Ref. Ares(2024)4472079 - 20/06/2024

SMARTBEAR

CALL H2020-SC1-FA-DTS-2018-2020

Trusted digital solutions and Cybersecurity in Health and Care

TOPIC DT-TDS-01-2019

Smart and healthy living at home

SMART BEAR

"Smart Big Data Platform to Offer Evidence-based Personalised Support for Healthy and Independent Living at Home"

D4.6 (D18) - Report on SMART BEAR Cloud Enabling Components v3

Due date of deliverable: 2024-04-30
Actual submission date: 2024-06-20

Grant agreement number: 857172

Start date of project: 01/09/2019

Lead contractor: CNR

Duration: 66 months

Project funded by the European Commission within the EU Framework Programme for Research and Innovation HORIZON 2020	
Dissemination Level	
PU = Public, fully open, e.g. web	P
CO = Confidential, restricted under conditions set out in Model Grant Agreement	
CI = Classified, information as referred to in Commission Decision 2001/844/EC.	
Int = Internal Working Document	

D4.6 (D18) - Report on SMART BEAR Cloud Enabling Components v3

Editors
Anca Bucur (Philips)
Tim Hulsen (Philips)

Contributors
Mario Sicuranza (CNR)
Daniel Foulkes (Philips)
Erno Ledder (Philips)
Anca Bucur (Philips)
Gurnoor Singh (Philips)
Tim Hulsen (Philips)
Valerio Bellandi (UMIL)
Ernesto Damiani (UMIL)
Jonatan Maggesi (UMIL)
Nicola Paganotti (UMIL)
Nello Scarabottolo (UMIL)
Christos Kloukinas (CITY)
Qiqi Su (CITY)
Eleftheria Vellidou (NKUA)
Dimitrios G. Boucharas
Evita Karapintzou
Marina Georgoula

Reviewers
Paolo Ceravolo (UMIL)
Mattia Occhipinti (UMIL)
Razvan Trascu (ANA)

The work described in this document has been conducted within the project SMART BEAR, started in September 2019. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857172

ãCopyright by the SMART BEAR Consortium.

Executive Summary

This document reports on the third (final) version of the components of the SMART BEAR backend Cloud platform. The deliverable documents the backend components and the analytics, together with planned work of WP4 focused on the integration of the platform, the further development of the key components, and on the implementation and evaluation of the analytics approaches adopted by the project.

The main components of the SMART BEAR cloud platform include the database and its underlying information model, the clinical repository interfaces, the Big Data Engine, the synthetic data generation component, and the analytics. The analytics and personalization components leverage the SMART BEAR FHIR-based Information Model. This final version reports on the work carried out since the release of D4.4.

Contents

1 Introduction 8
1.1 Overview 8
1.2 Goals of the deliverable 8
2 SMART BEAR Cloud Enabling Components 9
2.1 Data Model Specification compliant with FHIR 9
2.1.1 Introduction 9
2.1.2 Data model mapping 11
2.2 BDA Engine 18
2.2.1 BDA components 18
2.2.2 Data storage and data management implemented by BDA 19
2.2.3 Data analysis task 20
2.2.4 Data visualization 20
2.2.5 Platform management 20
2.3 Analytics workflow 21
2.3.1 Data Preparation 21
2.3.2 Data Quality Workflow 22
2.3.3 Workflow for designing a prognostic model 24
2.3.4 Data augmentation with generative models 25
2.3.5 Intrinsic Capacity 29
2.4 Analytics for Hearing Loss 32
2.5 Federated Learning Framework 33
2.6 AI models and interventions in the CVD scenario 34
2.6.1 Models 34
2.6.2 Applications of these models to support specific use cases in SMART BEAR 35
2.6.3 Data 36
2.7 Balance and Cognitive Analytics 38
2.8 Dashboard 38
2.8.1 Staging version 38
2.8.2 Baseline Assessment 38
2.8.3 Adverse events 39
3 Concluding Remarks 42
4 References 43
5 Annexes 46
5.1 Annex 1: Dashboard screenshots 46
5.1.1 Registration and login 46
5.1.2 Patient's registration 46
5.1.3 Baseline Assessment 49

5.1.4 Patient management 51
5.1.5 Tech support 53
5.2 Annex 2: Holobalance Analytics 55
5.2.1 Motivations and Objectives in Expanding Holobalance Solution 55
5.2.2 The Holobalance approach in SMART BEAR 56
5.2.3 Clinical Relevance 83
5.2.4 SMART BEAR Platform Integration 83
5.2.5 Conclusion 83
5.2.6 References 84

List of Figures

Figure 1. An example of SUSHI results....10
Figure 2. FHIR Resource Design for Hearing Aid Monitoring....12
Figure 3 SMART BEAR FHIR profiles....14
Figure 4 Differential Table for Hearing Aid Device....15
Figure 5 Differential Table for HA Observation....17
Figure 6 Differential Table....17
Figure 7. BDA Components....19
Figure 8 Excerpt of raw data in the BDA....22
Figure 9. Data quality workflow....24
Figure 10. Flow for designing a prognostic model....25
Figure 11: Implementation of CTGAN....26
Figure 12: Web interface....27
Figure 13: Synthetic data graphs generated....28
Figure 14: Predictive tables....28
Figure 15. Primary view to the patient data: From single measurement to calculated Intrinsic Capacity....31
Figure 16. Left: Calculated IC for a timeframe of study the status of a patient; Right: measured IC coverage for the selected patient....31
Figure 17. Visualizing the specific domain evolution; from the right panel it is possible to select the domain to visualize....32
Figure 18. Figure 4: The composition of IC and the contribution of each domain in the final IC score....32
Figure 19 LSTM Cell Architecture....35
Figure 20 Walking distance in the dataset....37
Figure 21 Aggregated heart rate values....37
Figure 22. Medical History....39
Figure 23. Adverse events button....40
Figure 24. Adverse events tab....40
Figure 25. Suspect entity management....41
Figure 26. The login form....46
Figure 27. The home page....47
Figure 28. Create patient form....47
Figure 29. Exclusion criteria....48
Figure 30. Dexterity questionnaire....48
Figure 31. Exclusion criteria are matched....49
Figure 32. Risk of exclusion alert....49
Figure 33. Exclusion criteria are matched....50
Figure 34. Profiling shortcut....50
Figure 35. Patient profiling....50
Figure 36. Edit a patient....51
Figure 37. Change the patient's status....52

Figure 38. ID management 52
Figure 39. Devices management 53
Figure 40. The Tech Support page 53
Figure 41. Tech Support. Press the SHOW button 54
Figure 42. Tech Support. The Overview tab 54
Figure 43: The HOLOBALANCE prediction workflow diagram 56
Figure 44: Patients' sessions distribution. 60
Figure 45: Total conducted exercises distribution across all patients 60
Figure 46: Questionnaires completion dates distribution. 61
Figure 47: Observation and prediction window. 72
Figure 48: Exercise IDs against timepoints / sessions. 73
Figure 49: Distribution of final score (outcome variable). 73
Figure 50: AI models performance with past horizon equal to 3 sessions 74
Figure 51: AI models performance with past horizon equal to 4 sessions 74
Figure 52: AI models performance with past horizon equal to 5 sessions 75
Figure 53: AI models performance with past horizon equal to 6 sessions 75
Figure 54: Top 10 most important features with past horizon equal to three sessions. 77
Figure 55: Top 10 most important features with past horizon equal to four sessions. 77
Figure 56: Top 10 most important features with past horizon equal to five sessions 78
Figure 57: Top 10 most important features with past horizon equal to six sessions. 78
Figure 58: Distribution of total score across cognitive exercises 79
Figure 59. Cognitive exercises performance per patient ID. 79
Figure 60. Sessions distribution for performing cognitive exercises. 80
Figure 61: Feature importance for 'holobalance_cognitive_s3_animal_feeding' exercise. 81

List of Tables

Table 1. Hearing Aid Properties 12
Table 2. Device resource data elements 13
Table 3. Observation resource data 13
Table 4. Possible codes of Codeable Concept 15
Table 5: Comparison of performance of models for predicting the Depression state 27
Table 6. Intrinsic Capacity variables 30
Table 7: Exercise data acquisition sensors. 59
Table 8. Data description per exercise 62
Table 9: Extracted features from original dataset 70
Table 10: Balance exercises models performance. 76
Table 11: Cognitive exercises models performance. 80
Table 12: Hyperparameter tuning details. 82

1 Introduction

1.1 Overview

Deliverable 4.6 Report on SMART BEAR Cloud Enabling Components v3 reports on the final version of the components that compose the SMART BEAR backend Cloud platform. This deliverable documents the infrastructure and analytics development activities part of WP4, which focuses on delivering all the components needed for the SMART BEAR Cloud infrastructure, including the data repository, the analytics models, the decision support system, the big data approaches, the developed interventions and their evaluation, and the visualisation.

The main components of the SMART BEAR cloud platform include the database and its underlying information model, the clinical repository interfaces, the Big Data Engine, the synthetic data generation component, and the analytics.

1.2 Goals of the deliverable

This deliverable provides the technical specification of the SMART BEAR cloud enabling components, describing their design, interactions, interfaces, and functionality. A core topic of this deliverable is the solution to data representation and management, including a full specification of the FHIR-compliant information model for clinical data, the description of the full data model (including clinical and non-clinical data) and of the database implementation.

We present approaches selected for the implementation of AI and analytics to support the development of effective interventions that will be deployed and evaluated in the SMART BEAR pilots. Each analytics approach is presented in the context of a specific clinical scenario, describing the targeted interventions, the process of developing and validating the model, and the used data.

2 SMART BEAR Cloud Enabling Components

2.1 Data Model Specification compliant with FHIR

2.1.1 Introduction

2.1.1.1 Overview on FHIR and available tools

FHIR is a standard created by HL7 International for exchanging digital health information by exploiting the best characteristics of HL7 Version 2 and Version 3, although it is not compatible with them. The standard is spreading rapidly in the international context and the most important international organizations that provide solutions to specific problems in healthcare, like Integrating the Healthcare Enterprise (IHE) and Personal Connected Health Alliance (PCHA), are updating their technical specifications to include FHIR.

FHIR defines the key health entities as resources, which are a collection of information models specifying data elements, constraints and relationships for the "business objects". The FHIR specification Release 4, version 4.01, which is based on the FHIR Composition Framework that includes 145 different resource types covering many clinical and administrative concepts of the healthcare sector. Each resource type, like *Patient*, *Observation*, and *Device*, permits to represent a number of properties related to a specific concept. In order to facilitate a homogeneous representation of the data into these resources, the FHIR specification provides a concrete formalization of them in three different data formats: eXtensible Markup Language (XML), JavaScript Notation Object (JSON), and Turtle (TTL). In FHIR, many elements are used to memorize terminologies represented by coded values, like the element *code* of the *Observation* resource. Some of these codes are directly defined by the FHIR specification, while other codes may be defined elsewhere, like: external terminology systems.

Another important component of the FHIR specification is represented by the RESTful Application Programming Interfaces (APIs), which are a collection of well-defined interfaces for making different applications able to interoperate. Their use is useful to use shared communication protocols via Hypertext Transfer Protocol (HTTP), even if not mandatory.

The FHIR specification is a "platform specification", as it describes a large number of base resources, frameworks and APIs. However, in order to implement a FHIR-based solution for a specific subdomain of the healthcare domain, one needs to take into account different regulations, requirements, etc. Therefore, the FHIR specification requires further adaptations to the particular context. These adaptations include a specification of the particular standard resource elements that will be used, possible additional elements, the APIs and terminologies to use, etc. Several FHIR artefacts may be used for adapting the specification to a specific context: i) Implementation Guide (IG), which is a set of rules well documented that specify how FHIR resources have to be used to solve a particular problem; ii) Package, which is a group of related adaptations published within an IG; iii) Conformance Resource, which is a specific FHIR resource in a package that represents the adaptations in a computable way.; iv) Profile, which is a set of constraints and extensions on a single resource.

A number of frameworks and tools exist for facilitating the implementation and adaptation of solutions in compliance with the FHIR standard.

Simplifier.net is a web platform for publishing, collaborating and developing the FHIR specification, offering capabilities for resource management and team collaboration. The platform provides a multitude of available FHIR resources organized in projects, consisting of profiles, extensions, value sets, dictionaries, mappings, examples and more. Forge can be used for creating FHIR profiles through a Windows desktop application, in an integrated way with Simplifier.net.

ART-DÉCOR is an open source tools suite that supports the creation and maintenance of standard HL7 templates, value sets, scenarios, and datasets. It therefore represents both a methodology for the creation of standard artefacts and a set of tools to simplify the operations of managing artefacts. The tool also includes the management of the templates and value sets through cloud-based federated Building Block Repositories and

therefore allows to simplify the entire phase of use and reuse of published templates and value sets. ART-DECOR supports collaboration between members of the same team or between different teams through governance operations.

FHIR Shorthand (FSH) is a language specified by HL7 for defining the contents of the FHIR IGs. It permits to express resource profiles, value sets, examples, etc. using any text editor, enabling thus distributed, team-based development using source code control tools such as GitHub. It is based on a formal grammar defined in the ATLR parser generator. SUSHI Unshortens ShortHand Inputs (SUSHI) is a FSH compiler, which convers scripts written according to FSH language to FHIR artefacts. As shown in Figure 1, SUSHI permits to compile and produce artefacts concerning profiles, extensions, value sets, code systems and instances.

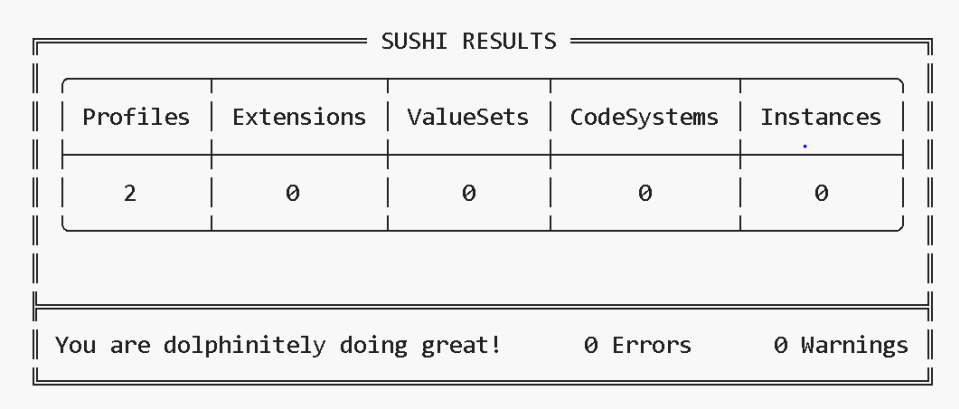

Figure 1. An example of SUSHI results

IG Publisher is a tool provided by the FHIR team for translating the IG content in: i) generated resources that can be included into the published guide (XML, JSON, TTL formats); ii) HTML files; iii) zip files used by implementers. The tool is used after the compilation of the FSH scripts by the SUSHI compiler.

ART-DÉCOR and FSH are free, whereas Simplifier.net has both free and paid profiles and Forge is free only for educational purposes.

2.1.1.2 Methodology used in the SMART BEAR project

The nature of the data treated in the project, in accordance with the rules of the FHIR standard, led to the necessity of using a specific IG. For this reason, an analysis of the IGs published on the FHIR registries was carried out. Among these, particular attention was dedicated to the Personal Health Device (PHD) and International Patient Summary (IPS) IGs.

The PHD IG adapts FHIR resources to convey measurements and supporting data from PHDs to different kind of systems, like platforms for electronic medical records, clinical decision support, etc. The interest for this IG was captured considering that it is based on the Continua Design Guidelines and upon the ISO/IEEE 11073 PHD Domain Information Model (DIM). Anyway, considering that in SMART BEAR many health data are not acquired by PHDs, but by collecting patients' answers to specific questionnaires, this IG was not considered adequate for the SMART BEAR project.

The IPS IG defines the rules for the production of a document containing the essential healthcare information about a subject of care, designed for supporting unplanned, cross border care, although it is not limited to it. Although this IG provides an important contribution to identify a minimal, specialty-agnostic, condition-independent, clinically relevant dataset for a patient, it was not considered relevant for the SMART BEAR project.

For these reasons, in compliance with the FHIR standard and in line with the choices adopted in many European projects, the approach taken for the definition of the information model for the project is to define a dedicated

SMART BEAR IG by profiling a set of identified FHIR resources and individuating the terminologies from international standard code systems as well as internal value sets. The tools chosen for modelling the FHIR information model are FHIR Shorthand, SUSHI and IG publisher, considering that they are the official tools provided by HL7.

Starting from Information model we have defined and published a dedicated Smart Bear FHIR Implementation Guide, using the HL7 tools.

In particular, the tools used were:

- FHIR Shorthand (FSH)
  It is a domain-specific language for defining the content of FHIR implementation guides. This allows resource profiling, value sets, examples, using any text editor. Use formal grammar.
- SUSHI (Unshortens ShortHand Inputs)
  it is a reference implementation and de facto standard for a FSH compiler that translates FSH into FHIR artifacts such as profiles, extensions, and value sets. It therefore allows you to convert scripts written according to the FSH language into FHIR artifacts.
- IG Publisher
  It is a tool provided by the FHIR team to translate IG content into:
  - generated resources that can be included in the published guide (XML, JSON, TTL formats);
  - HTML files;
  - zip files used by implementers.

The tool was used after the FSH scripts were compiled by the SUSHI compiler. And then get the different web pages of the IG. These tools are the official tools provided by HL7 international.

2.1.2 Data model mapping

This section gives the specification of the data collected, and the associated constraints and extensions to the base FHIR model. In a subsequent iteration, this specification will be formalized in a FHIR Implementation Guide. Where applicable, relevant profiles and definitions are mentioned.

The rest of the section details the SMART BEAR patient information, the mapping to and constraints on FHIR resources, relevant profiles and notes. Here we extend the model described in the previous version of this deliverable (which was D4.2 Report on SMART BEAR Cloud Enabling Components v1), and focus on the hearing loss use case, which was not covered in D4.2.

Hearing aid monitoring takes place by means of an App that polls the hearing aids for data via Bluetooth. Polling takes place on regular basis (nominally 15 min), and the data derived from the hearing aids is formed into a set of FHIR Observation resources each of which contains a single aspect of the hearing aid data. All of these Observation resources are linked to a single “parent” Observation resource that maintains the collection of information. All Observation resources reference the Device resource that defines a specific hearing aid device. Properties of the Device resource are also updated with the latest polled information to provide a current view of the hearing aid status.

- Device resource for each HA updated (stores point in time data)
- Multiple light weight Observation resources created for each poll
- Properties stored in individual Observation resources (eg. A parent observation resource plus one for each of 9 attributes)

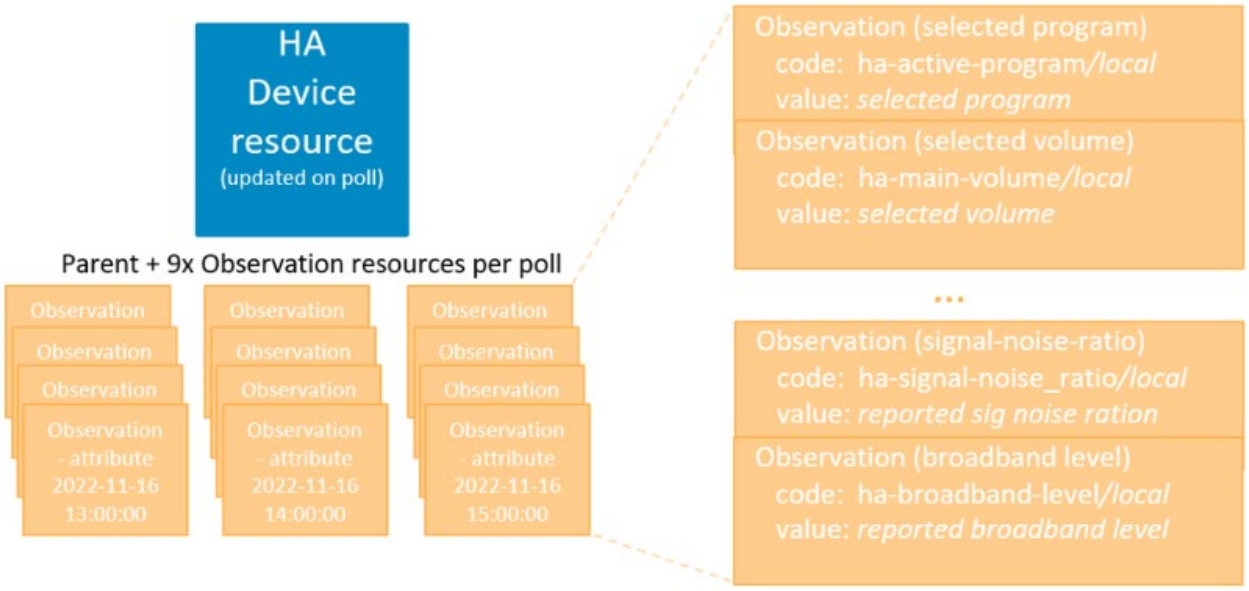

Figure 2. FHIR Resource Design for Hearing Aid Monitoring

Table 1. Hearing Aid Properties

Code	Definition/Notes	Related Value (Observation)	Related Property (Device)
property-set	Code for the "parent" Observation to reference a set of component Observations.		N/A
mute-status	Code for Observation containing mute status of the hearing aid. In Device this is represented by statusReason=paused.	valueBoolean with options: true=muted | false=unmuted	N/A
main-volume	Main volume setting of the hearing aid.	valueCodableConcept.coding[0].code with options: ["-4","-3","-2","-1","0","1","2","3","4"]	valueCode[0].coding[0].code with options: ["-4","-3","-2","-1","0","1","2","3","4"]
main-program	Main program setting of the hearing aid.	valueCodableConcept.coding[0].code with options: ["0","1","2","3"]	valueCode[0].coding[0].code with options: ["0","1","2","3"]

time-since-boot	Time since last boot in seconds.	valueInteger	valueQuantity[0].value
total-usage-time	Total time device has been used in seconds.	valueInteger	valueQuantity[0].value
signal-noise-ratio	Signal to Noise Ratio Estimator.	Array of decimal values in: component[n].valueQuantity.value	Array of decimal values in: valueQuantity[n].value
noise-level	Noise Level Estimator.	Array of decimal values in: component[n].valueQuantity.value	Array of decimal values in: valueQuantity[n].value
broadband-level	Broadband Level Estimator	Single decimal value in: valueQuantity/value	Single decimal value in: valueQuantity[0]/value

Table 2. Device resource data elements

Element	Contents
status	Normally "active". Use "inactive" if the device is no longer operational or a part of the project
statusReason	"**online**" if device can communicate, "**paused**" for muted state, or "**offline**" if device is not accessible
manufacturer	Name of device manufacturer
type	Left HA "**6012004:272741003=7771000**" or Right HA "**6012004:272741003=24028007**"
property[]	Set of properties as outlined in the Related Property (Device) column of the Hearing Aid Properties table
patient	Reference to related Patient resource

Table 3. Observation resource data

Element	Contents
status	Normally "active". Use "inactive" if the device is no longer operational or a part of the project
code	Item from the Code column of the Hearing Aid Properties table indicating the value of this Observation
subject	Reference to the related Patient resource
effectiveDateTime	Date time of observation
bodySite	Left Ear "**89644007**" or Right Ear "**5577004**"
device	Reference to the relevant Device resource

hasMember[]	Only required when code == 'property-set': a set of references to the hearing aid Observations taken at the same time
value[]	The item in the Related Value (Observation) column of the Hearing Aid Properties table relevant to the code value of this resource.

2.1.2.1 FHIR artefacts produced

The specifications described in the previous sections have been translated in a formal FHIR IG using FSH and IG Publisher. The corresponding web pages have also been created.

The final published version (v3.4) IG (Implementation Guide) consists of 153 Resource Profiles (of type Observation, Condition, Questionnaires, Bundle, Patient, DeviceUseStatement, FamilyMemberHistory, MedicationStatement, ResearchSubject), 7 Extensions, 45 Value Sets and 258 examples. The FHIR IG is available at https://smart-bear.eu/fhir/ .

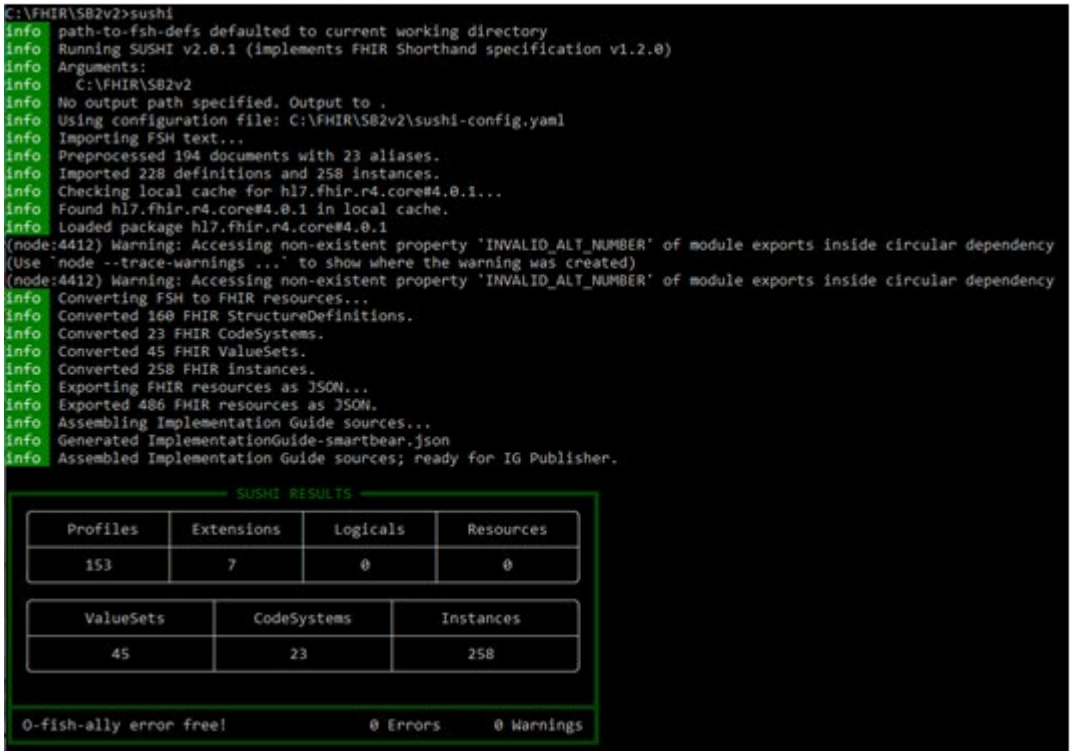

Figure 3 SMART BEAR FHIR profiles

This section shows some illustrations of the web specifications produced for the Hearing Aids FHIR Artefacts. The main profiles produced are listed below.

Profile HADevice

The resource profile HADevice is a profile to capture data for a hearing aid device (left and right). This resource profile is built using parent HL7 resource is Device.

Figure 4 shows the Differential Table.

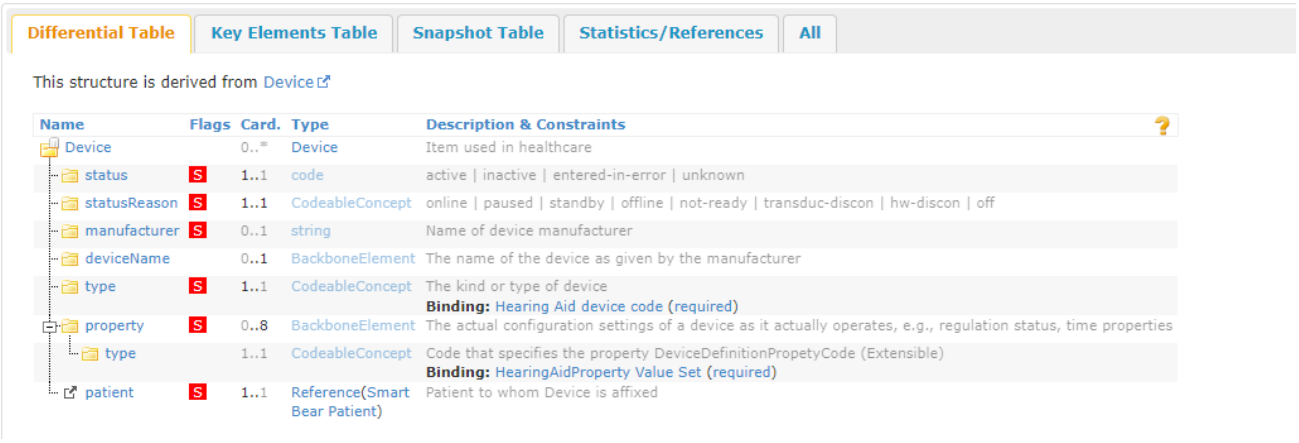

Figure 4 Differential Table for Hearing Aid Device

The type of device is a CodeableConcept, where the possible code are:

- 6012004:272741003=7771000 Left hearing aid
- 6012004:272741003=24028007 Right hearing aid

The type of property is a CodeableConcept, where the possible values are showed in the table below.

