NOAH AAL Project
Documentation

Deliverable 1.3

Behavioral Analysis Report

Lead Partner	UNIPR
Authors	Niccolò Mora
Contributors:	G. Debard
Date:	21-12-2017
Revision	1.0
Dissemination Level	Public

Project Acronym:	NOAH
Project full title:	NOAH Not Alone At Home
AAL Project Number:	AAL-2015-2-115
With Support of:	

Noah Project

Tel. +39 0521905828
info@noahproject.eu

University of Parma
Parco Area Delle Scienze
181/a 40124 Parma, ITALY

www.noahproject.eu

Introduction

The Behavioral Analysis (BA) module takes care of extracting meaningful insights from raw data gathered by the field sensors and stored in the relative DataBase (DB). Figure 1 below illustrates how the BA module fits within the Noah cloud ecosystem.

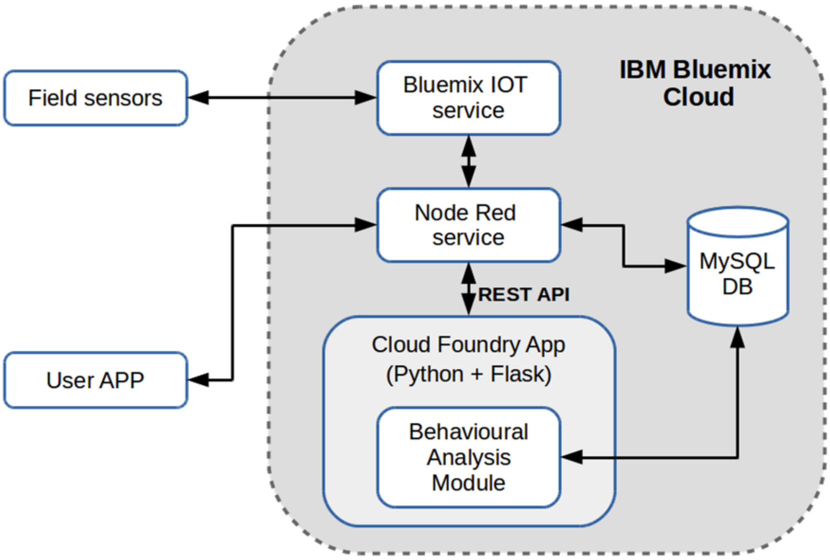

Fig. 1: NOAH Cloud Architecture - high level view.

The data stemming from field sensors are ingested into the cloud by a Bluemix IOT service. Meanwhile, a Node Red module takes care of parsing such data, implementing simple rule-based actions, and, eventually, it inserts them into a MySQL DB for further processing. The IOT service and the Node Red module are tightly coupled to implement those functionalities. On the other hand, the BA module is conceived as a web service, which exposes an API (Application Programming Interface) to serve different analytics requests. Design is modular and such a decoupling allows better organization and maintainability. The API is based on a REST architecture, and requests are served over secure HTTPS connections.

Overall, the BA module performs the following tasks:

- Request parsing. Each request is parsed to extract the type of analysis to perform, along with the associated parameters.
- Data retrieval from the MySQL database
- Data cleaning and pre-processing. Raw data are inspected to remove potential errors and transformed to a richer representation (sensor event matching and filtering), better suited for further processing.
- Processing. The transformed data are exploited to infer relevant information and, possibly, make prediction on future states

- Response serving through HTTPS protocol, using JSON encoding.

BA module implementation notes

The BA module is implemented exploiting the Python programming language and is split into two main components:

- Web-service management. This part handles incoming requests, parse them to determine the kind of analysis to perform and serves results. It is based on the Bluemix Cloud Foundry App container for Python and the Flask web framework, an open-source package for easy and modular web service design. In order to access the analytics services, users must supply a username and password, in order to protect possibly relevant data. Requests are only accepted through a secure HTTPS link.
- Analytics engine. This part implements all the analytics components to extract meaningful information from the data. It exploits popular data-science oriented packages, including P*andas, Numpy, Scipy, Scikit-learn, Statsmodels*.

