Università degli Studi di Padova

Department of Information Engineering

Master Thesis in Telecommunications Engineering

Unsupervised anomaly detection techniques for smart assisted living using environmental sensors

Supervisor
Michele Rossi
Università di Padova

Master Candidate
Matteo Ciprian

1 April 2019

Academic Year 2018/2019

Dedicated to my family and to everyone who gets close to me during these years

Abstract

The continue increment of life expectancy is given rise to an increasing aging population in almost all the countries. In order to guarantee senior people to live independently in their own community, there is a need of Smart Assisted Living Systems able to monitor the Activities of Daily Living of elders, supporting them in every day life. Although such systems can be deployed to provide a wide range of services, one of the most interesting aspect concerns "behavioural monitoring and understanding" with the possibility of detecting upcoming critical situations such as a fall, loss of consciousness or other health problems.

In this thesis we present an analysis of sensor data collected in a real Smart Home scenario from 17 households of elderly people who live alone, monitored through environmental sensors. Such sensing devices are motion sensors (PIR) and pressure sensors, able to detect "Occupancy Activities" of the patient as the permanence in a given room or the usage of a specific object for a specific interval of time. In specific, we elaborate a software to detect anomalies on single activities (events) in respect to a specific interface of alarms: "Anomalous Duration" (an activity which lasts for too long or too short) or "Anomalous Occurrence" (an activity repeated too many times) and Unusual Activity. The anomaly detection system here presented is adaptive to the patient and exploits a data-driven approach that works in a completely unsupervised setting since the data-set is completely unlabelled. To cope with this problem, we elaborate a first filtering phase based on GAS Growing Neural Network Clustering technique which selects the events respectful of an "operative definition of recurrence". Only the recurrent events are used to build a probabilistic model utilized to compute an Anomaly Score function and perform classification. Moreover, a set of tests have been defined to assess the performances of the system in terms of:

- Reliability: 1) ability of our system to detect as anomalies the anomalous events artificially generated and injected in the data-set (True positive rate) 2) ability of detecting as normal not anomalous events previously selected (True Negative Rate).
- Adaptability: capacity of the system to adapt with respect to changes in patient's habit.

As general trend, the results achieved show good performances in respect to both the two metrics despite future works have to be addressed to properly tune some project parameters.

Sommario

Il continuo aumento della popolazione anziana nella società odierna sta comportando una sempre più grande richiesta di Servizi di Assistenza Remota (Smart Assisted Living Services) integrati in ambiente Smart Home per il monitoraggio delle attività giornaliere (ADL) dell'assistito. Tali sistemi danno la possibilità di individuare anomalie nel comportamento delle persone monitorate permettendo il rilevamento precoce di situazioni pericolose come perdite di conoscenza, malori o presenza di intrusi.

In questa tesi un software per la detection di anomalie è stato sviluppato come supporto ad una piattaforma digitale di Telemedicina chiamata Ticuro Reply.
L'algoritmo è stato sviluppato sulla base di un dataset raccolto in uno scenario reale Smart Home, in cui, 17 anziani che vivono da soli sono stati monitorati attraverso sensori ambientali. Tali dispostivi sono nello specifico sensori di movimento a infrarossi (PIR) e sensori di pressione in grado di rilevare attività come la presenza di una persona in una data stanza o l'utilizzo di un oggetto per un certo intervallo di tempo. In particolare, il software rileva anomalie su singole attività (eventi) rispetto a una specifica interfaccia di allarmi: "Durata anomala" (un'attività che dura troppo a lungo o troppo poco) o "Occorrenza anomala" (un'attività ripetuta troppe volte in un certo intervallo di tempo), "Attività insolita". Il sistema è stato progettato per essere adattivo rispetto al paziente e utilizza un approccio data-driven. Inoltre, l'algoritmo è stato concepito per funzionare in un ambiente completamente non-supervisionato, date le specifiche del dataset di partenza. Per far fronte a questo problema iniziale, è stata predisposta una prima fase di filtering basata su una tecnica di clustering chiamata GAS Growing Neural Network in grado di filtrare solamente gli eventi che sottendono ad una precisa definizione di ricorrenza. Solo questi eventi ricorrenti vengono utilizzati nella una fase successiva (fase di classificazione) per costruire un modello probabilistico grazie al quale è possibile calcolare una funzione di "Anomaly Score"; sulla base di tale funzione un dato evento viene classificato come anomalo/ non-anomalo.
Oltre alla definizione di una procedura per Anomaly Detection, in questa tesi anche una serie di test sono stati definiti per valutare le prestazioni del sistema in termini di:

- Affidabilità: 1) capacità del nostro sistema di rilevare come anomalie, degli eventi anomali generati artificialmente e inseriti nel dataset (True positive rate); 2) rilevare come non anomali, eventi normali precedentemente selezionati (True Negative Rate).
- Adattabilità: capacità del sistema di adattarsi ai cambiamenti nella ruotine del paziente.

Come considerazione generale, i risultati raggiunti mostrano buone prestazioni rispetto ad entrambe le due metriche, nonostante altri lavori futuri debbano essere effettuati per una selezione più accurata dei diversi parameteri di progetto.

Contents

Listing of figures

Listing of tables

Listing of acronyms

AAL .... Ambient Assisted Living

SHAAL ...... Smart Home Ambient Assisted Living

ADL .... Activity of Daily Living

GNG .... Gas Growing Neural Network

SOM .... Self Organizing Map

1

Introduction

1.1 Background and Motivations

Smart Assisted Living or Ambient Assisted Living (AAL) can be summarized as an ensemble of concepts, products and services which combine new technologies to allow people to stay active longer, remain socially connected and live independently in their community. Having such target, AAL systems have been widely exploited in *Elderly Care* context as an assistance tool for elderly people. Research in the AAL community covers a wide range of topics but at the present time the most important ones are focused on *health parameters monitoring* for health status control, *human activity recognition* for recognizing activities of daily living (ADLs) and *behavioural understanding* for apprehending the behavioural habits of a specific person. Although such lines of research are different, they all boil down to the design of methods for early detection of critical and anomaly situations within a specific environment. In this sense research on Anomaly Detection can be considered a crucial point at realizing the goals previously cited. In recent years, work in Smart Assisted Living field has intensified, taking advantage of a flourishing technological progress in several different branches of ICT. At first, the developments in sensor technology, reduced sizes and costs, have contributed to the diffusion of a wide range of sensing devices which have enabled more sophisticated and accurate measurements. These are for example wearable sensors for movement analysis as IMUs (Inertial Measurements Units) or pulse-oximeter for oxygen saturation (SpO2) or environmental sensors for motion detection. A second fundamental reason concerns the progress in Communication Networks. The definition of new protocols and architectures over the so-called "Internet of Things" (IoT) paradigm allows a continuous and reliable exchange of information between such devices. Third but not least, the spreading of Artificial Intelligence and Machine Learning has provided useful frameworks to process the big amount of signals gathered, making possible a data-driven analysis with a strong impact on personalized medicine and diagnosis as well as on anomaly detection in health and behavioural status.

Beyond a positive technological background, research in AAL has gained momentum due to the fact it provides innovative approaches to face the increasing of the socials issues brought by the aging of population. According to World Health Organization, by 2050, the worlds population aged 60 years and older is expected to total 2 billion. This trend can lead to several sociological and economical challenges as an increasing costs for health-care systems. In this sense AAL technologies serve as valid framework of solutions to help senior people in continuing to lead independent lives and play an active role in the society, whether that be in the house, work or in their community.

In this context, home monitoring of old people can be seen as a first fundamental step to achieve the aforementioned goals. This is intrinsically connected with the concept of "Smart Home", used to define an integrated system of communicating sensors and actuators which enables remote monitoring and management of the appliances. In a Smart Home the administration and services in the house are centralized, supporting in such way the daily activities of elderly people with the possibility of provisioning remote assistance including care succor within the home environment. In some papers the term SHAAL [3], Smart Home Ambient Assisted Living, is used to define specifically AAL technologies applied to smart home framework which can be considered the main area of interest in this community.

As proof of a more and more growing interest about this thematic, it is possible to see that in the literature many studies have been conducted so far and some commercial solutions are starting to be available on the market. Some early implementations of smart home are for example research projects as Georgia Tech Aware Home [4], and MIT intelligent room [1]. Beyond purely academic studies, some AAL in Smart Home Service have been implemented and offered by companies operating in the Telemedicine and HealthCare field as Bell Canada [5], AT&T [6] and British Telecom (BT) [7]. In many of these systems, cameras and motion sensors are utilized to track the activities of the elders residing at home and, in the case of an unexpected event, notifications are sent to the family members and the caregivers via e-mails, text messages and voice messages. In others, wearable GPS-enabled device that can be easily attached to a keychain, are used to trace the activity of the person in and out of the home.

1.2 Objectives of the thesis and contributions

The work presented in this thesis deals with *behavioral monitoring and understanding* on a real commercial AAL system developed by Healthy Reply, a company of Reply Group based on Milan. The core of this project is "Ticuro Reply" a telemedicine platform created by Healthy which integrates a wide number of sensors to allow medical and behavioral monitoring in IoT and IoMT environment. Ticuro can be used both in clinical and non-clinical context including a quite large variety of services ranging from Vital Signs Monitoring, Remote Consultation with clinicians and creation of personalized plans [8]. An important project of Healthy Reply is precisely called "Elderly Care" which involves Ticuro platform for developing an Assisted Living system to monitor a group of old people who live alone. In this case, environmental sensors as Motion Sensors (PIR) based on infrared light and pressure sensors have been installed on the houses of these people. These sensors detect the presence of a person in a specific room and at a specific time noticing the open/lock of windows and doors as well (occupancy activities). The devices are connected through an IoT hub to a server processing unit which elaborates the events and sends electronic notifications to a secondary service provider once an anomaly is detected. The processing unit, also referred to Reasoner, can detect anomalies in the behaviour of the person following a pre-defined interface of Alarms as for example Excessive presence of the patient inside a room or the Repetition of an activity for too many times in a specific period referred to unusual behaviours that might be triggered by a dangerous event such as a fall, loss of consciousness, health problems or intruder. So far, such anomalies have been identified using a "deterministic" reasoner based on static thresholds chosen a priory by an operator. In this context, once a certain threshold is exceeded, for example the patient stays in bathroom for more than 4 hours, an alarm is raised. This can be considered a very naive and inefficient approach, unable to adapt to the habits of the patient and therefore less robust to false alarms (false positive).

In this work a new software module integrated with Ticuro's reasoner has been realized to cope with this inconvenient. Specifically, this software performs anomalies detection on single events-activities using a data-driven approach based on advanced tools and frameworks available in the literature.

The anomaly detection problem here presented is not new in the literature although in some works, up to now, the solutions proposed are thought to work in very specific "ad-hoc scenarios", difficult to be reproduced on a real setting. Such discrepancy can be caused by a lack of standardization about the sensor type to use and the way the data have to be collected [3], a limited availability of open-source and reliable data-sets [9] and an intrinsic difficulty in properly defining the anomalies in the scenario [9]. In any case, as better described in section 2.4, two main paths can be identified [9]: a first path concerns the detection of event-related anomalies such as anomalies on the single activity [10], [11] while a second path is about spotting anomalies in pattern (or sequence) of activities [9],[12], [13], [14]. With respect to both approaches, the architecture and the frameworks utilized depends mainly on type of dataset available at beginning. In [13], Support Vector Data Description (SVDD) method has been used to classify pattern of activities using a labelled dataset where abnormal and normal pattern were previously identified by an expert. Despite the very good performance, such methods are far from being applicable in a real scenario where the datasets are usually unlabeled. [9] tries to overcome this limits proposing a method based on HTM, Hierarchical Temporal Memory to find anomalies also in unsupervised setting. An event-related approach is instead investigated in [10] where an algorithm based on SOM self-organizing map clustering technique has been used to perform anomaly detection in respect to Unusual Activity, Unusually short activity, Unusually long activity. [10] works with a relatively small dataset, ad-hoc created in ([15]), under the assumption that the training data are "Anomaly Free" (semi-supervised setting).

The dataset considered in this study has been collected through Ticuro in a *real* context considering around two years of recording on 17 different households of patients. Since no manual annotation is present in the dataset, the only labels directly available are represented by the alarms launched by the previous version of the reasoner. From the considerations made before, such labels are not trustworthy and for this reason the considered problem is completely unsupervised. In this work we have followed an event-based approach as in [10]. The novelty proposed in this work consist in defining a way to move the problem from a completely unsupervised to semi-supervised setting exploiting a filtering strategy where an object event, properly represented as a feature vector, is given in input to an event-filter based on Gas Growing Neural Networks which selects only the events respectful of an operative definition of *recurrence*. At this point only such recurrent events are processed to build a probabilistic model used to perform anomaly detection based on the computation of an Anomaly Score function (Statistical Anomaly Detection procedure). Being the algorithm here discussed thought to be used in a real Smart Assisted living service, a particular attention has been taken in respect to the applicability of the procedure in a real scenario. The main contributions of this work can be thus summarized as follows:

- The development of a new standalone software for behavioural monitoring in Smart Assisted Living context using a real digital platform named Ticuro Reply.

- The definition of a new protocol/methodology for Anomaly Detection to find out anomalous events in an non-supervised setting.

- The definition of a new protocol/methodology for events filtering.
- The definition of a set of tests to assess the performances of the system proposed.

This thesis is organized as follows: in chapter 2 the main concepts of Smart Assisted Living and Anomaly Detection are discussed with a final focus on the past works that had tried to combine such disciplines; in chapter 3 the case study considered in this work is completely presented providing details about the dataset and the type of anomalies we are trying to detect; in chapter 4 the anomaly detection software is presented including the description of each single component as well as with the analysis of the tests; finally in 5 and 6 the results and conclusion are disclosed.

2

Literature Overview

Internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines with the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. In an healthcare context the usual "Internet of Things" paradigm is more often referred with the term "Internet of Medical Things" to differentiate it from the other applications and scopes. Smart home with integrated e-health and assisted living technology, is an example of an IoT and IoMT platform where a variety of sensing and acting devices are used for healthcare purposes. In a smart home, sensors and actuators are connected through a Personal Area Network (PAN) or Wireless Body Area Network (WBAN) where physiological signals are continuously measured in real-time manner, sampled and sent to a central BSN node which performs limited data processing and functions as the gateway to external network. In the following two sections a review of the main topics related to the project are presented. Although the work of this thesis can be confided to the scope of "Data Science", in section 2.1, also a brief survey on sensors and AAL technologies is given for completeness. In 2.2, we provide an overview about the experimental projects related to Smart Home for AAL, while in 2.3 the theme of "Anomaly Detection" is discussed underling the most diffused approaches and the main problematics connected. Finally, in 2.4, it is presented a list of works existing in literature which coped with a similar problem.

2.1 Sensors technology for Smart Assisted living

Sensor type and effectiveness largely depend on the parameter to be recognized. As stated in [16], in the scope of AAL two main categories of sensors can be distinguished: wearable sensors and non-wearable sensors. Wearable sensors are usually attached to a person directly (e.g., bracelet sensors or cardio sensors) or to her/his

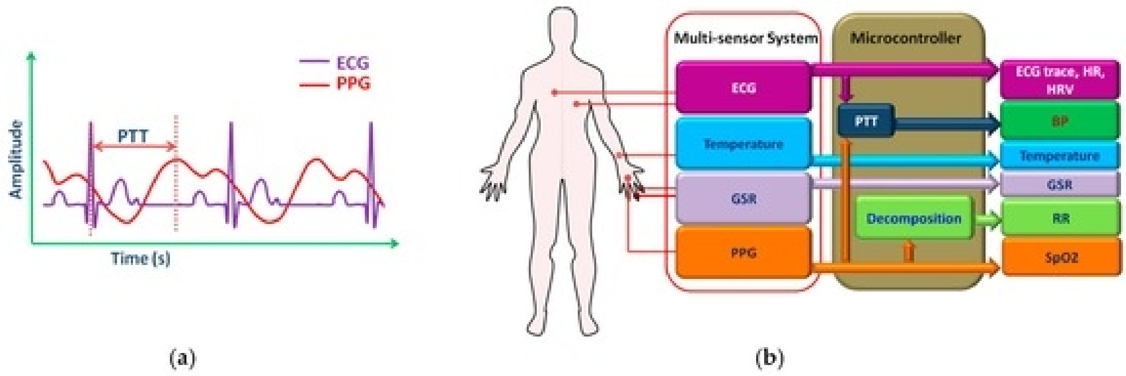


Figure 2.1: Health parameter monitoring scheme

clothes (e.g., an accelerometer, step counter) and are used to measure vital parameters as well as to assess motion activity characteristics (Low-level sensory data analysis 2.4); non-wearable sensors are deployed in stationary locations of a house or a room and can be exploited often to detect a person and his/her activities.

Wearable sensors are usually deployed for two main tasks: vital signs monitoring and Movement Activity Analysis. In the first task, it is possible to include a large variety of sensors that measure some specific body parameters as blood glucose, blood pressure, heart rate and activity (ECG) [17], blood saturation, CO2 gas, or brain activity. Despite in the past such measurements were only possible in a clinical environment, nowadays many cost-effective sensing devices are available on the market [18], figure 2.1.

In Movement Activity Analysis, it is possible to include a set of wearable sensors which provide quantitative assessment of the movement and can be deployed for activity recognition or posture analysis. These sensing devices are particularly exploited in AAL technologies for ADL (Activity of Daily Living) classification [19] and to differentiate different types of motion [20] (e.g., running, walking, sitting, scrubbing, etc.) as well as useful tool for fall detection [21]. In this domain Inertial Movement Sensors (IMU) are the most diffused ones. These units are usually composed by an accelerometer integrated with one or a set of gyroscopes and/or a magnetometer. Inertial units are usually embedded in many devices of daily using such as wristwatch or smartphones. Specifically for ADL classification, another technology named RFID, "Radio-Frequency Identification Devices", is sometime utilised. Such tools use radio waves to identify objects or persons tags and are often used to detect the interaction of a person with an object. In [22], RFID tags were deployed on various kitchen utensils such as bowls, cutlery, dishes, and jars to detect food preparation, eating, and drinking.

Despite this kind of sensors had a great diffusion over the last years, such technology can be considered largely impacting on the final user. Moreover, the quality of the data gathered by wearable sensors principally depends on the way these units are applied on the body. Just a small error in the application of the sensing devices can have a huge influence on the signal, compromising the reliability of the measurements. In a non clinical context and without the supervision of an expert, this requirement is not always respected with a considerable impact on the diffusion of wearable sensors for AAL.

From this perspective, nonwearable sensors can be a valid solutions because they are less intrusive and do not require any interaction from the users side [16]. On the other hand this kind of sensors are able to gather less informative data with a lower accuracy. They can be deployed for measuring the operational status of objects, measure water flow, room temperature, presence of a person or door/cupboard openings/closings [16] (High-level sensory data analysis 2.4). Often times, such kind of sensors are commonly referred with the term *environmental sensors*. In Smart Home Assisted Living domain, the most used type of non-wearable sensors are the following. *Infrared sensors* (IR) which uses infrared light to discover human presence in a room, detect motion in a specific area, or to locate a human within a house [23]; *ultrasonic sensors* which exploit ultrasonic vibrations for movement detection and localization by measuring distances to objects; photoelectric sensors, which detect a light source and output a signal when the light intensity is greater or less than the predefined threshold value [24]. Other solutions can come from vibration sensors [25] usually deployed to detect a person falling or interaction with various objects, or measuring water flows. Pressure sensors can be considered a variant of the last ones, and are utilized to detect the presence of a person, steps, and fall, aperture/disclosure of doors and windows [26].

In a broad sense, it is possible to include in "non-wearable sensors" category also Video Cameras and Audio sensors. In video-based approaches, a camera is installed in a specific place of a house to detect person movements and to classify ADL through Computer Vision tools [27]. Differently, in [28], Audio Sensors such microphones and audio transducers are used to perform the same task discriminating different types of sounds and connecting them to the related ADL. Although this two last solutions considered to be the most reliable from a technological point of view, they are rarely implemented in a real context because considered too much obtrusive on patient's privacy.

The sensors integrated in the Ticuro Platform for this project are: *PIR sensors and pressure sensors*. The former, are used to detect movement and presence of a person in a specific location and consequently checking the presence of a person in a specific room at a specific time; the latter are used to capture the opening and lock of the fridge, section 3.

All sensors and actuators in the smart home are connected with the central communication and decision making platform though a communication network. All physiological and environmental signals measured by the sensors are transmitted to the central computing node over a wireless and/or wired communication medium.

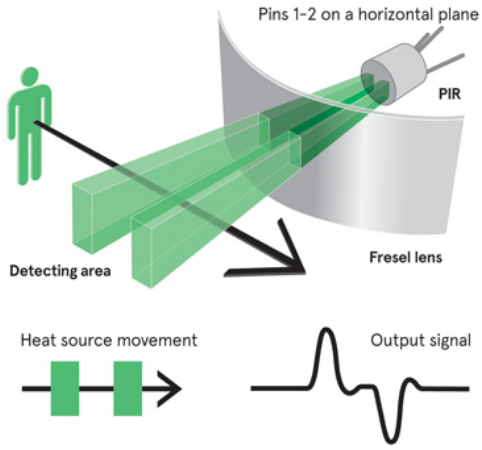


Figure 2.2: PIR sensors functioning

Figure 2.3: Real PIR Sensors

Although wired connection is a feasible solution for fixed-position based environmental sensors, it is not suitable for wearable and long-term monitoring systems. The wearable medical sensors can be connected in a Body Sensor Network, where the central BSN node (the hub) is connected with all environmental sensors and actuators through the WSN to form a Local Area Network (LAN) or Personal Area Network (PAN) and to provide data communication inside the smart home. As underlined in [29] communication technologies used in these systems may vary from supplier to supplier in a quite fragmented IoT market. The widely used protocol are Low-Power Wireless Personal Area Networks (6LoWPAN) over Bluetooth, ZigBee IP, and 6LoWPAN over DECT, ULE, and Thread.

2.2 Smart Home Projects and Datasets

Smart Assisted Living services for monitoring elderly daily behaviour require the existence of good datasets that enable testing and validation of the methods proposed. However, the cost to build real smart homes and the collection of datasets for such scenarios is very high and sometimes impracticable for many projects; beyond a pure economical reason, other problems have to be taken into account as finding the optimal placement of the sensors [30], finding appropriate participants ensuring them a certain level of privacy [31]. For these reasons, there are few numbers of smart home projects being established worldwide. One of the first pioneer project in this field was MavHome, started in 2003 [15], which proposed a first Smart Home architecture and standard for behavioural monitoring through cameras and environmental sensors. Since 2003 to the the present, other projects have begun:

- Smart AHRI (Aware Home Research Initiative) [4] at Georgia Institute of Technology.
- CASAS (The Center for Advanced Studies in Adaptive Systems) at Wash-

ington State University [32], [33] where Cook et. al created a toolkit called smart home in a box where they integrate several types of environmental sensors that can fit in a single box. The toolkit has been installed in 32 homes to capture the participants interactions.