Table 4. Possible codes of Codeable Concept

Code	System	Display	Definition
property-set	https://smart-bear.eu/fhir/CodeSystem/BelatonHearingAidProperty	Beleton Hearing Aid Property Set	Code for the "parent" Observation to reference a set of component Observations.
usetime_down	https://smart-bear.eu/fhir/CodeSystem/BelatonHearingAidProperty	Use time Mi link down	Use time Mi link down
usetime_up	https://smart-bear.eu/fhir/CodeSystem/BelatonHearingAidProperty	Use time Mi link up	Use time Mi link up
usetime_down_during_usage	https://smart-bear.eu/fhir/CodeSystem/BelatonHearingAidProperty	Use time Mi link down in hours during usage	Use time Mi link down in hours during usage
usetime_up_during_usage	https://smart-bear.eu/fhir/CodeSystem/BelatonHearingAidProperty	Use time Mi link up hours	Use time Mi link up hours

		during usage	during usage
total_usetime	https://smart-bear.eu/fhir/CodeSystem/BelatonHearingAidProperty	Total use time	Total use time
usetime_per_preset	https://smart-bear.eu/fhir/CodeSystem/BelatonHearingAidProperty	Use time per preset	Use time per preset
numberof_device_boots	https://smart-bear.eu/fhir/CodeSystem/BelatonHearingAidProperty	Number of device boots	Number of device boots
numberof_presetswitches	https://smart-bear.eu/fhir/CodeSystem/BelatonHearingAidProperty	Number of preset switches	Number of preset switches
histogram_environment_per_preset	https://smart-bear.eu/fhir/CodeSystem/BelatonHearingAidProperty	Histogram On Classified Environment Per Preset	Histogram On Classified Environment Per Preset
histogram_meanvolume_preset1	https://smart-bear.eu/fhir/CodeSystem/BelatonHearingAidProperty	Histogram On Mean Volume Control Level Preset 1	Histogram On Mean Volume Control Level Preset 1
meanvolume_per_preset	https://smart-bear.eu/fhir/CodeSystem/BelatonHearingAidProperty	Mean Volume Control Level Per Preset	Mean Volume Control Level Per Preset
meanvolume_environment_preset1	https://smart-bear.eu/fhir/CodeSystem/BelatonHearingAidProperty	Mean Volume Control Level Per Environment Preset 1	Mean Volume Control Level Per Environment Preset 1

ParentResourceHAObservation

It is an Observation for Parent Resource Hearing Aids Observation

Figure 5shows the Differential Table.

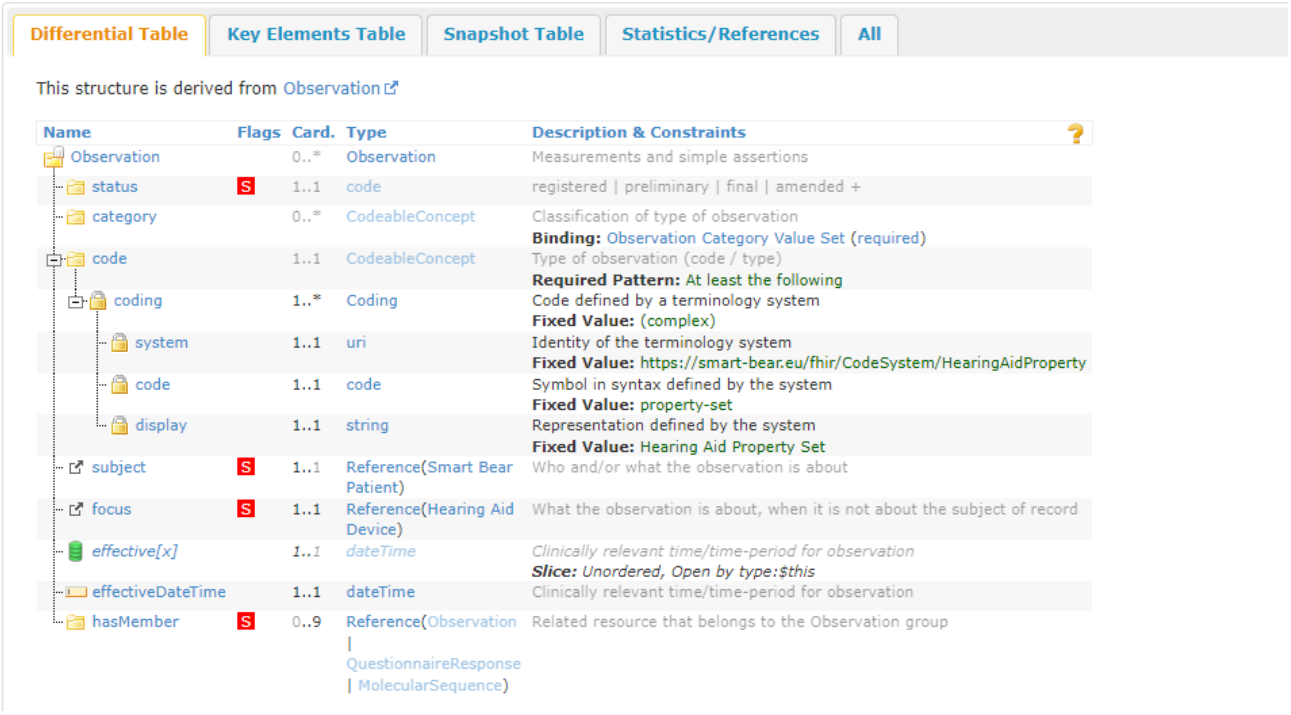

Figure 5 Differential Table for HA Observation

MemberResourceHAObservation

It is an Observation for Member Resource Hearing Aids Observation

The figure shows the Differential Table.

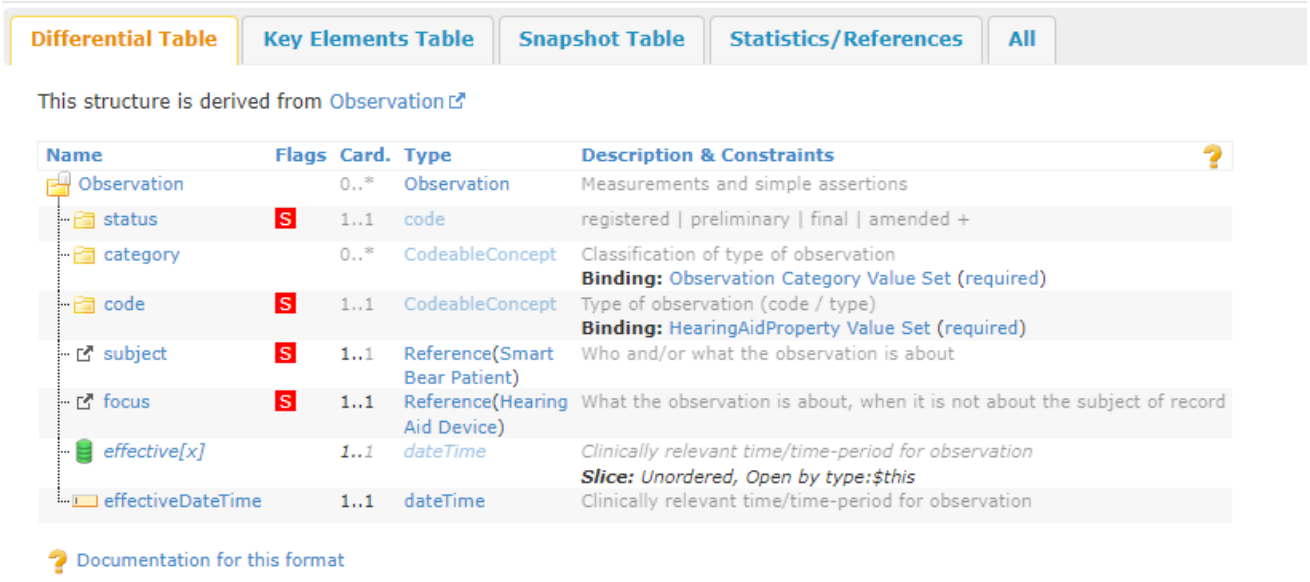

Figure 6 Differential Table

The code of this Observation is an CodeableConcept, where the possible values are showed in theTable 4.

2.2 BDA Engine

The BDA engine mainly addresses the functionalities required for processing DAWs (Data Analysis Workflows) and providing/storing execution results.

The choice of technologies for the BDA engine, particularly for executing data science and machine learning tasks, is important. At its core lies Apache Spark, a unified analysis engine designed for large-scale data processing. Specifically, PySpark[1]—an interface to Apache Spark in Python—was utilized. PySpark not only facilitates the development of applications through its Python API but also offers a shell for interactively analyzing data within a distributed environment. Through PySpark, patient clinical data undergo analysis and manipulation, primarily through pre-processing activities. Additionally, Pandas[2], a Python library renowned for its optimized tools and frameworks tailored for handling tables and time series, was employed to support data manipulation tasks.

2.2.1 BDA components

The BDA engine is designed to achieve scalability by allowing resources to be easily added to the platform to support increasing workloads. Its configuration and deployment of resources are straightforward, and because it runs on Docker containers, it offers high flexibility [29].

The BDA engine uses the following components (see Figure 7):

- Apache Hadoop[3]: Utilizes the Hadoop Distributed File System (HDFS) for storing and transforming extracted data from the FHIR repository into a tabular format.
- Apache Hive Metastore[4]: Contains information about databases, tables, and their relationships, facilitating data handling on the distributed file system. It ensures interoperability among components accessing the data and supports SQL-like queries via Trino and Spark.
- Apache Spark[5]: Facilitates data management, engineering, and ML tasks on large datasets, treating HDFS data like SQL tables for ease of use and scalability.
- Trino[6]: A distributed SQL query engine for querying large datasets from various sources, which is suitable for data warehousing, analytics, and ETL transformations, and can run SQL queries on HDFS.
- Apache Airflow[7]: it develops, schedules, and monitors batch workflows through an extensible Python framework, and runs custom analytic tasks with Docker.

[1] https://spark.apache.org/docs/latest/api/python/index.html

[2] https://pandas.pydata.org/

[3] https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

[4] https://hive.apache.org/

[5] https://spark.apache.org/

[6] https://trino.io/

[7] https://airflow.apache.org/

- BDA API: this component was developed by the SMART BEAR consortium and provides a REST interface for interacting with workflows and managing execution results, with a catalog of atomic analytics for workflow composition.
- Delta Lake[8]: an open-source storage framework based on Hadoop, which enables a Lakehouse architecture supporting ACID transactions and unified streaming and batch data processing [30].
- Apache Zeppelin[9]: A web-based tool for interactive data analysis, prototyping analytics, and sharing results in a collaborative environment for data scientists.

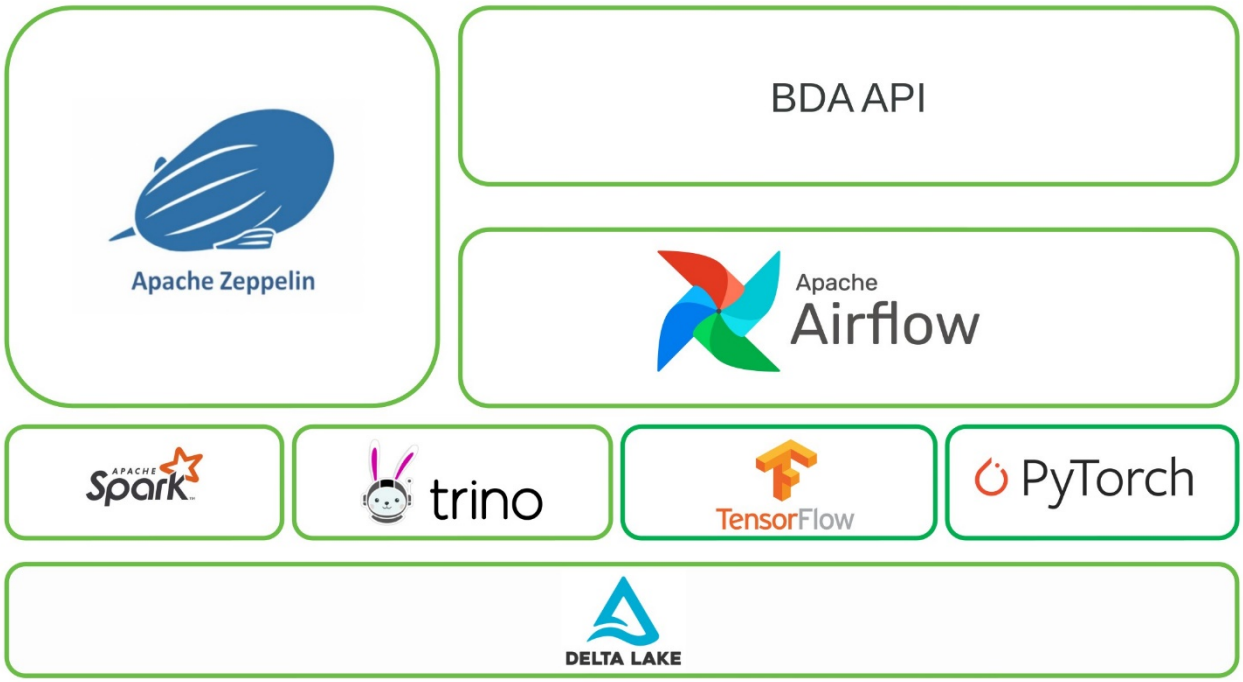

Figure 7. BDA Components

2.2.2 Data storage and data management implemented by BDA

Data extracted from various storage sources, such as the FHIR repository, undergoes a series of transformations before being stored in the format requested by Delta Lake. This format facilitates the creation of a Lakehouse architecture, which integrates the strengths of both data lakes and data warehouses. Key features of this architecture include (i) ACID transactions, (ii) schema enforcement, (iii) support for business intelligence, (iv) openness, (v) compatibility with diverse workloads, and (vi) handling of structured and unstructured data [31]. Through Delta Lake, we meet these requirements and provide a robust version management process for the data environment. Utilizing the Time Travel component, we can specify a particular version of a table, enabling us to rerun analyses on specific

[8] https://delta.io/

[9] https://zeppelin.apache.org/

data versions for comparative purposes. Additionally, it furnishes a mechanism for result replication and facilitates data and result audits.

2.2.3 Data analysis task

Data analysis tasks predominantly rely on Spark and Python libraries capable of processing data in Delta Lake format. Depending on workflow complexity, these tasks may involve multiple steps. In certain scenarios, it's more efficient to manipulate and prepare data using Spark and subsequently utilize specialized libraries or tools for analytics. Leveraging Apache Airflow, we can construct Directed Acyclic Graphs (DAGs) comprising various tasks and employ HDFS as a distributed file system to store data required for different steps. Moreover, this flexibility allows for the use of preconfigured Docker images containing all necessary tools for model creation and analytics, if needed. Thus, the primary requirement remains the ability to access data from a distributed file system like HDFS.

The daily export workflow comprises various tasks, roughly categorized as: (i) extracting from FHIR, (ii) flattening the data, and (iii) upserting into Delta Table. Data extraction leverages the capabilities of the FHIR REST API, specifically utilizing the Bulk Data Export API. This allows for requesting data that has been inserted or modified in the repository since the last successful export process. The exported data, formatted in ndjson, is then stored on HDFS. The subsequent step involves transforming the data into a flat format, akin to that of a typical SQL table. This entire process is orchestrated through Apache Airflow as the primary engine. With Airflow, workflows can be configured to run at predefined intervals, either with predefined schedules or, in specific cases, using custom schedules defined via Cron-like expressions.

2.2.4 Data visualization

For data visualization, the BDA Engine primarily utilizes two distinct tools: Apache Echarts for showcasing analytics results within the Dashboard interface, and Apache Zeppelin for enabling data scientists to conduct data exploration and prototype analytics for deployment. Apache Echarts, an open-source JavaScript visualization tool, operates seamlessly on web browsers and mobile devices. It boasts a robust library of fundamental charts and offers customization options to suit our output requirements. Many available components, among the most used there are certainly: (i) Datazoom that is used for zooming a specific area, which enables users to investigate data in detail, get an overview of the data, or get rid of outlier points. (ii) Timeline which provides functions like switching and playing between multiple Echarts. (iii) Toolbox that contains some functionalities like the export to PNG and zooming. (iv) Legend that shows symbol, color, and name of different series. The user can click legends to toggle displaying series in the chart.

2.2.5 Platform management

Given that all tools within the BDA Engine are containerized, Kubernetes (K8s) serves as our primary platform for managing the entire system. K8s handles scaling, automatic deployment, and management of all containerized applications in the Cloud environment [32]. The basic configurations for all described components are stored as ConfigMaps in the dedicated Smart Bear repository for the BDA Engine, while passwords and sensitive data are stored as Secrets. Helm Charts are utilized to deploy certain components, such as the Spark cluster, while others are deployed ad-hoc. Final configurations are then performed using the provided web UI tool. These administrative UIs are accessible only to

system administrators, who have the authority to modify tool deployments and manage resources to ensure infrastructure performance meets requirements. Since all the tools used within the BDA Engine have been containerized, to manage the entire platform the tool we rely on is the deployment platform itself, i.e., Kubernetes (K8s). K8s is the system that we use to handle scaling, automatic system deployment and to manage all the containerized applications that we use in the Cloud10Analytics.

2.3 Analytics workflow

This section describes the range of approaches to implement AI-leveraged analytics in the various with the BDA engine clinical scenarios of SMART BEAR.

2.3.1 Data Preparation

The data managed by the BDA are organised in 8 main tables: Components, Observations, Encounters, Conditions, Patients, Medications, Research Subjects, Diagnostic Reports, and Questionnaire Responses.

Patients: Contains master patient information.

Observations: Contains the medical observations of the patients, such as the results of examinations and measurements performed by the smart devices.

Conditions: Contains information about the patient's medical status, such as the presence of medical conditions.

Questionnaire Responses: Contains the results of some of the questionnaires that the patients have completed.

To align with the FHIR standard entries are associated to a well-defined semantics captured using LOINC and SNOMED, which are used to uniquely identify medical observations transmitted by universal numeric codes. The initial step is to extract all the information necessary for the analysis and prediction activities. This data is referred to as raw data because no preprocessing has been done. Each patient is uniquely identified by a numeric ID, which is used to track all his information across other datasets. Each observation is associated with a set of information according to the FHIR schema, including the medical ID code according to the LONIC and SNOMED standards. Within the medical domain, a further distinction can be made between data extracted from time series and non-time series. Time series data, such as blood pressure or heart rate, always has an additional value associated with it: the time at which the measurement was taken. This represents repeated data over time and distinguishes it from non-time series data, such as gender, which is likely to remain unchanged. When

dealing with time series data, it is important to ensure that the extracted information is current (see Figure 8).

Patient_id	Age	Gender	Educational Status	Household Composition	Smoking	BMI	Physical activity
1045034429	69	Male	Elementary school	null	null	null	no
1065442966	67	Female	University	Lives with family	Former smoker	31.4	yes
1067687190	77	Female	null	null	null	null	no
1079961145	71	Female	Elementary school	Lives with family	Never smoker	26.6	yes
1086760558	71	Female	University	Lives alone	Never smoker	null	yes
1135122000	81	Female	null	null	null	null	no
1142775641	72	Female	Elementary school	null	null	null	no
1158131383	67	Male	Elementary school	Lives with family	Former smoker	null	no
1170694944	74	Male	Elementary school	Lives with family	null	null	no
1178099045	75	Female	University	Lives alone	Former smoker	34.8	yes
1192125641	70	Male	null	null	null	null	no
1207244471	74	Female	null	null	null	null	no
1243869239	72	Male	Elementary school	Lives with family	Former smoker	38.3	yes
1278905543	72	Female	Elementary school	null	null	null	no
128350896	69	Male	Elementary school	Lives with family	Former smoker	null	yes
1284035951	73	Male	Elementary school	null	null	null	no
1314082283	81	Female	University	null	null	null	no
1352594320	71	Female	Elementary school	null	null	null	no
1387732224	71	Female	Elementary school	Lives with family	Current every day smoker	null	no
1396750978	84	Female	Elementary school	null	null	null	no

Figure 8 Excerpt of raw data in the BDA

2.3.2 Data Quality Workflow

The process of ensuring data quality involves several steps aimed at enhancing accessibility, usability, and efficiency of analytical results. In the context of our BDA, we focus on exploring various aspects of data quality. Our proposed workflow consists of six steps, that are described here below and are shown in Figure 9

- Step 1: Data Homogeneization. Homogeneization is carried out involving the diverse clinical data sources. SMART BEAR creates the interoperable data model for structural homogenization while for contextual homogenization purposes takes advantage of the identified LOINC16 and SNOMED-CT17 codes in defining observations, encounters, and biological considerations that will be stored in FHIR repositories [33]. In the context of the SMART BEAR project, a data model for mapping the questionnaires according to the FHIR principles has been developed. According to this model, an FHIR questionnaire template requires resources where the URL, name, and title shall have a value while the version might have a value. This way, the questionnaires are mapped on the FHIR data model.
- Step 2: Data Organizing and Restructuring. This step has the goal to adapt data into a format compatible with data analytics and machine learning algorithms. In order to bring them into a tabular format compatible with the vectors ingested by data analytic algorithms, an ad-hoc script is created using Apache Spark consists of the following steps:
    1. Flatten all the properties, which means that each property and object nested in the JSON creates a column by concatenating the name of the properties with those that are children of the main property.

2. Create a new row for each element of an array if the element contains arrays within the structure. In this case, a new row is created in the resulting table for each element of this array.

- Step 3: Data Transformation. This step requires three operations: Extraction, Transformation, and Loading (ETL). By extraction, we query and gather the required data in the original data set; while by Transformation we could apply multiple changes such as Data Filtering, Data Mapping, Data Deduplication, Data Cleansing, Derived Variables, Data Sorting, or Ordering. All these operations are needed to proceed incrementally as the SMART BEAR is running and new data is arriving. For Extraction and Loading purposes, the BDA engine exploits the Trino and Delta lake.
- Step 4: Data Virtualization. Data virtualization creates a layer where users can access, retrieve, and manipulate data as needed. It brings all the information together in one virtual location, allowing real-time access with no need to perform ETL. Data virtualization is often more cost-effective and accurate. SMART BEAR uses the Apache Zeppelin environment for this purpose. Zeppelin is a web-based notebook that enables data-driven, interactive data analytics and collaborative documents. Using Zeppelin data scientists can implement step 3 and proceed with the rest of the procedure serving the homogenization and organizing while exploring the available data. Notebooks created in this way are a prototype to obtain feedback and then move on to the analytics deployment phase in the system. After the notebook is finalized by the experts we can start to use the code and test it, the accepted code can be released and included in the BDA catalog to make it accessible via the BDA API. Through Zeppelin, it is possible to launch Python, Spark, SQL Query, or R scripts. This is possible by the different interpreters that are provided by the system and that are launched using Docker images containing the required components and libraries already configured to execute the snippets.
- Step 5: Data Normalization. Data normalization is the organizing of data to make them more similar in form and scale across all records and fields. It increases the cohesion of entry types leading to the cleansing the unstructured data, redundancies, and duplicates. Data normalization is basically the mapping functions unifying the scales of numerical values without loss of generality. For example, normalization by max/min values and z-score are advised regarding the scope of analytics.
- Step 6: Data imputation. The sparse multivariate temporal series collected by smart devices in smart-health projects requires specific methodologies to leverage feature sparsity by effectively studying the correlation between features, the achieved data quality levels, and appropriate techniques for filling the missing values to be adopted. The proportion of missing data only provides limited information about the bias and efficiency gains that can be made from data imputation. The recent literature clarified that data imputation needs to be anticipated by correlation analysis to verify the pattern followed by missing values [20, 23, 24, 30, 37]. This implies that researchers should consider whether all the variables related to missingness can plausibly be included in the imputation model to limit bias and improve accuracy. An additional challenge of imputation techniques in longitudinal studies is related to the consistent integration of temporal series and scalar features in imputation models.

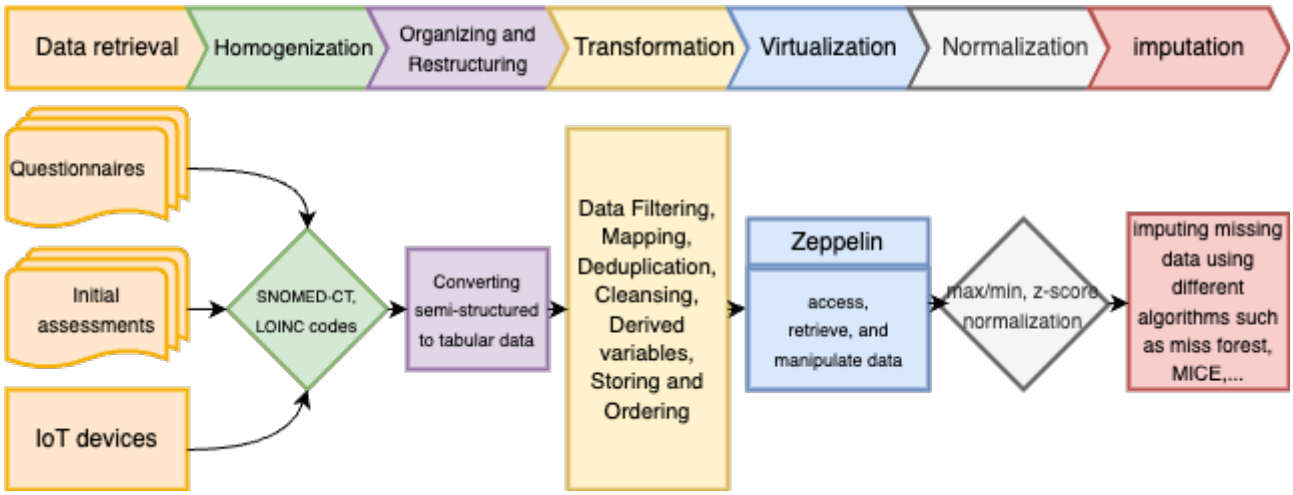

Figure 9. Data quality workflow

2.3.3 Workflow for designing a prognostic model

A workflow has been developed that fulfils the requirements for designing a prognostic model. This workflow involves multiple steps that can be executed separately, with a validation step proposed at the end to ensure the accuracy of the model. The steps followed for designing a diagnostic model are described here below and are shown in Figure 10.

- Step 1: Feature extraction. The extraction of suitable features is crucial in managing the complexity of a problem when dealing with heterogeneous data types in machine learning (ML) models. This involves unifying a dataset containing diverse data types like time series, categorical data (e.g., gender), and numerical data (e.g., age). The goal is to create a unified form and temporal dimension for these inputs in a d-dimensional feature space required for ML model training. Various algorithms are available, including those extracting static statistical features and others based on embedding algorithms that maintain the temporal relevance of data points while abstracting records [34].
- Step 2: Sample size approximation. estimating the number of data points required for validating an ML model is a critical aspect when starting a study with limited resources. The SMART BEAR project outlines a three-stage process for defining, implementing, and validating prognostic models. The first stage involves converting multidimensional data into a single-valued representation. The second stage uses correlation analysis to detect patterns in missing data, facilitating their integration into ML models for imputation and predictions. The third stage examines the proportion of missing information to predict its impact on prognostic models. Evaluating the sample size with a temporal plan is essential to establish a credible timeline for acquiring realistic data volumes. Planning with complex models is impractical if the data size is inadequate. For instance, literature evidence indicates that Random Forest performs reliably with a sample size of 200 times the number of features. Therefore, once the sample size is determined, the feature space's dimension and the appropriate modeling tool can be identified.

- Feature selection (the variables to be used in the learning process): for a specific model, we need to select specific features out of the existing measurements that are carried out in the SMART BEAR project.
- Target definition (the variables to be predicted): as we will discuss more specifically the target of the model should be defined, in other words, which is the output of the model.
- Sample size continuous evaluation and drift detection: in a running project, we repeatedly run the sample size approximation since new data brings new insight into the feature space.