Home sensor kit overview

Data collected from smart home sensors offer a comprehensive set of information which can be exploited in many different ways: from the monitoring of very specific actions (such as user's sleeping patterns) up to building models of a *typical resident's day.*

Three major types of sensors are usually exploited to gather information about home residents' routines:

- *motion* sensors (mainly Passive InfraRed, PIR), which carry information about motion occurring within their sensing range.
- *interaction* sensors, which require the user to specifically interact with them. Examples are:
    - magnetic switches for checking on doors, windows, cabinets
    - presence sensors, such as pressure pads for detecting bed/chair occupancy
    - energy meters for keeping track of an appliance's use
    - accelerometer-based units for detecting when a person moves them

- *environmental* sensors, such as temperature, humidity, light, which can provide further context to the data.

Each one carries different information and, in general, the first type is most commonly encountered in literature, being the less obtrusive. However, the information they are able to convey is much coarser than that of the second type of sensors. In fact, interaction sensors, are much more specific and related to a given activity, but, on the other side, a larger amount of such sensors must be deployed to address different activity monitoring analyses.

The amount of data being generated by the home obviously depends on the number of sensors, as well as their type (motion sensors are, indeed, likely to be triggered much more than an *interaction sensor*, e.g. a bed sensor), and generally up to some thousand events per day can be generated, in a typical scenario. However, having such a volume of data does not guarantee to be able to extract and monitor each user's behavior precisely. In fact, while some target behavior may be quite specific and easy to monitor with a few sensors (e.g. toilet use), many other routines can vary so much in terms of duration, sequence of activated sensors and time of the day that modeling each possible scenario would require an intractably large amount of data.

The NOAH home sensor kit features the following sensing devices:

- Bed presence sensor
- Chair presence sensor
- Toilet sensor
- PIR motion sensor
- Magnetic contact (e.g. for medicine cabinet or front door)

Data Analytics services

The primary purpose of all analytics services developed for the NOAH project is to model the usual user's behaviors and to detect new or deviating observations. In this sense, a great emphasis is laid on *explanatory models*: their purpose is to model behaviors as originating from given factors, which are relevant to the understanding of patterns.

The analytics services feature three main

- *Regression analyses* and *outliers detection*. The aim of such investigations is to spot longitudinal trends, accounting for differences between week days and week-ends. These models also allow to detect data that do not fit well the general observed trend, and the system is able to flag them as anomalous, possibly triggering further investigations.
- *Activity curves*. Activity curves model the average user-sensor interaction throughout the day. Such curves may be used to detect changes in patterns of use, by comparing different periods and testing for statistically significant deviations. Another application is as features to assess similarity between users' patterns.
- *Activity curves clusters*. The activity curves can be further investigated to extract different patterns which occur sparsely through days. For example, a bed sensor may exhibit different behaviors during week days or weekends. Clustering operators may automatically extract those information, which may be missed by the "average" behavior as detected by the activity curves.

All models can be automatically applied to each sensor type, as detailed before. The analytics pipeline handles all steps of data retrieval and cleaning. This is particularly important, since spurious or missing activations are likely to occur. Exceptions and errors during model computation are handled carefully to prevent service blocking.

As a final note, multivariate extensions, attempting to jointly model multiple sensors, are being investigated.

Analytics by example: real-world data case study

This section provides an example of analytics capabilities, based on data collected by UniPR from a real-life pilot. Such data are not part of NOAH pilot studies; they share, however, most of the inclusion criteria, and are therefore suitable to present NOAH analytics services. The user is a 75-year old male, living in a sheltered home. His apartment consists in a single bedroom and a toilet, whereas the common spaces (e.g. dining room and hall) are shared with other residents.

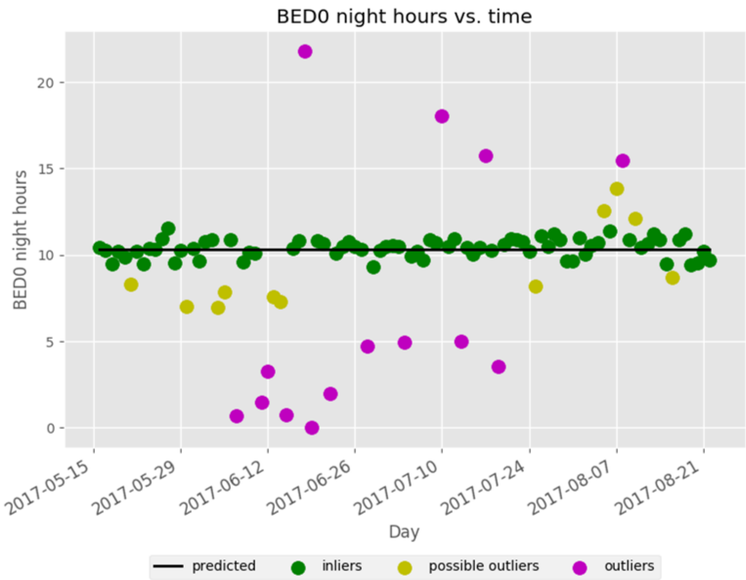

Fig. 2: nightly bed usage plot. The explanatory linear model does not detect any significant linear trend, nor the effect of week days vs. weekends.

