



Project Title: Sensing and predictive treatment of frailty and associated co-morbidities using advanced personalized models and advanced interventions

Contract No: 690140

Instrument: Collaborative Project

Call identifier: H2020-PHC-2014-2015

Topic: PHC-21-2015: Advancing active and healthy ageing with ICT: Early risk detection and intervention

Start of project: 1 January 2016

Duration: 36 months

Deliverable No: D4.1

Offline analysis of data (vers. a)

Due date of deliverable: M18 (30/06/2017)

Actual submission date: 30/06/2017

Version: 1.0

Lead Author: Konstantinos Deltouzos (UoP)

Lead partners: Konstantinos Deltouzos, Dimitrios Vlachakis, Evangelia Zacharaki, Emilia Papagiannaki, Spyros Kalogiannis, Evangelia Pippa, Vasilis Megalooikonomou (UoP), Ilias Kalamaras (CERTH), Carlo Mancuso, Tommaso Faetti, Gianluca De Toma (Smartex)



Horizon 2020
European Union funding
for Research & Innovation

EXECUTIVE SUMMARY

The FrailSafe project aims to study all domains of frailty and to create new measures of assessments leading to a model which will be able to better understand, detect, predict, delay, or even revert frailty. To achieve these aims plans are made to devise a comprehensive clinical assessment, to develop a real-life sensing and intervention platform, and to provide a digital patient model of frailty, sensitive to dynamic parameters. Recommendations will be provided to delay frailty, and all this through a safe, unobtrusive, acceptable system and cost-effective system.

The aim of work package WP4 is to develop methods for the offline and online management, fusion, and analysis of multimodal and advanced technology data from social, behavioral, cognitive, and physical activities of frail older people and apply them to manage and analyze new data. Results from the analysis of existing and new data will be also used to create user-profiling virtual models of elderly patients.

The main focus of the deliverable D4.1 is to report on the usage of existing and new developed techniques within the FrailSafe project towards offline data management, preprocessing and analysis. In particular, techniques for data pre-processing are examined involving data cleaning (handling of missing, noisy or inconsistent data, identification and/or removal of outliers), dealing with contaminated/noisy data segments, data integration, data transformation (normalization and aggregation), data reduction (production of reduced representations of data using dimensionality reduction (feature selection), discretisation and numerosity reduction techniques. New techniques for data reduction and summarization of streaming sensor data are also been developed, in order to explore meaningful measuring units for frailty.

Additionally, the state-of-the-art of existing database technologies was examined in order to support the organization of data (both raw signals and analyzed data), including the support of efficient storage and retrieval capabilities such as multidimensional indexing and content-based queries. An investigation of data compression issues and of the ability to analyse compressed data was performed, with respect to the most representative frailty features acquired from feature selection.

Finally, we present our work in the detection of patterns and associations between clinical indicators and frailty states, and in the analysis of multidimensional time series towards revealing associations among signals and symptoms that are connected to the frailty syndrome.

DOCUMENT INFORMATION

Contract Number: H2020-PHC-690140

Acronym: FRAILSAFE

Full title Sensing and predictive treatment of frailty and associated co-morbidities using advanced personalized models and advanced interventions

Project URL <http://frailsafe-project.eu/>

EU Project officer Mr. Jan Komarek

Deliverable number: 4.1 **Title:** Offline analysis of data (vers. a)

Work package number: 4 **Title:** Data Management and Analytics

Date of delivery **Contractual** 30/06/2017 (M18) **Actual** 30/06/2017

Status Draft Final

Nature Report Demonstrator Other

Dissemination Level Public Consortium

Abstract (for dissemination) The main focus of the deliverable D4.1 is to report on the usage of existing and new developed techniques towards offline data management, preprocessing and analysis.

Keywords Data Management, Data Preprocessing, Offline Data Analysis, Multivariate statistical analysis, Activity classification, Multi-dimensional time series analysis

Contributing authors (beneficiaries) Konstantinos Deltouzos, Dimitrios Vlachakis, Evangelia Zacharaki, Emilia Papagiannaki, Spyros Kalogiannis, Evangelia Pippa (UoP), Ilias Kalamaras (CERTH), Carlo Mancuso, Tommaso Faetti, Gianluca De Toma (Smartex)

Responsible author(s) Konstantinos Deltouzos **Email** deltouzos@upatras.gr

Beneficiary UoP **Phone** +30 2610 996 994

Table of contents

Table of contents	5
List of figures	7
List of Tables.....	7
1 Introduction	8
2 Data Management.....	10
2.1 Frailsafe Data Description.....	10
2.1.1 Clinical data (e-CRF)	10
2.1.2 Sensorized strap/vest.....	11
2.1.3 GPS logger (smartphone).....	13
2.1.4 Beacons	14
2.1.5 Games	14
2.1.6 Auxiliary medical devices	16
2.2 FrailSafe Database (HBase).....	16
2.2.1 Database schema	17
2.2.2 NoSQL Databases	20
2.2.3 HBASE.....	23
2.2.4 Hadoop framework.....	24
2.3 FrailSafe Data processing (Apache Spark)	25
2.3.1 Hadoop and Spark.....	25
2.3.2 Spark Features	26
2.3.3 Spark Architecture	27
2.3.4 Resilient Distributed Datasets.....	28
3 Data Preprocessing.....	29
3.1 Clinical data (eCRF)	29
3.2 Handling missing values.....	30
3.3 Data from sensors (time series)	31
4 Data Analysis.....	31
4.1 Group-wise histogram analysis of clinical data	31
4.2 Multivariate statistical analysis of clinical data	36
4.2.1 Calculation of FI_1	36

4.2.2	Calculation of Fl_2	38
4.3	Analysis of data from sensors (multi-dimensional time series)	40
4.3.1	Activity Classification	40
4.3.2	Towards prediction of frailty	48
References.....		51

List of figures

FIGURE 1: VISUALIZATION OF A PARTICIPANT'S OUTDOOR MOVEMENT FOR A SPECIFIC DAY.....	14
FIGURE 2: WP4 CLOUD RESOURCES FOR DATA COLLECTION AND AGGREGATION.	17
FIGURE 3: DATABASE SCHEMA FOR FRAILS SAFE.....	18
FIGURE 4: SENSOR RECORDING ER DIAGRAM.....	19
FIGURE 5: DATA ANALYSIS ER DIAGRAM.....	20
FIGURE 6: HUMAN MOTION IDENTIFICATION MODULE ARCHITECTURE.....	41
FIGURE 7: CLASSIFICATION ACCURACY FOR DIFFERENT SUBSETS OF N-BEST FEATURES (N=10,20,..., 550).	47