- TigerPlace project [34] at University of Missouri, USA, where passive sensor networks are deployed in 17 flats within an elder care establishment. The devices include motion sensors, proximity sensors, pressure sensors and others. The data collection took more than two years.
- PlaceLab at MIT [1], [35] is a 1000 sq.ft. smart flat developed in several rooms. The flats have many sensors distributed throughout each room, such as electrical current sensors, humidity sensors, light sensors, water flow sensors, etc. Volunteering participants can live in PlaceLab to generate a dataset of their interactions and behaviours. The project produced several datasets for different scenarios.
- HomeLab [36] is a smart home equipped with 34 cameras distributed over several rooms. HomeLab is promoted and has an observation room that allows the researcher to observe and monitor the conducted experiments. HomeLab aims to provide datasets to study human behaviour in smart environments and to investigate technology acceptance and usability.
- Smart home Lab at Iowa State University [37]

These smart home projects monitor occupancy daily living activities, enhancing comfort and deploying heterogeneous sensors for to better regulate daily activities. Obtrusive camera sensors are sometimes included as well. Only small part of the data generated in the experimental settings just cited are free available for research purposes; a quite good percentage of the datasets are proprietary based and so not available to the whole scientific community. To cope with this problematic, Smart home simulation tools have been proposed as an alternative solution. A wide explanation of such types of tools can be found in [38], where the authors distinguish two approaches: model based and interactive. In the first approach predefined models of activities are used to generate synthetic data [39], [40] while, in the interactive approach, the data are generated by an avatar that can be controlled by a researcher or by another human participant. The avatar moves and interacts with the virtual environment which is endowed of virtual sensors and/or actuators [41]. Although the last method can be considered more realistic it requires an effort from the user side reducing the amount of data that can be generated.

As underlined in [42], despite a certain amount of investigations and data collection have been conducted so far, most of the datates do not contain labeled data and usually anomalies are generated forcing a certain anomalous behavior in the

Figure 2.4: Placelab Smart Home Environment at MIT [1]

Figure 2.5: Smart Home Scenario for Smart assisted living

monitored person [9] or injecting artificially the events, [10] [32]. In both cases a certain degree of arbitrariness is present on what the authors of the study has defined as anomaly.

2.3 Anomaly detection techniques

"Anomaly detection" is a very challenging problem widely explored in the scientific literature as well as in the philosophical one. The anomaly detection problem can be shortly summarized as identifying single data point or patterns of them that do not lie in a normal region. Therefore, the problem can be approached by recognising this normal region and flagging as anomaly anything outside it.

As underlined in one of the most important survey about this subject [2], the Anomaly Detection problem is very challenging in practice due to the absence of a criteria to distinguish what can be considered normal and what abnormal. Moreover, the concept of normality can change from one domain to another and evolve in time also within the same domain. From a more practical point of view, a lack of representative data-sets with or without labels is usually another major problem and often it is very difficult to distinguish true anomalies from noisy data. To properly address this problem in [2] anomalies are categorised into three types:

- *Punctual anomalies*: Where anomalous data points are regarded so different from the rest of the data. In Figure 2.6, regions N1 and N2 are considered normal because most of the data points are in these two regions. On the other hand, O1, O2 and O3 are far from the normal regions and considered anomalies.

- *Contextual anomalies*: Where the context of the data points is anomalous and not the data points itself. As shown in 2.8, the data point t1 is identical

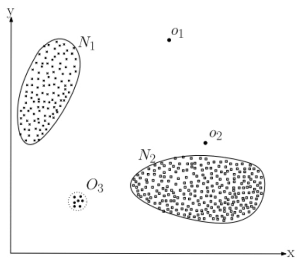


Figure 2.6: Point anomalies in a two dimensional space [2]

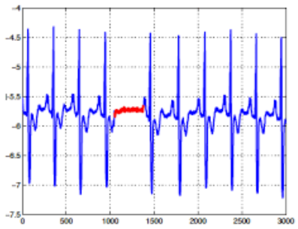


Figure 2.7: Collection anomaly in a human electrocardiogram [2]

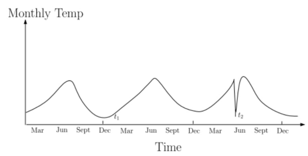


Figure 2.8: Contextual anomalies in a monthly temperature data [2]

to the data point t2 but the latter point is considered anomalous because it appears in an anomalous context.

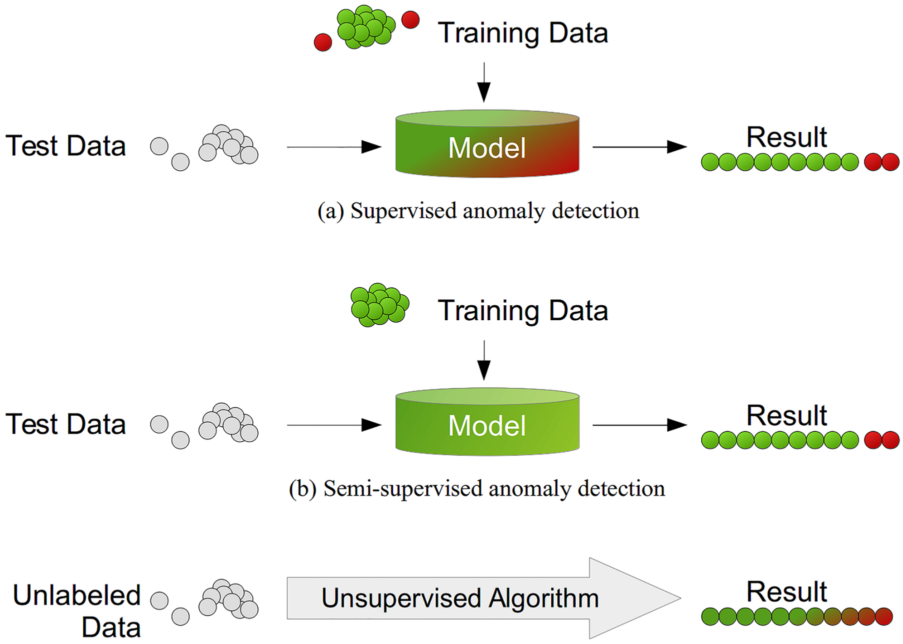
- *Collective anomalies*: Where a collection of data points is considered anomalous not because of the data point themselves but because of the collection of these data points together. In Figure 2.7, The data points in the electrocardiogram are considered anomalies because of their appearance as a collection in this data and not because of the data points themselves.

With this premises it is possible to define an *Anomaly Detection Technique* as a method which allows one to detect anomalies in a given data set. With regards to the availability of the data labels, Anomaly Detection Techniques can be divided into three categories:

- *Supervised*: The entire dataset (or a part of the it) is identified with labels which define if data points are normal or abnormal.
- *Semi-supervised*: The assumption here is that the available training data is all normal and the deviation from these normal data points is considered an anomaly.
- *Unsupervised*: The data are not labelled and no training data is needed. The techniques belonging to this category assume that the majority of the data points are normal and thus any isolated points are considered anomalies.

Beyond this first distinction Anomaly detection techniques can be differentiated in respect to the so-called *categorisation factor* used. A summary of the main approaches is reported in the following.

Classification These methods need labelled data points for the models to learn from. The main idea is to train a classifier on the normal data points and then evaluate the accuracy of the model on unseen testing data. These techniques can be further divided into two categories: one-class and multi-class anomaly detection techniques. The one-class models group all normal data points as one big class and



(a) Supervised anomaly detection

(b) Semi-supervised anomaly detection

(c) Unsupervised anomaly detection

Figure 2.9: Anomaly detection approaches in respect to the availability of training and testing data

learns the characteristics of these data points trying classify them as normal. The points residing outside this class are flagged as anomalies. To learn the normal region, several algorithms can be used although the most popular one is Support Vector Machine (SVM) [43]. The multi-class category is similar to one-class except that instead of learning one normal region, multiple regions can be learned, into several categories. The tools exploited in this case are Multi-Class Support Vector Machines [44], Artificial Neural Networks, Bayesian networks [45].

Nearest Neighbours This type of anomaly detection techniques are widely exploited in unsupervised setting and are based upon a proper definition of a *distance metric* among data points. According to this definition of distance, the data are organised in neighbourhoods to understand the structure of the dataset. The algorithms in this category can be further divided into two sub-categories: algorithms based on the Kth nearest neighbour and density-based algorithms.

A Kth nearest neighbour algorithm has been applied for example in [46] by Guttormsson to detect anomalies in the operation of turbine motors. In this case the Kth nearest neighbour distance has been used as an anomaly score for a collection of data points and a threshold can be set by a field expert to separate anomalous data points from normal ones.

Differently, density-based techniques measure the density of data points neighbourhoods. Any data point that resides in a low density neighbourhood is flagged as an anomalous data point. On this idea it is based the popular Local Outlier Factor (LOF) density based algorithm that takes into account the ratio of the average density in each neighbourhood or cluster. To calculate the density of a neighbourhood, a radius of a small hypersphere is defined for a neighbourhood with a data point at its centre. Then, the hypersphere volume is divided by the number of data points in the neighbourhood and the resulting number is the density score for this neighbourhood. The anomalous data points can be easily defined as the data points that reside outside of the neighbourhood [47]. From 2000 when LOF algorithm has been proposed, many extensions in the literature were introduced with the aim of improving the calculation time and reducing the computational complexity [48].

Nearest Neighbours techniques do not require the availability of target labels for every data point and work with any type of data assuming that an appropriate distance measure is defined. For these reasons such algorithms are suitable and can be used in an unsupervised and a semi-supervised fashion. The main disadvantages of these techniques lie in the fact that the normal data is assumed to have a cluster structure. Moreover they can be computationally expensive, with a complexity around $O(N^2)$ since the distance for each data point is calculated against all data points.

Clustering approach Clustering based anomaly techniques can be considered very similar to nearest neighbours techniques for many aspects. While nearest neighbours techniques perform the calculation between a data point and its local nearest neighbour, the clustering based techniques perform the calculation between each data point and the group or the cluster that it belongs to. As in the previous cases, also in this type of techniques some sub-groups can be defined. A first group of the clustering techniques presume that the normal data has a cluster and any data point outside of this cluster is flagged as an anomalous data point. Examples of these techniques are DBSCAN which was proposed in [49], ROCK [50] and SNN [51].

Another group of these techniques works under the assumption that normal data points are organised around the cluster centre or centroid. The data points that are not close to the centre are identified as anomaly data points. The general procedure of these techniques starts by using a clustering algorithm to group the data points. Then for every data point, the distance from the data point to the cluster centroid is defined as the anomaly score. Several algorithms were successfully used to achieve this goal such as Self-Organising Maps [52], Expectation Maximisation (EM), K-means [53], Gas Growing Neural Network [54].

A third family of these clustering methods suppose that normal data points reside in dense clusters and anomalous data points are concentrated into low density clusters. There are many algorithms in the literature that follow this assumption. Examples of these algorithms are Cluster-Based Local Outlier Factor (CBLOF) which is proposed in [55] in 2003. Similar to the other families, many extensions to these algorithms were proposed in the literature [56].

As with the Nearest Neighbor approach, Clustering techniques can work in an unsupervised in a semi-supervised fashion.

Statistical Statistical techniques try to fit a statistical model on the normal data points distribution and holds on the assumption that an anomaly is an observation which is suspected of being partially or wholly irrelevant because it is not generated by the stochastic model assumed.

The statistical approaches for anomaly detection problems can be divided into two categories: parametric and non-parametric techniques.

Parametric techniques assume the existence of a distribution and its parameters can be learned from the data points. The distribution parameter is referred to as Θ which is estimated from the training data points to compute the probability density function (pdf) $f(x, \Theta)$. At this step an anomaly score function is defined in respect to $f(x, \Theta)$. When the model is defined, simple thresholds can be applied to filter out the normal data from the anomalous data or differently a statistical test can be applied [57]. Another class of parametric techniques exploits a regression model to detect anomalies with frameworks used in time-series analysis [58]. These techniques fit a regression model based on the training data points and then the anomaly score is calculated as how far this testing data point is from the predicted ones.

Differently, non-parametric techniques do not assume the existence of a distribution of the data points. Rather, distribution of the data is derived from the dataset itself. These techniques can be further divided into histogram based and kernel function based techniques. Kernel based methods use kernel functions (better define later) to estimate the density of the dataset. Examples of these techniques are [59] and [60].

2.4 Related Works

Although most of the techniques and concepts related to Anomaly Detection have been formulated in the literature for more than 25 years, the applications of such frameworks in Smart Home Assisted Living has been explored only over the last 15 years.

As stated in one of the most recent survey about application of Anomaly Detection in Smart Homes [61], it is possible to distinguish among three main types of research areas in this domain: *health parameter monitoring*, for control of the health status, *environmental monitoring* intended as the monitoring of the ambient parameters as temperature, power consumption etc., *Behavioural Monitoring and*

Understanding intended as the analysis of behavioral patterns and daily activities, the one treated in this work. Always in [61] two kind of level of analysis are differentiated in respect to the type of data available: *Low-level sensory data analysis* in where row signals coming from the sensors are processed and *High-level sensory data analysis* where only a higher pre-processed data output is accessible from the analyzer side.

Low-level sensory data analysis is usually possible when signals in time series fashion are available at the input. The signals are processed through a specific pipeline such as pre-processing, segmentation and feature extraction. In *Behavioural Monitoring* domain, such types of analysis are usually performed for Activity Recognition (AR) or Discovery (AD) processing (for example) the inertial signals (3D acceleration, turn velocity..) coming from IMUs and ambient sensors [62]. Such topics have been widely explored so far in many studies exploiting tools as Hidden Markov Models (HMM) [62], SVM-Based Multi-modal Classification [63] and Artificial Neural Networks [64]. Even many projects and data-set developed over the Smart Home Environments cited in the previous section 2.2 are focused on this direction.

The eventual output of low-level data analysis is *occupancy activities* or labels, such as sleeping, eating, showering, entering or leaving home. These different activity labels could be further analyse from *higher-level context*, to get more information about occupancys model of lifestyle (living habit) with the goal of spotting anomalous activities such falling, suffering a stroke or loss of consciousness. This is the case treated here, where only the binary output streams of the environmental sensors are available in the dataset (better explicated in section 3).

As partially anticipated in the introduction, focusing on this specific research area, two main approaches can be identified in the literature. A first path concerns the adoption of an event-related approach which aims at spotting anomalies on a single activity (or event) and can be re-conducted to a "Punctual Anomaly" problem. A second alternative concerns a pattern-related approach where a pattern of activities is considered with the interest of finding anomalous behavioural patterns and can be re-conducted to a "Collective Anomaly" problem.

One of the first work is an article written by Jain et. al in 2006 [14] referred to MavHome architecture [15] cited in the previous section. In this paper different algorithms have been used to perform activity prediction on sequences of events measuring at the same time the divergence between the detected activity and the one predicted by the algorithm (Markovian approach). To validate the different methods, a mixture of artificial events have been generated (1400 events per day of the week).

An "event-related" anomaly detection system is proposed in the work of Novak et al. in [10]. The authors have used the MavHome project dataset and have built a clustering anomaly detection algorithm using Self Organising Maps (SOM) to detect several types of anomalies such as unusual long or short periods of inactivity, unusual presence or absence and changes in the daily rhythms. This work follows a semi-supervised approach and can be considered one of the most complete works for the detection of such types of anomalies. For this reason the mathematical description and tools used in [10] have been taken as reference for the anomaly detection system presented in this thesis.

Other works as [13], [65] have examined the problem of detecting anomalous patterns of events. Specifically, Shin et al. (2011) [13] proposed a method for the anomaly detection of daily living patterns exploiting a 1-class Support Vector Data Description (SVDD) method to classify a given pattern as normal or abnormal. The dataset for this project is collected using infrared (IR) motion sensors installed in elderly inhabitants homes. The test subjects were suffering from mild to sever other diseases and the data collection lasted for seven months. They track the changes in motion sensors by measuring: (1) how many times (the percentage) the motion sensors were triggered and (2) how often (the frequency) an occupant is moving between two sensors. However, Wong et al. [66] pointed out that this system would often generate false alarms for anomaly detection due to the reason that occupancy activity patterns are irregular. Such as waking up late in the morning or performing usual activities during different times in the day. Differently Kang et al. (2010) [65] used a Hierarchical Hidden Markov Model (HHMM) to recognise and predict the states of a smart home inhabitant and proposed an algorithm based on that model to detect anomalies in the inhabitants behaviour. 77 sensors were installed in two single-person apartments and gathered data for two weeks. The sensors were installed on devices such as refrigerators, drawers, etc.
Another work is the one presented by Aran et. al (EPFL) in [67], where a probabilistic model has been built taking into account the location of the subject at each our of the day defining in this way a likelihood of the subjects behavior based on her/his location and outings.

Aztiria et al. [68] developed a system that could identify possible shift of profile data. A shift or known as concept drift is the change detected, when comparing new occupancy measurements with the recorded frequent activities represented from the profile data. The modification steps required to turn the frequent activity to newly observed activity were used as a measure to decide whether the newly observed activity is to be classified as anomalous.
In conclusion, one of the most recent and advanced work published about Anomaly Detection on Behavioural Patterns is [9] published in 2018. In this paper HTM, Hierarchical Temporal Memory (HTM) theory, is applied on real data and artificial data-set to spot anomalous sequences of activities.

As general consideration it is possible to state that, despite the pattern approach can be considered a more general solution in respect to first one, the event related approach is simpler to be tested and be deployed in a real-context as underlined by [66].

Each of the papers cited has elaborated its own definition of anomaly in respect to its scope. Novak et al. in [10] assumes a part of its starting data-sets is free from anomalies using a Semi-Supervised approach and the anomalies are artificially generated but very few specifics are given in details. In [9] and [67] a validation dataset has been created through real measurements and simulated one. In the first case, different people have been monitored in smart home environment and the anomalies have been manually annotated by the participants themselves. In the second case, the data has been generated by controlling an avatar with OpenSHS simulator tool [69]; in this scenario the user decides to create anomalous behaviours according to her/his own interpretation. Despite using an ad-hoc smart-home dataset can be a valid solution to train and validate a specific algorithm, in real scenario such as the one considered in this study, usually a limited amount of information is available and the data are in most of the cases unlabeled and affected by systemic anomalies (better defined in section 3). Moreover, the huge size of this dataset make unfeasible and sometimes impossible labeling in retrospect as done in [13].

In this study, where an event-related approach is contemplated, we start from a *proper definition of the type of anomalies* and we propose a method to automatically clean and filter the events, preserving in such way only those ones which comply to an operative definition of "recurrence" (later defined). The algorithm proposed is inspired on "Near Cluster Centroid" strategy proposed in [10], but in this case Gas Growing Neural Networks are used instead of SOM. Besides presenting a new algorithm to realize this filtering operation, one of the novelty brought by this work consists in defining a set of specific metrics (with related tests) to assess the confidence of the system proposed. Specifically, beyond testing the robustness of the system in terms of true positive score (ability of classifying as anomalies artificial anomalous events injected on the dataset) and true negative score (capacity of classifying as normal events that are supposed to be normal), we propose a method to test *adaptability* of the system which reflects the capacity of the system to adapt to the changes in the patient's behaviour.

As a general consideration, it is possible to state that the quality and the quantity of the data available can be considered one of the main issues at the moment. Most of the datasets are not open-source and a lack of standardization as well as a differentiation on the way the anomalies are defined, impacts the possibility of comparing the performance among the different approaches.

3

Case Study Presentation

The "Elderly Care" service here considered is based on a real digital platform which collects data from two main types of environmental sensors: PIR (Passive Infrared sensors) and pressure sensors. PIR sensors are installed in different rooms of the patient's habitation and provides a binary output: state ON if a motion is detected, state OFF if no motion is detected. Differently, pressure sensors are usually applied on the windows or on the fridge and are sensible to mechanical strains signaling in this way the interaction between the patient and the object itself. With these premises, in this context the term activity (occupancy activity) is referred to two main typologies of actions: the type 1 indicates the presence of a patient in a given room as for example "presence in bathroom between 8.15 am to 8.25 am", the type 2 indicates the time of usage of an object by the patient as for example "fridge opened at 12.15 am and closed at 12.16 am". All the outcomes produced by the sensors are transmitted to an IoT gateway which sends them to a Reasoner Processing Unit that performs a first processing and elaboration of the data. The binary signals are combined by the Reasoner, providing at the output an interface with the occupancy activities of the patient in time domain. A possible example is represented in Fig. 3.1. A single instance of an activity is called *"event"*, which is therefore characterize by the following attributes: s_t, the starting time, e_t, ending time, $ID\_PATIENT$, the ID of the patient and a, the type of activity. At the current state in the Ticuro platform seven activities have been considered, six of type 1: "presence of the patient in room A" with A=*{kitchen,bathroom,living-room,hallway,bedroom,outside}* and one of type 2: "time open the fridge".

With this initial setup, the Reasoner is able to capture the main aspects of the patient's behavior, enabling the possibility of spotting anomalies whenever the behaviour deviates too much from the usual pattern. As stated in the introduction, the Reasoner considers a specific set of different type of anomalies related to a spe-

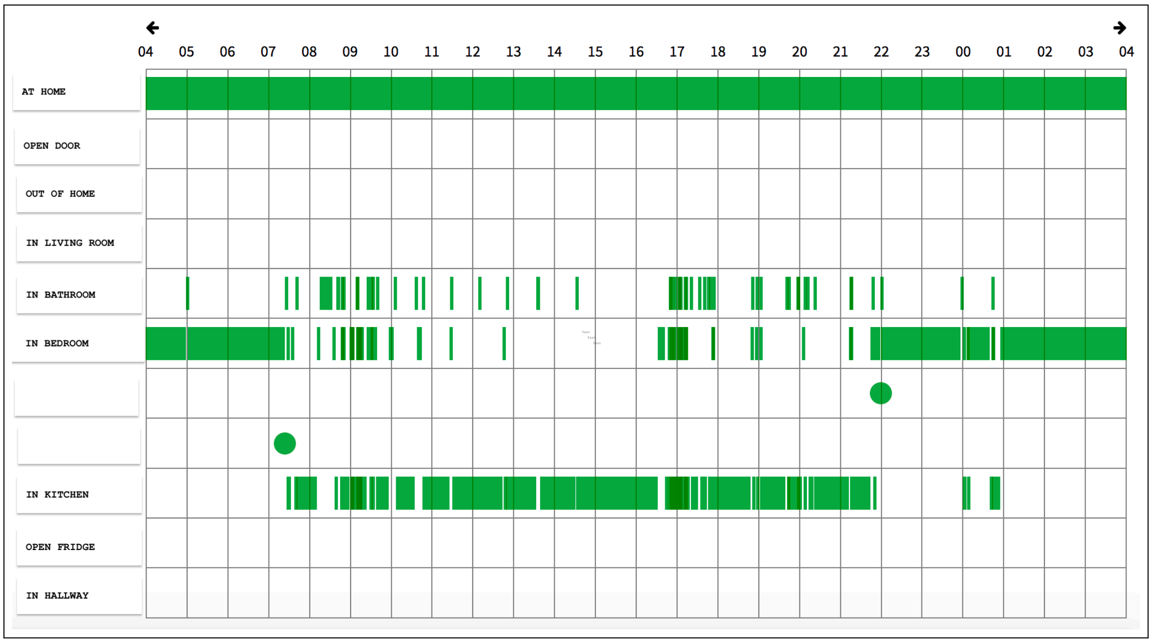


Figure 3.1: Example of the Binary Data Stream at the output of the Reasoner Unit

cific interface of alarms. Given a single event e referred to activity a, the Reasoner can launch different kinds of "Event-Related" alarms each referred with an ID. In this project, we have directed our efforts in studying two types of alarms which can be explained as follows:

1. Anomalous duration of activity a

2. Anomalous repetition of activity a (Anomalous occurrence)

Such specific anomalies have been chosen because they can be considered simple and valid indicators of dangerous situations. An anomalous high duration of an activity can be for example caused by a fall or loss consciousness. Differently, a high occurrence of the same activity could be indicator of weirdness on the behaviour as for example the presence of intruder or an illness. Although this can be seen as a simplification, this is necessary to properly address the problem. Another important specificity of the project lies in the fact that these alarms have to be spotted separately since the reasoner performs different actions according to which alarm/alarms is/are launched. As just discussed, the Reasoner exploits a very simple approach to find these behavioural anomalies which basically consist in checking if the considered quantity (i.e. duration, occurrence of the event) exceeds a predetermined threshold, no matter the habits of the patient and not taking in consideration the time of the day in which the event has occurred. The software here presented tries to overcome this "threshold approach", exploiting a new algorithm and protocol based on machine learning techniques which spot behavioural anomaly according to a data-oriented strategy which considers the past events and the previous story of the patient.