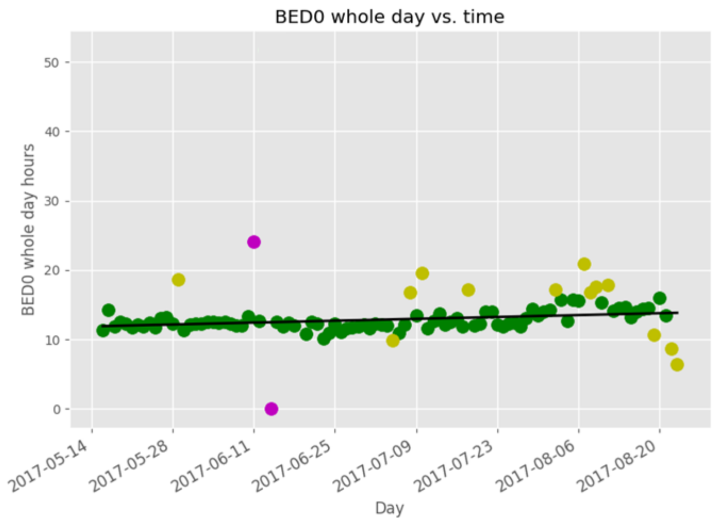

Fig. 3: bed usage through the whole day. A linearly increasing trend of approximately 1h 45min is detected.

. We begin by observing some historic trends on bed usage during evening and night hours (19:00 - 09:00), as shown in Fig. 2.

A robust linear model is fitted to data, to discover linearly increasing trends, as well as to assess the impact of weekend vs week days (the model simply consists in two features and a bias term). The result is that night presence in bed is quite stable, approximately 10h20' per day, and no trends or weekend effects are noticeable. Only a few points were detected as gross outliers (purple points in the scatter plot), and some "unusually distant" points were observed (marked as yellow points). Values above the observed window (14 hours, from 19:00 to 09:00) refer to intensive and uninterrupted bed use, extending over the window. Such observations are quite rare.

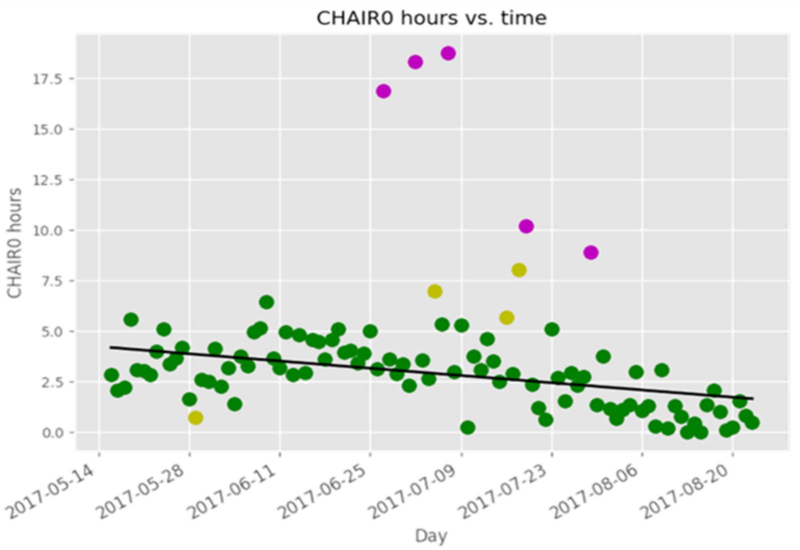

Fig. 4: Chair usage through whole day. A linearly decreasing trend is noticeable, opposite to the trend observed for the bed