List of Tables

TABLE 1: SENSORIZED STRAP/VEST RECORDED PARAMETERS.....	12
TABLE 2: GPS LOGGER RECORDED PARAMETERS.....	13
TABLE 3: BEACON RECORDED PARAMETERS.....	14
TABLE 4: RED WINGS RECORDED PARAMETERS.....	15
TABLE 5: COMPARISON BETWEEN RDBMS AND NOSQL.	22
TABLE 6: CLINICAL METRICS. DOMAINS INVESTIGATED BY THE CLINICAL EVALUATION.	31
TABLE 7: PRE-PROCESSED SIGNALS	42
TABLE 8: ADDITIONAL SIGNALS	43
TABLE 9: SUBJECT DEPENDENT HUMAN MOTION IDENTIFICATION ACCURACY.....	45
TABLE 10: MEAN ACROSS SUBJECTS CONFUSION MATRIX. ROWS REPRESENT THE ACTUAL CLASS AND COLUMNS THE PREDICTED CLASS.....	46
TABLE 11: RELIEFF FEATURE RANKING	47

1 Introduction

Managing FrailSafe's multimodal data is a task of great importance. The large data files that contain the raw sensor data generated by the devices, the medical records of the older people, and the analysis results produced by medical experts or by developed software, need to be stored effectively, aiming to fulfill the data access requirements that arise during offline analysis.

After contacting all partners and the vendors that produce the devices, a summary of the expected input data was made. This summary was used as a guide towards the design of the database. Based on the nature of the data of the FrailSafe project, a NoSQL database was decided that will be more appropriate. Among the numerous NoSQL solutions the Apache HBase was chosen. The motivation behind this choice is that HBase is part of the Hadoop ecosystem, which provides high scalability in data analysis and knowledge discovery algorithms. Towards the analysis of the data, Apache Spark was selected as it is one of the state-of-the-art data processing engines, which can efficiently execute streaming, machine learning or SQL workloads that require fast iterative access to datasets.

The data management and analysis platform has been deployed in the cloud, using the cloud service by Amazon Web Services (AWS) which has been set up by the integrators of the project, Gruppo Sigla. The integration of the submodules which generate the data with the data management and analysis platform has started and is expected to be completed in the following months.

Regarding the task of the offline data analysis, our work has been focused on several areas. Having collected all clinical data from the eCRF platform, the first step to start the data analysis was to summarize and describe the population's behavior. To that end, a group-wise univariate analysis was performed across three different bases: a. Frailty status based analysis, b. Age based analysis and c. Gender based analysis. Additionally, the clinical measurements from eCRF were used for multivariate statistical analysis. Specifically, their predictive ability towards the development of a frailty index was examined. Two different frailty indexes (FI) were computed, one aiming to predict the discrete Fried classification score (FI_1) and one trying to estimate a continuous score as a linear combination of the 5 criteria related to Fried classification (FI_2). The ultimate goal is to investigate whether the proposed frailty indexes are more reliable predictors of frailty transition than standard classification scores.

Finally, multi-dimensional time series analysis has been targeted towards the problem of activity classification, and towards the prediction of frailty. For the former, temporal, and spectral features extracted from the sensor signals (accelerometer and gyroscope) were combined and used to train motion dependent binary classification models. Each individual model was capable to recognize one motion versus all the others. Afterwards the decisions were weighted by their sensitivity on the training set combined by a fusion function. The proposed methodology was evaluated using SVMs for the motion dependent classifiers and

was compared against the common multiclass classification approach optimized using either feature selection or subject dependent classification. For prediction of frailty, some preliminary work has started on the investigation of deep learning techniques for seamless extraction of a features' hierarchy and in-depth analysis of the time series data.

2 Data Management

Managing FrailSafe's multimodal data is a task of great importance. The huge data files that contain the raw sensor data generated by the devices, the medical records of the older people, the annotations generated by the experts (both clinicians and researchers), and the files that contain the analysis results need to be stored effectively, aiming to fulfill the data access requirements that arise during offline analysis.

2.1 Frailsafe Data Description

2.1.1 Clinical data (e-CRF)

The data which are collected by medical personnel during the clinical evaluation, are inserted to the eCRF (Electronic Case Report Form) using the Clinical Web Portal that Gruppo Sigla has developed. The eCRF is mainly composed by a series of forms assessing the participant's clinical status:

- Generalities
- Medical history
- Clinical examination / measurements
- Balance and gait evaluation
- Fried's criteria of frailty
- Mini-Mental State Examination
- Sensory system evaluation
- Nutritional assessment
- Activities of daily living
- Cognitive/emotional evaluation
- Self-evaluation scales

Additionally, the eCRF hosts forms to collect additional data for the participants such as:

- Housing evaluation
- The Big-five assessment
- Social media questionnaire
- Phone follow-up
- Devices registered to the participant during FrailSafe home visits
- Undesirable events during monitoring time

Gruppo Sigla has developed an API which can be used to export the summary of the stored data into csv files, so that they can be inserted to the FrailSafe database and can be used for the analysis.

2.1.2 Sensorized strap/vest

The sensorized strap/vest which is manufactured by our partner Smartex, is equipped with a series of sensors which provide useful measurements for FrailSafe participants. These measurements can be grouped in these categories:

1. **ECG measurements:** The main measurement of this category is the value of the ECG signal coupled together with a quality index which shows how accurate the measurement actually is. This helps ignoring measurements for which the quality is low because the strap was not placed properly. Using the ECG signal, the vest software calculates useful clinical measurements such as Heart Rate, Heart Rate Variability, and R-R interval distance in ECG signal.
2. **IMU measurements:** The strap (WWS) is equipped with a “light” IMU measuring only the participant’s acceleration in X-Y-Z axis (using an accelerometer), while he/she is wearing the strap. The new vest (WWBS) is equipped with one IMU placed on the chest which is measuring the participant’s acceleration, the angular rate, and the magnetic field surrounding the body in X-Y-Z axis using accelerometer, gyroscope and magnetometer, respectively. There are additional two IMUs placed on the arms which just send extracted quaternions. These measurements might not be directly connected with clinical parameters, however they are needed in order to run Fall Detection and Activity Classification algorithms.
3. **Respiration measurements:** The strap is also equipped with a piezoresistive point placed on the thorax, which is used to measure the strain on the thorax caused by the participant’s breathing. The strap uses this measurement to calculate the Breathing Rate, and the Breathing Amplitude of the participant.
4. **Activity attributes:** Additionally, some measurements are provided about the activity the participant performs while wearing the strap. There is a simple activity recognition (lying, standing, walking, running) which however is not as accurate as the activity classification algorithm developed by the UoP. Also, there is a counter measuring the number of steps the participant has done while wearing the strap, and the step period which shows how fast/slow the steps are being done.