PATIENT_ID	ACTIVITY_ID	START_TIME	END_TIME	ALARMS
123	8	18:39	19:12	No Alarm
221	17	18:25	19:12	Anomalous duration of activity
221	21	18:14	18:25	No Alarm
...	...	...	...	...

Table 3.1: Event Streams at the output of the Reasoner Unit

The **dataset** used to train and validate the system has been collected from measurements taken in a real scenario from 17 different houses of elder people who live alone around the urban and suburban areas of Milan over a time period of 2 years. The data-set consists in a *time-stamp of events* each of which is represented by a tuple of features cited above (s_t, e_t,ID_PATIENT,a) plus a possible list of related alarms launched by the reasoner (represented in Table 3.1). The total number of events collected considering all the patients is around 2 millions, a considerable size for such kind of dataset. As the measurements are taken in a real smart home scenario, the data are sometimes corrupted by so-defined "systemics anomalies" such as problems with the transmission of the signals, hub offline or "crazy sensors". All these events come with anomalies from an analyzer side although they are not triggered by dangerous events but are simply problems in the system itself. In Table 3.2, a summary of the dataset is provided. Specifically, the number of events for each activity a (column) is represented for all the patients (IDs of the patients reported on the row). In the last column, the total amount of time for the monitoring period is summarized in respect to each patient.

ID	7	8	16	17	22	21	19	Total Time
364	1337	15769	21335	20278	13928	41901		943 days, 10:05:31
323	295	23592	33554	34918		25455	3163	943 days, 13:45:45
285	162	17587	21190			21436	3579	790 days, 14:02:31
286	837	18188	11328	55427		38029	5811	868 days, 13:59:01
385	819	38048	66199	153838		167639	9527	943 days, 14:04:49
424	476	7984	9613			12881		774 days, 19:24:14
464	219	10017	9811	10251	13868	14001	750	734 days, 10:39:21
526	1219	22633	14510		41197	21530	1136	943 days, 9:38:28
563	114	18991	5789		28624	8332	510	921 days, 19:37:17
525	1057	7540	17774		43631	44211	368	809 days, 20:58:24
585		422	812		273	1267		327 days, 7:26:50
694			1437			1421		622 days, 12:19:16
695			2369			2379		622 days, 13:13:34
705	115	974	2136	1531				933 days, 15:16:39
444	180	20471	27881	26494	24666	32083	6409	810 days, 15:03:31
303	380	25240	14814	4747		23859	5108	624 days, 10:28:42
523	299	4801	23313		8147	27338	189	867 days, 9:53:23

Table 3.2: Summary of the input data for each patient. In the table the number of events for each activity a (column) is represented for all the patients (rows). In the last column, the total amount of time for the monitoring period is reported for each patient

4

Anomaly detection system

In this chapter the entire anomaly detection system is presented. As better clarified in section 4.2 the procedure here proposed comprises three main phases: feature extraction, event-filtering and classification. In "Feature extraction", referenced in subsection 4.3, the event e is described by a numerical tuple $x = (p, d, o)$ which represents respectively the time position of its start-time, the duration and the occurrence; such formalization has been chosen because it enables a topological description of the anomalies examined. Then, in the "filtering phase", subsection 4.4, the events are drained according to a precise definition of recurrence based on Gas Growing Neural Network clustering technique; at the end in classification phase, illustrated in 3, the probability density function of the recurrent events is estimated, with the possibility of defining an anomaly score function representing the likelihood of the event. At this point a threshold on this function is computed through an dynamic threshold assignation algorithm regulated by an α hyper-parameter. In the recognition phase, if the value of the anomaly score computed on e is lower than the threshold, the event is spotted as anomalous.

In the first part of the chapter, section 4.1, we define the most important concepts used as baseline for the algorithms presented in section 4.2. Specifically in subsection 4.1.1, we present a possible subdivision of time line introducing the notion of Time class. With this concept in 4.1.2, we propose different types of approaches contemplated for Anomaly Detection on Occupancy Activities, defining in proper manner the approach used in this thesis. In subsection 4.1.4, the rationale behind the algorithms for Event-Related Anomaly Detection is explained in relation to other works previously done. In subsection 4.1.5, the Gas Growing Neural Network algorithm is discussed to define the concept of "Recurrent event" in subsection 4.1.1.

In conclusion, in section 4.6, the tests for performances assessment are introduced.

4.1 Preliminary Remarks

4.1.1 Time Subdivision

Each event e is identified by a start-time (s_t) and an end-time (e_t) which identifies precisely the event in time line. For this reason, a precise time subdivision is essential to address properly the problem. In this direction we split a full 24 hour day into a number n of "Time Classes" of same duration. By definition, an event e belongs to a time class if its starting time is included in it. The choice of n is arbitrary but a reasonable value can be identified with $n = 12$ conducting to the following subdivision: $[00:00-02:00], [02:00-04:00], ..., [22:00-24:00]$. The number 12 has been selected mainly because it is a good trade-off between two conditions. At first, we want that events associated to the same time class correspond to similar behaviours (preferring an high value of n). For example, time class [00:00-02:00] contains often events correlated to sleeping activity or, inversely, time class [12:00-14:00] includes events referred to lunch activity. On the other hand if n is too high, each time class hold very few events, compromising the possibility of extrapolating valid statistics.

Time-slot can be seen as a single instance of Time Class. Specifically, while a Time-Slot is related to a specific day, the concept of time class is not; in other words, the Time lot [10-12] referred to the date 10/12/2001 is different to the time slot [10-12] referred to the date 11/12/2001 although they belong to the same time class.

The definitions of these concepts will be very useful for the characterization of the anomalies as well as in the testing phase.

4.1.2 Type of Anomalies and Approaches

Once clarified the context in which the anomaly detection system has to work, a precise definition of anomaly has to be given in respect to the task. It is worth to mention that all the considerations written from now on, hold on the assumption that **anomalies represent a minority with respect to the rest of the data**. If this assumption is not verified, the work done here cannot represent a valid solution.

As indicated more times during this thesis, different types of anomalies can be contemplated.

- A first type of anomalies is related to the specific activity, considering the single event as the object of analysis. Such type of anomalies are examined in "Event-Related" approach where the properties of the event are investigated (when it starts, its duration etc.). We are thus interested in spotting anomalous *events*.

- A second type of anomalies is related to "Time Slot" as considered in [13] and [67]. In this case we are interested in studying the properties of "Time Slot" as for example the number of events occurred in it, the type of events etc. It is clear that "Time slot -related" approach can be considered a valid solution to define anomalies referred to multiple activities (each time slot can be associated to more events), while keeping the problem referred to a punctual anomaly (spotting an anomalous Time-Slot). Despite this could lead to a more general solution, the high dimensionality of the mathematical formalization related to this approach could affect the performances of the anomaly detection system so designed.

- Another possible type of anomaly can be identified in *anomalous pattern of events* i.e. "collective" anomalies, already discussed in 2.4. This approach is interested in spotting anomalies related to a sequence of events considering a wide variety unusual behaviours as for example "moving between the bathroom and the bedroom for too many times" or "going outside and inside repetitively". As stated in section 2.4, this approach has been addressed in many studies as [9] and [65].

As indicated in section 3, in our case the definitions of the anomalies is a project's specific since it derives from the definitions of the *Alarm Interface* considered by the Reasoner module. As indicated in the previous section, the alarms of the Reasoner considered in this work can be summarized with the following semantic:

1. "Anomalous duration of activity a.. "

2. "Anomalous occurrence of activity a.."

Being the alarms related to the single activity a, an "Event-Related" approach seems to be the most valid solution. Specifically we address our efforts in detecting anomalous events according to three types of anomalies: **"Anomaly of duration"**, events that have an anomalous duration as for example an excessive/too short permanence in a room; **"Anomaly of occurrence"**, activities repeated for too many times during the last time interval; **"Unusual Activity"** activities that are simply performed in an unusual time (for example going out at 3.00 a.m. in the morning).

4.1.3 Other remarks

Once the task has been formulated properly, it is possible to state the following remarks.

- The problem here considered is unsupervised since all the events contained in the dataset are unlabeled. The only possible labels could derive from the alarms launched by the previous version of the reasoner. Such alarms were anomalies spotted following deterministic rules, so cannot be taken as reference in our analysis. Moreover, in the data-set, no manual annotation is available about the consistency of the alarm itself. In other words, the use of a Supervised Strategy is infeasible.

- Following an "Event Related" approach, the anomalies are referred to a specific activity a and so it is not reductive considering the activities one independent from the other, performing separated analysis for each of them as done in [10].

- From the peculiarities of the project enounced in previous chapter, it is clear that, although a common analysis can be performed, at a certain point the anomalies have to be spotted separately since the reasoner performs different actions in respect to which alarm/alarms is/are launched.

4.1.4 Unsupervised Anomaly Detection Techniques for Event-related approach

All the considerations made in 4.1.3 and 4.1.2 define in a proper manner the problem, allowing an accurate selection of the analytical tools used to achieve the objective.

Choosing an "Event-related" approach, a first essential step consists in the selection of an appropriate mathematical description of the event e, associating it to a tuple $\mathbf{x} = [x_0, x_1, ..., x_m] \in \mathbb{R}^m$. The mathematical formalization has to be defined in relation to the types of anomalies contemplated. In [10] for example, where they are interested in detecting "Unusual long and short activity", the single event is described as 2D features-vector $e = [p, d]$, where p is the relative position of the starting-time in the day and d is the duration of the event; of course, each event is differentiated on the basis of its activity ID (a parameter) conducting separated analysis for each activity.

With respect to the methods used to spot the anomalies, the remarks contained in section 4.1.3 are particularly useful in this sense. Being the initial context not-supervised, a clustering approach can be considered the most suitable solution at our purpose as underlined in section 2.3. In the literature, various clustering techniques have been considered: K-means, Fuzzy c-means, etc. In [10], Self Organizing Maps (SOM) have been deployed as clustering technique owing to it does not require a number of clusters to be defined in advance as it is with the K-means. SOM are single layer feed-forward networks having an input layer of source nodes that projects directly onto an output layer of neurons. The SOM input layer has m source nodes, each associated with a single component of the input vector x and each neuron in the lattice is connected to all the source nodes. The links (synapses) between the source nodes and the neurons are weighted, such that the $j-th$ neuron is associated with a synaptic-weight vector denoted by $w_j = [w_{j1}, w_{j2}, ..., w_{jm}]^T \in \mathbb{R}^m$, $j = 1, ..., L$, where L is the total number of neurons. In [10], the tool has been exploited to perform anomaly detection in online fashion. Specifically, they divide their procedure into two main phases: learning phase and prediction phase. For a selected time period, that is the learning period (e.g. one month), they consider all the data as anomaly free and let the SOM adapts to the topology of the events-points according to a modified version of the SOM algorithm. The resulting clusters are named normal clusters. In the recognition phase, new events arrive in the system and an anomaly is spotted if the considered event does not belong in a normal cluster.

As just repeatedly stated in this thesis, the anomaly-free assumption made by Novak et. al can be guaranteed only on data-set coming from a monitored scenario, not in a real context. For this reason, we propose a similar clustering algorithm not to perform Anomaly Detection but to perform events filtering, a preliminary phase where the events are filtered online according to a precise definition of "recurrence", section 4.1.1. This phase allows our system to maintain only such events with an *high probability of being normal*: recurrent events. Thanks to this procedure it is possible to switch the problem from unsupervised to semi-supervised where the labels are partially available. As better discussed in the "Processing Pipeline" at section 4.2, only at this stage, the anomalies are detected exploiting a statistical approach. Differently from the study [10], the procedure here explained leverages a Gas Growing Neural Network (GNG) structure to learn and maintain a set of centroids in the events feature space in a totally unsupervised fashion. The set of centroids can be seen as "behavioural prototypes", and the euclidean distance from the event e and its nearest centroid can be seen as divergence metric of e from the usual habit (Near Cluster Centroid approach). To allow the adaptation be performed only on the recurrent events, the algorithm exploits a set of different dictionaries, taking inspiration by the paper [70]. In subsection 4.1.5 we present the GNG algorithm providing in section 4.1.1 an operative definition of recurrence used in event filtering .

4.1.5 Gas Growing Neural Network

Growing Neural Gas is an artificial neural network that is trained using an unsupervised learning algorithm.
The network can be seen as an undirected weighted graph of N nodes, the neurons n_j, each identified by a weight vector $w_j = [w_{j1}, w_{j2}, ..., w_{jm}]^T \in R^m$. When an input $x = [x_1, x_2, ..., x_m]^T \in X$, $X \subset R^m$, is presented to the network, the Euclidean distance between the input and each node is calculated. This procedure divides the feature space into a number of subregions: $X_j = \{x \in X : ||x - w_j|| \leq ||x - w_i||, \forall i\}$,

called Voronoi polyhedra, within which each data vector x is described by the corresponding reference vector w_j, called Best Matching Unit (BMU). When an input is extracted, it induces a synaptic excitation detected by all the neurons in the graph, adapting their weight vectors according to x. As shown in [54], the adaptation rule can be described as a winner-takes-most instead of winner-takes-all rule:

$$\Delta w_{n_1} = \epsilon_b(x - w_{n_1}) \tag{4.1}$$

$$\Delta w_{n_j} = \epsilon_n(x - w_{n_j}), \text{for all neighbors } n_j \text{ of } n_1 \tag{4.2}$$

Simultaneously to the updating of the weights, other operations are performed as follows.

Edge removing: be n_1 and n_2 respectively the first and second Best Matching Units (BMUs). Such units develop a connection between each other called *edge*. An edge is characterized by a parameter called *age* that reflects the frequency with which the pair has been responsive in tandem; at each iteration it is reset to zero whenever a connection occurs. When the age of a connection exceeds a pre-specified lifetime T then the edge is removed.

Adding a new node: In contrast with SOM which considers a fixed number of nodes, in the GNG network, new units are successively added to an initially small network. Precisely, at each iteration, an error quantity is computed and updated only for the neuron corresponding to the Best Matching Unit as follows: $error(n_1) = error(n_1) + ||w_{n_1} - x||$. If the number of input patterns generated is an integer multiple of a parameter λ, a new node is n_r is added to the network. The insertion is performed as follows: be node n_q the neuron with highest error and n_f the neighbor of n_q with highest error variable, then the node n_r is a neuron placed halfway between n_q and its neighbor n_f, or better $w_r = 0.5(w_q + w_f)$.

In the lines above, only the main steps of the GNG clustering algorithm are summarized; a full explanation of the procedure can be found in [54].

4.1.6 Operative Definition of Recurrence

As possible to see from the previous section, GNG incrementally learns the topological relations in a given input data set, dynamically adding and removing neurons and connections, adapting itself to previously unseen input vectors. If GNG is trained on event-points, the centroids represent the "most common behaviours" and can be seen as "compressed representation" of the events themselves. These properties make GNG a good candidate as tool to learn behavioural habits enabling us to formulate an operative definition of recurrence applicable as filtering rule. Such definition has to take into account the following considerations:

- The filtering algorithm has to work in online fashion, being able to filter the events but also to adapt to new unseen events (that reflects for example a change in behaviour). In other words we want that the algorithm is dynamic in respect to time t.

- GNG has not to be trained on all the events generated but only on such events that can be considered recurrent.

These context-related considerations are useful for the formulation of an operative definition of *"recurrence"*.

Before giving this definition, it is necessary to define some quantities useful for the formalization.

1. Given an event e_0 with starting time t which belongs to activity a, being $S_a(e_0, t, r)$ the *"neighbourhood of ray r of event e_0 at time t"* defined as the set of event-points whose distance from the point e_0 is lower than r or $S_a(e_0, t, r) = \{e \in S_a(t) : ||(e, e_0)|| <= r\}$ where $|| \cdot ||$ is the euclidean norm

2. Given the set $S_a(e_0, t, r)$, it is possible to define a set $Date(S_a(e_0, t, d))$ containing the dates on which the events in $S_a(e_0, t, d)$ have been occurred.

With these premises it is possible to give the following definition of "recurrence":

Definition 4.1.1. An event e_0 is considered recurrent if:

1. The distance between the events e_0 and its nearest centroid a time t is lower than a certain threshold d_{th}.

2. Being τ_{min} and τ_{max} two intervals of time, there exists a τ such that:

   (a) The distance between the events e_0 and its nearest centroid at time $t+\tau$ is lower than a certain threshold d_{th};

   (b) $\frac{|Date(S_a(e_0, t+\tau, r))|}{\tau[days]} >= 0.4$, where $\tau[day]$ means the quantity τ expressed in days.

The **first condition** is a natural definition of recurrence. If an event e_0 is relatively close to the nearest centroid, there is an high probability that the event belongs to a behavioural habit and an high number of events are contained within the same neighborhood.
A very basic and practical example is given as follows. Be the event e_0 described by a mathematical tuple $x = (p, d) \in \mathbb{R}^2$ as done in [10]. Let's assume that we are monitoring the occupancy activities of a person in kitchen (a=*"Presence in Kitchen"*), using a monitoring system as the one considered here. Let's assume as well that the daily routine of this person consists in waking up regularly at 7:00 a.m. in the morning and staying in the kitchen for a certain amount of time to

have breakfast and prepare to leave. In this case, for activity a, we will expect to observe a clustered structure around (for example) position $p \in [7.15, 7.20]$ and d, duration, around 30 minutes with a consequent formation of a GNG centroid near the center of this space region.

Differently from the first condition, the **second condition** introduces in the definition the concept of time t and allows our system to be adaptive on new future behaviours not observed in the past. According to this view, if the event e_0 starting at time t does not verify the first condition, it is not totally discarded but it is buffered for a certain period in a local memory structure. At this point, the two sub-conditions for the re-admission of the event on the recurrent set are:
2.a) The network has adapted, and at time $t+\tau$ the distance from e_0 and its nearest centroid is $d < d_{th}$ (according to previous definition).
2.b) we observe a similar behaviour also in the period around t, or formally, the rate between the number of dates which present an event in the neighborhood of ray d of e_0 and number of days passed since e_0 has occurred ($\tau[days]$) is greater than 40%. The condition 2.b is used to quantify the following concept: if a new behaviour occurs repetitively at least in approximately 40% of the following days, then this can be considered a new habit. In this sense choosing a minimum value τ_{min} allows us to define "recurrent" only those behaviours observed for at least a certain amount of time, for example a week. Inversely fixing a τ_{max} prevents the Events Filter by the influence of too old events which cannot be considered any longer a habit. A graphical interpretation of the condition 2.b) is provided in figure 4.1.

A practical implementation of a filtering algorithm based on this definition of recurrence, is possible using an Unsupervised Dictionary Algorithm inspired by the work done in [70]. In this paper, three different dictionaries have been defined to perform vector quantization applied in Subject-Adaptive Unsupervised ECG Signal Compression (SURF) using a GNG structure. The dictionaries are: the current dictionary, the reserved dictionary and the updated dictionary. The rationale behind this procedure can be summarized as follows: the current dictionary learns and maintain a set of prototypes in the data feature space while the remaining two are used for the evaluation of the new and unseen data. In this work a slightly modified version of this method has been adopted in respect to the task. In this case, D_1, the current dictionary, contains the recurrent events which satisfy the condition of recurrence given before; $D3$, the reserved dictionary, contains the not-recurrent events discarded; $D2$, updated dictionary, contains both the recurrent and not recurrent events over a certain period. The main idea is the following: once an event e is generated, if its distance in respect to its nearest centroid n_e is lower than a certain distance threshold d_{th} (Condition 1), it is considered recurrent and it is put on D_1. Else, it is added both on D_3 where, in a future time, the system will check

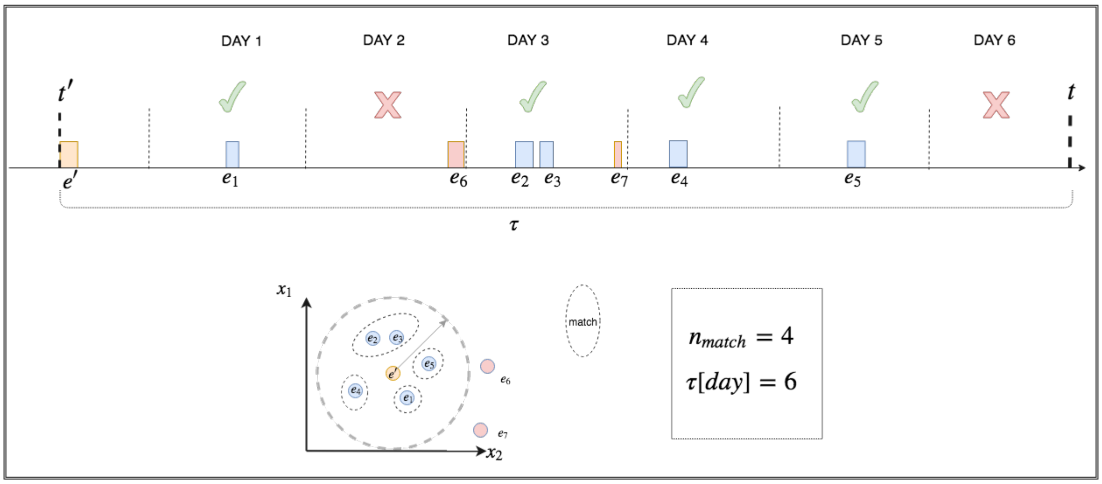


Figure 4.1: Visual representation of Condition 2.b considering a 2D mathematical description of the event $e \rightarrow x = (x_1, x_2)$.

periodically if the other two conditions (2.a, 2.b) are verified. After this check the algorithm adds in any case the event on D_2. A full version and explanation of the algorithm can be found in 4.4.1.

4.2 Processing Pipeline

Considering the remarks made on the previous section, the processing pipeline can be divided into three main phases: the "Feature Extraction" phase, "Filtering Phase" and the "Classification Phase". A schematic view of the pipeline can be found in 4.2.

