which activity the patient is performing within a set of activities of interest, which includes for example sitting, running, and exercising. On demand, the CDSS can report the patient's current status (in terms of ongoing activity) to its users, i.e., the patient's family or a nurse. In [128], Juarez et al. presented a tool that collects data from motion sensors placed within the house and visualizes them in a specific graphical form that should help the user identify unusual movement patterns that may correspond to unwanted events (e.g., occurred fall, burglary). However, most of the research effort related to smart houses is focused on the detection or management of falls [129, 130]. In that field, most of the existing projects are aimed at providing support when a fall occurs. Usually such systems are designed to detect the fall event, reach for the family member who is the closest to the patient's position, and possibly arrange an healthcare intervention [131]. Few systems focus on fall prevention. The majority of them assist the subject in performing specific exercises to maintain his/her walking ability [132]. It is in fact known that elderly subjects prefer to carry out any prescribed physical activity at home, instead of going to dedicated health centers [133]. Other systems [134] ask the subject to perform a specific action when receiving a specific audio/video signal. By analyzing the subject's reaction time, they detect changes in his/her physical/mental state, which could correspond to an increase in his/her risk of falling. However, this means asking the subject to periodically perform exercises that he/she would not do spontaneously, thus imposing a variation on his/her daily habits. This could be a limitation, since it is known that patient compliance is higher for systems that do not alter the subject's daily routine [21]. To identify potential changes in the patient's health condition it would be more appropriate to detect changes in his/her daily routine, by monitoring the activities that he/she performs spontaneously. Nevertheless, to our knowledge in the literature there are no systems that aim at that goal, despite the current availability of several kinds of PGHD.

To conclude this overview, many of the systems for supporting chronic patients collect PGHD over time, more or less frequently. Recently, CDSSs tend to collect PGHD more and more automatically, although many systems still ask the patient to manually enter data. However, CDSSs that perform advanced analysis on the time series of the collected PGHD are still not widespread.

2.4. Temporal analysis of patient generated (health) data

This section is divided in four sub-sections. In the first one, the challenges posed by the analysis of PGHD are discussed. The second sub-section describes a technique for analyzing time series of data, namely knowledge-based temporal abstraction. The third sub-section describes a software tool for performing temporal abstraction, which will be used in the framework object of this thesis. The fourth sub-section describes an example of application of the JTSA tool.

2.4.1. Challenges in analyzing temporal patient generated (health) data

Analyzing time series of PGHD raises several challenges, some of which are specific to the nature of this kind of data. In the following we will discuss the challenges related to the following six aspects:

- Data quality
- Patient involvement
- Data loss
- Evaluation of the temporal component of the collected measurements
- Integration of data from multiple heterogeneous sources
- Signal characteristics (e.g., sampling frequency)

First, as in any data analysis, attention must be paid to the quality of the collected data before proceeding with any processing. In the clinical setting, there are guidelines to follow to assess and ensure the quality of the measurements collected within the hospital. On the contrary, PGHD is a relatively recent and very heterogeneous type of data, so there are still open issues on how to guarantee their reliability [135]. Compared to measurements gathered in the hospital, the collection of PGHD can lead to more critical issues, since the patient has a central role in the procedure. On the one hand, using sensors that collect measurements automatically facilitates the procedure, potentially limiting the occurrence of human errors. On the other hand, some PGHD sensors must be positioned appropriately to collect reliable measurements, and it must be assumed that the subject is able to do so. For example, activity trackers must be worn on the wrist in a specific position and must not fit loosely, otherwise the device is not able to measure the subject's HR correctly. This obviously has an impact also on the variables that the device computes as a function of HR, i.e., detection of physical activity and sleep. In some cases, the device does not record any data if it is not worn correctly, thus leading to data loss. Often, though, data is recorded anyway, leading to unreliable measurements, which will be difficult to identify.

Furthermore, wearable sensors often need the subject to actively participate in collecting data, although the necessary contribution is limited. Usually, the subject is asked to recharge the device regularly (e.g., on average once a week for activity trackers) and check that the measurements collected by the device is transferred to an external repository, since wearable sensors have a limited memory to store data. In the case of activity trackers, data must be synchronized to the cloud. It is often possible to personalize the system settings to synchronize the tracker automatically and continuously. However, also in that case, the patient must make sure to activate the Bluetooth module on his/her mobile phone, to receive data from the tracker, and an internet connection to transfer the data to the cloud. If the synchronization does not occur timely, the collected measurements might be lost. For example, for the Fitbit tracker, if the measurements are not synchronized to the Fitbit cloud within a week, a lot of data is canceled, including the time series of HR measurements and sleep staging. The same consideration holds for FGM devices. For example, the sensor in the Freestyle Libre System can store approximately 32 measurements, that correspond to eight hours of monitoring. If the patient does not scan the glucose sensor with the reader for more than eight hours, the oldest measurements are deleted from the sensor to allocate memory for storing the most recent values.

Data loss may occur even when the patient is very compliant, or when the sensor does not require an active participation of the subject in collecting the data, as in the case of environmental sensors, which generally transmit the measurements without any manual intervention. However, connectivity issues might prevent the sensors from sending data. If the sensor has no back-up systems for storing values in case of temporary lack of connectivity, the data can be lost.

As for the reliability of the collected values, the accuracy of the measurements depends on the specific device [136, 137]. When designing a CDSS, clinical experts should be involved in the selection of the devices for PGHD collection, to make sure they have an accuracy deemed sufficient for the purpose of the system. For example, as we anticipated in Section 2.2.2, the Fitbit trackers show acceptable accuracy in the detection of sleep, but they are not accurate in identifying the different stages within the sleep occurrence [86]. In our use cases, the clinical experts believed that the detailed information on sleep stages was not necessary, and we used the Fitbit trackers mainly to monitor the patient's sleep quantity. For each variable of interest, a device with suitable accuracy must be selected.

In addition to providing accurate measurements, it is fundamental that the sensors are able to unequivocally associate each value with the instant in which that value occurred. In the following, we will use the term *temporal tag* to refer to the time reference associated with a measurement. Knowing the temporal tag of all the collected measurements is fundamental to study how the variable of interest evolves over time. In case of multiple variables of interest, knowing the temporal tag of all the measurements of all the variables, allows carrying out an integrated analysis. In the clinical context, it is often useful to exploit the information given by a variable to contextualize the information given by others. For example, the diabetologists we have worked with are particularly interested in identifying those patients who experience hypoglycemia during sleep, as it is a life-threatening condition. This is possible by combining information from two sensors, namely an activity tracker and a FGM or CGM device. To combine the two sources of information correctly, it is necessary for the two devices to be synchronized (i.e. that measures with the same temporal tag actually correspond to measurements taken at the same time stamp), so that we can correctly identify the time intervals in which the patient sleeps and the time intervals in which the patient experiences hypoglycemia, and that we can compare them. In case of more than two devices, all the involved sensors must be synchronized.

Although it might seem a trivial assumption, ensuring the reliability of all the temporal tags is challenging for several reasons. First, wearable systems for collecting PGHD often use their own internal clock and require a contribution from the subject to maintain the correct time synchronization. Usually, if the device is not connected to a computer or smartphone for weeks, its clock might stop working correctly. The same issue might occur if the device is not recharged for weeks. In our tests, this occurred both for the Fitbit activity trackers and for the Freestyle Libre System. When the device's clock falls behind, the collected measurements are associated with an incorrect temporal tag, which undermines temporal analytics. Therefore, it is important to make the user aware of this issue, and instruct him to always pay attention to the time reported by the PGHD devices.

Other reasons may cause the time lag of the device's clock. Among all, one of the most frequent is daylight saving time (DST), i.e., shifting the clock one hour ahead during the brightest months of the year to take advantage of sunlight and save electricity, and adjusting the clocks back in the autumn. In the following, we will define *forward DST change* the operation of advancing the clock for DST, and *backward DST change* the opposite one. DST changes often cause issues in the synchronization of all electronic devices, including PGHD sensors. Some devices, like the activity trackers, manage DST changes automatically, using an internet connection, if they are synchronized regularly. Other systems require the subject to handle DST changes manually, and this might rise a challenge for temporal analytics. For example, the Abbott Freestyle Libre system does not apply DST. The system has an internal clock, which must initially be set by the patient, and can be re-set any time. After being set, the clock counts the time starting from the time provided by the subject, without any additional control. In the best case scenario, in which the patient sets the new time exactly in the moment in which DST is planned (i.e., at 2 or 3 in the morning, in Italy), this would lead to errors only in case of backward DST change. In that case, adjusting the clock back by one hour at 3 AM causes the time interval between 2 AM and 3 AM to exist twice. This leads to duplicate measurements, i.e., two measurements having the same temporal tag, but different values. Duplicates are also generated when the patient travels to a country with a different time zone from the country where he/she lives and adjusts the time of the reader back and forth. Duplicate measurements must be filtered out before applying any temporal analytics, and a strategy for filtering should be set up. However, the assumption that the patient will set the new time exactly when the DST is planned is not realistic. The time change on the device is performed either before or after the actual DST takes place, usually depending on when the patient remembers to do it. Therefore, we have to foresee a time interval of unknown duration that may last even months, in which the device's clock is shifted by one hour (forward or backward) with respect to the actual time possibly reported by other devices. A shift of one hour in the temporal tags may be critical when contextualizing the BG profile. For example, it might cause a high BG value to be interpreted as a post-prandial response, when the patient had not yet eaten. Given the consequent lack of synchronization with the Fitbit, this may cause an altered BG value during a workout to be interpreted as an altered BG value at rest. Therefore, it is important to detect those shifts and correct them. The time changes performed manually are logged by the Freestyle Libre system. With appropriate precautions, it is possible to correct the temporal data tags provided by the device, considering both the expected date of DST change, and any DST changes performed manually by the patient. As a further problem, the device does not run any check on the date set by the patient. For example, if the patient sets a future date by mistake, the system does not detect it and all the measurements reported by the device will have a temporal tag that represents a future date. During our tests we detected a frequent mistake in the manual DST change when performed by the patients around midnight. For example, in a DST backward change, one patient adjusted the device's clock back, from 00:30 to 23:30, but forgot to update the date. As a result, the device's clock was erroneously shifted forward by 23 hours, instead of 1 hour backward. All these types of human errors must be identified and corrected by analyzing the Freestyle Libre logs, before analyzing the data. We used the Freestyle libre system as an example, but similar problems can arise with different PGHD sensors. For each specific type of PGHD sensor it is therefore necessary to understand which types of errors may affect the assignment of temporal tags to the collected measurement and define strategies to correct them before integrating all the sources and analyzing the data.

Even assuming that the temporal tags are correct, additional difficulties may arise, including challenges which derive from analyzing data from different sources. One of the most important is the sampling frequency of the time series of measurements, which highly depends on the measured variable. Some parameters are collected more frequently than others. For example, Fitbit activity trackers provide one measurement of HR per minute, while the Freestyle Libre system produces on BG measurement every 15 minutes. The discrepancy in sampling frequencies must be taken into account when variables from multiple sources are analyzed simultaneously.

Furthermore, even considering only one variable at a time, the sampling frequency may not always be constant. For example, when processing data from wearable devices, we have to expect to detect time intervals in which data is missing because the device was not worn by the patient (e.g., recharge time for activity trackers). In addition, the duration of the time interval between two consecutive measurements may slightly vary. For example, from the data collected with the Freestyle Libre, we observed that the time between two consecutive measurements is in general 15 minutes, but it can be also 14 minutes. Since the output measurement of the device is an average over BG measurements taken every minute, the observed behavior is probably due to the algorithm used by the producer to process the data. When analyzing the data, it is therefore important to select a technique that can tolerate this characteristic.

2.4.2. Time series analysis techniques

Depending on the characteristics of the collected data and on the purpose of the analysis, several techniques for analyzing time series (TS) of measurements are available. Such techniques can belong to traditional signal processing, or can operate on qualitative representations of the collected data.

Among signal processing techniques, artificial neural networks (ANNs) are currently one of the most popular [138]. Their main application is to the prediction of the behavior of a TS in a future instant, taking into account the history of the values assumed by the same TS in the past. In the clinical domain, several research efforts have been recently devoted to predicting the behavior of BG, in terms of value and intensity of variation, at a 30-minute horizon [139, 140]. ANNs are networks formed by several layers, each of which contains nodes, called *processing elements*, connected to each other through acyclic links. Each link is associated with a weight, which is part of the set of parameters to be estimated during the model construction phase. The output of the model is then computed by applying a mathematical formula, which depends on the input (i.e., the observed TS) and on the specific structure of the network, including the disposition of the nodes and the values of their weights [138]. An advantage of ANNs is that they are data-driven, i.e., they do not require any prior knowledge of the characteristics of the TS. One of the main drawbacks of such technique is that it requires a significant amount of data to train the model, being unaware of the context of the analysis, i.e., the specific clinical domain in which the TS was collected [141, 142]. Moreover, it is not trivial to associate a clinical interpretation to the result provided by the analysis.

Signal processing techniques have specific requirements and cannot be applied to any type of data. For example, in general they cannot be applied to data showing an irregular sampling frequency. Since the irregular sampling frequency is one of the main characteristics of PGHD, in our framework we did not include signal processing techniques. For analyzing

PGHD, we focused on knowledge-based Temporal Abstraction (KBTA). KBTA can be defined as an intelligent and context-sensitive interpretation of temporal data and its visualization in a synthetic representation [143–147]. KBTA allows processing TS of measurements, often referred to as *raw data*, to obtain high-level views of such data. In particular, by analyzing the temporal profile of the considered variable, KBTA allows converting the TS of measurements into a qualitative and interval-based representation, in which each time interval (TI) has a label that summarizes the qualitative behavior of the variable in that interval. The set of admissible labels depends on the domain and purpose of the analysis. Figure 2**Error! Reference source not found.** shows an example of temporal abstraction applied to the diabetes domain. In this case, the raw data (on the top) consist of the TS of the patient's BG measurements collected by an FGM device over a selected time interval, T. The bottom part of the figure shows the result of applying KBTA for extracting increase, decrease and stationarity intervals in the TS. In the final interval-based representation (on the bottom, T is split in 9 sub-intervals according to the observed BG trend. The admissible labels for each TI are *Increasing*, *Stationary*, and *Decreasing*.

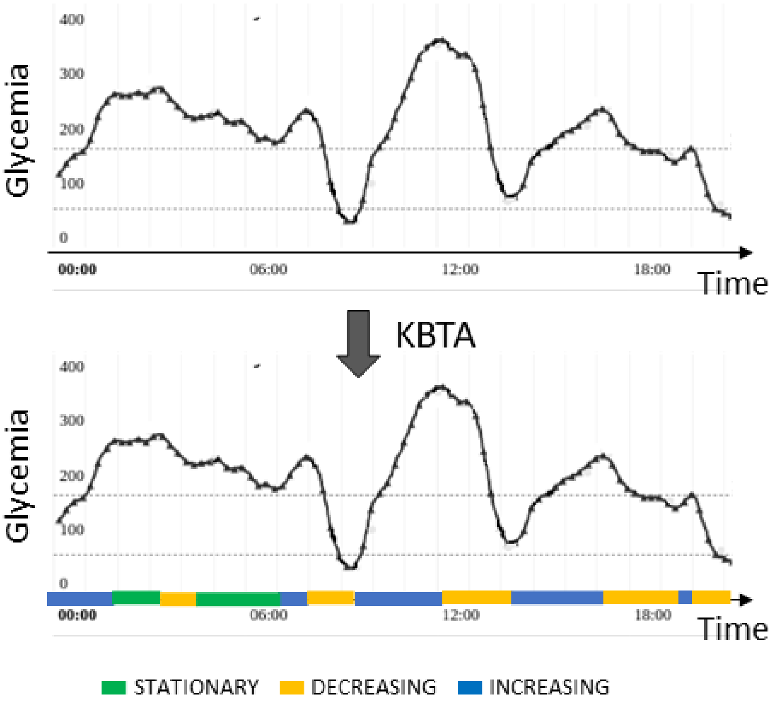


Figure 2: Example of knowledge-based temporal abstraction applied to extract trends in BG data.

In the following, we will refer to the behavior we want to detect in the temporal data as *pattern*. It is essential to define together with clinical experts the patterns to be extracted from the TS of interest. Some patterns are simple, such as the increasing and decreasing trends considered in the previous example. Other patterns represent more complex phenomena. For example, to optimize the patient's therapy, diabetologists need to check the subject's BG profile for the occurrence of the *dawn effect*, that corresponds to normal BG values during sleep, followed by hyperglycemia before breakfast. Complex patterns may also be multivariate. For example, diabetologists are interested in identifying when an increase in the subject's HR profile precedes an episode of hypoglycemia.

To explicitly model the knowledge necessary to define the patterns unambiguously, the experts' contribution is essential. For example, if we need to identify all the time intervals in which the patient's BG value increases, several questions need to be answered: what is the minimum rate of increase in the BG value that is considered significant from the clinical point of view? If the trend lasts for a very limited time, is it informative to report it as a pattern? If multiple time intervals show the desired trend, should we aggregate them as a single occurrence of the pattern, or as distinct occurrences? Answering these questions allows including the domain knowledge necessary to properly exploit the KBTA framework. A richer set of questions arises in the case of complex patterns, since they are composed of a combination of behaviors, linked by very specific temporal relationships. For example, for the previously described heterogeneous pattern, how long can it take between the increase in HR and the occurrence of hypoglycemia to consider them as related episodes? Is the increase in HR supposed to end before the occurrence of hypoglycemia? Which time interval is considered critical, only the TI characterized by hypoglycemia or also the TI showing the HR increase? KBTA techniques must be personalized, taking into consideration these aspects. In the following section we will present a software tool that allows performing user-defined temporal data analyses based on KBTA.

2.4.3. Java Time Series Abstractor: a tool for temporal abstraction

In the proposed framework, to perform KBTA on PGHD we use Java Time Series Abstractor (JTSA), a framework that has recently been developed by a team of researchers from our department at the University of Pavia [148]. In general, KBTA is performed by applying a sequence of algorithms that abstract the input data. JTSA formalizes several algorithms for KBTA, that can be personalized by tuning their parameters. JTSA is convenient since it is modular, i.e., multiple algorithms can be combined in workflows to detect any user-defined pattern. Thus, it allows a context-specific summarization of the collected PGHD data. JTSA methods are available within a JAR, which can be easily integrated into any third-party application.

JTSA operates on two basic temporal primitives, namely the *event* and the *episode*. An event is a tuple composed of one-time stamp and the value assumed by the variable of interest at that time. An episode is composed of one-time interval, defined by a start date and an end date, and the value assumed by the variable in that interval, which is usually a qualitative label. Given their qualitative nature, TS of episodes have a higher level of abstraction than TS of events. In the following, we will refer to TS of episodes as *A-TS*.

The ontology that describes the set of algorithms provided by JTSA is shown in Figure 3.

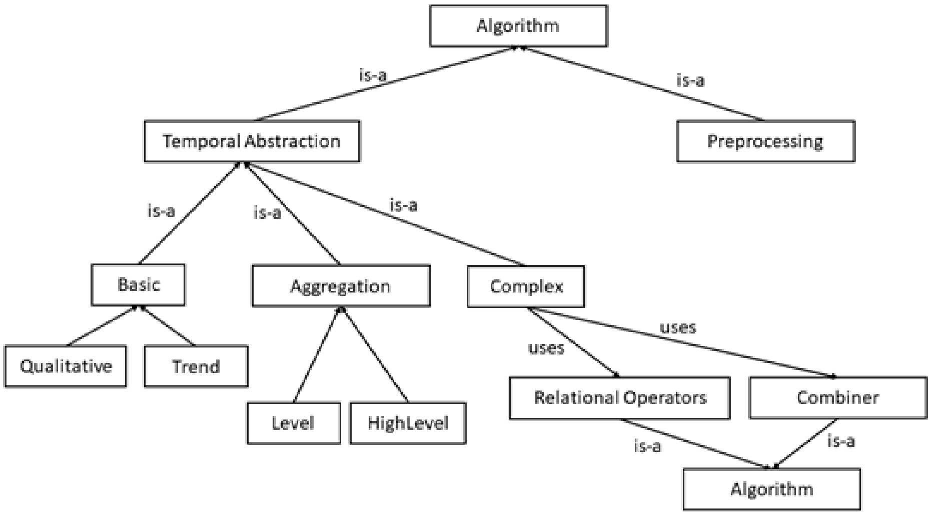
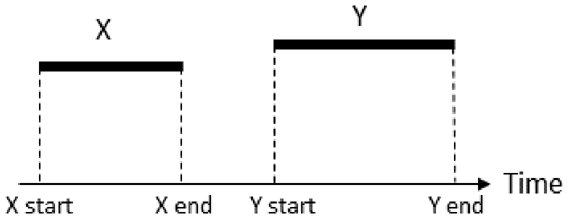
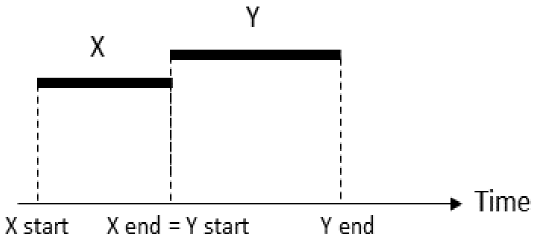
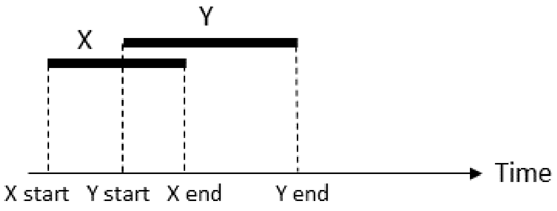
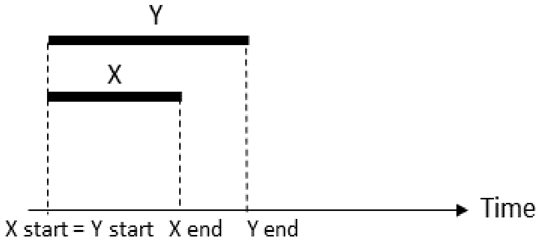
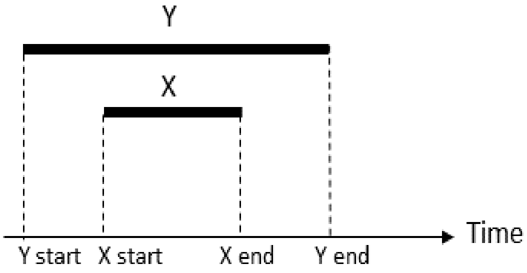
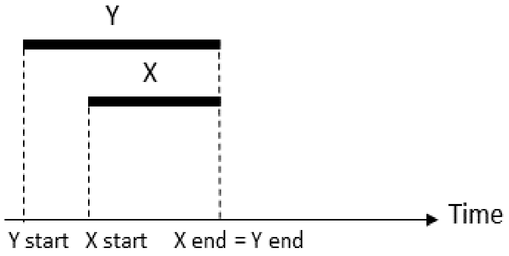
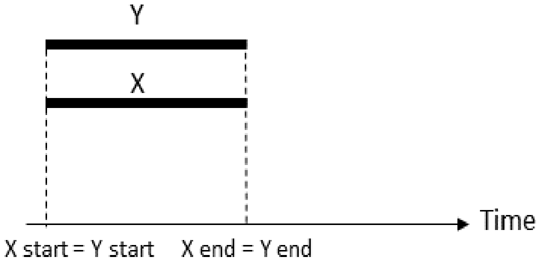


Figure 3: Ontology of the algorithms provided by JTSA. Source: [148]

Pre-processing algorithms allow performing operations such as normalization, filtering and polynomial interpolation of data. The algorithms for KBTA can be classified into three different groups, namely *Basic*, *Aggregation* and *Complex*, which differ both in the type of input data and in the type of output. Basic algorithms receive a TS of events as input and return an A-TS as output. The Basic algorithms are further divided into two groups: *Qualitative* methods discretize the input variable using a set of thresholds provided by the user; *Trend* methods detect increase, decrease or stationarity in the profile of the input variable. *Aggregation* methods receive an A-TS as input, and aggregate subsequent episodes having the same label into a single episode, based on parameters defined by the user. Thus, the result of the aggregation is an A-TS. Two parameters need to be specified to apply an aggregation algorithm: the *minLen*, i.e., the minimum length that an aggregated episode must have to be included in the output series, and the *gap*, i.e., maximum distance between two consecutive episodes to be aggregated in a single episode. *Complex* algorithms detect complex abstractions, created combining Basic abstractions or complex abstractions on the basis of a set of temporal operators, namely Allen's relational operators [149], which define the different temporal relations that can occur between two time intervals. Table 1 lists Allen’s operators, each one complemented with a textual description and a graphic representation.

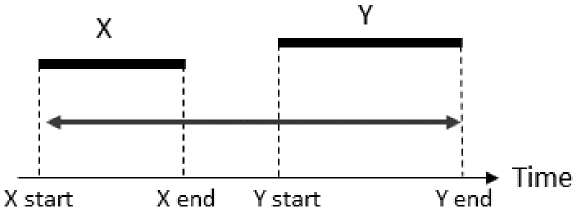
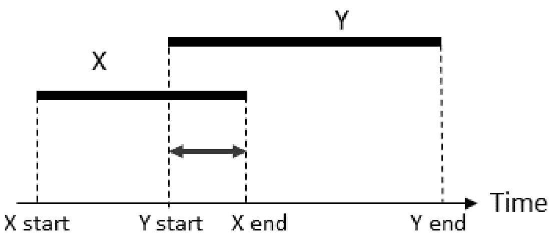
***Table 1**:* Allen's temporal operators

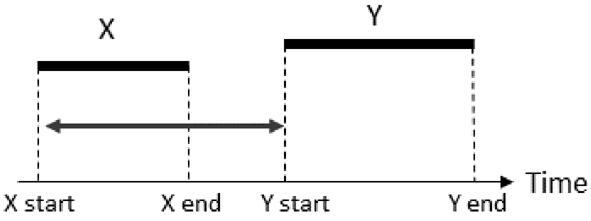
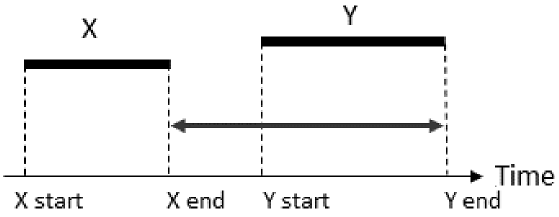
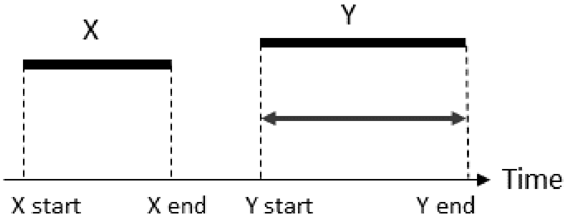
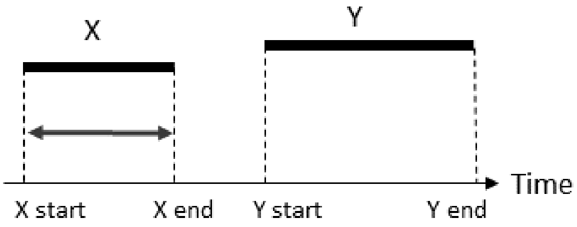
OPERATOR	DESCRIPTION	REPRESENTATION
BEFORE	X ends before the start of Y	
MEETS	X ends when Y starts	
OVERLAPS	X begins before Y AND Y begins before the end of X AND X ends before the end of Y	
STARTS	X starts at the same time as Y AND X ends before the end of Y	
DURING	Y starts before X AND X ends before the end of Y	

FINISHES	Y starts X AND X ends at the same time as Y	
EQUALS	X starts at the same time as Y AND X ends at the same time as Y	

Complex algorithms accept two A-TS, *S1* and *S2*, as input and try to apply the selected temporal relation to all possible <episode of S1, episode of S2> pairs. When a pair verifies that temporal relation, an output episode is created, and inserted in the A-TS returned by the algorithm. The time interval of the new episode is built based on the time intervals of the two episodes in the considered pair, through operators known as *combiners*. Table 2 lists the combiners available in JTSA, each one complemented with a textual description and a graphic representation.

Table 2: Combiners provided by JTSA

COMBINER	DESCRIPTION	REPRESENTATION
UNION	The resulting time interval includes both X and Y, in addition to the possible time interval between them.	
INTERSECTION	Time interval that X and Y share.	

GAP BETWEEN STARTS	Time interval between the beginning of X and the beginning of Y.	
GAP BETWEEN ENDS	Time interval between the end of X and the end of Y.	
LONGEST	The longest time interval among X and Y. In the provided example, the combiner returns Y.	
SHORTEST	The shortest time interval among X and Y. In the provided example, the combiner returns X.	

To detect a user-defined pattern, the described JTSA algorithms can be combined in a *workflow*, a document that formalizes the steps to be followed for abstracting one or more time series into the desired pattern. A workflow is composed of several sections, known as *blocks*, which may be in turn composed of sub-sections, known as *steps*. Each block can use one or more JTSA algorithms. Based on the type of input and on the content, a block may be either a *Pipeline Block* or a *Complex Block*. Pipelines blocks consist of a sequence of steps that use a Basic algorithm or an Aggregation algorithm and require only one time series, either TS of events or A-TS, as input. Complex Blocks, on the other hand, work on two A-TS as input and use a Complex algorithm to combine them through one Allen's operator and one combiner. Workflows are XML documents based on a JTSA-specific XML schema, which are interpreted and executed by the JTSA engine.

To clarify how the JTSA framework works, in the following sub-section we will describe an example of workflow related to the diabetes domain.

2.4.4. Example of JTSA workflow

The workflow described in this section processes the patient's BG data and detects the occurrence of the previously mentioned dawn effect, i.e., normal BG value during sleep, followed by hyperglycemia on awakening. The XML file formalizing the workflow is shown in Figure 4. A schema summarizing the workflow is shown in Figure 5. The workflow consists of two Pipeline blocks and one Complex block. Each block has a unique identifier, specified by the *id* attribute. The Pipeline block with the identifier *Normal_Night* extracts the time intervals in which the subject's nocturnal BG value is in the normal range [Th_{HYPO}; Th_{HYPER}], where Th_{HYPO} and Th_{HYPER} are two patient-specific thresholds, set by the diabetologist. In this example, we will consider [70 ; 180] as the normal glycemic range. The Pipeline block with the identifier *Hyper_Morning* extracts the time intervals in which the subject's BG value is higher than the Th_{HYPER} threshold. The Complex block combines the sub-results provided the two Pipeline blocks, to detect the time intervals in which the Dawn Effect occurs. The following paragraphs provide a detailed description of the three blocks, complemented with the list of the used algorithms, and the parameters set to customize them.

```
<workflow>
        <block>
          <pipeline id="Normal_Night" type="pipeline"
          dataType="file_BASIC_TS_CSV_READER" dataIn="BG_SLEEP.csv">
                <step order="1" title="Normal_Night"
                        parameters="properties/qualitativeG.properties"
                        type="BASIC" subtype="BASIC_QUALITATIVE">
                </step>
                <step order="2" title="Aggregation_Normal"
                        parameters="properties/HighLevelAgg_N.properties
                        type="AGGREGATION" subtype="AGGREGATION_HIGHLEVEL">
                </step>
                </pipeline>
        </block>
        <block>
          <pipeline id="Hyper_Morning" type="pipeline"
          dataType="file_BASIC_TS_CSV_READER" dataIn="BG.csv">
                <step order="1" title="Hyper_Morning"
                        parameters="properties/qualitativeG.properties"
                        type="BASIC" subtype="BASIC_QUALITATIVE">
                </step>
                <step order="2" title="Aggregation_Hyper"
                        parameters="properties/HighLevelAgg_H.properties
                        type="AGGREGATION" subtype="AGGREGATION_HIGHLEVEL">
                </step>
           </pipeline>
        </block>
        <block>
          <complex id="DawnEffect" type="complex"
                dataType1="block" dataIn1="Normal_Night"
                dataType2="block" dataIn2="Hyper_Morning"
                rel_operator="BEFORE" combiner="Union"
                parameters="properties/BEFORE.properties"
                label="DawnEffect">
          </complex>
        </block>
</workflow>
```

***Figure 4:** JTSA workflow to detect the Dawn Effect*

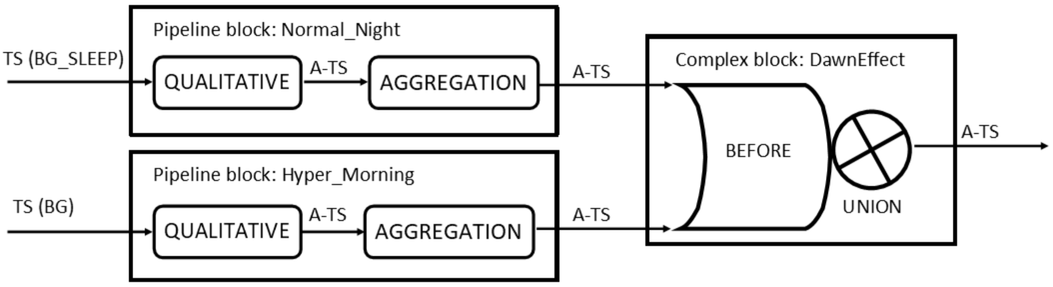


***Figure 5**: Schema summarizing the JTSA workflow to detect the Dawn Effect*

The *Normal_Night* block is characterized by three attributes, excluding the attribute that specifies the unique identifier. The *type* attribute specifies the nature of the block, which in this case is *pipeline*. Two attributes, *dataIn* and *dataType* are used to specify the data that the block must receive as input. A Basic block can read the input data from a text file, or it can receive them from the application that integrates JTSA. Since in this example the attribute *dataType* has value *file_BASIC_TS_CSV_READER*, the block will read its input data from the file specified by the *dataIn* attribute. In case the file, BG_SLEEP.csv contains all the BG values related to the subject's sleep. The block is composed of two steps whose order of execution is established by the value of the *order* attribute. Each step has a *title* attribute, which provides a qualitative description of the expected result of the operation and does not represent an identifier. The first step uses a Basic Qualitative algorithm, as specified by the *type* and *subtype* attributes. As anticipated, this algorithm discretizes the input values using a set of thresholds, which are set in a configuration file, whose path is specified in the value of the *parameters* attribute. The configuration file has two lines, the first line specifies the set of thresholds, the second line specifies the set of labels to be assigned to the values according to the defined thresholds.

```
th=70, 180
label=hypoglycemia,normal,hyperglycemia
```

Consequently, every BG value lower than 70 will be labeled as *hypoglycemia*; every value greater than 180 will be labeled as *hyperglycemia*; each value in the range between 70 and 180 will be labeled as *normal*.

The second step aggregates the episodes that have the same label, according to the parameters specified in the configuration file, *properties/HighLevelAgg_N.properties*. In this case the file includes five parameters, as shown in the following lines.

```
levels=normal
label=Normal
gap=60
minLen=60
granularity=MINUTES
```

The *levels* parameter determines which categories to consider in the construction of the A-TS to be returned. Among the episodes produced by the first step, only episodes with the *normal* label are considered by the second step. The gap is the maximum distance between two episodes with the *normal* label to be aggregated. In particular, the algorithm considers all the pairs $<E_1, E_2>$, where each E_i is an episode of the A-TS produced in the previous step, and E_1 starts before E_2. If the temporal distance between the end of E_1 and the beginning of E_2 is more than 60 minutes, the two episodes are combined into a single episode, starting when E_1 starts and ending when

E_2 ends. The *minLen* parameter sets the minimum duration of an episode to be considered relevant. In this case, only episodes longer than 60 minutes will be included in the A-TS returned by the step. The episodes in the output A-TS have the label specified by the *label* parameter, i.e. *Normal*, in this example. The *granularity* parameter specifies the unit of time in which the *gap* and *minLen* variables are expressed. In this case they are expressed in minutes.

The *Hyper_Morning* block works similarly to the *Normal_Night* block, but receives in input a different time series, *BG*, that is not limited to the BG values measured during sleep. The block consists of two steps. Despite having a different title, step 1 produces the same result as step 1 of the *Normal_Night* block, i.e., a series of episodes, each one having one among the three possible labels, *hypoglycemia*, *normal*, and *hyperglycemia*. Step 2 aggregates episodes having the *hyperglycemia* label. While step 1 uses the same configuration file described for step 1 of the *Normal_Night* block, step 2 uses a separate configuration file, shown below.

```
levels=hyperglycemia
label=Hyperglycemia
gap=20
minLen=60
granularity=MINUTES
```

Thus, step 2 aggregates episodes having the *hyperglycemia* label if the gap between them does not exceed 20 minutes. After the aggregation, episodes with duration longer than 60 minutes are given the *Hyperglycemia* label and are included in the A-TS returned by the block.

The complex block receives as input the two A-TS produced by the pipeline blocks, as specified by the *dataType* and *dataIn* parameters. Specifically, *dataIn1* specifies that the result produced by the block with identifier Normal_Night is used as the first input; according to *dataIn2* the second input is the A-TS produced by the *Hyper_Morning* block. The order assigned to the two inputs is important and must be set carefully, because it determines how the selected Allen's operator will be applied to the two series. The selected Allen's operator is specified by the *rel_operator* parameter. In this case, the block applies the BEFORE operator, which was described in Table 1. The BEFORE operator is personalized according to a set of properties set in the configuration file, *properties/BEFORE.properties* having the following content.

```
ls=540
rs=240
gap=60
granularity=MINUTES
```

If we define *A-TS1* and *A-TS2* the two input A-TSs, and <*x* , *y*> the pair consisting of one episode of A_TS1 (x) and one episode of A-TS2 (y), the parameters in the configuration file are used to constrain the desired temporal relationship between x and y. Their meaning is summarized in Figure 6.

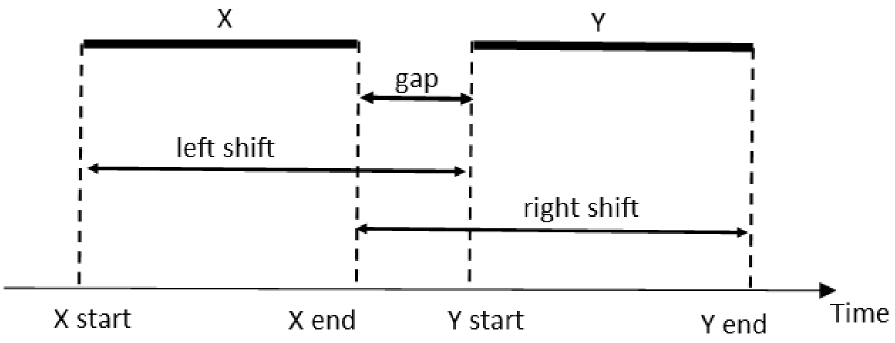


Figure 6: Parameters for personalizing Allen's temporal operators

In particular, *ls* is the acronym for *left shift* and identifies the maximum distance between the beginning of x and the beginning of y. The *right shift* (*rs*) defines the maximum distance between the end of x and the end of y. As in the previous cases, the gap represents the maximum distance between the end of x and the beginning of y. The block applies the BEFORE operator to all the possible <x , y> pairs. For each pair that verifies the BEFORE relation, the two episodes, x and y, are combined using the *Union* combiner (Table 1) as specified by the *combiner* parameter. All the resulting episodes are assigned the *DawnEffect* label and included in the A-TS returned by the workflow. As previously discussed, the *granularity* parameter specifies the unit of time in which the temporal variables are expressed.

Figure 7 shows a detail of the output of the described JTSA workflow for the selected patient.

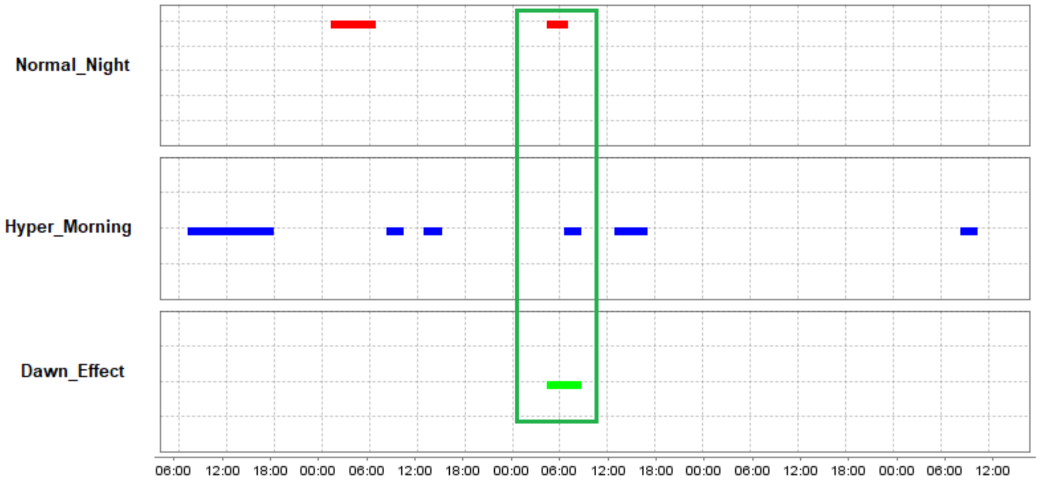


Figure 7: Example of occurrence of the Dawn effect

In particular, Figure 7 is divided in three vertical sections, each one representing the output of one block of the described workflow. Time intervals in which the subject's BG value is normal during the night are shown in red. Time intervals that correspond to hyperglycemic episodes are shown in blue. The identified occurrence of the dawn effect is shown in green.

3. The proposed framework

As anticipated in Section 2.4, when clinicians examine time series of measurements, they usually search for specific behaviors, or *patterns*, in the collected data. Depending on the clinical problem, the valuable information may be the *presence* of a specific behavior in the data, its *absence*, or its *modification* over time in terms of a specific set of characteristics (e.g., duration, frequency, or intensity). Such information often supports the clinician in evaluating the patient's health status and personalizing the care process.