These measurements can be summarized in the following table:

Table 1: Sensorized strap/vest recorded parameters.

Recorded parameter	Description	Values (1 unit)	Sampling rate
ECG Value	Electric signal measuring the ECG	0.8 mV	250Hz
ECGquality Value	ECG signal quality	0-255 (0=poor, 255=excellent)	1/5sec
ECGHR Value	Heart rate	Beats/minute	1/5sec
ECGRR Value	R-R intervals	number of samples between R-R peaks	1/5sec
ECGHRV Value	Heart rate variability	ms	1/60sec
AccX-Y-Z Value	Accelerometer in X-Y-Z axes	0.97 10 ⁻³ g	25Hz
GyroX-Y-Z Value	Gyroscope in X-Y-Z axes	0.122 °/s	25Hz
MagX-Y-Z Value	Magnetometer in X-Y-Z axes	0.6 μT	25Hz
RespPiezo Value	Electric signal measuring the chest pressure on the piezoelectric point	0.8 mV	25Hz
RespQuality Value	Respiration signal quality	0-255 (0=poor, 255=excellent)	1/5sec
BR Value	Breathing rate	Breaths/minute	1/5sec
BA Value	Breathing Amplitude	logic levels	1/15sec
Activityenergy Value	estimation of energy activity	is just an estimation (0=no activity, 255=max of activity)	1/5sec
Activityclass Value	Activity performed	0=other, 1=lying, 2=standing/sitting, 3=walking, 4=running	1/5sec
Activity1Pace Value	Step period	ms	1Hz
ActivityPace Value	Pace	steps/min	1/5sec
Q0-Q1-Q2-Q3 values	Quaternions from main electronic device (Q0, Q1, Q2, Q3 components)	Q14 format	25 Hz
QEL0-QEL1-QEL2-QEL3 values	Quaternions from external left arm device (Q0, Q1, Q2, Q3 components)	Q14 format	25 Hz
QER0-QER1-QER2-QER3 values	Quaternions from external right arm device (Q0, Q1, Q2, Q3 components)	Q14 format	25 Hz

The WWS/WWBS data are downloaded by the medical personnel once the home visit session has been performed (at the end of the 5-day period) in a compressed

file format. This file is then uploaded by the medical personnel to the Amazon cloud and the Data Grabber (described in Section 2.2) loads it to the database.

2.1.3 GPS logger (smartphone)

The GPS logger application for the smartphone (developed by CERTH) collects measurements about the geographic location of the participants. The location is obtained by receiving a signal from GPS satellites, thus it is accurate only for the outdoor localization of the participant (in a macroscopic scale). The specific measurements obtained are the latitude, longitude, and elevation of each geographic location, together with the accuracy of the measurement and the orientation of the movement. Combining subsequent points of the location of the participant, we can derive other measurements with more significant clinical value such as the speed of movement, the distance covered etc. The GPS logger application additionally measures the number of steps the participant has made, using the phone sensors.

These measurements can be summarized in the following table:

Table 2: GPS logger recorded parameters.

Recorded parameter	Description	Sampling rate
Latitude	Satellite estimation of the latitude of the geolocation point	variable
Longitude	Satellite estimation of the longitude of the geolocation point	variable
Elevation	Elevation of the geolocation point (sea level)	variable
Speed	Indicative speed of movement	variable
Accuracy	Accuracy of the geolocation	variable
Bearing	Orientation of the movement	variable
Steps	Step counter (based on android sensor)	variable

These recorded data can also be visualized in order to evaluate the outdoor area in which the participants are moving. In Figure 1 we can see an example of such a visualization.

GPS logger currently uploads the data in an FTP server at the premises of CERTH, and Data Grabber collects data from there. In the near future, CERTH will integrate a service in the Amazon cloud and the uploading will be performed directly there.

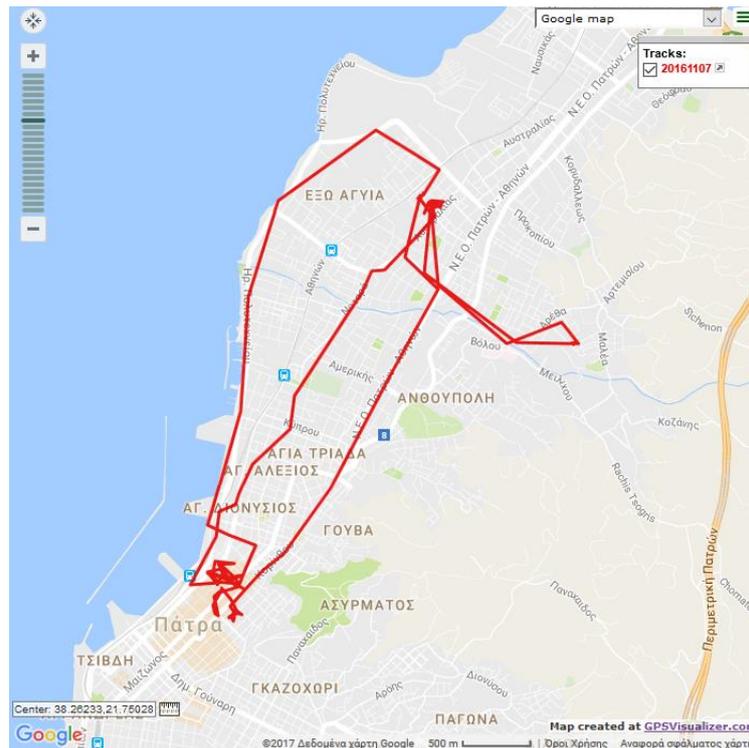


Figure 1: Visualization of a participant's outdoor movement for a specific day.

2.1.4 Beacons

CERTH has developed an application for the smartphone, which can be used with the beacons to perform indoors localization of the participant. Each measurement obtained from the developed app contains the room name that the participant is located. Combining subsequent measurements we can derive the information of the aggregated time the participant has spent in each room.

Table 3: Beacon recorded parameters.

Recorded parameter	Description	Sampling rate
Room name	Label describing the room that the participant is in.	1/15sec

The app is currently in testing phase, thus no data from participants has been generated.

2.1.5 Games

So far, the older persons were exposed to two games, the Red Wings game which was developed by Brainstorm and the Virtual Supermarket game which was developed by CERTH.

The red wings game records a log file with measurements connected with the game such as the speed that the plane is moving, the distance it has covered, the height which it is at, and the number of lives the player still has. Additionally, as the game is operated by the dynamometer, the force of the participant is being collected. Combining the subsequent measurements of the log files, we can derive the total time the participant played the game, the total distance covered (total score), and the maximum force on the dynamometer.