- Step 3: Validation. For ML models, the new dataset can act as the test set, while the k-fold cross-validation algorithm is utilized to find the optimal parameters. Temporal validation can also be suggested as an external validation method. In projects like SMART BEAR, accuracy is not the only metric evaluated due to potential biases; clinician approval is also necessary. This decision is based on active collaboration with clinicians, ensuring that any adopted model or sample size not only demonstrates stable accuracy across all cross-validated folds (low variation) but also should be evaluated by clinicians for the output's suitability and the relevance of the extracted information.

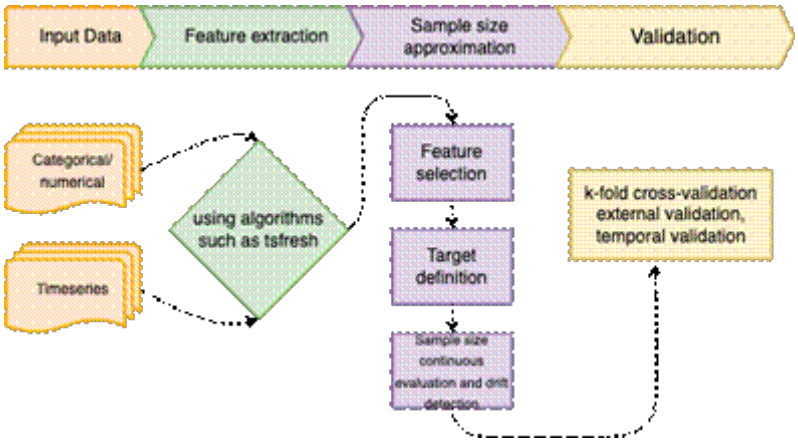

Figure 10. Flow for designing a prognostic model

2.3.4 Data augmentation with generative models

As discussed previously, accurate Machine Learning models require high-quality, large-volume data. Even if we take care of the quality of the data, we may still encounter challenges related to the scarcity of data and the unbalanced distribution of the same.

The quality of acquired data from the SMART BEAR project has been discussed before in D4.3 and D4.4.

In the context of predictive models, data augmentation plays a crucial role in improving the accuracy and generalization of models. This practice involves expanding the existing dataset through various techniques, including adding noise and generating synthetic data from the real distribution. Adding noise to existing data is one of the most common augmentation techniques. This process can help make the model more robust and adaptable to variations in the input data. However, it is important to consider

that adding numerical data noise may not be appropriate for all features, as in the case of "gender" which having only two categorical values, adding random values would compromise coherence and significance. Another strategy for augmentation is the generation of synthetic data starting from a real data seed. This approach allows you to introduce new data instances that maintain the essential characteristics of the reference dataset.

With synthetic data generation we are trying to address data scarcity and imbalance classes and propose solutions to improve data management and analysis. To do so, we have implemented an application of synthetic data generation to be used by predictive models. This analysis will focus on the effectiveness of predictive models, examining evaluation metrics such as accuracy, precision and the data distribution. To have a complete view on the real versus synthetic dataset, we have developed a web application capable of illustrating the results on a web platform. Using the SMART BEAR reference dataset as a seed for generating synthetic data, we have employed several techniques, including Conditional Tabular Generative Adversarial Networks (CTGAN), Anyway Conditional Tabular Generative Adversarial Networks (ACTGAN) and Deep Generative Adversarial Networks (DGAN), each of which has specific advantages and limitations. To stay more efficient, we report only the output using CTGAN. The synthetic data generated will be applied in the prediction of a specific comorbidity, which in our case is depression. We will use the metrics mentioned to evaluate the predictive models and determine the model improvements using the augmented dataset.

2.3.4.1 Back end

As for the back end, we use Python to exploit the open-source CTGAN library as depicted in the Figure 11.

Since generative models require a starting dataset for generating synthetic data, we consider the dataset relevant to the SMART BEAR case study based on patient depression, suggested by the clinicians. This dataset contains different data with different types:

• Numeric: BMI (Body Mass Index), SQS (Sleep Quality Scale), GDS (Geriatric Depression Scale), Physical activity, Diastolic, Systolic, Hear Beat, Heart Rate.

• Categorical: Gender, Smoker, Household, Alcohol, Education, Depression.

```
from ctgan import CTGAN
discrete_columns = [
'Smoker',
'Gender',
'Household',
'Alcohol',
'Education',
'Depression',
'Cardiovascular Disease',
'Cognitive Impairment',
'Age',
'SQS',
'GDS'
]

ctgan = CTGAN(epochs=10)
ctgan.fit(df, discrete_columns)
# Create synthetic data
synthetic_data = ctgan.sample(100)
```

Figure 11: Implementation of CTGAN

After generating the new dataset containing the synthetic data, we proceed with the evaluation and comparison of these data with the real ones through a statistical analysis. This allows us to understand whether the distribution of the synthetic data was consistent with that of the real data.

The objective is to predict the state of patients, determining whether or not they suffer from depression, and subsequently calculate the accuracy of the model obtained. We have tried Logistic Regression, Random Forest, K-Nearest Neighbors Classifier, Support Vector Classifier, and Decision Tree Classifier. As shown previously for synthetic data, here too we have a first phase which concerns data preprocessing. The outputs are presented in the following Table 5.

Table 5: Comparison of performance of models for predicting the Depression state

	Accuracy	Precision	Recall	F1 Score
LR	**0.714**	0.667	**0.500**	**0.571**
RF	0.667	**1.000**	0.125	0.222
KNNC	0.667	0.667	0.250	0.364
SVC	0.667	**1.000**	0.125	0.222
DTC	0.571	0.400	0.250	0.308

2.3.4.2 Front end

To have a more detailed overview of the experiment results, we developed a client-server platform with a responsive web interface. The starting page is presented in Figure 12.

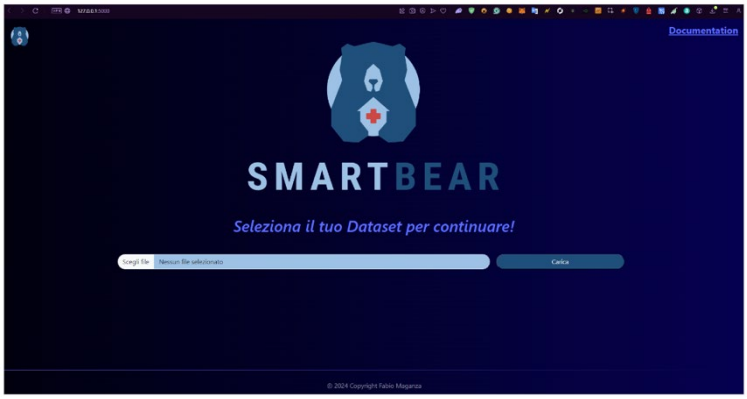

Figure 12: Web interface

This system allows the user to upload their csv dataset directly to the site. Once the data has been uploaded, the user can view the graphs and results of the experiment on their device. Once the correct loading of the .csv has been verified, a success message is shown with a carousel containing the graphs relating to each feature of the dataset (synthetic data in yellow and real data in blue in the Figure 13), including a legend indicating two statistical metrics (mean and median) useful for understanding the correlation between synthetic and real data.

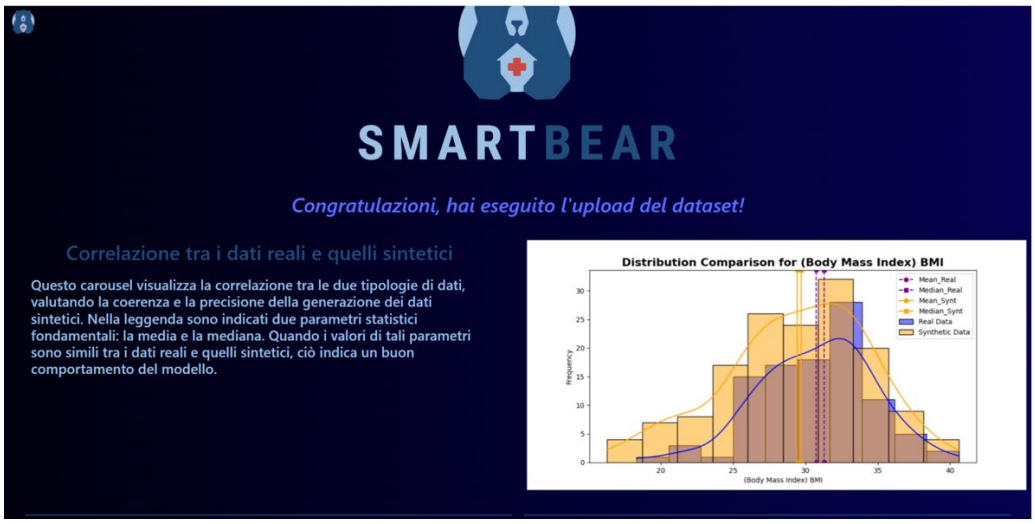

Figure 13: Synthetic data graphs generated

In addition to the graphs, the platform also displays two tables of predictions relating to a specific column. Two tables and two sets of predictions are therefore generated with the first 10 records: the first relating to the original dataset and the second relating to the real dataset combined with the synthetic data. Theoretically, the latter should present greater precision having a greater amount of data available. Scores for all metrics covered (Accuracy, Precision, Recall, F1) are shown in Figure 14, allowing users to directly compare how the model works and how the data correlates. The idea was to create a direct interface that offered a quick and intuitive way to examine and interpret data and analysis results.

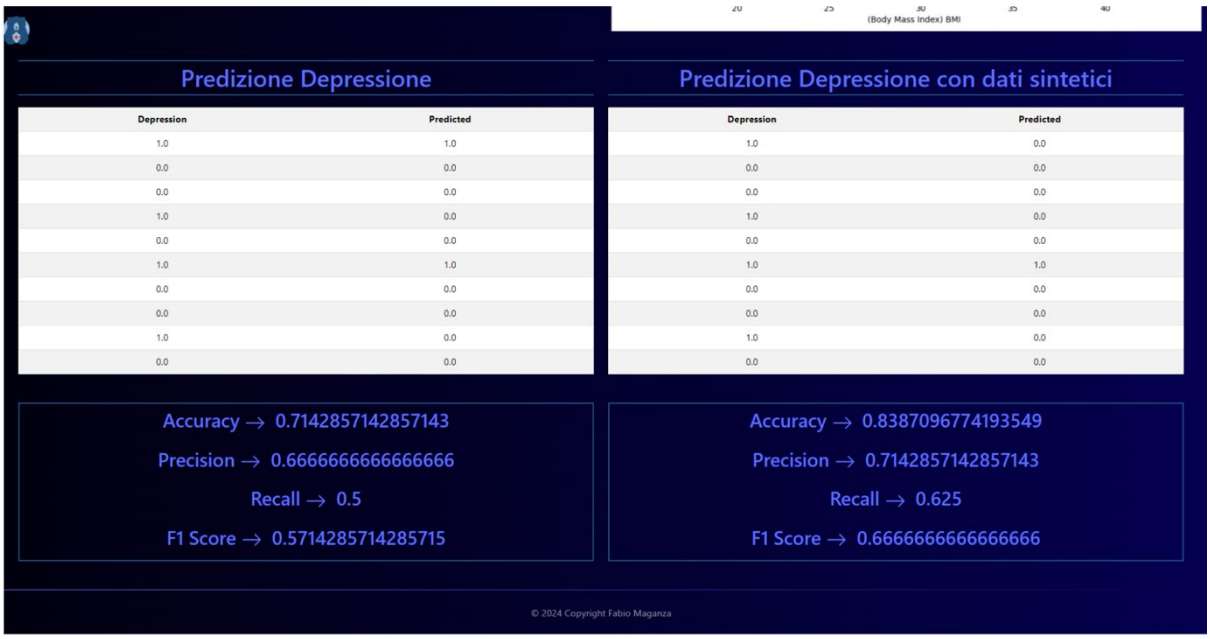

Figure 14: Predictive tables

2.3.4.3 Utilized technologies

As for the web platform and for the implementation of the ML algorithms, we relied on different technologies.

For developing the site structure, we used HTML, CSS and JavaScript, taking advantage of the Bootstrap CSS framework to easily customize the page style and manage responsiveness. As for the server side, we opted for Python in combination with the Flask[11] framework, which greatly simplified the creation of the server and the management of requests coming from and going to the client. As for the machine learning algorithms, we mainly used the open-source scikit-learn (sklearn) library[12] for model predictions and training. Additionally, we leveraged Matplotlib[13] for data visualization via graphs.

We employed Conditional Tabular Generative Adversarial Network (CTGAN) for synthetic data generation; considering the input SMART BEAR dataset as an example, we did not have many records to generate, CTGAN therefore proved to be an ideal choice to meet our basic needs, given the easy implementation and the excellent performance in terms of processing time.

2.3.5 Intrinsic Capacity

Intrinsic Capacity (IC) is an index determined by the composition of all physical and mental abilities that generally describe the biological, personal and psychological state of an individual. This model is an evolution of the previous framework published by the WHO, namely the International Classification of Functioning, Disability and Health (ICF), which consists of multiple interacting elements that influence contextual factors (personal and environmental) and determine health status [35]. Over the years researchers have tried to determine the functions of the body that are most useful for maintaining the individual's independence and healthy living, but also those functions whose impediment is connected to functional loss and loss of independence.

The only guidance available for the implementation of IC is the Integrated Care for Older People (ICOPE) manual, published by the WHO, which defines that IC is the composition of six domains: Vitality (vitality), locomotor capacity (locomotion), visual capacity (vision), auditory capacity (hearing), psychological capacity (psychological) and cognitive capacity (cognition). In reality, auditory and visual ability are usually aggregated into a single domain, i.e. sensory ability, resulting in a total of five domains [36]. According to research conducted by Gonzalez et al. in 2020 [37] on ten studies, there were no standardised IC index validated for research or clinical use, and all these studies analyzed by them were not designed to measure IC and used retrospective data. Also, not all calculated a global index and only one study operated with multi-country data.

This section presents the work carried out to develop an IC index using the data provided by SMART BEAR. We will start from the organization of the data provided, through the pre-processing, to the calculation of the index, explaining the methodology used. SMART BEAR offers a good correspondence with the dimensions of the IC index. Table 6below shows the SMART BEAR variables associated with IC domains.

[11] https://flask.palletsprojects.com/en/3.0.x/

[12] https://scikit-learn.org/stable/

[13] https://matplotlib.org/

Table 6. Intrinsic Capacity variables

Domain	Variable	Unit of Measure	Frequency	Risk Values
Locomotion	Number of steps	steps	daily	< 4000
	Walked distance	km	daily	< 2400
	Frail score	hours	6 months	>= 3
	Sarcopenia score	hours	6 months	>= 4
Sensory	Sensory conditions1	questionnaire	daily	negative rating: none
	Light sleep duration	hours	daily	< 7
	Deep sleep duration	hours	daily	>= 10
	Sleep Duration	hours	daily	>= 5
Psychological	Personal Health Questionnaire Depression Scale (PHQ-9)	questionnaire	daily	negative rating: < 1.75
	Geriatric Depression Scale (GDS)	questionnaire	daily	>= 10
	Psychological conditions2	questionnaire	bi-weekly	negative rating: < 26
	Montreal Cognitive Assessment (MoCA)	assessment	6 months	<= 7
	Rapid Cognitive Screen (RCS)	questionnaire	6 months	negative rating: < 12
	Cognition conditions3	assessment	12 months	negative rating: < 12
Vitality	MNA Short Form	questionnaire	monthly	> 18.5 ∧ < 24.9
	SNAQ	questionnaire	weekly	> 13 ∧ < 24
	Max Heart Rate	bpm	6 months	> 100
	Min Heart Rate	bpm	daily	< 60
	Average Heart Rate	bpm	daily	> 70 ∧ < 100
	HDL cholesterol	mg/dl	6/12 months	< 40
	LDL cholesterol	mg/dl	6/12 months	> 130
	BMI	kg/m3	daily	< 129
	Body Fat Percentage	%	daily	> 100
	Standing systolic blood pressure	mmHG	6/12 months	< 129
	Standing diastolic blood pressure	mmHG	assessments	< 180

We decided to create a dynamic web page containing all the useful views, to facilitate their consultation, but in particular to make it interactive, allowing the end user, i.e. the medical staff, to select the desired patient and to customize viewing charts using Apache ECharts.

As regards the frontend part, the open-source bootstrap framework was used, as it allows to simplify and speed up the creation part of the web page, as it contains HTML, CSS and javascript elements for the creation of the interface components.

An important part of bootstrap is the fact that it provides easy to use elements to allow you to organize the page in a grid, defining the rows and the number of columns to define the width of the element.

In healthcare facilities it is not necessarily the case that only large screens are used, but it is possible that smartphones or tablets are used to consult this information, and here also bootstrap offers us features that simplify the creation of responsive web pages, allowing us to define breakpoint, indicating the behavior of our page based on the size of the device. In order to provide a representative page, we leveraged custom CSS, redefining the style of some elements. In Figure 15, Figure 16, Figure 17, Figure 18, we are showing the primary page to select patient, and the Intrinsic Capacity Coverage.

Figure 15. Primary view to the patient data: From single measurement to calculated Intrinsic Capacity

Figure 16. Left: Calculated IC for a timeframe of study the status of a patient; Right: measured IC coverage for the selected patient

Figure 17. Visualizing the specific domain evolution; from the right panel it is possible to select the domain to visualize

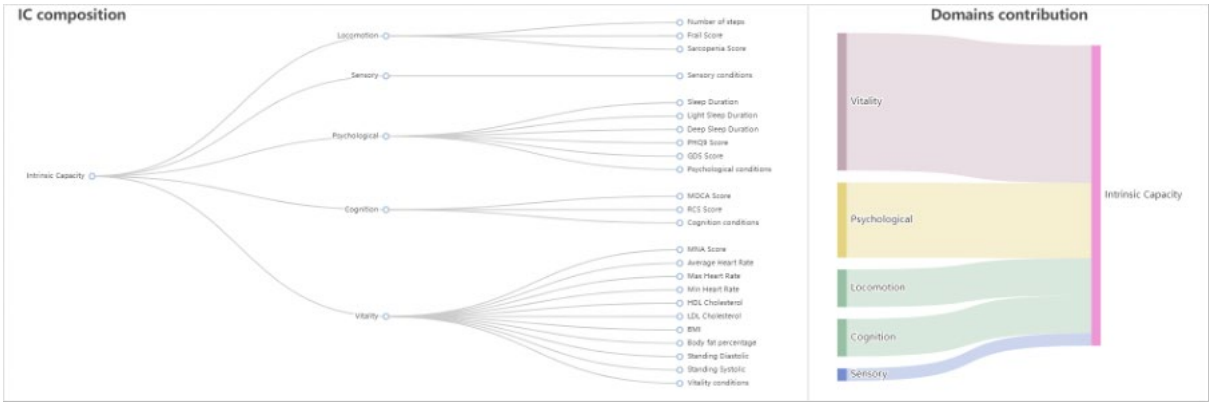

Figure 18. Figure 4: The composition of IC and the contribution of each domain in the final IC score

2.4 Analytics for Hearing Loss

This section concerns the clinical use case of the Hearing Loss scenario in the SMART BEAR project. In the Hearing Loss (HLoss) clinical use case several parameters could be improved: 1) increase the hours of usage, 2) increase the patient satisfaction with the Hearing Aid (HAid), 3) decrease the risk of dropping out, and 4) reduce the number of visits to the Audiologists' office for HA fine tuning. Previous deliverable, D4.4 considered a pipeline for predicting future HA usage, which considered all the required steps – from pre-processing the data, hyper-tuning the hyper-parameters, training the model with the selected best set of hyper-parameters, testing the model, and finally explaining the model. That pipeline, using an Encoder-Decoder NN with Attention mechanism, and the SHAP algorithm as the main post-hoc explanation of the NN model (LIME & DiCe were also evaluated) was published as "Predicting and Explaining Hearing Aid Usage Using Encoder-Decoder with Attention Mechanism and SHAP" at the 2022 16th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) [20]. The code itself is available at [21]. That work used synthetic data [12] that

had been produced by the EU Evotion project[14], one of the precursors of SMART BEAR. The Evotion Synthetic Data (EvoSynth) [12] has 399,500 observations of 53 participants.

Following that work, we considered the problem of designing a NN architecture that would be interpretable by design, so that no post-hoc explainability algorithm would be needed (which can by definition only approximate what the actual model has learnt). This strand of work led to the development of FocusLearn, which will be presented at the 2024 International Joint Conference on Neural Networks [22], as part of the IEEE World Congress on Computational Intelligence [23]. The full version of that paper (containing extra appendices too) is provided freely at arxiv.org [24]. Unlike the previous work (Attn-ED), FocusLearn used real data that had been collected by Evotion researchers [25] among the data sets that it considered. On all the data sets we considered, FocusLearn [24] surpassed comparable interpretable models (NAM, SPAM) in predictive power, and matched or even surpassed state-of-the-art non-interpretable models (LSTM, XGBoost), while providing simple explanations for what the underlying Neural Network (NN) had learnt that are in the same form as NAM. So FocusLearn [24] provides researchers with shape functions that inform them about the correlation between the outcome and each feature, along with an indication of how many data in the data set were used to train the model in each interval (and therefore how trustworthy the model can be in each interval). FocusLearn has been evaluated for both regression and classification tasks of multivariate time series — see [24] for more details and [26] for the code repository.

2.5 Federated Learning Framework

As described in Deliverable 4.2 Report on SMART BEAR Cloud Enabling Components v1, Federated learning is a machine learning technique that can train an algorithm across multiple systems or devices holding local datasets without the need of exchanging data. Instead of data, models and model parameters are exchanged. This approach is well suited as well for personalization of models to individual users, in a network of IoT devices and smart devices such as mobile phones or dedicated edge devices. Even when no geographic distribution of data is necessary, this architecture enables the separation of local datasets for training and supports the implementation of increased autonomy with respect to local model updates and contribution to the global model between the coordinator and the individual worker nodes.

The data management scenario in SMART BEAR is centralized, all data being collected in the repository hosted on the cloud. Therefore, we do not leverage the privacy and distribution benefits of the federated learning approach. The key benefit is in the ability to personalize to individuals and to their data. In SMART BEAR we develop techniques to tailor the federated learning solution towards individual user-level personalization. Specific extensions are required to assess the performance of the local and global models, to implement validation loops to help detect any performance degradation, to identify data outliers and make decisions on their use in the system, and to design patterns of data and parameters exchange between the workers responsible for the local models specific to individual patients and the coordinator responsible for the global model. This solution facilitates as well future scenarios in SMART BEAR, ensuring scalability when data from external systems becomes available for model development without the need to be transformed and loaded into the SMART BEAR repositories, or when additional cloud resources need to be leveraged for data storage or for computation.

[14] https://h2020evotion.eu

2.6 AI models and interventions in the CVD scenario

2.6.1 Models

There is an extensive number of different models and networks that one can use to create predictions [10]. Choosing the best suited model for a use case depends on multiple factors such as type of input, predictive strength, and explainability. For the cardiovascular disease (CVD) deliverables we use recurrent neural networks (RNNs) as well as Long Short-Term Memory (LSTM) models which have great predictive strength for time series input.

2.6.1.1 RNN

RNN (see figure 14) is a type of neural network that is preferred for time-series data [REF: https://towardsdatascience.com/understanding-recurrent-neural-networks-the-prefered-neural-network-for-time-series-data-7d856c21b759]. The RNN trains its units by adjusting their weights following a slight modification of a feedback process known as backpropagation: backpropagation through time (BPTT). This makes RNNs well-suited for tasks where context and past information are crucial factors. RNNs are able to produce more accurate predictions than traditional neural networks in many cases.

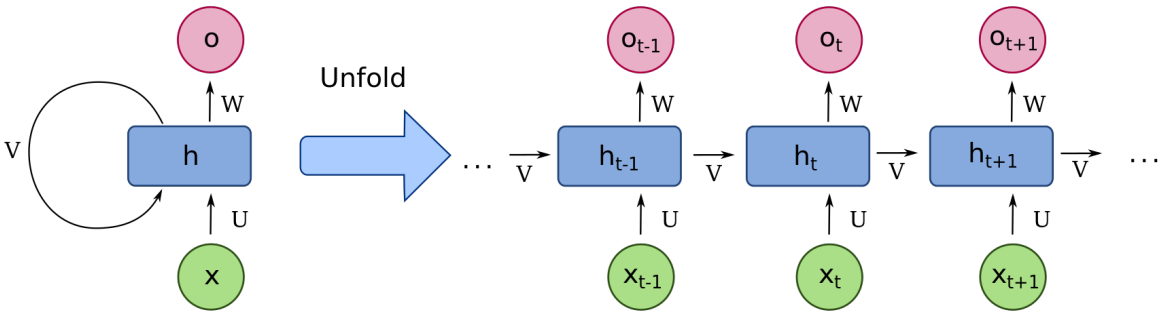

Figure 14 RNN Architecture [15]

2.6.1.2 LSTM

LSTM (Long Short-Term Memory, see figure 15) is a type of neural network that is designed to handle sequences of data, such as text, speech or continuous vitals [8]. Unlike traditional neural networks that can struggle to remember previous information, LSTMs have "“memory cells"” that allow them to store important information from earlier in the sequence. Additionally, LSTMs have "“gate units"” that regulate the flow of information into and out of the memory cells, enabling the network to selectively remember and forget information as needed. This makes LSTMs well-suited for tasks where context and past information are crucial factors. By incorporating these memory and control mechanisms, LSTMs are able to produce more accurate predictions than traditional neural networks in many cases.

[15] https://en.wikipedia.org/wiki/Recurrent_neural_network#/media/File:Recurrent_neural_network_unfold.svg

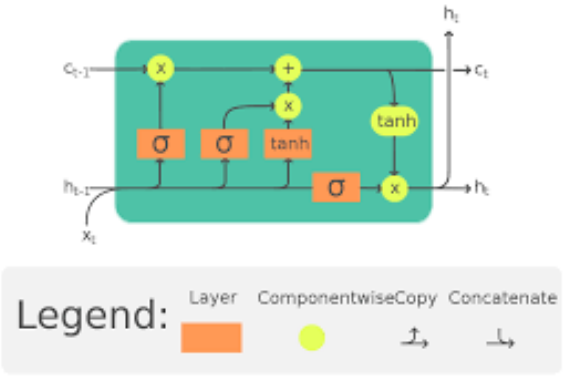

Figure 19 LSTM Cell Architecture [16]

2.6.2 Applications of these models to support specific use cases in SMART BEAR

2.6.2.1 Heart rate prediction (exercise vs. non-exercise)

High heart rates have shown to be a risk factor for patients with a cardiovascular disease [15]. Furthermore, research has shown that planning the optimal exercise using heart rates can lead to better health outcomes [16]. By monitoring heart rate during exercise and adjusting the intensity based on set targets, individuals can improve their health more efficiently and safely. RNN and LSTM models can be applied in this field by combining heart rate data with other relevant information, such as age but not limited to, continuous heart rates during exercises, weight, history of previous exercises and medical history, to create personalized exercise plans and predict the optimal heart rate zones for different individuals. This can help improve the effectiveness and efficiency of exercise programs, leading to better health outcomes. We developed three scripts using Python and Tensorflow:

- 1) a script with a LSTM model that uses FHIR data to predict the maximum heart rate. As input data we used the following data items: `effectiveDateTime`, Average Heart rate, Max heart rate, Min heart rate, Number of steps and Walking distance 24 hour Calculated. It uses a window to make prediction 14 time-units in the future, given 14 time-units of history.
- 2) a script that compares exercise with non-exercise FHIR data, to see if exercise has an effect on the heart rate. It can be concluded that, for most patients, physical exercise has a positive effect on the minimum heart rate: all patients with >1.75M steps have a lower minimum heart rate, and 80% of patients with >1M steps have a lower minimum heart rate.
- 3) a script with a RNN or LSTM model to predict the future heart rate of a single patient, using non-FHIR time series data, with timestamps and heart rates for every 15 seconds (from the `timeOffsetHeartRateSamples` data item).

These scripts will be made available as Jupyter notebooks for further research.

[17] https://holobalance.eu/

2.6.2.2 Sleep duration prediction

Sleep is an essential component of overall health and well-being. It plays a crucial role in repairing and rejuvenating the body, regulating hormones, and consolidating memories. Inadequate or poor-quality sleep can have negative impacts on physical and mental health, including weakened immunity, decreased cognitive function, and increased risk for chronic conditions such as obesity, diabetes, and cardiovascular disease. For patients, the importance of adequate and restful sleep cannot be overstated. By prioritizing sleep and addressing any sleep-related issues, patients can support their overall health and well-being. Neural tree models can be trained to predict the hours of sleep. When we can identify in each patient which variables are most important in predicting sleep time, recommendations can be provided if for example a patient should not exercise in the evening or should take a walk before going to bed etc. For this model to be trained we need data such as but not limited to, continuous heart rates, history of hours of sleep, age, weight, and medical history.