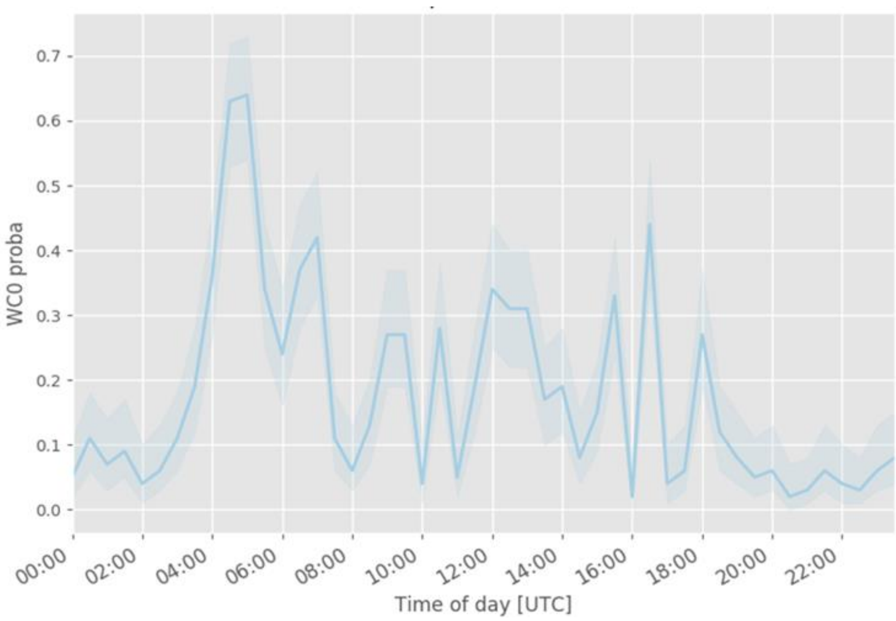

Fig. 5: Activity curves represent the expected average user-sensor interaction throughout the day. In this case, the activity curve of a toilet sensor is reported

Bed usage through the whole day, instead, shows a linearly increasing trend, as reported in Fig.3. Overall, the person spends 13h45’ a day in bed, with a linearly increasing trend of 1h45’ during the observed period.

Still, no difference was noted between weekend and week days. Daily bed presence is more stable than night presence (less outliers or potential ones), suggesting an overall adjustment through the day in case the evening-night period was not satisfactory to the user.

Interestingly, the increase in bed presence throughout the day can be explained by an opposite decreasing trend in chair usage, approximately -2h30’ over the period. Average presence is around 2h45’ per day. Some gross outliers are flagged, probably due to a heavy bag or item being left over the chair. Fig. 4 summarizes such findings.

The same information may be looked from a different perspective, using sensors’ characteristic profiles. Such curves represent the likelihood of the user interacting with the sensor throughout the day.

An example of such curve is reported in Fig. 5 for a toilet presence sensor: solid lines represent the average expected pattern, shaded areas the confidence intervals.

These curves are particularly interesting in order to detect changes in patterns over time: A comparative analysis can be made between different time periods, and statistical tests can be performed to spot significant changes.

Two such examples are given below in Fig.6 and 7, for the bed and chair sensors, respectively. Red curves represent the initial reference period (first 20 days), whereas blue lines represent patterns from more recent period (last 20 days).

The changes are significant around noon, and such changes (increasing bed presence, decreasing chair presence) are in line with the results presented above on historic trends.

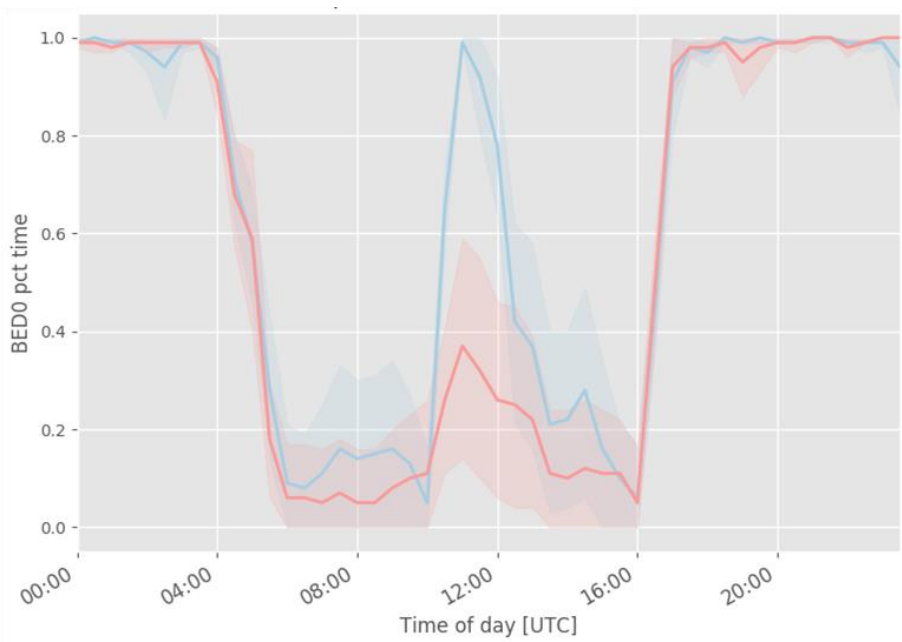

Fig. 6: Comparison of bed activity curves between different windows of 20 days each. A statistically significant deviation is detected around noon, increasing use in the last days. This is in agreement with the linear models.