In this chapter, we propose a framework for building CDSSs that perform pattern detection on temporal series of PGHD and provide decision support accordingly. The proposed framework collects considerations that do not depend on specific decision problems; thus, it may be applied to any clinical context in which extracting information from longitudinal PGHD is needed. We identified three main key steps that the CDSS must be able to support:

1. The download and integration of PGHD
2. The pattern detection
3. The provision of pattern-based decision support

We then identified a minimum set of components that need to be integrated into a CDSS to allow performing the above-mentioned key steps. The proposed set includes five main components, namely:

- the data integration module
- the PGHD repository
- the knowledge base
- the inference engine
- the graphical user interface.

The relationships between the different components are shown in Figure 8, in which the dashed box delimits the CDSS.

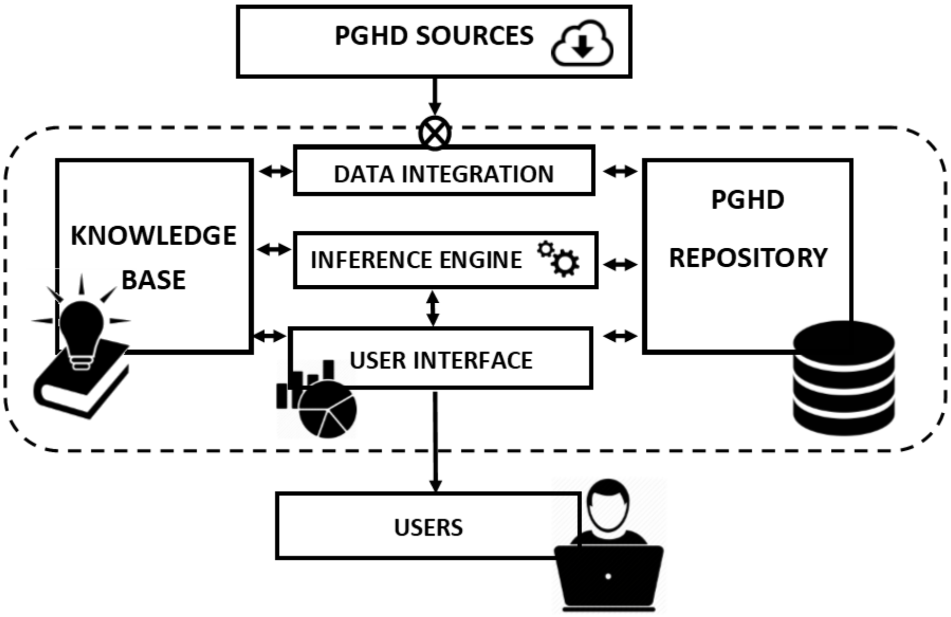


***Figure 8**: Components needed in a CDSS that analyzes time series of PGHD*

In particular, we use the term *data integration module* to indicate an interface between the system and external PGHD sources. Its functionality is to periodically download PGHD data, and to prepare it for further analyses. For example, the data integration module should convert the downloaded time series into a format that is suitable for the system. It should also combine the information obtained from multiple sources, when the integration can provide insights that are meaningful for the decision problem.

A dedicated PGHD repository should store the PGHD prepared by the data integration module, and possibly persist the results of the analyses.

The knowledge base (KB) should define all the concepts related to the domain in which the CDSS operates, the relationships between them, the patterns of interest for the specific decision problem, and how they need to be used in order to provide decision support.

An inference engine should act upon such formalization of knowledge to actually deliver decision support to the users.

A graphical interface should be integrated, to allow the system to communicate with the user. Most system communications are outgoing, to present the results of the analysis and possible warnings, but often graphical interfaces allow users to provide data that are necessary for the analysis.

Figure 9 details the general architecture that was presented in Figure 8, listing the technical sub-components required for a CDSS that exploits JTSA for analyzing time series of PGHD. The role of each component and sub-components will be discussed in the following of this chapter.

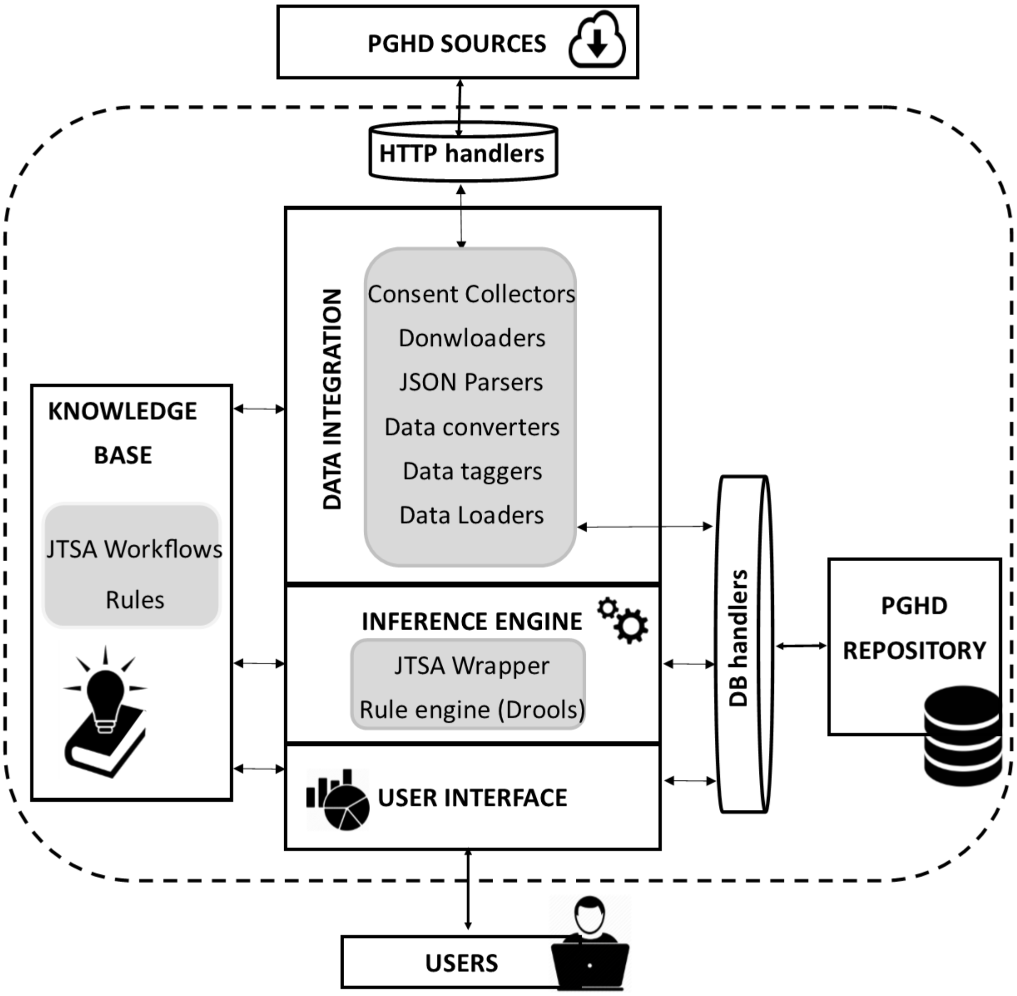


Figure 9: System architecture

For each main component, a sub-section will describe the reasons why it is necessary in a CDSS for analyzing time series of PGHD and its desired characteristics, regardless of the implementation technologies. In this work, we focused on knowledge-based temporal abstraction (KBTA) for performing pattern detection. Thus, the desired characteristics of the proposed components were defined assuming that KBTA would be performed.

3.1. The data integration module

We have identified two main needs that this component should address

- the need to download data from multiple sources
- the need to give a meaning to the downloaded data

To address these needs, we suggest to include six types of components in the data integration module (Figure 10).

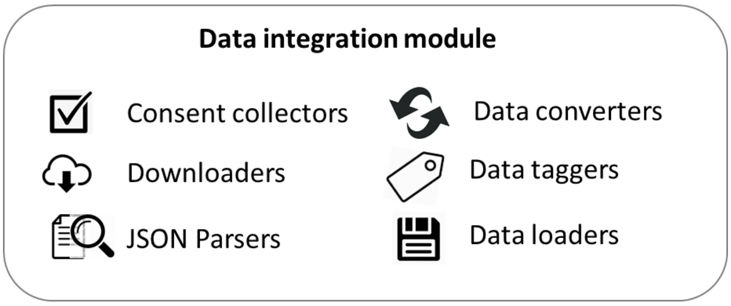


Figure 10: *Components of the data integration module*

The two needs and the components that are necessary to fulfill them will be discussed in the following subsections.

3.1.1. The need to download data from multiple sources

The data integration module should contain several modules for downloading PGHD data, possibly one for each type of PGHD source, as each source might have specific data download modes. These components will be referred to as *downloaders*. Most commonly, downloaders request the data sending HTTP requests to the dedicated server provided by the PGHD source, which queries the cloud repository and supplies the retrieved information. In most cases, the collected measurements are not available immediately, but only after the PGHD has been synchronized with the cloud repository. Thus, the downloader must activate only when data become available, and when the latter have not yet been downloaded. Our strategy for synchronizing the download will be discussed in a following sub-section.

Since usually the PGHD source provides the requested data in the JSON format, the data integration module should also include *json parsers*, i.e., components that parse the JSON file to extract the data.

Furthermore, PGHD are sensitive data, since they may reveal relevant information about the health of the individual who generated them. Therefore, they must be protected from any access, use or distribution that is not authorized by the subject himself/herself [150–152]. As discussed in Section 2.2.2, commercial PGHD sensors often store the collected data in a proprietary cloud repository, whose content can be accessed through dedicated servers provided by the PGHD source. By default, such servers provide the collected data only to the person who generated it, to preserve his/her privacy. However, the monitored subject can grant third-party applications the permission to download his/her data. Thus, the data integration module needs to contain components for collecting the subject's consent to download his/her own data. In this thesis, these components will be referred to as *consent collectors*. The method for obtaining consent may vary depending on the PGHD source. Consent may be valid for a limited time and may be withdrawn from the monitored subject at any time. Therefore, the consent collectors must be able to monitor the consent status over time, and ask the subject to renew it, in case it has expired.

Solution adopted in the framework to perform the download

In our applications we developed a component, to which we will refer to as *HTTP handler*, dedicated to sending and processing HTTP requests. Our HTTP handler exploits *httpclient*, a library developed by Apache that provides methods for formalizing and sending HTTP requests, and for inspecting the answer [160].

The answer often contains the requested data in the JSON format. In particular, each JSON document is composed of a set of elements, organized in a hierarchical structure. For each type of PGHD, the hierarchical structure is known, since it is usually provided by the source documentation. Based on the hierarchical structure, the *parsers* exploit the methods provided by *json-simple* [153], a library developed by Apache to retrieve the content of each element (i.e., the PGHD) in the JSON document.

Solution adopted in the framework to synchronize the download

As anticipated, the downloader must activate only when the data is available and not yet downloaded, i.e., stored in our dedicated PGHD repository. Checking this condition is not trivial and must be done separately for each type of data. It also requires monitoring a specific set of events (e.g., start of the monitoring session, synchronization of the PGHD device with the cloud) and the temporal relations between them.

The events of interest and their temporal relations will be described in the following. To facilitate reading the following description, Figure 11 shows three timelines, namely (from top to bottom):

- the sequence of the events related to the PGHD monitoring device
- the sequence of the events related to the cloud in which the PGHD device stores the data
- the sequence of the events related to the CDSS repository.

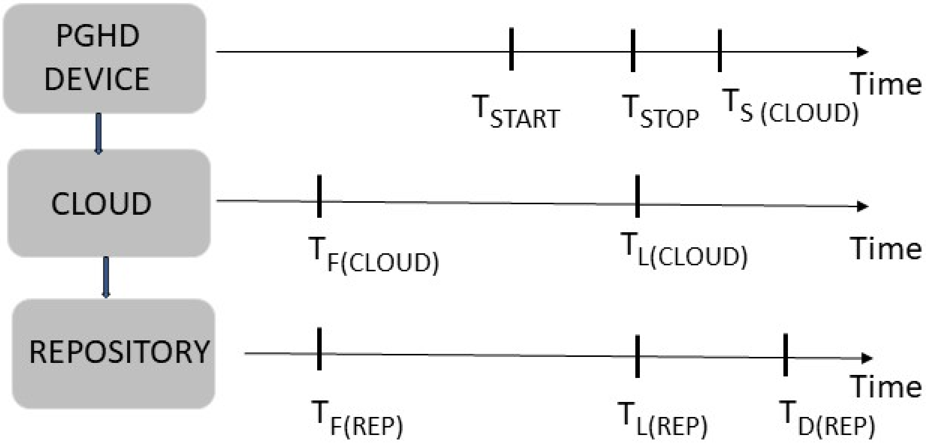


Figure 11*: Time variables of interest for downloading PGHD*

The first variable to monitor is the time interval that includes the temporal tags of the data that are currently on the PGHD device, which we will refer to as [T_{START}; T_{STOP}]. Since PGHD devices often have limited memory, this range contains the most recent data and not the complete monitoring history. $T_{S(CLOUD)}$ represents the date and time in which the PGHD device was synchronized with the corresponding PGHD cloud. It is fundamental to monitor $T_{S(CLOUD)}$, to check that synchronization occurs regularly. The CDSS should generate an alarm if the discrepancy between $T_{S(CLOUD)}$ and the current date is greater than a selected threshold duration. The measurements that were collected before $T_{S(CLOUD)}$ can be downloaded from the cloud. For each type of data, $T_{F(CLOUD)}$ represents the date and time of the oldest value stored in the cloud, which ideally is also the first measurement (of that type) collected by the PGHD device. When the PGHD device has been running for some time, $T_{F(CLOUD)} < T_{START}$. $T_{L(CLOUD)}$ represents the date and time of the latest value stored in the cloud. This value coincides with the last data measured by the PGHD device only at the time of synchronization. When the PGHD device contains data that are waiting to be synchronized, $T_{L(CLOUD)} < T_{STOP}$. It is also necessary to memorize the date and time of every attempt to download data made by the downloaders. In particular, $T_{D(REP)}$ represents the date and time of the last download attempt. If $T_{D(REP)} < T_{L(CLOUD)}$ the downloader must activate. For an early detection of possible issues in the downloading process, it is necessary to monitor that $|| T_D - T_{CURRENT}|| < TH$, where $T_{CURRENT}$ represents the current datetime, TH represents a selected threshold duration. In three cases $|| T_D - T_{CURRENT}|| < TH$ is not verified. These three cases will be briefly discussed in the following.

CASE 1

Most probably, the PGHD device may have not been synchronized with the cloud for too long and also $T_{S(CLOUD)}$ and $T_{L(CLOUD)}$ might not be updated.

CASE 2

As a second possible explanation, the device may be synchronized but may have not collected any measurements (Figure *12*). In the case of an environmental sensor, intended to emit data almost continuously, this indicates that the sensor is malfunctioning. In case of wearable sensors, the absence of measurements does not necessary indicate malfunctioning. For example, the device may be charging, thus not worn by the patient.

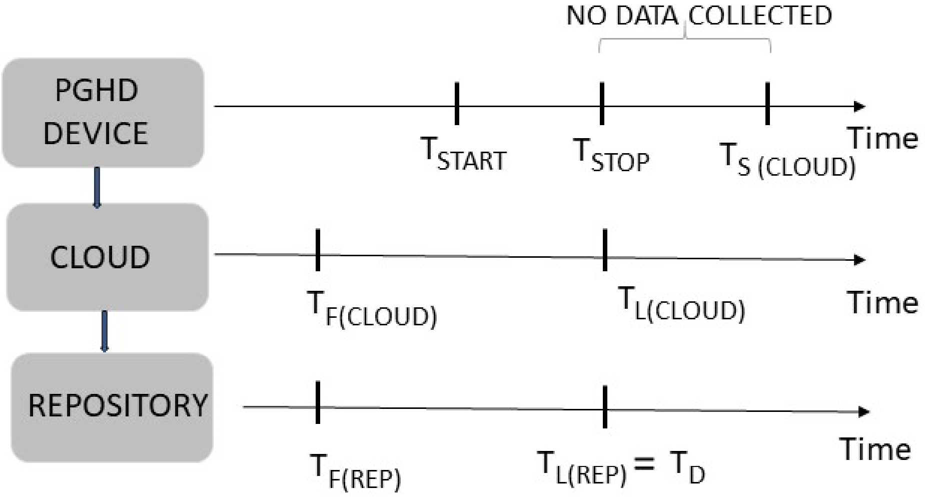


Figure 12: Time variables of interest for downloading PGHD

CASE 3

In the third case, the downloader failed activating. It is necessary to check that all the data downloaded from the cloud is transferred to the repository. Thus, for each type of data, $T_{F(REP)}$ represents the date and time of the oldest value downloaded into the CDSS repository, while $T_{L(REP)}$ specifies the date and time of the latest value in the repository. Ideally, $T_{F(REP)}$ is equal to $T_{F(CLOUD)}$, while and $T_{L(REP)}$ is equal to $T_{L(CLOUD)}$. Otherwise, the downloader must activate. However, the method discussed so far involves sending requests for data too frequently if the PGHD device has been set up to synchronize with the cloud continuously. An improper use of data requests may overload the system, slowing it down. In addition, the PGHD server often provides $T_{S(CLOUD)}$, not $T_{L(CLOUD)}$, making it difficult to determine when to activate the downloader. Thus, more often each downloader has its own predefined activation frequency, which depends on the frequency with which the data are produced and on the use case for which the system is designed. The checks on the dates previously described are still necessary to assess what are the time intervals to consider in data requests, but they are carried at a fixed frequency.

Solution adopted in the framework to manage patient consent

As anticipated, the communication with the PGHD servers is usually based on the HTTP (Hypertext Transfer Protocol) protocol. In general, an additional protocol needs to be applied to ensure that the PGHD servers provide the patient's PGHD only to applications that have been authorized by the subject himself/herself. One of the most known protocols for managing the patient's consent is the *OAuth 2.0 Authorization Framework* [154], published by the Internet Engineering Task Force (IETF), a community that develops Internet standards [155]. According to the OAuth 2.0 authorization framework, the patient is the *resource owner*, since he/she owns the collected PGHD. The OAuth 2.0 protocol allows the *resource owner* to grant third-party application the permission to access the data server on his/her own behalf. Some producers of sensors for collecting PGHD adhere to the OAuth 2.0 authorization framework and provide protocols for managing the resource owner's consent. This facilitates the development of applications for downloading analyzing the collected PGHD. For example, the well-known companies that produce activity trackers also provide documentation on how to apply the authorization protocol to download the collected PGHD. In our applications, consent collectors are implemented in Java, as all the components in the data integration module, and comply to the Oath protocol provided by the PGHD sources to explicitly obtain the patient's consent to data download.

In the following, to provide an example of application of the OAuth 2.0 protocol, we list three links to the documentation related to authorization provided by popular brands of activity trackers.

Fitbit
https://dev.fitbit.com/build/reference/web-api/oauth2/

Polar
https://www.polar.com/accesslink-api/#authentication

Garmin
https://developer.garmin.com/connect-iq/programmers-guide/communication/

3.1.2. The need to give meaning to the downloaded data

The content returned by each PGHD source is not informative until it has been given a meaning that is understandable for the CDSS. In particular, for each content retrieved from the JSON document returned by the source, three aspects must be clarified unequivocally:

- the type of measurement
- the value
- the exact time the measurement refers to

The first two items are highly related. Most commonly, the same variable is formalized differently from the different PGHD sources. Even more commonly, the formalization of the same variable in the CDSS differs from the formalization used by the PGHD sources. For example, the same variable can be expressed in different units of measurement. Generally, the format used by the PGHD source is provided in the source documentation. In our framework, we develop specific components (*data converters*) that process the PGHD in the JSON document, and translate it, so that it complies to the format known to the CDSS.

As regards the third item, *temporal information normalization* is a key-step in the data integration process [156]. It consists in standardizing the temporal component of the data, i.e., the *temporal tag* of each measurement (see Section 2.4.1), to allow integrated data analysis. In fact, PGHD sources might express temporal tags using different notations, that can also differ from the notation used by the CDSS. Also in this case, the temporal notation used by the PGHD source is known from the source documentation. Thus, data converters can uniform all the temporal tags to the specific notation used by the CDSS.

Once the downloaded measurements have a clear meaning and a clear temporal reference, additional meaning can be assigned by combining the information collected from multiple kinds of PGHD sources. Often, a specific variable (*dependent* variable) may be contextualized, or influenced by, another variable (*independent* variable). These two variables may be measured by the same device, or by different kinds of PGHD monitoring devices. For example, to support diabetologists in analyzing the patient's blood glucose profile, it is useful to determine if a given BG value (dependent variable) was recorded during the subject's sleep (independent variable), or during a workout (independent variable), since the subject's metabolic response depends on sleep and physical activity. Since contextualizing some kinds of PGHD may facilitate understanding the subject's health condition, in our framework we suggest a strategy to enrich the definition of a selected dependent variables according to the value of its independent variables. The following subsection describes that strategy and provides an example of its application in our tool for supporting diabetes care.

Solution adopted in the framework to contextualize dependent variables

The simplest strategy to enrich the definition of a variable is to add a further dimension. Thus, we decided to enrich the dependent variables of interest by assigning them a *context label*, whose value summarizes the behavior of the independent variables at the same time. In general, assigning a *context label* means relating a variable of interest, collected from one PGHD source, with another variable having a different nature, that might help to contextualize the variable of interest. For example, as anticipated, for each BG value we want to associate a context label, whose value indicates whether that value was collected during the subject's sleep, during a workout, or during regular daily activities. In our framework, in the data integration module we include dedicated components, *data taggers*, that associate the context labels to the collected data.

As anticipated, the independent variable may be collected from the same PGHD source, or from a different data source. This entails the need to define rules on how to handle those kinds of data that need to be contextualized based on other PGHD. The data taggers must take into consideration that the two kinds of data are not always downloaded at the same time, although they need to be interconnected. Rules should define how to proceed when the data to be contextualized (dependent variable) is available but the contextualizing data (independent variable) is missing, thus leading to a missing *context label*. This may occur often when the two kinds of data are provided by different PGHD sources. In that case, the *context label* may be missing temporarily, waiting for the independent variable to be downloaded, but it may also be permanently missing, if the contextualizing data was not collected at all. Distinguishing these two situations is fundamental to assess whether it is necessary to repeat the tagging attempt. The distinction can be made by checking the above described dates (i.e., $T_{F(CLOUD)}$, $T_{L(CLOUD),}$ $T_{S(CLOUD)}$, $T_{F(REP)}$, $T_{L(REP)}$, and $T_{D(REP)}$) for each type of data involved.

3.2. PGHD repository

After the data have been assigned a meaning, they can be transferred to the dedicated PGHD repository.

Maintaining a repository within the CDSS might seem superfluous, since most of the data to be analyzed can be provided by PGHD servers on demand. We have analyzed advantages and disadvantages of integrating a PGHD repository into a CDSS, and we have concluded that it represents a fundamental component in the proposed framework. The PROs and CONs of having a centralized repository are summarized in Table 3 and will be discussed in the following paragraphs.

Table 3: Advantages and disadvantages of integrating a PGHD repository into a CDSS

PROs	- It is possible to save the complete history of the collected PGHD measurements. - For data analysis, it is faster to extract the data from the repository rather than from the sources. - It is possible to store content that is not served by PGHD sources (e.g., patient-reported information, inputs by clinicians, results of data analysis). - It is possible to assign context labels to the collected data.
CONs	- It is necessary to allocate memory for data storage. - It is necessary to design and maintain the repository. - It is necessary to design, develop and maintain components to interact with the repository. - It is necessary to backup the content regularly, to avoid possible data loss.

Integrating a repository requires some effort, both in terms of resources (e.g., allocation of memory) and in terms of time, since designing and maintaining the repository and the CDSS components that must interact with it may be time-consuming. However, we believe that the advantages of having a PGHD repository, or the disadvantages of not having it, justify the effort invested in its integration. In fact, first of all, not all PGHD systems maintain the complete history of the collected measurements. For example, the Abbott's Freestyle Libre system can only store data covering a period of three months. Saving data in a dedicated repository allows to keep track of the subject's BG profile over time. Otherwise, such data would be lost after three months.

Furthermore, requesting data from the different sources at the time of the analysis requires more time than extracting it from a dedicated repository. In case of analysis over an extended timeframe, the same operation performed on the fly would cause a significant delay in producing the results for the user.

In addition, saving the data in the dedicated repository as they are produced also allows to associate to each measurement the context labels (see Section 3.1) which enrich the value, contextualizing it. Assigning context labels at run time would not be feasible, since it is burdensome from a computational point of view.

Finally, in any case, maintaining a dedicated repository is essential for storing user-specific parameters, including login credentials, patient-reported information, personalized settings or information entered by the clinician manually. For example, if the system supports the management of diabetic patients, for each subject it is necessary to memorize his/her hyperglycemia and hypoglycemia thresholds, to perform a personalized analysis of his/her glycemic profile. This information could not be obtained from any external source. Thus, we concluded that a dedicated repository is needed.

Integrating a dedicated repository that collects data from heterogeneous sources makes our CDSSs similar to systems for multidimensional online analytical processing, known as MOLAP [157]. Systems for MOLAP are usually based on data warehouses, into which the data are regularly imported from other sources (e.g., hospital information systems) using ETL *(Extraction, Transformation and Load)* techniques. MOLAP systems allow their users to perform analytical queries on the collected data. However, usually MOLAP systems perform analyzes that are less complex than KBTA, which is one of the foundations of our framework.

After identifying the PGHD repository as a fundamental component of the framework, we evaluated what type of repository to include. The selection of the type depends on the purpose of the CDSS, on the expected volume of data manipulated by the system, and on the characteristics of its users. We evaluated two commonly used options, namely relational databases (RDBs) and NoSQL databases. Their characteristics are summarized in Table 4 and will be discussed in the following paragraphs.

Table 4: Characteristics of RDBs and NoSQL databases

	RDBs	NoSQL databases
PROs	- Complex queries can be performed, using SQL - Vertically scalable - Suitable for transaction-based application.	- Suitable when data structure is unknown. - Suitable for hierarchical data storage - Horizontally scalable
CONs	- Not suitable when data structure is unknown. - Not suitable for hierarchical data storage.	- SQL is not applicable - Less support than RDBs - Updates may not be effective immediately.

Relational databases (RDB) are a widely used type of repository when dealing with structured data. In RDBs data is usually stored in tabular form. Each entity of the domain is stored in a different table, in which each row represents an instance of that entity, and each column represent an attribute. RDBs can be queried using the Structured Query Language (SQL), which is very versatile and allows to perform complex queries. RDBs are vertically scalable: in the event of increased data traffic, the single server can be enhanced, by upgrading its features, such as RAM (Random-Access Memory) or CPU (central processing unit). Several software solutions are available for managing RDBs, including MySQL [158] and PostgreSQL [159].

Non-relational databases, also known as *NoSQL databases*, represent an alternative to RDBs. NoSQL databases are used to store large volumes of unstructured or non-uniform data. In general, NoSQL databases do not store data in tabular form, but the storage may be document-oriented, column-oriented, graph-based or data may be organized as key-value pairs. Thus, storage is not constrained by the data structure, which may not even be known when designing the database. When the structure of the data to be saved is not known, NoSQL databases are to be preferred to RDBs. In addition, NoSQL databases are sometimes preferred if the volumes of data to be manipulated are very large, since they are more scalable than RDBs. In fact, NoSQL databases are horizontally scalable, which means that data traffic may be partitioned on multiple servers. However, an update on one of these servers may be acknowledged by the other servers with a delay. Thus, NoSQL are not recommended in case it is important to analyze new data promptly. As for RDBs, several commercial solutions are available for managing NoSQL databases, including RavenDB [160], Cassandra [161], MongoDB [162], and CouchDB [163].

Solution adopted in the framework

The considerations that constitute our framework are valid for both RDBs and NoSQL databases. In our applications, we preferred the RDBs for three reasons. First, more support and more documentation are available for setting up and deploying RDBs, while for NoSQL technologies often only community support is available. Second, NoSQL are not optimized for complex queries, while RDBs are, since they can be queried using SQL. Finally, we do not expect our applications to have data traffic that cannot be managed by RDBs. For example, one of the two applications described in the thesis is now deployed in two hospitals. However, to preserve the patients' privacy, each hospital has a separate installation, with a dedicated relational database. Therefore, the number of users of the single installation is limited and the data volume is not such as to give up query power in exchange for scalability. In the future, we might consider using NoSQL databases for long-term storage of data that is not often used in analyses, such as the less recent history of PGHD.

Thus, in our applications we included a relational database and we managed it using MySQL, the database management system provided by the Oracle Corporation [164]. MySQL provides connectors that allow applications to perform operations on the database using programming language, including Java, which is the one our CDSSs are based on.

In our CDSSs we develop dedicated components (*data loaders*) to upload data into our repository. Since the data loaders need to communicate with the RDB, we also create mediation components (*DB handlers),* to facilitate such communications. In particular, the DB handlers act as wrappers around the methods provided by MySQL connectors, to convert the java objects manipulated by the application into tuples that can be saved in the database tables, or vice versa.

In general, for each java object that needs to be saved, one row needs to be generated in a specific table in the database. Each attribute of the object must be saved in a specific column of that table. The data type of the java attribute and the data type of the database column must be compatible. To formalize and implement the mapping between any java object and its representation in the database we use Hibernate, a java framework distributed by the RedHat company [165]. Hibernate allows separating the object-based representation of concepts used in the java application from the representation of the same concepts as database tuples. For each object *O* to be saved in the database, a mapping file must be formalized using the Hibernate-specific XML language. The mapping file specifies which java class corresponds to which table in the database and, for each attribute in *O*, which column of the table should contain it. Once all the necessary mapping files have been defined, Hibernate provides methods for converting the java objects into database tuples and vice versa. DB handlers provide methods that wrap the Hibernate methods that can be called by the data loaders for opening the connection to the database and for storing, updating, or fetching data.

3.3. CDSS: knowledge base and engine

In general, the knowledge base (KB) of the CDSS should formalize in detail what the system needs to know in order to provide decision support. Of course, this highly depends on the purpose of the system. In this framework, we provide considerations which are refined for systems that analyze continuous PGHD. In particular, we consider CDSSs that summarize continuous PGHD using KBTA and provide decision support accordingly. Thus, the KB should formalize all the concepts needed to complete three steps

- The download and integration of PGHD from multiple sources
- The analysis of the collected data using KBTA
- The exploitation of the results of the analysis to provide decision support

In general, it is good practice to formalize the KB so that it can be easily maintained and expanded over time. The choice of the type of formalization depends on the CDSS.

For example, ontologies represent one of the most known formalisms for constructing a KB [166]. An ontology is a formal and explicit description of the concepts of a specific domain, and of the relationships and the relationships between them. Ontologies for CDSSs can be formalized using different software applications, including Protégé [167], which is free and open source.

Rules represent another widely used formalisms to define a KB. In particular, rules represent one of the most straightforward formalisms, since they allow specifying the expected behavior of the system under specific conditions. The complexity of the rules needed to process the PGHD for decision support depends on the specific application domain. When the rules are simple, it may be convenient to formalize them in the CDSS using a language of choice. In this case, it is necessary to develop a dedicated rules engine to run them. In case of complex rules, it may be appropriate to rely on third-party rules management systems (RMS). RMSs may require the developers to use a proprietary language for formalizing the rules. In general, the rules engine is provided by the RMS. Integrating an RMS may be useful for several reasons. First, it allows to clearly separate the formalization of the rules from the rest of the application. Thanks to this separation, both the maintenance of the existing rule set and the addition of new rules to the set is easier. Furthermore, RMSs are optimized for the execution of rules. Generally, they allow running multiple rules in parallel on the same data or running the same rule in parallel on more data, reducing the computational time of the CDSS.

Considerations related to the KB needed for the three steps identified in this section will be provided in the following subsections.

3.3.1. Download and integration of PGHD from multiple sources

First, the KB needs to formalize how to collect PGHD from the different sources and how to prepare them for further analysis, with particular focus on how to manage interconnected data provided by different devices [168].

For each variable of interest, the KB must define its main characteristics, including the unit of measurement used by the CDSS, the range of admissible values, and the meaning of the different values (e.g., normal ranges and ranges that correspond to impaired health conditions). The KB may also formalize possible relations between the collected variables. For example, as anticipated in Section 3.2.1, for each dependent variable, the KB should specify the list of independent variables. In our applications, the KB also specifies how to assign context labels on the dependent variables, based on the value of its independent variables.

In addition, the same variable may be downloaded from multiple sources. Thus, for each possible source, the KB needs to declare the download mode (e.g., source, communication protocol, recommended frequency for downloading data) and the main characteristics of the provided signal (e.g., type of data returned, unit of measurements used by the source, sampling frequency). The KB should also formalize how to collect the patient's consent, if needed (see Section 3.1.1). For each source, it is also necessary to build a mapping between the language it uses to describe the data and the language used by the CDSS. This ensures that the same signal taken from different PGHD sources is treated equally by the CDSS. Compatibility between different signals is also guaranteed once their relationships are correctly defined in the KB.

Solution adopted in the framework

In our applications, we favored simplicity when we formalized the KB for data download and integration.

We did not formalize a proper ontology to describe the concepts of interest for our domains, since formalizing it would have required integrating our

CDSS with software for the formalization of ontologies. However, our CDSS are developed in Java, which is a language for object-oriented programming and allows defining hierarchies of concepts through the inheritance mechanism [169]. In particular, each Java class can represent one concept, describing it with a set of parameters. It is then possible to create subclasses that extend the considered Java class to describe the same concept in more detail. Subclasses inherit the set of parameters defined in the parent class, which is complemented by a more detailed set, specifically formalized for the subclass. For example, in our applications we include a Java class to represent a generic PGHD measurement, which is characterized by a set of parameters that are common to the measurements of all the variables of interest for the domain (e.g., time of occurrence, identifier of the patient who owns the measurement). The generic PGHD measurement is then extended by different subclasses, one for each type of PGHD measurement, which contain the parameters that are specific for that variable. While the inheritance mechanism allows representing hierarchies of concepts naturally, expressing relationships between concepts is not straightforward. In our case, we formalized relationships mostly by defining attributes in our Java classes. For example, to describe the relationship “A time series *is made of* measurements”, we would assign to the Java class that represents the time series a specific attribute ("measurements"), which formalizes the list of PGHD measurements in the time series.

We used simple rules to define how to download the data from the PGHD sources and convert them into the formats used by the CDSS. Since the rules were not complex, we coded them in plain Java language. We use rules also to assign context labels to our dependent variables (Section 3.2.1). For example, in our application for analyzing the diabetic patient’s glucose profile, one of the most important tasks of the Data Integration Module is tagging each blood glucose measurement, to contextualize them within the day of the subject [170]. To do this, we consider the information about workout and sleep collected by an activity tracker. In this case, the context label can assume the following values: *sleep*, *workout*, *routine*, and *NA*. The *sleep* and *workout* values are assigned when the BG measurement occurs during a tracked sleep session or during a tracked workout, respectively. The *routine* value is assigned when the event occurs in an instant in which the patient is not sleeping and is not training. The *NA* value is assigned to each BG event occurring when the patient is not wearing the Fitbit tracker. In particular, we assume that the subject was not wearing the tracker at time t_i if no HR measurements are available in the interval $[t_i - 5\text{ min}; t_i + 5\text{ min}]$.

The data integration module acts upon the information formalized in the KB coded in Java to download and prepare the data for further analysis.

3.3.2. Analysis of the collected PGHD using KBTA

The aim of this framework is to facilitate developing CDSSs able to identify *scenarios* of interest in the collected data. The term *scenario* may indicate several concepts, depending on the purpose of the CDSS. Scenarios may be simple, when the information of interest for clinical decision making is the presence (or absence) of a specific pattern in the collected data. For example, for diabetic patients, diabetologists are interested in checking whether a specific alteration of the blood glucose profile (e.g., the dawn effect, described in detail in Section 2.4.4) occurs or not for the considered patient. Scenarios may also be complex. For example, in the application for monitoring the elderly living alone at home, we need to detect when a specific sequence of daily activities occurs.

To detect scenarios, both complex and simple, the KB must specify all the monitoring variables on which the KBTA must be applied and must describe in detail how to abstract those data to perform pattern detection. Commonly, consultation with clinical expert is needed, to identify the patterns of interest for the considered clinical domain, and to formalize them to apply KBTA.

In this framework, we suggest JTSA (See Sections 2.4.3) as a tool to perform KBTA on the collected data. We chose JTSA since it is flexible; as anticipated in Section 2.4.3 this tool provides several algorithms for temporal abstraction, that can be combined in workflows to detect any user-defined patterns. We discussed in detail an example of workflow in Section 2.4.4. For each pattern of interest for the considered clinical domain, one or more JTSA workflows must be formalized to allow its detection. For each JTSA algorithm used in the workflow, one configuration file must be provided to define the values of the parameters used by the algorithm. Such values should be determined with a clinical expert, to ensure that the formalized pattern have clinical relevance. The KB of the CDSS must include all the XML file that describe the workflows formalized for the CDSS, complemented with their configuration files.

Each workflow expects to receive specific time series as input. We use rules to feed each workflow with the correct input time series. We also use rules to specify the sequence of operations to be performed on the PGHD fetched from the database before feeding them to the workflow.

Furthermore, as anticipated, sometimes detecting a pattern requires running more JTSA workflows and combining their results. In that case, it is also necessary to formalize rules that determine the sequence of execution of the different JTSA workflows and the method of integration of the results. Examples of patterns that require running multiple workflows will be described in Chapter 4. When the formalized rules are not complex, they may be encoded in Java, without integrating the CDSS with a rules management system. In case of complex rules, our framework integrates Drools as rules management system [171], a tool distributed by the RedHat company [172]. Thus, in our framework the inference engine is composed by the rules engine provided by Drools. By running our rules, it prepares the PGHD data for the analysis, activates the integrated JTSA engine for running the JTSA workflows, and processes the results before presenting them to the users.

Details on how we integrated our applications with Drools and JTSA will be discussed in the following subsections.

Solution adopted in the framework to formalize and run complex rules

In our applications we adopted Drools for formalizing and running complex rules since it is open source and can be easily integrated into third party applications.

To integrate Drools into a third-party java application, it is necessary to use three libraries: one library to define the rules (kie-api-7.5.0.Final.jar), one library to pre-compile them (drools-compiler-7.5.0.Final.jar), and one library to run them (drools-core-7.5.0.Final.jar). The rules must be defined within one or more files with the .drl extension. Within the .drl file, each rule is uniquely identified by a string identifier, and consists of two parts, a premise that contains the condition to be tested, and a body listing the operations to be performed on the data when the condition is verified. The premise is formulated in a Drools-specific language which is very similar to Java, while the body is written in Java. All the .drl files containing the rules must be stored in a Java package within the application. The name of the Java package containing the rules must be specified in the Drools configuration file, which must be written in a specific XML language. When the Drools work session is started, by invoking a method (i.e., *fireRules*) provided by the drools-core-7.5.0.Final.jar library, Drools execute all the rules in the selected package.

Solution adopted in the framework to run JTSA workflows

Like Drools, JTSA is integrated into the CDSS as an external library. The schema in Figure 13 summarizes the sequence of steps needed to run each JTSA workflow.

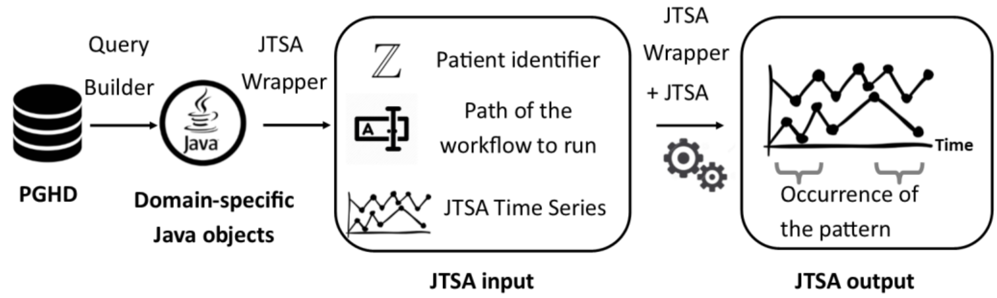


Figure 13: Steps to run a JTSA workflow in the proposed framework

The first step exploits the methods of the *QueryBuilder,* a java class where we code the methods for constructing the data queries according to the type of input data requested by the selected JTSA workflow and to the time interval of interest for the analysis. The QueryBuilder can activate the DB handlers to execute the queries and fetch the PGHD time series of interest, which are represented as java objects. In each application we program a component, which we call *JTSA Wrapper*, to encode the PGHD time series in the input format requested by JTSA. For each workflow, JTSA requires a set of three input objects: an integer to identify the patient the analyzed data belongs to, a string that specifies the path of the workflow to be run, and a list containing all the necessary time series of measurements. Each time series is a list of *TimeseriesDot* objects, which represents the single measurement within the time series. Each TimeseriesDot consists of three attributes, namely a String which identifies the type of value within an admissible set (e.g., *Double* or *Integer*), the value itself, and the time and date of the measurement. The datetime of the measurement must be expressed in the Java Long format as the number of milliseconds between the Unix reference date (i.e., 1st January 1970, 00:00:00) and that datetime. In the following we will refer to this date format as *Unix format*. After packaging the 3-item JTSA input, the JTSA Wrapper calls the JTSA library methods for running the workflow and collects the output, i.e., the time intervals in which the selected pattern is satisfied by the analyzed PGHD. In particular, each time interval in the output is represented in a JTSA-specific type, the *AbstractionDot*. Each AbstractionDot is defined by three attributes, namely the *label*, a string which identifies the pattern selected for the analysis, and *start* and *end*, the dates that delimit that time interval, expressed in the Unix format. In each application, rules are then applied on the AbstractionDots found, to provide decision support.