The virtual supermarket game simulates a supermarket, where the participant has a shopping list and needs to navigate inside the supermarket, collect the items of the shopping list and pay at the cashier. The game collects measurements about the time the participant needed to buy each item, the time the participant spent on each part of the supermarket, the value of the products, money paid etc.

These measurements can be summarized in the following tables:

Table 4: Red wings recorded parameters.

Recorded parameter	Description	Sampling rate
Distance	The distance achieved.	60Hz
Force	The force applied by the user.	60Hz
Height	The height of the plane.	60Hz
Lives	The lives remaining.	60Hz
Speed	The plane's speed.	60Hz

Table 5: Virtual supermarket recorded parameters for each session.

Recorded parameter	Description
Total time	Total session duration.
Item types bought	How many types of items were bought (regardless of the quantities for each type).
Item types in list	How many types of items were listed in the shopping list (i.e. which was the goal).
Item quantities bought	The total quantity of the items of all types bought.
Item quantities in list	The total quantity of the items of all types listed in the shopping list (i.e. which was the goal).
Item types not in list	How many types of items were bought that were not listed in the shopping list.
Item quantities not in list	The total quantity of the unlisted items that were bought.
Items value	The total price of the items (and quantities) that were bought.
Money paid	The total price paid by the user at the end.

Currently data are being collected by Brainstorm and CERTH at their local premises, and are being inserted periodically to the FrailSafe database in batches. In the near future both Brainstorm and CERTH will integrate their services directly in the cloud.

2.1.6 Auxiliary medical devices

In the FrailSafe project, there are some auxiliary medical devices which are used to measure clinical parameters. These devices are used a limited number of times, thus they do not generate a large number of data for each participant. The auxiliary devices are:

- **Mobil-o-graph (by Agaedio):** This device is used during clinical examination to measure the arterial stiffness of participants. It collects measurements about blood pressure, heart rate, cardiac output, vascular resistance, augmentation pressure etc.
- **Impedance scale (by FORA):** This device is also used only during clinical examination and collects measurements about weight, body fat, BMI etc.
- **Blood pressure monitor (by FORA):** This device is used during the FrailSafe sessions, so the participant is operating the device 3 times per day in order to measure his/her blood pressure.

2.2 FrailSafe Database (HBase)

One of the main tasks of WP4 is to gather all the data which are either generated by the various medical devices, or collected by the medical personnel, and store them consistently into a database. The data are then analyzed and aggregated towards providing the Virtual Patient Model with which the clinicians will interact to design their interventions. The data collection and aggregation conceptual plan is shown in Figure 2.

In the heart of the system, there is a cluster of 4 Amazon EC2 machines, which hosts two services:

- a) A distributed NoSQL database (Apache HBase) which stores the FrailSafe data, and
- b) A distributed data processing engine (Apache Spark), which is used to process and aggregate the data.

There is an additional Amazon EC2 machine called the “Data Grabber”, which is responsible for collecting the data uploaded to the Amazon cloud by the different submodules of the FrailSafe project, or the external servers of the machine vendors (Agaedio and FORA).

The data collection and integration from these submodules and external servers is still an on-going process. Some features have been fully or partly integrated, while

other features are planned to be integrated at a later point. We can summarize the integration state as follows:

- eCRF data: Fully integrated process
- WWS data: Fully integrated process
- GPS data: Partly integrated process (however data at the moment are not uploaded directly to the amazon cloud)
- FORA data: Integration planned for M19
- Games data: Integration planned for M20-21
- Beacons data: Integration planned for M20-21
- Mobilograph data: Integration planned for M21

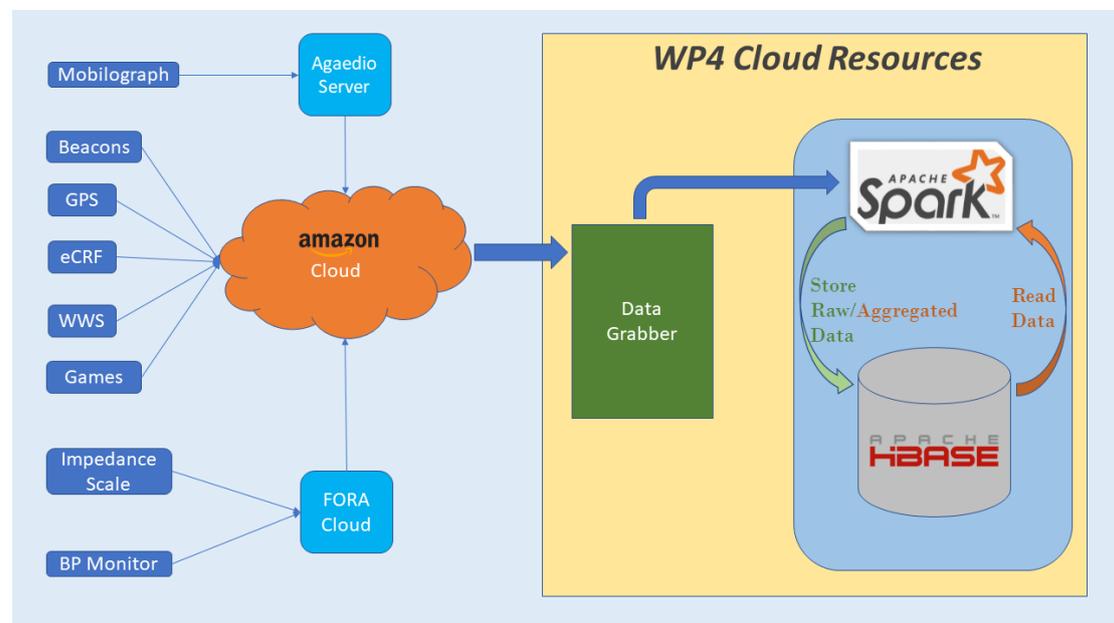


Figure 2: WP4 cloud resources for data collection and aggregation.

2.2.1 Database schema

Dealing with the multimodal data generated in the FrailSafe project is a demanding task. We started our efforts by designing a scheme capable of capturing the complexity of the data and their relations. In this section we will show some parts of the ER (Entity-Relationship) diagram that correspond to the most important data to be stored.

The overview of the schema is presented in Figure 3 and captures the data that are recorded by the FrailSafe devices, the data acquired from the clinical evaluations, and the data that are generated as a result of the offline analysis process.