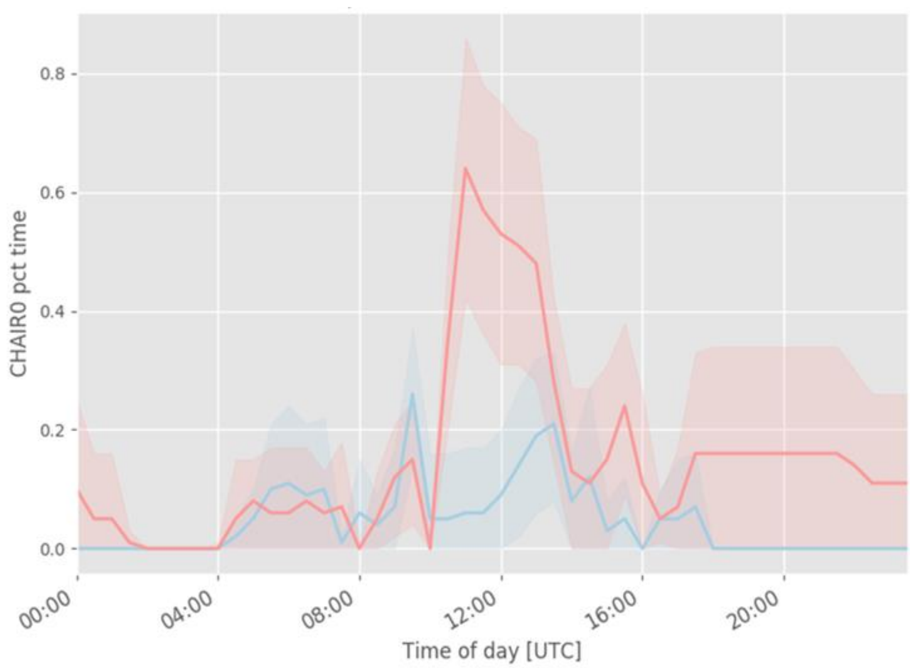

Fig. 7: Activity curves comparison for the chair sensor, during the same observation windows.

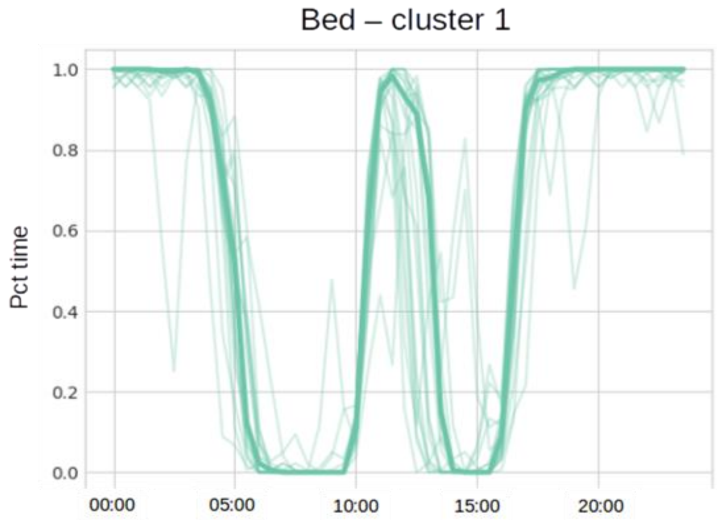

Fig. 8: Clustering of bed traces (first cluster). The bold solid line represents the median, light lines the single day traces