1. **"Feature Extraction":**

   > *Input*: Event time-stamp e
   > *Output*: Event-tuple $x = (p, d, o)$

   Given a new event e of user u_i, the mathematical description of e is performed in this phase according to the definition of the type of anomalies made in 4.1.2. By this definition, an appropriate mathematical description of e is the tuple $\mathbf{x} = (p, d, o) \in R^3$. This three parameters correspond respectively to: $p =$ the relative position in the day of "start time", $d =$ duration of the event, $o =$ occurrence. There is also an additional feature, a, that indicates the type of activity of e and is used to associate e to its related learning process.

2. **"Filtering Phase"**:

   > *Input*: Event-tuple $x = (p, d, o)$
   > *Output*: Updated dictionaries $D_1, D2, D3$

Once the tuple is computed, in this phase the event is filtered according to the *"operative definition of recurrence"* defined in section 4.1.1. As just anticipated, this filtering procedure is based on three different dictionaries D_1, D_2, D_3 used to implement the definition of recurrence. This phase makes possible to drain in first instance the outliers and the points which do not satisfy the definition of recurrence. Through filtering it possible to pass from a completely unsupervised to semi-supervised context, excluding as well the systemic anomalies existing in the dataset coming from problems as ill-sensors or network issues (more details in section 3). There is a filtering phase and a related event filter E for each activity of a given patient.

3. **"Classification Phase**:

> *Input*: Event tuple $x = (p, d, o)$, Dictionary D_1
> *Output*: Binary Value $\rightarrow A_d$, A_o

* In classification phase the event is classified as Anomalous/ Non-Anomalous in respect to duration (d) and occurrence (o) separately. The input is represented by the event-point x and the set of recurrent events contained in D_1. Differently from clustering approach used in the previous stage, in the Classification phase, a probabilistic model is build following a statistical approach. This phase is divided in two main sub-phases: the training phase, and recognition phase, each repeated as a new event arrives in the data-set. In the training phase the Probability Density Function is estimated in respect to a sub-set of the *recurrent event-points* contained in $D1$, so the ones expected to be anomaly-free (labeled as 0). In this work a Multidimensional Kernel Estimation has been chosen as estimating method.

Once the pdf has been estimated, an anomaly function score (ϕ) is defined. This function reflects the likelihood that an event is described by a certain vector of features x. After this stage, a set of anomalous points X_{an} are generated according to an exogenous distribution Γ_α regulated by a tuning α parameter. Such procedure allows to estimate a value of threshold th through an optimization problem ("Dynamic Threshold Estimation" algorithm). In the recognition phase the event is classified as anomalous if the score ϕ computed on e is lower than th.

* A_d = { True, if the event e has an anomalous duration; False, if not}, A_o = { True, if the event e has an anomalous occurrence; False, if not}

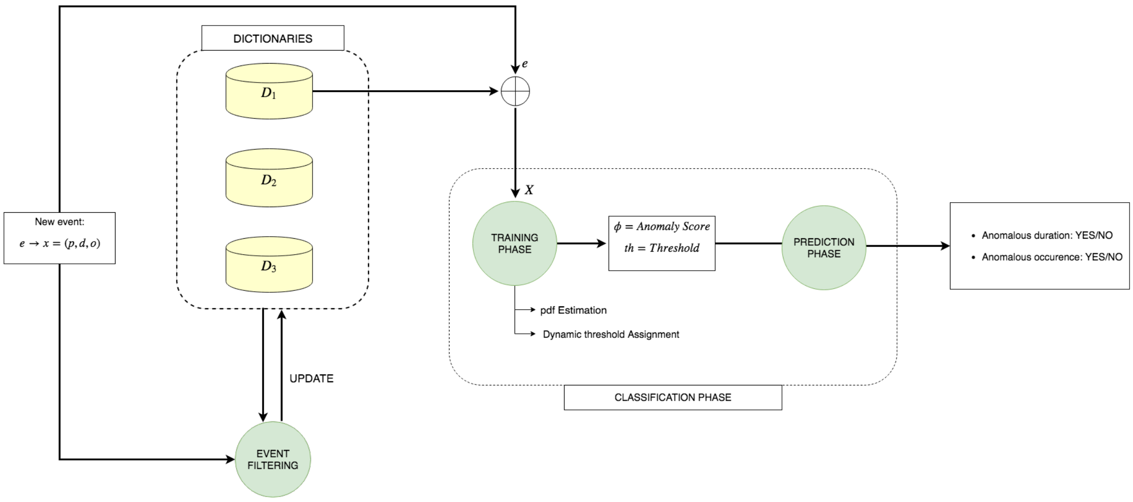


Figure 4.2: Schematic view of the processing pipeline

4.3 Feature Extraction

The event is properly described by a mathematical tuple:

$$e \to \begin{pmatrix} p \\ d \\ o \end{pmatrix} = \begin{pmatrix} Position \\ Duration \\ Occurrence \end{pmatrix}, \text{ with } a \text{ ID of the activity} \tag{4.3}$$

The feature a indicates the activity of the event and it is used to associate the event itself to the related learning process. For this reason the event itself can be considered as a 3D mathematical object described by the tuple (p, d, o). This formalism is very similar to the one proposed in [10] despite in that case only a (p, d) description is considered because they are not interested in spotting anomalies of the "occurrence".

- p, is the "position" parameter and it indicates the relative position during the day of the "Starting Time". It is a decimal number whose value is bounded in the interval $0 < p < 24$. Given a starting time with the following syntax hh:mm:ss which stands respectively for hours (hh), minutes (mm) and seconds (ss), the parameter p is computed as following:

$$p = \frac{(hh) \cdot 3600 + (mm) \cdot 60 + (ss)}{24 \cdot 3600} \tag{4.4}$$

  The introduction of the parameter p is necessary if we want our system be adaptive in respect to the time position during the day.

- d, is the "duration" parameter and it indicates the duration in seconds of the event. It is a positive decimal number whose value is obtained as the difference between the end-time e_t and start-time s_t.

- o is the occurrence parameter and it is an indicator of the *inter-arrival frequency of events* for activity a over the last interval of time preceding the event e. A high occurrence could be indicator of weirdness on the behaviour of the patient as for example the presence of intruder or an illness. A natural estimator of this quantity is represented by the inverse of the mean of inter-arrival times of the events considered. From a mathematical point of view this can be clarified as follows: given an event e_0 with Start Time t and activity ID a, and $S_a = \{e_0, e_1..\}$ the set of events with ID equal to a earlier than e_0, be T_a the set of the starting times t_i of the events $e_i \in S_a$. Considering these definitions it is possible to order the set S_a according to t_i in a way that $t_i > t_{i+1}$; this implies that the event e_i occurs after the event e_{i+1} and the event e_1 has occurred just before the event e_0 (the one considered). In this way it is possible to define the subset $S_a^{(1h)} = e_0.., e_l$ which includes: all the elements of S_a occurred at most *1 hour* before the time t, *plus* the last element, e_l, which correspond to the last event occurs at least 1 hour before the time t. It is thus possible to define the mean of inter-arrival times and the occurrence o of the event e_0 as follows:

$$t_{mean} = \sum_{i=0}^{l} (t_i - t_{i+1})/l \qquad e_i \text{ in } S_a^{(1h)}, \tag{4.5}$$

$$o = \frac{1}{t_{mean}} \tag{4.6}$$

This definition holds on the assumption that the inter-arrival process of the events is stationary as for example a Poisson process. The stationary property is not always verified since the events in the stream can belong to different behaviours. For this reason, after having computed the inter-arrival times, the algorithm verifies the Poisson hypothesis (and so the stationary) through Kolmogorov-Smirnov test, a nonparametric test which allows to know the probability that a given sample of the data is produced by a certain theoretical distribution. The test gives at output a score named p-value bounded in the interval 0-1; the higher the score is, the more probable the sequence has been generated by the given distribution. In this case the KS-test has been used to estimate the p-value that the inter-arrival process is a Poisson Process with a given rate "o" which is equivalent to test that the inter-arrival times of the event sequence $S_a^{(1h)}$ are distributed according to an exponential random variable of mean t_{mean}. If the p-value resulting from the KS-test is > 0.05 the Poisson Hypothesis is accepted, if not it is discarded. In this last case o is simply estimated through a counting process of the events of the same activity a occurred over the last hour.

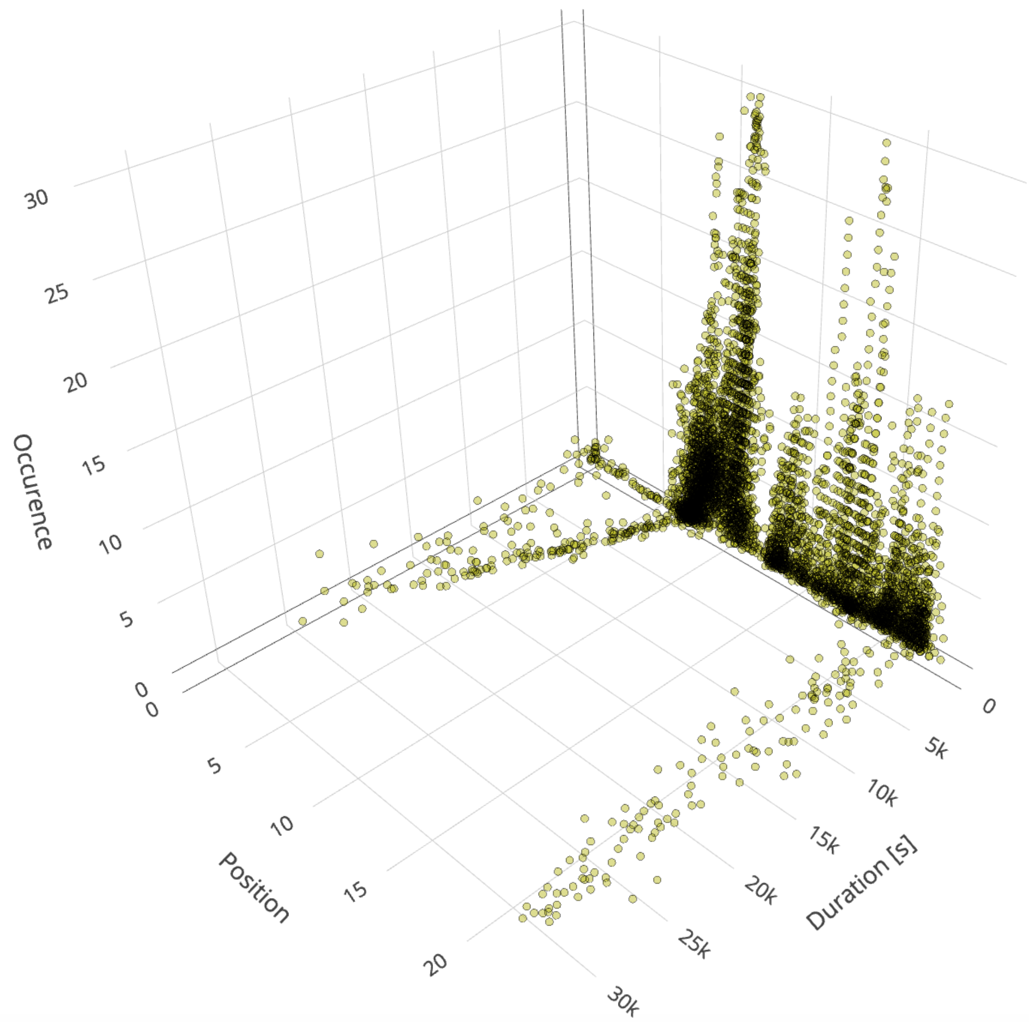


Figure 4.3: Example of events 3-D representation operated after feature extraction. In the figure, it is reported the plot for activity 16 i.e. "Presence in bedroom" for a certain patient.

4.4 Events filtering

As possible to see from the pictures above 4.3, the mathematical description operated in the Features Extraction Phase allows us to have a "topological" interpretation of the anomalies we are looking for. In this sense "Anomalies in duration" and "Anomalies in occurrence" can just be seen as outliers in the 3D domain or simply as points placed in low dense neighborhood.

This concept has been fully examined in 4.1.1 providing a definition of "recurrent event". In the following we exploit such definition to characterize the filtering rule.

4.4.1 Adaptive Filtering Algorithm

As stated in 4.1.1, to practically implement the filtering rules in respect to the definition of recurrence, three different Dictionaries have been created: D_1 which contains the recurrent events (and so the events that are passed to classification phase as training set) occurred in the last period of time $t_{D_1} = 5$ months; D_3 which contains the events that are temporally discarded occurred in the last period of time $t_{D3} = 1$ month; D_2 which contains all the events occurred in the last period of time $t_{D_2} = 1$ month.

Once an event e_0 is generated at time t_0, the algorithm checks if $d(e_0, n_e) < d_{th}$. If this is true, it is considered recurrent and it is added to D_1, where GNG adapts its topology to the new input vector.

If not, it is buffered on D_3 while in any case it is added to D_2. At this point, a periodic checking is performed in respect to all the events e' contained in dictionary D_3.

Let's assume that t' is the starting time of e' and t_0 is the current time. This checking verifies that the distance d from the event e' and its nearest centroid is $d < d_{th}$. If this condition is verified then the event e' is readmitted to D_1 (condition 2.a); if not, thus $d > d_{th}$, the algorithm checks condition 2.b creating the set $Date(D_2(e', t_0, 0.2))$ from the dictionary D_2.

This set contains the dates (days) on which we observe a match with the event e'. A match, in this case, means that at least one event on that date is located in a neighborhood or ray $r = 0.2$ of e' (considering the feature space is normalized [0,1]). The number of elements of this set is called n_{match}. If the rate between n_{match} and the number of days passed since e' is occurred ($n_{day} = \tau[days] = t_0 - t'$) is greater than 40%, then the events contained in $D_2(e', t_0, 0.2)$ are considered recurrent; the events that belong to $D_2(e', t_0, 0.2)$ and also to D_3 are then moved to D_1.

The filtering phase has to be performed considering the (p, d, o) representation because of two main reasons. At first, it is possible to catch the correlation between d and o dimensions, making "more recognizable" such events that present anomalies both in duration and in occurrence (outliers in both dimensions). Secondly, we do not want that in the classification phase the statistics of anomalous events only in one dimension interferes with the statistics of the normal events.

As possible to see in the pseudo-code 1, in the first part of the algorithm two thresholds d_{max} and o_{max} are defined; these two numbers are respectively the threshold on "duration" and the threshold on "occurrence" and represent an upper bound on the two dimensions beyond which an anomaly is spotted in any case. The values represent extensive high values that can be chosen by the Service Care provider in respect to a previous knowledge on the patient behaviour. These *"hard limits"* are useful mainly for two reasons. At first, we need to be conservative in reporting the alarms so if, for example, a patient stays in bathroom for more than 8 hours or a fridge remains opened for more than 10 minutes, an anomaly has to be spotted no matter his/her habits. Secondly, these two thresholds are used as a reference to normalize the events-feature points and avoid numerical errors or dimensional discrepancy (line 4 of 1).

In this way, as a new event e_0 enters in the data-set at time t_0 the following operations are performed:

1. Check if d or o are above the maximum thresholds. If at least one of this condition is true $\longrightarrow$ *Launch an anomaly* (hard limits overcome)

2. Normalize the feature vector (p, d, o) with respect to the maximum thresholds;

3. If the event e is the first $n_{first} = 50$, put it on dictionary D_1

4. If not, check its "recurrence":

    (a) Get the e's nearest neuron n_e, and the threshold $d_{th}(e)$ according to the algorithm explained in algorithm 2.

    (b) If the distance between the event e and the neuron n_e is lower than the threshold put it on dictionary D_1 and trains the *GNG*

    (c) If not, put it on D_3

    (a) For each event in e' in D_3:

        i. $\tau = t_0 - t'$, where t_0 is the current time and t' is the starting time

        ii. If the event e' is buffered in D_3 for a time $\tau > \tau_{max}$ then discarded it;

        iii. If the event e' is buffered in D_3 for a time $\tau < \tau_{min}$ then pass to examine the next event;

        iv. If $\tau_{min} < \tau < \tau_{max}$ then continue;

        v. Check if e' is recurrent according to point 4 and put it on D_1 if the condition is true;

vi. If the conditions of the previous point are not satisfied, getting the events in D_2 such that the distance from e' is below a fixed distance threshold $r = 0.2 \rightarrow D_2(e', t_0, 0.2)$;

vii. Count how many different dates n_{match} are present in the batch $n_{match} = |Date(D_2(e', t_0, 0.2))|$;

viii. Being n_{day} the number of the days of the time τ, if the rate $\frac{n_{match}}{n_{day}} > 0.4$ move the events contained in $D_2(e', t_0, 0.2) \cap D_3$ to D_1 and train the *GNG*.

5. Put e in dictionary D_2

The pseudo code of the algorithm is in Algorithm 1, while a graphical representation with flow charts is in figure 4.5.

4.4.2 Distance Threshold

The definition of *recurrence* brings with it a problem in the definition of the threshold d_{th}. This threshold can be a static value which does not vary with time. This can be a valid solution if the designer of the algorithm owns a specific domain knowledge on the dataset as in [70] but cannot be an optimal choice in terms of adaptability. For this reason, in the filtering algorithm here presented, the threshold is set dynamically and adapts to the topology of the dataset. As possible to see in Algorithm 2, once a new event e enters in filtering process, the threshold on distance d_{th} is assigned dynamically with respect to the position of the nearest node through a simple empirical rule. Basically, once the "filtering algorithm" computes the node n_e, d_{th} is set equal to the distance between the node n_e and its nearest neighbor (n_1). This is done for *enhancing the adaptation* of the system. In high density regions we will observe neurons very close one tho the other. On the other hand, in lower density regions neurons are more sparse. Setting a unique threshold could lead to an excessively strict criteria for the admission of the events belonging to these sparse regions, compromising the capability of the system of adapting also to new unseen events and exploring a new area in the input data space.

Moreover, as possible to see in algorithm 2, two additional thresholds are set respectively as lower and upper bound to prevent some inconveniences coming from some specific cases of point distributions. Specifically, setting an upper bound on the distance threshold prevents the filter to be too much lax in draining the events; this is the case when the nearest neighbor node n_1 is relatively far from n_e, leading to an high value of d_{th}. On the other hand a lower bound on the distance threshold prevents the filter from being too strict in the admission of the events to D_1. This is a relatively common phenomenon which occurs when the Gas Growing Neural Network is trained on more and more events leading to a very dense neural

Algorithm 1 Dynamic Filtering Algorithm

Input: Event $e_0 \to x = (p, d, o)$, occurred at time t_0

1: **if** $(d > d_{max}$ or $o > o_{max})$ **then**
2:     $e \to Anomaly$
3: **end if**
4: *NORMALIZATION:*
    $p \leftarrow p/24$, $d \leftarrow d/d_{max}$, $o \leftarrow o/o_{max}$
5: $e \to Event\_List$
6: **if** $|Event\_List| < 50$ **then**
7:     $e_0 \to D_1$
8: **end if**
9:
10: **CHECK CONDITION 1:**
11: $n_e = get\_nearest\_node(e_0)$
12: $d_{th} = get\_threhsold\_distance(n_e)$
13: $d = ||x - n_e||$
14: **if** $(d < d_{th})$ **then**
15:     $e \to D_1$
16:     $GNG.train(x)$
17: **else**
18:     $e \to D_3$
19: **end if**
20:
21:
22: $e \to D_2$
23:
24: **CHECK CONDITION 2:**
25: **for** e' in D_3 **do**
26:     $t = current\_date$
27:     $\tau = t - date(e')$
28:     **if** $(\tau < \tau_{MAX})$ AND $(\tau > \tau_{MIN})$ **then**
29:         Repeat lines 10,11,12 for e'
30:         **if** $(d < d_{th})$ **then**
31:             remove e' from D_3
32:             $e' \to D_1$
33:             $GNG.train(x)$
34:         **else**
35:             N_{match}=$|Date(D_2(e', t, r = 0.2))|$
36:             $N_{day} = \tau[days]$
37:             **if** $(N_{match}/N_{day} >= 0.4)$ **then**
38:                 each e in $\{D_2(e', t, r = 0.2) \cap D_3\} \to D_1$, train GNG on such points
39:
40:                 $p_{new} = mean(D_2(e', t, r = 0.2))$
41:                 $GNG.add\_node(p_{new})$
42:             **end if**
43:         **end if**
44:     **end if**
45: **end for**

Algorithm 2 Threshold Computation algorithm

```
Function Get_Threshold:
d_min = 0.05
d_max = 0.2
for  a new event e: do
    n_e=get_nearest_neighbor(e)
    n_1=get_nearest_neighbor(n_e)
    d=distance(n_e,n_1)
    if d > d_max then
        d_th = d_max
    else
        d_th = max(d, d_min)
    end if
end for
return  d_th;
```

graph; in this case d_{th} can degenerate into a very low value with the result that many events are excluded and putted in D_3. As general remark, it is possible to state that the dynamic assignment of the threshold can be seen as feedback on the algorithm to cope with some issues in the adaptability of the network. If the network adapts too fast, the distance becomes lower and there is a stricter policy for the admission of the event in D_1; contrary, if the network is too slow in adapting, the threshold is larger. Of course this is only an empirical consideration and for a rigorous description further studies have to be carried out.

Practical implementation and tools The algorithm has to work in a real environment so some empirical choices have to be made.

As regard to the hyper-parameters of the Gas Growing Network, we adopt the following set of hyper-parameters: $n\_starting\_node = 4$, $\epsilon_b = 0.2650$, $\epsilon_n = 0.0315$, $max\_age = 50$, $\lambda = 550$, $\alpha = 0.5$, $d = 0.995$, $max\_nodes = 600$ well explained in [54]. At this purpose, we use MDP Pyhton framework (The Modular toolkit for Data Processing) which provides a complete implementation of GAS Growing Neural Network. For more details visiting: [71].

To increase adaptability in respect to new behaviours, each time a new set of events is readmitted from D_3 to D_1 (line 38) a new node of position equal to the mean of such set is added to the node GNG graph (new habit inserted on the network) (line 40-41).

For computational reasons, the checking at line 24 (CONDITION 2) is not performed each time a new event arrives in the network but it is done periodically as for example twice a day. Another policy adopted by the software consists in completely removing the events which exceed the hard limits o_{max}, d_{max} because we do not want our system adapt to such anomalies. In the first case, if $o(e) > o_{max}$ not only the event e is removed but also all the events in its "one-hour interval" (events occurred one hour before as explained in equation 4.5).

4.5 Classification Phase

Once the filtering phase has been completed, the events contained in dictionary D_1 satisfy the *recurrence property* and can therefore be assumed non-anomalous, targeting them with a label 0 (negative). If in the filtering phase a 3D representation of the events $S = \{e \to x = (p, d, o)\}$ has been considered, the specifics required by the project (the anomalies on occurrence and duration have to be spotted with two different alarms) impose that in the training phase two analyses have to be performed: one analysis with regard to *duration*, considering X_d, the 2D projection of S into the plan (p, d) and the other analysis with regard to *occurrence* considering X_o, the 2D projection of S into the plane (p, o). As just discussed in section 3, differently from the filtering phase where only topological properties of the points (clustering approach) have been used to select the recurrent events, in the classification phase a probabilistic interpretation has been introduced (statistical approach). This interpretation is rightful if the assumption made at the beginning is valid: "anomalies represent a minority in respect to the rest of the data".