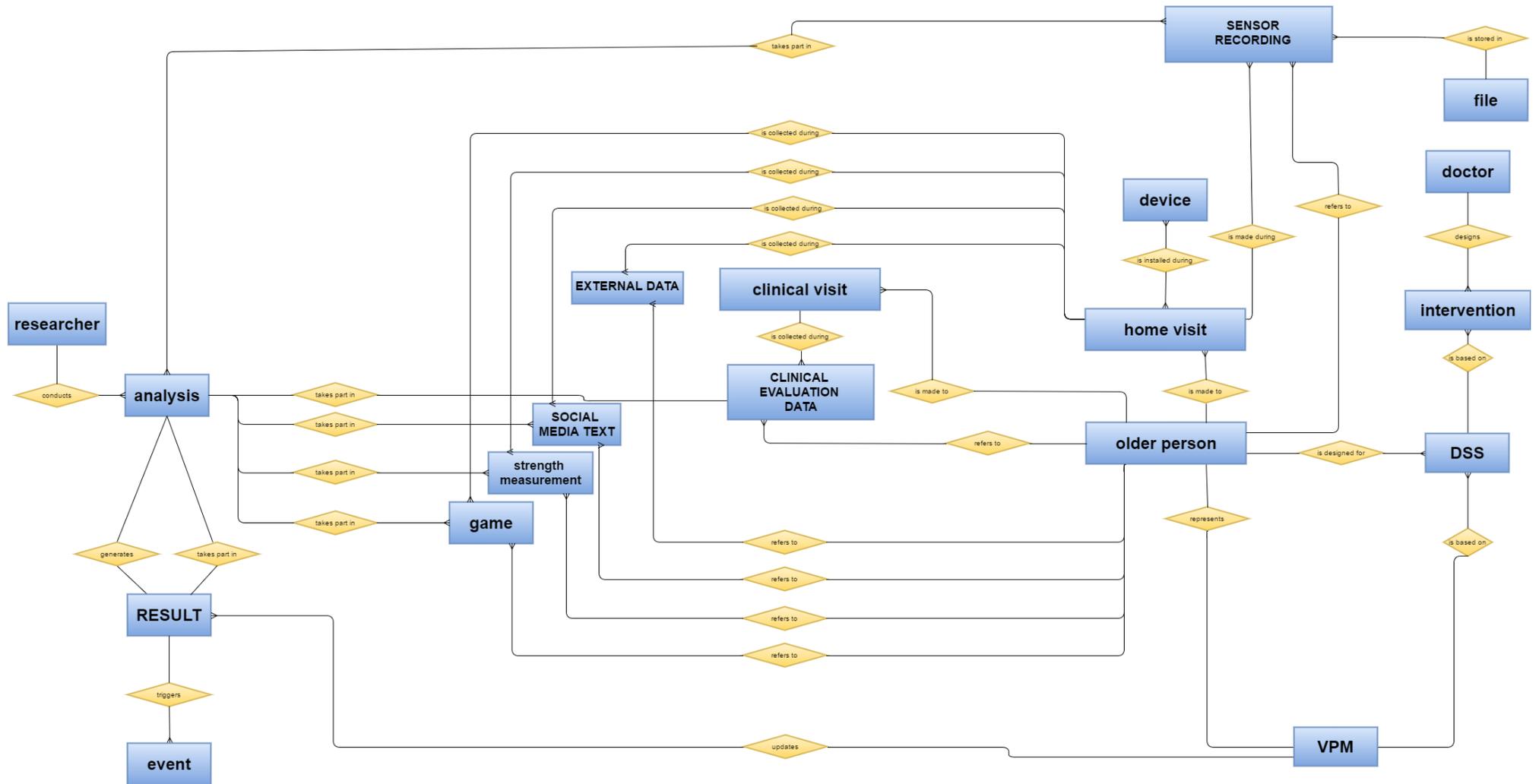


Figure 3: Database Schema for FrailSafe.

Furthermore, the utilization of this schema gives us the ability to store both clinical and physiological data and metadata and thus, gives us the ability to directly search for relationships and correlations among the data.

Sensor recording-related entities

One of the most important entities of the database is the Sensor Recording. By this term we refer to the physiological data coming from the sensorized vest that the FrailSafe participants wear. Important information that should be stored for every sensor recording includes:

- The subjects' ID.
- A timestamp.
- The type of sensor (e.g. ECG, Heart Rate, Respiration Rate).
- The value of the recording.

The actual data (the signals) from the recording are also stored in one or multiple files and thus, one recording is associated with a file. A more detailed description of the recording-related information is shown at the ER diagram in Figure 4. We note here that not all entity attributes are shown due to spacing issues.

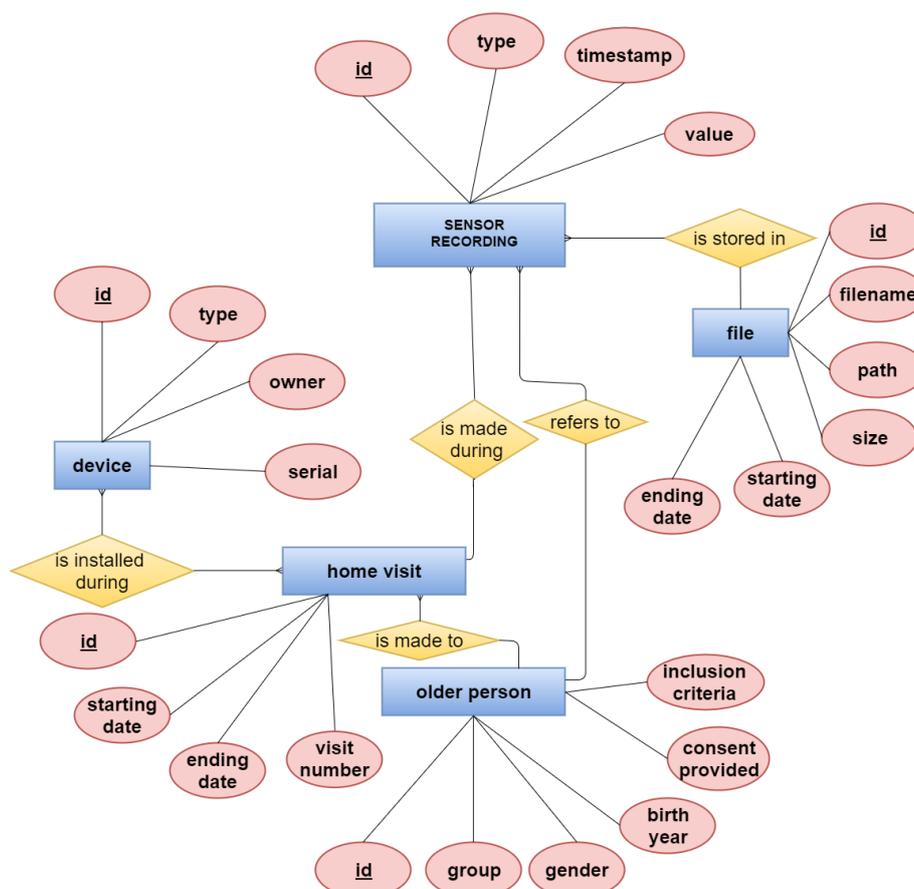


Figure 4: Sensor recording ER diagram.