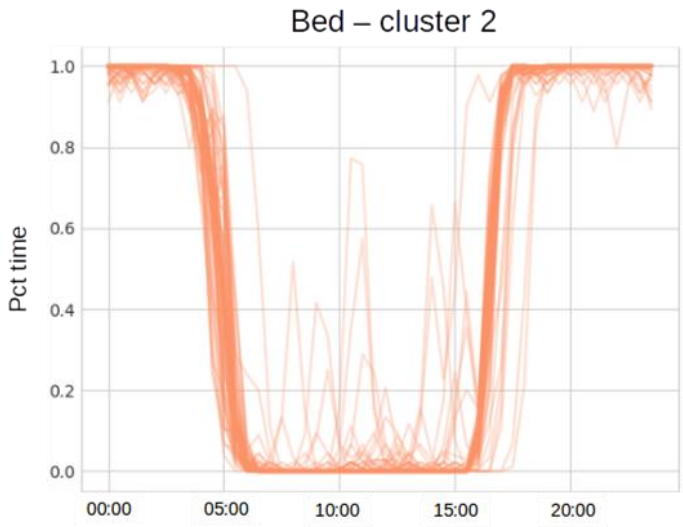

Fig. 9: Bed traces clustering (second cluster)

The activity curves are quite useful to detect mean changes in sensor profiles. However, they might miss some characteristic patterns: for example, there might be a day in the week during which the patterns are regularly different from others. Automatic discovery of such patterns is useful to gather insights in user's behavior. This can be accomplished thanks to clustering, a machine learning technique which aims at finding similarities within groups of data.

In particular, daily sensor traces are grouped together according to a suitable similarity metrics between waveforms. A specific algorithm was then designed to automatically determine the optimal number of clusters, i.e. different user behaviors in the observed data. Therefore, by looking at the gathered data, the algorithm is able to determine which are the most common patterns in the data: this is shown in Fig. 8 and 9 for the bed sensor, where two different clusters are found.

Taking the median allows to model cluster prototypes, representative of underlying users' behaviors. The algorithm is also able to identify patterns which are far from the extracted cluster prototypes, i.e. potentially anomalous. Such situation is shown in Fig. 10

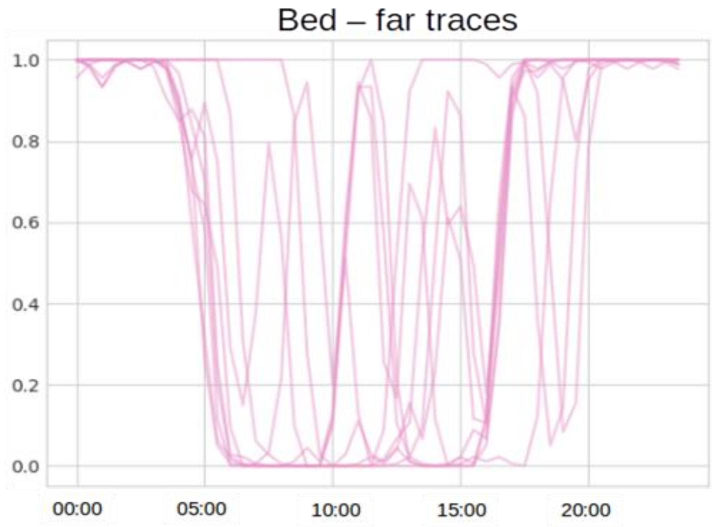

Fig. 10: Once cluster prototypes are extracted, it is possible to identify traces which are quite different from all others ("far traces")

Conclusions

This document presented the data-analytics services of the NOAH platform. The system gathers data from many different sources, including

Bed presence sensor

Chair presence sensor

Toilet sensor

PIR motion sensor

Magnetic contact (e.g. for medicine cabinet)

Such data are then used to provide quantitative information about:

Trend analysis & anomaly detection (regression analysis)

Routine/habits modeling and deviation assessment (activity curves)

presence of multiple different behaviors (activity curves clustering)

The services are implemented using the Bluemix cloud-computing frameworks, from data ingestion up to device management and advanced data mining. Services are resilient and can handle common errors which happens in real-world data, such as missing data, incorrect readings and so on. All analytics services are exposed as web-services over secure HTTPS links, and require a username and password to be served.

Finally, a real-case scenario was finally described as case study to showcase the analytics platform capabilities. Such services are able to produce summarized information (e.g. trend detection, incipient behavior change in sensor traces), which can be further refined before being presented to the end user. Furthermore, depending on the type of end-user, such information can be customized: for example, a formal care giver may find useful being notified by the system about incipient trends or on change in specific sensor patterns (e.g. bed/toilet), whereas the elderly person may find more useful to have just a simple indicator that everything is going all right or whether there are some problems with a specific behavior. Such information can be easily provided through the customized end-user app.