This strategy has been also adopted in other works which investigate Anomaly Detection as [59], [60] and can be summed-up in the following three points: 1) *pdf estimation*, 2) *Anomaly score function computation* 3) *Thresholding*. Roughly speaking, this approach consists in defining for each point of the training set X a function ϕ called *Anomaly Score* which reflects some properties of the pdf $p(X)$ in such way that the points placed in space regions with low values of the pdf (anomalies) are differentiated from those placed in regions with greater $p(X)$ (normal points). Once a threshold th is chosen, the anomalous points are such points whose ϕ is greater or lower (depending on the way ϕ is defined) than this threshold [67].

This procedure can be seen as a standard statistical technique for anomaly detection but it is far from being complete since no directions are given on the way the function ϕ and the threshold th has to be chosen. In the following it is reported the implementation adopted in this study:

- *PDF estimation with Kernel:* Given a 2D distribution resulting from the events contained in D_1, the probability density function is estimated through a 2-D Kernel pdf which approximates the pdf in the given domain.

- *Anomaly score function:* In this respect we considered as anomaly score of a point $\mathbf{x}'$ function the function ϕ defined as following:

$$\phi(\mathbf{x}') = P\begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} = P(x'_1 - \epsilon_1 < x_1 < x'_1 + \epsilon_1, x'_2 - \epsilon_2 < x_2 < x'_2 + \epsilon_2) \quad (4.7)$$

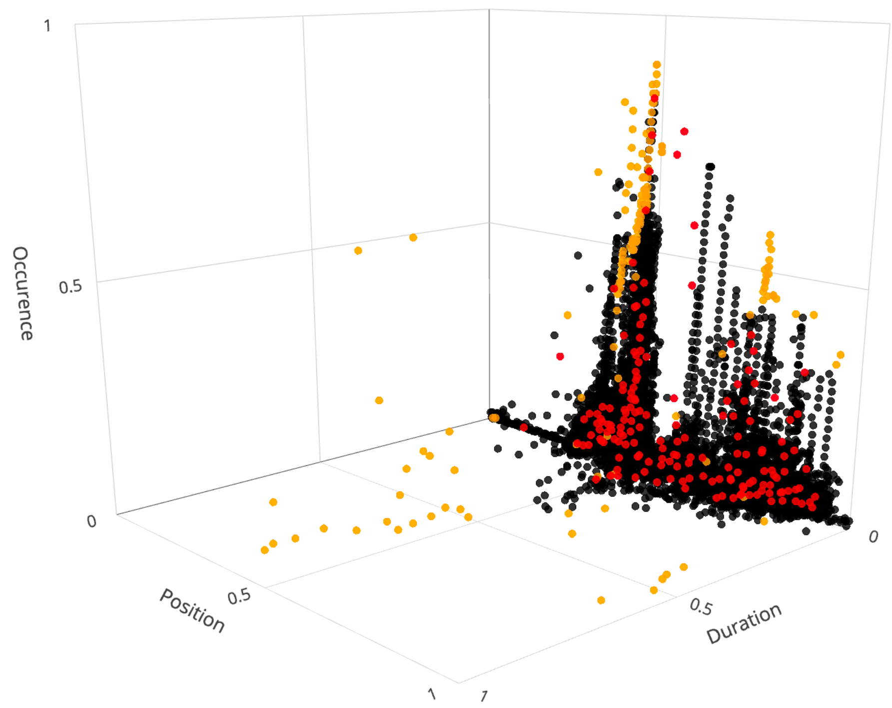


Figure 4.4: Example of the filtering phase: D_1 points (recurrent) are coloured in black, $D3$ points (not.recurrent) are coloured in orange; in red it is underlined node-graph of the GNG

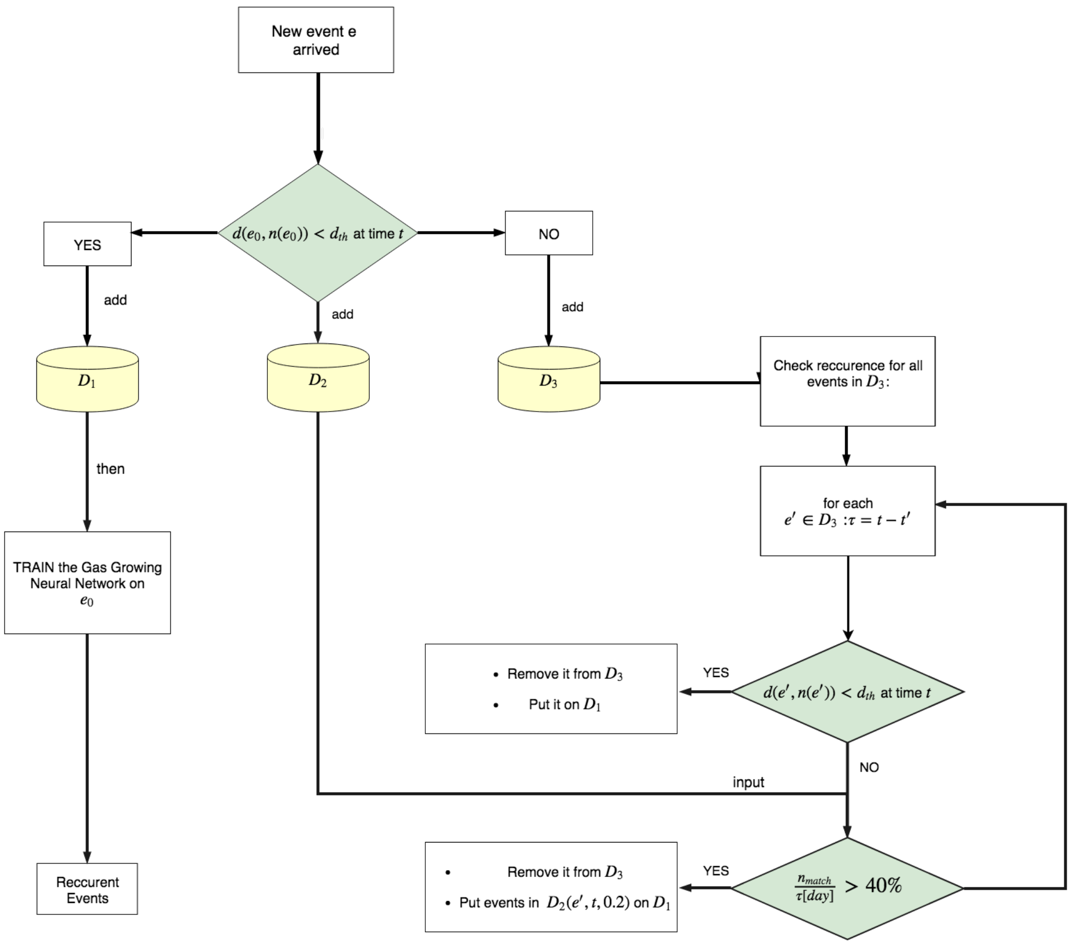


Figure 4.5: Schematic view of the filtering phase

This reflects the likelihood of finding x in a neighbor of x' which, in practical terms, is equivalent to the probability of having an event started at that time position with a value of duration or occurrence in a certain interval.

- *Thresholding:* With the previous definition of $\phi(x)$ it is clear that anomalies are points whose values of ϕ are *below* a certain threshold. Giving to ϕ a probabilistic meaning, the choice of this threshold can be done also from the user side according to an empirical evaluation as for example selecting only such events whose ϕ is under a certain value as 0.002 or 0.001. Choosing a priory this number can be very impractical in a real context also because the function ϕ depends on the values of ϵ_1 and ϵ_2 (size of the square neighborhood). For these reasons, this value has been computed through a procedure named *"Dynamic Threshold Selection"* addressed in the next paragraph.

Kernel Density Estimation For the classification of a new event $e_0 \rightarrow x = (p_0, d_0, o_0)$ passed through the filtering phase, a subset $S(e_0)$ is created selecting from dictionary D_1 only those events which belong to the same time-slot of e_0. Precisely this selection picks only those events whose starting time (parameter p) lies in the range $p_0 \pm 1$, where $p_0 - 1$ is the starting time of e_0 minus 1 hour, and $p_0 + 1$ starting time of e_0 plus 1 hour. To avoid numerical issues given by the fact that p_0 is a time value included in range [0,24], once $S(e_0)$ has been created, the values of p are normalized in the range [0,2]; this conducts to $p_0 = 1$.

At this point two parallel analyses are carried out: the first to detect "Anomalies in occurrence" considering X_o,the second to detect "Anomalies in duration" considering X_d. Being these two analyses identical from the theoretical point of view, from now on we use $X = \{(p, x_2)\}$ to define the sets X_d and X_o, considering $x_2 = \{d, o\}$. A visual representation of the projected X domain is represented in figure 4.6. Once X is created, the 2D pdf-estimation is operated through KDE, kernel density estimation, a non - parametric statistical method which allows the estimation $p(X)$ through the sum of different Bi-dimensional Gaussian Kernels better described in [72]. This method has been implemented through the library "statsmodels" of Pyhton which furnishes a non fft-based approach for KDE providing as well useful tools to tune important parameters in Kernel Density Estimation as the for example Band-with and the shape of kernels.

Anomaly score Function Given a pdf $p(X)$ on set X, the function ϕ used for anomaly discrimination of the point x' is:

$$\phi(\mathbf{x'}) = \phi \begin{pmatrix} p' \\ x'_2 \end{pmatrix} = P(p' - \epsilon_1 < p < p' + \epsilon_1, x'_2 - \epsilon_2 < x_2 < x'_2 + \epsilon_2), \text{with } x_2 = \{p, o\} \tag{4.8}$$

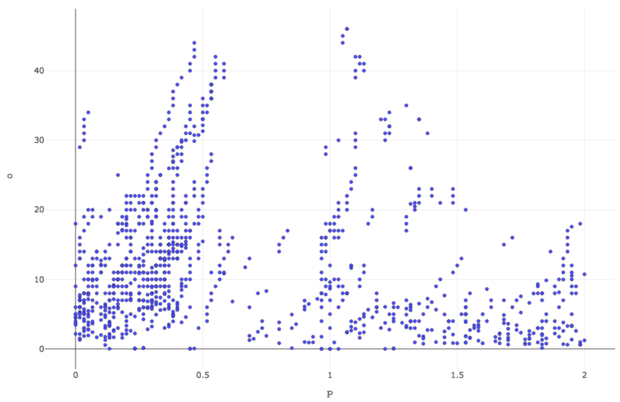


Figure 4.6: Event points projected in (p,o)= (position,occurrence) domain for occurrence analysis. The parameter p is normalized [0,2] since events contained in a 2 hours interval are considered.

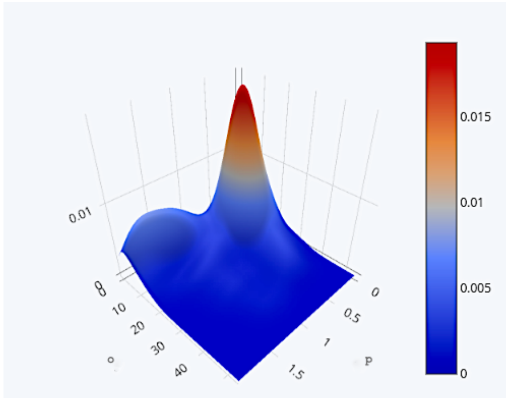


Figure 4.7: Anomaly score computed in (p,o) domain.

From a mathematical point of view this is equivalent to calculate the probability that a point x is contained in a square of vertices: $\{A = (p' - \epsilon_1, x_2' - \epsilon_2), B = (p' - \epsilon_1, x_2' + \epsilon_2), C = (p' + \epsilon_1, x_2' + \epsilon_2), D = (p' + \epsilon_1, x_2' - \epsilon_2)\}$ with the result that $\phi(x')$ can be simply estimated through a sum/differences of Cumulative Density Functions (CDF or $F_X(x)$) calculated on the vertices. From a trivial geometric consideration this function can be calculated as following:

$$\phi(x') = F_X(C) - F_X(D) - F_X(A) + F_X(B) \tag{4.9}$$

The value ϵ_1 referred to p is set equal to $\epsilon_1 = 0.1$ which is equivalent to 6 min. in 24h time domain; in respect to ϵ_2 we have set $\epsilon_2 = 1$ for occurrence analysis and $\epsilon_2 = d_{max}/100$ for duration analysis. In figure 4.7, the Anomaly Score on a (p,o) domain is represented.

Threshold Selection Once $\phi(x)$ has been computed $\forall x \in X$, the threshold can be set manually by an external user or can be selected dynamically through a procedure called *Dynamic threshold assignment*. Due to the "recurrence property", events in X are supposed free from anomalies with the consequence that the threshold th has to be set in such way that the number of $x \in X$ such that $\phi(x) > th$ is maximized. Considering only this criteria, a trivial solution of this maximization problem is $th = 0$ which is of course incorrect. For this reason it is necessary to introduce in the problem a counterbalance given by the introduction of an artificial distribution X_{an}, complementary to the one observed in X. Being these points generated by an exogenous distribution, the classification procedure have to recognize them as "anomalies" $\to \phi(x_{an}) < th$. Before presenting the algorithm some definitions are in order.

Definition 4.5.1. Given the feature $x_2 = \{d, o\}$, be $X[x_2] = \{x_2 | x = (p, x_2) \ \ \& \ x \in X\}$ the set of the features x_2 in X. Be $p(x_2)$ the one-dimensional distribution estimated through kernel method on $X[x_2]$

Definition 4.5.2. Given a parameter $0 < \alpha < 1, \alpha \in \mathbb{R}$, be Γ_α a one-dimensional gamma random-variable of parameters k (shape parameter) and θ (scale parameter) whose distribution is defined by the function $\gamma_\alpha(x, k, \theta) = \frac{1}{\theta^k \Gamma(k)} x^{k-1} e^{-(\frac{x}{\theta})}$ and such that the area of the intersection surface under the two functions and $p(x_2)$ is equal to α.

This means the following integral:

$$\int_0^{+\infty} (\min[\gamma_\alpha(x_2), p(x_2)] dx_2) = \alpha \tag{4.10}$$

The distribution of Γ_α, can be estimated imposing a condition on the variance that, in a gamma distribution, is related to the parameters k and θ by: $var(\Gamma_\alpha) = k\theta^2$.If we want the variance of Γ_α is equal to the variance of $X[x_2]$, the problem can be formalized as following:

$$\begin{cases} var(\Gamma_\alpha) = k\theta^2 = var(X[x_2]) \\ \min_{k,\theta} | \int_0^{+\infty} (\min[\gamma_\alpha(x_2), p(x_2)] dx_2) - \alpha | \end{cases}$$

Once the parameters k, θ are estimated, it is possible to create the distribution X_{an}, generating the parameter x_2 through $\gamma_\alpha(x_2, k, \theta)$ and the parameter p through a uniform random variable $U([0, 2])$. At this point the problem of finding an optimum threshold th can be addressed as presented in the following lines. Be $X_{tot} = \{X \cup X_{an}\}$ the entire training set and $l_{tot} = \{l(X) \cup l(X_{an})\}$ the labels, where $l(X) = 0$ and $l(X_{an}) = 1$:

$$th_{opt} = \operatorname*{argmax}_{th} F_1(l_{tot}, \mathbb{1}\{\phi(X_{tot}) < th\}) \tag{4.11}$$

where $\mathbb{1}$ is the indicator function and F_1 is the f1-score defined as $F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$ between the labels l_{tot} and the classifications $\mathbb{1}\{\phi(X_{tot}) < th\}$.

The α parameter is a hyper-parameter which can be tuned by the user. In practical terms finding a value for this parameter is equivalent to give a definition of anomaly. The higher α is, the more similar the distribution $\gamma_\alpha(x)$ is in respect to the estimated not-anomalous $p(X)$; so it is more probable that an event is classified as anomalous. Inversely a lower α conducts to a $\gamma_\alpha(x)$ very different from $p(X)$ so it is less probable that an event is classified as anomalous because the definition of anomaly is less strict. In this way, the service provider can tune this parameter online according to the real feedbacks observed in reality.

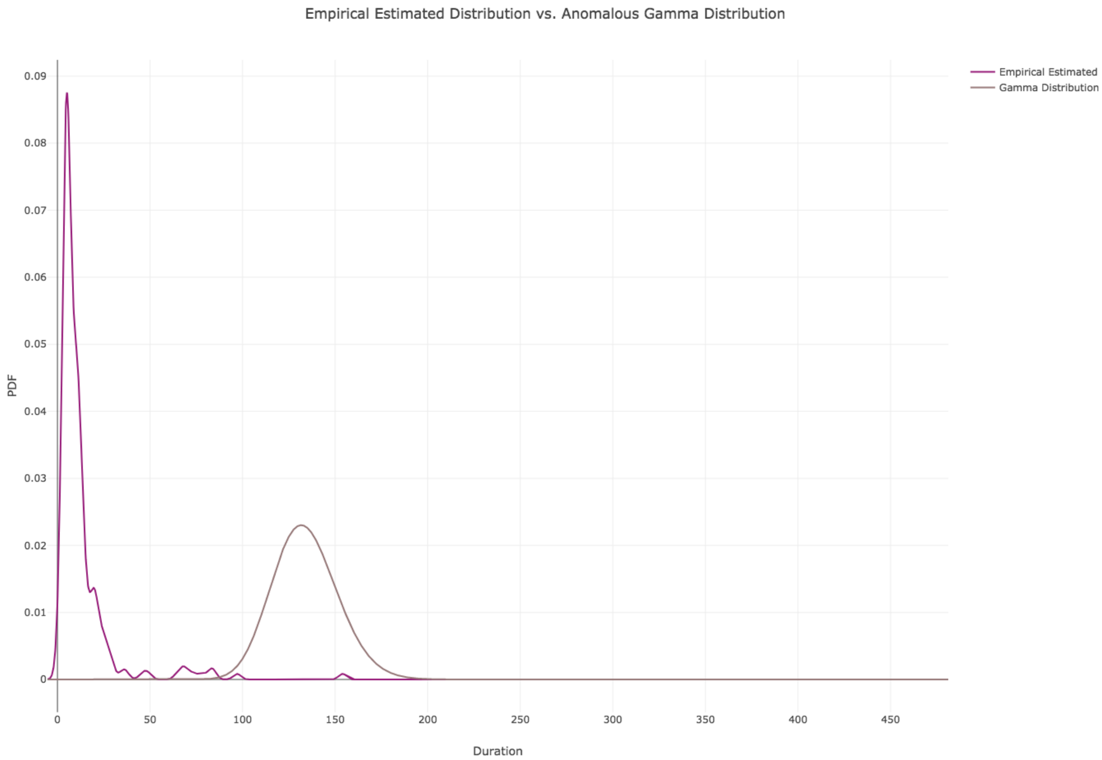


Figure 4.8: Example of exogenous distribution Γ_α wiht $\alpha = 0.01$

RECOGNITION PHASE In the previous paragraphs, the training phase is presented. This phase is executed each time a new event is generated and it is repeated both for the occurrence and for the duration analysis giving a "classifier" as output. This classifier can be defined by a pair $(\phi(X), th)$ respectively, anomaly score function and its related threshold. In the recognition phase the event $e_0 \rightarrow x = (p_0, d_0, o_0)$ is therefore classified as following:

$$y_{pred}(d) = \mathbb{1}\{\phi_d([p_0, d_0]) < th_d\} \tag{4.12}$$

$$y_{pred}(o) = \mathbb{1}\{\phi_o([p_0, o_0]) < th_o\} \tag{4.13}$$

where $y_{pred}(d)$ and $y_{pred}(o)$ are respectively the predictions made for *duration* and for *occurrence*. Being the training phase performed on a sub set of the recurrent events contained in D_1, X, it is possible that in some cases this set is very poorly populated because very few events occur in that given time slot compromising the validity of all the statistics computed in the classification phase. To avoid such problem, in this case the classification through probabilistic approach is performed only if X counts a certain number of events: $|X| > 30$. Differently, if this condition is not satisfied, no classification is performed, and the output of the filtering phase can be seen as the only valid result. If the event is considered recurrent (inserted in D_1) then the event is considered not anomalous; inversely if the event is not recurrent (it is thus inserted in $D3$) then it is considered as anomaly and signaled to the Reasoner unit's as *Unusual Activity.*

4.6 Tests for performances assessment

Working in an unsupervised setting where no label is available, the testing phase is crucial to properly assess the confidence of the proposed Anomaly Detection Algorithm. The tests discussed in the following have been designed to show two main properties of the system. The first property is the *"reliability"* intended as the capacity of the system of recognizing as anomalous events that are anomalous indeed (True Positive Score) and as Non Anomalous events that are that are, in fact, non anomalous (True Negative Score). The second property is the *"adaptability"* intended as the capacity of the system to adapt in respect to the habits of the patient. All the tests have to be conducted both for duration and occurrence and in respect to all the activities and patients considered in the platform.

Despite differences in the mechanisms, all the tests cited share the same baseline. From the whole list of events S occurred for a given patient, it is extracted the subset S_a defined as the set of events occurred in respect to the activity a. The set S_a is split into two parts S_{a1} and S_{a2}, respectively the 85% and 15% of the entire set. The first part is given in input to an Event Filter, E_{a1} which drains the events in S_{a1} furnishing at the output a set of recurrent events $D_1(S_{a1})$. A schematic view of the tests is available on Fig. 4.11.

Reliability Tests These tests aim at providing an estimation of True Positive Score and True Negative Score. To properly compute these quantities two different tests have been modeled respectively: "No Anomaly Test" and "Artificial Anomaly Test" performed for the classification process for duration and for occurrence.

- **"No Anomaly Test"**: In the first part of the test, the set S_{a2} is filtered with a new empty filter E_{a2} . The events contained in $D_1(S_{a2})$ are given in input to E_a and then classified through the classification process. The filtering process performed by E_{a2} maintains only the recurrent events so we can suppose such events being free from anomalies and can be labelled as 0. The result of the test is given by the percentage of events classified as "Non anomalous" on the total of events contained in $D_1(S_{a2})$.

- **"Artificial Anomaly Test"**: This analysis furnishes at the output a possible estimation of the True Positive Score. At the beginning of the test, a certain number n_{an} of anomalous events are generated artificially (S_{an}) and then inserted into a stream of recurrent events coming from the set $D_1(S_a)$. The whole stream is given in input to the filter E_{a1} and the anomalous artificial events are classified. The procedure to generate the artificial anomalies is described below:

- **Test for duration**: all the events belonging to the set S_a are subdivided in 12 time classes. In respect to the distribution of each subset $X_{d,i} = \{(p,d)\}$, it is computed an anomalous distribution $\gamma_{\alpha,i}(k,\theta)$ with the same procedure discussed before in the training phase 4.8. Given a time class j, an anomalous event is generated at the beginning of this class with duration d given by $\gamma_{\alpha,j}(k,\theta)$. An anomalous event is inserted in a different time class randomly chosen once every 5 days. In this test the number of anomalous events injected are $n = 25$ and the result is given by the percentage of "Anomalous" events classified as "Anomalous".
- **Test for occurrence**: As in the previous case the time class approach has been adopted computing for each time class i an anomalous distribution $\gamma_{\alpha,i}(k,\theta)$ in respect to the distribution of each subset $X_{o,i} = \{(p,o)\}$ with the same procedure discussed in the training phase. Differently from the duration test in this case for a given a time class j, a *sequence* of anomalous events is created according to an anomalous occurrence o_j, generated from the distribution $\gamma_{\alpha,j}(k,\theta)$. Effectively once o_j is computed, the events in the anomalous sequence $\{e_0, e_1 ... e_n\}$ are generated with inter-arrival time equal to an exponential random variable of mean equal to $1/o_j$ ($\Delta t(e_i, e_{i+1}) \sim Exp(\lambda = o_j)$) for the whole duration of the time class (Poisson arrival).