Analysis-related entities

Another very important task in the database is to keep track of every analysis that takes place and the corresponding results. For this purpose, our database schema includes entities that are able to describe the whole analysis process: the purpose of the analysis, the utilized data and the results.

As we mentioned earlier, since marking events of interest is a very common analysis task, we added the Event entity for the description of events. Also, the result of the analysis might significantly affect the Virtual Patient Model, and subsequently the Decision Support System that is connected to it.

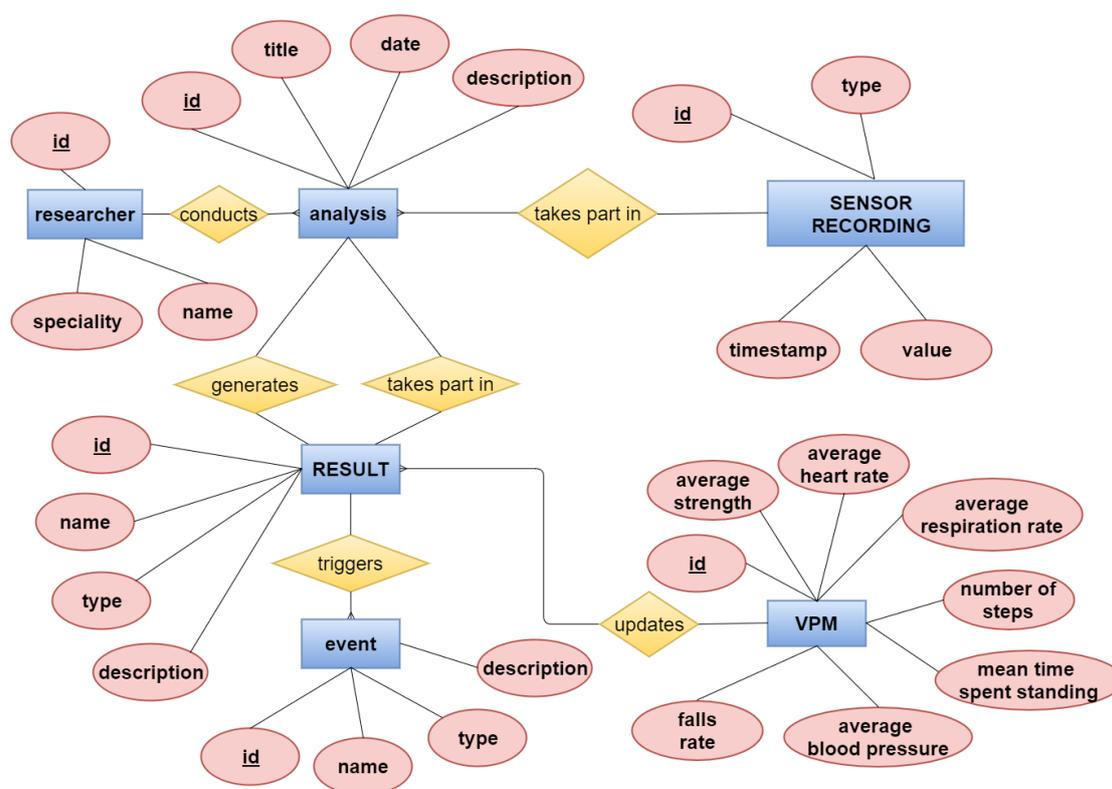


Figure 5: Data Analysis ER diagram.

2.2.2 NoSQL Databases

Based on the nature of the data of the FrailSafe project, a NoSQL (Not-only-SQL) database was decided that will be more appropriate. A NoSQL database is one that has been designed to store, distribute and access data using methods that differ from relational databases (RDBMSs), where data is placed in tables and data schemas are carefully designed before the database is built. NoSQL Databases especially target large sets of distributed data.

NoSQL technology was originally created and used by Internet leaders such as Facebook, Google, Amazon, and others who required database management systems that could write and read data anywhere in the world, while scaling and delivering performance across massive data sets and millions of users.

Today, almost every company and organization has to deliver cloud applications that personalize their customer's experience with their business, with NoSQL being the database technology of choice for powering such systems.

Types of NoSQL Databases

Several different varieties of NoSQL databases have been created to support specific needs and use cases. These fall into four main categories:

Key-value data stores: Key-value NoSQL databases emphasize simplicity and are very useful in accelerating an application to support high-speed read and write processing of non-transactional data. Stored values can be any type of binary object (text, video, JSON document, etc.) and are accessed via a key. The application has complete control over what is stored in the value, making this the most flexible NoSQL model. Data is partitioned *and replicated* across a cluster to get scalability and availability. For this reason, key value stores often do not support transactions. However, they are highly effective at scaling applications that deal with high-velocity, non-transactional data.

Document stores: Document databases typically store self-describing JSON, XML, and BSON documents. They are similar to key-value stores, but in this case, a value is a single document that stores all data related to a specific key. Popular fields in the document can be indexed to provide fast retrieval without knowing the key. Each document can have the same or a different structure.

Wide-column stores: Wide-column NoSQL databases store data in tables with rows and columns similar to RDBMS, but names and formats of columns can vary from row to row across the table. Wide-column databases group columns of related data together. A query can retrieve related data in a single operation because only the columns associated with the query are retrieved. In an RDBMS, the data would be in different rows stored in different places on disk, requiring multiple disk operations for retrieval.

Graph stores: A graph database uses graph structures to store, map, and query relationships. They provide index-free adjacency, so that adjacent elements are linked together without using an index.

How NoSQL Differs from Relational Databases

NoSQL databases are not a direct replacement for a relational database management system (RDBMS). For many data problems, though, NoSQL is a better

match than an RDBMS, as they are designed to support different application requirements (Table 5).

Table 5: Comparison between RDBMS and NoSQL.

RDBMS	NoSQL
Centralized applications (e.g. ERP)	Decentralized applications (e.g. Web, mobile and IOT)
Moderate to high availability	Continuous availability; no downtime
Moderate velocity data	High velocity data (devices, sensors, etc.)
Data coming in from one/few locations	Data coming in from many locations
Primarily structured data	Structured, with semi/unstructured
Complex/nested transactions	Simple transactions
Primary concern is scaling reads	Concern is to scale both writes and reads
Philosophy of scaling up for more users/data	Philosophy of scaling out for more users/data
To maintain moderate data volumes with purge	To maintain high data volumes; retain forever

Advantages over RDBMSs

The advantages of NoSQL databases are no secret, especially when cloud computing has gained wide adoption.