3.3.3. Exploitation of the results to provide decision support

As anticipated, the information of interest for the considered decision problem may be the presence (or absence) of a specific pattern, or sequence of patterns, in the collected data. In these cases, rules may be formalized for generating warnings based on the patterns found (or not found). In other cases, it may be meaningful to monitor the occurrence of a specific pattern, or sequence of patterns, over time to detect any changes of interest for the specific decision problem. Examples of relevant changes maybe the modification of the duration of the pattern occurrence, or the variation of its frequency over time. Also in this case, rules can be formalized to process the results of pattern detection, identify the change in the occurrence of the relevant patterns, and produce warnings to inform the users about it.

However, the purpose of the CDSS may not be to automatically produce recommendations based on pattern detection, but simply to allow the user to search for relevant patterns in the collected data. In this case, rules can be formalized to specify how to optimize the presentation of the results to the final user.

Examples of exploitation of the results of the pattern analysis to provide decision support will be presented in Chapter 4 and 5.

3.4. User interface

The user interface is the component of the CDSS that allows communication with the user.

The technologies used for its implementation highly depend on the nature of the CDSS and on the target user. Most commonly, users prefer either smartphone applications or web applications, so that the they can access the CDSS wherever they are. Sometimes CDSS developers prefer to provide a web application because it is easier to maintain, being installed centrally on a server, instead of the user's own smartphone. Furthermore, a web application can be accessed from any device having a web browser, regardless of the nature of the device. Thus, it may be compatible with any devices owned by the user. However, if the CDSS is developed with the purpose of providing reminders to its users, it is preferable to include a smartphone application in the system, since web applications can present reminders to the user only when he/she is online.

Independently of the selected technologies, user interfaces must be designed to overcome the critical issues discussed in Section 2.3. The main task of the interface must be to make the use of the system easy, and not intuitive. It is necessary to respect some precautions in the development of the user interface, in order to guarantee two fundamental qualities of the CDSS: *usability* and *accessibility*. The International Organization for Standards (ISO) defines *usability* as <<*the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*>> [173]. As regards *accessibility*, a system is said to be *accessible* when it can be easily used by individuals with disabilities. In the following sub-section we will describe the principles we followed in our applications to reach and assess usability. We did not focus on accessibility in the pilot versions of our applications, since the patient populations in our pilot studies did not have disabilities. However, in the following of this chapter a sub-section collects the precautions to be adopted to make a system accessible, besides usable. We will adopt such precautions in future releases of our applications.

Solution adopted in the framework to reach and assess usability

As anticipated in Section 2.3, the graphical interface is one of the factors that most affect the usability of the system. In fact, even systems with functionalities that are perceived as useful by the users are not appreciated if their interface is not user-friendly. Several factors influence the users' perception of the interface. The most relevant are the following:

- Ease of navigation
- Waiting time
- Detection of input error
- Supported languages
- Management of user accounts

- Reminders

Navigating the application should be intuitive for the user and the interface must contain only elements that are essential for the functioning of the system. The system must contain interactive elements, meant for dialoguing with the user, such as links, buttons, selection menus, or input text areas. Each interactive element must have an explicit meaning. For example, the label shown on buttons must clearly indicate the action that will be triggered by pressing that button. All the textual contents provided by the interface must be well organized and concise, and the user must be allowed to request further details, if interested in additional information. It is necessary to carefully evaluate which content should be presented in the interface by default, and which should be provided to the user on demand. Multimedia elements should be used only when needed and their size should be limited, to limit the memory occupation, in case of smart applications, or to limit the use of the internet connection, in the case of web applications. Web applications should adapt automatically to the browser window to be independent of the screen size of the device and should avoid requiring the installation of plugins.

Each interactive element must return feedback quickly, to notify the interaction occurred [174]. It is known that the user perceives a reply as instantaneous if he/she receives it in less than 0.1 seconds. The maximum waiting time that is considered user-friendly is 10 seconds. In addition, when it is necessary for the user to wait, the interface must explicitly show that the system is producing a response.

For each element that asks information from the user, the interface must check that the user input is valid. This will limit input errors and consequent unexpected system behaviors, which would discourage the user from using the CDSS.

If the application is meant for an international audience, the content should be available in multiple languages. In addition, the application content should also avoid concepts or images that may be misunderstood or perceived disrespectful in other cultures. The dates should be expressed using an unambiguous notation, that mentions the time zone explicitly. The time zone should be tuned according to the user's country, specified by the browser o by the smartphone's settings.

Each user must be able to create and manage his/her own account and must be reminded to change the password frequently. The system must also provide a functionality for recovering the password in case the user does not remember it.

The CDSS should provide settings for personalizing the frequency of reminders received by the users. To prevent user fatigue, the number of reminders sent must be limited.

After applying the described precautions when designing the CDSS, an evaluation of the usability perceived by the user must be carried out. Usually, such evaluation takes place during the pilot studies of the application, asking users to fill in questionnaires. The System Usability Scale (SUS) is one of the most used indicators to evaluate the usability of software applications

[175]. It is a 10-item questionnaire that focuses on evaluating the system's user interface, by asking users questions related to the perceived ease of use of the system and assessing their willingness to continue using it after the test phase. The rules for computing the numeric SUS score based on the subjects' answers and the rules for evaluating the results are available online [175]. Obviously, if the functionalities of the CDSS are not perceived as useful by the users, the SUS score will be low regardless the characteristics of the user interface. It is good practice to ask the users to fill in the SUS questionnaire at least twice, once after a short time of the use of the application and once after a few months of use (e.g., 6 months). Repeating the assessment helps to understand if the availability of users to use the CDSS varies over time. In case it varies, it is possible to analyze the answers to the SUS questions, to understand which feature of the system needs to be improved.

Solutions to reach accessibility

In Italy, the accessibility of computerized tools is regulated by the Stanca Act, issued in 2004 [176], which protects everyone's right to take advantage of information technology. Similar laws regulate the same subject in other countries. In general, using multimodal communication methods facilitates the accessibility of the interface by people having impairment. The Stanca Act particularly focuses on precautions that can facilitate three categories of individuals:

- Individuals with visual impairment
- Individuals with hearing impairment
- The elderly

In particular, according to the Stanca Act, to guarantee accessibility by subjects having visual impairment, each element of the user interface should be assigned a textual description that can be read by a screen reader. For example, each image should be complemented with a description that comments on the content. It is not required that such description is the same description presented in the interface. In addition to blind people, the system must not discriminate against color-blinded individuals, therefore it is necessary to make sure that the information is not provided only through color.

For deaf individuals, a textual alternative must be provided for each audio content included within the interface.

Finally, accessibility also means ensuring that the application is suitable for the elderly, if such age group is included in the target audience of the system. In this case, the text fields of the interface must have fonts of adequate size and the contents of the application must be easy to understand for subjects who may have a physiological mental decline.

4. Case study on falls prevention

Falls in the elderly are a known social problem, being a major cause of loss of independence, hospitalization or increase of hospital stay, decreased quality of life, and increased social costs [177]. They are also associated with psychological and functional sequelae, independently from the injury severity. Since falls often have consequences, such as fractures, they also involve high costs for the healthcare system. Since the number of the elderly at risk has been increasing, also the costs sustained to manage fall events are expected to increase.

Falls are associated with a vast and heterogeneous set of risk factors, which may be related both to the characteristics of the environment in which the elderly live, and to the clinical, socio-demographic and behavioral characteristics of the individual [76–79, 178–182]. Some factors are related to the subject's health status and include chronic diseases (e.g., neurological disorders, diabetes, and cardiac arrhythmias), simultaneous intake of multiple drugs or specific drug therapies (e.g., antiepileptics), sensory limitations (e.g., visual and/or hearing impairment), and alterations of the subject's balance. Among behavioral factors, incorrect eating habits may cause nutritional deficiencies such as vitamin deficiency or dehydration, which are known to increase the individual's risk of falling. Behavioral factors also include lack or excess of physical activity, fear of falling, and irrational behavior, such as moving within the house at night without turning on the lights or walking without using a cane although it is prescribed. As regards the living environment, the most dangerous risk factors are inadequate lighting, slippery floors, physical obstacles to the movement of the elderly (e.g., portable oxygen dispenser, presence of pets within the apartment), and inadequate room temperature and humidity conditions, which can weaken the elderly. Among the socio-demographic factors, an inadequate economic condition is one of the most critical, since it may lead to malnutrition or impossibility of adapting the home to the subject's needs.

Within NONCADO (Italian expression meaning *I do not fall*), a project funded by the Lombardy Region in Italy, we aim at preventing falls in elderly people living alone at home [183]. Living alone implies a difficult or delayed detection of a possible decline that in turn may increase the subject's risk of falling. Reasonably, a decline influences the patient's habits, concerning for example daily activity, sleep quality and/or quantity, time spent outside the house, and consumption of hot meals[177]. These considerations motivated the development of a monitoring system able to detect such changes, and to inform the subject's family of a possible decline. Early detection of changes in the patient's daily habits is the innovative feature of the NONCADO system, compared to the devices known in the literature [129, 130], which were described in the overview on CDSS in Section 2.3.

The project was carried out in collaboration with two information technology companies, namely Spindox [184], a company based in Milan in Italy, and Biomeris [185], an academic spin-off born from the University of Pavia, Italy. Furthermore, for the NONCADO project a collaboration has been activated with the Department of Neurorehabilitation Sciences of Casa di Cura Privata del Policlinico (CCPP) Hospital in Milan in Italy [186]. A team of clinicians working for CCPP provides feedback on the system's use cases as experts in the clinical field. In addition, CCPP provides a pilot site, where tests can be carried out to assess the usability and accuracy of the system on a real-world population of elderly patients.

In this chapter we present the prototype of the NONCADO system. First, we describe the use cases of the system and how it detects daily activities by integrating data from a network of monitoring sensors. Furthermore, we present the results of its application in a preliminary evaluation study on a small group of neurological patients.

4.1. The NONCADO system

Section 4.1.1 will describe the use cases of the NONCADO system and its architecture, which complies to the framework presented in Chapter 3. In section 4.1.2 we will focus on how the CDSS detects the subject's daily activities by applying JTSA on the monitoring data.

4.1.1. The architecture

The architecture of the NONCADO system is shown in Figure 14. In the following paragraphs we will describe the five main components of the system, namely:

- the PGHD sources, i.e., the network of sensors which collects the monitoring variables;
- the PGHD repository, where the raw data collected by the sensors are stored;
- the CDSS, which analyzes the collected data;
- the event repository, which stores the results of the analysis;
- user interface, which reports the results to the users.

In the NONCADO system, we designed and developed the CDSS component and its communication with the other components. Spindox developed the PGHD repository and the user interface; Biomeris provided the event repository. This thesis will focus on the CDSS, while providing an overview of the entire system.

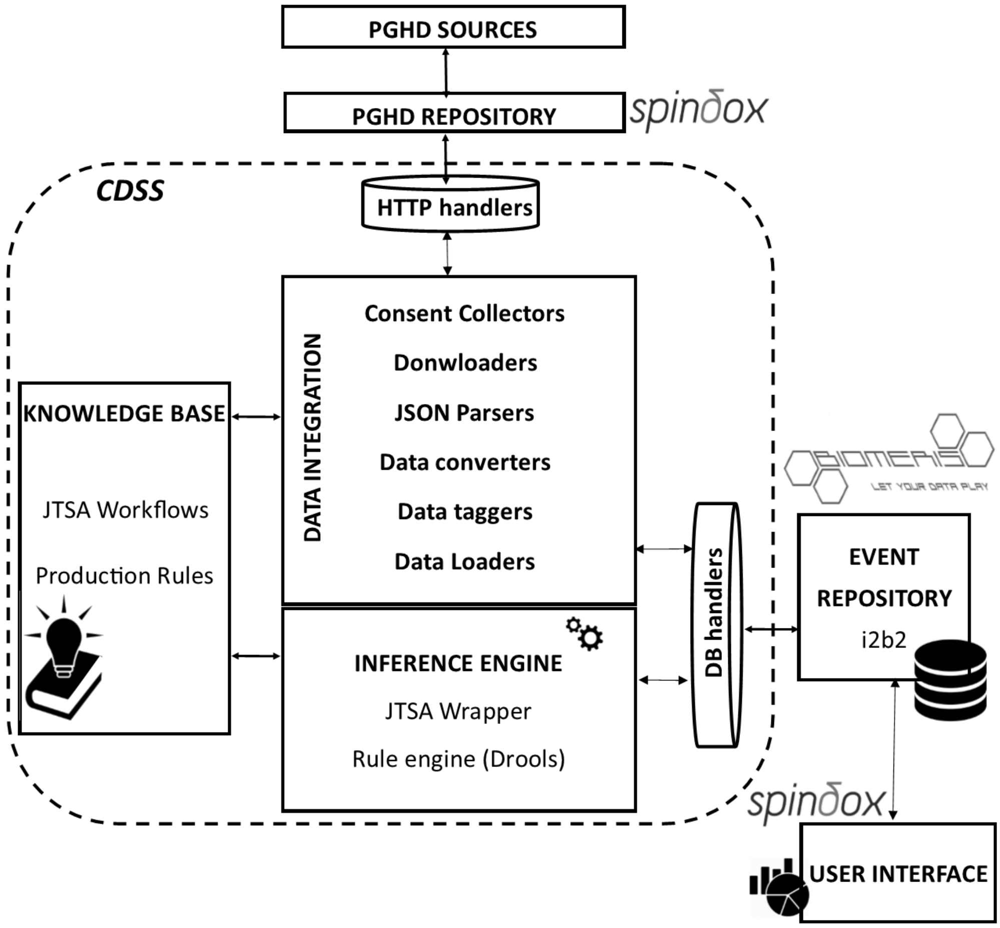


Figure 14: The NONCADO system's architecture

PGHD sources and PGHD repository

The network of sensors includes both wearables and environmental sensors. In particular, a set of environmental sensors collect measurements of motion using PIR technology, temperature, humidity, and luminosity within each room of the house. From these sensors we collect one measurement every 4 seconds. In addition, pressure mats can be placed under the mattress to detect the subject's presence in bed. For the prototype developed in the project, the bed was conceptually divided into three sections, i.e., feet, back, and head. Under each section we placed a pressure mat. As the for the other sensors, from each pressure mat we collect one pressure measurement every 4 seconds. Furthermore, two photo-cells positioned in sequence, close to the door, are used to detect when the subject passes through the door, and in which direction, i.e., enter or exit. The network of environmental sensors also include a sensorized carpet (6 meters long), which monitors the subject's walking speed and step length [187]. Since the carpet is expensive, it is not realistic to install it at each patient's home. Thus, the idea is to install it at a healthcare facility and ask the patient to use it regularly, for example weekly or monthly, to monitor improvements or deteriorations in his/her gait over time. In addition to environmental sensors, a Fitbit activity tracker (model: Fitbit Alta HR) monitors the subject's sleep and physical activity.

All the environmental sensors are connected to Raspberry PI boards, which gather the collected measurements, and transmit them to a PGHD repository managed by Spindox, which stores all the raw measurements. As anticipated in Section 2.2.2, the Fitbit data are stored into the Fitbit Cloud and served by dedicated Fitbit servers. Thus, Spindox developed an extension of the PGHD repository, which automatically collects data from the Fitbit cloud and saves it along with the other data.

The PGHD repository provides the data to the CDSS through the HTTP protocol.

CDSS and event repository

A CDSS analyzes the collected data. The system's lifecycle is shown in Figure 15.

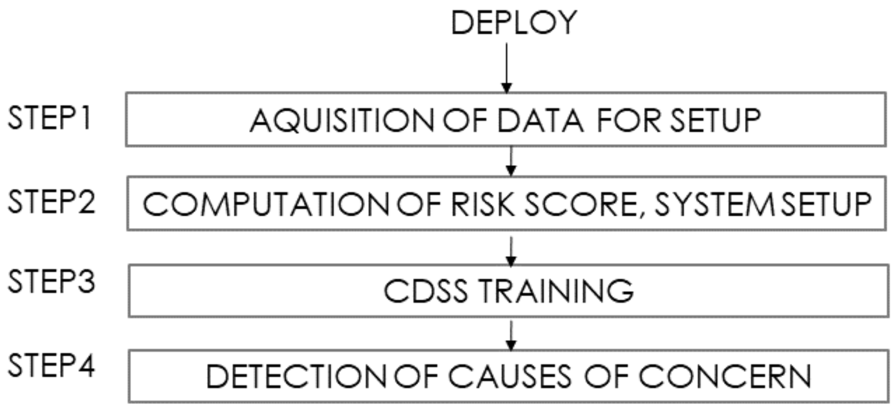

DEPLOY
STEP1
AQUISITION OF DATA FOR SETUP
STEP2
COMPUTATION OF RISK SCORE, SYSTEM SETUP
STEP3
CDSS TRAINING
STEP4
DETECTION OF CAUSES OF CONCERN

Figure 15: The NONCADO system's lifecycle

Following the installation of the CDSS, the user is asked to fill in a series of questionnaires for the collection of information useful for framing the monitored subject (STEP 1). The required data includes all the information that the CDSS needs to calculate the subject's risk score according to nine risk models that are applicable to non-hospitalized subjects aged over 65 years, identified in a review of the literature on falls [188–196]. Some information is mandatory since it is easily available. Mandatory data includes personal data (e.g., age and gender) and qualitative assessments of the subject's independence. Other information is optional since it may be unavailable or more challenging to retrieve. Optional information includes the results of specific clinical tests such as the postural oscillation test or sight examination. On the basis of the collected information, the CDSS calculates the subject's risk score, by integrating the estimates of the different models, which will be described in Appendix A (STEP 2). In the next phase (STEP 3), the network of sensors begins to collect the monitoring variables. By processing the data collected in this first monitoring phase, the CDSS defines a profile of the subject. For example, the CDSS learns its habits in terms of amount of daily activity, number of night-time awakenings, frequency of exits from the house. After the initialization phase, the system starts working in its steady state (STEP 4). The sensors regularly send the collected time series of measurements and the CDSS processes them to detect two kinds of possible causes of concern: deviations from the patient's habits, that could indicate changes in the his/her health status, and behaviors to correct, since they are known to increase the subject's risk of falling. Deviations from the subject's profile include for example increase in night-time awakenings or decrease in cooking activity. The set of risky behaviors we aim to detect includes for example moving within the house with poor light conditions. When the system detects that the patient is moving within the house in the dark, it immediately switches a light on, and then generates a warning to invite the subject to avoid such behavior in the future. The complete list of causes of concern is provided in Table 5.

Table 5: Causes of concern that may indicate a decline in the patient's health status

DEVIATIONS FROM THE SUBJECT'S PROFILE	Increase in night-time awakenings
	Increase/decrease in resting time
	Increase/decrease in time spent in specific rooms (e.g., kitchen, bathroom)
	Decrease in time spent outside the house
	Decrease of cooking activity
	Decrease in bath/shower activities
	Decrease in physical activity
	Decrease in walking speed
	Decrease in sleep quality (e.g. decrease in sleep efficiency)
BEHAVIORS TO CORRECT	Moving within the house in poor light conditions
	Getting up from the bed in poor light conditions
	Exiting the house during the night
	Taking a bath/shower during the night
	Exiting the house while cooking (and using the stove)

	Absence of cooking activity, i.e., not having hot meals
	Spending time in a room with inadequate humidity and/or temperature conditions

The CDSS also produces a warning when the subject is not detected within the home for a long time, despite not having detected any exit from the house. This may indicate a system malfunction, but it may also indicate that the subject is inactive for a long time, but not resting on a bed or chair equipped with a pressure mat, and it may indicate the occurrence of a fall event.

Detecting causes of concern requires searching for domain-specific patterns. To detect such patterns, the CDSS runs a set of JTSA workflows we formalized specifically for the project. The parameters of each JTSA algorithm in these workflows were tuned after functional tests aimed at observing the response of the involved sensors to stimuli designed to simulate the patient's activities of daily living. Besides the JTSA workflows and the rules to run them, the KB contains the description of the house of each patient, to assess what types of sensors are present within each room and, consequently, what patterns can be detected. The description of the house is filled in by the users when the sensors are installed in the apartment and is stored in the JSON format. For each room, the description provides an identifier of the room, a textual description provided by the users, which will be used to refer to that room in the reports, and the list of the sensors installed in the room. To distinguish different sensors of the same type, each sensor is characterized by an attribute that identifies the Raspberry pin to which the sensor is connected. Sensors of the same type may monitor different variables. For example, in the same room one temperature sensor may monitor the temperature perceived within the room, while one temperature sensor may monitor the temperature of the area close to the stove to detect when the subject uses it. Even if both time series collect temperature measurements, they have an intrinsically different meaning. Thus, each sensor is assigned an attribute, the *context_id*, that identifies the meaning of the measurement. In the example, the first sensor has the context_id equal to *Generic* while the second sensor has the context_id equal to *Stove*. In Appendix B we provide the JSON document that describes our pilot site as an example of house description.

We define *event* the occurrence of a specific pattern. For further analysis, the CDSS stores the detected events variables into a dedicated i2b2 [159] data warehouse managed by Biomeris. According to the detected events, at the end of each day the CDSS produces a report, meant both for the monitored elderly subject and his/her family. This report warns against the potentially risky behaviors that have been detected during the day (Table 3, second section) and provides a summary of the monitored variables. The CDSS produces an additional report at the end of each week, to compare the considered week to the previous ones in terms of detected events. Daily and weekly reports are stored in the i2b2 data warehouse, along with the detected events.

User interface

Daily and weekly reports are delivered to the users through a mobile application developed by Spindox, which retrieves them from the i2b2 repository. The reports should allow the subject's family to remotely monitor the user, and to detect significant changes in his/her habits, to early identify a possible decline.

In the daily report the user can visualize the following parameters:

- Total time spent within the home
- Total time spent outside the home
- Number of night-time awakenings with getting up
- Total time spent in bed (during the day)
- Total time spent in bed (during the night)

The weekly report contains the following information:

- Number of days the patient did not leave the house
- Report of the exercise on the carpet (i.e., number of repetitions and walking speed for each repetition)
- If detected, failure to carry out the exercise on the carpet
- Comparison to the previous week, in terms of the following parameters
  - time spent in each room, with particular focus on the rooms of interest (e.g., kitchen, bathroom)
  - time in bed during the day
  - time in bed during the night
  - number of exits from the house
  - time spent outside

4.1.2. Domain-specific pattern detection

To detect the causes of concern in Table *5*, we have identified 10 *use cases* (Table 6). Eight use cases (use cases 1-8) represent typical activities that compose the day of an elderly person who is independent enough to live alone, with particular attention to the activities that may influence his/her risk of falling. Two use cases (use cases 9 and 10) represents two environmental conditions we need to detect. The NONCADO CDSS is implemented to detect when these use cases occur, by processing the monitoring data of the network of environmental sensors. For each use case, Table 6 provides a brief description.

Table 6: The use cases of the NONCADO system

USE CASE	DESCRIPTION
1. AWAKENING	The subject gets up from the bed and possibly exits the bedroom

	(e.g., he/she goes to the bathroom or he/she goes to the kitchen to have breakfast).
2. NIGHT-TIME AWAKENING	The subject is in bed, gets up and then goes back to bed.
3. EXIT	The subject leaves the house.
4. HOT MEAL	The subject cooks at the stove.
5. ACTIVITY	The subject moves within the room.
6. RESTING	The subject is in bed, or on a sofa or armchair that is equipped with a pressure mat.
7. BATH/SHOWER	The subject is in the bathroom to take a shower or a bath.
8. WASHING THE DISHES	The subject is at the sink and washes the dishes.
9. POOR LIGHT CONDITIONS	The room is in poor light conditions.
10. ENVIRONMENTAL DISCOMFORT	One or more of the following events occur: - The room temperature is not in the optimal range [18°C-26°C]. - The room humidity is not in the optimal range [40%-65%]. - A rapid increase/decrease of the room temperature occurs (corresponding to a variation of 2°C in 10 minutes)

Both the causes of concerns and the variables in the reports are computed according to the use cases detected in the monitoring data. Some of them are identified/estimated by detecting a specific use case in the data. For example, in the report the total time in bed during the day is estimated by detecting the *use case 8* and selecting its occurrences during the day. Among the causes of concern, *exiting the house during the night* is detected when the CDSS detects *use case 3* is activated during the night hours (e.g., from 10 PM to 6 AM in the morning, in the default settings). Other variables are estimated by detecting a specific combination of use cases. For example, among the causes of concern *Moving within the house in poor light conditions* is detected when *use case 5* activates when *use case 9* is active. Finally, other causes of concerns are identified by the absence of activation of a use case. For example, *Absence of cooking activities* is detected when the *use case 4* is not detected in the monitoring data for at least two days when the patient is at home.

The detection of the use cases (Table 6) is based on the identification of 9 domain-specific patterns in the time series of PGHD collected by the sensors:

- *Pattern 1: Movement within the room*
- *Pattern 2: Movement within the room (in poor light conditions)*
- *Pattern 3: Presence in bed*
- *Pattern 4: Permanence in the rooms*
- *Pattern 5: Cooking*
- *Pattern 6: Bath/Shower*
- *Pattern 7: Washing the dishes*
- *Pattern 8: Environmental discomfort (Temperature)*
- *Pattern 9: Environmental discomfort (Humidity)*

In the following of this section, we will describe each pattern, listing the type of sensors needed for its detection, the JTSA workflows formalized for processing the data and the steps needed to run them. In Section 4.1.3 we will describe the correspondence between the causes of concern and the patterns we need to detect for identifying them.

Pattern 1: Movement within the room

This pattern identifies the time intervals in which the subject moves within the room. The steps to detect the pattern, implemented using Drools are summarized in Figure 16.

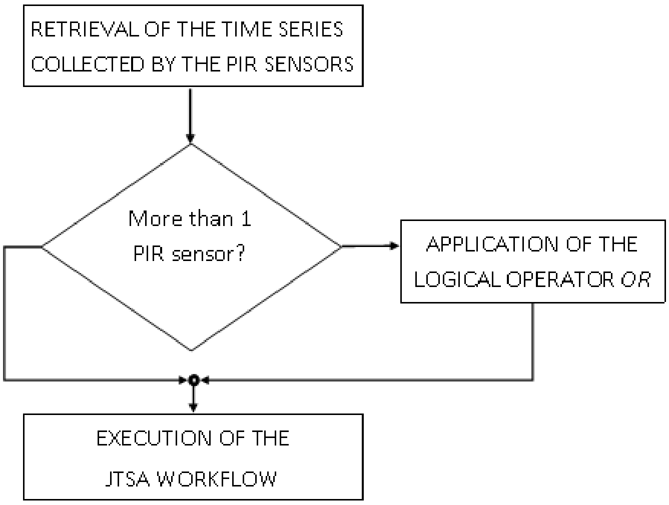

RETRIEVAL OF THE TIME SERIES COLLECTED BY THE PIR SENSORS
More than 1 PIR sensor?
APPLICATION OF THE LOGICAL OPERATOR *OR*
EXECUTION OF THE JTSA WORKFLOW

Figure 16: Sequence of operations to detect movement within the room

First, the CDSS retrieves from the PGHD repository the time series of measurements collected by the PIR sensors located in the room. As anticipated, each point in this time series is composed of a tuple, <DATE, VALUE> where DATE represents the time instant (date and time) of the measurement, and VALUE is 0 when the sensor does not detect movement, 1 otherwise. If multiple PIR sensors are present in the room, we apply a logical operator on the retrieved time series to obtain a single time series of pairs <DATE, VALUE> where VALUE is 1 if at least one PIR has detected movement in that instant temporal, 0 otherwise. Since several PIRs may not be synchronized to the second, when applying the OR operator, we consider two measurements, M_PIR1 and M_PIR2 (M_PIR1 precedes M_PIR2), as *simultaneous* if they are no more than 4 seconds apart. Finally, we run the JTSA workflow summarized in Figure 17. We will refer to this workflow as *movement_within_a_room*.

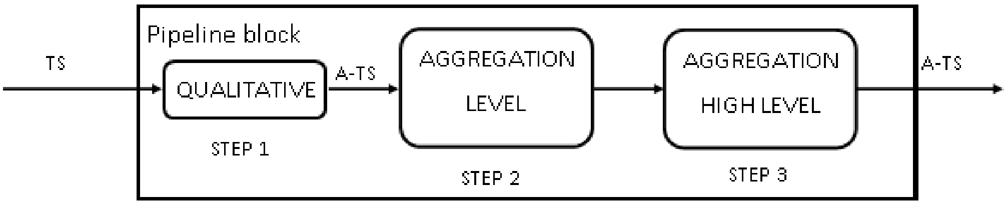


Figure 17: JTSA workflow to detect movement within a room

The workflow is composed of a pipeline block which includes three steps. The configuration parameters of the algorithms in each step are listed in Table 7. In STEP 1 the Basic Qualitative algorithm applies a qualitative abstraction to the values in the provided time series. In particular, the *absence* (of movement) label is assigned to all the values below 0.5, while the *movement* label is assigned to the values above 0.5. STEP 2 performs an aggregation of the episodes with the same label that are no more than 20 seconds apart. Since the minLen is low, none of the obtained episodes is filtered out. In STEP 3 an Aggregation HighLevel algorithm performs a further aggregation of the episodes found in the previous step, extracting only those with the *movement* label, longer than 8 seconds.

Table 7: Parameters of the JTSA workflow for detecting movement within a room

ALGORITHM	PARAMETERS

Qualitative	th=0.5
	label=absence,movement

Aggregation Level	gap=20
	minLen=3
	granularity=SECONDS

Aggregation HighLevel	gap=60
	minLen=8
	granularity=SECONDS
	label=MovementInRoom
	levels=movement

Pattern 2: Movement within the room (in poor light conditions)

This pattern is a variant of Pattern 1 and identifies the time intervals in which the subject moves within the room in poor light conditions. The sequence of operations to detect the pattern are summarized Figure 18.

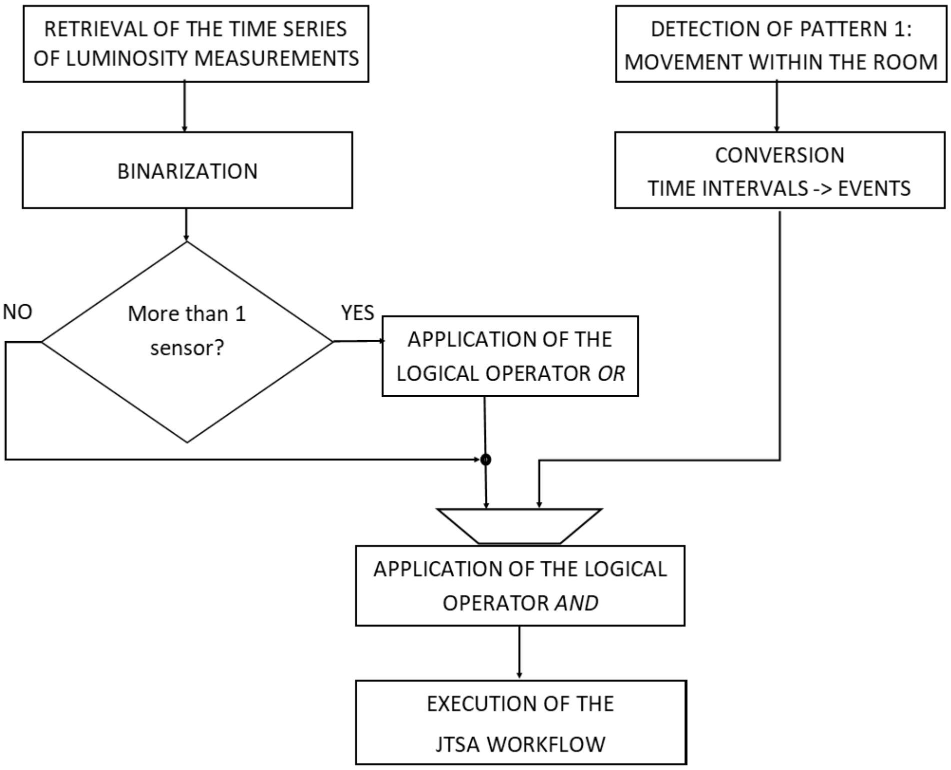

RETRIEVAL OF THE TIME SERIES OF LUMINOSITY MEASUREMENTS
BINARIZATION
More than 1 sensor?
NO
YES
APPLICATION OF THE LOGICAL OPERATOR *OR*
DETECTION OF PATTERN 1: MOVEMENT WITHIN THE ROOM
CONVERSION TIME INTERVALS -> EVENTS
APPLICATION OF THE LOGICAL OPERATOR *AND*
EXECUTION OF THE JTSA WORKFLOW

Figure 18: Sequence of operations to detect movement within the room in poor light conditions

First, two operations are carried out in parallel. On one hand, the CDSS retrieves the time series measurements collected by the luminosity sensors located in the room. In this case, each point in the series is a tuple <DATE, VALUE> where VALUE is an integer ranging from 0 to 1500, proportional to the intensity of the sensed luminosity. The CDSS binarizes this series, converting each value lower than 200 to 1, that represents luminosity, and each value higher than 200 to 0, representing darkness. The threshold was set to 20 after analyzing the profile of measurements provided by the sensor in different light conditions, artificially simulated. If multiple sensors are present in the room, we apply the OR logical operator to obtain a single time series of pairs <DATE, VALUE> where VALUE is 1 if at least one sensor has detected darkness in that instant temporal, 0 otherwise.

On the other hand, we identify the time intervals in which the subject moves within the room, by detecting *Pattern 1: Movement within the room*, as previously described. By detecting Pattern 1, we obtain a time series of episodes. To combine them with the time series that conveys the information on the light conditions, we convert the time series of episodes in a time series of events, as shown in Figure 19.

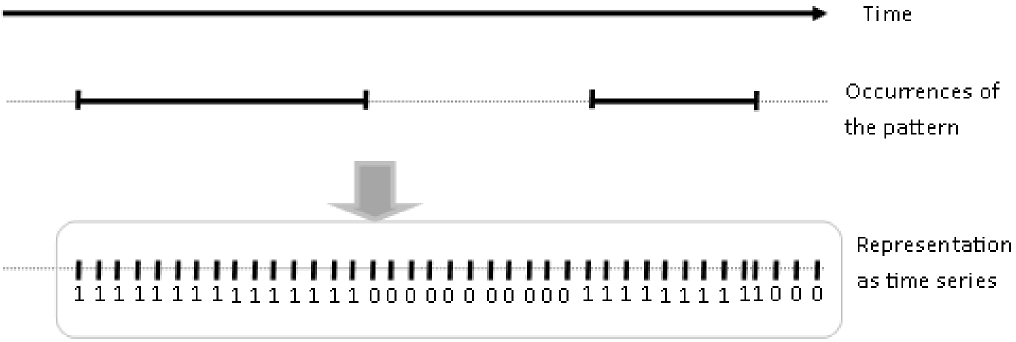


Figure 19: Conversion from time intervals to events

In particular, we create a time series of pairs <DATE, VALUE> where VALUE is 1 if DATE belongs a time interval that verifies *Pattern 1: Movement within the room*, 0 otherwise. The DATE elements are generated with a granularity of 4 seconds, starting from the start date of the first time interval in the time series of episodes. For each interval, we also generate the DATEs corresponding to the start date and end date of the interval, if they are not generated automatically, i.e., when they do not belong to the sampling grid. This ensures that the exact start and end time of the occurrence is not lost in the conversion. In the following, we will refer to this conversion as *conversion from time intervals to events* (TIs->EVs).

To detect when the subject moves in poor light conditions, we then apply the AND operators to the obtained two binary time series. We obtain a time series of pairs <DATE, VALUE> where VALUE is 1 if movement in the darkness has occurred in that instant temporal, 0 otherwise.

Finally, we apply a JTSA workflow to aggregate the time intervals in which the selected pattern occurs for more than 5 seconds. In terms of the structure, the workflow is equal to the workflow shown in Figure 17. However, the set of parameters is slightly different (Table 8).

Table 8: Parameters of the JTSA algorithms for detecting movement within the room in poor light conditions

ALGORITHM	PARAMETERS

Qualitative	th=0.5
	label=ok, warning

Aggregation Level	gap=20
	minLen=3
	granularity=SECONDS

Aggregation HighLevel	gap=20
	minLen=5
	granularity=SECONDS
	label=MotionWhenDark
	levels=warning

Pattern 3: Presence in bed

This pattern identifies the time intervals in which the subject is on the bed. As anticipated, the bed is conceptually divided into three sections, i.e., feet, back, and head. For each section, a pressure mat detects the pressure exerted by the patient's body on the section and provides a time series of measurements whose value ranges from 0 to 1024, proportionally to the exerted pressure. The operations to detect the pattern are summarized in Figure 20.

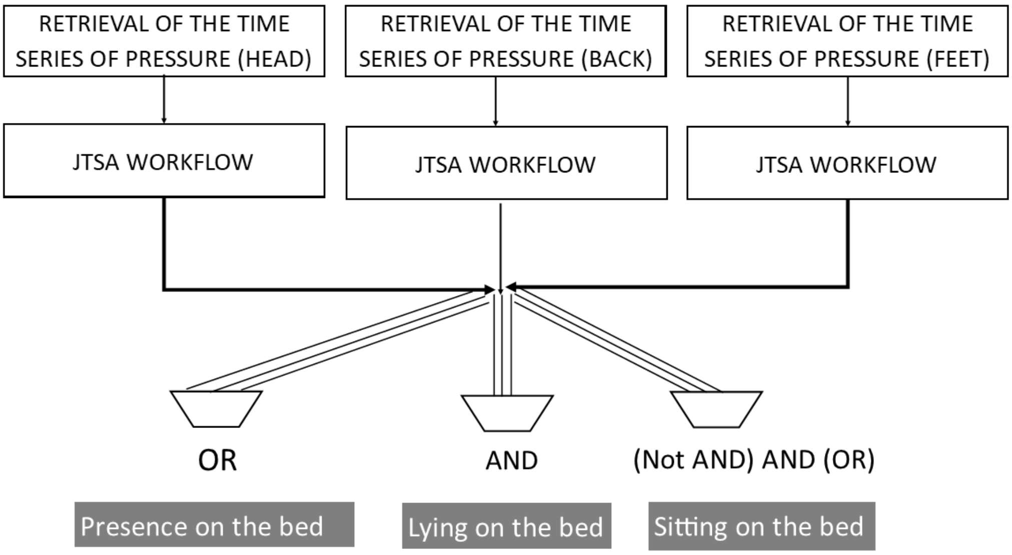

RETRIEVAL OF THE TIME SERIES OF PRESSURE (HEAD)
RETRIEVAL OF THE TIME SERIES OF PRESSURE (BACK)
RETRIEVAL OF THE TIME SERIES OF PRESSURE (FEET)
JTSA WORKFLOW
JTSA WORKFLOW
JTSA WORKFLOW
OR
AND
(Not AND) AND (OR)
Presence on the bed
Lying on the bed
Sitting on the bed

Figure 20: Sequence of operations to detect presence in bed and its variations

First, for each section the collected pressure measurements are retrieved from the PGHD repository. For each section, the collected time series is fed to the JTSA workflow in Figure 21, which identifies presence on the section by detecting when a rapid increase in pressure is followed by stationarity in the same variable. The workflow consists of three blocks, two pipelines and one complex block.

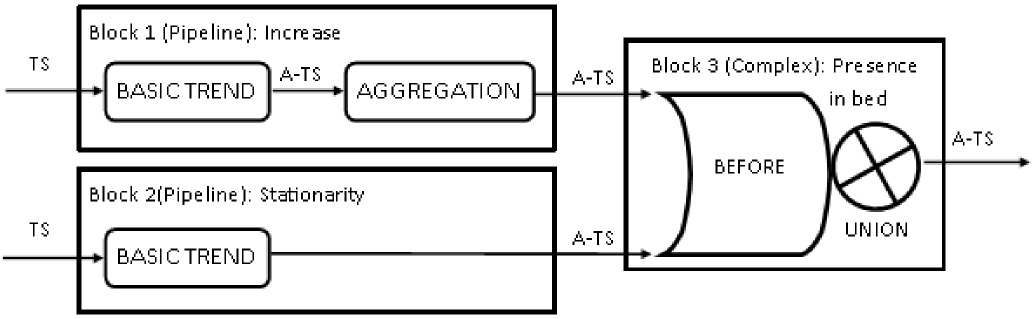

Block 1 (Pipeline): Increase
TS
BASIC TREND
A-TS
AGGREGATION
A-TS
Block 2(Pipeline): Stationarity
TS
BASIC TREND
A-TS
Block 3 (Complex): Presence in bed
BEFORE
UNION
A-TS

Figure 21: JTSA workflow for detecting presence in bed

Block 1 detects the time intervals in which an increase in pressure occur and is composed of 2 steps. The parameters of the algorithms in each step are shown in Table 9. STEP 1 uses the Basic Trend algorithm to detect the time intervals in which the pressure increases by at least 50 units in 4 seconds. This step returns a time series of episodes, one for each detected time interval longer than 4 seconds. STEP 2 uses the Aggregation Level algorithm and aggregates the episodes having the same label that are no more than 15 seconds apart.

Block 2 is composed of a single step, that uses the Basic Trend algorithm to detect the time intervals in which the pressure value is stationary.

Block 3 is a complex block that combines the results obtained from Block 1 and Block 2. In particular, it detects the complex pattern *presence in bed*, composed of an episode of increase in pressure followed by an episode of stationarity.