  An anomalous sequence is inserted in a different time class randomly chosen once every 5 days. In this test the number of anomalous sequences injected are $n = 25$ and the result is given by the percentage of sequence where at *least one event* is classified as "Anomalous".

ADAPTABILITY TESTS Different from the first ones, this kind of tests have been designed to assess the adaptability of our system to new anomalous behaviors. The test is based on a simple consideration: if an anomaly happens repetitively in time, at a certain point this has not to be longer observed as an anomaly.

- **"Adaptability for high occurrence"**: The setting of this test is very similar to the one proposed in the previous "Artificial Anomaly Test". The anomalous sequences are created in the same manner using an anomalous distribution $\gamma_{\alpha,j}$ but they are inserted on a fixed the time class j. Once the anomalous sequences are generated in reference to this class, they are inserted once a day in the stream of events. To asses the adaptability two parallel process are started: the first in which the adaptation filtering is enabled, the other in which the adaptation is disabled. In the second case the initial distribution

X on which the training process is performed does not change with time, so we will expect that in the first process at certain point the anomalies will be recognized as normal while in the second the anomalies are always recognized.

- **"Adaptability Test for duration"**:
  - **Adaptability for high duration**: The setting of this test is identical of the previous one except from the fact that an anomaly is represented by a single event with anomalous duration.
  - **Adaptability for range duration**: This test aims at testing the ability of our system in recognizing unusual events with duration in a certain range. The setting of this test is a bit different from the other ones: being d_m the maximum value of duration of the events in the set S_{a1}, the events whose duration within the interval $[0.3 \cdot d_m, 0.7 \cdot d_m]$ are firstly excluded from the dataset. The remaining events are given in input to an empty filter E_{a3} which adapts only on events with durations out of this range. At this point a stream of *all* anomalous events S_{range} is generated with duration equal to $\frac{0.3 \cdot d_m + 0.7 \cdot d_m}{2}$. The inter-arrival time between the events in S_{range} is exponential with mean $t_{mean} = 4$ hours. Since E_{a3} has "never learned" events in range $[0.3 \cdot d_m, 0.7 \cdot d_m]$ we expect that event S_{range} will be classified as anomalous until a certain time, when the learning framework will add them to the recurrent class. From there, they will be classified as normal. Besides proving adaptability this test shows as well the robustness of our system in identifying anomalous events of unseen durations as for example too short duration.

Multi Patient Test In conclusion, to check the adaptability from a general point of view, a test has been proposed considering a set of patients $\{u_i\}$ where u_0 is the starting patient. Initially an event-filter $E_{a1}(u_0)$ filters the events of u_0 patient related to the activity a, then all the events contained in the $D_1, D_2, D3$ are removed maintaining only the Neuron Graph of the Gas Growing Neural Network. At this point a classification process is started inserting sequentially the events contained in S_{a2} of the next user u_i. This procedure is repeated each time a patient is changed. If the system adapts to the habits of each user, we will observe a sudden sharp increment in the number of anomalies detected as events of a different new patient are inserted. Then we expect to observe a more and more flattening trend with the passing of time since the system adapts to the habits of the new patient.

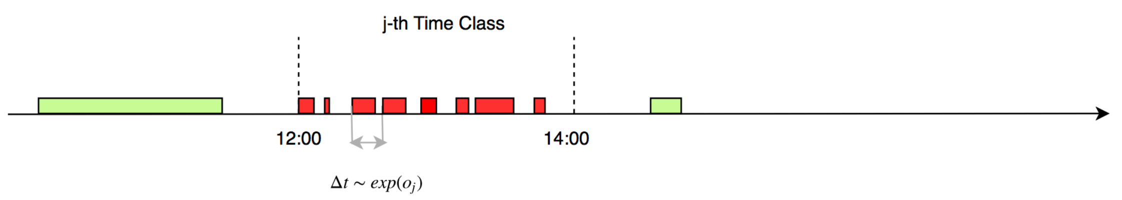


Figure 4.9: High occurrence test: in red the events of the anomalous sequence injected in the events stream

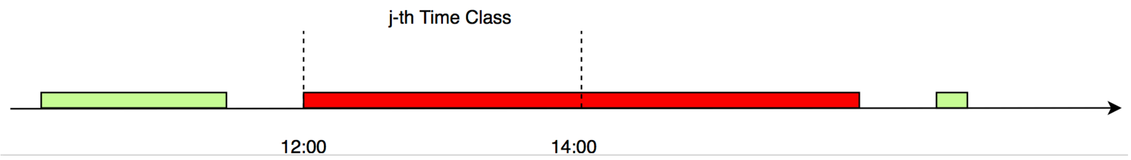


Figure 4.10: High duration test: in red the anomalous event

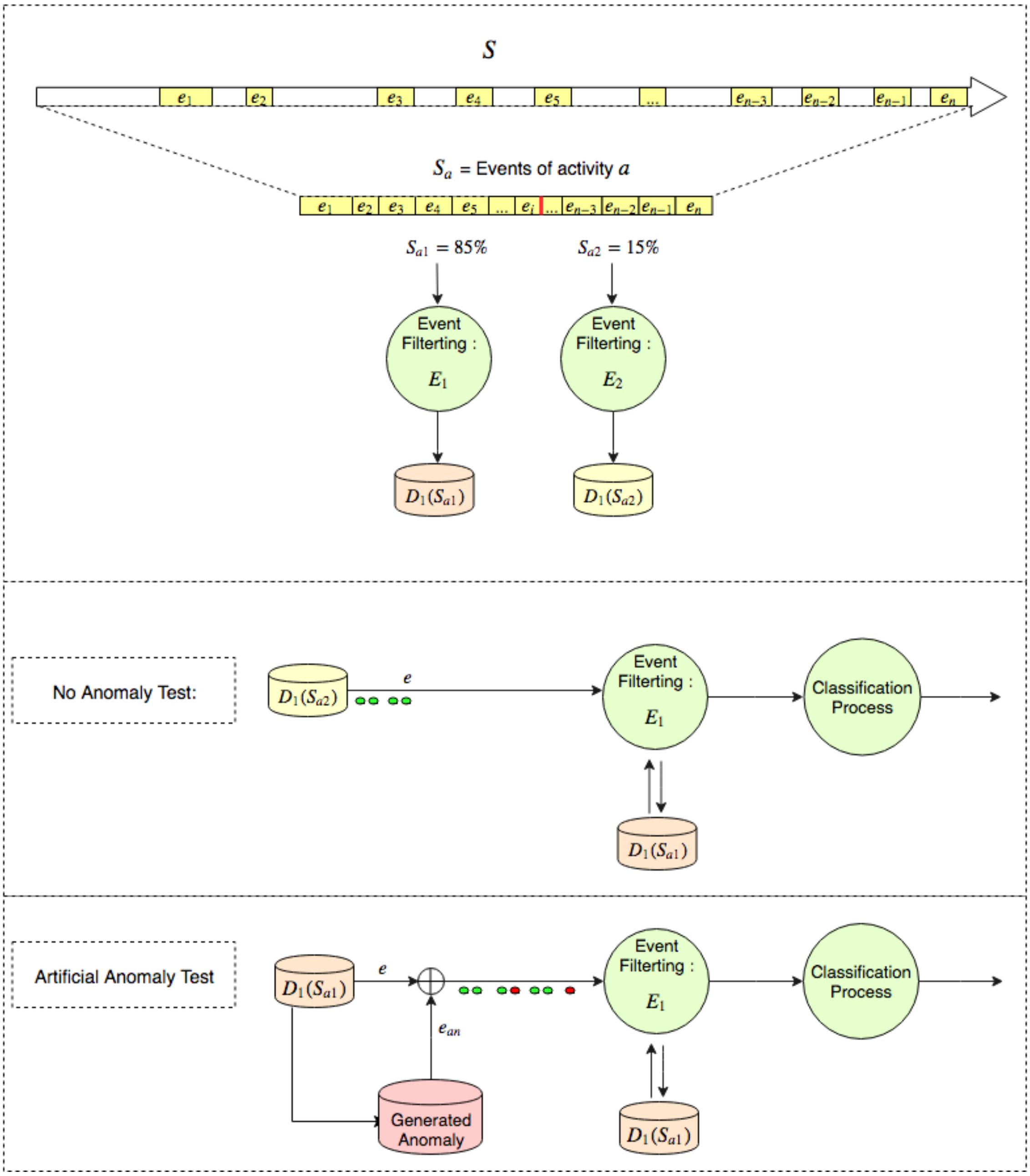


Figure 4.11: Scheme of the tests

5
Results

In the following, the main results are summarized. As discussed in the previous chapter, two types of metric will be discussed: "reliability" metrics to show the effective performances of the system and "adaptability" metrics to show the capacity of our system to adapt to the patient's habit. In the table 3.2 presented in chapter 3, the "input" data of all the patients (IDs reported on the rows) are collected including the number of events available for each activity and the duration of the monitoring period (last column). In the following, this table will be used often to explain some outcomes.

A visual exemplification of the the classification process is reported in the following figure 5.1. In this particular situation, an anomalous sequence of events is considered (anomalous occurrence); in the x-axis the time line is reported and the events are indicated through the scatters. The black-dashed line represents the values of the dynamic threshold corresponding to the event (th) while the orange one corresponds the anomaly scores (ϕ) in logarithmic scale. As can be seen from the plot, as the anomaly scores ϕ goes under the threshold the value of the threshold, the events are spotted as anomalous (red colour of the scatters).

5.0.1 Reliability Tests

In the tables 5.2 5.1, the results of the the reliability tests are summarized. On the rows, the patients IDs are reported while the columns indicate the different activities considered so far in Ticuro Platform: $a = \{$ *'Presence in Kitchen'=21,'Presence in Bathroom'=8, 'Presence in Bedroom'=16, 'Presence in Hallway'=17, 'Presence in Sitting Room'=22, 'Presence Outside'=7, 'Time Open Fridge'=19* $\}$.

The α chosen for the analysis is equal to 0.01 and it is used *both for the training phase both for generating the artificial anomalies* in different time classes. The time

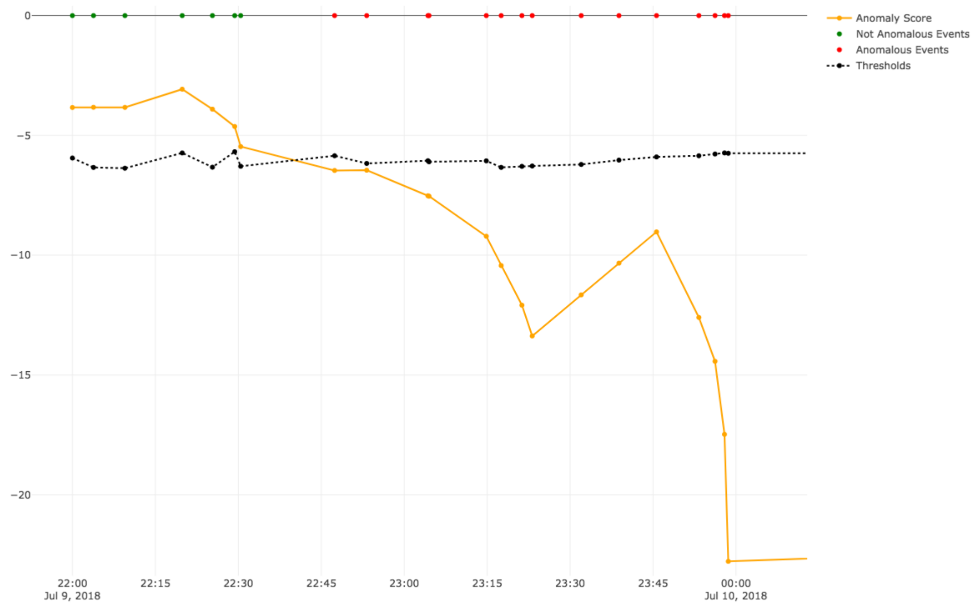


Figure 5.1: Classification process: the orange line corresponds to the anomaly score while the black line to the dynamic threshold

class $j = \{1, 2, ..12\}$, in which the anomalies are inserted, uniformly at random. In this case, if the time class is under-populated, or in other words, the number of events in time class j is < 30, the anomaly distribution is always a gamma but the parameters (k, θ) are fixed.

In the tables 5.1 5.2, for each column it is shown the "True Negative Rate" (TNR) and the "True Positive Rate" (TPR).
TNR is referred to the "No anomaly Test" and is obtained with the following expression: $TNR = \frac{|\{classification(e)==0\}|}{|D1(S_{a2})|}$, where the numerator represents the amount of events e that has been classified as non anomalous and the total number of events, $D1(S_{a2})$. TPR is referred to Artificial Anomaly test and represents the percentage of anomalies classified as anomalous (positive) in respect to the number the total number of anomalies injected in the event stream. TPR is calculated differently in respect to the type of analysis.
For duration analysis: $TPR = \frac{|[classification(e)==1]|}{|S_{an}|}$ where the numerator represents the number of anomalous events classified as 1 and the denominator represents the total number of anomalous events inserted.
Differently, for the occurrence analysis, TPR is calculated as
$TPR = \frac{|s=\{e_0,e_1...e_k\}|\exists e:classification(e)==1|}{|\{s\}|}$ where the numerator represents the number of anomalous sequences where *at least one event has been classified as 1* while the denominator represents the total number of anomalous sequences.

Activity Test	21 TNR	TPR	8 TNR	TPR	16 TNR	TPR	7 TNR	TPR	19 TNR	TPR	22 TNR	TPR	17 TNR	TPR
364	1	0.92	0.97	0.84	1.00	0.92	0.93	0.72			1	0.8	0.97	0.8
323	1	0.92	0.98	0.92	1.00	0.92	0.79	0.6	0.97	1			1	0.84
286	0.99	0.88	0.97	0.8	0.96	0.84	0.58	0.92	0.92	0.96			0.98	0.84
285	0.99	0.76	1	1	1.00	0.76	0.58	0.64	0.98	1				
385	0.88	0.8	0.99	0.8	0.98	0.80	0.79	0.48	0.96	0.92			0.89	0.92
525	1	0.92	0.99	0.6	1.00	0.64	0.96	0.76	0.91	0.8	0.99	0.96		
444	0.96	0.88	0.99	0.96	0.98	0.92	0.63	0.76	0.89	0.92	0.98	0.92	0.94	0.84
303	0.99	0.84	0.84	1	0.90	0.88	0.6	0.88	0.96	1			0.93	0.88
323	1	0.92	0.98	0.92	1.00	0.92	0.79	0.6	0.97	1			1	0.84
523	1	0.88	0.99	0.8	1.00	0.68	0.91	0.72	0.88	0.96	1	0.84		
424	0.96	0.68	0.94	0.8	0.94	0.92	0.95	0.68						
526	0.97	0.76	0.98	0.88	0.95	0.72	0.96	0.68		0.8	0.96	0.68		
694	0.99	0.72			0.94	0.84								
695	0.86	0.72			0.89	0.80								
705			0.94	0.72	1.00	0.76	0.88	0.72					1	0.64
563	0.97	0.84	0.98	0.84	0.95	0.68	0.59	0.88	0.94	0.92	1	0.6		
585	0.95	0.84	0.94	0.84	1.00	0.76					0.8	0.72		
Median	0.99	0.84	0.98	0.84	0.98	0.80	0.79	0.72	0.95	0.96	0.99	0.8	0.97	0.84
Mean	0.97	0.83	0.96	0.848	0.97	0.81	0.78	0.71	0.94	0.93	0.96	0.78	0.96	0.83

Table 5.1: True Negative Rate (TNR) and True positive Rate (TPR) for 'Occurence Analysis'

As can be seen from above, TPR and and TNR have been obtained with different tests and are calculated in different ways in respect to the cases interested. For this reason, performance metrics as Recall, Precision are misleading in this context and will not be considered.

In respect to **duration analysis**, figure 5.2, it is possible to see that our system can be considered robust for both the tests. Specifically TPR rate is, except for one case, always greater than 75% for each activity and patient. At thee same time, the values of TNR, whose median is always greater than 90% for each activity, prove that the system is robust to false alarms as well. This result is particularly good if we consider the following aspects. At first, in the training phase, the normal and the anomalous distribution has certain "common area" represented by α, implying that a certain value of False Negative is expected. Secondly, the recurrent events in $D1(S_{a2})$, used in the No Anomaly Test, are effectively "future events" so it is possible that new habits of the patient have been captured by the filtering process E_2.

As regards to **occurrence analysis**, very similar outcomes can be observed. The TPR rate is higher than 80% in almost all the considered cases, preserving at the same time a quite good score for what concerns TNR.

Activity Test	21 TNR	 TPR	8 TNR	 TPR	16 TNR	 TPR	7 TNR	 TPR	19 TNR	 TPR	22 TNR	 TPR	17 TNR	 TPR
364	1	1	0.97	0.72	1.00	1.00	0.95	0.64			1	0.88	1	0.84
323	1	1	0.84	0.96	0.96	0.88	0.93	0.64	0.98	0.88			1	0.96
286	0.92	0.88	0.96	0.8	0.96	0.84	0.83	0.52	0.95	0.96			0.97	0.84
285	1	0.88	1	1	1.00	0.88	0.95	0.36	0.99	1				
385	0.98	0.88	0.98	1	0.98	0.80	0.88	0.64	0.98	0.96			0.99	0.84
525	0.96	0.84	1	0.92	0.98	1.00	0.96	0.76	0.96	0.96	0.97	0.96		
444	0.99	0.92	1	1	0.99	0.92	0.85	0.36	0.96	0.92	0.96	0.8	0.97	0.88
303	0.98	0.92	0.98	1	0.96	0.84	0.96	0.84	0.96	0.96			0.97	0.8
323	1	1	0.84	0.96	0.96	0.88	0.93	0.64	0.98	0.88			1	0.96
523	0.98	0.84	0.97	0.96	0.98	1.00	0.86	0.6	0.88	1	0.96	0.76		
424	0.97	1	0.99	0.96	0.96	0.80	0.95	0.68						
526	0.98	1	0.98	0.8	0.96	0.96	0.97	0.68		1	0.95	0.96		
694	1	0.96			0.97	0.92								
695	0.92	0.72			0.99	0.84								
705			0.97	0.88	0.93	0.92	0.82	0.32					0.98	0.96
563	0.97	0.92	0.98	0.92	0.99	0.80	0.71	0.36	0.96	1	0.98	0.48		
585	0.73	0.52	0.92	0.88	0.94	0.76					0.75	0.52		
Median	0.98	0.92	0.98	0.96	0.97	0.88	0.93	0.64	0.96	0.96	0.96	0.8	0.985	0.86
Mean	0.96	0.8925	0.95	0.92	0.976	0.88	0.89	0.57	0.96	0.95	0.94	0.77	0.985	0.885

Table 5.2: True Negative Rate (TNR) and True positive Rate (TPR) for 'Duration Analysis'

As possible to see in both the two analysis, the activity 7 exhibits the lowest performances for both analyses. This can be explained by the values contained in table 3.2, where it is possible to see that activity 7 counts the lowest number of events compared to the others activities. This fact can condition the adaptability of the event filter with the consequent inability of the system to compute valid statistics for the classification phase. The relationship between scores and number of events is also confirmed by the fact that the patients who present an higher number of events for activity 7, present at same time an higher scores both for *duration* and *occurrence* analyses.

All the tests cited above have been conducted considering a single value of $\alpha = 0.01$. In this direction, we conduct other tests on a subset of patients considering another value of α, namely $\alpha = 0.005$ and the results are reported in 5.3 and 5.4. As expected, if the value of alpha parameter is decreased, the TNR score slightly augments in almost every patient. This is caused by the fact that decreasing α, we provide a "less strict definition of anomaly to the system" and the threshold th is calibrated to recognize more different anomalies in respect to the normal distribution.

Activity	21		8		16		7		19		22		17	
Alpha	0.01	0.005	0.01	0.005	0.01	0.005	0.01	0.005	0.01	0.005	0.01	0.005	0.01	0.005
285	0.995	0.998	0.995	0.998	0.998	0.998	0.947	0.947	0.993	0.996				
286	0.917	0.930	0.960	0.963	0.965	0.965	0.828	0.828	0.948	0.952			0.968	0.973
323	0.998	0.998	0.838	0.843	0.965	0.973	0.929	0.929	0.984	0.984			1	1
525	0.962	0.968	0.995	0.995	0.980	0.982	0.959	0.959	0.962	0.962	0.968	0.977		
585	0.733	0.819	0.925	0.925	0.939	0.955					0.75	0.75		
Median	0.962	0.968	0.960	0.963	0.965	0.973	0.938	0.938	0.973	0.973	0.859	0.863	0.984	0.986
Mean	0.921	0.943	0.943	0.945	0.969	0.975	0.916	0.916	0.972	0.974	0.859	0.863	0.984	0.986

Table 5.3: Occurence Analysis: a comparison of TNR performances between two values of $\alpha = \{0.01, 0.005\}$

Activity	21		8		16		7		19		22		17	
Alpha	0.01	0.005	0.01	0.005	0.01	0.005	0.01	0.005	0.01	0.005	0.01	0.005	0.01	0.005
285	0.990	0.998	0.998	0.998	1.000	1.000	0.579	0.737	0.985	0.987				
286	0.993	0.993	0.972	0.978	0.958	0.973	0.578	0.562	0.92	0.928			0.98	0.98
323	1.000	1.000	0.983	0.983	1.000	0.998	0.786	0.833	0.966	0.969			0.997	1
525	1.000	1.000	0.993	0.993	1.000	1.000	0.959	0.979	0.906	0.925	0.992	0.998		
585	0.952	0.952	0.943	0.925	1.000	1.000					0.8	0.8		
Median	0.993	0.998	0.983	0.983	1.000	1.000	0.682	0.785	0.943	0.948	0.896	0.899	0.988	0.99
Mean	0.987	0.989	0.978	0.975	0.992	0.994	0.726	0.778	0.944	0.952	0.896	0.899	0.988	0.99

Table 5.4: Duration Analysis: a comparison of TNR performances between two values of $\alpha = \{0.01, 0.005\}$

Global Threshold As just stated, the parameter α is used as reference for the dynamic computation of the threshold th on the anomaly score ϕ. Possibly, the threshold can be also fixed not considering the dynamic assignation. In the following, we assess the relation between the value of the threshold and the related performances (TPR and TNR rate). Although the classification process is performed in respect to the time class of the event e and differs according to the type of activities and patients, the following two ROC curves show that it is possible to identify a range of "reasonable" values of global thresholds maintaining a consistent performance. These two plots have been designed contemplating all the events of all the activities for every patient examined in "No Anomaly Test" and "Artificial Anomaly Test". Specifically we considered as Negative set the union of the events contained in $D_1(S_{a2})$ ("No Anomaly Test") of each single patient, while we consider as Positive the anomalous events artificially generated in "Artificial Anomaly Test" (with $alpha = 0.01$). During the tests, the value of the anomaly score $\phi(e)$ has been saved for each single event, using these data as input to draw the ROC curve. Of course occurrence analysis and duration analysis have been split.