2.6.2.3 Deterioration prediction and Regimen adherence

Regimen adherence is a critical component in the successful management of a patient's health [17]. When a patient adheres to their prescribed treatment plan, it helps to ensure that they receive the full benefit of their medication, leading to improved health outcomes. Furthermore, regimen adherence can reduce the risk of adverse events and deterioration.

Neural Tree models can be used to predict if a patient is going to deteriorate or if a patient is adhering to the prescribed regimen. Continuous heart rates can be used as a proxy for labelling deteriorations and disobedience of the described regimen. Patient specific information such as but not limited to, continuous heart rates, history of hours of sleep, age, weight, and medical history is needed to create a well performing model.

2.6.3 Data

To train such models data with all relevant data elements and of good quality is needed. In order to predict the heart rate during an exercise (semi) continuous HR measurements are needed as well as the time at which an exercise is performed. Furthermore, other variables such as weight, drinking status, smoking status, hours of sleep, etc. could be used to increase the performance of such models. We have explored the data collected in the PoP to ensure that all the elements required for these models are collected at the needed level of detail and accuracy.

2.6.3.1 Current challenges

Data such as hours of sleep, smoke status and weight are already available in the data set. Even though a lot is already available there are still some issues we are trying to resolve. First, the overall quality of the data can be improved, problems are mainly related to the observations table, below are some examples that were found but the problems are not limited to those. For example, Figure 20 shows the walking distance of a patient over a period of 24 hours. As you can see there are multiple entries at the exact same data for the same patient with different values. The correct value should be the largest one as the other values are measured during the day, what should be fixed here is the timestamp of the other values. It is quite valuable information when a patient took a walk, definitely in combination with the heartrate.

	patient_id	effectiveDateTime	valueQuantity_value	code_coding_code	valueCodeableConcept_coding_code	valueCodeableConcept_coding_display	code_coding_display
4417	792607756	2021-11-26 01:00:00	11.466	41953-1	None	None	Walking distance 24 hour Calculated
4418	792607756	2021-11-26 01:00:00	4.194	41953-1	None	None	Walking distance 24 hour Calculated
4419	792607756	2021-11-26 01:00:00	1.842	41953-1	None	None	Walking distance 24 hour Calculated
4420	792607756	2021-11-26 01:00:00	4.088	41953-1	None	None	Walking distance 24 hour Calculated
4425	792607756	2021-11-27 01:00:00	5.726	41953-1	None	None	Walking distance 24 hour Calculated
4423	792607756	2021-11-27 01:00:00	0.038	41953-1	None	None	Walking distance 24 hour Calculated
4424	792607756	2021-11-27 01:00:00	1.611	41953-1	None	None	Walking distance 24 hour Calculated

Figure 20 Walking distance in the dataset

Secondly, Figure 21 shows aggregated heart rate values over the last 24 hours. As can be seen in the figure is that again there are multiple entries on the same exact moment. For maximum and minimum this is not a big problem as we can take the maximum of the Max and the minimum of the Min but for the average there is no way of knowing what the actual average is.

473	32578327	2022-01-14 01:00:00.000	137.000	55426-1	None	None	Max heart rate
552	32578327	2022-01-14 01:00:00.000	128.000	55426-1	None	None	Max heart rate
798	32578327	2022-01-14 01:00:00.000	70.000	55425-3	None	None	Average Heart rate
1060	32578327	2022-01-14 01:00:00.000	71.000	55425-3	None	None	Average Heart rate
1100	32578327	2022-01-14 01:00:00.000	86.000	55425-3	None	None	Average Heart rate
1660	32578327	2022-01-14 01:00:00.000	53.000	96935-2	None	None	Min heart rate
1724	32578327	2022-01-14 01:00:00.000	64.000	96935-2	None	None	Min heart rate
1843	32578327	2022-01-14 01:00:00.000	48.000	96935-2	None	None	Min heart rate

Figure 21 Aggregated heart rate values

Additionally, the observations table is populated with an extreme amount of NaN values, in the observations table there are 67 columns of which on average 53 are filled with NaN values.

Finally, for the advanced machine learning models semi continuous heart rate data is needed which is not yet available in the data set but is measured by the wearable. Without these variables the previously described models cannot be trained.

2.6.3.2 Possible solutions

A few simple steps can be taken to improve the quality of the data. First, providing the exact moment of when a patient was walking will solve the problem of multiple entries. Secondly, only adding the correct aggregated heart rate values in the data set will also improve the data quality. Additionally, taking a look at what columns are needed and not needed and removing the columns with only NaN values will also improve the quality of the data. Finally, and most importantly, the continuous heart rate values are collected, but not sent to the FHIR-based repository. So, by either creating a FHIR resource and extending the repository or saving the values in a different kind of format in another repository so they can be used for analytics is of the highest priority.

Furthermore, in order to collect the walked distance and speed we can collect the differences in locations between two time points (close together). This information is needed to better correlate heart rate values with exercising.

2.6.3.3 Chosen solutions

After discussing with the team, we decided to store the semi-continuous heart rates (as recorded from the devices) in a non-FHIR repository as it is not relevant for the dashboard. For advanced analytics the data is not needed to be accessed in real time and the database can iteratively be built up. Therefore, storing it in a non-FHIR repository where it can be retrieved in an on-demand fashion was chosen as the most efficient solution. The non-FHIR data consists of heart rate data for every 15 seconds, together with data items such as activeKilocalories, activeTimeInSecondsnumber, distanceInMeters, steps and stressLevel. This data is much more useful for the model doing heart prediction during exercise.

2.7 Balance and Cognitive Analytics

Due to its size, this part has been submitted as a separate document. Please see the separate document "D4.6 Report on SMART BEAR – Balance and Cognitive Analytics".

2.8 Dashboard

The Dashboard of SMART BEAR is the main UI interface for clinicians. The graphical interfaces allow the registration of the patients, the monitoring of the measurements by the devices, the investigation of the analytics, along with the interventions and alerts for each of the patients that participate in the pilot studies. In order to support the clinicians, an online support guide is online at https://smartbear-unimi.github.io/sb-ui-doc/docs/.The access to pages depends on the role, which should be chosen when the users have their accounts activated. For example, the SysAdmin type user cannot view the clinical data, while the clinician cannot manage users within the system. The TechSupport type user can view patient data, but can monitor the sending of data and notify managers to check that there are no communication problems that could affect the sending of data in the system.

In this section, an overview of the main improvements is provided. More information is available in D4.5. Also, some screenshots of the main tabs are provided in the Annex.

2.8.1 Staging version

From M49, the staging version of the app is online for clinicians and technical personnel, for training and troubleshooting purposes at this link: https://cloud-staging.smart-bear.eu/.

The following roles are currently supported:

- System Administrator: This role grants access to GDPR requests and user approvals.
- Clinical Case Manager: This is the most common role, assigned to clinicians, and provides access to the following pages: Home, Patients, GDPR, Inbox, and Tech Support.
- Help Desk Operator: This role grants access to the Home and Tech Support pages and allows technical teams to monitor inactivity time.

2.8.2 Baseline Assessment

In this section of the UI, the clinician enters the data that will be mapped into FHIR-type resources and then saved in the repository via REST calls (see **Errore. L'origine riferimento non è stata trovata.**).

Obviously, also in this case each call passes through the Security Component which takes care of translating the ID and avoiding having a link between sensitive data and the patient's ID.

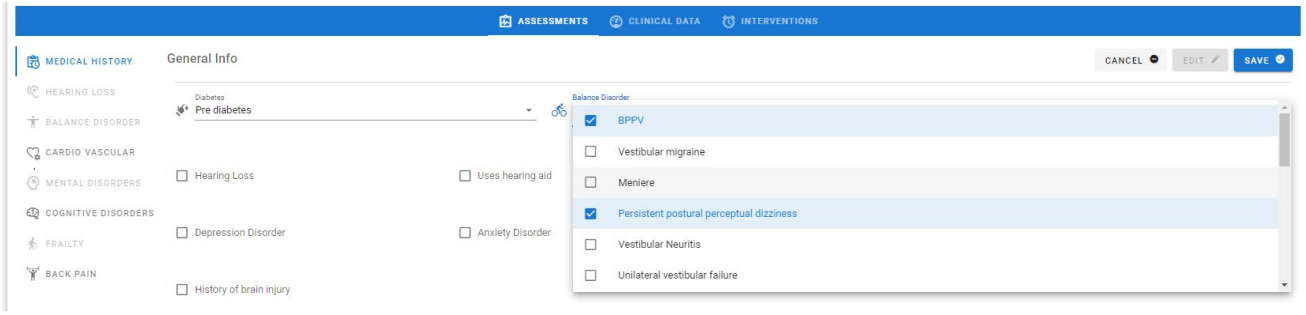

Figure 22. Medical History

2.8.3 Adverse events

The Adverse events form is being developed, which will be a tab for clinicians to record any actual or potential medical care, research study or other event causing physical injury. The tab will be activated when the patient is officially enrolled in the project. This subsection can show the past adverse events linked to the patient and provide a form where the clinicians can modify or insert a new event of this type. The clinician will be capable of inserting a suspect entity to be associated to this type of event, the suspected entities can be either a device or a medication (see **Errore. L'origine riferimento non è stata trovata.**, **Errore. L'origine riferimento non è stata trovata.** and **Errore. L'origine riferimento non è stata trovata.**).

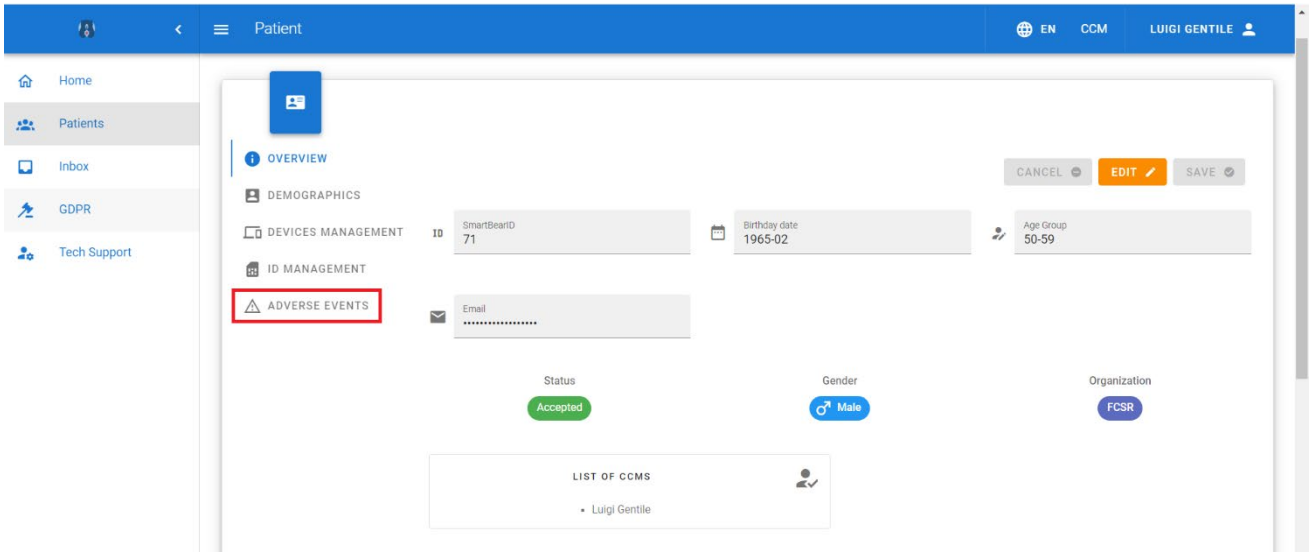

Figure 23. Adverse events button

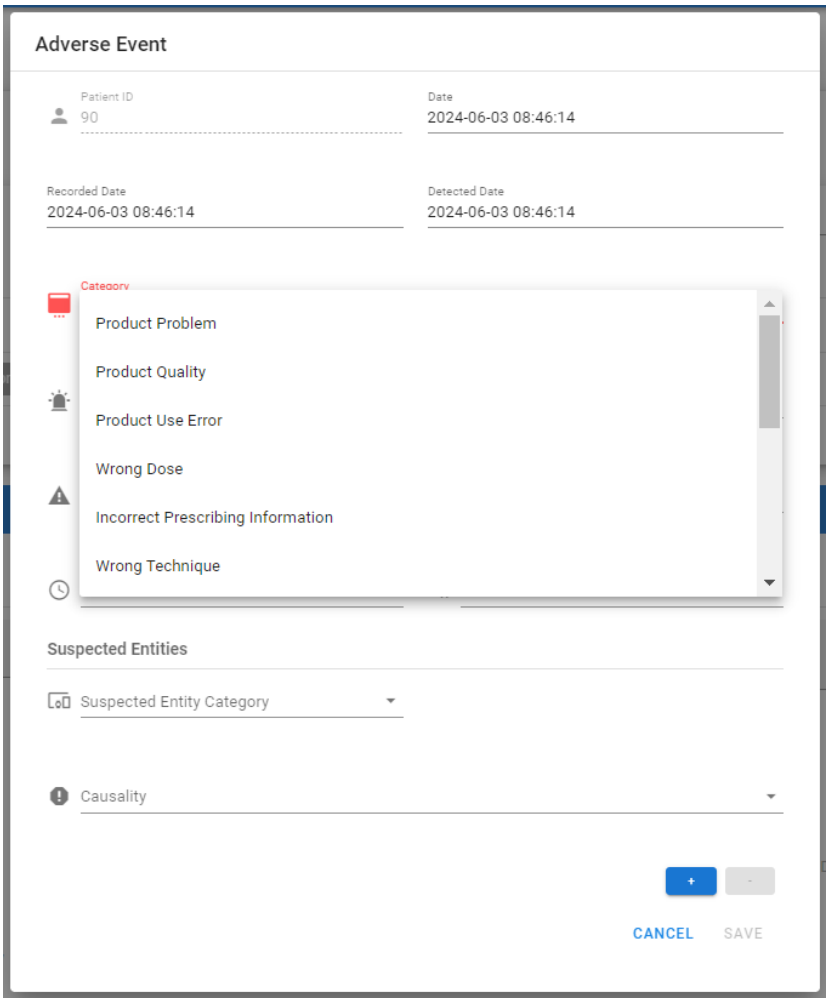

Figure 24. Adverse events tab

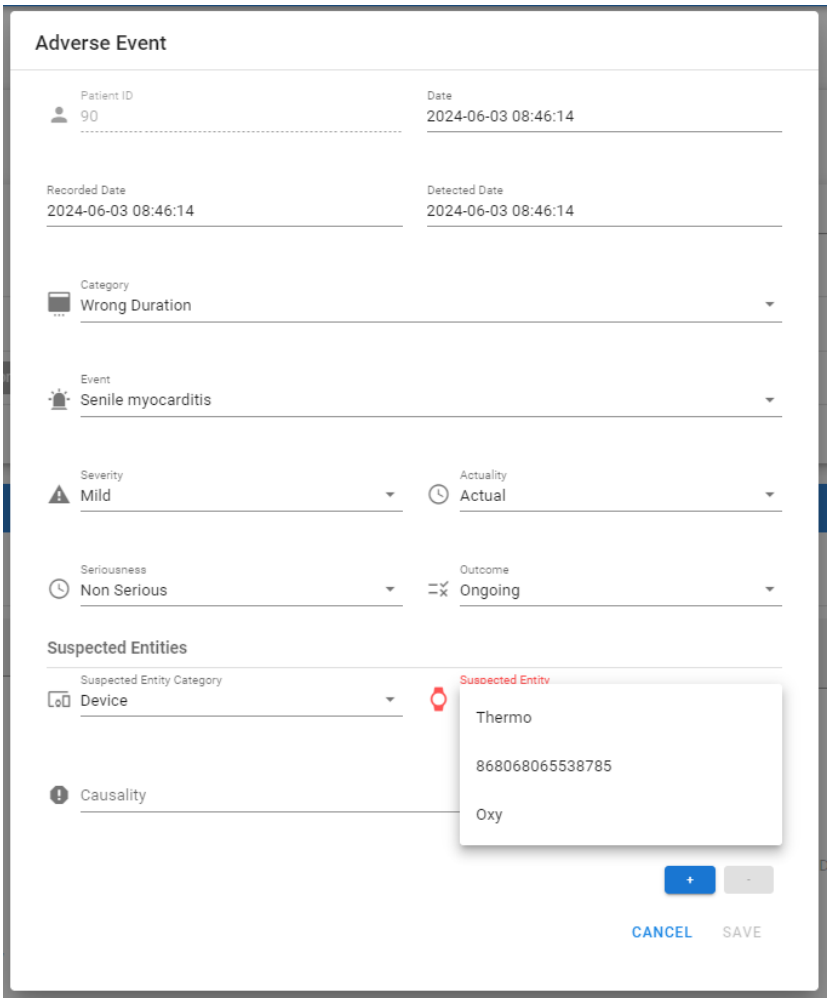

Figure 25. Suspect entity management

3 Concluding Remarks

This deliverable provides the specification of the final versions of the relevant SMART BEAR infrastructure components that compose the central cloud platform, including the data model, the implementation guide, the Big Data Engine and the analytics components.

The FHIR data model and the Implementation Guide were extended to support additional data for the hearing loss use case.

We also present the current analytics results with SMART BEAR data, the analysis we carried out for the data collected and its ability to support the desired interventions described in the clinical use cases. For some planned analytics we identified issues with the data for which we have implemented a solution that enabled us to develop those models with the relevant data. The models will be further refined in the next reporting period, using the large datasets collected by the pilots.

Our solutions are in line with recent EU publications around advanced digitalisation of research [27] and FAIR data productivity [28].

4 References

[1] P. Katrakazas, T. Androutsou, A. Smith, J. Milas, D. Brdarić, L. Murdin, N. Nina Koloutsou, A. Dudarewicz, M. Pawlaczyk-Łuszczyńska, M. Śliwińska-Kowalska, N. H. Pontoppidan, L. Trenkova, G. Kaloyanova, N. Tsokova, D.-E. Bamiou, G. Dritsakis, A. Bibas and D. Kikidis, "Public Health Policy Decision Models (PHPDM) v1, Deliverable D3.1 to the EVOTION-727521 Project funded by the European Union," ICCS, Greece, 2017.

[2] P. Katrakazas, D. Koutsouris, K. Koloutsou, A. Smith, G. Spanoudakis, K. Pozdniakov, L. Murdin, M. Sladen, D. Brdaric, S. Samardžić, I. M. Huskić, M. Kowalska, J. H. Christensen, L. Trenkova, D.-E. Bamiou, G. Dritsakis, M. Anisetti, A. Bibas and D. Kikkidis, "Public Health Policy Decision Models (PHPDM) v2, Deliverable D3.2 to the EVOTION-727521 Project funded by the European Union," ICCS, Greece, 2019.

[3] I. Basdekis, Y. Bin, G. Spanoudakis, N. Dimakopoulos, P. Kanelopoulos, P. Kokkinakis, M. Smyrlis, P. Katrakazas, N. H. Pontoppidan and L. H. Tietz, "Data Repository and Collection Components, Deliverable D5.2 to the EVOTION-727521 Project funded by the European Union," City, University of London, London, 2017.

[4] W. Zucchini, I. L. MacDonald and R. Langrock, Hidden Markov models for time series: an introduction using R, CRC Press, 2017.

[5] I. L. MacDonald and W. Zucchini, Hidden Markov and Other Models for Discrete-valued Time Series, London: Chapman and Hall, 1997.

[6] L. R. Rabiner, "A tutorial on hidden Markov models," *In IEEE,* vol. 77, no. 2, pp. 257-286, 1989.

[7] G. D. Forney, "The Viterbi algorithm,," *In IEEE,* vol. 61, pp. 268-278, 1973.

[8] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306.

[9] Chen, Yuehui, et al (2005). "Time-series forecasting using flexible neural tree model." *Information sciences* 174.3-4: 219-235.

[10] Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.

[11] J. H. Christensen et al., "Fully Synthetic Longitudinal Real-World Data From Hearing Aid Wearers for Public Health Policy Modeling," Zenodo https://doi.org/10.5281/zenodo.2668210. 2019.

[12] J. H. Christensen et al., "Fully Synthetic Longitudinal Real-World Data From Hearing Aid Wearers for Public Health Policy Modeling," Front Neurosci, vol. 13, no. 850, 2019.

[13] A. Vaswani et al., "Attention Is All You Need," Jun. 2017.

[14] Francesca Manni, Aleksandr Bukharev, Anshul Jain, Shiva Moorthy, Asif Rahman, Anca Bucur:
Neural gradient boosting in federated learning for hemodynamic instability prediction: towards a distributed and scalable deep learning-based solution. AMIA 2022

[15] Perret-Guillaume, C., Joly, L., & Benetos, A. (2009). Heart rate as a risk factor for cardiovascular disease. *Progress in cardiovascular diseases*, *52*(1), 6-10.

[16] Karvonen, J., & Vuorimaa, T. (1988). Heart rate and exercise intensity during sports activities: practical application. *Sports medicine*, *5*, 303-311.

[17] Ingersoll, K. S., & Cohen, J. (2008). The impact of medication regimen factors on adherence to chronic treatment: a review of literature. Journal of behavioral medicine, 31, 213-224.

[18] Hughes, R.A.; Heron, J.; Sterne, J.A.; Tilling, K. Accounting for missing data in statistical analyses: multiple imputation is not always the answer. International journal of epidemiology 2019, 48, 1294–1304.

[19] Madley-Dowd, P.; Hughes, R.; Tilling, K.; Heron, J. The proportion of missing data should not be used to guide decisions on multiple imputation. Journal of clinical epidemiology 2019, 110, 63–73.

[20] Q. Su and E. Iliadou, "Predicting and Explaining Hearing Aid Usage Using Encoder-Decoder with Attention Mechanism and SHAP," 2022 16th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Dijon, France, 2022, pp. 308-315, doi: 10.1109/SITIS57111.2022.00053.

[21] Q. Su, "AttnED" https://github.com/qisuqi/Attn_ED (visited Apr. 2024)

[22] 2024 International Joint Conference on Neural Networks, https://www.inns.org/wcci-2024-call-for-papers (visited Apr., 2024)

[23] IEEE World Congress on Computational Intelligence, https://2024.ieeewcci.org/ (visited Apr. 2024)

[24] Q. Su, C. Kloukinas, and A. Garcez "FocusLearn: Fully-Interpretable, High-Performance Modular Neural Networks for Time Series", https://arxiv.org/abs/2311.16834 (visited Apr. 2024)

[25] J. H. Christensen, G. H. Saunders, M. Porsbo, and N. H. Pontoppidan 2021. "The everyday acoustic environment and its association with human heart rate: evidence from real-world data logging with hearing aids and wearables" R. Soc. Open Sci. 8201345 http://doi.org/10.1098/rsos.201345

[26] Q. Su, "FocusLearn" Code repository https://github.com/qisuqi/FocusLearn (visited Apr. 2024)

[27] European Commission, Directorate-General for Research and Innovation, Opinion paper on advanced digitalisation of research, Publications Office of the European Union, 2024, https://data.europa.eu/doi/10.2777/932733

[28] European Commission, Directorate-General for Research and Innovation, Opinion paper on FAIR data productivity, Publications Office of the European Union, 2024, https://data.europa.eu/doi/10.2777/49194

[29] Valerio Bellandi, Paolo Ceravolo, Jonatan Maggesi, and Samira Maghool. 2024. Data management for continuous learning in EHR systems. ACM Trans. Internet Technol. https://doi.org/10.1145/3660634

[30] Michael Armbrust, Ali Ghodsi, Reynold Xin, and Matei Zaharia. 2021. Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. In Proceedings of CIDR, Vol. 8

[31] Valerio Bellandi, Paolo Ceravolo, Samira Maghool, and Stefano Siccardi. 2022. Toward a general framework for multimodal big data analysis. Big Data 10, 5 (2022), 408–424.

[32] Marko Luksa. 2017. Kubernetes in action. Simon and Schuster.

[33] Duane Bender and Kamran Sartipi. 2013. HL7 FHIR: An Agile and RESTful approach to healthcare information exchange. In Proceedings of the 26th IEEE international symposium on computer-based medical systems. IEEE, 326–331.

[34] Francesco Bagattini, Isak Karlsson, Jonathan Rebane, and Panagiotis Papapetrou. 2019. A classification framework for exploiting sparse multi-variate temporal features with application to adverse drug event detection in medical records. BMC medical informatics and decision making 19, 1 (2019), 1–20.

[35] Matteo Cesari, Islene Araujo de Carvalho, Jotheeswaran Amuthavalli Thiya- garajan, Cyrus Cooper, Finbarr C Martin, Jean-Yves Reginster, Bruno Vellas, and John R Beard. Evidence for the domains supporting the construct of in- trinsic capacity. The Journals of Gerontology: Series A, 73(12):1653–1660, 2018.

[36] Valerio Bellandi, Paolo Ceravolo, Ernesto Damiani, Samira Maghool, Matteo Cesari, Ioannis Basdekis, Eleftheria Iliadou, and Mircea Dan Marzan. A methodology to engineering continuous monitoring of intrinsic capacity for elderly people. Complex & Intelligent Systems, 8(5):3953–3971, 2022.

[37] Emmanuel Gonzalez-Bautista, S Andrieu, LM Guti´errez-Robledo, RE Garc´ıa- Chanes, and P de Souto Barreto. In the quest of a standard index of intrinsic capacity. a critical literature review. The journal of nutrition, health & aging, 24:959–965, 2020.

5 Annexes

5.1 Annex 1: Dashboard screenshots

5.1.1 Registration and login

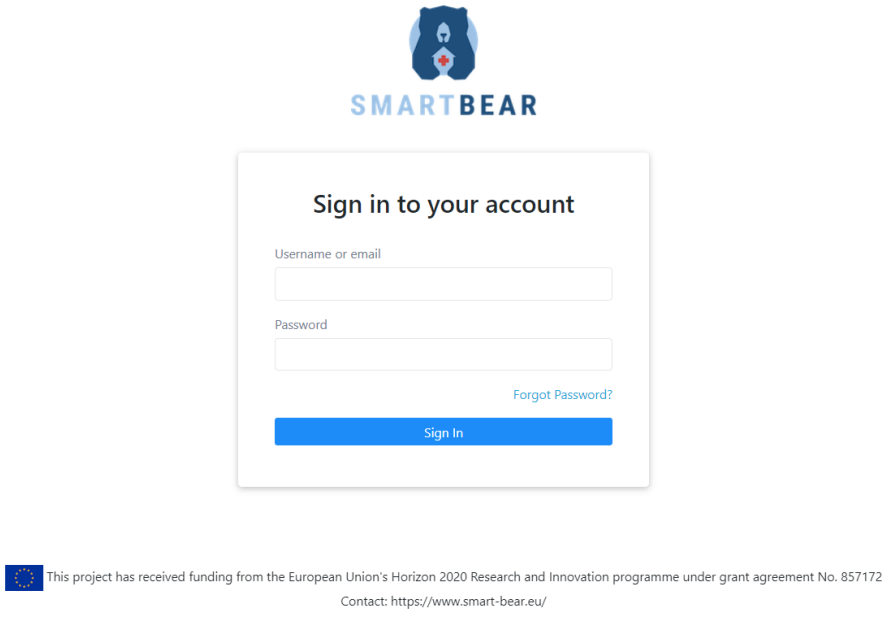

Figure 26. The login form

Upon the platform migration to Kubernetes was completed, the registration procedure requires contacting STS to request, and user do not have to fill the registration form anymore. In fact, STS is responsible to activate the new accounts. The login page was slightly modified, and a Reset password functionality was added to recover the credentials (see Figure 26).

5.1.2 Patient's registration

The patient page allows to create and manage the patients. In order to register a patient, a form must be filled with data including the responsible CCM. In caseNo personal data must be collected from the patient.