Table 9: Parameters of the JTSA workflow for detecting presence in bed

BLOCK	STEP	ALGORITHM	PARAMETERS

Block 1	Step 1	Basic Trend	minLen=4
			label=Increasing
			minSlope=20
			maxSlope=200
			gap=240
			granularity=SECONDS

	Step 2	Aggregation Level	gap=15
			minLen=1
			granularity=SECONDS

Block 2	Step1	Basic Trend	minLen=1
			label=Stationary
			minSlope=-5
			maxSlope=20
			gap=240
			granularity=SECONDS

Block 3	-	Operator:
		BEFORE
		Combiner:
		UNION
			ls=30
			rs=43200
			gap=20
			granularity=SECONDS

Once the JTSA workflow has been run for each section, the conversion from time intervals to events is applied to the returned time series of episodes. We obtain three time series of events, TS1, TS2, and TS3.

Finally, logical operators are applied to TS1, TS2, and TS3 to distinguish the time intervals in which the subject is sitting on the bed from the time intervals in which he/she is lying on the bed. In particular, we define *presence on the bed* as presence on at least one section. Thus, it is detected as TS1 OR TS2 OR TS3. After analyzing the response of the pressure mats to experiments simulating presence in bed, we assume that lying on the bed activates all the sections. Thus, the pattern *lying on the bed* is detected as TS1 AND TS2 AND TS3. The time intervals that verify TS1 OR TS2 OR

TS3 and do not verify TS1 AND TS2 AND TS3 represent the time intervals in which the subject is sitting on the bed.

Pattern 4: Permanence in the rooms

This pattern is based on two assumptions. The first assumption is that the subject is present within a room if movement is detected within that room or if presence in bed is detected within that room. The second assumption is that the subject cannot be in two rooms at the same time. Thus, we combine the results obtained by running *Pattern 1: Movement within the room* for all the rooms in the house and *Patter 3: presence in bed* for the bedroom, to identify, for each room, the time intervals in which the subject stays in that room. This pattern also allows to detect when the subject is absent from the house.

The sequence of operations to detect the *permanence in the rooms* pattern are shown in Figure 22.

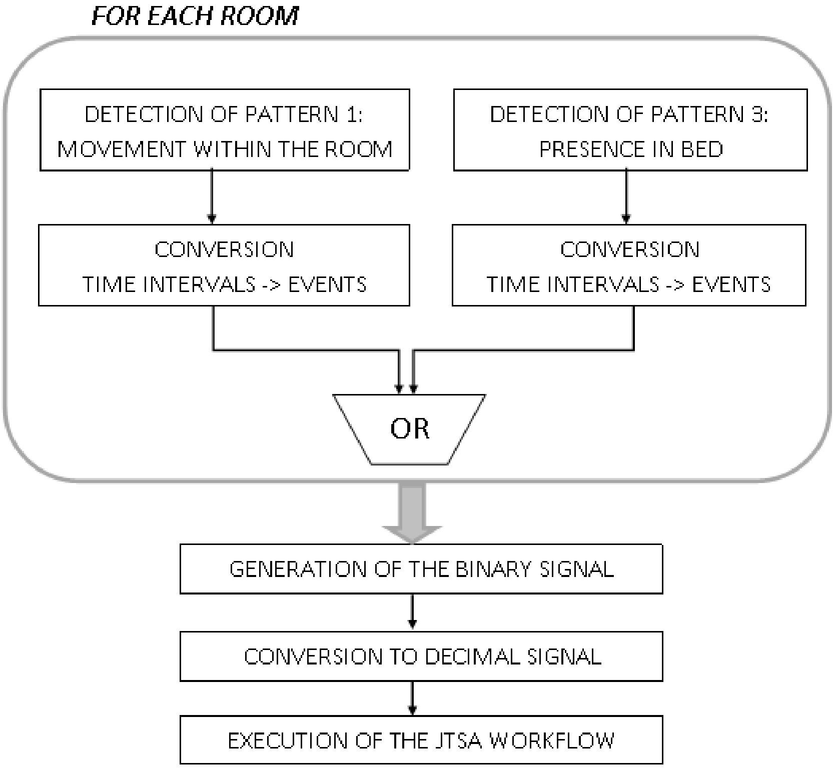

FOR EACH ROOM
DETECTION OF PATTERN 1: MOVEMENT WITHIN THE ROOM
DETECTION OF PATTERN 3: PRESENCE IN BED
CONVERSION TIME INTERVALS -> EVENTS
CONVERSION TIME INTERVALS -> EVENTS
OR
GENERATION OF THE BINARY SIGNAL
CONVERSION TO DECIMAL SIGNAL
EXECUTION OF THE JTSA WORKFLOW

Figure 22: Sequence of operations to detect permanence in the rooms

First, for each room we detect the time intervals that verify *Pattern 1*, as previously described. We then apply the conversion from time intervals to events to the returned time series of episodes to obtain a time series of events, which we will refer to as *T_mov*.

In parallel, for each room having a bed or chair equipped with pressure mats, we detect presence in bed. Again, we apply the conversion from time intervals to events to obtain a time series of events, which we will refer to as *T_bed*.

For each room, we apply the OR operator to *T_mov* and *T_bed*. Potentially, the obtained time series, *T_pres,* identifies presence within the room, if presence is not detected in other rooms.

In the following step we create a signal that conveys the information on *T_pres* for all the rooms. In particular, we create a signal by concatenating, time instant by time instant, the binary value obtained from each room. For example, for a house having three rooms (*R1, R2* and *R3*), if at a specific time instant (T_1) presence is detected in R1, absence in R2 and absence in R3, the value of this signal will be 001. The binary value (e.g., 001), is then converted into a decimal signal (e.g., 001 translates to 2), which is processed through a JTSA workflow. Of course, the list of the rooms of the house is stored in a configuration file, that is filled in when the NONCADO system is set up.

The JTSA workflow aggregates consequent time intervals of presence in a room, when they are separated by short time intervals that verify absence, in which no presence is detected elsewhere (Figure 23).

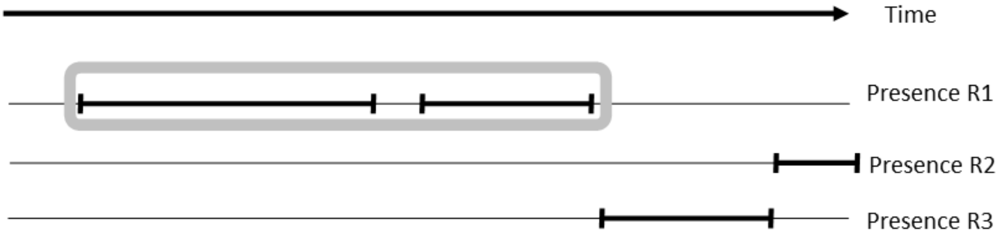


Figure 23: Representation of the aim of the JTSA workflow for detecting presence in the rooms

To this aim, the JTSA workflow shown in Figure 24 detects the following complex pattern:

(presence in R1, followed by absence) followed by presence in R1

where *absence* indicates absence from all the rooms.

The workflow is composed of 4 blocks, 2 pipelines and two complex blocks. It must be run separately for each room, since the parameters are room-dependent. For each room the workflow identifies three kinds of time intervals:

- The time intervals in which the subject is in the selected room
- The time intervals in which the subject is in another room, that is not specified
- The time intervals in which the subject is outside the house.

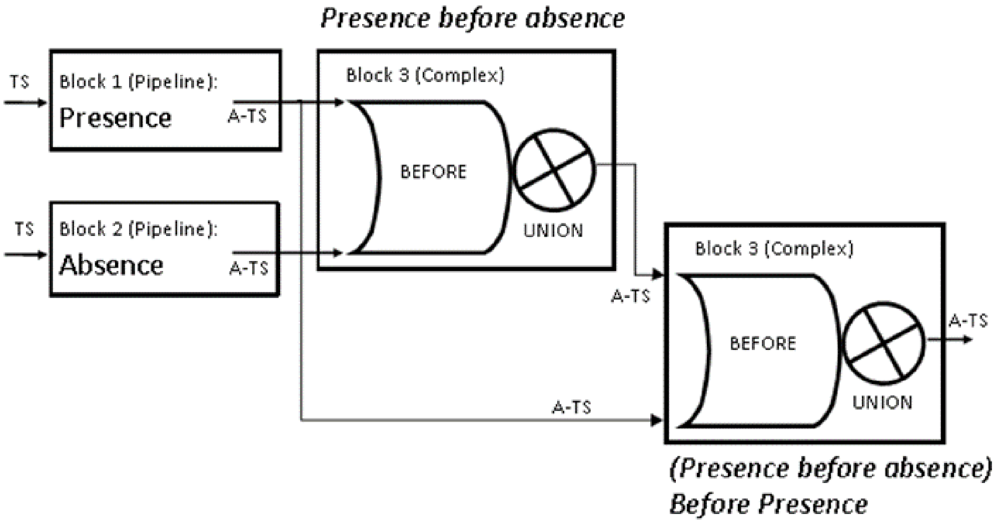


Figure 24: JTSA workflow for detecting permanence in the rooms

Block 1 is the pipeline block that identifies presence in the selected room and is composed of 3 steps. STEP 1 uses the Basic Qualitative algorithm and applies a qualitative abstraction to the decimal signal. The *absence* label indicates absence from all the rooms of the house, while *presence* denotes presence within the selected room. The *overlap* label indicates detection of simultaneous presence in several rooms of the house, including the selected one. This should not happen, unless the subject is moving from one room to another and the rooms are contiguous. The *other* label identifies presence in one (or more rooms) other than the one examined. The set of thresholds used for assigning the labels depends on the room. For example, if we consider two rooms, R1 and R2, and we select R1, the decimal signal will have the following values:

- 0 - absence, since the corresponding binary signal is 00;
- 1 - presence in room R1, since the corresponding binary signal is 01;
- 2 - presence in room R2, since the corresponding binary signal is 10;
- 3 - presence in both rooms, thus *overlap*, since the corresponding binary signal is 11.

The parameters set for the JTSA workflow when it runs for R1 (in a house having maximum 10 rooms equipped with sensors) are shown in Table 10.

Table 10: Parameters of the JTSA algorithm for detecting permanence in the rooms

BLOCK	STEP	ALGORITHM	PARAMETERS

Block 1	Step 1	Qualitative	th=0.1,1.1,2.1,3.1,4.1,5.1,6.1
			label=absence,presence,other,overlap, other,overlap,other,overlap

	Step 2	Aggregation Level	gap=20
			minLen=2
			granularity=SECONDS

	Step 3	Aggregation HighLevel	gap=60
			minLen=1
			granularity=SECONDS
			label=PresenceInRoom1
			levels=presence

Block 2	Step 1	Qualitative	th=0.1,1.1,2.1,3.1,4.1,5.1,6.1
			label=absence,presence,other,overlap, other,overlap,other,overlap

	Step 2	Aggregation Level	gap=20
			minLen=2
			granularity=SECONDS

	Step 3	Aggregation HighLevel	gap=60
			minLen=1
			granularity=SECONDS
			label=Absence
			levels=absence

Block 3	-	Operator: BEFORE
		Combiner: UNION
			ls=86400
			rs=86400
			gap=20
			granularity=SECONDS

Block 4	-	Operator: BEFORE
		Combiner: UNION
			ls=86400
			rs=86400
			gap=10
			granularity=SECONDS

Pattern 5: Cooking

This pattern allows to identify when subject uses the stove for preparing a hot meal and is detected through the JTSA workflow in Figure 25.

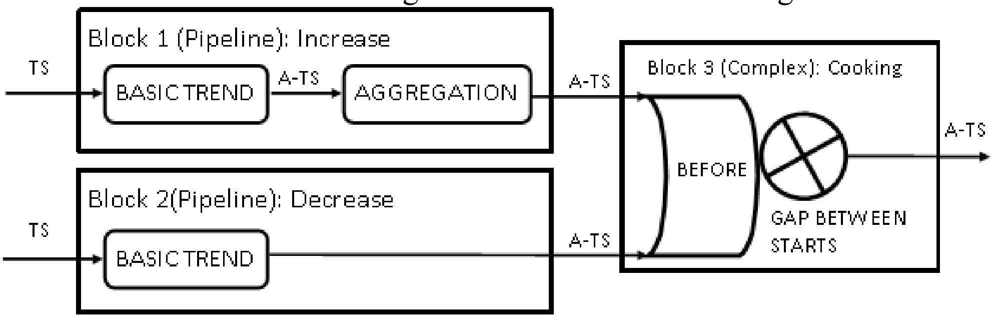

Block 1 (Pipeline): Increase
TS
BASICTREND
A-TS
AGGREGATION
A-TS
Block 3 (Complex): Cooking
BEFORE
GAP BETWEEN STARTS
A-TS
Block 2(Pipeline): Decrease
TS
BASICTREND
A-TS

Figure 25: Schema of the JTSA workflow for detecting cooking activity

The JTSA workflow is fed with the time series of measurements collected by the temperature sensor positioned at 60 cm from the stove. Since cooking causes a peak in the values measured by this sensor, the cooking activity is detected as a complex pattern, composed of an increase in temperature, followed by a decrease in temperature. The workflow is composed of three blocks.

The first block is the pipeline that detects the time intervals in which the temperature increases. The second block is the pipeline that detects the time intervals in which the temperature decreases. The third block is a complex block that detects the complex pattern, i.e., an increase followed by a decrease. The parameters of the algorithms used in each block are listed in Table 11.

Table 11: Parameters of the JTSA algorithm for detecting the cooking activity

BLOCK	STEP	ALGORITHM	PARAMETERS

Block 1	Step 1	Basic Trend	minLen=10
			label=IncreasingTemp
			minSlope=0.032
			maxSlope=1000
			gap=300
			granularity=SECONDS

	Step 2	Aggregation Level	gap=300
			minLen=1
			granularity=SECONDS

Block 2	Step 1	Basic Trend	minLen=12
			label=DecreasingTemp
			minSlope=-100
			maxSlope=-0.03
			gap=180
			granularity=SECONDS

Block 3	-	Operator:
		BEFORE
		Combiner:
		GAP BETWEEN STARTS
			ls=3600
			rs=3600
			gap=600
			granularity=SECONDS

Pattern 6: Bath/Shower

The bath or shower activity is associated with an increase in the relative humidity in the bathroom, that can be detected by a humidity sensor placed close to the shower box. The time series of measurements collected by the sensors are fed to a JTSA workflow composed of a single pipeline. The pipeline is composed of a single step, which applies the Basic Trend algorithms, with the following parameters, set according to the sensor's response to stimuli designed to simulate the shower activity.

minLen=10
label=IncreasingHum
minSlope=0.1
maxSlope=3
gap=40
granularity=SECONDS

In this case, the sensor's response to the activity is not immediate. Thus, we cannot identify exactly the instants in which the activity begins or ends. However, we can assess whether or not it occurred within the day.

Pattern 7: Washing the dishes

Initially, we supposed that washing the dishes would cause an increase in the humidity level in the area next to the sink. We formalized a workflow to detect such increase in the time series of measurements collected by a humidity sensor located next to the sink. However, a set of testes proved that such increase is not very significant, and that the response of the sensor is too slow to detect it. Thus, to detect this activity we use a PIR sensor located next to the sink, having a visual field limited to the sink itself.

The time series of measurements are fed to a JTSA workflow in composed by a single pipeline, made of three steps. This JTSA workflow works exactly as the JTSA workflow for detecting movement within a room (Pattern 1 in this chapter), with slightly different thresholds used when aggregating the time intervals in which movement occurs. The parameters of the algorithm in each step are listed in Table 12.

Table 12: Parameters of the JTSA algorithm for detecting the *Washing the dishes* activity

STEP	ALGORITHM	PARAMETERS

Step 1	Qualitative	th=0.5
		label=absence,movement

Step 2	Aggregation Level	gap=60
		minLen=1
		granularity=SECONDS

Step 3	Aggregation HighLevel	gap=60
		minLen=5
		granularity=SECONDS
		label=WashingDishes
		levels=movement

Pattern 8: Environmental discomfort (Temperature)

According to the literature, the optimal range of environmental temperature for a subject aged over 65 years is between 18°C and 26°C. We use a JTSA workflow to detect the time intervals in which the temperature is not in this range. If the room is equipped with multiple temperature sensors, excluding the sensor that monitors the area next to the stove, the workflow must be run for each sensor and the results must be combined by applying the OR operator.

Again, the workflow has the same structure as the workflow for detecting movement within the room, with a different set of parameters (Table 13). In particular, it consists of three steps. STEP 1 applies the Basic Qualitative algorithm to perform a qualitative abstraction to the time series of temperature events. Specifically, all values lower than 18°C are assigned the *LOW* label, those above 26°C are labelled as *HIGH*, and the remaining values are labelled as *OK*. STEP 2 uses the Aggregation Level algorithm and aggregates the episodes having the same label. STEP 3 uses the Aggregation HighLevel algorithm and performs a further aggregation, selecting those having the *LOW* or *HIGH* label, to detect the time intervals in which the temperature is not ideal for at least 5 minutes.

Table 13: Parameters of the JTSA workflow for detecting inadequate environmental temperature

STEP	ALGORITHM	PARAMETERS

Step 1	Qualitative	th=18,26
		label=LOW,OK,HIGH

Step 2	Aggregation Level	gap=15
		minLen=2
		granularity=SECONDS

Step 3	Aggregation HighLevel	gap=15
		minLen=300
		granularity=SECONDS
		label=UncomfortableTemperature
		levels=LOW,HIGH

Pattern 9: Environmental discomfort (Humidity)

According to the literature, the optimal range of environmental humidity for a subject aged over 65 years is between 40% and 65%. We use a JTSA workflow to detect the time intervals in which humidity is not in this range. It works as the JTSA workflow for detecting temperature discomfort, with a slightly different set of thresholds in STEP 1 (Table 14).

Table 14: Parameters of the JTSA workflow for detecting inadequate environmental humidity

STEP	ALGORITHM	PARAMETERS

Step 1	Qualitative	th=40,65
		label=LOW,OK,HIGH

Step 2	Aggregation Level	gap=15
		minLen=2
		granularity=SECONDS

Step 3	Aggregation HighLevel	gap=15
		minLen=300
		granularity=SECONDS
		label=UncomfortableHumidity
		levels=LOW,HIGH

4.1.3. Decision support

The detection of the patterns described in the previous section provides an overview of the activities performed by the subject over time. Based on the results of the analysis, for each day we can assess which use cases (Table 6) were verified, and we can produce the contents of the daily and weekly report for the subject's family.

The first section of the daily report contains the warnings related to any risky behavior detected during the day (Table 3). For each considered behavior, Table 15 describes how it is detected. The *condition* column describes the condition that triggers the generation of the warning. The *patterns* column lists the patterns that are search for to check if the condition is verified.

Table 15: Patterns used to detect the behaviors to correct

BEHAVIOR TO DETECT	CONDITION	PATTERNS
Moving within the house in poor light conditions	USE CASE 5 (ACTIVITY) activates when USE CASE 9 (POOR LIGHT CONDITIONS) is active.	Pattern 2: Movement within the room (in poor light conditions)
Getting up from the bed in poor light conditions	USE CASE 1 (AWAKENING) and USE CASE 5 (ACTIVITY) activate when USE CASE 9 (POOR LIGHT CONDITIONS) is active.	Pattern 2: Movement within the room (in poor light conditions) Pattern 3: Presence in bed
Exiting the house during the night	The photocell positioned close to the external door reports an exit during the night hours and absence from the house is detected for more than a set duration.	Pattern 4: Permanence in the rooms
Taking a bath/shower during the night	USE 7 (BATH/SHOWER) activates during the night hours.	Pattern 6: Bath/Shower

Exiting the house while cooking (and using the stove)	The photocell in the kitchen reports an exit when USE CASE 4 (HOT MEAL) is activated.	Pattern 5: Cooking
Absence of cooking activity, i.e., not having hot meals	USE CASE 4 (HOT MEAL) does not activate for more than a set duration.	Pattern 5: Cooking
Spending time in a room with inadequate humidity and/or temperature conditions	In a selected room, USE CASE 10 (ENVIRONMENTAL DISCOMFORT) activates when presence in that room is detected.	Pattern 4: Permanence in the rooms Pattern 8: Environmental discomfort (Temperature) Pattern 9: Environmental discomfort (Humidity)

In a second section, the daily report provides a summary of the activities performed by the subject during the day. In particular, the total time spent by the subject within the home is estimated as the total duration of the time intervals that verify *Pattern 4: Permanence in the rooms*. By running the same pattern and extracting the time intervals labeled as *absence*, we assess the total time by the subject outside the home.

By analyzing the occurrences of *Pattern 3: Presence in bed* we can assess the number of times the subject gets up during the night, and the total time spent in bed, during the day and the night. By default, we define *night* as the time between 22:00 and 6:00, but the definition is customizable.

We also run *Pattern 5 (Cooking)*, *Pattern 6 (Bath/Shower)*, and *Pattern 7 (Washing the dishes)* and report if the corresponding activities occurred during the considered day.

As anticipated, the weekly report lists any changes identified with respect to the previous weeks, i.e., the deviations from the patient's profile (Table **5**) For each possible deviation, Table 16 describes how it is detected.

As in the previous table, the condition *column* describes the condition that triggers the generation of the warning. The *patterns* column lists the patterns that are search for to check if the condition is verified. The terms *increase* (or *decrease*) means an increase (or decrease) of at least *X* percent. with *X* being a customizable threshold. At the beginning of each condition <<*Compared to previous weeks*>> is implied.

Table 16: Patterns used to detect deviations from the subject's profile

DEVIATION TO DETECT	CONDITION	PATTERNS
Increase in night-time awakenings	The number of times USE CASE 2 (NIGHT_TIME AWAKENING) activation increases.	Pattern 3: Presence in bed
Increase/decrease in resting time	The duration of the occurrence of USE CASE 6 (RESTING) increases/decreases.	Pattern 3: Presence in bed
Increase/decrease in time spent in specific rooms (e.g., kitchen, bathroom)	For a selected room the duration of the occurrence of *Pattern 4: Permanence in the rooms* increases/decreases.	Pattern 4: Permanence in the rooms
Decrease in time spent outside the house	The duration of the time intervals in which absence is decreased.	Pattern 4: Permanence in the rooms
Decrease of cooking activity	The number of times USE CASE 4 (HOT MEAL) activation decreases.	Pattern 5: Cooking
Decrease in bath/shower activities	The number of times USE CASE 7 (BATH/SHOWER) activation decreases.	Pattern 6: Bath/Shower
Decrease in physical activity	Either the duration of USE CASE 5 (ACTIVITY) decreases, or the number of minutes of activity recorded by the Fitbit tracker decreases.	Pattern 1: Movement within the room
Decrease in walking speed	The walking speed provided by the sensorized carpet decreases.	-
Decrease in sleep quality (e.g. decrease in sleep efficiency)	Either the duration of sleep and/or the sleep efficiency provided by the Fitbit activity tracker decreases.	-

4.2. Pilot study

The system underwent two test phases. The first test phase aimed at tuning the parameters of the JTSA workflows and assessing the functionality of the system. The tests were carried out on healthy volunteers in an environment set up within our department. Following the preliminary study on healthy volunteers, in September 2018 a prototype of the system was tested in a 2-weeks pilot study involving 16 patients with history of falls treated at the CCPP Hospital. The study was approved by the Ethical Committee of Fondazione IRCCS Cà Granda Area 2, Milan, Italy (nr 570_2018bis) and was held in a dedicated environment within the hospital. The following two sections describe these two test phases.

4.2.1. Preliminary tests on volunteers

The network of sensors was installed in two rooms of our department, arranged to simulate a home. One room was equipped with a bed and one room was equipped with a kitchen stove to simulate a bedroom and a kitchen, respectively. The volunteers were enrolled among the students and researchers of the department.

This test phase was divided into two sub-phases. In the first phase, the volunteers were asked to perform specific atomic activities, designed to trigger the patterns described in Section 4.1.2. More than to test the system's performance, these tests were aimed at tuning the parameters for the JTSA workflows described in Section 4.1.2. Table 17 lists the functional tests performed. The first column specifies the activity performed and, in brackets, the patterns it should trigger. The second and third columns contain the start and end date of the activity. The start and end dates of the occurrences of the patterns detected by the CDSS were compared with these two dates, noted by an external observer.

Table 17: Preliminary functional tests on healthy volunteers

ACTIVITY (PATTERNS)	START	STOP
Movement within the room in appropriate light conditions (*Movement within the room, Permanence in the rooms)*	17/10/2017 11:46:50	17/10/2017 11:49:00
	17/10/2017 12:16:34	17/10/2017 12:18:34
Sitting on the bed (*Presence in bed*, *Permanence in the rooms*)	17/10/2017 11:49:00	17/10/2017 11:54:00
	17/10/2017 12:18:34	17/10/2017 12:23:34
	23/10/2017 10:20:00	23/10/2017 10:40:00

	25/10/2017 15:50:00	25/10/2017 16:00:00
	25/10/2017 16:00:00	25/10/2017 16:10:00
	07/02/2018 09:43:03	07/02/2018 10:01:00
	07/02/2018 11:40:00	07/02/2018 11:50:00
	16/02/2018 09:58:00	16/02/2018 10:13:00
	16/02/2018 12:05:00	16/02/2018 12:10:00
	20/02/2018 11:40:36	20/02/2018 11:50:36
	20/02/2018 12:05:00	20/02/2018 12:15:00
Switch from one room to another (*Permanence in the rooms*)	17/10/2017 11:54:00	17/10/2017 11:58:00
	17/10/2017 12:24:00	17/10/2017 12:26:18
Movement within the room in poor light conditions (*Movement within the room in poor light conditions, Permanence in the rooms)*	19/10/2017 18:01:29	24/10/2017 16:00:23
	20/02/2018 16:57:03	20/02/2018 17:09:04
Sitting on the bed, getting up, and sitting on the bed again (*Presence in bed*)	17/10/2017 11:58:53	17/10/2017 12:00:20
	23/10/2017 10:42:00	23/10/2017 11:03:00
Moving within the room, exiting, and entering again (*Permanence in the rooms*)	17/10/2017 12:08:40	17/10/2017 12:14:18
	17/10/2017 12:38:30	17/10/2017 12:42:50
	31/10/2017 10:47:00	31/10/2017 11:00:00
	31/10/2017 11:05:00	31/10/2017 11:18:00
Sitting on the bed, then lying on the bed (*Presence in bed*)	23/10/2017 11:07:00	23/10/2017 11:19:00
	25/10/2017 15:09:00	25/10/2017 15:39:55
	07/02/2018 10:20:00	07/02/2018 10:40:00
	16/02/2018 10:35:00	16/02/2018 10:55:00
	20/02/2018 17:15:00	20/02/2018 17:35:00
Movement across multiple rooms (*Permanence in the rooms*)	31/10/2017 11:54:00	31/10/2017 12:00:00
	31/10/2017 12:06:00	31/10/2017 12:12:00
	31/10/2017 12:25:00	31/10/2017 12:30:00
	31/10/2017 12:33:00	31/10/2017 12:35:00
Boiling a pot of water (*Cooking*)	09/11/2017 09:49:01	-
	09/11/2017 10:46:51	-
	09/11/2017 10:55:55	-
	09/11/2017 11:05:45	-

	09/11/2017 11:17:36	-
Washing a dish (*Washing the dishes*)	11/12/2017 09:49:39	-
	11/12/2017 10:00:23	-
	11/12/2017 10:11:16	-
	11/12/2017 10:48:44	-
	11/12/2017 11:01:58	-
Washing a dish in poor light conditions (*Washing the dishes, Movement within the room in poor light conditions*)	11/12/2017 11:55:27	-
	11/12/2017 12:00:09	-
	11/12/2017 12:08:00	-
	11/12/2017 12:11:22	-
Getting up from the bed in poor light conditions (*Movement within the room in poor light conditions, Presence in bed*)	16/02/2018 11:15:00	16/02/2018 11:25:00
	20/02/2018 11:10:00	20/02/2018 11:20:00

The second test phase on the volunteers aimed at assessing the performance of the NONCADO system in daily use. During the test, we asked the volunteers to simulate a sequence of activities of daily living, organized in sessions lasting about 2 hours. The subject was asked to manually note the details (i.e., start time and end time, mainly) of each activity performed. Finally, we compared the events detected by the CDSS with the subject's record.

For each session, Table 18 provides a textual description of the performed activities, their start time and end time as reported by the volunteer. The *CDSS* column reports whether the detection by the CDSS was successful (*OK*) or not (*NO*).

Table 18: Real life simulation test performed by healthy volunteers

SESSION	ACTIVITY	START	END	CDSS
27/03/2018	PRESENCE IN BED (LYING DOWN)	10:56:00	11:03:00	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	11:03:00	11:05:00	OK
	PRESENCE IN BED (LYING DOWN)	11:05:00	11:12:00	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	11:12:00	11:14:00	OK
	PRESENCE IN BED (LYING DOWN)	11:14:00	11:16:00	OK
	PRESENCE IN THE KITCHEN	11:16:00	11:56:00	OK
	ABSENCE	11:56:00	12:06:00	NO
	PRESENCE IN THE KITCHEN	12:06:00	12:55:00	OK
14/06/2018	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:58:40	11:09:00	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	11:14:40	11:25:00	OK
	PRESENCE IN THE KITCHEN	11:30:00	12:00:00	OK

	ABSENCE	12:00:00	12:15:00	OK
	PRESENCE IN THE KITCHEN	12:15:00	12:30:00	OK
	COOKING	12:18:00	12:23:00	OK
	PRESENCE IN BED (SITTING)	12:30:40	12:35:40	NO
	PRESENCE IN BED (LYING DOWN)	12:35:40	12:40:40	OK
	EXITING THE KITCHEN WHILE COOKING	12:43:00	12:51:00	OK
	PRESENCE IN THE KITCHEN	12:51:00	13:01:00	OK
14/06/2018	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	15:35:40	15:46:00	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	15:51:40	16:02:30	OK
	PRESENCE IN THE KITCHEN	16:07:30	16:37:30	OK
	ABSENCE	16:37:30	16:53:30	OK
	PRESENCE IN THE KITCHEN	16:53:30	17:08:30	OK
	COOKING	16:55:00	17:00:00	OK
	PRESENCE IN BED (SITTING)	17:09:30	17:14:30	NO
	PRESENCE IN BED (LYING DOWN)	17:14:30	17:19:30	OK
	PRESENCE IN THE KITCHEN	17:20:00	17:30:00	OK
	MOVEMENT IN POOR LIGHT CONDITIONS	17:30:00	17:40:00	OK
15/06/2018	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:15:10	10:20:50	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:24:40	10:32:10	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:36:00	10:41:50	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:45:20	10:49:20	OK
	PRESENCE IN THE KITCHEN	10:50:00	11:35:00	OK
	PRESENCE IN BED (LYING DOWN)	11:36:00	11:56:00	OK
	PRESENCE IN THE KITCHEN	11:56:00	12:06:00	OK
15/06/2018	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	15:28:00	15:33:30	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	15:37:00	15:44:30	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	15:48:00	15:53:50	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	15:57:30	16:01:30	OK
	PRESENCE IN THE KITCHEN	16:02:00	16:47:00	OK
	PRESENCE IN BED (LYING DOWN)	16:48:00	17:08:00	OK
	PRESENCE IN THE KITCHEN	17:09:00	17:19:00	OK
18/06/2018	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:21:40	10:29:00	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:32:00	10:37:40	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:41:20	10:47:00	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:51:00	10:55:00	OK
	PRESENCE IN THE KITCHEN	10:55:50	11:40:50	OK
	PRESENCE IN BED (LYING DOWN)	11:42:00	12:02:00	OK
	PRESENCE IN THE KITCHEN	12:04:20	12:14:20	OK
25/06/2018	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	14:53:45	15:05:30	OK

	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	15:11:45	15:23:00	OK
	PRESENCE IN THE KITCHEN	15:28:00	16:00:20	OK
	ABSENCE	16:00:20	16:15:20	OK
	PRESENCE IN THE KITCHEN	16:15:40	16:32:20	OK
	PRESENCE IN BED (SITTING)	16:33:10	16:38:40	OK
	PRESENCE IN BED (LYING DOWN)	16:38:40	16:44:15	OK
	PRESENCE IN THE KITCHEN	16:45:15	16:55:15	OK
	MOVEMENT IN POOR LIGHT CONDITIONS	16:55:20	17:05:30	OK
26/06/2018	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:13:25	10:24:30	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:30:55	10:41:55	OK
	PRESENCE IN THE KITCHEN	10:46:55	11:17:00	OK
	ABSENCE	11:17:00	11:32:00	OK
	PRESENCE IN THE KITCHEN	11:32:00	11:48:45	OK
	PRESENCE IN BED (SITTING)	11:49:20	11:54:30	OK
	PRESENCE IN BED (LYING DOWN)	11:54:30	11:59:55	OK
	PRESENCE IN THE KITCHEN	12:00:55	12:29:55	OK
26/06/2018	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	14:57:30	15:08:30	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	15:14:25	15:25:15	OK
	PRESENCE IN THE KITCHEN	15:30:15	16:00:30	OK
	ABSENCE	16:00:30	16:15:05	OK
	PRESENCE IN THE KITCHEN	16:15:05	16:31:10	OK
	PRESENCE IN BED (SITTING)	16:31:50	16:36:50	OK
	PRESENCE IN BED (LYING DOWN)	16:36:50	16:43:10	OK
	PRESENCE IN THE KITCHEN	16:44:00	16:53:25	OK
	MOVEMENT IN POOR LIGHT CONDITIONS	16:53:25	17:03:40	OK
27/06/2018	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:25:30	10:31:20	NO
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:35:05	10:43:00	NO
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:47:00	10:52:40	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:56:25	10:59:35	OK
	PRESENCE IN THE KITCHEN	11:00:20	11:45:50	OK
	PRESENCE IN BED (LYING DOWN)	11:46:35	12:06:55	OK
	PRESENCE IN THE KITCHEN	12:07:40	12:19:25	OK
27/06/2018	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	14:28:15	14:34:00	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	14:37:35	14:45:20	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	14:49:10	14:54:55	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	14:59:00	15:03:05	OK
	PRESENCE IN THE KITCHEN	15:03:55	15:51:10	OK
	PRESENCE IN BED (LYING DOWN)	15:51:55	16:11:35	OK
	PRESENCE IN THE KITCHEN	16:12:15	16:22:00	OK
28/06/2018	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	14:39:30	14:47:10	OK

	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	14:54:10	15:01:40	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	15:05:45	15:11:30	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	15:15:05	15:19:05	OK
	PRESENCE IN THE KITCHEN	15:19:40	16:05:00	OK
	PRESENCE IN BED (LYING DOWN)	16:05:45	16:25:30	OK
	PRESENCE IN THE KITCHEN	16:26:10	16:26:30	OK
24/07/2018	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	10:55:00	11:05:00	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	11:11:00	11:22:00	OK
	PRESENCE IN THE KITCHEN	11:28:00	11:58:00	OK
	ABSENCE	11:58:00	12:13:00	OK
	PRESENCE IN THE KITCHEN	12:13:00	12:28:00	OK
	PRESENCE IN BED (SITTING)	12:29:00	12:34:00	OK
	PRESENCE IN BED (LYING DOWN)	12:34:00	12:39:00	OK
	PRESENCE IN THE KITCHEN	12:40:00	12:50:00	OK
	MOVEMENT IN POOR LIGHT CONDITIONS	12:50:00	13:00:00	OK
24/07/2018	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	14:32:00	14:43:00	OK
	PRESENCE IN BED (LYING DOWN) AND EXIT FROM THE ROOM	14:49:00	15:00:00	OK
	PRESENCE IN THE KITCHEN	15:05:00	15:35:00	OK
	ABSENCE	15:35:00	15:50:00	OK
	PRESENCE IN THE KITCHEN	15:51:00	16:06:00	OK
	PRESENCE IN BED (SITTING)	16:07:00	16:12:00	OK
	PRESENCE IN BED (LYING DOWN)	16:12:00	16:17:00	OK
	PRESENCE IN THE KITCHEN	16:18:00	16:38:00	OK

According to Table 18, the obtained results were positive; only 5 actions (in grey in the table) were not correctly verified by the system. In general, the errors are due to incorrect calibrations of the sensors (in particular, the pressure mats) or to a sub-optimal tuning of the parameters in the JTSA workflows. For example, one error occurred in detecting a short time interval of absence that occurred during a long time interval of presence within the room. Such error was due to a non-optimal calibration of the aggregation algorithms used by the JTSA workflow for identifying presence in the rooms (Section 4.1.2). Following the described test sessions, the parameters were recalibrated, and all the errors were corrected.

4.2.2. Pilot study on the real-world patient population

For the pilot study we enrolled 16 patients aged over 65 years (6 females and 10 males; age: 72.69 ± 8.53 years) being prescribed occupational therapy at the CCPP hospital as rehabilitation after a fall event. All the subjects were affected by a neurological disease (i.e., hemiparesis, medullary paraparesis, medullary or cerebellar tetra paresis, cerebral hemorrhage, polyneuropathy, or Parkinson's disease), in sub-acute or chronic stage. As required by the inclusion criteria, all the subjects were able to walk. The pilot study was carried out during rehabilitation sessions held in the *Living Lab*, i.e., an environment specifically devoted to practice activities of daily living with a therapist. The next three section describes, respectively, the layout of the Living Lab, the protocol of the pilot study, and the obtained results.

The Living Lab

Figure 26 shows the layout of the Living Lab and the layout of the sensors installed within this environment.

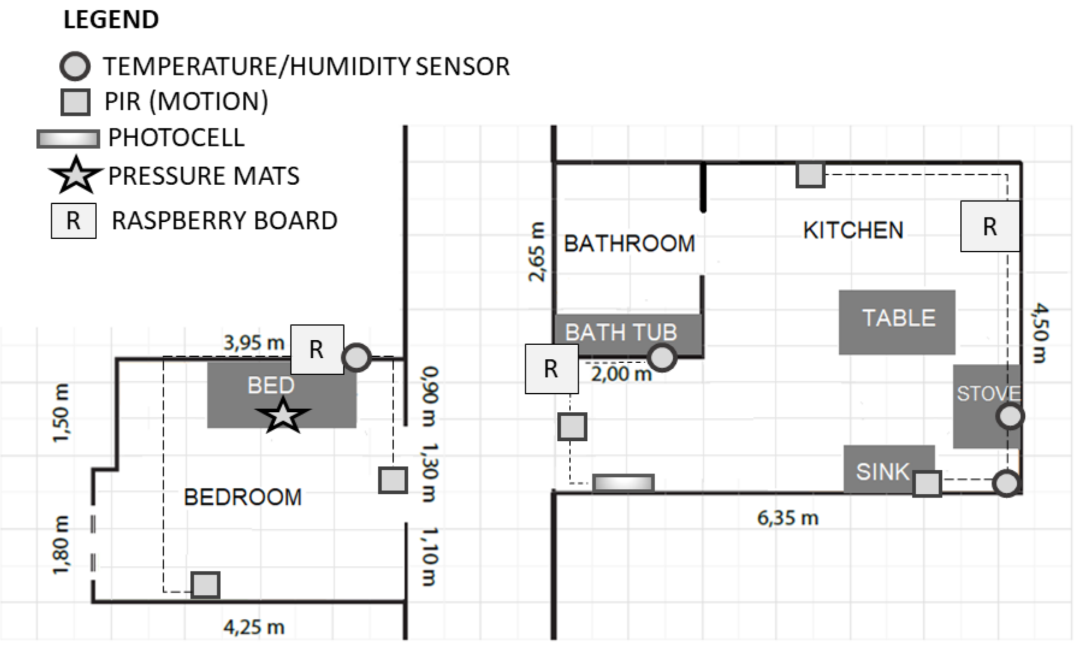


Figure 26: Layout of the Living Lab

The Living Lab consists of two rooms, one bedroom and one kitchen, separated by a corridor.

The bedroom contains a single bed, that we equipped with the three pressure mats, as described in Section 4.1.2. To detect movement within the bedroom, we placed two PIRs sensors: one on the door and one in front of the bed. The bedroom was also equipped with a temperature sensor and a humidity sensor. The sensors for measuring luminosity were not installed, since the patients were not allowed to perform any activity in poor light conditions, for safety reasons. All the sensors were connected to a Raspberry card, placed under the bed.

In the room simulating the kitchen two Raspberries were installed. One was located near the door and was configured with a motion sensor, a temperature sensor, a humidity sensor, and a photocell. The other one was positioned near the stove and was configured with two motion sensors (one of which was dedicated to identifying the *Washing the dishes* pattern) and two temperature sensors, one of which was dedicated to identifying the *Cooking* pattern.

The protocol

The tests were carried out during conventional sessions of occupational therapy under the supervision of a therapist. The expected duration of each session was 30 minutes. During the session, the subjects were asked to perform 5 macro-activities, to which we will refer to as *test use cases,* outlined in collaboration with the clinicians of the CCPP hospital to be compatible with the rehabilitation process. The set of test use cases included resting in bed, resting in bed with getting up, cooking, leaving the kitchen while cooking, and washing the dishes. To standardize the test for all the patients, we prepared a form (Table 19) that detailed the sub-activities that compose the 5 test use cases. For each test use case, the form lists the sub-activities that compose it. During each session, an observer was present in the room together with the patient and the therapist and was asked to fill in the form with the start time and end times of each activity performed by the subject.

Table 19: Form listing the activities performed by the patients in the pilot study

TEST USE CASE	ACTIVITIES	START TIME	END TIME
1. Resting in bed	The patient lies on the bed.		
	The patient gets up.		
	The patient lies back on the bed.		
	The patient gets up.		
2. Resting in bed with getting up (i.e., exiting the bedroom)	The patient lies on the bed.		
	The patient gets up.		
	The patient leaves the bedroom.		
	The patient enters the bedroom again.		
	The patient lies back on the bed.		
	The patient gets up.		
3. Cooking	The patient turns on the stove.		
	The patient turns off the stove.		
4. Exiting the kitchen while cooking	The patient turns on the stove.		
	The patient leaves the kitchen.		
5. Washing the dishes	The patient opens the tap.		