As possible to see in the graphs 5.2 and 5.3, for occurrence analysis, reasonable values of the threshold are $0.001 < th < 0.005$, which show the best compromise in terms of TNR and TPR scores. In duration analysis, instead, an optimal range can be found between $0.001 < th < 0.016$. As indicated before, these are just indications for a clever assignment of a global threshold in case the dynamic assignment through

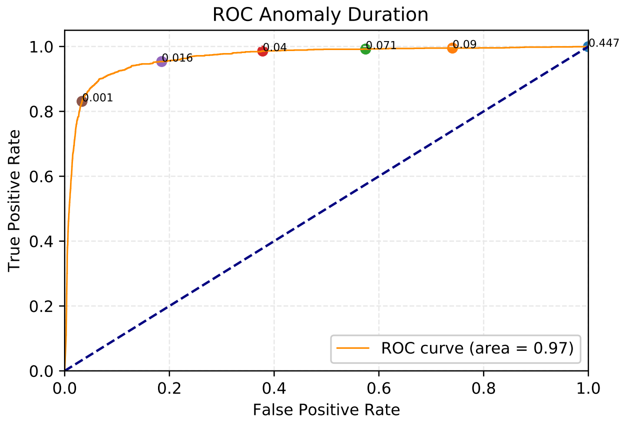


Figure 5.2: ROC curve for "Duration Analysis" in respect to the threshold th

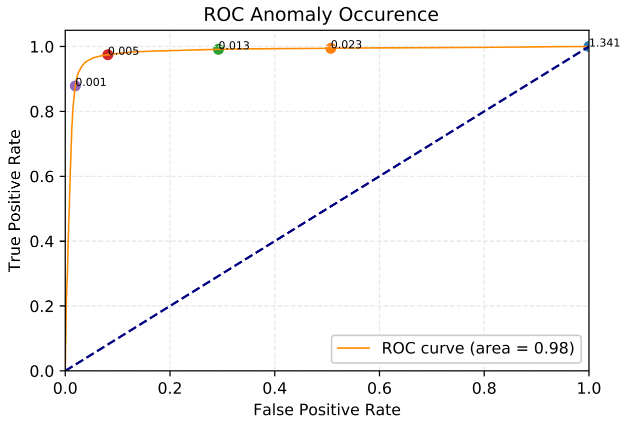


Figure 5.3: ROC curve for "Occurence Analysis" in respect to the threshold th

α is not possible.

5.0.2 Adaptability Tests

As in the previous analysis, the adaptability tests have been conducted with respect to all the activities and patients, considering as well 3 different seeds for each case. As explained before, for each single test two parallel assessments have been conducted: the first with a filter which adapts to the new events and the second with a filter that does not update, performing the classification always in respect to the initial distribution.

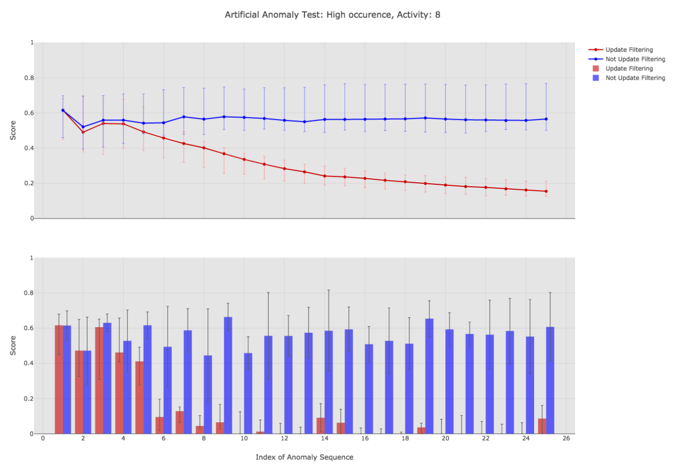


Figure 5.4: Adaptability in respect to "high occurrence test"; Results for activity $a = 8$

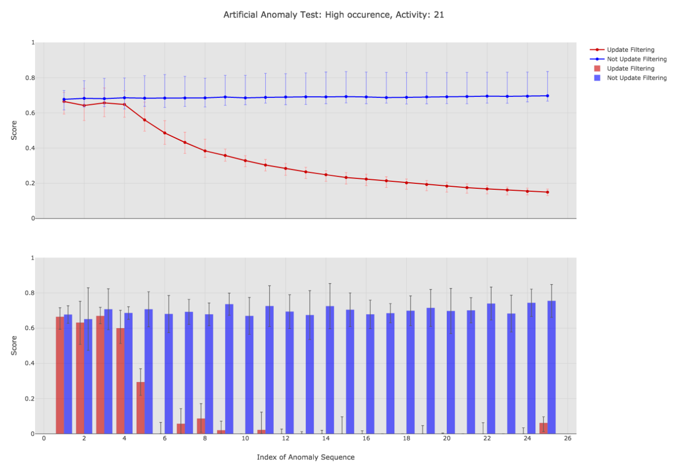


Figure 5.5: Adaptability in respect "high occurrence test"; Results for activity $a = 21$

To investigate **"Adaptability for high occurrence"**, an anomaly sequence of occurrence $o_j = \gamma_{\alpha,j}(k, \theta)$ is inserted in time class $j = 11$ (22:00-00:00) with daily cadence for $n = 25$ days. As possible to see in the graphs 7.8, 7.7, in these tests two scores have been considered: R_1 and R_2 defined in what follows.

Being i the index of the anomalous sequence s_i, R_1 is defined as the percentage of anomalous events detected as anomalous in the single sequence: $R_1(i) = \frac{|\{classification(e \in s_i)==1|\}}{|\{e \in s_i\}|}$. This score is reported below in the graphs and is expected to be < 1 since the first part of the anomalous events in the sequence is not recognized as anomalous.
Differently R_2 (reported above in the graphs) is a cumulative value and for the sequence s_i represents the percentage of events detected as anomalous up to i in respect to the total of amount anomalous events inserted up to i: $R_2(i) = \frac{\sum_{k=0}^{i} |\{classification(e \in s_k)==1\}|}{\sum_{k=0}^{i} |\{e \in s_k\}|}$.

Considering figure 7.7,7.8, in the x-axis it is indicated the progressive index of the sequence, while, the two scores are indicated in y-axis, R_2 above, R_1 below. The values plotted on the charts are the median values of the scores obtained for each different patient, with the related 25% and 75% percentile. The results are reported for activities 21 and 8 (5.7 and 5.6 respectively) while the plots for the other activities can be found in the appendix.

As expected, both the scores related to the updated classification process (indicated in red) decrease with the progress of time; this is related to the fact that the filtering process has adapted to the new behaviours till the point at which no events in the sequence are detected as anomalous i.e. $R_1(i) = 0$ (complete adaptation). Inversely, the not-updated classification process shows quite steady performances in respect to both the metrics because it continues to recognize as anomalies the events contained in inserted sequences.

To investigate **"Adaptability for high duration"**, an anomalous event with duration $d_j = \gamma_{\alpha,j}(k, \theta)$ is inserted in time class $j = 8$ (16:00-18:00) with daily cadence for $n = 25$ days. In contrast with the previous case, only one score, R, has been considered in this context. Being i the index of the anomalous event e_i, the score is defined as the percentage of anomalous events detected as anomalous in respect to the total amount of anomalous events inserted up to the time i: $R(i) = \frac{|\{classification(e_i \in \{e_0, e_1, ..e_i\})==1\}|}{|\{e_0, e_1, ..e_i\}|}$. In the x-axis, it is indicated the progressive index of the event, while in the y-axis the score $R(i)$. The results are reported for activities 21 and 8 (5.7, 5.6) while the plots for the other activities can be found in the appendix 7.1.

As in the previous cases the values indicated on the graph is a median value in respect to the scores obtained for each different patient with the related 25% and 75% percentile. A decreasing in the graph corresponds to mis-classification (anomaly considered as normal) while an increasing correspond to a correct classification. Also in this case, the score related to the updated classification process (indicated with red) decrease with the progress of time while the not updated classification process shows quite steady around 1.

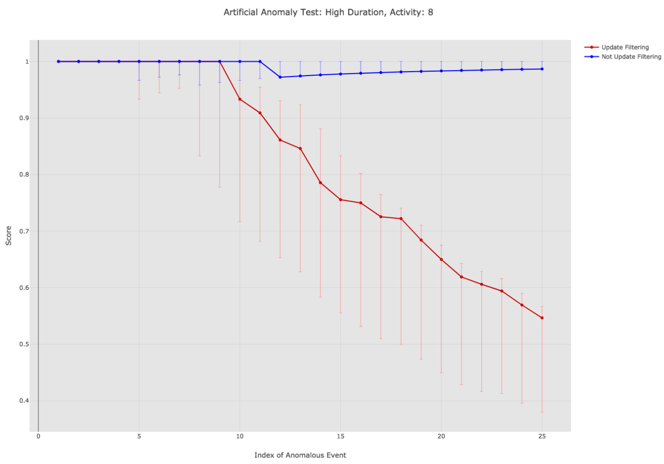


Figure 5.6: Adaptability in respect to "high duration test"; Results for activity $a = 8$

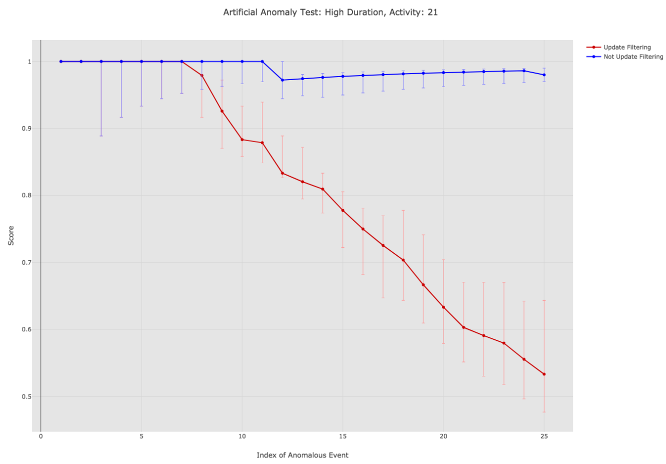


Figure 5.7: Adaptability in respect to "high duration test"; Results for activity $a = 21$

The third and last test to assess Duration Adaptability is the *Adaptability for range duration* whose outcomes are shown in plot 5.8. As in the previous test, the index of the event is indicated on x-axis while the score, computed as before, is reported on y-axis.

In this case only the updated process has been considered and in the graph it is possible to see the median result considering all the activities and all the patients.

The trend observed confirms the expectations. Specifically, the fact that the curve remains stable around 1 for the first part of the event streams shows a good capacity of the architecture to detect anomalies in duration not only in respect to "high duration" but more generally in respect to unseen durations. In addition the fact that the median of the score declines with time verifies the adaptability of the system also in this context.

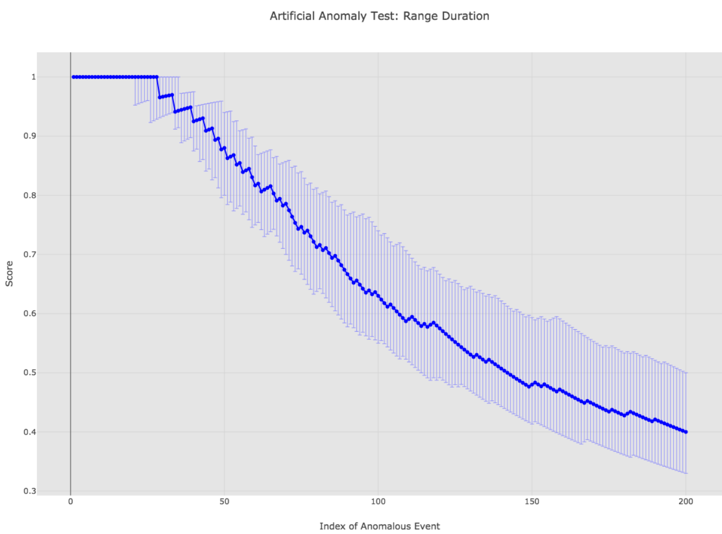


Figure 5.8: Test for "Range Duration"

Adaptability of the filtering process The filtering process based on GAS Growing Neural Network clustering technique exploits 3 different dictionaries: D_1, D_2, D_3. In D_3 dictionary the "non-recurrent" events are contained. If the filter adapts to the patient's habits we expect to observe a pick in the number of events buffered in D3 as soon as the process is started, with a consequent decreasing once the filter starts to adapt (events transferred from D3 to D1). This trend can be observed in the graphs below where the number of events in D3 in figure 5.9 is plotted against the the time for two different activities.

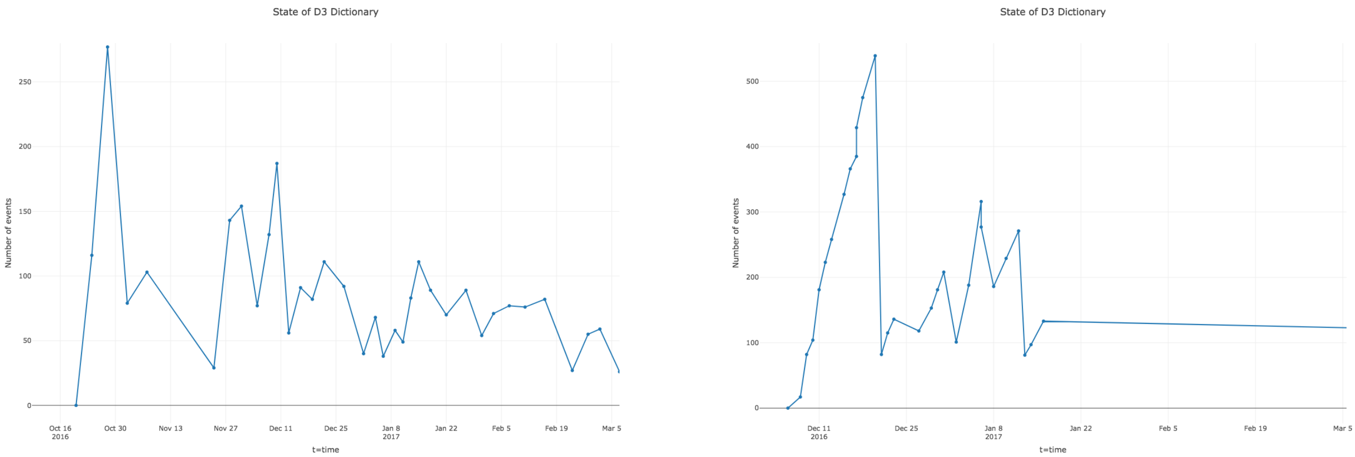


Figure 5.9: Number of events contained in D_3 as function of time

Multi patient test   The last test proposed aims at verifying adaptability in respect to data coming from several patients. The results of this test are summed up in figure 5.10 where 5 different patients: $286, 323, 563, 526, 525$ have been considered in this order in respect to activity $a = 21$. Initially an event filter E' has been trained on the events of patient 285 then, as discussed in section 4.6, all the events are removed from the dictionary and only the graph of the GAS network is left. This procedure is repeated each time a patient is changed. In the x-axis the time line is depicted, while the y-axis represents the number of anomalies detected up that time by the system. Even in this case the expectations have been verified as possible to see in the graph. An initial rapid growth of the curve (relative high value of the derivative) is followed by a more and more stable trend as time pass. It is possible to see that for patient 323 the adaptation does not occur. This can be explained by the fact that this patient presents a "crazy sensor" which generated anomalies in the occurrence as shown by the fact that in a very short time (around middle of February 2019) the curve augments around 100 units in a very short time.

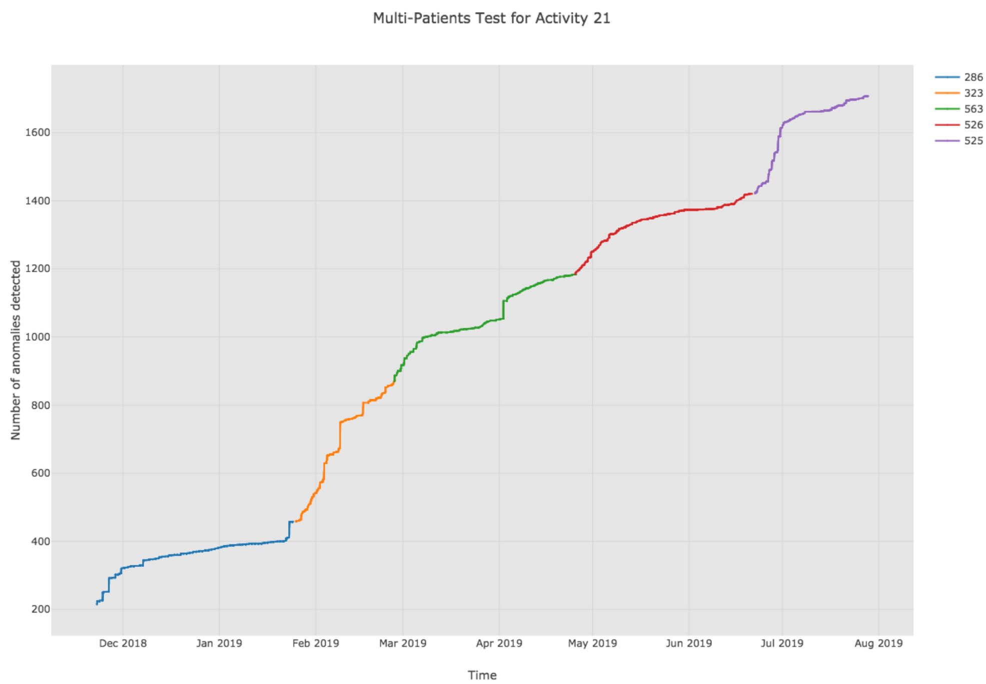


Figure 5.10: Multi-patients test results

6
Conclusion

In this thesis, the problem of Anomaly detection in Behavioural Monitoring has been addressed. Specifically, a new software has been developed for Ticuro Reply, a Digital Platform created by Healthy Reply to furnish Smart Assisted Living Services to elderly people who live alone (Elderly Care Project). The aim of the project consists in using a new data-driven approach to spot anomalies in occupancy activities of daily living ("Presence in a certain room" or "Time of usage of an object") of senior patients monitored through environmental sensors (PIR and Pressure Sensors). Differently from other existing studies which have investigated the problem of spotting anomalies in behavioural patterns, in this work, anomalies related to the single activity (or event) are contemplated. Specifically the algorithm has been designed to recognize three main types of anomalies related to the single activity: "Anomaly in Duration", an activity last for too long or that is too short; "Anomaly in Occurrence", repetition of a given type of activity for too many times over a certain interval, "Unusual Activity". All these anomalies are connected to a specific interface of alarms defined on Ticuro Platform.

The dataset has been generated in a real Smart Home Scenario where 17 people who live alone, in the suburban area of Milan, have been monitored over a period of almost 2 years. The dataset was completely unlabelled making the initial problem defined in a completely unsupervised setting.

In this work, to cope with this problem, a specific processing pipeline has been designed. The event object, opportunely defined according to a vector of features, is firstly filtered through an "Events-filter" structure based on Gas Growing Neural Network clustering technique. The events kept by the filter (recurrent events) are used to calculate a probability density function and the related anomaly score. Through a dynamic assignation, a threshold on the anomaly score is computed in respect to an α parameter manually chosen by the user. The event is considered anomalous if its anomaly score is below such threshold. Beyond the definition of this anomaly detection procedure, in this thesis a set of tests have been designed to assess the confidence of our system.

At first type of tests, "Reliability tests" shows quite good performances in term of True Negative Rate and True Positive Rate in respect to duration and occurrence analyses. Such rates have been evaluated separately considering in the first case (True Negative Rate) recurrent events filtered through filtering process, in second case (True Positive Rate), anomalous events artificially generated.
As a general trend it is possible to state that very good performances are achieved in both the cases. The poorest performances have been obtained for each patient in respect to activity 7, i.e. "Permanence Outside" due to the fact that such activity presents the lowest amount of events with a possible impact on the adaptability of Gas Growing Neural Network as well as on the capacity of extrapolating reliable statistics. Moreover, it has been shown that the α parameter has a direct dependence of the True negative rate enabling the possibility of tuning the system by the user side.

The second type of tests, "Adaptability tests", has demonstrated the capacity of the proposed architecture to adapt to the user's habit in respect to different situations (high duration test, high occurrence test, range duration test, Multi Patient Test).

The software has been implemented in Python language and has been released to the Service Care Provider to be directly applied in a real context. Nonetheless, some other tests should be addressed. A possible future work consists in performing very similar analysis also considering a different data-set generated in a monitored Smart Home Scenario where a basic labeling has been performed. A possible solution in this sense can derive from the datasets cited in the section 2.2 or through a generation of synthetic data through simulator. Beyond being useful to validate our procedure, this can represent a valid solution to tune in a precise manner some hyper-parameters that have not been adjusted in our analysis as, for example, the parameters of the Gas Neural Network.