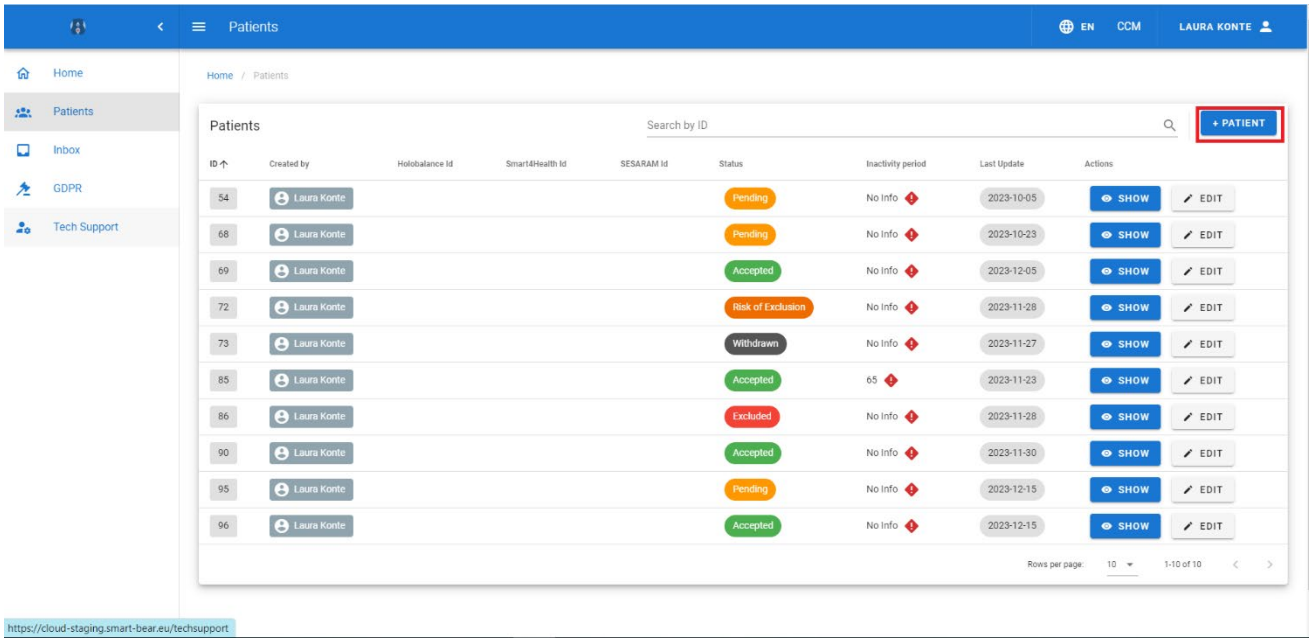

Figure 27. The home page

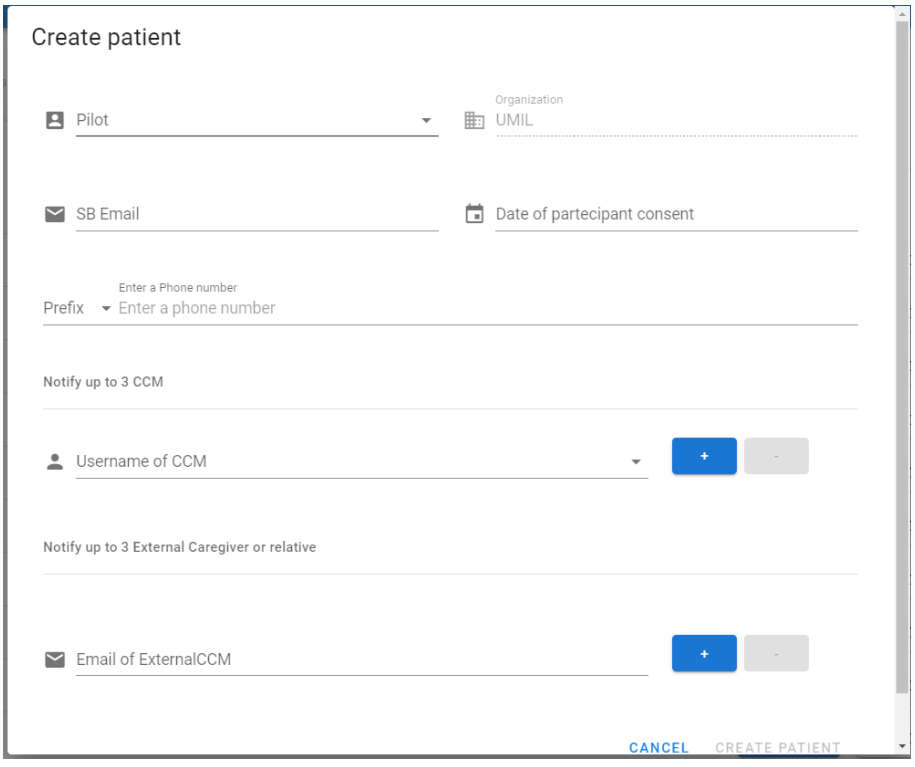

Figure 28. Create patient form

After the patient is registered and some demographics are inserted it is also necessary to carry out a pre-test, in order to verify if the patient meets the requirements of the protocol and for this, they must pass a MoCA Test and show that they have a certain level of Dexterity.

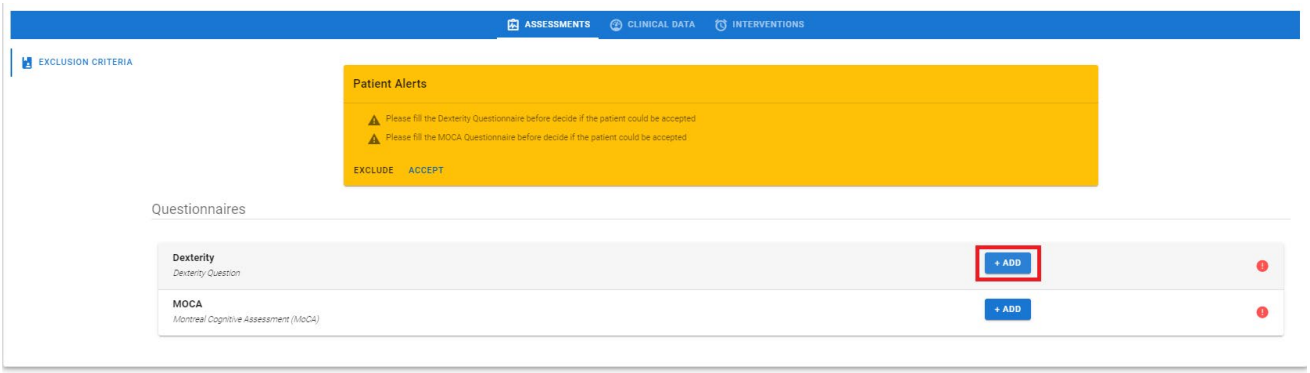

Figure 29. Exclusion criteria

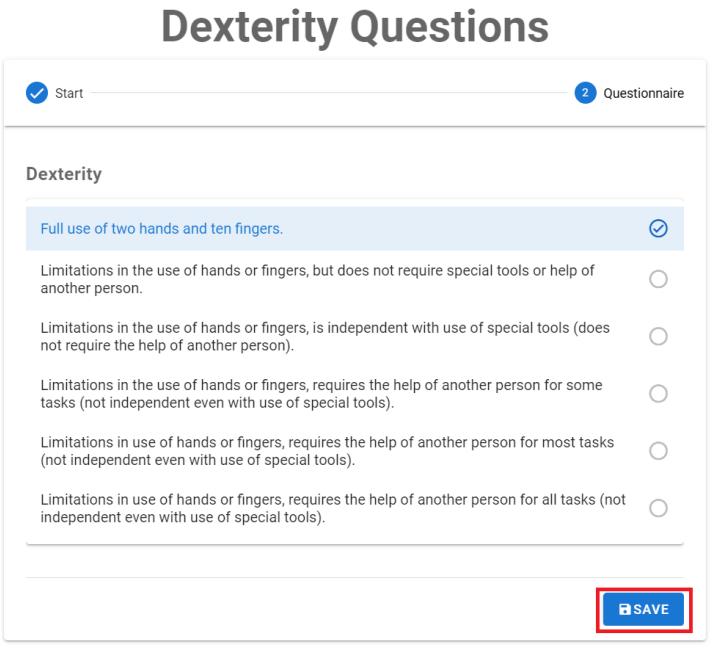

Figure 30. Dexterity questionnaire

After the clinician verifies the exclusion criteria, a popup appears on the screen, to confirm whether the patient has passed the tests or exclusion is suggested.

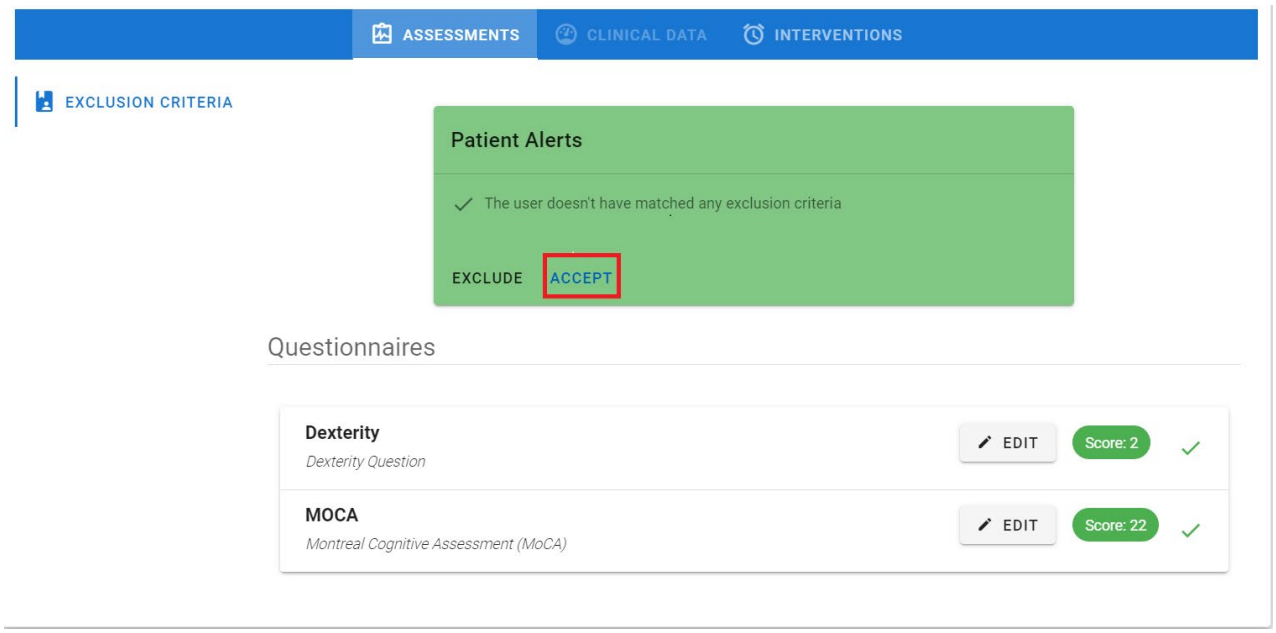

Figure 31. Exclusion criteria are matched

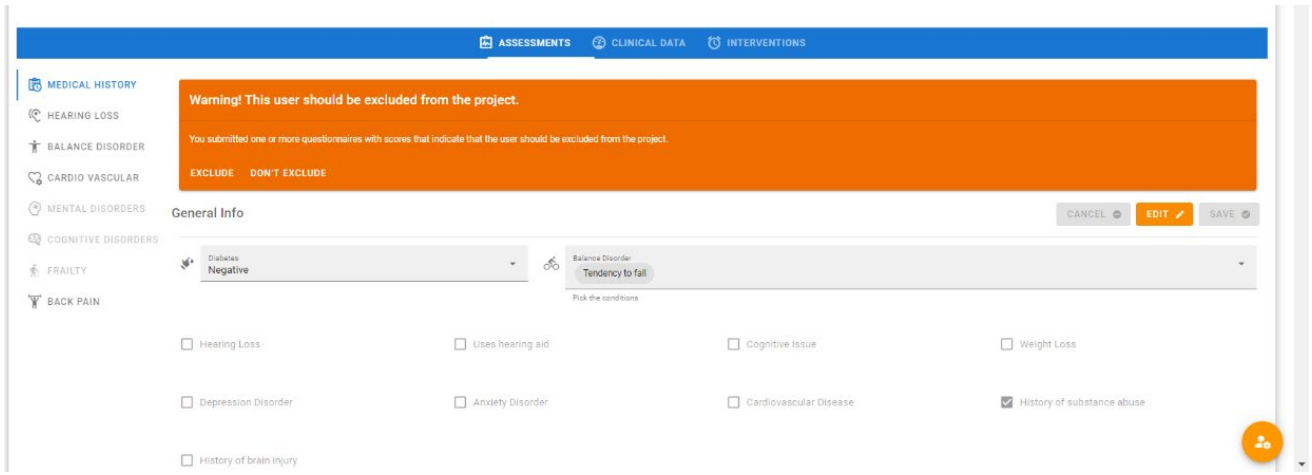

Figure 32. Risk of exclusion alert

5.1.3 Baseline Assessment

During the Baseline Assessment, a profiling functionality is available for the users to confirm which comorbidities will be monitored. There is an orange button in the lower right corner which opens a form, where the proposed profile of the patient is shown: select/unselect the comorbidities to be monitored, then click on ACCEPT. Consequently, the specific assessment tabs will be activated according to the selection.

Figure 33. Exclusion criteria are matched

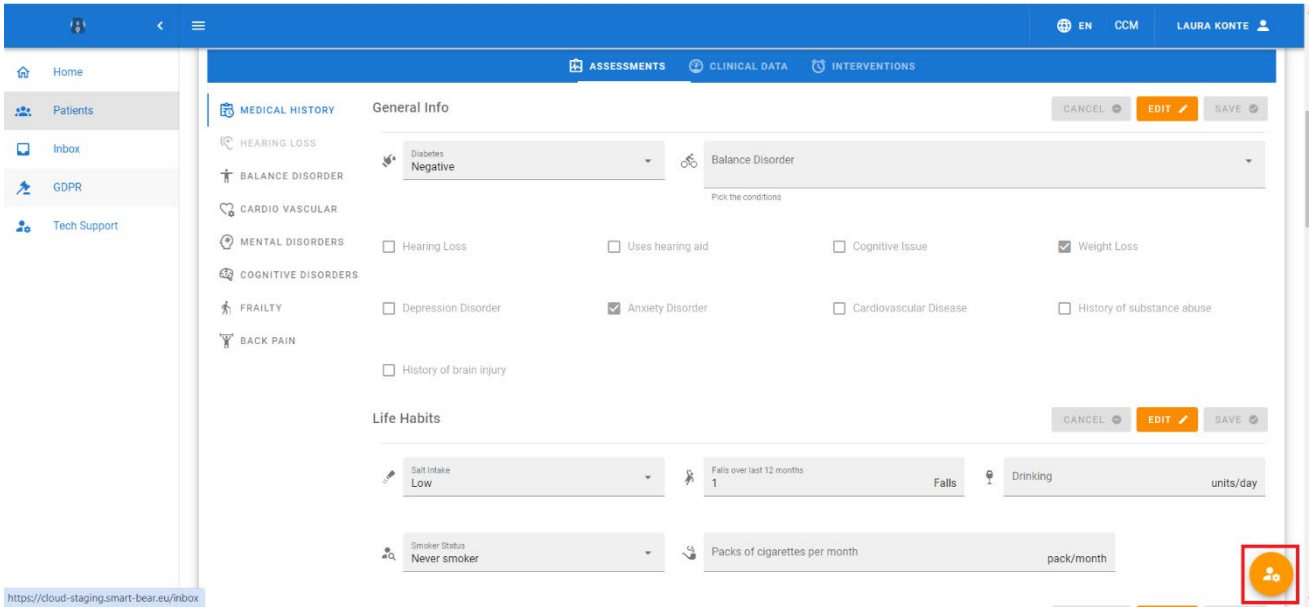

Figure 34. Profiling shortcut

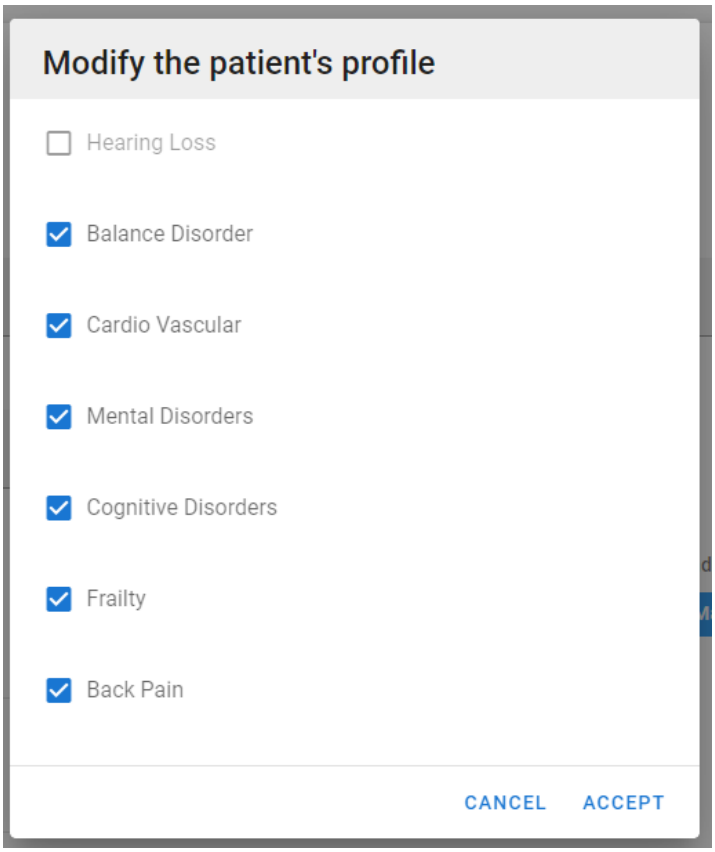

Figure 35. Patient profiling

Based on the clinical profile, the specific tabs and devices are activated.

5.1.4 Patient management

The dashboard supports the patient management by providing the following functionalities.

- EDIT PATIENT. It allows a clinician to edit the patient status to "ACCEPTED" to "EXCLUDED" or "WITHDRAWN".
- ID MANAGEMENT. In case a patient participates to a synergy, for example HOLOBalance or Smart4Health, a specific ID is assigned to him/her and can be inserted and modified.
- DEVICE MANAGEMENT. This tab is available clinicians and help desk operators to add and manage the devices that a patient receives after the enrollment. The available device types are available according to the profile selected.

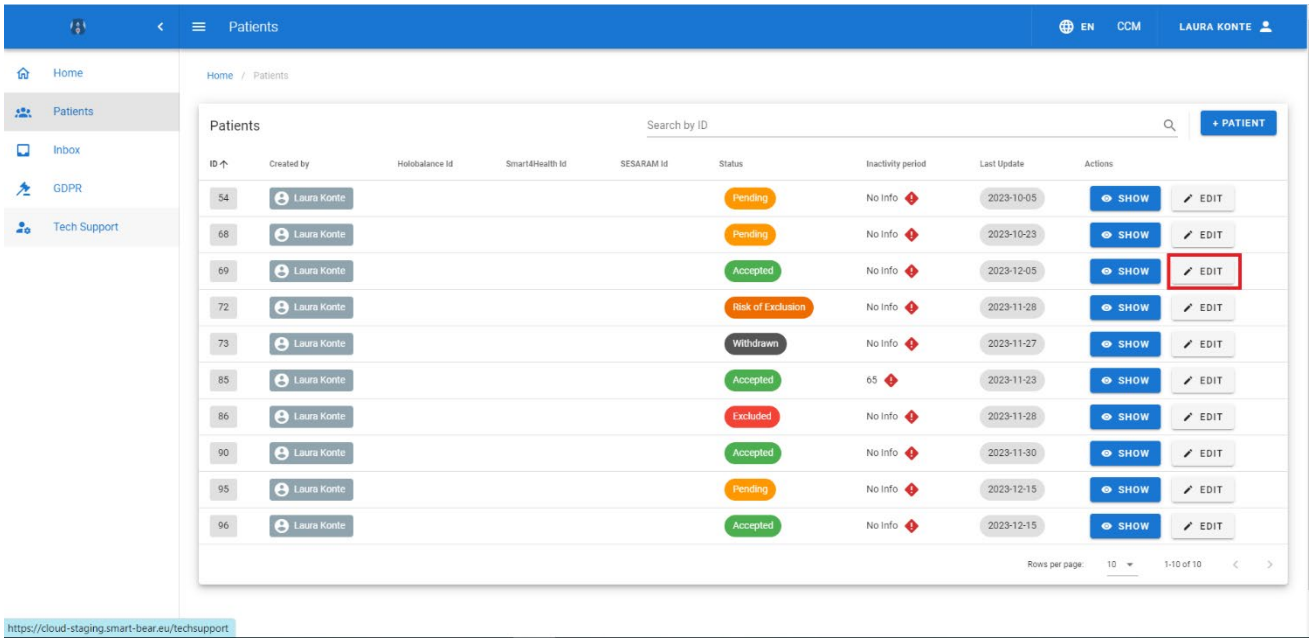

Figure 36. Edit a patient

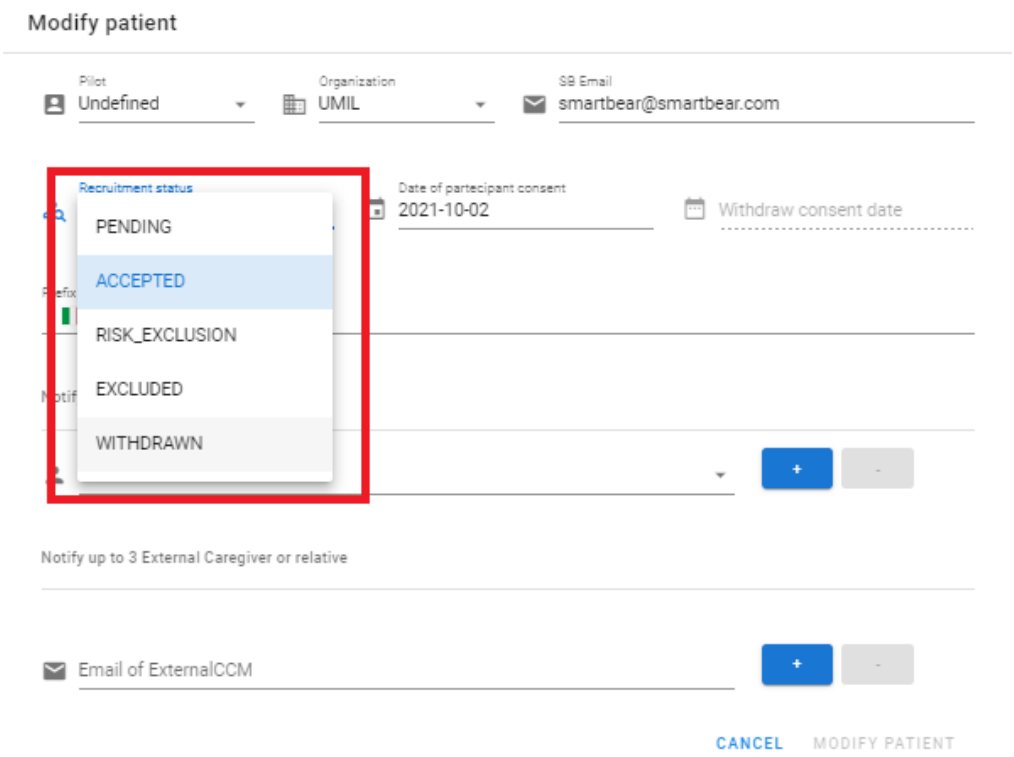

Figure 37. Change the patient's status

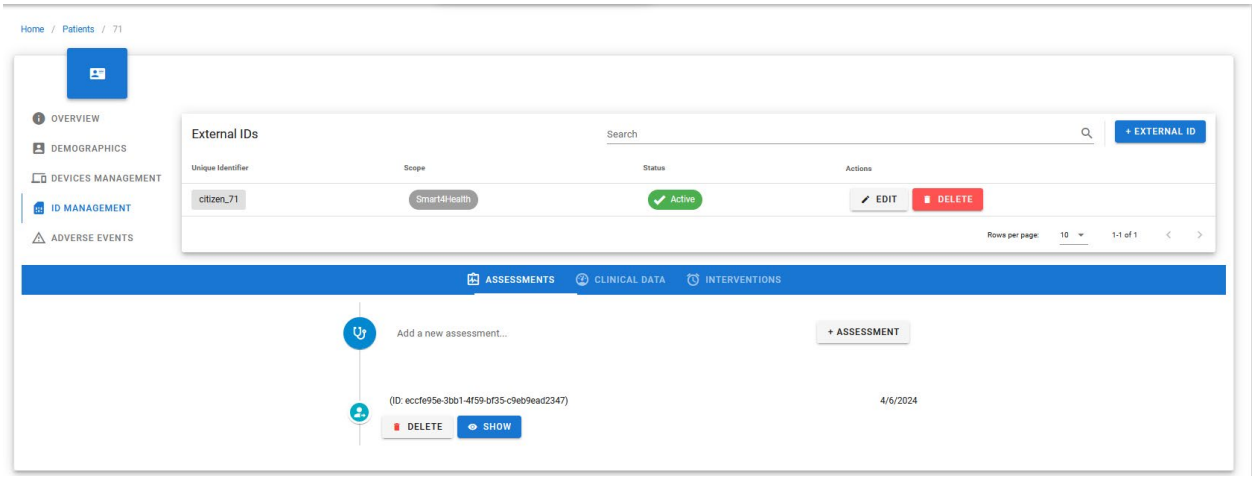

Figure 38. ID management

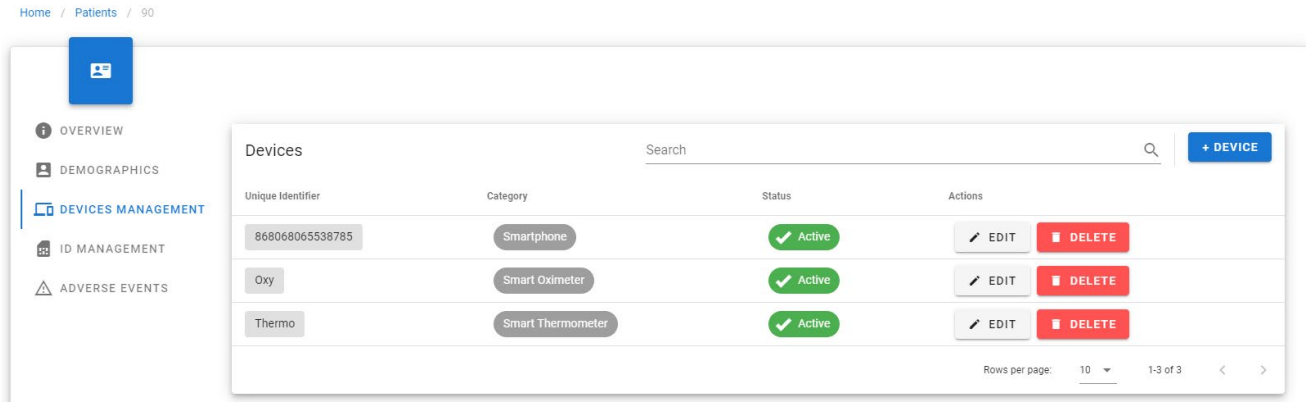

Figure 39. Devices management

5.1.5 Tech support

The Tech Support page is a new page that provides a functionality for clinicians and help desk personnel for monitoring the ways the device, the patient's activity and the inactivity time, which can hint at possible problems related to be examined by the technical team. This page also displays a button redirecting the users to the device management page.