	The patient closes the tap.		

Every patient participated in two test sessions, to which we will refer to as *T1* and *T2*. The list of the sessions performed in the pilot study is presented in Table 20. For each session, the table provides the date, the subject identifier, the identifier of the test session (either *T1* or *T2*), the start time and end time of the session as recorded by the observer.

Table 20: List of the sessions in the pilot study

DATE	SUBJECT	TEST	START TIME	END TIME
04/09/2018	S1	T1	09:13	09:51
	S2	T1	09:57	10:27
	S3	T1	10:35	11:10
05/09/2018	S4	T1	09:06	09:38
	S5	T1	09:38	10:09
	S6	T1	10:10	10:47
	S7	T1	10:48	11:01
	S8	T1	11:05	11:30
06/09/2018	S1	T2	09:09	09:49
	S2	T2	09:49	10:20
	S3	T2	10:37	11:08
07/09/2018	S4	T2	10:13	10:41
	S5	T2	09:40	10:12
	S6	T2	11:15	11:45
	S7	T2	09:06	09:37
	S8	T2	10:44	11:12
	S10	T1	09:26	10:05
	S11	T1	10:15	10:42
	S12	T1	10:54	11:29
	S13	T1	11:36	12:03
12/09/2018	S14	T1	09:13	09:47
	S15	T1	09:52	10:24
	S16	T1	10:29	10:57
	S17	T1	11:05	11:35
13/09/2018	S10	T2	10:25	10:58
	S11	T2	11:14	11:45
	S12	T2	09:49	10:25
	S13	T2	09:01	09:35
14/09/2018	S14	T2	09:08	09:37
	S15	T2	10:48	11:18
	S16	T2	09:38	10:05
	S17	T2	10:16	10:45

In each session the subject completed each test use case once. However, one patient interrupted one session early due to a concomitant visit, thus completing only 3 test use cases out of 5.

Results and discussion

To evaluate the performance of the system, the forms were used as a gold standard. In particular, the results of the CDSS elaboration in terms of intervals of validity of each activity were compared to the details reported by the observer in the form. Each test use case was considered to be correctly detected if all its actions were correctly detected by the system. Tables in in Appendix C (Table 33 -Table 37) report in detail the performance of the CDSS in identifying the 5 test use cases.

In summary, the test use case 1 (Resting in bed) was performed 32 times and was correctly detected by the CDSS in 31 cases, thus reaching an accuracy equal to 97%. In one case, the first repetition of the sub-activity *lying in bed* was not correctly detected. In particular, the increase in pressure that determines the beginning of the subject's presence in bed was not detected. In fact, the increase in pressure lasted less than 10 seconds, thus being too short to be considered significant by the DSS. Afterwards, the value measured by the pressure mats decreased unexpectedly, although remaining high. For further insight, we provide the pressure values registered in the time interval in which the detection error occurred:

12/09/2018 11:07:21	306
12/09/2018 11:07:25	353
12/09/2018 11:07:29	**740**
12/09/2018 11:07:34	673
12/09/2018 11:07:38	673

As the reported measurements underline, the pressure profile showed a peak, in which the signal reached the value 740, before decreasing to 673. The observed spike was probably due to the intervention of the therapist, who had to help the patient position himself comfortably in the bed. The spike prevented the system from detecting the increase in the signal. To correctly manage similar cases, we have to implement filters for eliminating possible peaks from the signal before feeding it to the JTSA workflows.

The test use case 2 (Resting in bed with getting up) was performed 31 times and was correctly detected in 30 cases, reaching an accuracy equal to 97%. In one case, the second presence in bed was not correctly detected due to the lack of pressure measurements relating to the time interval in which the action was performed. The lack of measurements was probably due to connection problems, which prevented the pressure mats from sending the measurements.

The test use case 3 (Cooking) was performed 31 times and was correctly detected in 30 cases, reaching an accuracy equal to 97%. The error was due to malfunctioning of the temperature sensor placed next to the stove, which provided implausible values (i.e., -100 ° C), which were automatically discarded by the CDSS. Due to the lack of reliable temperature measurements in the time interval in which the subject was using the stove, the cooking activity was not detected.

Identifying *exiting the kitchen while cooking* was the most challenging task for the CDSS. In fact, the test use case 4 (Exiting the kitchen while cooking) was performed 32 times and was correctly detected in 25 cases, with accuracy equal to 78%. In 7 cases, the CDSS was not able to correctly detect at least one of the two activities that compose the test use case, i.e., the cooking activity and the exit from the room. In particular, in 3 cases a malfunction of the temperature sensor occurred, as described for the previous test use case. In 2 cases out of 3, this resulted in the lack of detection of the cooking activity itself. In the third case the cooking activity was detected, but with a delay. Thus, according to the system the subject left the kitchen before the beginning of the cooking activity. In the same repetition of the test use case, even the subject's exit from the kitchen was detected with a delay. This error was probably due to the exit of the therapist, which occurred shortly after the patient left. The CDSS erroneously aggregated the two consecutive exits (of the patient and of the therapist) into a single exit event, occurring when the second exit occurred. In 2 cases, which occurred at the end of the test day, the photocell log was written with a delay, which caused the most recent data to be lost when the system was shut down. In two cases, the patient's exit was not detected at all. In particular, the photocell detected movement, but it was unable to determine its direction, i.e., whether the patient was entering or exiting the kitchen. These errors were probably due to the presence of the therapist who had to assist the patient on the way out of the living lab for safety reasons. This highlighted one limitation of the NONCADO system, which is designed to monitor a subject living alone.

The other test use case that showed low accuracy was *Washing the dishes*. In particular, this test use case was performed 32 times and was correctly detected in 26 cases, leading to accuracy equal to 81%. However, 5 out of the 6 errors happened during the same test day and were due to a cable throttling, which prevented the communication between the motion sensor used for detecting the use case and the Raspberry PI board. The sixth error was due to a temporary connection problem, causing the sensor to fail sending the collected data, as well.

In general, as shown in Table 21, the system achieved good results, reaching an overall accuracy of 90% in detecting the activity performed by the patient during the occupational therapy.

Table 21: Accuracy of the NONCADO system in detecting the test use cases

TEST USE CASE	ITERATIONS	ITERATIONS DETECTED BY THE CDSS	ACCURACY
1. Resting in bed	32	31	97%
2. Resting in bed with getting up	31	30	97%
3. Cooking	31	30	97%
4. Exiting the kitchen while cooking	32	25	78%
5. Washing the dishes	32	26	81%
OVERALL	**158**	**142**	**90%**

This result was affected by the incorrect positioning (i.e., throttled cable) of one PIR sensor, and the consequent lack of monitoring data during a whole day of testing, which led to the impossibility of verifying the test use case 5 (*Washing the dishes*) for the all the study sessions held that day. The other activity that was critical to identify was leaving the kitchen while cooking, which influenced the accuracy for the test use case 4. This test use case was always carried out as the last activity of the session and often the patient was supposed to reach another hospital department immediately after this activity. For this reason, as previously mentioned, in certain cases the therapist had to walk the patient out of the room. However, our system has been implemented and tested for detecting the entry/exit of one person at a time, since the target user of the system is an autonomous subject who lives alone.

The tests in the Living Lab also achieved another goal, i.e., to demonstrate that the environmental sensors of the system were transparent to the patients and did not interfere with his/her activities. This suggests that the system can be installed in the homes of the subjects to be monitored without making them uncomfortable in performing their daily routines.

5. Case study on Type 1 Diabetes

Diabetes is an autoimmune pathology that may either prevent the patient's pancreas from producing insulin and/or prevent the body cells from responding to insulin properly. In both cases, it leads to an alteration in the patient's blood glucose (BG) level, which needs to be controlled through a combination of diet, physical activity, and medication. Medication may include either oral hypoglycemic drugs or insulin injections, depending on the type of diabetes. In particular, diabetes may occur in three types. In patients affected by Type 1 diabetes, the β cells in the pancreas that are responsible for the production of insulin are compromised. Thus, the pancreas is irreversibly prevented from producing insulin, and the subject needs to undergo insulin injections daily. In subjects affected by Type 2 diabetes, the pancreas can produce insulin, but the body cells are prevented from using it, partially or completely. A third type of diabetes may occur in women during pregnancy and is often a temporary condition. In this work, we will focus on supporting the management of Type 1 diabetes.

Achieving good glycemic control in patients affected by Type 1 diabetes often represents a challenge, due to the presence of several variables that might influence glycemic values, to the possible non-compliance of adult and pediatric patients, and to the daily self-control of the disease [197–199]. To obtain an adequate glycemic control, the patient is required to take several BG measurements during the day, and to undergo periodic encounters with the physician [200]. A close interaction between the patient and the physician takes place also beyond the scheduled periodic visits, and it often requires frequent telephone or email contacts to manage any therapeutic adjustment.

As anticipated in Section 2.3, a significant effort has recently focused on developing applications for supporting diabetic patients in managing their disease. Two main reasons have driven the focus of information technology on diabetes. First, the high prevalence of this disease highlights the need for IT solutions that may facilitate the management of a wide patient population. According to the report presented by the World Health Organization (WHO), in 2016 diabetes affected 8.5% of the global population [201]. In Europe, in the same year the diabetic populations consisted of 52 million of subjects, with the prevalence reaching 10% of the population in the most affected countries. Furthermore, the expected prevalence for the next years is even higher, due to the population aging and to the diffusion of specific risk factors, including sedentary lifestyle and obesity. In addition to the emerging need for IT solutions, the focus on diabetes has intensified thanks to the diffusion of new IT technologies, that are now available on the market at affordable costs. For example, as anticipated in Section 2.2.2, recent years have seen the introduction of novel wearable devices for continuous BG monitoring. However, despite their widespread adoption, such devices still have some limitations, such as the possibility to store only the most recent history of the subject's BG measurements, the generation of static reports that include only partially useful information, the lack of instruments to analyze more than one patient at the same time, and, most important, the impossibility of sharing the raw data with the physician, who can access only the predefined reports.

The diffusion of wearable monitoring devices has also led focus to integrating the patient's BG profile with lifestyle information. As anticipated in Section 2.2.2, recently the use of wearable sensors for monitoring sleep and physical activity has intensified, both in healthy subjects and in patient populations, providing a significant amount of data to be analyzed. Thus, smart applications to facilitate data integration and analysis are needed. In this chapter, we present the Advanced Intelligent Distant-Glucose Monitoring (AID-GM) system [202, 203], a tool designed to improve disease management in diabetic patients allowing data sharing and an advanced analysis of glycemic trends.

5.1. AID-GM: an advanced system supporting continuous monitoring of T1DM patients

AID-GM [170, 202] is a web-application developed at the Biomedical Informatics Lab of the University of Pavia, Italy, in collaboration with the Pediatric Endocrinology and Diabetes Outpatient Service of the Policlinico San Matteo Foundation hospital in Pavia, Italy. It provides an advanced platform for the analysis and summarization of continuous BG monitoring data, complemented with the information on the subject's sleep and physical activity collected by an activity tracker. AID-GM offers a wide range of visualization and analysis tools, available for different users, i.e., the patient and his/her diabetologists. In the following paragraph we will introduce the technical details of the platform.

5.1.1. The architecture

AID-GM is mainly developed in Java, integrated with JavaServer Faces (JSF), JavaScript (JS), Hibernate, and MySQL technologies. The system architecture is compliant with the framework described in this thesis and is shown in Figure 27.

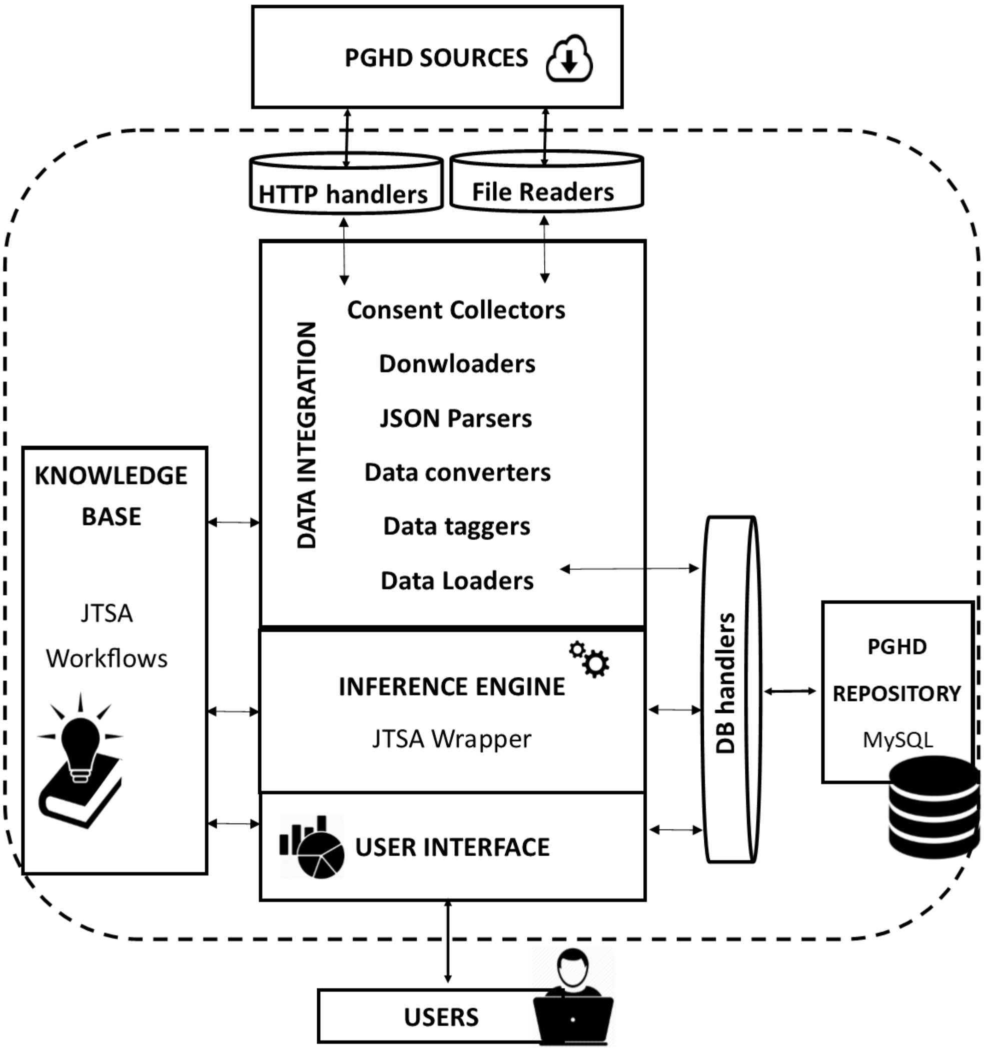


Figure 27: AID-GM architecture

Data integration and PGHD repository

AID-GM combines data from three PGHD sources, namely a BG monitoring device, an activity tracker, and the patient's self-reported information. AID-GM is designed to be independent of the specific BG monitoring device. For our use-case we focus on the Abbott Freestyle Libre FGM system, described in Section2.2.2 [98]. This choice is mostly due to the fact that this system has been approved for use in pediatric patients, who represent the target of our study. To collect information on the subject's HR value, on the amount of daily activity, and on sleep quality and quantity, AID-GM uses a Fitbit activity tracker. Finally, information on the subject's habits, including the meal schedule on each day of the week, is asked to the patient directly through the AID-GM system interface. Data acquisition requires the patient's collaboration, with a level of commitment that depends on the type of data.

Information on the patient's habits is collected at registration, when the patient is asked to fill-in a form to provide, for each day of the week, his/her usual time schedule concerning primary meals, snacks, and sleep. This information can be modified in case the patient thinks it is necessary.

As anticipated in Section 2.2.2, the BG data collected by the Freestyle Libre system are not stored in a proprietary cloud from which it can be downloaded automatically. Thus, BG data should be periodically uploaded into AID-GM by the patient, using a dedicated form that takes as input the text file produced by the Abbott software. In the application, we implemented a File Reader to parse such textual file and extract the contained information. Each row in this file describes one event, that can be either a BG measurement automatically provided by the sensor, or information reported by the patient through the reader. In particular, the reader allows the patient to report a specific set of diabetes-related events, which include the insulin intake, possible health-related issues (e.g., occurrence of flu), or meals. Each event is defined by its time of occurrence and by other attributes that depend on the event type. For BG events, the measured BG value is specified, while insulin events are characterized by the bolus dosage, as inserted by the patient. Health-related events include a textual description of the reported issue.

When the patient registers to AID-GM he/she may also connect his/her AID-GM account with his/her Fitbit account, by explicitly authorizing our application to download his/her records from the Fitbit cloud. Once the patient has given consent to access his/her records, the data coming from Fitbit are automatically retrieved from the cloud every night. In particular, for each day we download a daily summary of the measured activity, reporting the total amount of time in which the patient has been moving. For each workout we store start time, intensity, and duration. For each sleep record, we memorize the time at which the subject falls asleep, the time he/she wakes-up, the amount of time in which the subject was awaken in bed, and the amount of time in which the subject's sleep was restless. The subject's HR profile, that includes one HR measurement per minute, is also downloaded and stored in a dedicated DB.

In AID-GM, the data taggers have a relevant role for contextualizing the BG measurements based on the data filled-in by the patient and on the data collected by the Fitbit tracker. Taking into account the patient's habits collected at the patient's registration, data taggers tag each event considering the time of day when that event takes place. For example, an event is assigned the *awakening* tag if it occurs between the subject's usual awakening time and his/her usual breakfast time. Other possible tag values are after *breakfast*, *before lunch*, *after lunch*, *before dinner*, *after dinner, night*. The *night* tag is assigned when the time of occurrence is between the subject's usual bed-time and his/her usual awakening time. The events and their tags are stored by the data loaders in a dedicated MySQL database.

Given the information collected by the Fitbit, the data taggers calculate an additional tag for each BG event, to contextualize it within the actual patient's activities, as registered by the activity tracker. We will refer to this tag as the *Fitbit tag*, to distinguish it from the tag calculated considering the habits declared by the patient, which we will refer to as *Schedule tag*. The possible values for the Fitbit tag are *sleep*, *workout*, *routine*, and *not used*. The *sleep* and *workout* values are assigned when the BG event occurs during a tracked sleep session or during a tracked workout, respectively. The *routine* value is assigned when the event occurs in an instant in which the patient is not sleeping and is not training. The *not used* value is assigned to each event occurring when the patient is not wearing the Fitbit tracker. In particular, we assume that the subject was not wearing the tracker at a specific time (t_i) if no HR measurements are available in the interval [t_i - 5 minutes ; t_i + 5 minutes]. These tags are used for focusing the analysis on a specific time frame. For example, if the user is interested in detecting BG alterations occurred during the patient's sleep, only data having the *sleep* tag will be retrieved from the database by the DB handlers.

Knowledge base and inference engine

AID-GM allows the users to detect in the data a set of qualitative patterns that clinicians would manually search for, since they represent behaviors of interest to evaluate the patient's health status. The set of patterns was discussed and developed in collaboration with our clinical partners and will be described in Section 5.1.2. For example, the set includes hyperglycemia, hypoglycemia, and the dawn effect described in Section 2.4.4. For each pattern, in the KB we have formalized one JTSA workflow, whose parameters were tuned in collaboration with the clinicians. The description of the formalized KTSA workflows is provided in Appendix D.

The sequence of steps for running the domain-specific JTSA workflows is not as complex as in the NONCADO system (Chapter 4). Thus, in the KB we formalized simple rules for activating the workflows, and we did not need to integrate Drools to run them.

User interface

AID-GM provides tools for visualizing and analyzing the collected data, and for facilitating the communication between the patient and the diabetologists. In particular, Table 22 lists all the available functionalities, grouped by type of action, specifying to which user/s they are addressed.

The system is designed to analyze both individual patients and patients' groups, although the functionalities for analyzing data from multiple patients are available only to the clinician. This might help the clinician to monitor the patient population, by facilitating the analysis of the huge amount of longitudinal data made available by the wearable monitoring devices, which would not allow an easy manual identification of critical situations.

Table 22: AID-GM functionalities and corresponding users, namely patient (P) or/and clinician (C)

ACTION	FUNCTIONALITY	USER	
		P	C
Set up of the AID-GM account and access	Access through secure authentication	•	•
	Request to be registered in a clinical center	•	
	View and approval of registration request		•
	Set-up and update of daily habits (i.e., time of meals, wake-up and bedtime for each day of the week)	•	
	Set-up and update of patient-specific thresholds to identify glycemic alterations (i.e., hypoglycemia and hyperglycemia) and HR alteration (i.e., tachycardia and bradycardia).		•
Data upload	Upload of BG monitoring data	•	•
	Consent to download the Fitbit data	•	
Data analysis and visualization	Visualization of BG overall time series, daily trends, and AGP of one patient	•	•
	Visualization of a summary of the most recent hyperglycemic and hypoglycemic episodes	•	•
	Visualization of combined BG and HR daily profiles, complemented with information on sleep, workout, meal, and insulin intake	•	•
	Visualization of a summary of the physical activity in a selected period	•	•
	Visualization of a timeline that shows if the patient is regular in terms of sleep and activity	•	•
	Detection and visualization of patterns (Section 5.1.2) for one patient	•	•
	Detection and visualization of patterns (Section 1) for a group of patients		•
	Drill-down to the BG and HR profiles related to the time intervals in which a selected pattern occurred	•	•
	Visualization of statistics related to pattern detection for a group of patients		•
	Visualization of the patients' list, and list of the recently uploaded data		•
	Visualization of patient's information (demographics, contact information, onset date, weight, thresholds for BG and HR)		•
Communication between the patient and the physician	Request for a consultation	•	
	Visualization of consultation requests		•

Given the amount of interaction between AID-GM and its users, a lot of focus was also given to the graphic interface of the application. In particular, AID-GM is integrated with a set of technologies meant to improve the user experience in using web applications. For example, AID-GM is integrated with Primefaces [204], an open-source library for JSF-based applications that provides a set of customizable components for web-applications, including for example HTML editors, tables, calendars, and search bars with auto-completion mechanisms. As regards the data visualization, a JS library called AmCharts [205] was used to render the graphs that show the patterns detected in the patients' data. In particular, AmCharts allows to quickly implement a rich set of interactive graphs, including line charts, pie charts, and bar charts.

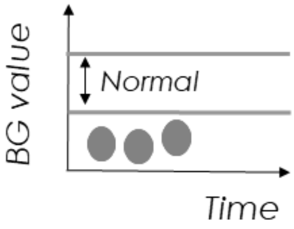
A prototype of the AID-GM application was already available when I started working on the AID-GM project. In particular, the functionalities for searching for patterns in the patients' BG data were already implemented. During my PhD, I contributed to the integration of the BG data with the information on lifestyle collected by the activity tracker and to the deploy of the web-application, first to test sites and then to the Policlinico San Matteo hospital.

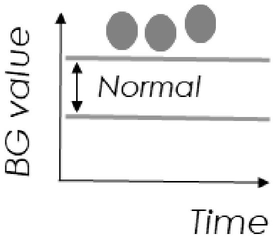
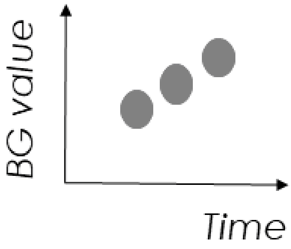
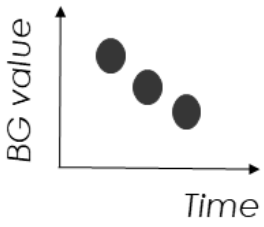
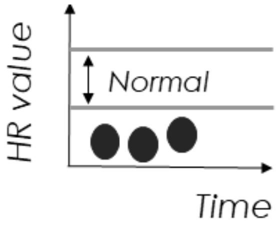
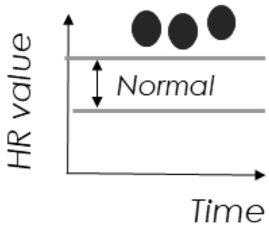
5.1.2. Domain-specific pattern detection

In agreement with the diabetologists we have defined a set of 10 knowledge-based patterns that are relevant for evaluating the diabetes outcome. In particular, these represent well-known clinical phenomena, which could identify potentially risky situations. Thanks to the Fitbit tag it is possible to contextualize the search for patterns within the patient's day (e.g. during sleep or workout).

Out of the 10 formalized patterns, 6 are basic and consist of a specific trend in a single variable, that may be either the subject's BG profile or his/her HR profile. Table 23 lists the basic patterns currently available in the AID-GM system. For each pattern, the table reports the input data and a graphical representation of the behavior of interest. For all the patterns that require the comparison of the measured value with a threshold (e.g., Hyperglycemia), the used threshold is patient-specific. The threshold values for each patient are personalized by the clinician through the system.

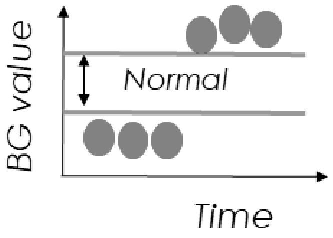
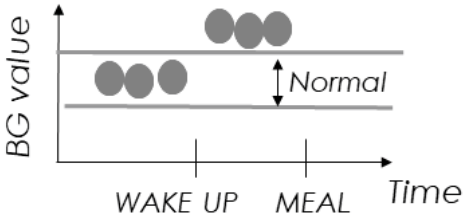
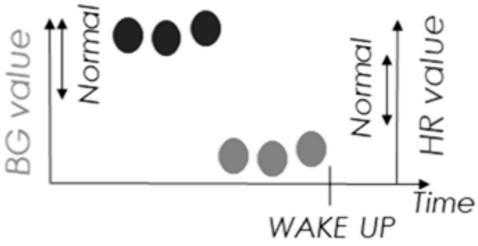
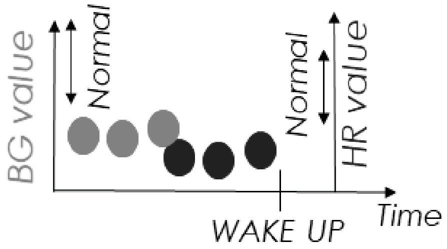
Table 23: Basic patterns implemented in AID-GM

PATTERN	INPUT DATA		GRAPHICAL REPRESENTATION
	BG	HR	
Hypoglycemia	•		

Hyperglycemia	•		
Increasing **BG value**	•		
Decreasing **BG value**	•		
Bradycardia		•	
Tachycardia		•	

We have also implemented Complex patterns, which consist of a combination of patterns, potentially extracted from different PGHD time series. Among these complex patterns, we have included the dawn effect (Section 2.4.4), and the rebound effect, which consists in a hypoglycemic episode followed by a rapid increase in the subject's BG leading to hyperglycemia. We also defined multivariate patterns that combine HR and BG trends. Table 24 lists the complex and/or multivariate patterns currently available in the AID-GM system.

Table 24: Complex and/or multivariate implemented in AID-GM

PATTERN	INPUT DATA			GRAPHICAL REPRESENTATION
	BG	HR	SLEEP	
Rebound Effect (hypoglycemia followed by hyperglycemia)	•			
Dawn Effect (normal BG value at night followed by hyperglycemia at wake up)	•		•	
Tachycardia PRECEDES hypoglycemia (DURING sleep)	•	•	(•)	
Hypoglycemia PRECEDES bradycardia DURING sleep	•	•	•	

In Appendix D we provide a description of the 10 JTSA workflows implemented in AID-GM, complemented with the selected parameter values.

5.1.3. Decision support

The AID-GM graphical interface guides the users through the process of selecting the patients for the analysis, selecting the patterns of interest, the time frame of interest, and visualizing the obtained results, i.e., the time intervals in which the selected patterns occurred.

For the individual patient, the possibility of detecting patterns supports the diabetologists in understanding the subject's BG temporal profile and in targeting the therapy accordingly. In fact, the visualization of the temporal patterns described in Section 5.1.2 (Figure 28) gives a synthetic overview of the patient's metabolic control over a selected time frame.

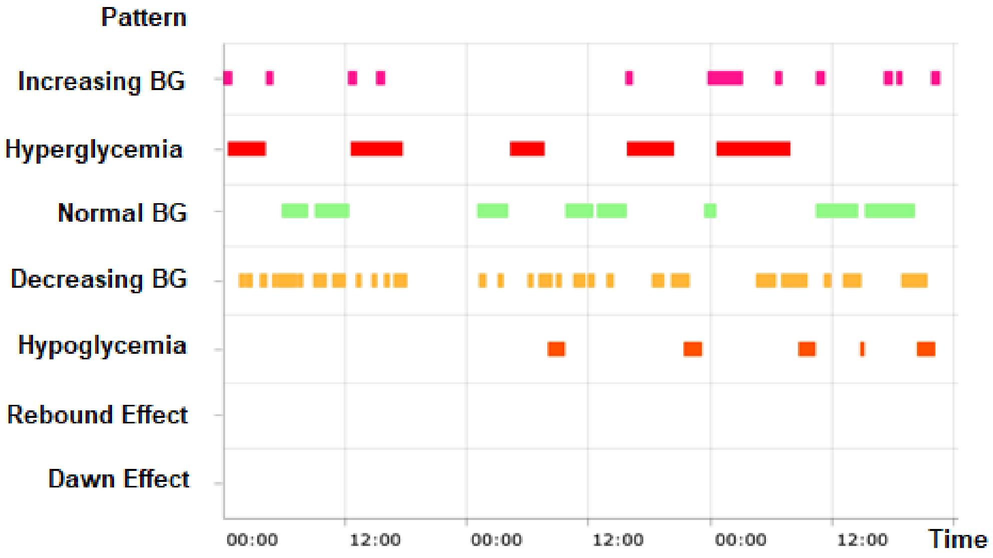


Figure 28: Visualization of the temporal patterns in AID-GM

In Figure 28 each colored bar represents one occurrence of a specific pattern, specified by the labels on the left of the graph. By clicking on a selected bar, the start time and end time of that occurrence of the pattern are provided. For each pattern occurrence, the corresponding colored bar also links to the chart for the integrated visualization of the subject's BG and HR profiles related to the time interval of that pattern occurrence.

The temporal analysis can be focused on specific time frames, such as selected days of the week, or moments of the day. For example, the user can search for night-time hypoglycemia episodes that occurred during weekends.

In addition to the functionalities for pattern detection, the AID-GM system provides the users with several reports that integrate both the subject's BG profile and the data collected by the activity tracker, summarizing both the BG levels and the patient's lifestyle. These data views are meant to support patients and physicians in taking decisions about timely changes of lifestyle, diet, or therapy and, more in general, for a more informed disease management. In particular, the system provides four main kinds of visualizations. First, AID-GM provides combined the subject's BG and HR daily profiles, supplemented with additional events like sleep start, sleep end, workout (with duration), insulin dosage, and meals (Figure 29). The upper section of Figure *29* shows an example of such visualization, while the lower section shows a detail of the graph, to highlight the icons that convey the information on the subject's lifestyle. The legend of the available icons is shown in Figure 30.

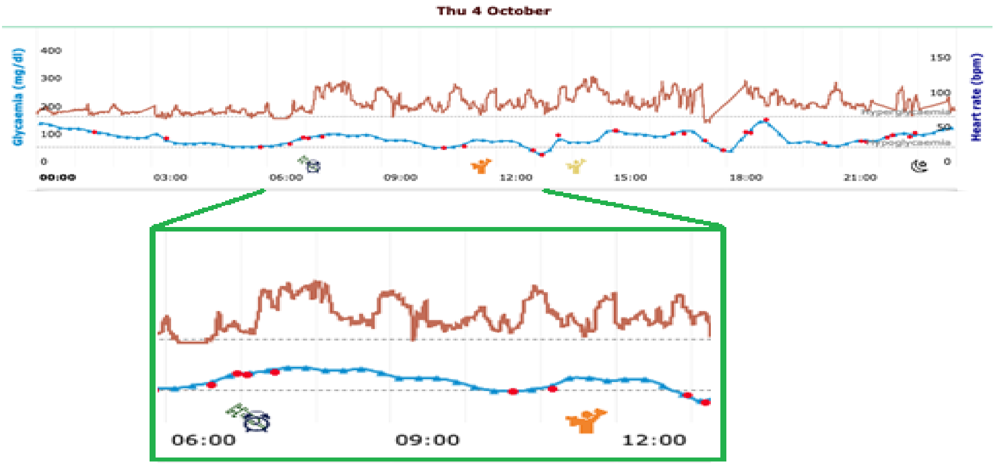


Figure 29: BG and HR daily profiles, supplemented with information on the subject's lifestyle

Figure 30: Legend of the icons used in the visualization of the BG and HR daily profiles

A second visualization provides a summary of the patient's lifestyle, by giving an overview of the activities of the patient over a selected period. The time frame of the summary can cover one day, one week, one month, or a user-defined period. For each day in the selected time frame, this visualization provides the patient's timeline (Figure 31), where the different activities registered by the tracker are represented by colored bars.

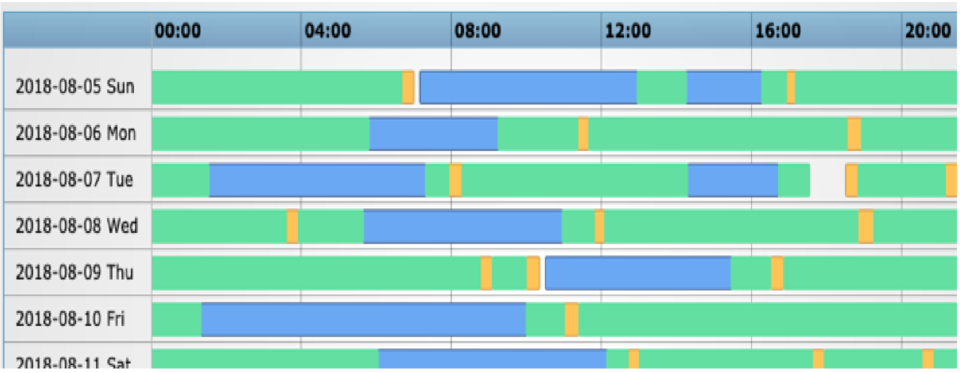


Figure 31: Visualization that provides a summary of the patient's lifestyle

Each type of activity has a different color (i.e., light blue for sleep, yellow for workouts, and green for routine), while the grey color indicates the time intervals in which the patient was not wearing the Fitbit tracker. The visualization of the timelines of different days can easily point out if the patient's lifestyle was regular in terms of sleep and physical activity. This visualization is described in this thesis to provide a complete description of the AID-GM system, but it was not object of my work.

A third visualization shows the AGP related to a selected period. The AGP is an internationally recognized visual representation that combines BG data from multiple days and collates them into a single 24-hour period, to provide an <<*average*>> BG profile over the selected time frame. As for the timelines that summarize the patient's activity, I have not contributed to developing the functionality for visualizing the AGP, but it is mentioned in this chapter for the sake of completeness.

In addition to the features dedicated to the individual patient, AID-GM allows the clinicians to analyze groups of patients. In particular, it is possible to perform the pattern detection on a selected group of patients (Figure 32).

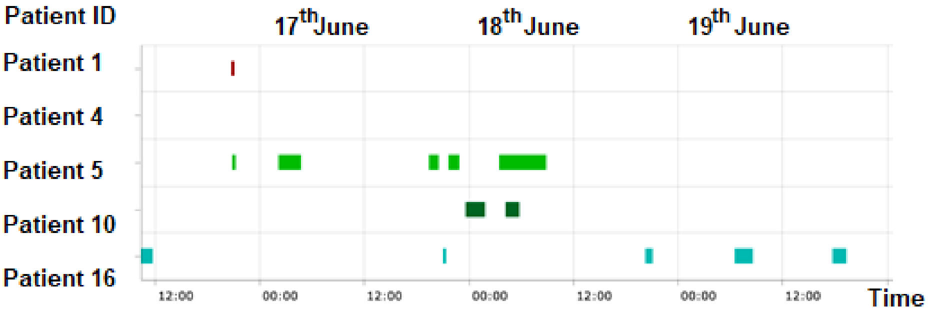


Figure 32: Example of visualization of patterns when analyzing a group of patients

In this case, the user can select one pattern at a time. The occurrences of the selected pattern are represented in a different color for each patient. Again, each bar links to the chart for the integrated visualization of the subject's BG and HR profiles related to the time interval of that pattern occurrence.

While on the individual level analyzing the collected PGHD supports the diabetologists in targeting care to the specific subject, analyzing groups of patients allows the detection of frequent patterns, and the identification of sub-groups in the patient population. The functionality for detecting patterns in data coming from multiple patients may also allow the clinician to quickly identify the individuals who need closer attention.

5.2. Pilot study and results

The AID-GM system was tested in two phases. In a preliminary test phase, the application was deployed locally on a machine running Windows that belonged to our clinical partners of the Policlinico San Matteo Hospital and the application was accessed only by the clinicians. In this test phase, the clinicians used AID-GM to detect patterns in BG data collected from a group of volunteers for a technical/functional assessment of the application, and an evaluation of the temporal data analysis features.

In a second test phase, we deployed the application on a server run by the Policlinico San Matteo Hospital, and we evaluated the usability of our application in a real-world pilot study involving 30 diabetic patients and 3 members of the health care personnel, namely a diabetologist, a resident, and a nurse.

The following two sub-sections describe the two test phases performed.

5.2.1. Preliminary tests on volunteers

The design and development of AID-GM followed an iterative process based on a close interaction with the clinicians through periodic meetings. During the first iterations, we tested the system as developers, on data collected by 4 healthy volunteers, who wore a Freestyle Libre sensor for 2 weeks in 2016. After the development tests, we deployed the application on the clinicians' laptop, and the testing was performed directly by the pediatricians, who evaluated the functionalities of the system during specific sessions carried out outside the routine clinical practice. This study allowed identifying a number of technical issues, missing functionalities, and possible improvements in the existing features of the system. In particular, the pediatricians asked us to implement a set of functionalities that were not available in the first version of AID-GM, namely:

- Allow the diabetologist to specify the diabetes onset date in the patient's profile.
- Integrate the patient's profile with his/her weight, whose value must be updated by the patient every 3 months.
- Allow the diabetologist to enter/modify/ and visualize the insulin therapy prescribed to the patient.
- Allow to drill down from the pattern visualization (Figure 28) to the corresponding BG profile.
- Allow the patient to request for a consultation, with the possibility of indicating which BG data triggered the need for consultation.

By testing the system, we also realized that the Freestyle Libre system does not adhere to the daylight saving time practice, as anticipated in Section 2.4.1. Thus, we implemented a functionality to correct the dates of the BG measurements accordingly.

The diabetologists also asked us to improve two existing features, namely the patients search settings in the dashboard that show the patients' profiles and in the menus for selecting the patient or groups of patients for pattern detection. They also asked us to improve the options for filtering the obtained results according to selected timeframes.

The AID-GM prototype developed during this phase was then evaluated by two diabetologists on the data voluntarily provided for the study by 24 pediatric patients, 12 boys and 12 girls. These patients were aged 3 to 22 years, showed an average disease duration of 4.2 years (range: [4 days – 15.2 years]), and provided data for an average of 172.2 days. The evaluation aimed at detecting the most clinically relevant BG patterns and tuning the parameters needed for their extraction in the JTSA workflows described in Section 5.1.2. The physicians used the group analysis functionality to examine the patients' sample. Patterns were searched on the complete monitoring period for all the patients. The night-time hypoglycemia pattern was found in 20 subjects. The average number of hypoglycemic events per patient was 29. Although most subjects had a limited number of events (11 patients with less than 20 patterns), it was possible to detect recurrent episodes in 9 patients. Among these, 2 subjects showed this pattern very frequently, with 99 and 162 events in a period of 5 and 7 months, respectively. Rebound patterns were detected in 8 subjects, with an average number of 2 events per patient. The dawn effect is a clinically relevant pattern not easy to be spotted using only paper-based glycemic diaries and, even more important, it must be differentiated from hyperglycemia at wakeup that may appear in response to night-time hypoglycemia. The dawn effect was detected in 6 out of 24 patients, with an average number of 1.84 patterns per patient.

5.2.2. Pilot study on the real-world patient population

The usability of the system has been evaluated in a pilot study carried out in collaboration with our clinical partners of the Policlinico San Matteo hospital in Pavia, Italy. The study was approved by the Institutional Review Board (IRB) of the hospital and involved 30 pediatric patients.

The protocol

We enrolled 30 subjects receiving care at the Policlinico San Matteo hospital and already using the Freestyle Libre device for monitoring their BG. Through AID-GM, the subjects themselves or their parents, in case of patients aged under 18 years, provided their consent to download their Fitbit data and use them in the application. The patients were asked to use the system to periodically upload their BG data and, if they wanted to, for visualizing information related to their BG profile and Fitbit data. After 2 and 6 months from enrollment, patients were asked to fill-in the System Usability Scale [175] questionnaire. For patients aged under 18 years, the

SUS questionnaires were filled in by the parents, whereas patients older than 18 years old filled in the questionnaires themselves. In terms of care path, the patients were treated following the usual clinical practice.

Besides the patients, also three pediatricians were asked to fill-in the SUS questionnaire at the end of the study.

Results and discussion

Out of the 30 patients originally enrolled in the study, 3 were discarded as they never login nor uploaded data to the system, leaving us with a sample of 27 subjects, including 14 females and 13 males. The mean age was 12 years (IQR= [8.5; 13.5]). The median duration of the BG monitoring was 97 days (IQR= [65; 167]). Not all the patients used the activity tracker. In particular, 9 out of 27 patients never used the Fitbit tracker. Considering the patients who wore it, the median duration of the Fitbit monitoring was 47 days (IQR= [38; 56]).