NoSQL databases were created in response to the limitations of traditional relational database technology. When compared against relational databases, NoSQL databases are more scalable and provide superior performance, and their data model addresses several shortcomings of the relational model. More specifically NoSQL databases have been widely adopted in many enterprises for the following reasons:

Elastic scalability: NoSQL databases use a horizontal scale-out methodology that makes it easy to add or reduce capacity quickly and non-disruptively with commodity hardware. This eliminates the tremendous cost and complexity of manual sharding that is necessary when attempting to scale RDBMS.

Big data applications: Given that transaction rates are growing from recognition, there is need to store massive volumes of data. While RDBMSs have grown to match the growing needs, but it's difficult to realistically use one RDBMS to manage such data volumes. These volumes are however easily handled by NoSQL databases.

Database administration: The best RDBMSs require the services of expensive administrators to design, install and maintain the systems. On the other hand,

NoSQL databases require much less hands-on management, with data distribution and auto repair capabilities, simplified data models and fewer tuning and administration requirements. However, in practice, someone will always be needed to take care of performance and availability of databases.

Economy: RDBMSs require installation of expensive storage systems and proprietary servers, while NoSQL databases can be easily installed in cheap commodity hardware clusters as transaction and data volumes increase. This means that you can process and store more data at much less cost.

Performance: By simply adding commodity resources, enterprises can increase performance with NoSQL databases. This enables organizations to continue to deliver reliably fast user experiences with a predictable return on investment for adding resources—again, without the overhead associated with manual sharding.

High Availability: NoSQL databases are generally designed to ensure high availability and avoid the complexity that comes with a typical RDBMS architecture that relies on primary and secondary nodes. Some “distributed” NoSQL databases use a masterless architecture that automatically distributes data equally among multiple resources so that the application remains available for both read and write operations even when one node fails.

Global Availability: By automatically replicating data across multiple servers, data centers, or cloud resources, distributed NoSQL databases can minimize latency and ensure a consistent application experience wherever users are located. An added benefit is a significantly reduced database management burden from manual RDBMS configuration, freeing operations teams to focus on other business priorities.

Flexible Data Modeling: NoSQL offers the ability to implement flexible and fluid data models. Application developers can leverage the data types and query options that are the most natural fit to the specific application use case rather than those that fit the database schema. The result is a simpler interaction between the application and the database and faster, more agile development.

2.2.3 HBASE

Apache HBase is a massively scalable, distributed big data store in the Apache Hadoop ecosystem. It is an open-source, non-relational, versioned database which runs on top of Amazon S3 (using EMRFS) or the Hadoop Distributed Filesystem (HDFS), and it is built for random, strictly consistent real-time access for tables with billions of rows and millions of columns. Additionally, Apache HBase has tight integration with Apache Hadoop, Apache Hive, and Apache Pig, so you can easily combine massively parallel analytics with fast data access. Apache HBase's data model, throughput, and fault tolerance is a good match for workloads in ad tech, web analytics, financial services, applications using time-series data, and many more.

Features and Benefits

Fast Performance at Scale: Apache HBase is designed to maintain performance while scaling out to hundreds of nodes, supporting billions of rows and millions of columns. It utilizes Amazon S3 (with EMRFS) or the Hadoop Distributed Filesystem (HDFS) as a fault-tolerant datastore. Amazon EMR supports a wide variety of instance types and Amazon EBS volumes, so you can customize the hardware of your cluster to optimize for cost and performance. Additionally, you can use Apache Phoenix for low-latency SQL over massive HBase tables or creating secondary indexes for increased performance.

Run Analytics Workloads on HBase Tables: Through tight integration with projects in the Apache Hadoop ecosystem, you can easily run massively parallel analytics workloads on data stored in HBase tables. You can easily install Apache Phoenix, Apache Hadoop, Apache Hive, Apache Pig, and other open-source big data applications on your Amazon EMR cluster alongside Apache HBase, and utilize these tools to run reporting, SQL queries, or other analytics workloads on your data in Apache HBase. Also, you can use these tools to bulk import/export data into Apache HBase tables, or use Apache Hive to join data from Apache HBase with external tables on Amazon S3.

Flexible Data Model: Apache HBase is wide-column store, allowing you to define arbitrary columns for each row for filtering purposes. Additionally, HBase adds a timestamp to each cell and can keep previous versions, allowing you to easily store and access the lineage of a dataset. Each cell is a byte array and can store a payload in the MB range, giving you flexibility in data types stored in your table. Apache Phoenix and Apache Hive enable SQL access over Apache HBase tables.

2.2.4 Hadoop framework

Apache HBase runs on top of the Hadoop, which is an open-source software framework used for distributed storage and processing of big data using the MapReduce programming model. It consists of computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework.

The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part which is a MapReduce programming model. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This approach takes advantage of data locality, where nodes manipulate the data they have access to. This allows the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.

The base Apache Hadoop framework is composed of the following modules:

- *Hadoop Common* – contains libraries and utilities needed by other Hadoop modules;
- *Hadoop Distributed File System (HDFS)* – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster;
- *Hadoop YARN* – a platform responsible for managing computing resources in clusters and using them for scheduling users' applications; and
- *Hadoop MapReduce* – an implementation of the MapReduce programming model for large-scale data processing.

2.3 FrailSafe Data processing (Apache Spark)

Apache Spark is a fast, in-memory data processing engine with elegant and expressive development APIs to allow data workers to efficiently execute streaming, machine learning or SQL workloads that require fast iterative access to datasets. With Spark running on Apache Hadoop YARN, developers everywhere can create applications to exploit Spark's power, derive insights, and enrich their data science workloads within a single, shared dataset in Hadoop.

Spark has several advantages compared to other big data and MapReduce technologies like Hadoop and Storm. First of all, Spark gives us a comprehensive, unified framework to manage big data processing requirements with a variety of data sets that are diverse in nature (text data, graph data etc.) as well as the source of data (batch vs. real-time streaming data). Spark enables applications in Hadoop clusters to run up to 100 times faster in memory and 10 times faster even when running on disk. Spark lets you quickly write applications in Java, Scala, or Python. It comes with a built-in set of over 80 high-level operators. And you can use it interactively to query data within the shell. In addition to Map and Reduce operations, it supports SQL queries, streaming data, machine learning and graph data processing. Developers can use these capabilities stand-alone or combine them to run in a single data pipeline use case.

2.3.1 Hadoop and Spark

Hadoop as a big data processing technology has been around for 10 years and has proven to be the solution of choice for processing large data sets. MapReduce is a great solution for one-pass computations, but not very efficient for use cases that require multi-pass computations and algorithms. Each step in the data processing workflow has one Map phase and one Reduce phase and you'll need to convert any use case into MapReduce pattern to leverage this solution.