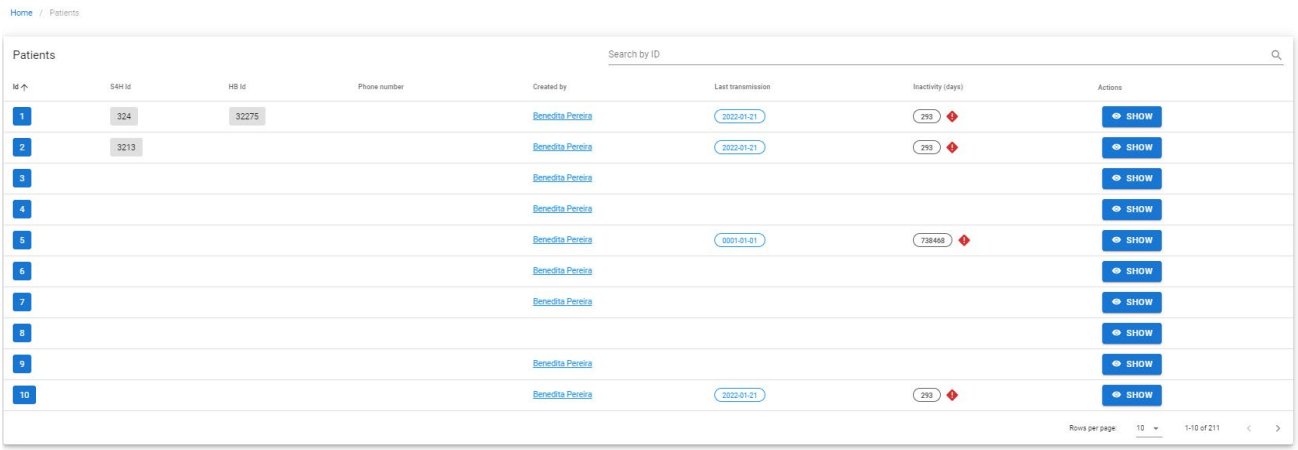

Figure 40. The Tech Support page

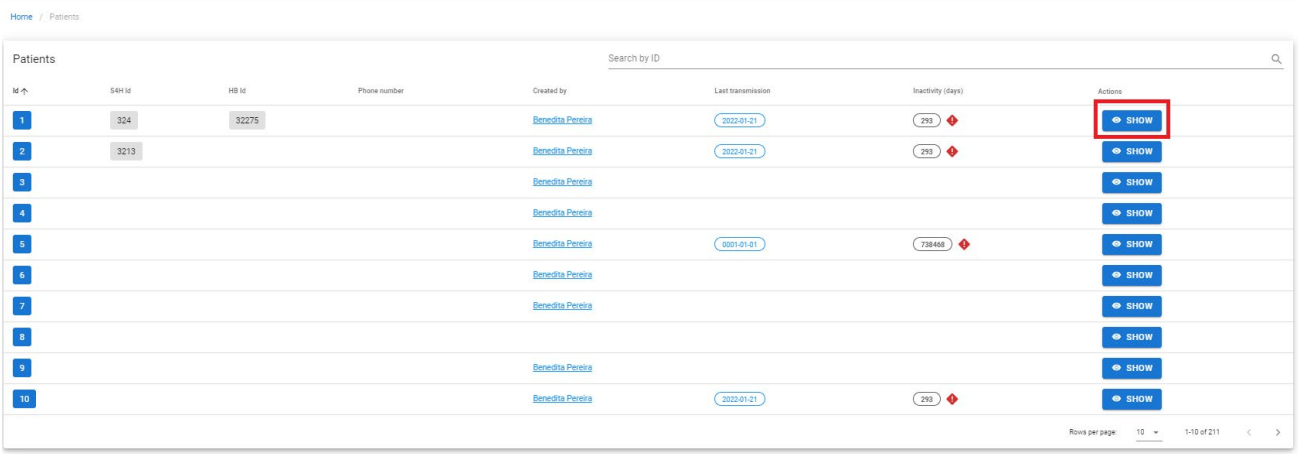

Figure 41. Tech Support. Press the SHOW button

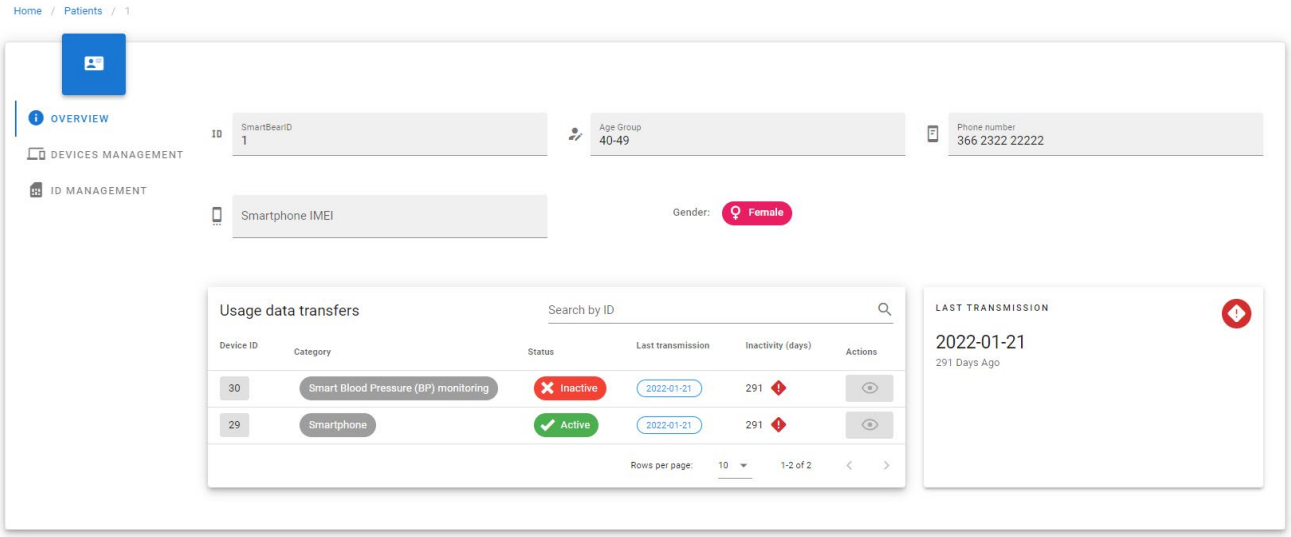

Figure 42. Tech Support. The Overview tab

5.2 Annex 2: Holobalance Analytics

The Holobalance[17] solution, acting as a synergy and now part of the SMART BEAR solution, marks a significant stride in assisting participants with balance disorders. This collaborative effort has led to the development of additional analytics including metrics tailored to every exercise conducted throughout the comprehensive 6-week program. These metrics serve as crucial indicators, providing insights into the progress and efficacy of the rehabilitation process.

Integral to these advancements are the two newly devised models. The first model accurately predicts the final score of balance exercises at the end of the 16th session, utilizing data from the initial sessions as input. This predictive capability empowers practitioners to identify and intervene effectively with participants whose predicted scores indicate potential challenges in their rehabilitation journey. Meanwhile, the second model forecasts the score of cognitive exercises in upcoming sessions based on historical data, offering practitioners a valuable tool for planning, and tailoring cognitive enhancement strategies.

Behind the scenes, specific attention has been paid to refining the modeling pipelines for each exercise. These pipelines encompass comprehensive strategies for handling missing values and outliers, uncovering correlations, and wisely selecting features. Such attention ensures that the models are robust and capable of providing meaningful insights to enhance the effectiveness of the Holobalance solution in assisting individuals with balance disorders.

This chapter encapsulates the demonstration of the aforementioned analytics, whether descriptive or predictive, utilizing data from the Pilot of pilots in Madeira and subsequently validating them against the data derived from the Athens pilot conducted within the Holobalance project. The analysis presented herein elucidates the methodology employed, which serves as a pivotal component within the SMART BEAR solution, particularly in the realm of balance analytics. This approach is instrumental in fulfilling one of the project's main objectives of facilitating the widespread management of balance disorders, not only within participants' homes via Holokits but also within clinical settings through Holoboxes.

5.2.1 Motivations and Objectives in Expanding Holobalance Solution

Balance is crucial for maintaining body orientation in various conditions. Impaired balance is a key cause of falls, a significant public health concern, especially among older adults. Around 35% of people aged 65 and over experience falls annually, leading to injuries and hospital admissions. Falls contribute to 40% of injury-related deaths and can result in a post-fall syndrome, affecting psychological and physical functions in the elderly. Recent technological advancements have introduced wearable solutions to assess and manage balance issues in patients with neurological disorders [1].

Kim *et al*. [2] explore post-stroke gait function in early rehabilitation, highlighting the importance of evaluating asymmetry and variability alongside walking speed. Through principal component analysis, three key components are identified: "Pace-Phase," focusing on speed, stride length, and double support phase, "Asymmetry" particularly temporal asymmetry in the single-support phase and

[17] https://holobalance.eu/

"Variability" associated with stride duration and stance phase on the more affected side. These findings offer crucial insights for tailoring rehabilitation strategies.

Silva-Batista *et al*. [3] investigate the feasibility and effectiveness of integrating Mobility Rehab, a sensor-based feedback system, into physical therapy for older adults with gait impairments. The system utilizes inertial sensors to provide real-time feedback on five gait metrics: step duration, stride length, elevation at mid-swing, arm swing range-of-motion (ROM), and trunk coronal ROM, aiming to enhance gait performance and balance confidence.

Zobeiri *et al*. [4] quantify head motion kinematics in individuals with vestibular schwannoma (VS) during prescribed balance stability exercises. Correlations between head motion during exercises and clinical measures in preoperative VS subjects suggest the potential for using head motion quantification to guide the assessment and prescription of balance exercises in this population.

In conclusion, the studies discussed underscore the significance of addressing balance impairments across various demographics, from post-stroke patients to older adults and individuals with neurological conditions. Advanced technologies, such as wearable sensors and feedback systems, offer promising avenues for assessing and managing balance issues in rehabilitation settings such as the Holobalance solution. By identifying key components like pace-phase, asymmetry, and variability in gait function, researchers can tailor rehabilitation strategies to target specific needs and improve outcomes. Moreover, integrating sensor-based feedback systems into physical therapy demonstrates potential in enhancing gait performance and boosting confidence in older adults. Furthermore, quantifying head motion during balance exercises in populations like vestibular schwannoma patients highlights the utility of objective measurements in guiding personalized rehabilitation programs. Overall, these findings contribute valuable insights into advancing balance assessment and rehabilitation methods, ultimately striving to mitigate falls and improve overall quality of life, particularly in vulnerable populations.

To this end, the synergy between HOLOBALANCE and SMART BEAR presents an opportunity for mutual enrichment and expanded value. By integrating the HOLOBALANCE solution as a pilot activity within SMART BEAR, researchers can leverage its novel approach to virtual coaching through holographic physiotherapist surrogates. This integration would facilitate the delivery of personalized interventions aimed at promoting healthy and independent living, aligning with the overarching goals of both projects. Moreover, by promoting evidence-based interventions across a broader spectrum of patients with varying comorbidities, the combined efforts of these platforms can address a wider range of healthcare needs. Through rigorous assessment comparing interventions facilitated by HOLOBALANCE with standard care, alongside the utilization of IoT and advanced sensing technologies, researchers can comprehensively evaluate the efficacy of these interventions and continuously improve the platform. This collaborative approach not only enhances the value proposition of SMART BEAR but also opens up new avenues for exploration and further exploitation, ultimately advancing the field of balance assessment and rehabilitation while promoting holistic well-being in aging populations and individuals with balance disorders.

5.2.2 The Holobalance approach in SMART BEAR

Figure 43: The HOLOBALANCE prediction workflow diagram.

In the context of the SMART BEAR project the method used is the workflow graphically illustrated in the Figure 43. To provide a more detailed explanation, the workflow commences by capturing

unprocessed time series data, which records consecutive observations over a period of time. Subsequently, this data is converted into a tabular format, thereby organizing time-dependent information into a structured arrangement of rows and columns. Next, the dataset is subjected to preprocessing procedures to address any missing values and outliers, so guaranteeing the integrity and dependability of the data. Feature engineering approaches follow which are utilized to enhance the dataset by generating novel informative features based on the current ones. Afterwards, feature selection approaches are utilized to identify the most pertinent features, hence lowering the number of dimensions and improving the effectiveness of the model. Finally, the preprocessed and refined dataset is employed to train machine learning models, enhancing their performance by adjusting parameters and evaluating metrics. This allows for precise predictions or insights to be derived from the data.

Bioinformatics workflows are ubiquitous across various studies in healthcare, medicine, and biology. Wratten *et al.* [5] delve into the high-throughput technologies have reshaped biomedical research, necessitating scalability and reproducibility in computational analysis. It introduces workflow managers as solutions to streamline pipeline development, optimize resource usage, and enhance reproducibility across different computing platforms. Additionally, it outlines their key features, compares common approaches for bioinformatics workflows, and provides guidance for users, showcasing their role in making computational analysis in biomedical research more shareable, scalable, and reproducible.

Wang *et al.* [6] introduced a bioinformatics technique known as Microbial Graph Attention (MEGA), designed to uncover the microorganisms most strongly associated with 12 distinct types of cancer. The effectiveness of MEGA was demonstrated using a dataset obtained from a collaboration of nine cancer institutes within the Oncology Research Information Exchange Network. The workflow of MEGA comprises four primary stages, including model training and the inference of biological gene networks. ORIEN datasets and two database dependencies served as the data sources. Preprocessing steps were applied to prepare the data for training a graph neural network for artificial intelligence. Following deep learning model training, cancer-associated microbial signatures were selected based on attention scores computed for each species at the sample level. The final results, showcasing the identified cancer-associated microbial communities, were presented in a tabular format for further visualization and analysis.

Chen *et al.* [7] provide a comprehensive survey of the latest developments in bioinformatics techniques used to proteomics, with a particular focus on their contribution to the understanding of biological systems. The paper commences by presenting specialized software and tools specifically designed for the purpose of protein identification and quantification using mass spectrometry. Afterwards, it explores different statistical and machine learning methods that have been developed to do thorough analyses in proteomics research. The authors emphasize the importance of quantitative protein data in the reconstruction of protein interactions and signaling networks. The stated workflow encompasses several steps in bioinformatics analysis for mass spectrometry-based proteomics. These steps include data cleaning, normalization, and visualization, followed by enrichment analysis, machine learning, network inference, and data transformation. This workflow highlights the iterative and integrative nature of the analysis process.

Iqbal *et al.* [8] analyze the integration of data science and soft computing techniques in several areas of bioinformatics, emphasizing its practical uses and future prospects. Their exploration encompasses basic techniques such as fuzzy logic, artificial neural networks, and support vector machines, as well as more advanced methods like artificial bee colony and ant colony optimization, which involve optimization and dimension reduction. The report highlights key study fields in bioinformatics, as well

as current trends and difficulties, providing a foundation for future research orientations. The proposed framework integrates the processing of omics data, dimension reduction, and machine learning modeling. The original omics data is processed to identify biomarkers, and then undergoes dimension reduction and machine learning-based prediction stages. These stages involve mapping reads to the human genome, counting reads, identifying genes with differential expression, and classifying diseases using machine learning or soft computing approaches. This complete method provides a deep understanding of the complex relationship between the processing of biological data and the creation of predictive models in bioinformatics research.

In conclusion, the current literature highlights the significant overlap between the methodologies employed in bioinformatics workflows and the approach outlined in the SMART BEAR project. Across various studies, a common thread emerges: the systematic processing of data, feature engineering, and application of machine learning techniques to extract meaningful insights. The HOLOBALANCE workflow described in SMART BEAR, from data preprocessing to model training and evaluation, aligns closely with the methodologies highlighted in bioinformatics research. This convergence is promising as it indicates a shared understanding of the importance of robust data analysis pipelines in extracting actionable insights from complex datasets. Furthermore, the emphasis on reproducibility, scalability, and integration of advanced computational techniques echoes the principles driving the SMART BEAR project. As such, the alignment between the methodologies employed in the current literature and the SMART BEAR pipeline bodes well for the project's success and its potential to contribute significantly to the field of healthcare analytics. By leveraging and building upon the existing knowledge and methodologies in bioinformatics, SMART BEAR stands poised to make substantial advancements in understanding and addressing healthcare challenges, ultimately leading to improved patient outcomes and quality of care.

In the SMART BEAR approach, HOLOBALANCE methods and data are further specifically explored and exploited to develop useful analytic and predictive tools for both balance and cognitive exercises. In this context, the data collected from patients participating to HOLOBalance and Smart4Health[18] training, the latter envisaging MedX machine usage in the Madeira pilot, are analyzed, and used for developing Machine Learning (ML) predictive models. The ML approaches are discriminated by balance and cognitive exercises outcomes, therefore the model outcomes focus on 1) predicting the final total score [9], [10] of participants performing balance exercises by using only early sessions data, and 2) predicting the score of cognitive exercise in the next session.

5.2.2.1 Example of input data

In the SMART BEAR project, the dataset employed for predictive analysis encompasses sensor data obtained from various sources, including head-placed inertial measurement units (IMUs), back-placed IMUs, insoles, and camera systems. These sensors are utilized to monitor and record movements during the execution of balance and cognitive exercises. This data is systematically collected and utilized for analytical purposes as it is showcased in Table 7. Furthermore, data from questionnaires at the beginning and the end of the rehabilitation program, as well as before and after each session are also gathered and exploited.

[18] https://smart4health.eu/

Table 7: Exercise data acquisition sensors.

Sensor	Components	Description	Exercise
Head IMU	Gyroscope Accelerometer Magnetometer	Track head movements during exercise	sitting 1,2,3
			standing 3
			walking 2
			walking3
Back IMU	Goniometer Gyroscope Accelerometer	Track movements during exercise placed at the back	standing 1,2
			walking 1,2,3
Insoles		Track movements during exercise placed at the soles (feet)	standing 1,2
			walking 1,2,3
Camera		Track back posture of patient	All

In relation to the balance exercises, a total of 19 patients undertook these activities. Figure 44 presents the temporal distribution of sessions for each patient, revealing that sessions were not consistently conducted on a weekly basis. Moreover, as it is illustrated in Figure 45, the exercise 'holobalance_standing_2' was primarily performed across all patients, followed by 'holobalance_sitting_1' and 'holobalance_sitting_2'. On average, the cohort completed 8 sessions, with a standard deviation of 1.8.

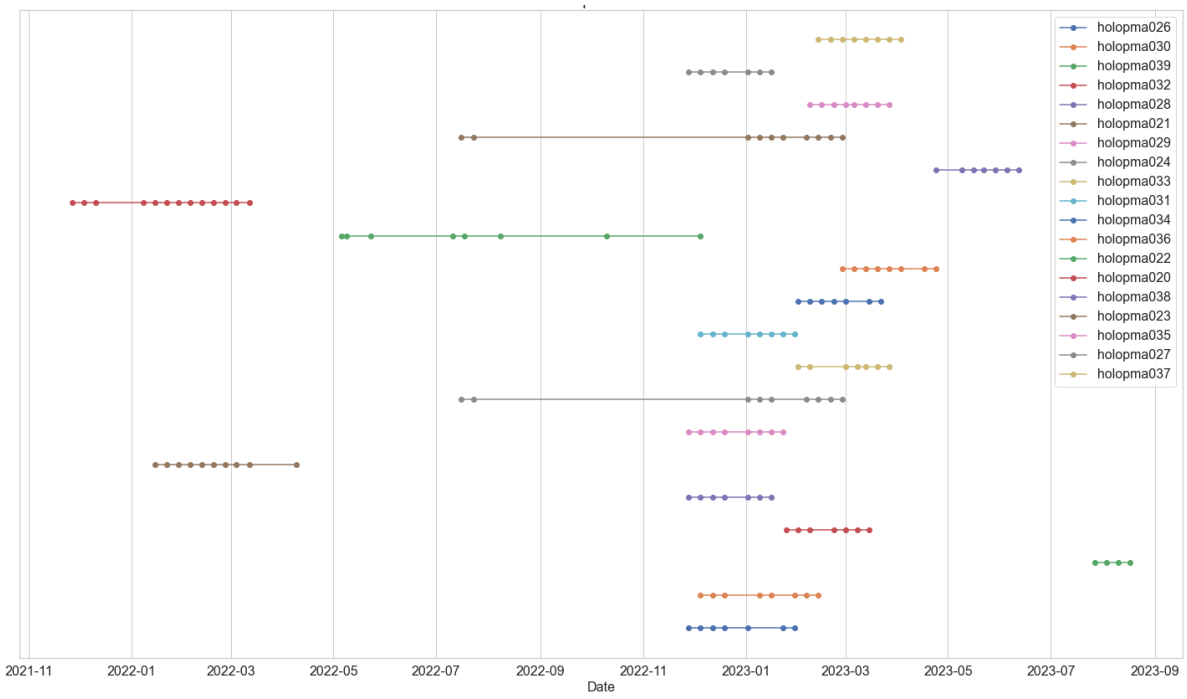

Figure 44: Patients' sessions distribution.

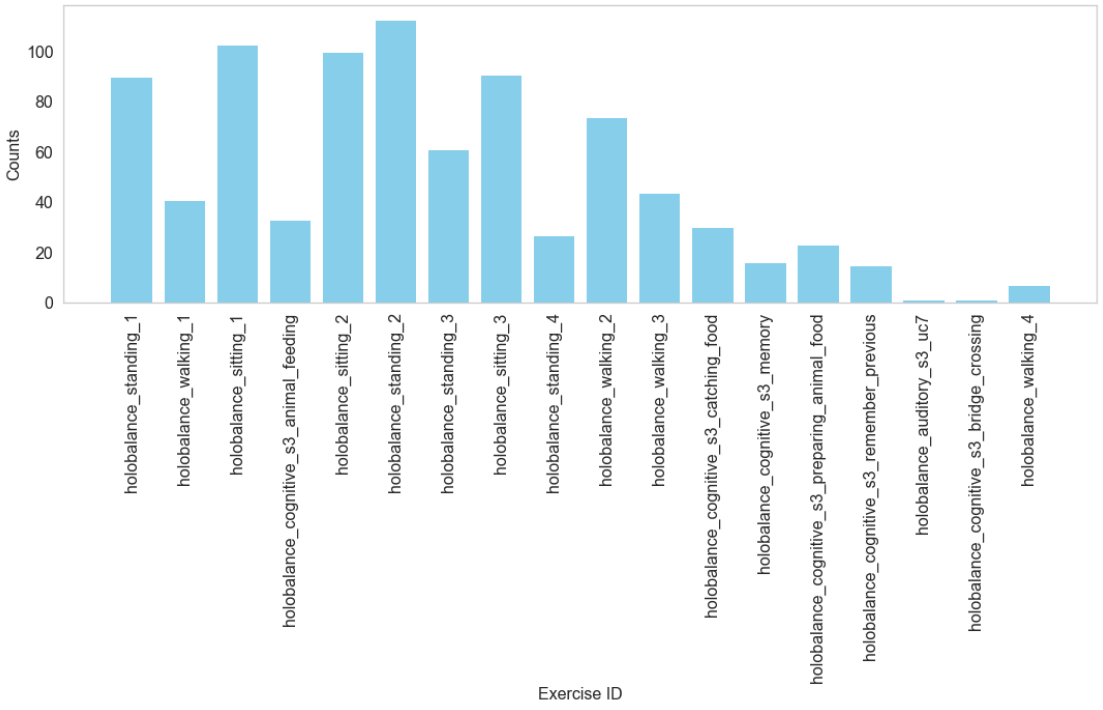

Figure 45: Total conducted exercises distribution across all patients.

Out of the total of 19 patients, responses to the MedX questionnaires were deemed satisfactory for only 16 participants. To incorporate the questionnaire data into the training dataset for the balance score prediction approach, solely the data from these 16 participants were employed. Figure 46 displays the distribution of completion dates for the questionnaires completed by each participant, indicating regular execution of these pain assessment surveys.

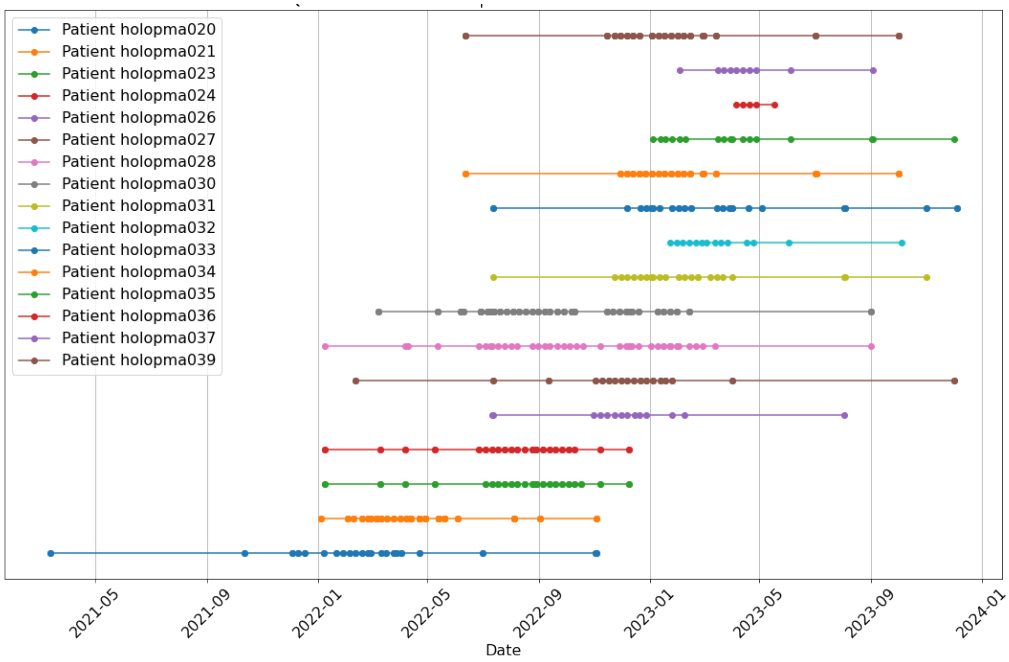

Figure 46: Questionnaires completion dates distribution.

5.2.2.1.1 Missing values and outliers

During the exploration of the dataset, both missing values and outliers were identified. Concerning missing values, features exhibiting more than 30% missing values across all patients and exercises were excluded from further analysis. For the remaining numerical variables, missing values were either imputed by employing the last valid value for the same patient and exercise or by utilizing the average value of the patient cohort if the variable was completely missing for a patient, but the exercise was performed. To maintain the sequence and frequency of exercises performed, if an exercise was not executed during a session, missing values were substituted with 0. For categorical variables, the most frequent value per patient was imputed.

In instances of outliers, these were addressed by replacing them with the average value of the corresponding variable within the respective exercise.

5.2.2.1.2 Updating the Feature Space

In pursuit of deeper insights into the sensor and questionnaire data collected during the rehabilitation sessions, the feature space was expanded through the extraction of additional analytics and descriptive statistics from the variables. An example of the comprehensive dataset for a patient is aggregated in Table 8.

Concerning the HOLOBALANCE solution, the variables extracted for each exercise are described extensively below. In total, there are 11 balance exercises and 7 cognitive games exercises.

Table 8. Data description per exercise.