To further characterize the patients' population, we used AID-GM to perform an analysis of the BG profiles through pattern detection. Table 25 gives a snapshot of the number of BG patterns detected in the dataset. The patterns are computed on all the patients taking part in the study. Since some of the patients didn't use the Fitbit tracker, the table doesn't show patterns like the dawn effect and the rebound effect, which require the information on sleep to be accurately detected. The table shows that there are some patterns that happen more frequently than others in our population, and that different patterns have different durations. For example, Increasing and Decreasing BG trends are the most frequent out of the considered patterns, but the episodes that last longer are the hyperglycemic ones.

Table 25: Number and duration of BG patterns detected in the dataset

PATTERN	NUMBER. OF EPISODES	AVERAGE EPISODE DURATION (MINUTES)
BG Decreasing	10570	75 (IQR = [45 ; 105])
BG Increasing	10892	60 (IQR =[45 ; 91])
Hyperglycemia	9673	165 (IQR = [60;404])
Hypoglycemia	2555	30 (IQR = [15;75])

For the 17 patients who have been wearing both the FGM and the Fitbit devices we also extracted additional patterns, involving BG, HR, and tracked sleep. The dawn effect and the rebound effect (see Section 5.1.2) occurred rarely in our population. We detected 10 episodes of dawn effect in 7 patients and only one episode for the rebound effect.

To evaluate system use, we analyzed the log files of AID-GM. These files include information about the type of action performed by the user, together with the date and time of execution. The available types of actions are *login*, *logout*, *visualize*, *modify*, and *find pattern*.

The most frequent action performed by both users was the *visualization* action. This action refers to the several types of visualization available in AID-GM (see Section 5.1.3), which include the visualization of BG profiles and summaries, the visualization of Fitbit data and summaries, and the visualization of patients' demographic and clinical information. In particular, patients were mostly interested into checking their daily BG trends. As regards the pediatricians' actions, we registered several visualizations of the patients' information, and visualizations of daily BG and HR trends. They also examined Fitbit summary information, which were instead less popular among patients, who probably checked these data directly through the Fitbit mobile app.

The *modify* action is registered when the user updates the data stored in the patient's account. In this pilot study it has mainly regarded the personalization of the patient's thresholds for detecting hypoglycemia and hyperglycemia, that can be performed by the physicians. In particular, we registered 19 accesses for setting the threshold for hyperglycemia, and 22 for hypoglycemia. As regards the patients' actions, the registered *modify* actions were related to the personalization of the daily routine habits. In this case, we registered about one access for each patient.

The action related to the analysis of temporal patterns registered 73 accesses for patients and 45 for physicians, who focused 30 times on the functionality for the single patient and 15 times on the extracting patterns on a group of patients. The patterns that were most frequently searched for were hypoglycemia, with a total of 75 views (23 by physicians and 52 by patients), and hyperglycemia, with a total of 54 views (19 by physicians and 35 by patients). Interestingly, both patients and physicians used several times the functionality that allows searching for multiple patterns to extract all the available patterns at once.

Besides the overall number of actions, we also considered the weekly trend, computed by total number of actions for each week of enrollment. This analysis showed that the number of actions in the first week was higher than the number of actions in the following weeks, both for patients and for physicians. In particular, the average number of actions performed by the patients in the first week of usage was 22, whereas in the following weeks we recorded an average of 1.23, with a standard deviation of 0.7. For physicians, the average number of actions in the first week was 89, with a decrease to 10 in the following weeks, with a standard deviation of 9.8. Out of the 24 weeks of monitoring, 10 patients accessed the system for 10 weeks or more (even not consecutive), 6 patients accessed the system between 5 and 9 weeks, whereas 11 patients accessed the system for less than 5 weeks. According to the logs, the pediatricians used the system more during weekdays and in the morning, whereas patients have a more uniform distribution of usage throughout the week and during the day, with an almost equal number of actions in the morning, afternoon, and in the evening. The duration of sessions was computed as the time interval between a login and a logout when the latter was available, and as the time interval between a login and the last action before the next login when logout was not available. This second case occurs when the user doesn't perform any action for 30 consecutive minutes, leading to an automatic logout. The average session duration for patients was 7.3 minutes, whereas for physicians it was 9.1 minutes. Since the first access to the system is performed by the patient together with the physician, who trains the new user by illustrating the functionalities of the system, we finally evaluate the duration of the first session for each patient to have an estimate of the training time. During our pilot, the average duration of the training session was 24.3 minutes, with a standard deviation of 13.6 minutes.

During the pilot study we collected some user-reported issues. The most frequent were two. First, the inability to upload the BG measurements when the patient sets the time in the Freestyle reader incorrectly, causing the collected BG measurements to have unreliable temporal tags. In particular, in AID-GM unreliable temporal tags are identified by analyzing the history of the time changes set by the patient in the reader. The history is collected from the file produce by the Abbott software and stored in the DB for future reference. For each recorded time change, the file lists the pre-change time and the post-change time. If they are more than one hour apart (with a certain tolerance), i.e., if the change does not correspond to a daylight saving time change (see Section 2.4.1), then AID-GM stops the automatic loading of the file and automatically sends us an e-mail indicating the problematic file. In that case, we correct the temporal tags of the BG measurements before uploading them to the AID-GM system. During the pilot study this issue occurred 8 times. Second, the inability to log-in to the personal account for entering the wrong password more than 5 times in row, since in that case we block the user's account, considering it an attempt to unauthorized access.

In addition to analyzing the user-reported issues, the system usability was assessed using the SUS questionnaire, which was delivered both to physicians and patients. For the patients, the average SUS score at 2 months was 82.6, whereas at 6 months we registered a slight decrease in the average score, which was 76.4. Both scores are considered above average, being above 68, i.e., the threshold conventionally used to evaluate the goodness of the SUS result. Although the score remained above average, we investigated the obtained results to better explain the reasons for its decrease. First of all, from the analysis of the system logs, we observed that 8 patients never used the system after two months, despite having completed the SUS questionnaire at the end of the study. Considering only the patients who performed at least one access after two months of usage, the average SUS score at 6 months is 81.3. For 11 patients belonging to this group, the SUS score increased or remained stable, whereas for 8 patients we registered a decrease in the score. Analyzing the single questions, the one that we found most critical was the following: <<*I found the various functions in this system were well integrated*>>. To this question, 7 patients gave a lower score after 6 months than at 2 months. Perhaps this question was not entirely understood by the patients, because of its technical formulation. Three physicians completed the questionnaire for our study. In this case, we had the maximum SUS scores (i.e. 100) for all three. As underlined by the results of the SUS questionnaires, in which both clinicians and patients provided positive assessments, the AID-GM system was considered user-friendly during the real-world pilot study.

The result conveyed by the SUS scores is particularly positive, considering that AID-GM has introduced a significant change in the care workflow, for both roles. Previously, the patients used to download the BG summaries from the Freestyle system less frequently, usually before the check-up visit, to discuss their BG profile with the diabetologist. The diabetologist could visualize the graphs produced by the Freestyle Libre System, or read the raw data, i.e. the time series of the collected BG measures, without having tools to analyze them. During the pilot study, the possibility to remotely share raw BG data with the clinician prompted the patients to download their BG data more frequently and to upload it into the AID-GM system on a regular basis. For example, one patient systematically uploaded data to AID-GM once a week. Thus, the high value of the patients' SUS points out that they believe it is worth investing time in uploading data to the system to facilitate remote monitoring of their health condition, and possibly receive better care. The advantage deriving from the use of the AID-GM system is even more evident when considering the health care personnel, who can gain deeper insights on how the patient’s condition evolves between check-ups, contextualizing the observed glycemic profile with the information on the patient's lifestyle as collected by the Fitbit activity tracker.

6. Discussion and conclusion

In this thesis we presented three main contributions. First, we proposed a conceptual framework for analyzing time series of PGHD collected from wearable and/or environmental sensors. In addition, we presented how we applied our framework to design and develop two decision support systems for assisting chronic patients.

This work was motivated by the emerging use of devices for collecting PGHD around the clock. As discussed in Section 2.2.2, nowadays wearable and environmental sensors are now commonly used for monitoring the individual's health status and lifestyle continuously. This is mostly due to the advancement of technologies, which made such devices both easy to use and affordable to purchase. As discussed in Section 2.3, although collecting time series of PGHD has become easier, systems that perform advanced analyzes of such data to provide clinical decision support are still rare. This might be due to the lack of comprehensive guidelines for designing and developing such systems. Although on one hand PGHD may be crucial to understand what happens to the chronic patient between medical appointments, on the other it is known that considering this additional information can lead to a longer duration of the patient's check-up, especially when it is not clear for the health care personnel how to interpret these data [206]. This consideration applies to all types of PGHD, and especially to measurements that are collected automatically at a high frequency, such as the BG measurements from CGM or FGM devices. In addition, as discussed in Chapter 2, PGHD might show data quality issues, which must be evaluated and managed carefully before analyzing the data. Since interpreting time series of measurements can be challenging and/or time consuming, providing systems able to summarize and guide the interpretation of time series of PGHD is crucial to fully exploit the potential of such data.

In the conceptual framework described in this thesis we identified and summarized the key steps needed to build applications for analyzing time series of PGHD. In our framework we collected considerations that do not depend on the clinical domain for which the system is developed. The described architecture design and a set of actual software components may be used for any application domain.

With the suggested architecture design, some software components may require a significant effort to be implemented. Some existing technological solutions could help to standardize the technologies used by the CDSSs for analyzing PGHD and to facilitate their implementation. For example, recently some platforms for collecting time series of measurements from a set of monitoring devices have been proposed. One of the most known is

Remote Assessment of Disease And Relapses (RADAR), an open source platform developed within a project funded by the Innovative Medicines Initiative [207]. The set of sensors supported by RADAR includes the Fitbit tracker, and other wearables which collect the subject's blood volume pulse, electro dermal activity, ECG, raw acceleration, heart rate variability, respiration rate, and oxygen saturation [208]. Integrating the RADAR platform in a CDSS might facilitate the development of the components meant for data collection. However, RADAR does not collect those parameters that are usually monitored in sick individuals only, such as BG monitoring data. Thus, other solutions should be explored to integrate this data. Recently, to enable real-time BG data collection from FCGM devices, wearable devices for continuously reading FGM measurements have been proposed [209, 210]. Such devices must be positioned on the patient's arm, next to the glucose sensor, and can scan it automatically, without requiring the patient to use the BG reader (Section 2.2.2). Among these, BluCon [209], produced by Ambrosia Systems, an American medical technology company, is a small wearable that can connect to the Freestyle Libre sensor using Near Field Communication (NFC) and exploits Bluetooth Low Energy for transmitting the BG measurements to a dedicated mobile application, that visualizes them. Given the growing interest in analyzing time series of PGHD, in the near future the same companies that produce FGM sensors will likely integrate devices similar to BluCon into their monitoring systems. However, as long as FGM systems rely on commercial protocols without providing interfaces for downloading data automatically, general purpose platforms such as RADAR will be prevented from offering ready-to-use functionalities for downloading time series of BG measurements. The same consideration holds for non-commercial sensors, such as the environmental sensors used in the NONCADO system. Thus, besides platforms similar to RADAR, the CDSS should include interfaces for receiving time series of measurements from non-commercial sensors or any commercial sensor that is not supported by existent platforms for collecting PGHD. In addition, the graphical interfaces of JTSA and Drools may facilitate the formalization of the knowledge base of a new CDSS, allowing the user to define domain-specific workflows (in JTSA) or rules (in Drools) through guided procedures, without needing to learn the specific languages used by these tools to represent the manipulated data.

We applied our framework in the design and development of two decision support systems, namely AID-GM and NONCADO. These two CDSS represent two main contributions of this work, since they have specific characteristics that make them innovative compared to the systems having similar functionalities. In particular, to our knowledge, AID-GM is the first application that exploits the Fitbit data on the subject's sleep and physical activity to contextualize his/her BG profile and perform advanced temporal analyses. In fact, most systems for analyzing FGM or CGM data do not integrate information collected from other sensors, but only provide statistics on the subject's glycemic profile, such as the percentage of time in which the patient's BG value remained within the target glycemic range. Another innovative feature is the possibility to analyze both the individual patient and patient groups. On the individual level AID-GM supports the diabetologists in better understanding the subject’s BG profile and in targeting care accordingly. The functionalities for analyzing groups of patients allow the clinicians to detect which clinical patterns occur frequently in the considered patient population, and to possibly stratify the population in sub-groups, identifying the individuals who need closer attention.

As regards the NONCADO system, the most innovative feature is the exploitation of temporal abstraction for early detection of changes in the daily habits of the elderly living alone at home. In fact, significant changes in the subject’s daily habits may correspond to a decline in his/her health condition and, consequently, to an increase in his/her risk of falling and/or a decrease of his/her ability to live without assistance. Giving the elderly the possibility to live independently for as long as it is safe is fundamental, since it is known that they prefer to age within their own home [211]. When this is not possible, it seems that the elderly prefers to live in homelike care facilities, i.e., small centers that recreate a private household, rather than in traditional nursing homes, which are organized similarly to a hospital [212]. In addition to the aversion to moving, the individuals who are used to living alone prefer not to live with other people, either family members or non-relatives, even as they age [213]. Thus, systems like NONCADO may support the subject’s family that lives remotely, helping to monitor the elderly and detect potential changes that may indicate that his/her condition is declining and that he/she may no longer be able to live alone. The NONCADO system differs from the known CDSSs for preventing or managing falls. In fact, most of the systems focus on supporting the subject and his/her family after the occurrence of the fall event, possibly forwarding requests for assistance to medical personnel. Other systems detect potential changes in the subject's health condition by evaluating his/her performance in programmed exercises that he/she would not do spontaneously. Thus, the elderly may perceive the proposed programmed exercises as a constraint in his/her daily routine. As a consequence, he/she may not be compliant and may not complete the exercises on a daily basis. We believe that the elderly may more easily accept to be monitored by CDSSs that detect changes in his/her habits, without requiring him/her to perform tasks that are not already included in his/her daily routine.

As the previous descriptions suggest, AID-GM and NONCADO were designed and developed for very different clinical domains. Nevertheless, the two CDSSs share the same approach to perform data analysis and knowledge extraction. In particular, the fact that JTSA could be used in both use-cases to perform KBTA emphasizes how this tool is flexible and re-usable in a heterogenous set of contexts. We believe that other researchers may be interested in exploiting JTSA in their CDSSs for analyzing time series of PGHD. Therefore, in this work we aimed to provide an overall picture on how this can be done, by providing a useful introductory reading for any researchers interested in exploiting JTSA to provide decision support in a different clinical domain. To provide a general guideline, we focused on the methodological aspects of the proposed framework and of the two CDSSs, rather than on the implementation details, which must be evaluated separately for the specific application. Thus, when describing our two CDSSs we focused on clarifying the behaviors of interest which were searched for in the collected PGHD, the clinical motivations underlying their relevance, and how their detection contributed in providing decision support.

In conclusion, we believe that reading this thesis may help researchers to assess which challenges they will encounter in developing their own application for analyzing PGHD, and which components need to be developed to overcome them. The proposed framework is not binding: it is not mandatory to use JTSA to perform KBTA; to exploit JTSA in a CDSS, it is not mandatory to adhere to all the solutions herein presented. However, following our suggestions when designing a new application may be useful. In particular, it may allow saving time by re-using both design choices that were already tested in applications deployed in real-world clinical environments, and a set of actual software components, that are independent of the purpose of the CDSS. We believe that this possibility may enhance the development of CDSS for analyzing PGHD, allowing researchers to focus on building the unique features of their CDSSs, leveraging on the design we experimented in the AID-GM and NONCADO projects.

References

1. Rimborso | FreeStyle Libre, https://www.freestylelibre.it/libre/aiuto/rimborso.html, last accessed 2019/09/07.
2. Dehghani, M.: An assessment towards adoption and diffusion of smart wearable technologies by consumers: the cases of smart watch and fitness wristband products. Presented at the July 13 (2016). https://doi.org/10.13140/RG.2.2.31366.37449.
3. Paré, G., Leaver, C., Bourget, C.: Diffusion of the Digital Health Self-Tracking Movement in Canada: Results of a National Survey. J. Med. Internet Res. 20, e177 (2018). https://doi.org/10.2196/jmir.9388.
4. Shih, P., Han, K., Poole, E., Rosson, M.B., Carroll, J.: Use and adoption challenges of wearable activity trackers. Presented at the March 1 (2015).
5. Henriksen, A., Haugen Mikalsen, M., Woldaregay, A.Z., Muzny, M., Hartvigsen, G., Hopstock, L.A., Grimsgaard, S.: Using Fitness Trackers and Smartwatches to Measure Physical Activity in Research: Analysis of Consumer Wrist-Worn Wearables. J. Med. Internet Res. 20, e110 (2018). https://doi.org/10.2196/jmir.9157.
6. Castillo, R.S., Kelemen, A.: Considerations for a successful clinical decision support system. Comput Inform Nurs. 31, 319–326; quiz 327–328 (2013). https://doi.org/10.1097/NXN.0b013e3182997a9c.
7. Khong, P.C.B., Holroyd, E., Wang, W.: A Critical Review of the Theoretical Frameworks and the Conceptual Factors in the Adoption of Clinical Decision Support Systems. Comput Inform Nurs. 33, 555–570 (2015). https://doi.org/10.1097/CIN.0000000000000196.
8. Sim, I., Gorman, P., Greenes, R.A., Haynes, R.B., Kaplan, B., Lehmann, H., Tang, P.C.: Clinical decision support systems for the practice of evidence-based medicine. J Am Med Inform Assoc. 8, 527–534 (2001). https://doi.org/10.1136/jamia.2001.0080527.
9. Kane-Gill, S.L., Achanta, A., Kellum, J.A., Handler, S.M.: Clinical decision support for drug related events: Moving towards better prevention. World J Crit Care Med. 5, 204–211 (2016). https://doi.org/10.5492/wjccm.v5.i4.204.
10. Middleton, B., Sittig, D.F., Wright, A.: Clinical Decision Support: a 25 Year Retrospective and a 25 Year Vision. Yearb Med Inform. S103–S116 (2016). https://doi.org/10.15265/IYS-2016-s034.
11. Greenes, R.A.: Clinical Decision Support: The Road to Broad Adoption. Academic Press (2014).
12. Jenders, R.A.: Advances in Clinical Decision Support: Highlights of Practice and the Literature 2015-2016. Yearb Med Inform. 26, 125–132 (2017). https://doi.org/10.15265/IY-2017-012.
13. Khairat, S., Marc, D., Crosby, W., Al Sanousi, A.: Reasons For Physicians Not Adopting Clinical Decision Support Systems: Critical

Analysis. JMIR Med Inform. 6, (2018). https://doi.org/10.2196/medinform.8912.

14. White, C.M., Sanders Schmidler, G.D., Butler, M., Wang, Z., Robinson, K., Mitchell, M.D., Berkman, N., Henderson, J., Fiordalisi, C., Hartling, L., Newberry, S.J., Guise, J.-M., Borsky, A., Niebuhr, D., Pham, D., Gozu, A., Iyer, S., Banez, L.: Understanding Health Systems' Use of and Need for Evidence To Inform Decisionmaking. Agency for Healthcare Research and Quality (2017). https://doi.org/10.23970/AHRQEPCWHITEPAPER2.
15. Alwan, K., Awoke, T., Tilahun, B.: Knowledge and Utilization of Computers Among Health Professionals in a Developing Country: A Cross-Sectional Study. JMIR Hum Factors. 2, (2015). https://doi.org/10.2196/humanfactors.4184.
16. Polyakov, A., Palmer, E., Devitt, P.G., Coventry, B.J.: Clinicians and Computers: Friends or Foes? Teaching and Learning in Medicine. 12, 91–95 (2000). https://doi.org/10.1207/S15328015TLM1202_6.
17. Vashitz, G., Meyer, J., Parmet, Y., Peleg, R., Goldfarb, D., Porath, A., Gilutz, H.: Defining and measuring physicians' responses to clinical reminders. J Biomed Inform. 42, 317–326 (2009). https://doi.org/10.1016/j.jbi.2008.10.001.
18. Backman, R., Bayliss, S., Moore, D., Litchfield, I.: Clinical reminder alert fatigue in healthcare: a systematic literature review protocol using qualitative evidence. Syst Rev. 6, (2017). https://doi.org/10.1186/s13643-017-0627-z.
19. Wendt, T., Knaup-Gregori, P., Winter, A.: Decision support in medicine: a survey of problems of user acceptance. Stud Health Technol Inform. 77, 852–856 (2000).
20. Jaspers, M.W.M., Smeulers, M., Vermeulen, H., Peute, L.W.: Effects of clinical decision-support systems on practitioner performance and patient outcomes: a synthesis of high-quality systematic review findings. J Am Med Inform Assoc. 18, 327–334 (2011). https://doi.org/10.1136/amiajnl-2011-000094.
21. Maeder, A., Poultney, N., Morgan, G., Lippiatt, R.: Patient Compliance in Home-Based Self-Care Telehealth Projects. J Telemed Telecare. 21, 439–442 (2015). https://doi.org/10.1177/1357633X15612382.
22. Gardner, R.M.: Clinical Information Systems – From Yesterday to Tomorrow. Yearb Med Inform. S62–S75 (2016). https://doi.org/10.15265/IYS-2016-s010.
23. Cahan, A., Cimino, J.J.: A Learning Health Care System Using Computer-Aided Diagnosis. J. Med. Internet Res. 19, e54 (2017). https://doi.org/10.2196/jmir.6663.
24. Routledge International Handbook of Participatory Design. Simonsen, J. and T. Robinson (Eds.) (2013): Routledge International Handbook of Participatory Design, Routledge.
25. Luna, D.R., Rizzato Lede, D.A., Otero, C.M., Risk, M.R., González Bernaldo de Quirós, F.: User-centered design improves the usability of drug-drug interaction alerts: Experimental comparison of

interfaces. J Biomed Inform. 66, 204–213 (2017). https://doi.org/10.1016/j.jbi.2017.01.009.

26. Jeffery, A.D., Novak, L.L., Kennedy, B., Dietrich, M.S., Mion, L.C.: Participatory design of probability-based decision support tools for in-hospital nurses. J Am Med Inform Assoc. 24, 1102–1110 (2017). https://doi.org/10.1093/jamia/ocx060.
27. Luna, D., Otero, C., Almerares, A., Stanziola, E., Risk, M., González Bernaldo de Quirós, F.: Participatory design for drug-drug interaction alerts. Stud Health Technol Inform. 210, 45–49 (2015).
28. Al-Itejawi, H.H.M., van Uden-Kraan, C.F., Vis, A.N., Nieuwenhuijzen, J.A., Hofstee, M.J.A., van Moorselaar, R.J.A., Verdonck-de Leeuw, I.M.: Development of a patient decision aid for the treatment of localised prostate cancer: a participatory design approach. J Clin Nurs. 25, 1131–1144 (2016). https://doi.org/10.1111/jocn.13120.
29. Jessen, S., Mirkovic, J., Ruland, C.M.: Creating Gameful Design in mHealth: A Participatory Co-Design Approach. JMIR Mhealth Uhealth. 6, e11579 (2018). https://doi.org/10.2196/11579.
30. Luna, D., Otero, C., Risk, M., Stanziola, E., González Bernaldo de Quirós, F.: Impact of Participatory Design for Drug-Drug Interaction Alerts. A Comparison Study Between Two Interfaces. Stud Health Technol Inform. 228, 68–72 (2016).
31. Cresswell, K., Majeed, A., Bates, D.W., Sheikh, A.: Computerised decision support systems for healthcare professionals: an interpretative review. Inform Prim Care. 20, 115–128 (2012).
32. Bright, T.J., Wong, A., Dhurjati, R., Bristow, E., Bastian, L., Coeytaux, R.R., Samsa, G., Hasselblad, V., Williams, J.W., Musty, M.D., Wing, L., Kendrick, A.S., Sanders, G.D., Lobach, D.: Effect of clinical decision-support systems: a systematic review. Ann. Intern. Med. 157, 29–43 (2012). https://doi.org/10.7326/0003-4819-157-1-201207030-00450.
33. Ortiz, D.R., Maia, F. de O.M., Ortiz, D.C.F., Peres, H.H.C., Sousa, P.A.F. de: Computerized clinical decision support system utilization in nursing: a scoping review protocol. JBI Database System Rev Implement Rep. 15, 2638–2644 (2017). https://doi.org/10.11124/JBISRIR-2016-003184.
34. Curtain, C., Peterson, G.M.: Review of computerized clinical decision support in community pharmacy. J Clin Pharm Ther. 39, 343–348 (2014). https://doi.org/10.1111/jcpt.12168.
35. Baysari, M.T., Lehnbom, E.C., Li, L., Hargreaves, A., Day, R.O., Westbrook, J.I.: The effectiveness of information technology to improve antimicrobial prescribing in hospitals: A systematic review and meta-analysis. Int J Med Inform. 92, 15–34 (2016). https://doi.org/10.1016/j.ijmedinf.2016.04.008.
36. Reis, W.C., Bonetti, A.F., Bottacin, W.E., Reis, A.S., Souza, T.T., Pontarolo, R., Correr, C.J., Fernandez-Llimos, F.: Impact on process results of clinical decision support systems (CDSSs) applied to medication use: overview of systematic reviews. Pharm Pract

(Granada). 15, 1036 (2017). https://doi.org/10.18549/PharmPract.2017.04.1036.

37. Porat, T., Delaney, B., Kostopoulou, O.: The impact of a diagnostic decision support system on the consultation: perceptions of GPs and patients. BMC Med Inform Decis Mak. 17, (2017). https://doi.org/10.1186/s12911-017-0477-6.
38. Miller, K., Mosby, D., Capan, M., Kowalski, R., Ratwani, R., Noaiseh, Y., Kraft, R., Schwartz, S., Weintraub, W.S., Arnold, R.: Interface, information, interaction: a narrative review of design and functional requirements for clinical decision support. J Am Med Inform Assoc. 25, 585–592 (2018). https://doi.org/10.1093/jamia/ocx118.
39. Kawamoto, K., Houlihan, C.A., Balas, E.A., Lobach, D.F.: Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. BMJ. 330, 765 (2005). https://doi.org/10.1136/bmj.38398.500764.8F.
40. Wu, H.W., Davis, P.K., Bell, D.S.: Advancing clinical decision support using lessons from outside of healthcare: an interdisciplinary systematic review. BMC Med Inform Decis Mak. 12, 90 (2012). https://doi.org/10.1186/1472-6947-12-90.
41. Bernell, S., Howard, S.W.: Use Your Words Carefully: What Is a Chronic Disease? Front Public Health. 4, (2016). https://doi.org/10.3389/fpubh.2016.00159.
42. CDC - NCHS - National Center for Health Statistics, https://www.cdc.gov/nchs/index.htm, last accessed 2019/07/09.
43. About Chronic Diseases | CDC, https://www.cdc.gov/chronicdisease/about/index.htm, last accessed 2019/07/09.
44. WHO | Integrated chronic disease prevention and control, https://www.who.int/chp/about/integrated_cd/en/, last accessed 2019/07/09.
45. Agency for Healthcare Research & Quality, https://www.ahrq.gov/, last accessed 2019/07/09.
46. Multiple Chronic Conditions Chartbook. 52.
47. Chapel, J.M., Ritchey, M.D., Zhang, D., Wang, G.: Prevalence and Medical Costs of Chronic Diseases Among Adult Medicaid Beneficiaries. Am J Prev Med. 53, S143–S154 (2017). https://doi.org/10.1016/j.amepre.2017.07.019.
48. World Health Organization: Noncommunicable diseases country profiles 2014. (2014).
49. States of Health in EU - Italy. (:unav). https://doi.org/10.1787/888933593627.
50. Barondess, J.A.: Scanning the Chronic Disease Terrain: Prospects and Opportunities. Trans Am Clin Climatol Assoc. 125, 45–56 (2014).
51. Meetoo, D.: Chronic diseases: the silent global epidemic. Br J Nurs. 17, 1320–1325 (2008). https://doi.org/10.12968/bjon.2008.17.21.31731.
52. McClusky, J.: “Epidemic” of chronic disease? BMJ. 339, b3753 (2009). https://doi.org/10.1136/bmj.b3753.

53. Suhrcke, M.: Chronic disease: an economic perspective. Oxford Health Alliance, London (2006).
54. YOUSEFI, M., ASSARI ARANI, A., SAHABI, B., KAZEMNEJAD, A., FAZAELI, S.: Household Health Costs: Direct, Indirect and Intangible. Iran J Public Health. 43, 202–209 (2014).
55. Health and Economic Costs of Chronic Disease | CDC, https://www.cdc.gov/chronicdisease/about/costs/index.htm, last accessed 2019/07/09.
56. Rapporto Osservasalute 2018 | Osservatorio sulla Salute, https://www.osservatoriosullasalute.it/osservasalute/rapporto-osservatorio-2018, last accessed 2019/07/10.
57. Milani, R.V., Bober, R.M., Lavie, C.J.: The Role of Technology in Chronic Disease Care. Prog Cardiovasc Dis. 58, 579–583 (2016). https://doi.org/10.1016/j.pcad.2016.01.001.
58. Kooij, L., Groen, W.G., van Harten, W.H.: The Effectiveness of Information Technology-Supported Shared Care for Patients With Chronic Disease: A Systematic Review. J. Med. Internet Res. 19, e221 (2017). https://doi.org/10.2196/jmir.7405.
59. Milani, R.V., Franklin, N.C.: The Role of Technology in Healthy Living Medicine. Prog Cardiovasc Dis. 59, 487–491 (2017). https://doi.org/10.1016/j.pcad.2017.02.001.
60. Raghupathi, W., Raghupathi, V.: An Empirical Study of Chronic Diseases in the United States: A Visual Analytics Approach to Public Health. Int J Environ Res Public Health. 15, (2018). https://doi.org/10.3390/ijerph15030431.
61. Dwarswaard, J., Bakker, E.J.M., van Staa, A., Boeije, H.R.: Self-management support from the perspective of patients with a chronic condition: a thematic synthesis of qualitative studies. Health Expect. 19, 194–208 (2016). https://doi.org/10.1111/hex.12346.
62. Lai, A.M., Hsueh, P.-Y.S., Choi, Y.K., Austin, R.R.: Present and Future Trends in Consumer Health Informatics and Patient-Generated Health Data. Yearb Med Inform. 26, 152–159 (2017). https://doi.org/10.15265/IY-2017-016.
63. Heikenfeld, J., Jajack, A., Rogers, J., Gutruf, P., Tian, L., Pan, T., Li, R., Khine, M., Kim, J., Wang, J., Kim, J.: Wearable sensors: modalities, challenges, and prospects. Lab Chip. 18, 217–248 (2018). https://doi.org/10.1039/c7lc00914c.
64. Andreu-Perez, J., Leff, D.R., Ip, H.M.D., Yang, G.: From Wearable Sensors to Smart Implants-–Toward Pervasive and Personalized Healthcare. IEEE Transactions on Biomedical Engineering. 62, 2750–2762 (2015). https://doi.org/10.1109/TBME.2015.2422751.
65. Majumder, S., Mondal, T., Deen, M.J.: Wearable Sensors for Remote Health Monitoring. Sensors (Basel). 17, (2017). https://doi.org/10.3390/s17010130.
66. Baig, M.M., GholamHosseini, H., Moqeem, A.A., Mirza, F., Lindén, M.: A Systematic Review of Wearable Patient Monitoring Systems - Current Challenges and Opportunities for Clinical Adoption. J Med Syst. 41, 115 (2017). https://doi.org/10.1007/s10916-017-0760-1.

67. Guk, K., Han, G., Lim, J., Jeong, K., Kang, T., Lim, E.-K., Jung, J.: Evolution of Wearable Devices with Real-Time Disease Monitoring for Personalized Healthcare. Nanomaterials (Basel). 9, (2019). https://doi.org/10.3390/nano9060813.
68. Bhavnani, S.P., Sitapati, A.M.: Virtual Care 2.0-a Vision for the Future of Data-Driven Technology-Enabled Healthcare. Curr Treat Options Cardiovasc Med. 21, 21 (2019). https://doi.org/10.1007/s11936-019-0727-2.
69. Weichert, I.: "My watch kept on alarming all night about my heart rate": diagnosis of asymptomatic atrial fibrillation with fast ventricular response in a patient with a recent TIA as the result of a smartwatch alarm. Oxf Med Case Reports. 2019, omz014 (2019). https://doi.org/10.1093/omcr/omz014.
70. Chen, H., Goldberg, M.S.: The effects of outdoor air pollution on chronic illnesses. Mcgill J Med. 12, 58–64 (2009).
71. Yi, W.Y., Lo, K.M., Mak, T., Leung, K.S., Leung, Y., Meng, M.L.: A Survey of Wireless Sensor Network Based Air Pollution Monitoring Systems. Sensors (Basel). 15, 31392–31427 (2015). https://doi.org/10.3390/s151229859.
72. PULSE, http://www.project-pulse.eu/, last accessed 2019/07/23.
73. Collins, K.J.: Low indoor temperatures and morbidity in the elderly. Age Ageing. 15, 212–220 (1986). https://doi.org/10.1093/ageing/15.4.212.
74. Krieger, J., Higgins, D.L.: Housing and Health: Time Again for Public Health Action. Am J Public Health. 92, 758–768 (2002).
75. Kenny, G.P., Flouris, A.D., Yagouti, A., Notley, S.R.: Towards establishing evidence-based guidelines on maximum indoor temperatures during hot weather in temperate continental climates. Temperature. 6, 11–36 (2019). https://doi.org/10.1080/23328940.2018.1456257.
76. Hignett, S., Masud, T.: A review of environmental hazards associated with in-patient falls. Ergonomics. 49, 605–616 (2006). https://doi.org/10.1080/00140130600568949.
77. Mackenzie, L., Byles, J., Higginbotham, N.: Reliability of the Home Falls and Accidents Screening Tool (HOME FAST) for identifying older people at increased risk of falls. Disabil Rehabil. 24, 266–274 (2002).
78. Pynoos, J., Steinman, B.A., Nguyen, A.Q.D.: Environmental assessment and modification as fall-prevention strategies for older adults. Clin. Geriatr. Med. 26, 633–644 (2010). https://doi.org/10.1016/j.cger.2010.07.001.
79. WHO Global Report on Falls Prevention in Older Age. World Health Organization (2008).
80. Kamel Boulos, M.N., Berry, G.: Real-time locating systems (RTLS) in healthcare: a condensed primer. Int J Health Geogr. 11, 25 (2012). https://doi.org/10.1186/1476-072X-11-25.
81. Fitbit Official Site for Activity Trackers & More, https://www.fitbit.com/home, last accessed 2019/07/19.

82. Heart Rate Monitors, activity trackers and bike computers | Polar Global, https://www.polar.com/en, last accessed 2019/07/19.
83. Garmin International | Home, https://www.garmin.com/en-US/, last accessed 2019/07/19.
84. Somboon, T., Grigg-Damberger, M.M., Foldvary-Schaefer, N.: Night Stepping: Fitbit Cracks the Case. J Clin Sleep Med. 15, 355–357 (2019). https://doi.org/10.5664/jcsm.7646.
85. Cook, J.D., Eftekari, S.C., Dallmann, E., Sippy, M., Plante, D.T.: Ability of the Fitbit Alta HR to quantify and classify sleep in patients with suspected central disorders of hypersomnolence: A comparison against polysomnography. J Sleep Res. e12789 (2018). https://doi.org/10.1111/jsr.12789.
86. de Zambotti, M., Goldstone, A., Claudatos, S., Colrain, I.M., Baker, F.C.: A validation study of Fitbit Charge 2™ compared with polysomnography in adults. Chronobiol. Int. 1–12 (2017). https://doi.org/10.1080/07420528.2017.1413578.
87. Salvi, E., Lanzola, G., Quaglini, S., Panzarasa, S., Bagarotti, R., Rodi, L., Larizza, C., Sacchi, L.: Permutation Entropy Applied to Fitbit Data: Long-Term Sleep Analysis on One Healthy Subject. Stud Health Technol Inform. 261, 156–161 (2019).
88. Hunter, J.C., Handing, E.P., Casanova, R., Kuchibhatla, M., Lutz, M.W., Saldana, S., Plassman, B.L., Hayden, K.M.: Neighborhoods, sleep quality, and cognitive decline: Does where you live and how well you sleep matter? Alzheimers Dement. (2018). https://doi.org/10.1016/j.jalz.2017.10.007.
89. Lee, S.W.H., Ng, K.Y., Chin, W.K.: The impact of sleep amount and sleep quality on glycemic control in type 2 diabetes: A systematic review and meta-analysis. Sleep Med Rev. 31, 91–101 (2017). https://doi.org/10.1016/j.smrv.2016.02.001.
90. Stone, K.L., Ensrud, K.E., Ancoli-Israel, S.: Sleep, insomnia and falls in elderly patients. Sleep Med. 9 Suppl 1, S18-22 (2008). https://doi.org/10.1016/S1389-9457(08)70012-1.
91. Brassington, G.S., King, A.C., Bliwise, D.L.: Sleep problems as a risk factor for falls in a sample of community-dwelling adults aged 64-99 years. J Am Geriatr Soc. 48, 1234–1240 (2000).
92. Stone, K.L., Blackwell, T.L., Ancoli-Israel, S., Cauley, J.A., Redline, S., Marshall, L.M., Ensrud, K.E.: Sleep Disturbances and Increased Risk of Falls in Older Community-Dwelling Men: The Outcomes of Sleep Disorders in Older Men (MrOS Sleep) Study. J Am Geriatr Soc. 62, 299–305 (2014). https://doi.org/10.1111/jgs.12649.
93. Danne, T., Nimri, R., Battelino, T., Bergenstal, R.M., Close, K.L., DeVries, J.H., Garg, S., Heinemann, L., Hirsch, I., Amiel, S.A., Beck, R., Bosi, E., Buckingham, B., Cobelli, C., Dassau, E., Doyle, F.J., Heller, S., Hovorka, R., Jia, W., Jones, T., Kordonouri, O., Kovatchev, B., Kowalski, A., Laffel, L., Maahs, D., Murphy, H.R., Nørgaard, K., Parkin, C.G., Renard, E., Saboo, B., Scharf, M., Tamborlane, W.V., Weinzimer, S.A., Phillip, M.: International Consensus on Use of Continuous Glucose

Monitoring. Diabetes Care. 40, 1631–1640 (2017). https://doi.org/10.2337/dc17-1600.

94. Kebede, M.M., Schuett, C., Pischke, C.R.: The Role of Continuous Glucose Monitoring, Diabetes Smartphone Applications, and Self-Care Behavior in Glycemic Control: Results of a Multi-National Online Survey. J Clin Med. 8, (2019). https://doi.org/10.3390/jcm8010109.
95. Dexcom, https://www.dexcom.com/it-IT, last accessed 2019/07/20.
96. What Is CGM? | Continuous Glucose Monitoring From Medtronic Diabetes, https://www.medtronicdiabetes.com/treatments/continuous-glucose-monitoring, last accessed 2019/07/20.
97. Rodríguez-Rodríguez, I., Rodríguez, J.-V., Zamora-Izquierdo, M.-Á.: Variables to Be Monitored via Biomedical Sensors for Complete Type 1 Diabetes Mellitus Management: An Extension of the "On-Board" Concept. J Diabetes Res. 2018, 4826984 (2018). https://doi.org/10.1155/2018/4826984.
98. FreeStyle Libre Continuous Glucose Monitoring System, https://www.freestylelibre.us/, last accessed 2019/07/18.
99. Ajjan, R.A., Cummings, M.H., Jennings, P., Leelarathna, L., Rayman, G., Wilmot, E.G.: Optimising use of rate-of-change trend arrows for insulin dosing decisions using the FreeStyle Libre flash glucose monitoring system. Diab Vasc Dis Res. 16, 3–12 (2019). https://doi.org/10.1177/1479164118795252.
100. Massa, G.G., Gys, I., Op 't Eyndt, A., Bevilacqua, E., Wijnands, A., Declercq, P., Zeevaert, R.: Evaluation of the FreeStyle® Libre Flash Glucose Monitoring System in Children and Adolescents with Type 1 Diabetes. Horm Res Paediatr. 89, 189–199 (2018). https://doi.org/10.1159/000487361.
101. Kooij, L., Groen, W.G., van Harten, W.H.: The Effectiveness of Information Technology-Supported Shared Care for Patients With Chronic Disease: A Systematic Review. J. Med. Internet Res. 19, e221 (2017). https://doi.org/10.2196/jmir.7405.
102. Smith, S.A., Shah, N.D., Bryant, S.C., Christianson, T.J.H., Bjornsen, S.S., Giesler, P.D., Krause, K., Erwin, P.J., Montori, V.M., Evidens Research Group: Chronic care model and shared care in diabetes: randomized trial of an electronic decision support system. Mayo Clin. Proc. 83, 747–757 (2008). https://doi.org/10.4065/83.7.747.
103. Polisena, J., Tran, K., Cimon, K., Hutton, B., McGill, S., Palmer, K.: Home telehealth for diabetes management: a systematic review and meta-analysis. Diabetes Obes Metab. 11, 913–930 (2009). https://doi.org/10.1111/j.1463-1326.2009.01057.x.
104. Chi, N.-C., Demiris, G.: A systematic review of telehealth tools and interventions to support family caregivers. J Telemed Telecare. 21, 37–44 (2015). https://doi.org/10.1177/1357633X14562734.
105. Harber, P., Redlich, C.A., Hines, S., Filios, M.S., Storey, E.: Recommendations for a Clinical Decision Support System for Work-Related Asthma in Primary Care Settings. J. Occup. Environ. Med. 59,

e231–e235 (2017). https://doi.org/10.1097/JOM.0000000000001182.