7 Appendix

7.1 Results for High Duration Analysis

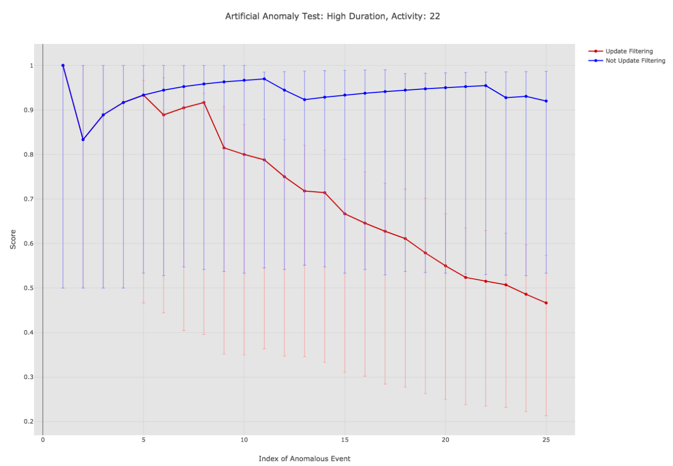


Figure 7.1: Adaptability in respect to "high duration test"; Results for activity $a = 22$ considering all the patients

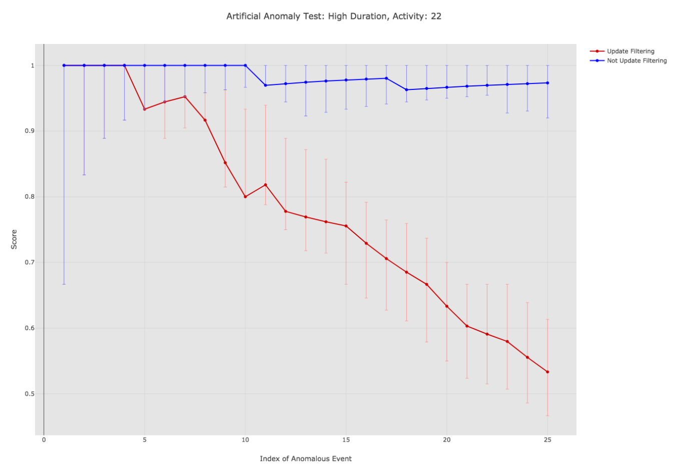


Figure 7.2: Adaptability in respect to "high duration test"; Results for activity $a = 22$ excluding patient 585 and 563

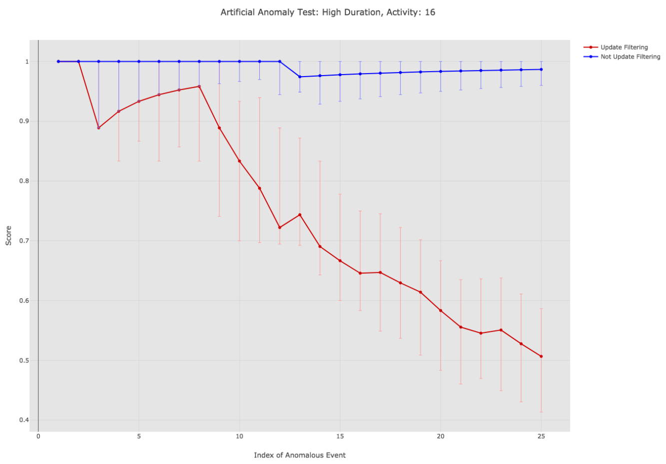


Figure 7.3: Adaptability in respect to ”high duration test”; Results for activity $a = 16$

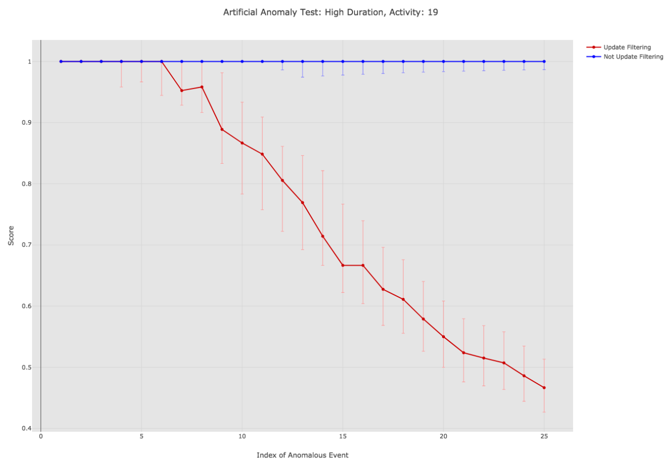


Figure 7.4: Adaptability in respect to ”high duration test”; Results for activity $a = 19$

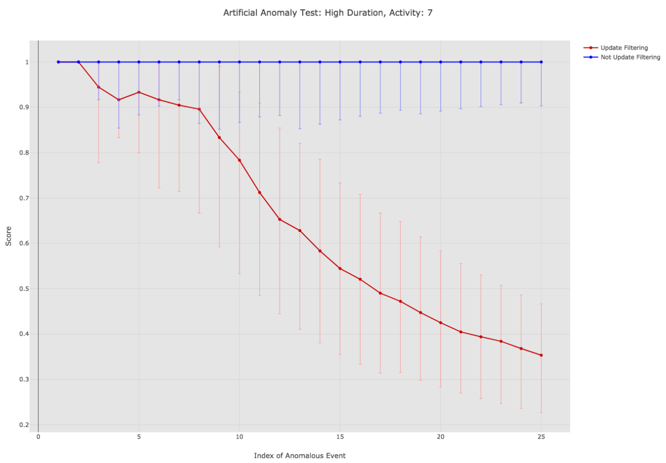


Figure 7.5: Adaptability in respect to ”high duration test”; Results for activity $a = 7$

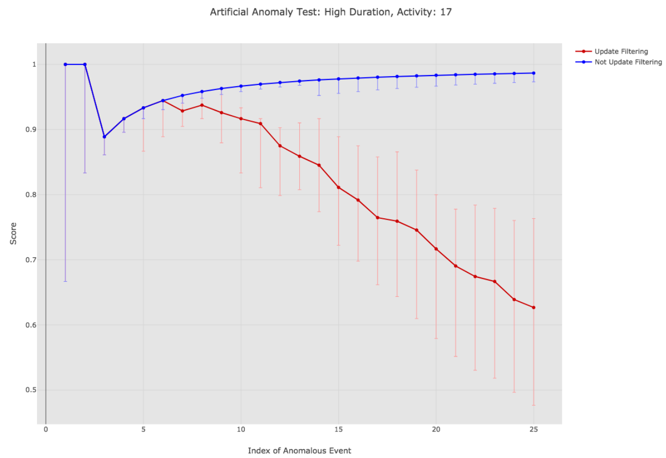


Figure 7.6: Adaptability in respect to ”high duration test”; Results for activity $a = 17$

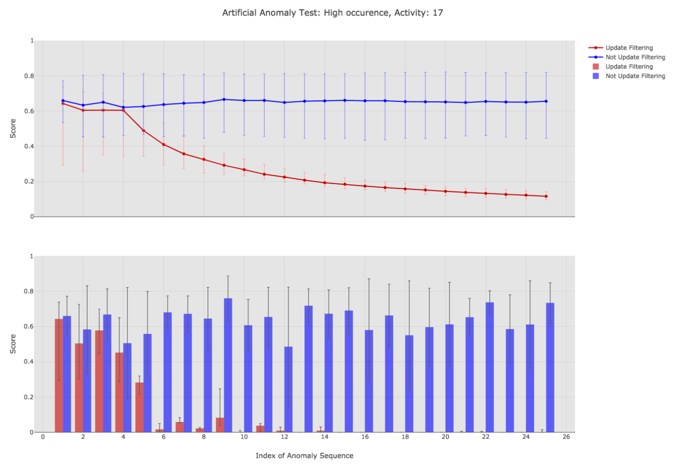


Figure 7.7: Adaptability in respect to "high occurrence test"; Results for activity $a = 17$

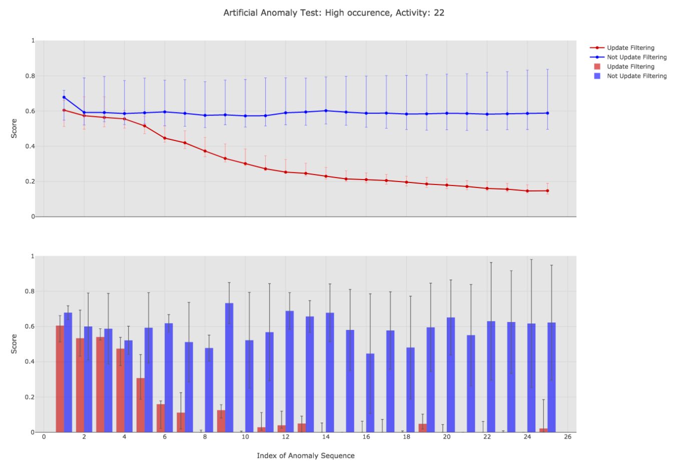


Figure 7.8: Adaptability in respect "high occurrence test"; Results for activity $a = 22$

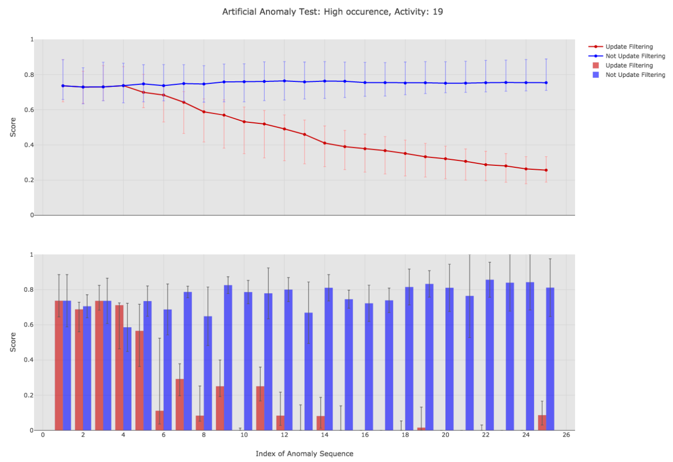


Figure 7.9: Adaptability in respect to "high occurrence test"; Results for activity $a = 19$

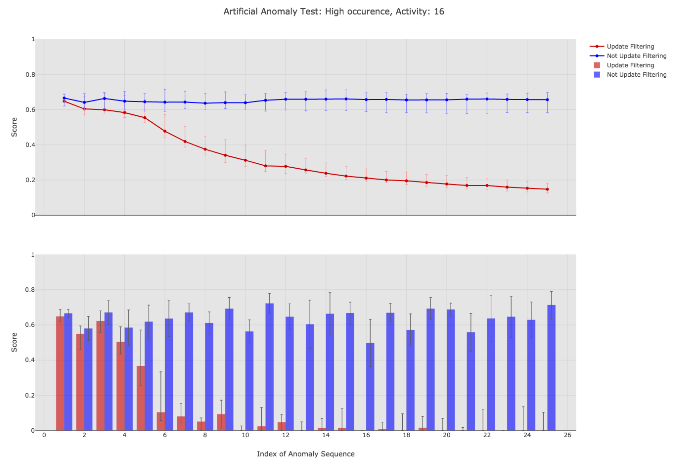


Figure 7.10: Adaptability in respect "high occurrence test"; Results for activity $a = 16$

References

[1] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homespast, present, and future," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1190–1203, Nov 2012.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, pp. 15:1–15:58, 2009.

[3] S. Sennan, "Internet of things based ambient assisted living for elderly people health monitoring," *Research Journal of Pharmacy and Technology*, vol. 11, pp. 1–5, 10 2018.

[4] R. Li, B. Lu, and K. McDonald-Maier, "Cognitive assisted living ambient system: A survey," *Digital Communications and Networks*, vol. 1, 11 2015.

[5] B. Canada. Remote monitoring. [Online]. Available: https://business.bell.ca/shop/enterprise/remote-monitoring-solutions

[6] AT&T. Remote monitoring. [Online]. Available: https://www.att.com/features/

[7] B. Telecom. Remote monitoring. [Online]. Available: https://www.bt.com/broadband/accessories/?s_intcid=userguideshelp_banner_productpages#home-monitoring

[8] H. Reply. Ticuro reply. [Online]. Available: https://www.reply.com/ticuro-reply/en/

[9] N. O. Alshammari, "Anomaly detection using hierarchical temporal memory in smart homes," 2018. [Online]. Available: http://eprints.staffs.ac.uk/4284/

[10] M. Novák, M. Bias, and F. Jakab, "Unobtrusive anomaly detection in presence of elderly in a smart-home environment," in *2012 ELEKTRO*, May 2012, pp. 341–344.

[11] J. Weisenberg, P. Cuddihy, and V. Rajiv, "Augmenting motion sensing to improve detection of periods of unusual inactivity," in *Proceedings of the 2Nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, ser. HealthNet '08. New York, NY, USA: ACM, 2008, pp. 2:1–2:6. [Online]. Available: http://doi.acm.org/10.1145/1515747.1515751

[12] D. J. Cook, G. M. Youngblood, E. O. Heierman, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja, “Mavhome: an agent-based smart home,” *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).*, pp. 521–524, 2003.

[13] J. H. Shin, B. Lee, and K. Suk Park, “Detection of abnormal living patterns for elderly living alone using support vector data description,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 3, pp. 438–448, May 2011.

[14] G. Jain, D. Cook, and V. Jakkula, “Monitoring health by detecting drifts and outliers for a smart environment inhabitant,” 01 2006.

[15] D. J. Cook, M. Youngblood, E. O. Heierman, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja, “Mavhome: an agent-based smart home,” in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).*, March 2003, pp. 521–524.

[16] C. Debes, A. Merentitis, S. Sukhanov, M. Niessen, N. Frangiadakis, and A. Bauer, “Monitoring activities of daily living in smart homes: Understanding human behavior,” *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 81–94, March 2016.

[17] Y. Wei, Y. Lee, and M. S. Young, “A portable ecg signal monitor and analyzer,” in *2008 2nd International Conference on Bioinformatics and Biomedical Engineering*, May 2008, pp. 1336–1338.

[18] A. Bulling, J. Ward, H. Gellersen, and G. Troster, “Eye movement analysis for activity recognition using electrooculography,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, pp. 741 – 753, 05 2011.

[19] L. Atallah, B. Lo, R. Ali, R. King, and G.-Z. Yang, “Real-time activity classification using ambient and wearable sensors,” *Trans. Info. Tech. Biomed.*, vol. 13, no. 6, pp. 1031–1039, Nov. 2009. [Online]. Available: http://dx.doi.org/10.1109/TITB.2009.2028575

[20] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, “Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers,” *Pattern Recognition Letters*, vol. 29, no. 16, pp. 2213 – 2220, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865508002560

[21] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” *SIGKDD Explor. Newsl.*, vol. 12, no. 2, pp. 74–82, Mar. 2011. [Online]. Available: http://doi.acm.org/10.1145/1964897.1964918

[22] and H. Kautz, "Hierarchical recognition of activities of daily living using multi-scale, multi-perspective vision and rfid," in *2008 IET 4th International Conference on Intelligent Environments*, July 2008, pp. 1–4.

[23] T. Kovácsházy and G. Fodor, "New approach to passive infrared motion sensors signal processing for ambient assisted living applications," in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, May 2012, pp. 2521–2525.

[24] T. Hori and Y. Nishida, "Ultrasonic sensors for the elderly and caregivers in a nursing home." 01 2005, pp. 110–115.

[25] S. S. Intille, K. Larson, J. Beaudin, J. Nawyn, E. Munguia Tapia, and P. Kaushik, "A living laboratory for the design and evaluation of ubiquitous computing technologies." 01 2005, pp. 1941–1944.

[26] J.-H. Lim, H. Jang, J. Jang, and S.-J. Park, "Daily activity recognition system for the elderly using pressure sensors," *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2008, pp. 5188–91, 02 2008.

[27] J. Han and B. Bhanu, "Human activity recognition in thermal infrared imagery," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops - Volume 03*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 17–. [Online]. Available: https://doi.org/10.1109/CVPR.2005.469

[28] L. Vuegen, B. V. D. Broeck, P. Karsmakers, H. V. hamme, and B. Vanrumste, "Automatic monitoring of activities of daily living based on real-life acoustic sensor data: a preliminary study," in *SLPAT*, 2013.

[29] S. Majumder, E. Aghayi, M. Noferesti, H. Memarzadeh-Tehran, T. Mondal, Z. Pang, and M. J. Deen, "Smart homes for elderly healthcarerecent advances and research challenges," in *Sensors*, 2017.

[30] S. Helal, J. W. Lee, S. Hossain, E. Kim, H. Hagras, and D. Cook, "Persim - simulator for human activities in pervasive spaces," 07 2011, pp. 192–199.

[31] M. P. Poland, C. D. Nugent, H. Wang, and L. Chen, "Development of a smart home simulator for use as a heuristic tool for management of sensor distribution." *Technology and health care : official journal of the European Society for Engineering and Medicine*, vol. 17 3, pp. 171–82, 2009.

[32] D. Cook, A. Crandall, B. Thomas, and N. Krishnan, "Casas: A smart home in a box," *Computer*, vol. 46, 07 2013.

[33] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "Casas: A smart home in a box," *Computer*, vol. 46, no. 7, pp. 62–69, July 2013.

[34] M. Skubic, G. Alexander, M. Popescu, M. Rantz, and J. Keller, "A smart home application to eldercare: Current status and lessons learned," *Technol. Health Care*, vol. 17, no. 3, pp. 183–201, Aug. 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1605363.1605370

[35] S. S. Intille, K. Larson, E. M. Tapia, J. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson, "Using a live-in laboratory for ubiquitous computing research," in *Pervasive*, 2006.

[36] R. de BER, E. Aarts, P. Markopoulos, and W. Ijsselsteijn, "Ambient intelligence research in homelab: Engineering the user experience," *Electronics Letters - ELECTRON LETT*, 01 2005.

[37] H.-I. Yang, R. Babbitt, J. Wong, and C. Chang, "A framework for service morphing and heterogeneous service discovery in smart environments," vol. 7251, 06 2012, pp. 9–17.

[38] J. Synnott, C. Nugent, and P. Jeffers, "Simulation of smart home activity datasets," *Sensors (Basel, Switzerland)*, vol. 15, pp. 14 162–79, 06 2015.

[39] A. Mendez-Vazquez, S. Helal, and D. J. Cook, "Simulating events to generate synthetic data for pervasive spaces," 2008.

[40] A. Helal, K. Cho, W. Lee, Y. Sung, J. W. Lee, and E. Kim, "3d modeling and simulation of human activities in smart spaces," 09 2012.

[41] J. Lundström, J. Synnott, E. Järpe, and C. D. Nugent, "Smart home simulation using avatar control and probabilistic sampling," in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, March 2015, pp. 336–341.

[42] N. O. Alshammari, "Anomaly detection using hierarchical temporal memory in smart homes," Ph.D. dissertation, Staffordshire University, 2018.

[43] and and, "Support vector machines for anomaly detection," in *2006 6th World Congress on Intelligent Control and Automation*, vol. 1, June 2006, pp. 2594–2598.

[44] V. N. Vapnik, *The Nature Of Statistical Learning Theory*, 01 2013, vol. 6.

[45] A. A. Sebyala, T. Olukemi, and D. L. Sacks, "Active platform security through intrusion detection using naive bayesian network for anomaly detection," 2002.

[46] S. E. Guttormsson, R. J. Marks, M. A. El-Sharkawi, and I. Kerszenbaum, "Elliptical novelty grouping for on-line short-turn detection of excited running rotors," *IEEE Transactions on Energy Conversion*, vol. 14, no. 1, pp. 16–22, March 1999.

[47] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," *SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, May 2000. [Online]. Available: http://doi.acm.org/10.1145/335191.335388

[48] V. Hautamäki, I. Karkkainen, and P. Fränti, "Outlier detection using k-nearest neighbour graph," vol. 3, 09 2004, pp. 430 – 433 Vol.3.

[49] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, pp. 226–231. [Online]. Available: http://dl.acm.org/citation.cfm?id=3001460.3001507

[50] S. Guha, R. Rastogi, and K. Shim, "Rock: A robust clustering algorithm for categorical attributes," *Inf. Syst.*, vol. 25, pp. 345–366, 2000.

[51] L. Ertöz, M. Steinbach, and V. Kumar, *Finding Topics in Collections of Documents: A Shared Nearest Neighbor Approach.* Boston, MA: Springer US, 2004, pp. 83–103. [Online]. Available: https://doi.org/10.1007/978-1-4613-0227-8_3

[52] T. Kohonen, M. R. Schroeder, and T. S. Huang, Eds., *Self-Organizing Maps*, 3rd ed. Berlin, Heidelberg: Springer-Verlag, 2001.

[53] R. Smith, A. Bivens, M. Embrechts, C. Palagiri, and B. Szymanski, "Clustering approaches for anomaly based intrusion detection," *Proceedings of Intelligent Engineering Systems Through Artificial Neural Networks*, pp. 579–584, 01 2002.

[54] B. Fritzke, "A growing neural gas network learns topologies," in *Proceedings of the 7th International Conference on Neural Information Processing Systems*, ser. NIPS'94. Cambridge, MA, USA: MIT Press, 1994, pp. 625–632. [Online]. Available: http://dl.acm.org/citation.cfm?id=2998687.2998765

[55] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recogn. Lett.*, vol. 24, no. 9-10, pp. 1641–1650, Jun. 2003. [Online]. Available: http://dx.doi.org/10.1016/S0167-8655(03)00003-5

[56] P. Sun and S. Chawla, "On local spatial outliers," *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pp. 209–216, 2004.

[57] R. J. Beckman and R. D. Cook, "Outlier . s," *Technometrics*, vol. 25, no. 2, pp. 119–149, 1983. [Online]. Available: https://doi.org/10.1080/00401706.1983.10487840

[58] A. J. Fox, "Outliers in time series," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 34, no. 3, pp. 350–363, 1972. [Online]. Available: http://www.jstor.org/stable/2985071

[59] D.-Y. Yeung and C. Chow, "Parzen-window network intrusion detectors," 05 2002.

[60] C. Bishop, "Novelty detection and neural network validation," *Vision, Image and Signal Processing, IEE Proceedings* -, vol. 141, pp. 217 – 222, 09 1994.

[61] U. A. B. U. A. Bakar, H. Ghayvat, S. F. Hasanm, and S. C. Mukhopadhyay, *Activity and Anomaly Detection in Smart Home: A Survey*. Cham: Springer International Publishing, 2016, pp. 191–220. [Online]. Available: https://doi.org/10.1007/978-3-319-21671-3_9

[62] T. van Kasteren, A. Noulas, G. Englebienne, and B. Krose, "Accurate activity recognition in a home setting," 01 2008, pp. 1–9.

[63] A. Fleury, M. Vacher, and N. Noury, "Svm-based multimodal classification of activities of daily living in health smart homes: Sensors, algorithms, and first experimental results," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 274–283, March 2010.

[64] F. Rivera-Illingworth, V. Callaghan, and H. Hagras, "A connectionist embedded agent approach for abnormal behaviour detection in intelligent health care environments," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, vol. 4, Oct 2004, pp. 3565–3570 vol.4.

[65] and and, "Detecting and predicting of abnormal behavior using hierarchical markov model in smart home network," in *2010 IEEE 17Th International Conference on Industrial Engineering and Engineering Management*, Oct 2010, pp. 410–414.

[66] K. Bing-Yung Wong, T. Zhang, and H. Aghajan, *Data Fusion with a Dense Sensor Network for Anomaly Detection in Smart Homes*, 11 2014, pp. 211–237.

[67] O. Aran, D. Sanchez-Cortes, T.-M.-T. Do, and D. Gatica-Perez, "Anomaly detection in elderly daily behavior in ambient sensing environments," *Human Behavior Understanding*, vol. 9997, pp. 17. 51–67, 2016. [Online]. Available: http://infoscience.epfl.ch/record/221398

[68] A. Aztiria, G. Farhadi, and H. Aghajan, "User behavior shift detection in intelligent environments," in *Ambient Assisted Living and Home Care*, J. Bravo, R. Hervás, and M. Rodríguez, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 90–97.

[69] I. Armac and D. Retkowitz, "Simulation of smart environments," in *IEEE International Conference on Pervasive Services*, July 2007, pp. 322–331.

[70] M. Hooshmand, D. Zordan, T. Melodia, and M. Rossi, "Surf: Subject-adaptive unsupervised ecg signal compression for wearable fitness monitors," *IEEE Access*, vol. 5, pp. 19 517–19 535, 2017.

[71] "Modular toolkit for data processing mdp," http://mdp-toolkit.sourceforge.net/api/index.html.

[72] A. Zambom and R. Dias, "A review of kernel density estimation with applications to econometrics," vol. 5, 12 2012.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Michele Rossi for the continuous support, motivation, and knowledge.
Besides my advisor, I would like to thank to my co-advisor Dr. Matteo Gadaletta who provided me an important technical support, using his expertise to guide me during all the steps of this project.
My sincere thanks also goes to Healthy Reply particularly, to Dr. Nicola Bottone, who gives me the opportunity to join their team as intern.
Finally, I must express my very profound gratitude to my parents Maria Angela and Diego and to my girlfriend Michela for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Thank you,

Matteo