Exercise	Description (Available in Holobalance dashboard)	API label
Balance exercises		
Exercise Sitting 1 (Yaw head movement)	Start time of exercise - we have the issued field which corresponds to time of post. E.g. "issued": "2021-06-01T09:08:35+02:00", The same applies to all start time.	"start_time"
	Exercise duration - We have it for everything	"duration"
	Total repetitions	"total_reps"
	Posture assessment – Back orientation	"posture"
	Range in yaw movement (Left and right head rotations. Allowed ranges 0-60 degrees	"range"
	Yaw speed of head turns/second (Speed of head rotations. Measured in head turns per second. Allowed ranges 0.5-3.0 head turns per second)	"speed"
	Exercise offline scoring	"score"
	Symptom-Disorientation	"disoriented"
	Symptom-Blurry vision	"blurryvision"
	Symptom-Dizziness	"dizzy"
	Symptom-Headache	"headache"
Exercise Sitting 2 (Pitch head movement)	Start time of exercise – As above (post time)	"start_time"
	Exercise duration	"duration"
	Total repetitions	"total_reps"
	Posture assessment – Back orientation	"posture"
	Range in pitch movement (Up and down head rotations. Allowed ranges 0-90 degrees)	"range"
	Pitch speed of head turns/second (Speed of head rotations. Measured in head turns per second. Allowed ranges 0.5-3.0 head turns per second)	"speed"
	Exercise offline scoring	"score"
	Symptom-Disorientation	"disoriented"
	Symptom-Blurry vision	"blurryvision"
	Symptom-Dizziness	"dizzy"
	Symptom-Headache	"headache"

Exercise	Description (Available in Holobalance dashboard)	API label
Exercise Sitting 3 (Bend)	Start time of exercise – As above (post time)	"start_time"
	Exercise duration - Reps is the duration we have it ok	"duration"
	Total repetitions	"total_reps"
	Posture assessment – Back orientation	"posture"
	Range in pitch movement: [Chest to chin, Chin to chest] (Head goes from chest to chin and chin to chest in pitch plane)	"range"
	Movement timing: [Chest to chin, Chin to chest] (Time required to move head from chest to chin and chin to chest in seconds)	"timing"
	Movement timing – Bending (Total time required to execute the bend over movement)	"timingbend"
	Exercise offline scoring	"score"
	Symptom-Disorientation	"disoriented"
	Symptom-Blurry vision	"blurryvision"
	Symptom-Dizziness	"dizzy"
	Symptom-Headache	"headache"
Exercise Standing 1 (Maintain balance)	Start time of exercise – As above (post time)	"start_time"
	Exercise duration	"duration"
	Total repetitions	"total_reps"
	Posture assessment – Back orientation	"posture"
	Pitch versus roll (Pitch versus roll plane points that indicate the sway in degrees)	"pitchvsroll"
	Cumulative percentage of pitch roll points (Percentages in 10 steps. The closer to center of pressure the patient is the closer to 100 percent most of the percentages are)	"rateofpitchvsroll"
	Exercise offline scoring	"score"
	Symptom-Disorientation	"disoriented"
	Symptom-Blurry vision	"blurryvision"
	Symptom-Dizziness	"dizzy"
	Symptom-Headache	"headache"
	Start time of exercise – As above (post time)	"start_time"

Exercise	Description (Available in Holobalance dashboard)	API label
Exercise Standing 2 (Maintain balance on foam)	Exercise duration	"duration"
	Total repetitions	"total_reps"
	Posture assessment – Back orientation	"posture"
	Pitch versus roll (Pitch versus roll plane points that indicate the sway in degrees)	"pitchvsroll"
	Cumulative percentage of pitch roll points (Percentages in 10 steps. The closer to center of pressure the patient is the closer to 100 percent most of the percentages are)	"rateofpitchvsroll"
	Exercise offline scoring	"score"
	Symptom-Disorientation	"disoriented"
	Symptom-Blurry vision	"blurryvision"
	Symptom-Dizziness	"dizzy"
	Symptom-Headache	"headache"
Exercise Standing 3 (Bend and reach up)	Start time of exercise – As above (post time)	"start_time"
	Exercise duration	"duration"
	Total repetitions	"total_reps"
	Posture assessment – Back orientation	"posture"
	Range in pitch movement: [Chin to chest, Chest to chin] (Head goes from chest to chin and chin to chest in pitch plane)	"range"
	Movement timing: [Chin to chest, Chest to chin] (Time required to move head from chest to chin and chin to chest in seconds)	"timing"
	Movement timing – Bending (Total time required to execute the bend over movement)	"timingbend"
	Bend down flexion (Total degrees of bending flexion of the body)	"benddownflexion"
	Range in pitch movement (The range in pitch plane while looking up executing the exercise. Allowed range from 0 - 90 degrees)	"range"
	Speed of head turn/sec (The speed of head rotations in head turns per second)	"speed"

Exercise	Description (Available in Holobalance dashboard)	API label
	Speed in pitch movement (The speed of head rotations in head turns per second)	"pitchspeed"
	Movement and standing timing: [Rise time, Stand time] (Time in seconds of executing the movement, rise time, and time in seconds standing still between movements)	"timing"
	Pitch range (The range in pitch plane while looking up executing the exercise. Allowed range from 0 - 90 degrees)	"rangepitch"
	Time of pitch head movement (Time required to execute a pitch movement looking up and going to the starting head position)	"thetimeofpitchmovement"
	Exercise offline scoring	"score"
	Symptom-Disorientation	"disoriented"
	Symptom-Blurry vision	"blurryvision"
	Symptom-Dizziness	"dizzy"
	Symptom-Headache	"headache"
Exercise Standing 4 (Turn 180°)	Start time of exercise – As above (post time)	"start_time"
	Exercise duration	"duration"
	Total repetitions	"total_reps"
	Posture assessment – Back orientation	"posture"
	Range of movement (Turn movement expressed in degrees. Optimal is 180 degrees)	"turn_degree"
	Time of movement (Time to turn 180 degrees)	"time_turn"
	Number of steps (Number of steps to turn 180 degrees)	"steps_turn"
	Exercise offline scoring	"score"
	Symptom-Disorientation	"disoriented"
	Symptom-Blurry vision	"blurryvision"
	Symptom-Dizziness	"dizzy"
	Symptom-Headache	"headache"
Exercise Walking 1 (Walk to horizon)	Start time of exercise – As above (post time)	"start_time"
	Exercise duration - yes	"duration"
	Total repetitions (walk routes)	"total_reps"

Exercise	Description (Available in Holobalance dashboard)	API label
	Posture assessment – Back orientation	"posture"
	Percentage single support Left, percentage single support Right, percentage double support (percentages of single support double support)	"metrixthree"
	Number of steps	"numofstep"
	Sway index rate	"metrixsway"
	Exercise offline scoring	"score"
	Symptom-Disorientation	"disoriented"
	Symptom-Blurry vision	"blurryvision"
	Symptom-Dizziness	"dizzy"
	Symptom-Headache	"headache"
Exercise Walking 2 (Walk and yaw head movement)	Start time of exercise – As above (post time)	"start_time"
	Exercise duration	"duration"
	Total repetitions (walk routes)	"total_reps"
	Posture assessment – Back orientation	"posture"
	Percentage single support Left, percentage single support Right, percentage double support	"metrixthree"
	Number of steps	"numofstep"
	Sway index rate	"metrixsway"
	Total repetitions (head turns)	"total_reps"
	Range in yaw movement	"range"
	Yaw speed of head turns/second	"speed"
	Exercise offline scoring	"score"
	Symptom-Disorientation	"disoriented"
	Symptom-Blurry vision	"blurryvision"
	Symptom-Dizziness	"dizzy"
	Symptom-Headache	"headache"
Exercise Walking 3 (Walk and	Start time of exercise – As above (post time)	"start_time"
	Exercise duration	"duration"

Exercise	Description (Available in Holobalance dashboard)	API label
pitch/V-shaped head movement)	Total repetitions (walk routes)	"total_reps"
	Posture assessment – Back orientation	"posture"
	Percentage single support Left, percentage single support Right, percentage double support * Reported at exergame only	"metrixthree"
	Sway index	"metrixsway"
	Number of steps	"numofstep"
	Total repetitions (head turns only)	"total_reps"
	Range in pitch movement	"range"
	Pitch speed of head turns/second	"speed"
	Exercise offline scoring	"score"
	Symptom-Disorientation	"disoriented"
	Symptom-Blurry vision	"blurryvision"
	Symptom-Dizziness	"dizzy"
	Symptom-Headache	"headache"
Cognitive exercises		
Cognitive game – Memory	Start time of exercise – As above (post time)	"start_time"
	Exercise duration (sec)	"exercise_duration"
	Progression level	"level"
	Total cards	"total_cards"
	Total score	"total_score"
	Time needed for answer	"time_needed"
	Questions	"question"
	Answers	"answer"
Cognitive game – Catching food	Start time of exercise – As above (post time)	"start_time"
	Exercise duration (sec)	"exercise_duration"
	Total rounds	"total_rounds"

Exercise	Description (Available in Holobalance dashboard)	API label
	Total rounds correct	"total_rounds_correct"
	Total score	"total_score"
	Average reaction time	"avg_reaction_time"
	Progression level	"level"
	Distance previous apple: [Right answer, Wrong answer]	"reaction_time"
Cognitive game – Remember previous	Start time of exercise – As above (post time)	"start_time"
	Exercise duration (sec)	"exercise_duration"
	Total rounds	"total_rounds"
	Total rounds correct	"total_rounds_correct"
	Total score	"total_score"
	Average reaction time	"avg_reaction_time"
	Progression level	"level"
	Reaction time: [Right answer, Wrong answer]	"reaction_time"
Cognitive game – Bridge crossing	Start time of exercise – As above (post time)	"start_time"
	Exercise duration	"exercise_duration"
	Total rounds	"total_rounds"
	Total rounds correct	"total_rounds_correct"
	Total score	"total_score"
	Progression level	"level"
	Time spawned - right answer	-
	Time spawned - wrong answer	-
Cognitive game – Animal feeding	Start time of exercise – As above (post time)	"start_time"
	Exercise duration (sec)	"exercise_duration"
	Total rounds	"total_rounds"
	Total rounds correct	"total_rounds_correct"
	Total score	"total_score"
	Average reaction time	"avg_reaction_time"

Exercise	Description (Available in Holobalance dashboard)	API label
	Progression level	"level"
	Reaction time: [Right answer, Wrong answer]	"reaction_time"
Cognitive game – Preparing animal food	Start time of exercise – As above (post time)	"start_time"
	Exercise duration (sec)	"exercise_duration"
	Total rounds	"total_rounds"
	Total rounds correct	"total_rounds_correct"
	Total score	"total_score"
	Average reaction time	"avg_reaction_time"
	Progression level	"level"
	Reaction time: [Right answer, Wrong answer]	"reaction_time"
MedX Questionnaires		
Questionnaires (pre & post training, training, presession, forcetest)	Date of questionnaire completion	"date"
	ID of HOLOBALANCE participant	"holo_id"
	Average pain level during the past four weeks	"average_back_pain_level"
	Current pain level	"current_back_pain_level"
	Maximum pain level during the past four weeks	"max_back_pain_level_4w"
	Taking painkillers at current assessment	"taking_painkillers"
	Recommended weight	"recommended_weight "
	Angle force set	"angle_force_set "
	Lower back pain level before training	"back_pain_level_before"
	Training number	"training_no"
	Well-being (0: All well, 10: All bad)	"well_being"
	Summary of health, general well-being, and activities in the last days/weeks	"well-being_summary"

Exercise	Description (Available in Holobalance dashboard)	API label
	Training weight recommendation for the next training	"training_weight_recommendation"
	The accuracy score of the exercise in percentage	"score"
	The number of repetitions executed during the exercise	"repetitions"
	The raw angle measurements during exercise	"sensor_raw"

The additional features extracted from the original dataset encompass a comprehensive range of parameters related to both balance and cognitive exercises. These include various metrics such as the number of symptoms reported, total range and offset movements, speed and rate of pitch versus roll, as well as detailed statistics like minimum, maximum, average, and interquartile ranges for different movement characteristics. For cognitive exercises, parameters such as reaction time statistics for correct and incorrect answers, as well as overall reaction time metrics, were extracted. Moreover, relevant features from the MedX questionnaire data, such as average and standard deviation of angle force set, were incorporated for analysis. These extracted features, aggregated in Table 9, provide a richer dataset for deeper analysis and insights into rehabilitation sessions.

Table 9: Extracted features from original dataset.

Original Features	Features extracted
Balance exercises	
(32) Patient ID, Date, Exercise ID, Start time, Exercise duration, Total repetitions, Score, Disoriented, Dizzy, Headache, Blurry vision, Speed, Range, Posture, Offset, Timing, Rate of pitch vs roll, Timing bend, Time of pitch movement, Bend down flexion, Pitch speed, Range of pitch, Turn degree, Time of turn, Steps of turn, Metrix three, Number of steps, Metrix way, Offset routes, Offset head, Total repetitions of head	(53) Number of symptoms, Number of speed movements, Total range movements, Total offset movements, Total/Average/Sum/IQR Q1/Q2/Q3 offset, Total/Average/Sum/ IQR Q1/Q2/Q3 range, Minimum/Maximum/Average/IQR Q1/Q2/Q3 speed, Minimum/Maximum/Average/ IQR Q1/Q2/Q3 rate pitch vs roll, Number of rate vs pitch roll steps, Total/Average/Sum/ IQR Q1/Q2/Q3 of bend down flexion movements, Minimum/Maximum/Average/ IQR Q1/Q2/Q3 time of turn, Number of time turns, Total offset head movements, Total offset routes movements, Sum/Average/ IQR Q1/Q2/Q3 offset head, Sum/Average/ IQR Q1/Q2/Q3 offset routes,
Cognitive exercises	
(17) Patient ID, Date, Exercise ID, Start time, Exercise duration, Level, Reaction time, Average reaction time, Total rounds of correct answers, Total rounds, Total score, Question, Answer, Time needed, Total cards, Total marbles, Total marbles correct	(12) Minimum/Maximum/Standard deviation/IQR Q1/Q2/Q3 of overall reaction time, Minimum/Maximum/Standard deviation reaction time for correct answers, Minimum/Maximum/Standard deviation reaction time for wrong answers,

MedX questionnaire (kept for analysis)	
(16) Patient ID, Date, Average back pain level, Current back pain level, Maximum back pain level 4w, Taking painkillers, Recommended weight, Angle force set, Back pain level before, Training session number, Well being coded and summary, Training weight recommendation, Score, Repetitions, Sensor raw	(5) Average/Standard deviation of angle force set, Minimum/Maximum/Average of raw sensor angle measurements,

5.2.2.2 Feature Selection

Since feature selection is considered important for enhancing model performance, reducing overfitting, and improving interpretability. For this pipeline, the Recursive Feature Elimination (RFE) method, coupled with the powerful XGBoost regressor, offers a robust approach to selecting the most informative features. RFE iteratively trains the model, starting with all features, and subsequently removes the least significant ones until the optimal subset is achieved. By leveraging the predictive power of XGBoost, which excels in capturing complex relationships within the data, RFE efficiently identifies the subset of features that contribute most to the model's predictive performance. This method was applied to each of the sub dataframes created in the balance exercises dataset (Sections 5.2.2.3.1 and 5.2.2.3.2).

Given the significance of feature selection in enhancing model performance, mitigating overfitting, and enhancing interpretability, the recursive feature elimination (RFE) method, in conjunction with the XGBoost regressor, presents a robust strategy for identifying the most informative features. RFE operates through iterative model training, initially incorporating all features and subsequently eliminating the least significant ones until an optimal subset is attained. Leveraging the predictive capabilities of XGBoost, renowned for its proficiency in capturing intricate data relationships, RFE efficiently discerns the subset of features contributing most significantly to the model's predictive accuracy. This methodology was applied to both sub dataframes within the balance exercises dataset, each pertaining to a distinct score prediction model, as described in the subsequent sections 12.5.1 and 12.5.2.

5.2.2.3 Model Training

Since both approaches are regression tasks, we consider XGBoost regressor, AdaBoost regressor, Random Forest regressor, and ElasticNet model. XGBoost Regressor stands out for its superior performance owning to its ensemble of decision trees, employing gradient boosting techniques that make it robust against overfitting and capable of handling large datasets efficiently. AdaBoost Regressor, another ensemble method, is beneficial when seeking a balance between simplicity and predictive power. It sequentially adjusts the weights of misclassified observations, thereby focusing more on challenging instances, making it suitable for tasks where interpretability and generalization are important. Elastic Net, a linear regression technique, combines the merits of Lasso and Ridge regression, effectively addressing multicollinearity and feature selection. It is particularly useful when dealing with high-dimensional data where feature redundancy is prevalent. On the other hand, Random Forest, a versatile ensemble of decision trees, excels in capturing complex relationships and interactions within the data while providing robustness against outliers and noise. Its ability to handle both numerical and categorical data, along with its built-in feature importance metrics, makes it a

suitable choice for a wide range of regression tasks, especially when interpretability is less critical compared to predictive accuracy.

In regression tasks, where the objective is to predict continuous numerical values, the performance of models is typically assessed using various metrics. Among these, Root Mean Squared Error (RMSE) measures the square root of the average of the squared differences between predicted and actual values. It's particularly sensitive to large errors due to the squaring operation. Mean Absolute Error (MAE), on the other hand, computes the average of the absolute differences between predicted and actual values, making it less sensitive to outliers. Mean Absolute Percentage Error (MAPE) calculates the average percentage difference between predicted and actual values, providing a relative measure of error. However, it may encounter issues with small actual values. Lastly, the Coefficient of Determination (R-squared) quantifies the proportion of the variance in the dependent variable that is predictable from the independent variables. R-squared values range from 0 to 1, where higher values indicate a better fit. To evaluate the regression models in both approaches developed in this task, it's crucial to consider multiple metrics to offer a comprehensive assessment of the model's performance.

5.2.2.3.1 Balance Model

In the initial approach aimed at predicting the final score of the balance exercises, utilizing only the initial time points of each session (as it is depicted in Figure 47), four distinct ML model architectures were investigated. These encompassed tree-based, kernel-based, and ensemble methods. An essential aspect of the training pipeline involved determining the optimal number of sessions to consider for training, thereby introducing an additional parameter to be explored. Specifically, four different input datasets were constructed, incorporating data from the first to fourth sessions, first to fifth sessions, first to sixth sessions, and first to seventh sessions, respectively. Notably, the first session from each dataset was excluded based on feedback derived from Madeira pilot representatives, aiming to mitigate potential performance bias, as participants typically utilize the initial session to acclimate to the rehabilitation solutions.

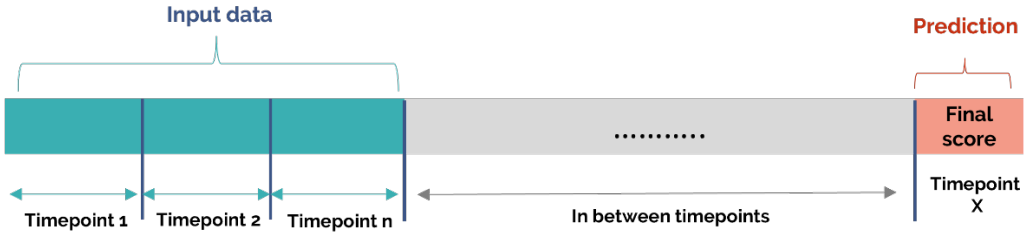

Figure 47: Observation and prediction window.

Figure 48 illustrates the distributions of exercises across all time points, with each of the four datasets annotated accordingly. It is crucial to highlight that, based on this distribution, the minimum number of sessions that can be considered is three, as all exercise data are recorded after the second time point.

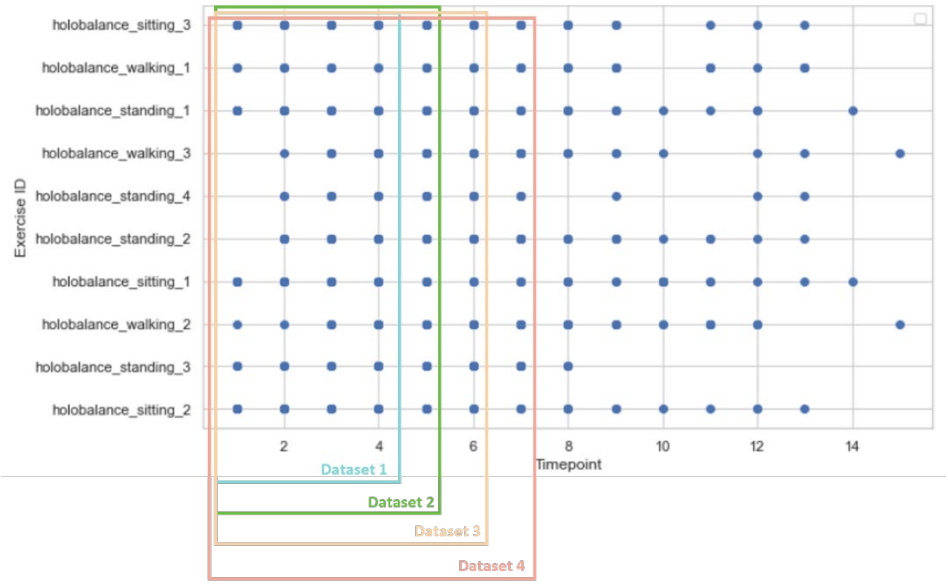

Figure 48: Exercise IDs against timepoints / sessions.

Concerning the output variable, the variable 'final score' was generated to reflect the average score of all exercises performed during the last session for each patient, as it is depicted graphically in Figure 49.

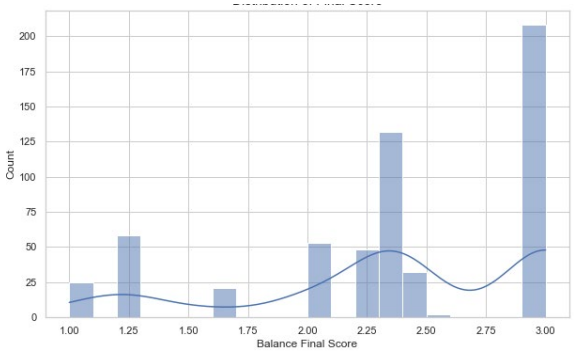

Figure 49: Distribution of final score (outcome variable).

Through a series of experiments conducted across all models utilizing the preprocessed dataset, the optimal number of sessions required for predicting the final score was investigated. Given the dataset's constraints and the evident correlations between features, exacerbated by the utilization of multiple descriptive statistics for each variable, the models' performance is deemed moderate with a noticeable trend towards overfitting. Particularly focusing on the R^2 metric, the results appear poor, with primarily negative values, indicative of challenges in generalizing the model to new, unseen data. As previously noted, these outcomes are unsurprising given the limitations of the dataset. Nonetheless, in most instances, XGBoost and AdaBoost consistently outperform other models as it is highlighted in Figure 50, Figure 51, Figure 52, and Figure 53. Notably, utilizing the first 4 sessions as the past horizon with the XGBoost regressor model yields the most favorable results in these experiments, primarily attributable to its non-negative and superior R^2 metric value.

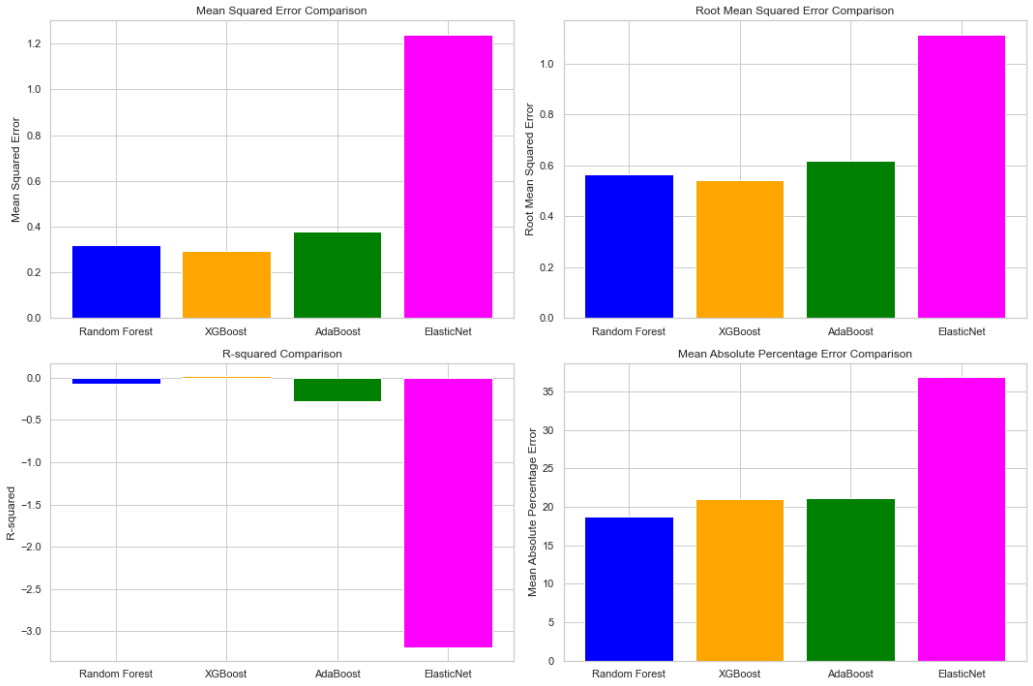

Figure 50: AI models performance with past horizon equal to 3 sessions.

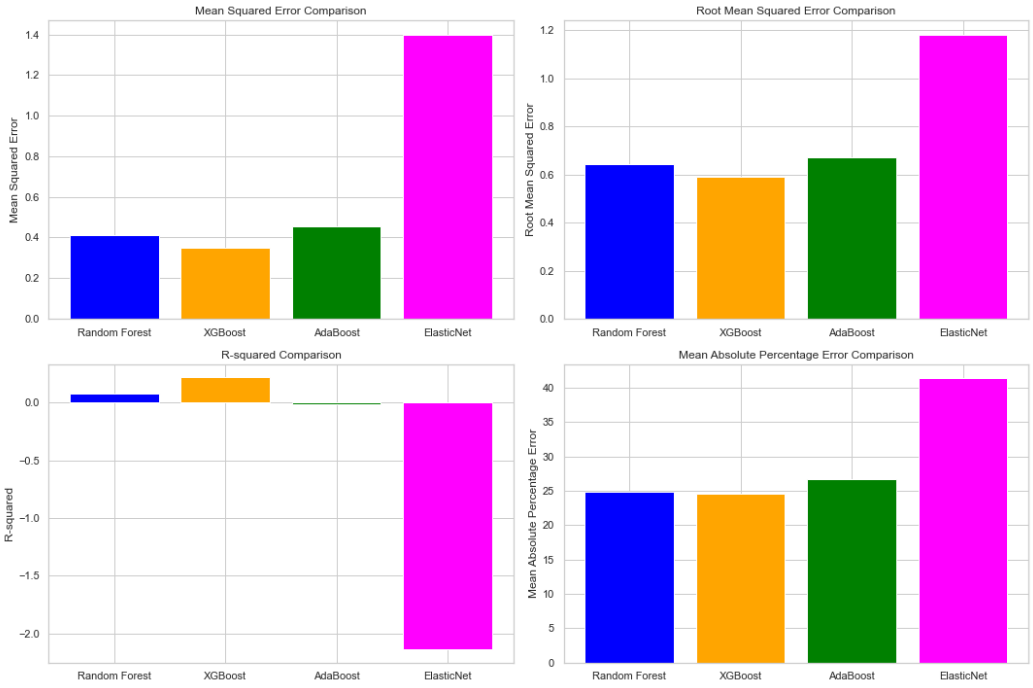

Figure 51: AI models performance with past horizon equal to 4 sessions.

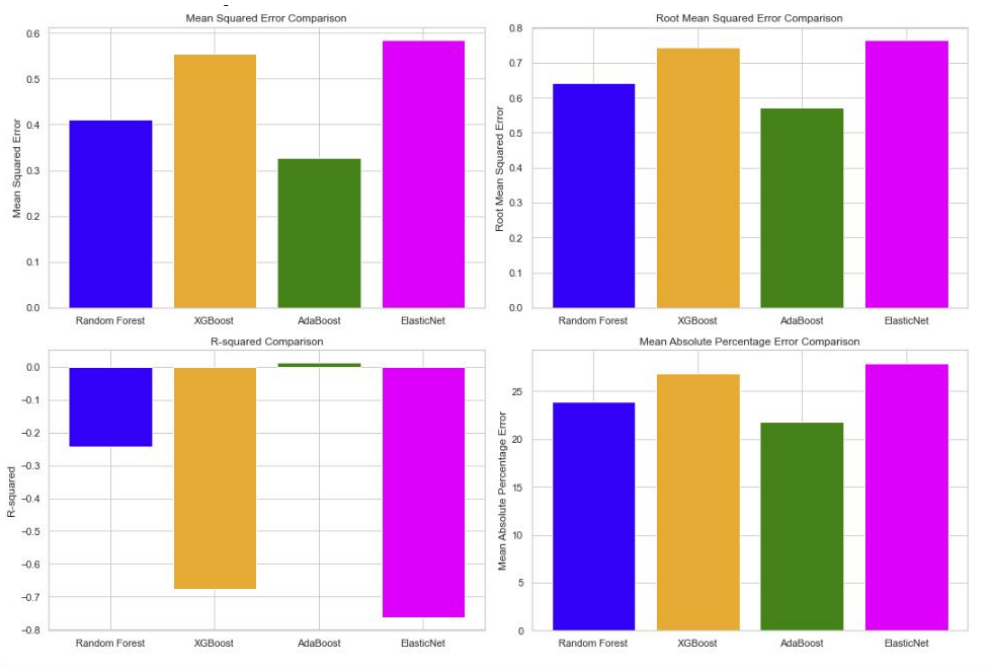

Figure 52: AI models performance with past horizon equal to 5 sessions.

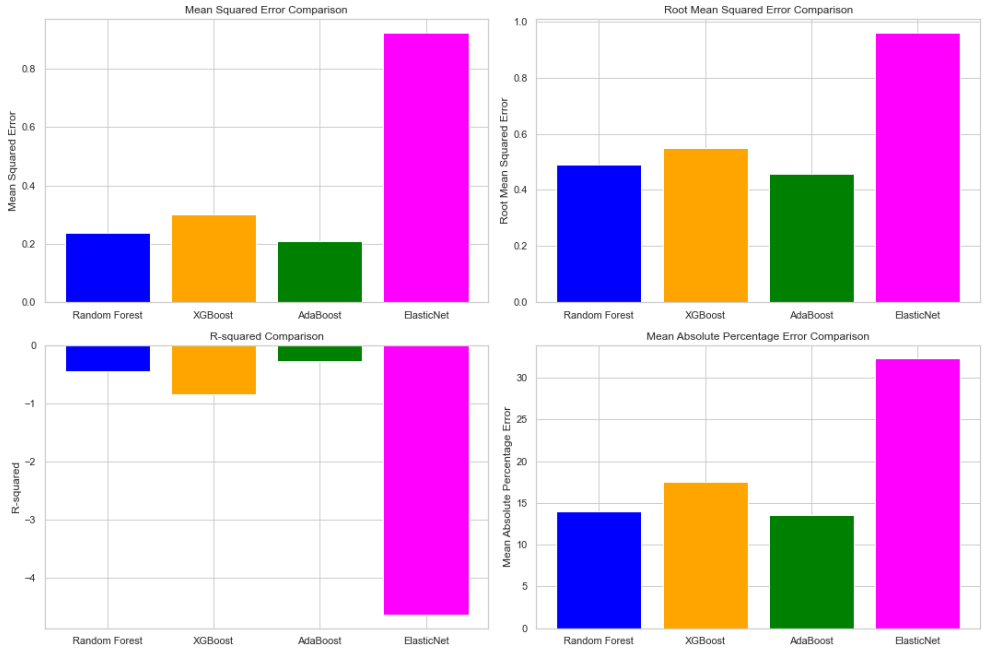

Figure 53: AI models performance with past horizon equal to 6 sessions.