106. Ebrahiminia, V., Riou, C., Seroussi, B., Bouaud, J., Dubois, S., Falcoff, H., Venot, A.: Design of a decision support system for chronic diseases coupling generic therapeutic algorithms with guideline-based specific rules. Stud Health Technol Inform. 124, 483–488 (2006).
107. van Weert, J.C.M., van Munster, B.C., Sanders, R., Spijker, R., Hooft, L., Jansen, J.: Decision aids to help older people make health decisions: a systematic review and meta-analysis. BMC Med Inform Decis Mak. 16, 45 (2016). https://doi.org/10.1186/s12911-016-0281-8.
108. Guardian Angel, http://groups.csail.mit.edu/medg/ga/manifesto/GAtr.html, last accessed 2020/01/01.
109. Ciccone, M.M., Aquilino, A., Cortese, F., Scicchitano, P., Sassara, M., Mola, E., Rollo, R., Caldarola, P., Giorgino, F., Pomo, V., Bux, F.: Feasibility and effectiveness of a disease and care management model in the primary health care system for patients with heart failure and diabetes (Project Leonardo). Vasc Health Risk Manag. 6, 297–305 (2010). https://doi.org/10.2147/vhrm.s9252.
110. Hui, C.Y., Walton, R., McKinstry, B., Jackson, T., Parker, R., Pinnock, H.: The use of mobile applications to support self-management for people with asthma: a systematic review of controlled studies to identify features associated with clinical effectiveness and adherence. J Am Med Inform Assoc. 24, 619–632 (2017). https://doi.org/10.1093/jamia/ocw143.
111. Cleveringa, F.G.W., Gorter, K.J., van den Donk, M., van Gijsel, J., Rutten, G.E.H.M.: Computerized decision support systems in primary care for type 2 diabetes patients only improve patients' outcomes when combined with feedback on performance and case management: a systematic review. Diabetes Technol. Ther. 15, 180–192 (2013). https://doi.org/10.1089/dia.2012.0201.
112. Yang, S.-W., Park, K.-H., Zhou, Y.-J.: The Impact of Hypoglycemia on the Cardiovascular System: Physiology and Pathophysiology. Angiology. 67, 802–809 (2016). https://doi.org/10.1177/0003319715623400.
113. Koivikko, M.L., Kenttä, T., Salmela, P.I., Huikuri, H.V., Perkiömäki, J.S.: Changes in cardiac repolarisation during spontaneous nocturnal hypoglycaemia in subjects with type 1 diabetes: a preliminary report. Acta Diabetol. 54, 251–256 (2017). https://doi.org/10.1007/s00592-016-0941-2.
114. Silva, T.P., Rolim, L.C., Sallum Filho, C., Zimmermann, L.M., Malerbi, F., Dib, S.A.: Association between severity of hypoglycemia and loss of heart rate variability in patients with type 1 diabetes mellitus. Diabetes Metab. Res. Rev. 33, (2017). https://doi.org/10.1002/dmrr.2830.
115. LLC, T.: Patient Generated Health Data & Engagement Platform | TapCloud, http://www.tapcloud.com, last accessed 2019/01/31.

116. Avigail: Case Studies - Datos Health - Transforming remote care, https://www.datos-health.com/case-studies/, last accessed 2019/01/31.
117. Home, https://www-int.glooko.com/, last accessed 2019/01/31.
118. Health2Sync, https://www.health2sync.com, last accessed 2019/05/09.
119. Scheiner, G.: CGM Retrospective Data Analysis. Diabetes Technol Ther. 18, S2-14-S2-22 (2016). https://doi.org/10.1089/dia.2015.0281.
120. Welcome to Nightscout, http://www.nightscout.info/, last accessed 2019/05/09.
121. Collaborating with Fitbit to Innovate and Help the Lives of People with Diabetes, https://www.medtronicdiabetes.com/loop-blog/10365-2/, last accessed 2019/05/10.
122. SA, C.G.: Online Platform | The Cellnovo Diabetes Management System, https://www.cellnovo.com/online, last accessed 2019/05/09.
123. Diabetes:M, https://www.diabetes-m.com/, last accessed 2019/07/19.
124. Baumgärtel, D., Mielke, C., Haux, R.: A Review of Decision Support Systems for Smart Homes in the Health Care System. Stud Health Technol Inform. 247, 476–480 (2018).
125. Rashidi, P., Mihailidis, A.: A Survey on Ambient-Assisted Living Tools for Older Adults. IEEE Journal of Biomedical and Health Informatics. 17, 579–590 (2013). https://doi.org/10.1109/JBHI.2012.2234129.
126. Delahoz, Y.S., Labrador, M.A.: Survey on Fall Detection and Fall Prevention Using Wearable and External Sensors. Sensors. 14, 19806–19842 (2014). https://doi.org/10.3390/s141019806.
127. Khattak, A.M., Vinh, L.T., Dang Viet Hung, Truc, P.T.H., Hung, L.X., Guan, D., Pervez, Z., Han, M., Lee, S., Lee, Y.: Context-aware Human Activity Recognition and decision making. In: The 12th IEEE International Conference on e-Health Networking, Applications and Services. pp. 112–118 (2010). https://doi.org/10.1109/HEALTH.2010.5556585.
128. Juarez, J.M., Ochotorena, J.M., Campos, M., Combi, C.: Spatiotemporal data visualisation for homecare monitoring of elderly people. Artif Intell Med. 65, 97–111 (2015). https://doi.org/10.1016/j.artmed.2015.05.008.
129. Majumder, S., Aghayi, E., Noferesti, M., Memarzadeh-Tehran, H., Mondal, T., Pang, Z., Deen, M.J.: Smart Homes for Elderly Healthcare-Recent Advances and Research Challenges. Sensors (Basel). 17, (2017). https://doi.org/10.3390/s17112496.
130. Nguyen, H., Mirza, F., Naeem, M.A., Baig, M.M.: Falls management framework for supporting an independent lifestyle for older adults: a systematic review. Aging Clin Exp Res. 30, 1275–1286 (2018). https://doi.org/10.1007/s40520-018-1026-6.
131. De Backere, F., Van den Bergh, J., Coppers, S., Elprama, S., Nelis, J., Verstichel, S., Jacobs, A., Coninx, K., Ongenae, F., De Turck, F.: Social-

aware Event Handling within the FallRisk Project. Methods Inf Med. 56, 63–73 (2017). https://doi.org/10.3414/ME15-02-0010.

132. Choi, S.D., Guo, L., Kang, D., Xiong, S.: Exergame technology and interactive interventions for elderly fall prevention: A systematic literature review. Appl Ergon. 65, 570–581 (2017). https://doi.org/10.1016/j.apergo.2016.10.013.
133. Ashworth, N.L., Chad, K.E., Harrison, E.L., Reeder, B.A., Marshall, S.C.: Home versus center based physical activity programs in older adults. Cochrane Database Syst Rev. CD004017 (2005). https://doi.org/10.1002/14651858.CD004017.pub2.
134. Ejupi, A., Gschwind, Y.J., Brodie, M., Zagler, W.L., Lord, S.R., Delbaere, K.: Kinect-based choice reaching and stepping reaction time tests for clinical and in-home assessment of fall risk in older people: a prospective study. Eur Rev Aging Phys Act. 13, (2016). https://doi.org/10.1186/s11556-016-0162-2.
135. Abdolkhani, R., Borda, A., Gray, K.: Quality Management of Patient Generated Health Data in Remote Patient Monitoring Using Medical Wearables - A Systematic Review. Stud Health Technol Inform. 252, 1–7 (2018).
136. Dunn, J., Runge, R., Snyder, M.: Wearables and the medical revolution. Per Med. 15, 429–448 (2018). https://doi.org/10.2217/pme-2018-0044.
137. Codella, J., Partovian, C., Chang, H.-, Chen, C.-: Data quality challenges for person-generated health and wellness data. IBM Journal of Research and Development. 62, 3:1-3:8 (2018). https://doi.org/10.1147/JRD.2017.2762218.
138. Adhikari, R., K. Agrawal, R.: An Introductory Study on Time series Modeling and Forecasting. (2013). https://doi.org/10.13140/2.1.2771.8084.
139. Woldaregay, A.Z., Årsand, E., Botsis, T., Albers, D., Mamykina, L., Hartvigsen, G.: Data-Driven Blood Glucose Pattern Classification and Anomalies Detection: Machine-Learning Applications in Type 1 Diabetes. J. Med. Internet Res. 21, e11030 (2019). https://doi.org/10.2196/11030.
140. Oviedo, S., Vehí, J., Calm, R., Armengol, J.: A review of personalized blood glucose prediction strategies for T1DM patients. Int J Numer Method Biomed Eng. 33, (2017). https://doi.org/10.1002/cnm.2833.
141. Karako, K., Chen, Y., Tang, W.: On medical application of neural networks trained with various types of data. Biosci Trends. 12, 553–559 (2019). https://doi.org/10.5582/bst.2018.01264.
142. Quisel, T., Foschini, L., Signorini, A., Kale, D.C.: Collecting and Analyzing Millions of mHealth Data Streams. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1971–1980. ACM, New York, NY, USA (2017). https://doi.org/10.1145/3097983.3098201.
143. Orphanou, K., Stassopoulou, A., Keravnou, E.: Temporal abstraction and temporal Bayesian networks in clinical domains: a survey. Artif

Intell Med. 60, 133–149 (2014). https://doi.org/10.1016/j.artmed.2013.12.007.

144. Stacey, M., McGregor, C.: Temporal abstraction in intelligent clinical data analysis: a survey. Artif Intell Med. 39, 1–24 (2007). https://doi.org/10.1016/j.artmed.2006.08.002.
145. Lavrac, N., Zupan, B., Kononenko, I., Kukar, M., Keravnou, E.: Intelligent Data Analysis for Medical Diagnosis: Using Machine Learning and Temporal Abstraction. AI Commun. 11, 191–218 (1998).
146. Moskovitch, R., Shahar, Y.: Temporal Data Mining Based on Temporal Abstractions. In: IEEE) ICDM-05 workshop on Temporal Data Mining (2005).
147. Shahar, Y., Musen, M.A.: Knowledge-based temporal abstraction in clinical domains. Artif Intell Med. 8, 267–298 (1996).
148. Sacchi, L., Capozzi, D., Bellazzi, R., Larizza, C.: JTSA: an open source framework for time series abstractions. Comput Methods Programs Biomed. 121, 175–188 (2015). https://doi.org/10.1016/j.cmpb.2015.05.006.
149. Allen, J.F.: Towards a general theory of action and time. Artificial Intelligence. 23, 123–154 (1984). https://doi.org/10.1016/0004-3702(84)90008-0.
150. Cohen, I.G., Mello, M.M.: HIPAA and Protecting Health Information in the 21st Century. JAMA. 320, 231–232 (2018). https://doi.org/10.1001/jama.2018.5630.
151. Izmailova, E.S., Wagner, J.A., Perakslis, E.D.: Wearable Devices in Clinical Trials: Hype and Hypothesis. Clin Pharmacol Ther. 104, 42–52 (2018). https://doi.org/10.1002/cpt.966.
152. EUGDPR – Information Portal, https://eugdpr.org/, last accessed 2019/08/06.
153. com.googlecode.json-simple » json-simple » 1.1.1, https://mvnrepository.com/artifact/com.googlecode.json-simple/json-simple/1.1.1, last accessed 2019/08/12.
154. RFC 6749 - The OAuth 2.0 Authorization Framework, https://tools.ietf.org/html/rfc6749#section-4.1, last accessed 2019/12/19.
155. IETF | About, https://ietf.org/about/, last accessed 2019/12/19.
156. Madkour, M., Benhaddou, D., Tao, C.: Temporal data representation, normalization, extraction, and reasoning: A review from clinical domain. Comput Methods Programs Biomed. 128, 52–68 (2016). https://doi.org/10.1016/j.cmpb.2016.02.007.
157. Jukic, N., Jukic, B., Malliaris, M., Burstein, F., Holsapple, C.: Online Analytical Processing (OLAP) for Decision Support. In: Handbook on Decision Support Systems 1. pp. 259–276 (2008). https://doi.org/10.1007/978-3-540-48713-5_13.
158. MySQL, https://www.mysql.com/, last accessed 2019/09/28.
159. PostgreSQL: The world's most advanced open source database, https://www.postgresql.org/, last accessed 2019/09/28.
160. RavenDB, https://ravendb.net/, last accessed 2019/09/28.

161. Apache Cassandra, http://cassandra.apache.org/, last accessed 2019/09/28.
162. MongoDB, https://www.mongodb.com, last accessed 2019/09/28.
163. Apache CouchDB, http://couchdb.apache.org/, last accessed 2019/09/28.
164. MySQL, https://www.mysql.com/, last accessed 2019/08/12.
165. Hibernate. Everything data., http://hibernate.org/, last accessed 2019/08/12.
166. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: Principles and methods. Data & Knowledge Engineering. 25, 161–197 (1998). https://doi.org/10.1016/S0169-023X(97)00056-6.
167. Protégé, https://protege.stanford.edu/, last accessed 2020/01/01.
168. King, R.C., Villeneuve, E., White, R.J., Sherratt, R.S., Holderbaum, W., Harwin, W.S.: Application of data fusion techniques and technologies for wearable health monitoring. Med Eng Phys. 42, 1–12 (2017). https://doi.org/10.1016/j.medengphy.2016.12.011.
169. Inheritance in Java, https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html, last accessed 2020/01/01.
170. Salvi, E., Bosoni, P., Tibollo, V., Kruijver, L., Calcaterra, V., Sacchi, L., Bellazzi, R., Larizza, C.: Patient-Generated Health Data Integration and Advanced Analytics for Diabetes Management: The AID-GM Platform. Sensors. 20, 128 (2020). https://doi.org/10.3390/s20010128.
171. Drools - Business Rules Management System, https://www.drools.org/, last accessed 2019/08/12.
172. Red Hat, https://www.redhat.com/en, last accessed 2019/08/08.
173. ISO 9241-11:2018(en), Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts, https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en, last accessed 2019/08/10.
174. Nielsen, J., Loranger, H.: Prioritizing Web Usability. Pearson Education (2006).
175. Brooke, J.: System Usability Scale (SUS), /how-to-and-tools/methods/system-usability-scale.html, last accessed 2019/08/08.
176. Legge 9 gennaio 2004, n. 4 aggiornata dal decreto legislativo 10 agosto 2018, n. 106|Agenzia per l'Italia digitale, https://www.agid.gov.it/it/node/79271, last accessed 2019/08/10.
177. Overview | Falls in older people: assessing risk and prevention | Guidance | NICE, https://www.nice.org.uk/guidance/cg161, last accessed 2019/08/16.
178. Sanford, J.A., Pynoos, J., Tejral, A., Browne, A.: Development of a comprehensive assessment for delivery of home modifications. Physical & Occupational Therapy in Geriatrics. 20, 43–55 (2001).
179. Ganz, D.A., Bao, Y., Shekelle, P.G., Rubenstein, L.Z.: Will my patient fall? JAMA. 297, 77–86 (2007). https://doi.org/10.1001/jama.297.1.77.

180. Moncada, L.V.: Management of falls in older persons: a prescription for prevention. Am Fam Physician. 84, 1267–1276 (2011).
181. Lusardi, M.M., Fritz, S., Middleton, A., Allison, L., Wingood, M., Phillips, E., Criss, M., Verma, S., Osborne, J., Chui, K.K.: Determining Risk of Falls in Community Dwelling Older Adults: A Systematic Review and Meta-analysis Using Posttest Probability. J Geriatr Phys Ther. 40, 1–36 (2017). https://doi.org/10.1519/JPT.0000000000000099.
182. Wetherell, J.L., Johnson, K., Chang, D., Ward, S.R., Bower, E.S., Merz, C., Petkus, A.J.: Activity, balance, learning, and exposure (ABLE): a new intervention for fear of falling. Int J Geriatr Psychiatry. 31, 791–798 (2016). https://doi.org/10.1002/gps.4393.
183. Salvi, E., Panzarasa, S., Bagarotti, R., Picardi, M., Boninsegna, R., Sterpi, I., Corbo, M., Lanzola, G., Quaglini, S., Sacchi, L.: NONCADO: A System to Prevent Falls by Encouraging Healthy Habits in Elderly People. In: Riaño, D., Wilk, S., and ten Teije, A. (eds.) Artificial Intelligence in Medicine. pp. 227–232. Springer International Publishing (2019).
184. Spindox: il partner italiano per l'innovazione digitale, https://www.spindox.it/en/, last accessed 2019/08/17.
185. Home – Biomeris Official Site, https://www.biomeris.it/en/home/, last accessed 2019/08/17.
186. Casa di Cura Privata del Policlinico SpA, https://www.ccppdezza.it/, last accessed 2019/08/18.
187. Bagarotti, R., Zini, E., Salvi, E., Sacchi, L., Quaglini, S., Lanzola, G.: An Algorithm for Estimating Gait Parameters Through a Commercial Sensorized Carpet. Presented at the September 1 (2018). https://doi.org/10.1109/RTSI.2018.8548404.
188. Lamb, S.E., McCabe, C., Becker, C., Fried, L.P., Guralnik, J.M.: The optimal sequence and selection of screening test items to predict fall risk in older disabled women: the Women's Health and Aging Study. J. Gerontol. A Biol. Sci. Med. Sci. 63, 1082–1088 (2008).
189. Ivziku, D., Matarese, M., Pedone, C.: Predictive validity of the Hendrich fall risk model II in an acute geriatric unit. Int J Nurs Stud. 48, 468–474 (2011). https://doi.org/10.1016/j.ijnurstu.2010.09.002.
190. Askari, M.: Improving quality of fall prevention and management in elderly patients using information technology: The impact of computerized decision support. (2014).
191. Whitney, J., Close, J.C.T., Lord, S.R., Jackson, S.H.D.: Identification of high risk fallers among older people living in residential care facilities: a simple screen based on easily collectable measures. Arch Gerontol Geriatr. 55, 690–695 (2012). https://doi.org/10.1016/j.archger.2012.05.010.
192. Tromp, A.M., Pluijm, S.M., Smit, J.H., Deeg, D.J., Bouter, L.M., Lips, P.: Fall-risk screening test: a prospective study on predictors for falls in community-dwelling elderly. J Clin Epidemiol. 54, 837–844 (2001).
193. Pluijm, S.M.F., Smit, J.H., Tromp, E. a. M., Stel, V.S., Deeg, D.J.H., Bouter, L.M., Lips, P.: A risk profile for identifying community-dwelling

elderly with a high risk of recurrent falling: results of a 3-year prospective study. Osteoporos Int. 17, 417–425 (2006). https://doi.org/10.1007/s00198-005-0002-0.

194. Stalenhoef, P.A., Diederiks, J.P.M., Knottnerus, J.A., Kester, A.D.M., Crebolder, H.F.J.M.: A risk model for the prediction of recurrent falls in community-dwelling elderly: a prospective cohort study. J Clin Epidemiol. 55, 1088–1094 (2002).
195. Yoo, S.-H., Kim, S.R., Shin, Y.S.: A prediction model of falls for patients with neurological disorder in acute care hospital. J. Neurol. Sci. 356, 113–117 (2015). https://doi.org/10.1016/j.jns.2015.06.027.
196. Deschamps, T., Le Goff, C.G., Berrut, G., Cornu, C., Mignardot, J.-B.: A decision model to predict the risk of the first fall onset. Exp. Gerontol. 81, 51–55 (2016). https://doi.org/10.1016/j.exger.2016.04.016.
197. Haller, M.J., Stalvey, M.S., Silverstein, J.H.: Predictors of control of diabetes: monitoring may be the key. J. Pediatr. 144, 660–661 (2004). https://doi.org/10.1016/j.jpeds.2003.12.042.
198. Rewers, M., Pihoker, C., Donaghue, K., Hanas, R., Swift, P., Klingensmith, G.J.: Assessment and monitoring of glycemic control in children and adolescents with diabetes. Pediatr Diabetes. 10 Suppl 12, 71–81 (2009). https://doi.org/10.1111/j.1399-5448.2009.00582.x.
199. Rewers, M.J., Pillay, K., Beaufort, C. de, Craig, M.E., Hanas, R., Acerini, C.L., Maahs, D.M.: Assessment and monitoring of glycemic control in children and adolescents with diabetes. Pediatric Diabetes. 15, 102–114 (2014). https://doi.org/10.1111/pedi.12190.
200. Schiffrin, A., Belmonte, M.: Multiple daily self-glucose monitoring: its essential role in long-term glucose control in insulin-dependent diabetic patients treated with pump and multiple subcutaneous injections. Diabetes Care. 5, 479–484 (1982). https://doi.org/10.2337/diacare.5.5.479.
201. WHO | Global report on diabetes, http://www.who.int/diabetes/global-report/en/, last accessed 2019/09/02.
202. Salvi, E., Sacchi, L., Madè, A., Calcaterra, V., Bellazzi, R., Larizza, C.: AID-GM: An Advanced System Supporting Continuous Monitoring of T1DM Patients. Stud Health Technol Inform. 247, 616–620 (2018).
203. Calcaterra, V., Sacchi, L., Salvi, E., Larizza, D., Made, A., Schiano, L.M., Montalbano, C., Regalbuto, C., Bellazzi, R., Larizza, C.: AID-GM System (Advanced Intelligent Distant - Glucose Monitoring) to Monitor Health Status and Metabolic Control of Young People with Type 1 Diabetes. Presented at the 57th Annual ESPE August 28 (2018).
204. PrimeFaces | Ultimate UI Framework for Java EE, https://www.primefaces.org/, last accessed 2019/09/02.
205. AmCharts - JavaScript Charts & Maps, https://www.amcharts.com/, last accessed 2019/09/02.
206. Steward, D.A., Hofler, R.A., Thaldorf, C., Milov, D.E.: A method for understanding some consequences of bringing patient-generated

data into health care delivery. Med Decis Making. 30, E1–E13 (2010). https://doi.org/10.1177/0272989X10371829.

207. Ranjan, Y., Rashid, Z., Stewart, C., Conde, P., Begale, M., Verbeeck, D., Boettcher, S., Hyve, Dobson, R., Folarin, A., RADAR-CNS Consortium: RADAR-Base: Open Source Mobile Health Platform for Collecting, Monitoring, and Analyzing Data Using Sensors, Wearables, and Mobile Devices. JMIR Mhealth Uhealth. 7, e11734 (2019). https://doi.org/10.2196/11734.
208. RADAR - Supported Devices, https://radar-base.org/index.php/data-sensors/supported-devices/, last accessed 2019/09/10.
209. Ambrosia Systems: Use NightRider BluCon, FreeStyle Libre Sensor, https://www.ambrosiasys.com/, last accessed 2019/09/09.
210. MiaoMiao FreeStyle Libre Reader, https://miaomiao.cool/, last accessed 2019/09/09.
211. Roy, N., Dubé, R., Després, C., Freitas, A., Légaré, F.: Choosing between staying at home or moving: A systematic review of factors influencing housing decisions among frail older adults. PLoS One. 13, (2018). https://doi.org/10.1371/journal.pone.0189266.
212. Ausserhofer, D., Deschodt, M., De Geest, S., van Achterberg, T., Meyer, G., Verbeek, H., Sjetne, I.S., Malinowska-Lipień, I., Griffiths, P., Schlüter, W., Ellen, M., Engberg, S.: "There's No Place Like Home": A Scoping Review on the Impact of Homelike Residential Care Models on Resident-, Family-, and Staff-Related Outcomes. J Am Med Dir Assoc. 17, 685–693 (2016). https://doi.org/10.1016/j.jamda.2016.03.009.
213. Siegel, J.S.: A Generation of Change: A Profile of America's Older Population. Russell Sage Foundation (1993).
214. Hofman, A., de Jong, P.T.V.M., van Duijn, C.M., Breteler, M.M.B.: Epidemiology of neurological diseases in elderly people: what did we learn from the Rotterdam Study? Lancet Neurol. 5, 545–550 (2006). https://doi.org/10.1016/S1474-4422(06)70473-2.
215. McHugh, M.L.: Interrater reliability: the kappa statistic. Biochem Med (Zagreb). 22, 276–282 (2012).
216. L. Fleiss, J.: Measuring Nominal Scale Agreement Among Many Raters. Psychological Bulletin. 76, 378 (1971). https://doi.org/10.1037/h0031619.

Acronyms

AGP	Average Glucose Profile
AID-GM	Advanced Intelligent Distant – Glucose Monitoring
AUC	Area Under The receiver operating characteristic Curve
BG	Blood Glucose
CCPP	Casa di Cura Privata del Policlinico (Hospital in Milan, in Italy)
CDC	Centers for Disease Control and Prevention
CDSS	Clinical Decision Support System
CGM	Continuous Glucose Monitoring
DST	Daylight Saving Time
FGM	Flash Glucose Monitoring
JS	JavaScript
JSF	JavaServer Faces
JTSA	Java Time Series Abstractor
HR	Heart Rate
HTTP	HyperText Transfer Protocol
IQR	InterQuartile Range
ISO	International Organization for Standards
KB	Knowledge Based
KBTA	Knowledge-Based Temporal Abstraction
MC	Majority Classifier
MCC	Matthews correlation coefficient
NA	Not Available
NPV	Negative Predictive Value
PGHD	Patient-Generated Health Data
PIR	Passive InfraRed
PPV	Positive Predictive Value
RDB	Relational Database
RMS	Rules Management System
SE	Sensitivity
SP	Specificity

SQL	Structured Query Language
SUS	System Usability Scale
TA	Temporal Abstraction
TI	Time Interval
TS	Time Series
WHO	World Health Organization
XML	eXtensible Markup Language

Appendix A

Computing the subject's risk of falling in the NONCADO system

This appendix describes how we computed the subject's risk of falling in the NONCADO system. First, we performed a literature search for models able to estimate the risk of falling of the target users of the NONCADO system, i.e., the non-hospitalized elderly living alone at home. The first subsection of this appendix describes the models identified in this literature review and how they were used for computing the risk score.

We then tested the concordance of the different models in assessing the fall risk of a small set of patients aged over 60 years receiving care at the CCPP hospital, and of a simulated patient population. The results of this analysis are described in the second subsection of this appendix.

Recently, our clinical partners provided us with the observed outcome (i.e., *fallen/not fallen*) collected during a 9 months follow-up study performed on the same patient population considered in the model concordance analysis. Thus, we evaluated the performance of the models on this population by comparing their predictions with the observed outcome. The results of this analysis will be described in the third sub-section of this appendix.

The models

Nine models for predicting the risk of falling in the elderly living alone were found in the literature. The characteristics of the identified models are summarized in Table 26. The first column (*model*) provides the reference to the publication illustrating the model, and specifies the type of model, which can have one among three possible values, namely *rules (R)*, *checklist (C)*, or *logistic regression (L)*. A rule-based model assigns a risk level according to a specific set of rules that can be derived from a classification tree. In a checklist model the subject's risk score is computed as a weighted sum of a specific set of risk factors. Based on the final score and a set of thresholds, the subject is then assigned a fall risk level. The logistic regression type can be considered as a subclass of checklist models, where the risk level is computed according to the results of a multivariate logistic regression. The weights for the variables included in the model are provided by the estimated regression coefficients. For each model, the table provides the following information: the characteristics of the population to which the model is applicable (column *eligibility*), the list of the variables included in the model (column *variables*), complemented with the variable weights when applicable, and the way in which the model assigns the subject a risk score/label (column *scoring type*).

After considering eligibility criteria, Model 3 was left out from further analyses, since it excludes patients with fall history, who are the main target of the NONCADO system.

Table 26: Models for predicting the risk of falling in the elderly living at home.

MODEL	ELIGIBILITY	VARIABLES	SCORING TYPE
1. Type: C Ref: [195]	Adult people with neurological problems.	Cardiovascular disease (2), falls in the last year (3), qualitative evaluation of the walking capability (2-6), overestimation of walking ability (7)	Range: 0-18 Levels: *High* (score > x*); *Low* (score <x*)

2.
Type: C
Ref:
[191]
	Age >= 60 years, expected life >=6 months	Cognitive problems (1), impulsivity /confusion (1), qualitative evaluation of the walking capability (1-2), falls in the last year (1), anxiolytic therapy (1) or antidepressant therapy (1)	Range: 0-7
			Levels:
			0% (score=1);
			10% (score=2);
			23% (score=3);
			45%(score=4);
			62%(score =5);
			82%(score 6);
			100%(score =7)

3. Type: R Ref: [196]	Age >= 65 years, no fall history, able to walk alone for at least 30 seconds.	Diet, age, BMI, fat mass index, visual or hearing problems, balance alterations, foot diseases	Rule-based levels: *At risk*; *Not at risk*
4. Type: C Ref: [194]	Age > 70 years, no neurological diseases	Depression (male:4, female:2), falls in last year (male:6, female:4), reduced grip strength (male:6, female:4), postural sway abnormalities (male:7, female:5)	Range: 0-23 Levels: *High* (score > 13); *Moderate* (8<= score <= 13); *Low* (score < 8)
5. Type: R Ref: [188]	Female aged over 65 years, need for gait assistance	Falls in last year, qualitative evaluation of the walking capability, need for assistance in daily activities, BMI, reduced knee muscle strength, low gait speed	Rule-based calculation of fall probability
6. Type: L Ref: [190]	Age>65 years, having history of falls	Age, qualitative evaluation of the walking capability, fear of falling, orthostatic hypotension	Fall probability according to the regression model
7. Type: C Ref: [192]	Age>65 years	Falls in last year (5), urinary incontinence (3), visual impairment (4), need for assistance in daily living (3)	Range: 0-15 Levels: High (score > x*); Low (score <x*)

8. Type: C Ref: [193]	Age>65 years	Falls in last year (4; 6 if afraid to fall), dizziness (4), need for assistance in daily living (3), low grip strength (3), weight (2), fear of falling (2; 4 if previous falls), pets (2), education (1), alcohol (1)	Range: 0-31 Levels: *High* (score > x*); *Low* (score <x*)
9. Type: C Ref: [189]	Age > 65 years	Impulsivity/confusion (4), depression (2), urinary incontinence (1), dizziness (1), male (1), antiepileptic therapy (2), benzodiazepine therapy (1), difficulty in getting up from chair (1-4)	Range: 0-16 Levels: *High* (score >= 5); *Low* (score < 5)

*in the referenced paper, the authors show results for different threshold values

As a first attempt to integrate the different models, we considered those that provide a binary prediction, either *High/Low* or *At risk/Not at risk.* We considered *High* comparable to *At risk*, and *Low* comparable to *Not at risk*. Thus, we assigned the label (either *High/At risk* or *Low/Not at risk*) that was most often predicted.

Analysis on model concordance

Our medical partner provided us with a retrospective dataset of 123 patients having history of falls. Unfortunately, this real dataset did not contain all the variables necessary for running all the models described in Table *26*. Nevertheless, 112 patients presented all the data necessary to apply models 2, 7 and 9, thus the real-world dataset could be used to compare these three models. To be able to test all the models, we used a simulation approach. We generated a population of 100000 subjects, aged between 65 and 85 years, by sampling variable values according to their probability distribution. Those distributions were derived from the literature, namely from the papers presenting the 9 models, from reviews[214], and from our real-world dataset. Moreover, the generation of simulated patients considered obvious constraints, to avoid, for example, generating a case where the measured *walking capability* is normal and the *subjective overestimation of walking ability* is TRUE, or a case where *Antidepressant drug* is TRUE, and *Depression* is FALSE.

In addition to a set of descriptive statistics, to assess the concordance of the models in rating the patient's risk we used the Cohen Kappa (k), a metric for testing the concordance of all the possible pairs of models[215]. We also computed the Fleiss (F) coefficient, which is the extension of the Cohen k to the case of multiple raters or, in our case, multiple applicable models [216]. Well-accepted thresholds for k are shown in Table 27 which also provides the corresponding interpretation in terms of model agreement. Thresholds for the Fleiss coefficient are shown in Table 28.

Table 27: Interpretation of the k coefficient

RANGE	AGREEMENT
k <= 0	No agreement
[0 ; 0.2]	Poor agreement
[0.21 ; 0.4]	Fair agreement
[0.41 ; 0.6]	Moderate agreement
[0.61 ; 0.8]	Substantial agreement
[0.81 ; 1]	Almost perfect agreement

Table 28: Interpretation of the Fleiss coefficient

RANGE	AGREEMENT
F <= 0.4	Poor agreement
[0.4 ; 0.75]	Intermediate/Good agreement
F > 0.75	Excellent agreement

As a first analysis on the simulated dataset, for each model we computed the percentage of eligible patients and the percentage of patients considered at high risk by the model (Table 29).

Table 29: Fall risk classification per model

MODEL	1*	2	4*	5*	6	7	8	9
% ELIGIBLE CASES	16.4	100	64.3	40.0	100	100	100	100
% PATIENTS AT HIGH RISK	67.1	24.4	1.7	13.7	20.3	28.6	79.4	45.9

*Models 1, 4 and 5 have specific eligibility criteria

The table shows that, even excluding models with very specific eligibility criteria, the model behaviour was quite different. On average, 6 out of the 8 models were applicable to each patient. As anticipated, each model predicts (i.e., votes for) a risk category. Thus, for each patient we considered the following variables:

n_p= number of models that can be applied to the patient,
n_1= number of models voting for high risk,
n_0= number of models voting for low risk,
with n_p=n_1+n_0.

Given n_1 and n_0, the patient can be assigned a label, specifically *high risk* (when n_1>n_0), *low risk* (when n_1<n_0), *none* (when n_1=n_0). To assess the label reliability, we calculated the absolute quantity $|n_1 - n_0|$, i.e. a measure of the *advantage* of that label with respect to the other one. Such indicator will be always 0 for patients being assigned the *none* label. The average *advantage* was 2.65 (±1.7). Then we excluded the cases having the *none* label (i.e., 11% of the sample), and we computed, for patients classified at high or low risk, the *supporter models ratio*, ranging 0-1, i.e. the number of models assigning that label divided by the number of models applicable to the patient. The average *supporter models ratio* was 0.74 (±0.13). Thus, when an informative label is assigned, we can say that its support is on average satisfactory.

Moreover, to formally assess the concordance of the models in rating the patient's risk, we computed the Cohen k for all the possible pairs of models, listed in Table 30. In the table, darker grey intensity indicates lower k level. According to the obtained k values, the best achieved concordance is *fair.* In particular, the k values obtained on the simulated dataset ranged from -0.0045 to 0.3, with a mean value of 0.085. Models 5 and 7 were the most concordant with a k value of 0.3.

Table 30: Cohen k for each pair of models when applied to the simulated dataset

MODELS	**1**	**2**	**4**	**5**	**6**	**7**	**8**	**9**
1	-	0.24	0	0.07	0.11	0.033	-0.004	0.001
2	-	-	0.26	0.20	0.07	0.24	0.05	0.24
4	-	-	-	0.24	0.005	0.03	0.008	0.008
5	-	-	-	-	0.003	0.30	0.05	0.003
6	-	-	-	-	-	0.0006	-0.0004	-0.001
7	-	-	-	-	-	-	0.11	0.03
8	-	-	-	-	-	-	-	0.06
9	-	-	-	-	-	-	-	-

We found 5 models that were applicable to all the simulated patients. Considering the predictions of these models, we computed the Fleiss coefficient, which was negative, indicating lack of concordance, as expected from the obtained paired k values.

We then performed the model concordance analysis on the real-world dataset. As mentioned, only 3 models (i.e., Model 2, 7, and 9) could be compared using this dataset. The results in terms k coefficient are shown in Table 31.

Table 31: Cohen k for each pair of models when applied on the real-world dataset

MODELS	**2**	**7**	**9**
2	-	0.31	0.43
7	-	-	0.18
9	-	-	-

We also computed the Fleiss coefficient, which was negative, confirming the poor/fair agreement among the models.

The experiments described in the section highlight the difficulty in stratifying the non-hospitalized elderly according to their risk of falling. Further work is needed to analyse potential factors influencing the model agreement level.

Analysis on model accuracy

As anticipated, for each model that was applicable to the real-world dataset (i.e., Model 2, 7, and 9) we compared its predictions with the patients' follow-up outcome (either *fallen* or *not fallen*), collected in a 9-months follow-up study performed by our clinical partners. In particular, we computed a set of performance indicators, including accuracy, Matthews correlation coefficient (MCC), sensitivity (SE), specificity (SP), positive predictive value (PPV), negative predictive value (NPV), and area under the receiver operating characteristic curve (AUC). For each model, we compared the obtained indicator values with those reported by the authors of the model, when available. For each model, we adapted the timeframe of the analysis to the duration of the follow-up studies described performed by the authors of the model. In particular, we considered the entire follow-up period for models 1 and 2, and only the first follow-up month for model 3. Table 32 shows the obtained performance indicators, and the performance of the majority classifier (MC) on the same dataset. The performance indicators reported by the authors of the model are shown in brackets when available, while *NA* is used when they were not available.

Table 32: Predictive performances observed by applying the three models and a majority classifier to the CCPP dataset. NA = not available.

INDICATOR	MODEL 1	MODEL 2	MODEL 3	MC
Accuracy	0.62 (NA)	0.67 (NA)	0.51 (NA)	0.63
MCC	0.21 (NA)	0.31 (NA)	0.05 (NA)	NA
AUC	0.69 (0.79)	0.69 (0.65)	0.59 (0.71)	0.5
SE	0.59 (NA)	0.62 (NA)	0.56 (0.86)	0
SP	0.63 (NA)	0.7 (NA)	0.51 (0.43)	1
PPV	0.48 (NA)	0.56 (NA)	0.17 (0.11)	NA
NPV	0.73 (NA)	0.75 (NA)	0.86 (0.97)	0.63

The receiver operating characteristic curves of the three models are shown in Figure 33.

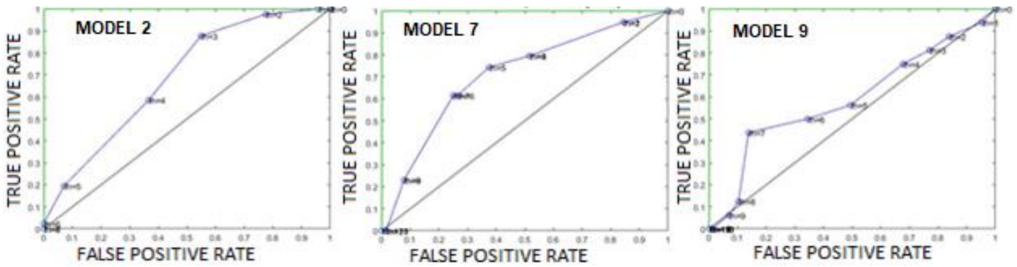


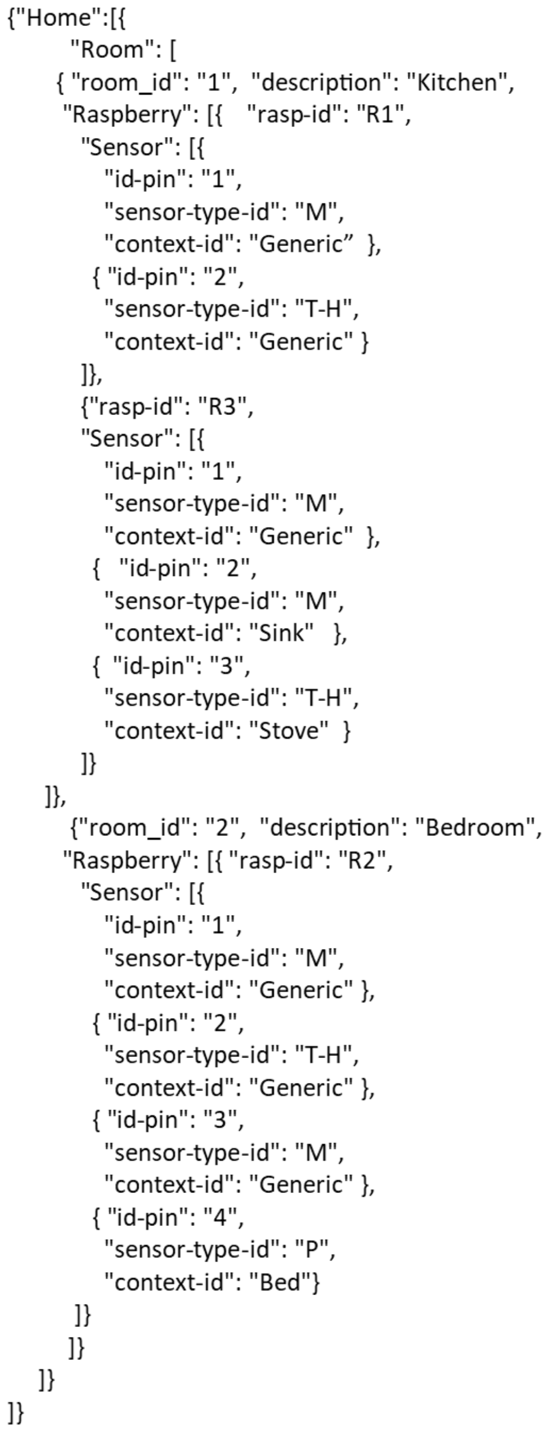
Figure 33: Receiver operating characteristic curves obtained by applying the three models to the real-world dataset.

As shown in Table 32, although models 2 and 7 performed better than the majority classifier, the overall performance is poor. This may be due to ignoring informative clinical variables (e.g., walking speed, specific clinical tests) that are not included in these three models, that are targeted (or applicable) to non-hospitalized patients. On the basis of these results, the need for developing more accurate models became apparent. Further work is needed to integrate the model predictions with variables that can be collected from the patient's medical record and with monitoring data, to build a comprehensive risk score for the individual subject

Appendix B

Representation of the living lab in NONCADO

This appendix contains the JSON document that describes the planimetry of the pilot site in terms of the sensors located in each room.



```
{"Home":[{
        "Room": [
      { "room_id": "1",  "description": "Kitchen",
       "Raspberry": [{    "rasp-id": "R1",
         "Sensor": [{
            "id-pin": "1",
            "sensor-type-id": "M",
            "context-id": "Generic”  },
          { "id-pin": "2",
            "sensor-type-id": "T-H",
            "context-id": "Generic" }
         ]},
        {"rasp-id": "R3",
         "Sensor": [{
            "id-pin": "1",
            "sensor-type-id": "M",
            "context-id": "Generic"  },
          {   "id-pin": "2",
            "sensor-type-id": "M",
            "context-id": "Sink"   },
          {  "id-pin": "3",
            "sensor-type-id": "T-H",
            "context-id": "Stove"  }
         ]}
     ]},
        {"room_id": "2",  "description": "Bedroom",
       "Raspberry": [{ "rasp-id": "R2",
         "Sensor": [{
            "id-pin": "1",
            "sensor-type-id": "M",
            "context-id": "Generic" },
          { "id-pin": "2",
            "sensor-type-id": "T-H",
            "context-id": "Generic" },
          { "id-pin": "3",
            "sensor-type-id": "M",
            "context-id": "Generic" },
          { "id-pin": "4",
            "sensor-type-id": "P",
            "context-id": "Bed"}
        ]}
        ]}
    ]}
]}
```

The environment is composed of two rooms. The first room is a Kitchen, equipped with two raspberries. One raspberry (R1) is connected with one motion (M) sensor and one sensor (type: T-H) that collects both temperature and humidity measurements. The other raspberry (R3) is connected with three sensors:

- One motion sensor with context-id equal to *Generic*, that monitors movement within the room.
- One motion sensor with context-id equal to *Sink*, that monitors movement in the area close to the sink.
- One sensor that monitors temperature and humidity in the area close to the stove.

The second room is the bedroom. It is equipped with one raspberry (R2), having four connections:

- Two motion sensors (connected at the first and third pin) with context-id equal to *Generic*, that monitor movement within the room.
- One sensor with context-id equal to *Generic* that monitors temperature and humidity within the room.
- The pressure mats located under the bed mattress (sensor-type-id = P, context-id=Bed).

Appendix C

Details on the results of the NONCADO pilot study

This appendix contains the tables that report in detail the activities performed by the patients enrolled in the NONCADO system pilot study.

Test use case 1 (Resting in bed)

For each repetition, Table 33 reports the date, the identifier of the patient, and the start time and end time of each activity that composes the test use case (*ACTIVITY WITHIN THE TEST USE CASE* column). The *ACTIVITY DETECTED* column reports whether the CDSS detected that activity successfully (*OK*) or not (*NO*). The TEST USE CASE DETECTED column reports whether the CDSS detected the test use case successfully or not. If at least one of the sub-activities is not properly detected, the test use case is not considered to be properly detected. Errors are highlighted in grey.

Table 33: Results of the pilot study on the NONCADO system, test use case 1 (Resting in bed)

DATE	SUBJECT	ACTIVITY WITHIN THE TEST USE CASE	ACTIVITY DETECTED	TEST USE CASE DETECTED
04/09/2018	S1	LYING IN BED (9:14:01-9:17:52)	OK	OK
		LYING IN BED (9:18:58-9:21:59)	OK	
	S2	LYING IN BED (9:58:29-10:01:35)	OK	OK
		LYING IN BED (10:03:32-10:06:38)	OK	
	S3	LYING IN BED (10:39:52-10:42:14)	OK	OK
		LYING IN BED (10:43:06-10:45:50)	OK	
05/09/2018	S4	LYING IN BED (9:07:47-9:10:38)	OK	OK
		LYING IN BED (9:11:42-9:14:44)	OK	
	S5	LYING IN BED (9:40:34-9:44:07)	OK	OK
		LYING IN BED (9:45:21-9:48:22)	OK	
	S6	LYING IN BED (10:11:27-10:14:27)	OK	OK
		LYING IN BED (10:15:41-10:18:49)	OK	
	S7	LYING IN BED (10:48:08-10:51:06)	OK	OK
		LYING IN BED (10:52:26-10:55:18)	OK	

	S8	LYING IN BED (11:06:43-11:09:22)	OK	OK
		LYING IN BED (11:10:44-11:13:44)	OK	
06/09/2018	S1	LYING IN BED (9:10:18-9:13:34)	OK	OK
		LYING IN BED (9:14:34-9:17:52)	OK	
	S2	LYING IN BED (9:50:00-9:53:57)	OK	OK
		LYING IN BED (9:55:10-9:59:28)	OK	
	S3	LYING IN BED (10:37:50-10:40:50)	OK	OK
		LYING IN BED (10:42:00-10:45:00)	OK	
07/09/2018	S4	LYING IN BED (10:13:46-10:16:50)	OK	OK
		LYING IN BED (10:17:58-10:21:05)	OK	
	S5	LYING IN BED (9:41:19-9:44:58)	OK	OK
		LYING IN BED (9:46:07-9:49:45)	OK	
	S6	LYING IN BED (11:15:18-11:18:31)	OK	OK
		LYING IN BED (11:19:38-11:23:18)	OK	
	S7	LYING IN BED (9:09:57-9:11:11)	OK	OK
		LYING IN BED (9:12:14-9:15:28)	OK	
	S8	LYING IN BED (10:44:18-10:47:33)	OK	OK
		LYING IN BED (10:48:42-10:51:51)	OK	
10/09/2018	S10	LYING IN BED (9:26:29-9:29:50)	OK	OK
		LYING IN BED (9:31:02-9:34:18)	OK	
	S11	LYING IN BED (10:15:23-10:18:53)	OK	OK
		LYING IN BED (10:20:18-10:23:46)	OK	
	S12	LYING IN BED (10:55:21-10:59:34)	OK	OK
		LYING IN BED (11:01:07-11:05:00)	OK	
	S13	LYING IN BED (11:37:29-11:40:49)	OK	OK
		LYING IN BED (11:42:03-11:45:22)	OK	
12/09/2018	S14	LYING IN BED (9:15:46-9:19:08)	OK	OK
		LYING IN BED (9:20:10-9:23:23)	OK	
	S15	LYING IN BED (9:52:47-9:56:27)	OK	OK

		LYING IN BED (9:57:31-10:00:53)	OK	
	S16	LYING IN BED (10:29:28-10:32:50)	OK	OK
		LYING IN BED (10:34:03-10:37:21)	OK	
	S17	LYING IN BED (11:07:21-11:10:49)	NO	NO
		LYING IN BED (11:11:51-11:15:22)	OK	
13/09/2018	S10	LYING IN BED (10:26:08-10:29:27)	OK	OK
		LYING IN BED (10:30:42-10:34:14)	OK	
	S11	LYING IN BED (11:14:22-11:17:41)	OK	OK
		LYING IN BED (11:18:56-11:22:16)	OK	
	S12	LYING IN BED (9:50:10-9:53:58)	OK	OK
		LYING IN BED (9:55:10-9:58:57)	OK	
	S13	LYING IN BED (9:02:41-9:05:59)	OK	OK
		LYING IN BED (9:07:02-9:10:19)	OK	
14/09/2018	S14	LYING IN BED (9:08:30-9:11:30)	OK	OK
		LYING IN BED (9:12:30-9:15:30)	OK	
	S15	LYING IN BED (10:48:55-10:52:14)	OK	OK
		LYING IN BED (10:53:14-10:56:38)	OK	
	S16	LYING IN BED (9:38:35-9:41:58)	OK	OK
		LYING IN BED (9:42:58-9:46:10)	OK	
	S17	LYING IN BED (10:17:10-10:20:22)	OK	OK
		LYING IN BED (10:21:22-10:24:33)	OK	

Test use case 2 (Resting in bed with getting up)

For each repetition, Table 34 reports the date, the identifier of the patient, and the start time and end time of each activity that composes the test use case (*ACTIVITY WITHIN THE TEST USE CASE* column). The *ACTIVITY DETECTED* column reports whether the CDSS detected that activity successfully (*OK*) or not (*NO*). The TEST USE CASE DETECTED column reports whether the CDSS detected the test use case successfully or not. If at least one of the sub-activities is not properly detected, the test use case is not considered to be properly detected. Errors are highlighted in grey.

Table 34: Results of the pilot study on the NONCADO system, test use case 2 (Resting in bed with getting up)

DATE	SUBJECT	ACTIVITY WITHIN THE TEST USE CASE	ACTIVITY DETECTED	TEST USE CASE DETECTED
04/09/2018	S1	LYING IN BED (9:23:04-9:25:56)	OK	OK
		ABSENCE (9:26:24-9:26.55)	OK	
		SITTING ON THE BED (9:27:12- 9:29:10)	OK	
	S2	LYING IN BED (10:07:58-10:11:39)	OK	OK
		ABSENCE (10:11:49-10:13:38)	OK	
		SITTING ON THE BED (10:13:57- 10:15:36)	OK	
	S3	LYING IN BED (10:47:09-10:50:16)	OK	NO
		ABSENCE (10:50:27-10:51:10)	OK	
		SITTING ON THE BED (10:51:16- 10:53:07)	NO	
05/09/2018	S4	LYING IN BED (9:15:48-9:18:06)	OK	OK
		ABSENCE (9:18:16-9:19:19)	OK	
		SITTING ON THE BED (9:19:29- 9:21:30)	OK	
	S5	LYING IN BED (9:49:27-9:51:58)	OK	OK
		ABSENCE (9:52:18-9:53:28)	OK	
		SITTING ON THE BED (9:53:38- 9:55:45)	OK	
	S6	LYING IN BED (10:20:18-10:22:35)	OK	OK
		ABSENCE (10:22:55-10:24:03)	OK	
		SITTING ON THE BED (10:24:13- 10:26:15)	OK	
	S8	LYING IN BED (11:23:06-11:25:13)	OK	OK
		ABSENCE (11:25:23-11:26:25)	OK	
		SITTING ON THE BED (11:26:34- 11:28:40)	OK	
06/09/2018	S1	LYING IN BED (9:31:34-9:34:05)	OK	OK
		ABSENCE (9:34:30-9:35:40)	OK	
		SITTING ON THE BED (9:36:11- 9:38:36)	OK	
	S2	LYING IN BED (10:06:40-10:09:55)	OK	OK
		ABSENCE (10:10:30-10:11:40)	OK	
		SITTING ON THE BED (10:11:54- 10:13:54)	OK	
	S3	LYING IN BED (10:51:26-10:53:26)	OK	OK
		ABSENCE (10:53:35-10:54:35)	OK	
		SITTING ON THE BED (10:54:50- 10:56:50)	OK	
07/09/2018	S4	LYING IN BED (10:27:57-10:30:02)	OK	OK

		ABSENCE (10:30:16- 10:31:18)	OK	
		SITTING ON THE BED (10:31:27- 10:33:28)	OK	
	S5	LYING IN BED (9:59:44- 10:02:14)	OK	OK
		ABSENCE (10:02:36- 10:03:40)	OK	
		SITTING ON THE BED (10:03:57- 10:05:42)	OK	
	S6	LYING IN BED (11:32:40- 11:36:21)	OK	OK
		ABSENCE (11:36:34- 11:37:44)	OK	
		SITTING ON THE BED (11:37:55- 11:40:21)	OK	
	S7	LYING IN BED (9:25:40- 9:27:54)	OK	OK
		ABSENCE (9:28:07-9:29:17)	OK	
		SITTING ON THE BED (9:29:35- 9:31:37)	OK	
	S8	LYING IN BED (10:59:41- 11:02:07)	OK	OK
		ABSENCE (11:02:21- 11:03:26)	OK	
		SITTING ON THE BED (11:03:39- 11:05:39)	OK	
10/09/2018	S10	LYING IN BED (9:45:31- 9:47:57)	OK	OK
		ABSENCE (9:48:08-9:49:15)	OK	
		SITTING ON THE BED (9:49:29- 9:51:34)	OK	
	S11	LYING IN BED (10:31:17- 10:33:32)	OK	OK
		ABSENCE (10:33:40- 10:35:03)	OK	
		SITTING ON THE BED (10:35:09- 10:37:15)	OK	
	S12	LYING IN BED (11:15:16- 11:18:36)	OK	OK
		ABSENCE (11:19:06- 11:20:11)	OK	
		SITTING ON THE BED (11:20:38- 11:22:46)	OK	
	S13	LYING IN BED (11:53:03- 11:55:26)	OK	OK
		ABSENCE (11:55:40- 11:56:47)	OK	
		SITTING ON THE BED (11:57:00- 11:59:03)	OK	
12/09/2018	S14	LYING IN BED (9:34:57- 9:37:11)	OK	OK
		ABSENCE (9:37:16-9:38:19)	OK	
		SITTING ON THE BED (9:38:26- 9:40:29)	OK	
	S15	LYING IN BED (10:11:18- 10:13:29)	OK	OK
		ABSENCE (10:13:38- 10:14:41)	OK	
		SITTING ON THE BED (10:14:47- 10:16:50)	OK	
	S16	LYING IN BED (10:45:45- 10:48:13)	OK	OK

		ABSENCE (10:48:24-10:49:31)	OK	
		SITTING ON THE BED (10:49:46- 10:51:50)	OK	
	S17	LYING IN BED (11:24:44-11:24:56)	OK	OK
		ABSENCE (11:27:05-11:28:09)	OK	
		SITTING ON THE BED (11:28:18- 11:30:20)	OK	
13/09/2018	S10	LYING IN BED (10:43:22-10:45:42)	OK	OK
		ABSENCE (10:45:53-10:47:06)	OK	
		SITTING ON THE BED (10:47:12- 10:49:22)	OK	
	S11	LYING IN BED (11:30:38-11:32:54)	OK	OK
		ABSENCE (11:33:02-11:34:12)	OK	
		SITTING ON THE BED (11:34:21- 11:35:25)	OK	
	S12	LYING IN BED (10:08:18-10:11:19)	OK	OK
		ABSENCE (10:11:33-10:12:42)	OK	
		SITTING ON THE BED (10:12:49- 10:14:54)	OK	
	S13	LYING IN BED (9:18:55-9:21:14)	OK	OK
		ABSENCE (9:21:22-9:22:26)	OK	
		SITTING ON THE BED (9:22:34- 9:24:38)	OK	
14/09/2018	S14	LYING IN BED (9:25:15-9:27:15)	OK	OK
		ABSENCE (9:27:26-9:28:26)	OK	
		SITTING ON THE BED (9:28:39- 9:30:39)	OK	
	S15	LYING IN BED (11:05:50-11:08:12)	OK	OK
		ABSENCE (11:08:24-11:09:24)	OK	
		SITTING ON THE BED (11:09:35- 11:11:35)	OK	
	S16	LYING IN BED (9:52:06-9:54:19)	OK	OK
		ABSENCE (9:54:30-9:55:30)	OK	
		SITTING ON THE BED (9:55:39- 9:57:39)	OK	
	S17	LYING IN BED (10:32:46-10:35:05)	OK	OK
		ABSENCE (10:35:20-10:36:40)	OK	
		SITTING ON THE BED (10:36:48- 10:38:55)	OK	

Test use case 3 (Cooking)

For each repetition, Table 35 reports the date, the identifier of the patient, and the start time and end time of the cooking activity. The *CDSS* column reports whether the CDSS detected that activity successfully (*OK*) or not (*NO*). Errors are highlighted in grey.

Table 35: Results of the pilot study on the NONCADO system, test use case 3 (Cooking)

DATE	SUBJECT	TIME INTERVAL	CDSS
04/09/2018	S1	9:34:03- 9:41:05	OK
	S2	10:17:02- 10:21:06	OK
	S3	10:54:50- 10:59:35	OK
05/09/2018	S4	9:22:42- 9:28:09	OK
	S5	9:56:52- 10:00:46	OK
	S6	10:27:07- 10:35:06	OK
	S8	11:14:55- 11:20:26	OK
06/09/2018	S1	9:22:02- 9:29:30	OK
	S2	10:00:40- 10:05:20	OK
	S3	10:46:02- 10:51:05	OK
07/09/2018	S4	10:22:26- 10:25:27	OK
	S5	9:50:51- 9:55:08	OK
	S6	11:25:24- 11:30:28	OK
	S7	9:16:32- 9:22:14	OK
	S8	10:52:58- 10:57:50	OK
10/09/2018	S10	9:35:30- 9:39:45	OK
	S11	10:24:35- 10:28:18	OK
	S12	11:06:29- 11:11:28	OK
	S13	11:46:16- 11:50:12	OK
12/09/2018	S14	9:25:08- 9:31:06	OK
	S15	10:02:18- 10:07:27	OK
	S16	10:38:47- 10:42:23	NO
	S17	11:17:00- 11:21:35	OK
13/09/2018	S10	10:35:31- 10:40:15	OK
	S11	11:22:56- 11:28:26	OK
	S12	10:00:48- 10:05:27	OK
	S13	9:11:25- 9:15:38	OK
14/09/2018	S14	9:16:30- 9:22:39	OK
	S15	10:57:57- 11:03:08	OK
	S16	9:47:29- 9:50:01	OK
	S17	10:25:42- 10:29:42	OK

Test use case 4 (Exiting the kitchen while cooking)

For each repetition, Table 36 reports the date and the identifier of the patient. The third column reports the time the patient turned on the stove and, in brackets, the time he/she left the kitchen. The *COOKING ACTIVITY DETECTED* column reports whether the CDSS detected the cooking activity successfully (*OK*) or not (*NO*). The *EXIT DETECTED* column reports whether the photocell detected the exit successfully or not. The *CDSS* column reports whether the CDSS detected the test use case successfully or not. If at least one of the two sub-activities is not properly detected, the test use case is not considered to be properly detected. Errors are highlighted in grey.

Table 36: Results of the pilot study on the NONCADO system, test use case 4 (Exiting the kitchen while cooking)

DATE	SUBJECT	STOVE ACTIVATED (EXIT)	COOKING ACTIVITY DETECTED	EXIT DETECTED	CDSS
04/09/2018	S1	9.47:25 (09:50:02)	OK	NO	NO
	S2	10:24:01 (10:26:46)	OK	OK	OK
	S3	11:05:49 (11:09:06)	OK	OK	OK
05/09/2018	S4	9:34:30 (9:37:51)	OK	OK	OK
	S5	10:05:51 (10:08:52)	OK	OK	OK
	S6	10:42:29 (10:46:15)	NO	OK	NO
	S7	10:57:48 (11:01:01)	NO	NO	NO
	S8	11:30:02 (11:35:01)	OK	OK	OK
06/09/2018	S1	9:42:07 (9:47:49)	OK	OK	OK
	S2	10:18:30 (10:20:01)	OK	OK	OK
	S3	11:01:00 (11:08:04)	OK	NO	NO
07/09/2018	S4	10:34:29 (10:41:54)	OK	OK	OK
	S5	10:07:19 (10:11:41)	OK	OK	OK
	S6	11:41:27 (11:46:05)	OK	OK	OK
	S7	9:32:51 (9:37:47)	OK	OK	OK
	S8	11:06:51 (11:11:51)	OK	OK	OK
10/09/2018	S10	10:00:22 (10:04:52)	OK	OK	OK
	S11	10:37:30 (10:41:39)	OK	OK	OK
	S12	11:24:25 (11:29:24)	OK	OK	OK
	S13	12:00:13 (12:03:30)	NO	OK	NO
12/09/2018	S14	9:42:11 (9:46:49)	OK	OK	OK
	S15	10:20:10 (10:24:10)	OK	OK	OK
	S16	10:53:57 (10:57:39)	NO	OK	NO
	S17	11:31:22 (11:35:03)	OK	OK	OK

13/09/2018	S10	10:50:20 (10:54:15)	OK	OK	OK
	S11	11:37:22 (11:40:43)	OK	NO	NO
	S12	10:15:55 (10:21:30)	OK	OK	OK
	S13	9:25:35 (9:29:50)	OK	OK	OK
14/09/2018	S14	9:31:30 (9:35:20)	OK	OK	OK
	S15	11:12:22 (11:17:00)	OK	OK	OK
	S16	9:58:29 10:02:56	OK	OK	OK
	S17	10:39:48 (10:43:20)	OK	OK	OK

Test use case 5 (Washing the dishes)

For each repetition, Table 37 reports the date, the identifier of the patient, and the time interval in which the considered activity occurred. The *CDSS* column reports whether the CDSS detected the test use case successfully (*OK*) or not (*NO*). Errors are highlighted in grey.

Table 37: Results of the pilot study on the NONCADO system, test use case 5 (Washing the dishes)

DATE	SUBJECT	TIME INTERVAL	CDSS
04/09/2018	S1	9:44:01- 9:46:02	OK
	S2	10:21:45 -10:22:55	OK
	S3	11:00:20 – 11:01:36	OK
05/09/2018	S4	9:29:00 – 9:30:03	OK
	S5	10:01:24 – 10:03:06	OK
	S6	10:35:40 – 10:36:44	OK
	S7	10:56:06 – 10:57:17	OK
	S8	11:20:47 – 11:22:08	OK
06/09/2018	S1	9:39:50 – 9:41:15	OK
	S2	10:15:17 – 10:17:15	OK
	S3	10:58:40 – 11:00:00	OK
07/09/2018	S4	10:25:50 – 10:26:41	NO
	S5	9:55:32 – 9:57:08	NO
	S6	10:30:40 – 11:31:48	NO
	S7	9:22:59 – 9:24:25	NO
	S8	10:58:06 – 10:59:04	NO
10/09/2018	S10	9:40:38 – 9:44:28	OK
	S11	10:29:02 – 10:30:27	OK
	S12	11:12:03 – 11:13:12	OK
	S13	11:50:36 – 11:51:57	OK
12/09/2018	S14	9:31:53 – 9:33:16	OK
	S15	10:08:10 – 10:10:11	OK
	S16	10:43:08 – 10:44:42	NO
	S17	11:22:01 – 11:23:30	OK

13/09/2018	S10	10:40:43 – 10:42:44	OK
	S11	11:28:43 – 11:30:05	OK
	S12	10:05:52 – 10:07:30	OK
	S13	9:16:26 – 9:17:45	OK
14/09/2018	S14	9:23:20 – 9:24:40	OK
	S15	11:03:30 -11:04:40	OK
	S16	9:50:15 – 9:51:15	OK
	S17	10:30:30 – 10:31:50	OK

Appendix D

JTSA workflows implemented in AID-GM

This appendix summarizes the 10 JTSA workflows implemented in AID-GM. For further details on the meaning of the parameters refer to Section 2.4.

Hypoglycemia and Hyperglycemia

The workflow for detecting hypoglycemia is composed of a single pipeline, consisting of two steps (Figure 34). The corresponding parameter values are listed in Table 38. As already stated, the threshold for defining hypoglycemic and hyperglycemic episodes (TH_{HYPO} and TH_{HYPER} in the table) are patient-specific and defined by the diabetologists.

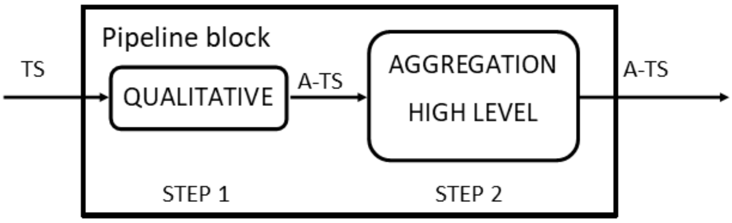


Figure 34: JTSA workflow for detecting Hypoglycemia

Table 38: Parameters of the JTSA workflow for detecting hypoglycemia

STEP	ALGORITHM	PARAMETERS

Step 1	Qualitative	th= TH_{HYPO}, TH_{HYPER}
		label=hypo, normal, hyper

Step 2	Aggregation HighLevel	gap=60
		minLen=13
		granularity=MINUTES
		label=hypo
		levels=hypo

The workflow for detecting hyperglycemia has the same structure shown in Figure 34. The algorithm parameters have the same values listed for the hypoglycemic episodes, excluding two the *label* and *levels* parameters in the second step of the workflow (Table 39).

Table 39: Parameters of the JTSA workflow for detecting hyperglycemia

STEP	ALGORITHM	PARAMETERS

Step 1	Qualitative	th= $\mathrm{TH}_{\mathrm{HYPO}}$, $\mathrm{TH}_{\mathrm{HYPER}}$
		label=hypo, normal, hyper

Step 2	Aggregation HighLevel	granularity=MINUTES
		gap=60
		minLen=13
		label=hyper
		levels=hyper

Increasing BG value/ Decreasing BG value

Increasing and decreasing episodes are defined as a variation of the BG level of at least 15 mg/dl every 15 minutes, lasting for at least 35 minutes. The workflow for detecting an increase or a decrease in the patient's BG value is composed of a single pipeline, consisting of a single step that uses a BASIC TREND algorithm. The parameter values set for detecting an increase are the following:

granularity=MINUTES
minLen= 35
label= Increasing
minSlope= 1
maxSlope= 200
gap=30

The parameter values set for detecting a decrease are the following:

granularity=MINUTES
minLen= 35
label= Increasing
minSlope= -200
maxSlope= 1
gap=30

Bradycardia and Tachycardia

The workflow for detecting bradycardia or tachycardia ha the same structure of the workflow for detecting hypoglycemia and hyperglycemia (Figure 34). The corresponding parameter values are listed in Table 40 for bradycardia and in Table 41 for tachycardia. Also in this case, the threshold for defining bradycardic and tachycardic episodes ($\mathrm{TH}_{\mathrm{BRADY}}$ and $\mathrm{TH}_{\mathrm{TACHY}}$ in the tables) are patient-specific and defined by the diabetologists.

Table 40: Parameters of the JTSA workflow for detecting bradycardia

STEP	ALGORITHM	PARAMETERS

Step 1	Qualitative	th= TH_{BRADY}, TH_{TACHY}
		label=brady, normal, tachy

Step 2	Aggregation HighLevel	gap=10
		minLen=5
		granularity=MINUTES
		label=brady
		levels=brady

Table 41: Parameters of the JTSA workflow for detecting tachycardia

STEP	ALGORITHM	PARAMETERS

Step 1	Qualitative	th= TH_{BRADY}, TH_{TACHY}
		label=brady, normal, tachy

Step 2	Aggregation HighLevel	gap=10
		minLen=5
		granularity=MINUTES
		label=hypo
		levels=hypo

Dawn effect

The dawn effect is defined as an episode of hyperglycemia at wakeup, which follows a night where BG values have remained in the normal range. The hyperglycemia episode is constrained to start maximum 45 minutes after the end of the normal BG period. The workflow for detecting the dawn effect was discussed in detail in Section 2.4.4.

Rebound effect

The rebound effect is defined as an episode of hyperglycemia that occurs after an episode of hypoglycemia. The episode of hyperglycemia should not start more than 45 minutes after the end of the hypoglycemic episode. The JTSA workflow for its detection is fed with the time series of BG measurements and is composed of three blocks, including two pipelines and one complex block (Figure 35). The first block is the pipeline that detects the time intervals in which hypoglycemia occurs. The second block is the pipeline that detects the time intervals in which hyperglycemia occurs. The third block is a complex block that detects the complex pattern, i.e., an episode of hypoglycemia followed by an episode of hyperglycemia. The parameters of the algorithms used in each block are listed in Table 42.

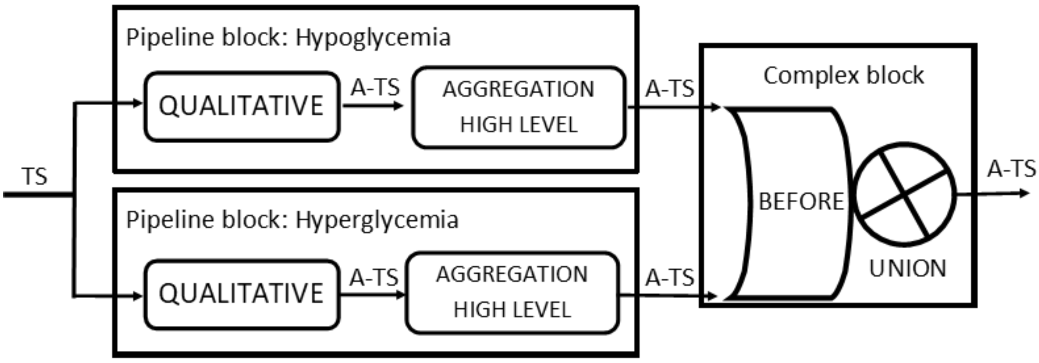


Figure 35: JTSA workflow for detecting the rebound effect

Table 42: Parameters of the JTSA algorithm for detecting the rebound effect

BLOCK	STEP	ALGORITHM	PARAMETERS

Block 1	Step 1	Qualitative	th= TH_{HYPO}, TH_{HYPER}
			label=hypo, normal, hyper

	Step 2	Aggregation High Level	granularity=MINUTES
			gap= 60
			minLen= 13
			label= hypo
			levels= hypo

Block 2	Step 1	Qualitative	th= TH_{HYPO}, TH_{HYPER}
			label=hypo, normal, hyper

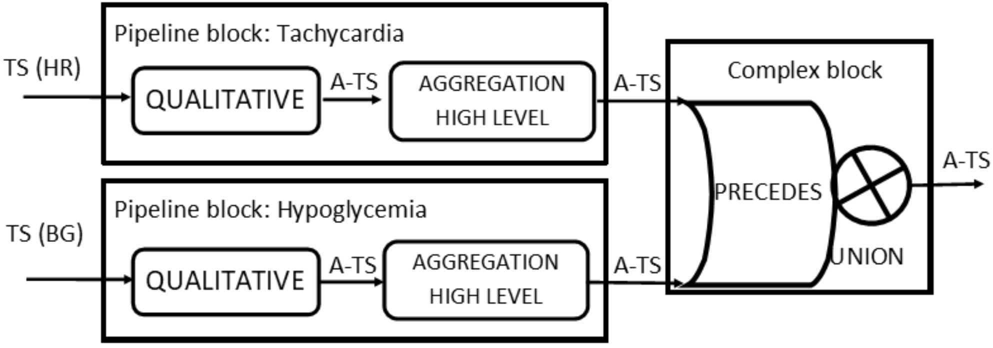
	Step 2	Aggregation High Level	granularity=MINUTES
			gap= 60
			minLen= 13
			label= hyper
			levels= hyper

Block 3	-	Operator:
		BEFORE
		Combiner:
		UNION
			granularity=MINUTES
			ls= 540
			rs= 240
			gap= 45

Tachycardia precedes hypoglycemia

This pattern consists of an episode of tachycardia immediately followed by an episode of hypoglycemia. The workflow is composed of three blocks, including two pipelines and one complex block (Figure 36). The first block is the pipeline that detects the time intervals in which tachycardia occurs. The second block is the pipeline that detects the time intervals in which hypoglycemia occurs. The third block is a complex block that detects the complex pattern, i.e., an episode of tachycardia followed by an episode of hypoglycemia. Since we are interested in hypoglycemic episodes which immediately follow tachycardic episodes, we use the PRECEDES temporal operator, instead of BEFORE. The parameters of the algorithms used in each block are listed in ***Table 43***.

Figure 36: Representation of the JTSA workflow for detecting tachycardic episodes immediately



followed by hypoglycemic episodes

Table 43: Parameters of the JTSA algorithm for detecting tachycardic episodes immediately followed by hypoglycemic episodes

BLOCK	STEP	ALGORITHM	PARAMETERS

Block 1	Step 1	Qualitative	th= TH$_{BRADY}$, TH$_{TACHY}$
			label=brady, normal, tachy

	Step 2	Aggregation High Level	gap=10
			minLen=5
			granularity=MINUTES
			label=tachy
			levels=tachy

Block 2	Step 1	Qualitative	th= TH$_{HYPO}$, TH$_{HYPER}$
			label=hypo, normal, hyper

	Step 2	Aggregation High Level	granularity=MINUTES
			gap= 60
			minLen= 13
			label= hypo
			levels= hypo

Block 3	-	Operator: PRECEDES
		Combiner: UNION
			ls= 40
			rs= 90
			gap= 0

Hypoglycemia precedes bradycardia

This pattern consists of an episode of hypoglycemia immediately followed by an episode of bradycardia. The workflow is composed of three blocks, including two pipelines and one complex block (Figure 37). The first block is the pipeline that detects the time intervals in which hypoglycemia occurs. The second block is the pipeline that detects the time intervals in which bradycardia occurs. The third block is a complex block that detects the complex pattern, i.e., an episode of hypoglycemia followed by an episode of bradycardia. The parameters of the algorithms used in each block are listed in Table 44.

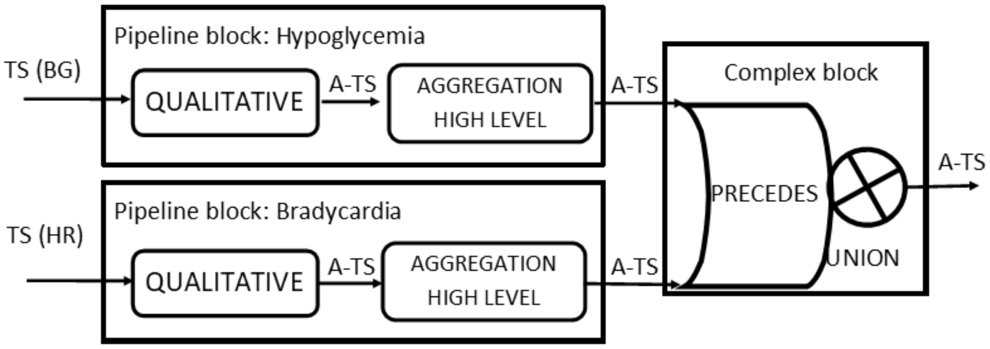


Figure 37: Representation of the JTSA workflow for detecting hypoglycemic episodes immediately followed by bradycardic episodes

Table 44: Parameters of the JTSA algorithm for detecting hypoglycemic episodes immediately followed by bradycardic episodes

BLOCK	**STEP**	**ALGORITHM**	**PARAMETERS**

Block 1	Step 1	Qualitative	th= TH$_{HYPO}$, TH$_{HYPER}$
			label=hypo, normal, hyper

	Step 2	Aggregation High Level	granularity=MINUTES
			gap= 60
			minLen= 13
			label= hypo
			levels= hypo

Block 2	Step 1	Qualitative	th= TH$_{BRADY}$, TH$_{TACHY}$
			label=brady, normal, tachy

	Step 2	Aggregation High Level	gap=10
			minLen=5
			granularity=MINUTES
			label=brady
			levels=brady

Block 3	-	Operator: PRECEDES
		Combiner: UNION
			ls= 40
			rs= 90
			gap= 0

List of publications

JOURNAL PAPERS

E. Salvi, P. Bosoni, V. Tibollo, L. Kruijver, V. Calcaterra, L. Sacchi, R. Bellazzi, and C. Larizza, *Patient-generated health data integration and advanced analytics for diabetes management: the AID-GM platform,* Sensors (Basel). 2019 Dec 24;20(1). doi: 10.3390/s20010128.

E. Salvi, E. Parimbelli, S. Quaglini, and L. Sacchi, *Eliciting and exploiting utility coefficients in an integrated environment for shared decision making*, Methods of Information in Medicine. 2019 Jun;58(1):24-30. doi: 10.1055/s-0039-1692416. Epub 2019 Jul 5.

INDEXED CONGRESS PAPERS

E. Salvi, S. Panzarasa, R. Bagarotti, M. Picardi, R. Boninsegna, I. Sterpi, M. Corbo, G. Lanzola, S. Quaglini, and L. Sacchi, *NONCADO: a system to prevent falls by encouraging healthy habits in elderly people*, Artificial Intelligence in MEdicine (AIME) 2019, June 2019.

E.Salvi, E.Parimbelli, L.Sacchi, S. Quaglini, E. Maggi, L. Duchoud, G.L.Armas, J. de Almeida, and C. Simon, *Towards the economic evaluation of two minimally invasive surgical techniques for head&neck cancer: a customizable model for different populations,* Artificial Intelligence in MEdicine (AIME) 2019, June 2019.

E.Salvi, G.Lanzola, S.Quaglini, S. Panzarasa, R. Bagarotti, L. Rodi, C. Larizza, and L. Sacchi, *Permutation entropy applied to Fitbit data: long-term sleep analysis on one healthy subject***,** pHealth2019, June 2019.

E. Salvi, L. Sacchi, A. Madè, V. Calcaterra, R. Bellazzi, C. Larizza, *AID-GM: an advanced system supporting continuous monitoring of T1DM patients,* Informatics for Health 2018, April 2018.

V. Calcaterra, L. Sacchi, **E. Salvi**, D. Larizza, A. Made, L.M. Schiano, C.Montalbano, C. Regalbuto, R. Bellazzi, and C. Larizza, *AID-GM system (Advanced Intelligent Distant – Glucose Monitoring) to monitor health status and metabolic control of young people with type 1 Diabetes*, 57th Annual ESPE, 2018

R. Bagarotti, E.M. Zini, **E. Salvi,** L. Sacchi, S. Quaglini, and G. Lanzola, *An algorithm for estimating gait parameter through a commercial sensorized carpet*, IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI), 2018.

E. Salvi, E. Parimbelli, G. Emalieu, S. Quaglini, and L. Sacchi, *A platform for targeting cost-utility analyses to specific populations,* Proceedings Artificial Intelligence in MEdicine (AIME) 2017, June 2017.

E. Salvi, E. Parimbelli, A. Basadonne, N. Viani, A. Cavallini, G. Micieli, S. Quaglini and Lucia Sacchi, *Exploring IBM Watson to extract meaningful information from the list of references of a clinical practice guideline* Proceedings Artificial Intelligence in MEdicine (AIME) 2017, June 2017.

NON INDEXED CONGRESS PAPERS

E. Salvi, L. Sacchi, F. Godtliebsen, T. D. Henriksen, K. Y. Yigzaw, M. A. Hailemichael, J. S. Hurley, S. Quaglini, J. G. Bellika
Generation of personalized decision trees from electronic health records, MedInfo Lyon 2019, August 2019.

E. Salvi, F. Fanfulla, N. D'artavilla Lupo, C. Boveri, G. Lanzola, S. Quaglini and L. Sacchi,
Is the Fitbit activity tracker a reliable device for the home monitoring of sleep?, VI Congresso Gruppo Nazionale di Bioingegneria (GNB), June 2018.

E. Salvi, A. Caronni, S. Quaglini, G. Lanzola, M. Corbo, and L. Sacchi, *Fall risk models: a literature review and an estimate of model score concordance,* VI Congresso Gruppo Nazionale di Bioingegneria (GNB), June 2018.

E. Salvi, A. Caronni, S. Quaglini, M. Corbo, and L. Sacchi, *Fall risk models: a literature review and an estimate of model score concordance,* Informatics for Health 2018, April 2018.

E. Salvi, E. Parimbelli, S. Quaglini, and L. Sacchi, *Supporting utility coefficient elicitation in a shared-decision making context,* Informatics for Health 2017, April 2017.

E. Salvi, E. Parimbelli, S. Quaglini, and L. Sacchi*, Supporting utility coefficient elicitation in a shared-decision making context,* V Congresso Gruppo Nazionale di Bioingegneria (GNB), June 2016.

BOOK CHAPTERS

L. Sacchi, **E. Salvi,** E. Parimbelli, S. Quaglini, *Medical shared decision making***,** XXXVI Scuola Annuale del Gruppo Nazionale di Bioingegneria

(2017), E-Health. Medicina digitale, by S. Quaglini, M. Cesarelli, M. Giacomini

Acknowledgments

Several people contributed to reaching the results described in this thesis.
First of all, I would like to thank my supervisor, Professor Lucia Sacchi, for her precious support, guidance, and patience in the past three years. Working with her has always been a source of motivation for me and I am very grateful for every moment she dedicated to my training.
I would like to thank Professor Silvana Quaglini for being an additional supervisor, albeit unofficially, and for involving me in several fruitful collaborations over these years.
I would like to thank Professor Cristiana Larizza for her precious suggestions for better coding, and for being my main teammate in developing AID-GM. It was a pleasure to code with her, debugging included.
I would like to thank all the team members from both the research projects I participated in, AID-GM and NONCADO, for sharing the efforts to reach our goals.
I would like to thank all my co-workers in Pavia, who made our office a friendly environment, in which I gladly spent these latest years.
I would like to thank the *Collegio Nuovo* for hosting me for the past two years, providing me with a stimulating and reliable living environment.
I would like to thank the students I co-supervised during my PhD, because working with them has always taught me a lot.
I would also like to thank the reviewers of this thesis for taking the time to read it, and for their constructive comments.

I am also grateful for all the support I received from my family and friends along the PhD journey.
Thanks to my family for being a constant certainty for me during these years, despite not being supporters of the research world.
Thanks to my flat mates in Pavia (Noemi, Giorgia, Alberto, and Paolo) and almost-flat mate in Tromsø (Alexandra), for being like a second family to me, turning the apartment I lived/live in into a real home.
Thanks to the bio-eng group (Adelina, Elisabetta, Lea, Palma, Paola, and loved ones) and Simona, for supporting me since our university years. Even when we could not meet often, I was always sure we would be there for each other.
Thanks to Anna, for being a great listener, also remotely, and for all the post-work memories, including my favorite, i.e., unfocused grocery shopping on Friday evenings.
Thanks to Pamela, for always asking about my PhD journey with curiosity and for our regenerating lunch appointments. Thanks to Marta and Eleonora,

for our spontaneous pizza breaks, never long enough to cover all the necessary updates.

Thanks to my “nephews” (Sophie, Adrien, and Roman), since spending time with them was one of the most powerful anti-stress solutions during these years.

Finally, I would like to thank two friends I could share my PhD journey with. It was crucial to have someone who could perfectly understand how I felt about the PhD experience, even before I could complete my sentences. Thanks to Natalia, for setting up an example for me, for being a great listener and advisor since I was a master student, and for sharing countless (research) experiences, including the congresses. Thanks to Ingrid for providing constant support since my period abroad, for her ability to convey calm and positivity, and for her invaluable company, especially during the writing phase.

I could not have been luckier in terms of traveling companions.