In Table 10, only the best performing model for each sub-dataset is described, accompanied by the corresponding feature importance plots in each case (i.e., Figure 54, Figure 55, Figure 56, Figure 57). Notably, it is evident that both questionnaire and exercise variables play significant roles in the models' performance.

Table 10: Balance exercises models performance.

AI Model	Time points (Past horizon)	Results
XGBoost Regressor	3 (2nd session + 3rd session + 4th session)	RMSE: 0.292 MAE: 0.424 MAPE: 0.21 R^2: 0.014
XGBoost Regressor	4 (2nd session + 3rd session + 4th session + 5th session)	RMSE: 0.351 MAE: 0.435 MAPE: 0.245 R^2: 0.216
AdaBoost Regressor	5 (2nd session + 3rd session + 4th session + 5th session + 6th session)	RMSE: 0.327 MAE: 0.427 MAPE: 0.218 R^2: 0.013
AdaBoost Regressor	6 (2nd session + 3rd session + 4th session + 5th session + 6th session + 7th session)	RMSE: 0.21 MAE: 0.35 MAPE: 0.135 R^2: -0.279

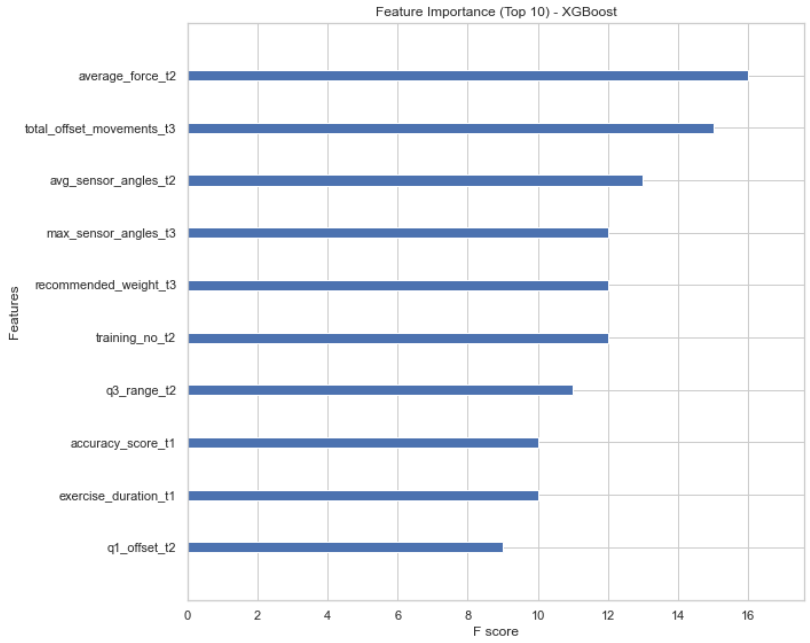

Figure 54: Top 10 most important features with past horizon equal to three sessions.

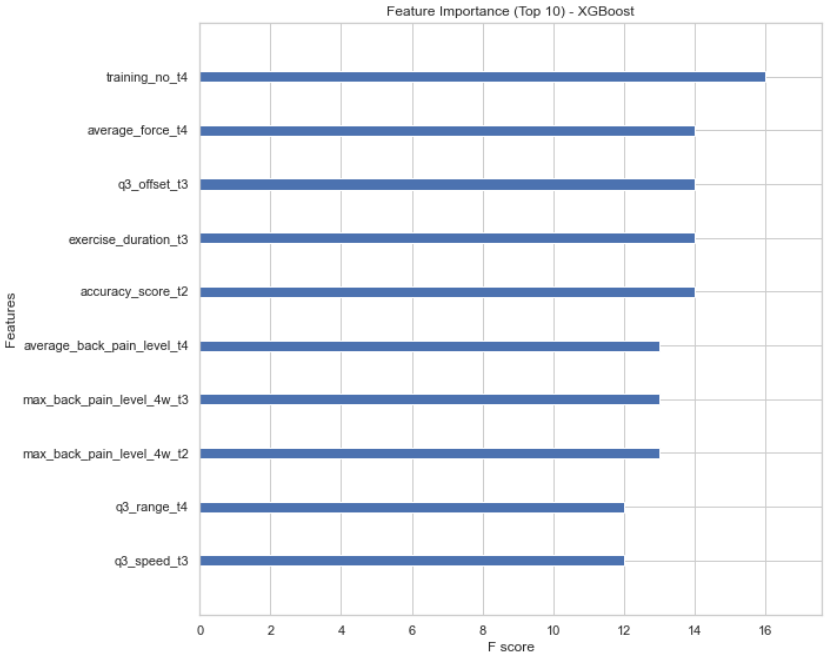

Figure 55: Top 10 most important features with past horizon equal to four sessions.

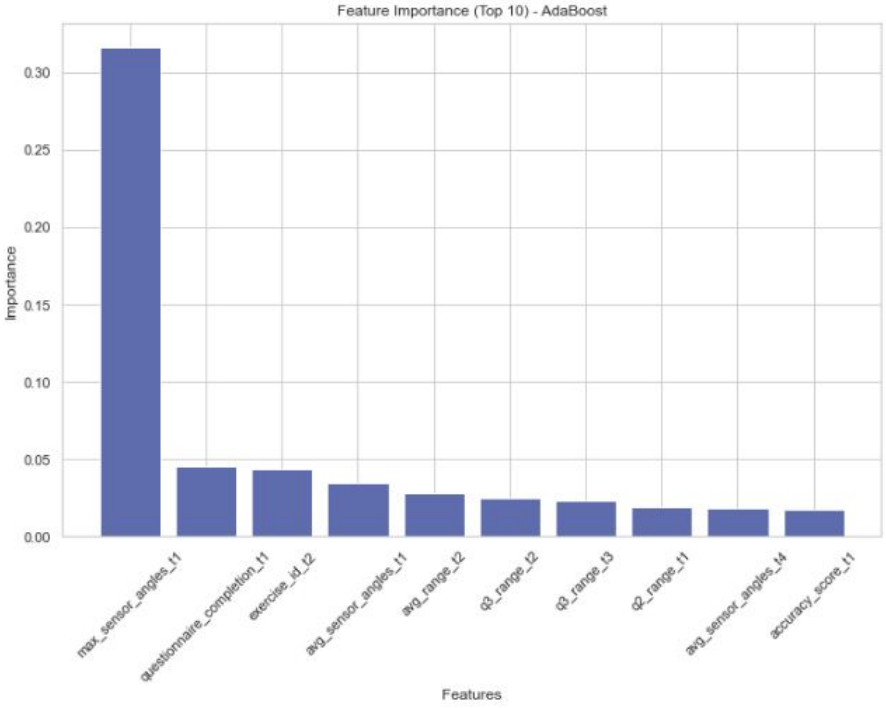

Figure 56: Top 10 most important features with past horizon equal to five sessions.

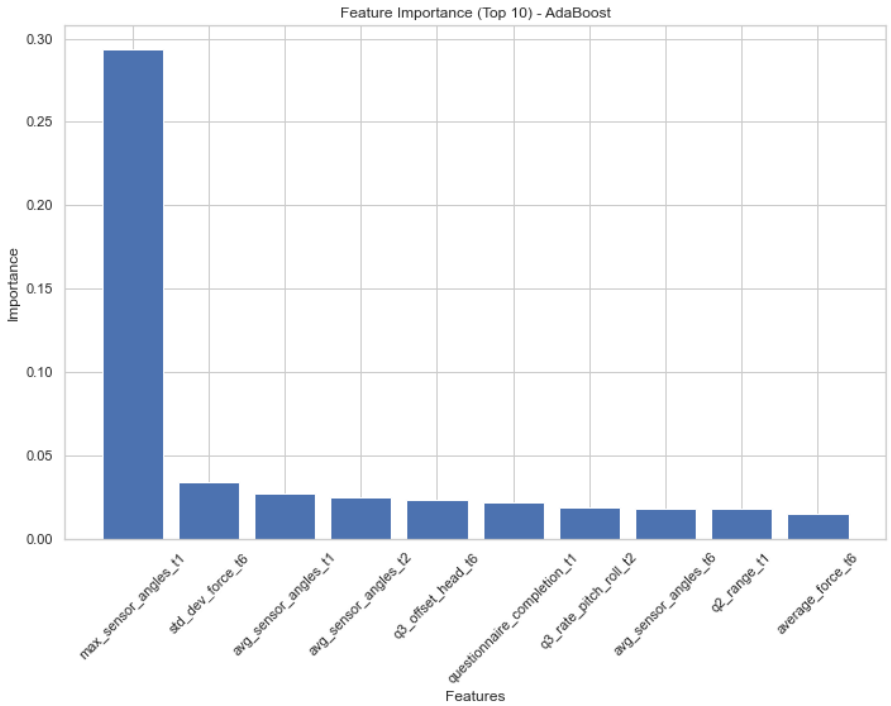

Figure 57: Top 10 most important features with past horizon equal to six sessions.

5.2.2.3.2 Cognitive Model

In the second approach, the focus shifted to predicting the score of cognitive exercises in the next session (timepoint). Three individualized models were developed for each cognitive exercise, as each exercise comprises a distinct set of features. Two exercises, namely 'holobalance_cognitive_s3_memory' and 'holobalance_cognitive_s3_bridge_crossing', were excluded

due to their limited number of features representing the exercises. Additionally, after preprocessing, the exercise 'holobalance_cognitive_s3_remember_previous' was not utilized further for training models, as there were insufficient data points available.

In this scenario, the input variables consisted solely of cognitive-related exercises (refer to Table 8). The prediction of the total score in the upcoming session was reflected by the 'total_score' variable in each of these exercises (Figure 58), shifted by one time point earlier to facilitate the regression task on the subsequent time point.

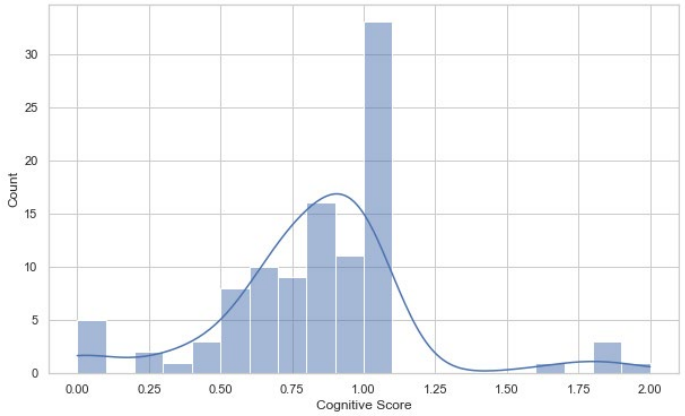

Figure 58: Distribution of total score across cognitive exercises.

Furthermore, the distribution of participants engaging in each exercise is depicted in Figure 59, alongside the total cognitive sessions. Additionally, Figure 60 illustrates the number of sessions involving the execution of cognitive exercises. Notably, there exists variability in performing cognitive exercises, with 'holobalance_cognitive_s3_animal_feeding' emerging as the most frequently performed exercise across all time points (4 in total) and patients.

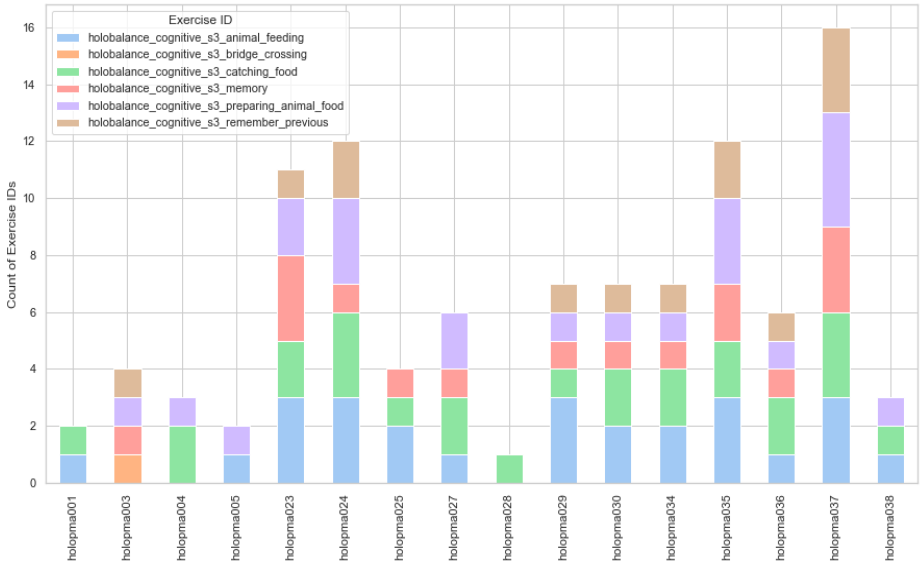

Figure 59. Cognitive exercises performance per patient ID.

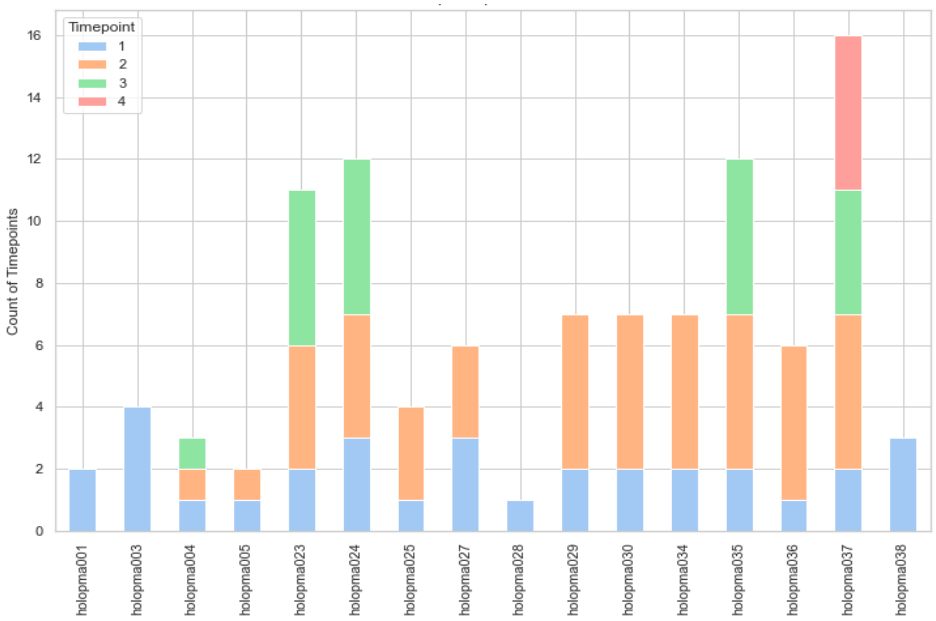

Figure 60. Sessions distribution for performing cognitive exercises.

As a result, only 12 patients have engaged in the cognitive exercises more than once, with just 5 participants completing 3 sessions and only one patient completing 4 sessions. Consequently, since the prediction task relies on the availability of multiple time points in the dataset, patients with only one time point recorded for each exercise were excluded from the model development process.

For each of the five remaining exercises, a dedicated dataset was created containing data solely related to that exercise. Subsequently, an XGBoost machine learning model was employed to undertake the regression task of score prediction. The performance of these models was evaluated using metrics such as RMSE, MAE, MAPE, and R^2. This performance evaluation is aggregated in Table 11.

Table 11: Cognitive exercises models performance.

Cognitive exercise (CE)	Exercise	Total number of participants	Number of features	Model results
CE1	holobalance_cognitive_s3_animal_feeding	7	15	RMSE: 0.046 MAE:0.167 MAPE: 0.185 R^2: -1.32
CE2	holobalance_cognitive_s3_catching_food	9	13	RMSE: 22.47 MAE: 4.67 MAPE: 0.774 R^2: -32.71
CE3	holobalance_cognitive_s3_preparing_animal_food	9	15	RMSE: 2.764 MAE: 1.457 MAPE: 0.513 R^2: -0.77

Based on these results, for each model and exercise, the only exercise with good performance is ‘holobalance_cognitive_s3_animal_feeding’, as the other exercises result in very poor evaluation metrics. For this exercise, the most important features for this prediction task are given. As observed exercise duration is recorded as the most important feature for this prediction.

Based on these results, it is evident that among all the models and exercises, only 'holobalance_cognitive_s3_animal_feeding' exhibits satisfactory performance, with the other exercises yielding very poor evaluation metrics. For this particular exercise, the most important features for the prediction task are illustrated in Figure 61. Notably, the exercise duration emerges as the most influential feature for this prediction, highlighting its significance in determining the cognitive exercise score.

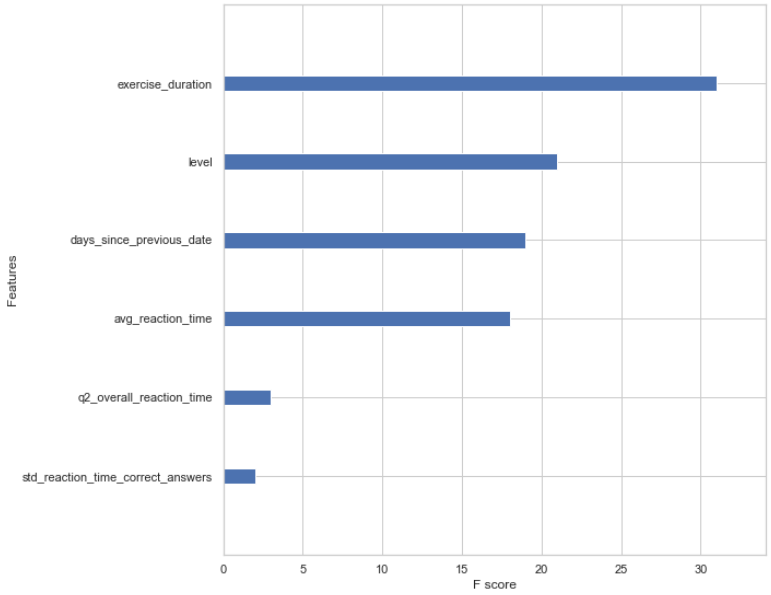

Figure 61: Feature importance for 'holobalance_cognitive_s3_animal_feeding' exercise.

5.2.2.4 Optimization

In both approaches, grid search with 5-fold cross-validation accuracy as the objective function was employed to determine the optimal models' hyperparameters. This comprehensive search was conducted across a wide range of parameters to ultimately identify the most well-performing models for each task. However, after numerous experiments, hyperparameter tuning did not yield improved results for the second approach concerning the cognitive exercises.

The specifics of the hyperparameters searched as well as those chosen for the balance exercises are outlined in Table 12.

Table 12: Hyperparameter tuning details.

AI Model	Prediction task	Hyperparameters space	Final hyperparameters
XGBoost	Final balance score prediction	min child weight: [1, 5, 10], max depth: [3, 7], n_estimators: [200,300,500, 1000], subsample: [0.4,0.8], reg_alpha: [0.1, 0.8], reg_lambda: [0.1,0.5], colsample by tree: [0,0.6, 1.0]	*(Past horizon =3)* min child weight: 5, max depth: 3, n_estimators: 500, subsample: 0.4, reg_alpha: 0.1, reg_lambda: 0.1, colsample by tree: 0.6
			(Past horizon =4) min child weight: 5, max depth: 7, n_estimators: 1000, subsample: 0.8, reg_alpha: 0.1, reg_lambda: 0.1, colsample by tree: 1.0
AdaBoost	Final balance score prediction	learning_rate: [0.01,0.05, 0.001], n_estimators: [200,400,500]	learning_rate:0.001, n_estimators: 400
Random Forest	Final balance score prediction	n_estimators: [100, 200, 300], max_depth: [None, 10, 20],min_samples_split: [2, 5, 10]	-
ElasticNet	Final balance score prediction	alpha: [0.1, 0.5, 1.0], l1_ratio': [0.1, 0.2, 0.9]	-

Further optimizations of the models' performance can be achieved by leveraging data augmentation and synthetic data generation methods, particularly considering the primary limitation encountered, which was the limited amount of data available. By augmenting the dataset with additional synthesized data points, the models can be trained on a more diverse and comprehensive set of examples, potentially improving their generalization capabilities.

Additionally, exploring further preprocessing approaches could enhance the integration of sensor data into the training procedure. This might involve refining feature engineering techniques or implementing advanced signal processing methods to extract more meaningful insights from the sensor data. By enhancing the preprocessing pipeline, the models can be provided with more informative input features, leading to improved predictive performance.

5.2.3 Clinical Relevance

In this study, the population evaluated were elderly people with chronic non-specific low back pain, who followed a rehabilitation protocol that included, in addition to standard care, multimodal balance exercises and cognitive retraining in an augmented reality environment. It is known that there is a correlation between cognitive deficit and chronic pain in the elderly. The mechanism has not been established but the possibility that there is difficulty in diverting attention from pain thus reducing the cognitive reserve [11] for additional brain functions is a hypothesis with strong arguments. Altered brain activity, up-regulation of inflammatory microglia could play a crucial role in the maintenance of chronic pain. These problems are reflected in disorders such as impaired processing and working memory [12].

The same population also shows balance disorders [13] with increased chances of recurrent falls [14] and/or falls-related injury. Altered muscle activation patterns, compromise postural stability and increase the risk of falls in this specific population. Catastrophizing and increased levels of anxiety reflect their impact on abnormal movement patterns and avoidance behaviors that affect gait characteristics and postural control [15].

Physical activity is the clinical recommendation to specialists for the management of chronic low back pain. A multifactorial approach [16] gives promising results and thus the application of balance exercises and cognitive retraining should be included in the treatment protocol.

Improvement of reaction time and processing exercise (animal feeding) is dose-dependent, based on the above prediction model. This is an unexpected and very positive result since it could lead to a modification of brain function and faster brain responses to neural stimuli. This evidence could reflect on the reduction of subjective perception of pain in this specific population. In the future, well-designed clinical studies should investigate the effect of cognitive retraining on pain reduction and increased tolerance of pain in older people diagnosed with non-specific low back pain. In any case, holistic and personalized physical therapy intervention is the appropriate solution for chronic pain.

5.2.4 SMART BEAR Platform Integration

The codes and models detailed in this section are specifically tailored to predict the overall balance and cognitive scores during the HOLOBALANCE program. They have been meticulously developed and are now poised for seamless integration within the SMART BEAR platform. They are accessible via the designated link:

https://drive.google.com/drive/folders/18MpTey6QTKWt-gYpXIjRTMpaO_-dVeut and under the name "***SMARTBEAR_AI_Services.zip***" primed to enhance the functionality and efficacy of the SMART BEAR platform, facilitating advanced predictive analytics and decision-making capabilities in the context of the Holobalance training program. Following the transmission of an official email request for authorization to **fotiadis@uoi.gr**, access to all produced versions of the codes/models can be obtained by navigating to the afore mentioned URLs.

5.2.5 Conclusion

In conclusion, the development of two predictive models for overall balance and cognitive scores within the HOLOBALANCE training program marks a significant stride forward. However, challenges such as limited data availability and complexities in assessing cognitive levels and back pain progression in the targeted demographic have been encountered. It is crucial to acknowledge that with additional data, there is potential for gaining deeper insights and refining and expanding these models,

potentially overcoming some of the current limitations. This foundational work also sets the stage for actual advancements in balance-related predictive analytics within the SMART BEAR platform. By addressing these challenges meanwhile leveraging a larger dataset, there is potential for more accurate and effective predictive analytics, ultimately enhancing the efficacy of the HOLOBALANCE program in improving both balance and cognitive function among older adults with balance disorders.

5.2.6 References

[1] A. Zampogna *et al.*, "Fifteen Years of Wireless Sensors for Balance Assessment in Neurological Disorders," *Sensors*, vol. 20, no. 11, Art. no. 11, Jan. 2020, doi: 10.3390/s20113247.

[2] W.-S. Kim, H. Choi, J.-W. Jung, J. S. Yoon, and J. H. Jeoung, "Asymmetry and Variability Should Be Included in the Assessment of Gait Function in Poststroke Hemiplegia With Independent Ambulation During Early Rehabilitation," *Arch. Phys. Med. Rehabil.*, vol. 102, no. 4, pp. 611–618, Apr. 2021, doi: 10.1016/j.apmr.2020.10.115.

[3] C. Silva-Batista *et al.*, "Mobility Rehab visual feedback system for gait rehabilitation in older adults," *J. NeuroEngineering Rehabil.*, vol. 20, no. 1, p. 144, Oct. 2023, doi: 10.1186/s12984-023-01260-2.

[4] O. A. Zobeiri, L. Wang, J. L. Millar, M. C. Schubert, and K. E. Cullen, "Head movement kinematics are altered during balance stability exercises in individuals with vestibular schwannoma," *J. NeuroEngineering Rehabil.*, vol. 19, no. 1, p. 120, Nov. 2022, doi: 10.1186/s12984-022-01109-0.

[5] "Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers - ProQuest." Accessed: Apr. 09, 2024. [Online]. Available: https://www.proquest.com/docview/2578915826/fulltextPDF?_oafollow=false&pq-origsite=primo&parentSessionId=BFlMhjEben5N4YMHNhY%2B1aXIGX6BpySpQ1s5hOl2vxU%3D&sourcetype=Scholarly%20Journals

[6] C. Wang *et al.*, "A Bioinformatics Tool for Identifying Intratumoral Microbes from the ORIEN Dataset," *Cancer Res. Commun.*, vol. 4, no. 2, pp. 293–302, Feb. 2024, doi: 10.1158/2767-9764.CRC-23-0213.

[7] C. Chen, J. Hou, J. J. Tanner, and J. Cheng, "Bioinformatics Methods for Mass Spectrometry-Based Proteomics Data Analysis," *Int. J. Mol. Sci.*, vol. 21, no. 8, Art. no. 8, Jan. 2020, doi: 10.3390/ijms21082873.

[8] N. Iqbal and P. Kumar, "From Data Science to Bioscience: Emerging era of bioinformatics applications, tools and challenges," *Procedia Comput. Sci.*, vol. 218, pp. 1516–1528, Jan. 2023, doi: 10.1016/j.procs.2023.01.130.

[9] V. Tsakanikas *et al.*, "Automated Assessment of Balance Rehabilitation Exercises With a Data-Driven Scoring Model: Algorithm Development and Validation Study," *JMIR Rehabil. Assist. Technol.*, vol. 9, no. 3, p. e37229, Aug. 2022, doi: 10.2196/37229.

[10] V. D. Tsakanikas *et al.*, "Evaluating the Performance of Balance Physiotherapy Exercises Using a Sensory Platform: The Basis for a Persuasive Balance Rehabilitation Virtual Coaching System," *Front. Digit. Health*, vol. 2, Nov. 2020, doi: 10.3389/fdgth.2020.545885.

[11] N. Samartin-Veiga, A. J. González-Villar, and M. T. Carrillo-de-la-Peña, "Neural correlates of cognitive dysfunction in fibromyalgia patients: Reduced brain electrical activity during the execution of a cognitive control task," *NeuroImage Clin.*, vol. 23, p. 101817, 2019.

[12] M. Schiltenwolf *et al.*, "The cognitive impact of chronic low back pain: positive effect of multidisciplinary pain therapy," *Scand. J. Pain*, vol. 17, no. 1, pp. 273–278, 2017.

[13] C. Tsigkanos, L. Gaskell, A. Smirniotou, and G. Tsigkanos, "Static and dynamic balance deficiencies in chronic low back pain," *J. Back Musculoskelet. Rehabil.*, vol. 29, no. 4, pp. 887–893, 2016.

[14] T. Bell, C. Pope, P. Fazeli, M. Crowe, and K. Ball, "The association of persistent low back pain with older adult falls and collisions: a longitudinal analysis," *J. Appl. Gerontol.*, vol. 40, no. 11, pp. 1455–1464, 2021.

[15] A. Meinke, C. Maschio, M. L. Meier, W. Karlen, and J. Swanenburg, "The association of fear of movement and postural sway in people with low back pain," *Front. Psychol.*, vol. 13, p. 1006034, 2022.

[16] M. Schiltenwolf *et al.*, "The cognitive impact of chronic low back pain: positive effect of multidisciplinary pain therapy," *Scand. J. Pain*, vol. 17, no. 1, pp. 273–278, 2017.