The Job output data between each step has to be stored in the distributed file system before the next step can begin. Hence, this approach tends to be slow due to

replication & disk storage. Also, Hadoop solutions typically include clusters that are hard to set up and manage. It also requires the integration of several tools for different big data use cases (like Mahout for Machine Learning and Storm for streaming data processing).

If you wanted to do something complicated, you would have to string together a series of MapReduce jobs and execute them in sequence. Each of those jobs was high-latency, and none could start until the previous job had finished completely.

Spark allows programmers to develop complex, multi-step data pipelines using the directed acyclic graph (DAG) pattern. It also supports in-memory data sharing across DAGs, so that different jobs can work with the same data.

Spark runs on top of existing Hadoop Distributed File System (HDFS) infrastructure to provide enhanced and additional functionality.

2.3.2 Spark Features

Spark takes MapReduce to the next level with less expensive shuffles in the data processing. With capabilities like in-memory data storage and near real-time processing, the performance can be several times faster than other big data technologies.

Spark also supports lazy evaluation of big data queries, which helps with optimization of the steps in data processing workflows. It provides a higher-level API to improve developer productivity and a consistent architect model for big data solutions.

Spark holds intermediate results in memory rather than writing them to disk which is very useful especially when you need to work on the same dataset multiple times. It's designed to be an execution engine that works both in-memory and on-disk. Spark operators perform external operations when data does not fit in memory. Spark can be used for processing datasets that are larger than the aggregate memory in a cluster.

Spark will attempt to store as much as data in memory and then will spill to disk. It can store part of a data set in memory and the remaining data on the disk. You have to look at your data and use cases to assess the memory requirements. With this in-memory data storage, Spark comes with a performance advantage.

Other Spark features include:

- Supports more than just Map and Reduce functions.
- Optimizes arbitrary operator graphs.
- Lazy evaluation of big data queries which helps with the optimization of the overall data processing workflow.
- Provides concise and consistent APIs in Scala, Java and Python.

- Offers interactive shell for Scala and Python. This is not available in Java yet.

Spark is written in Scala Programming Language and runs on Java Virtual Machine (JVM) environment. It currently supports the following languages for developing applications using Spark:

- Scala
- Java
- Python
- Clojure
- R

2.3.3 Spark Architecture

Spark Architecture includes following three main components:

Data Storage:

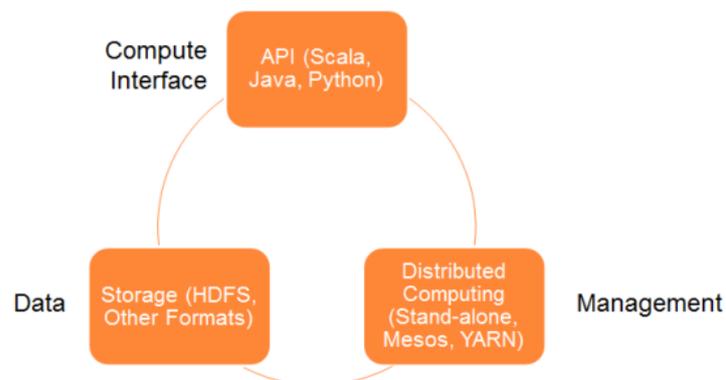
Spark uses HDFS file system for data storage purposes. It works with any Hadoop compatible data source including HDFS, HBase, Cassandra, etc.

API:

The API allows the application developers to create Spark based applications using a standard API interface. Spark provides API for Scala, Java, and Python programming languages.

Resource Management:

Spark can be deployed as a Stand-alone server or it can be deployed on a distributed computing framework like Mesos or YARN.



2.3.4 Resilient Distributed Datasets

Resilient Distributed Dataset (or RDD) is the core concept in the Spark framework. RDD can be thought as a table in a database. It can hold any type of data. Spark stores data in RDD on different partitions. RDDs help with rearranging the computations and optimizing the data processing. They are also fault tolerant because an RDD knows how to recreate and recompute the datasets. RDDs are immutable. You can modify an RDD with a transformation but the transformation returns you a new RDD whereas the original RDD remains the same.

RDD supports two types of operations:

- Transformation
- Action

Transformation: Transformations don't return a single value, they return a new RDD. Nothing gets evaluated when you call a Transformation function, it just takes an RDD and returns a new RDD.

Some of the Transformation functions are map, filter, flatMap, groupByKey, reduceByKey, aggregateByKey, pipe, and coalesce.

Action: Action operation evaluates and returns a new value. When an Action function is called on a RDD object, all the data processing queries are computed at that time and the result value is returned.

Some of the Action operations are reduce, collect, count, first, take, countByKey, and foreach.

3 Data Preprocessing

3.1 Clinical data (eCRF)

The analysis of clinical data from eCRF requires a number of steps that include the conversion and standardization of data as well as handling their missing values. Details on the individual steps are provided next.

Exclusion criteria (for samples and variables): An entire record (subject) is excluded if the Fried classification score or any of the 5 criteria used for Fried classification (involuntary weight loss, slow walking speed, poor handgrip quality, reported exhaustion, low physical activity) are not available. In respect to a first step of feature reduction, categorical variables, such as 'gender', are currently excluded due to their sparsity that hinders the analysis. Comorbidities are summarized as the number of significant co-morbidities. In respect to medication, the number of different drugs is currently computed ignoring the frequency of delivery. All numerical and ordinal variables are retained, except of variables that have many missing values (>20%) for which the imputation is highly prone to errors.

Label conversions: Binary variables ('True'/'False' or 'Yes'/'No') are converted to 1/0. Ordinal variables (categorical but with ordering) are converted to a pseudo-score in the range of [0, 1], as shown in the next Table.

Variable	Labels from E-CRF	Numeric conversions
frailty	'Non frail' 'Pre-frail' 'Frail'	0 0.5 1
medication delivery	'999' '888' '777'	empty 0.3 0.6
vision	'Sees poorly' 'Sees moderately' 'Sees well'	0.3 0.6 0.9
audition	'Hears poorly' 'Hears moderately' 'Hears well'	0.3 0.6 0.9
activity_regular	'< 2 h per week' '> 2 h and < 5 h per week' '> 5 h per week'	0.3 0.6 0.9
balance_single	'test non realizable' '<5 sec'	empty 0.3

	'>5 sec'	0.9
smoking	'Never smoked'	0
	'Past smoker (stopped at least 6 months)'	0.5
	'Current smoker'	1

The rest of the data (the numeric variables) are scaled also in the range [0 1].

3.2 Handling missing values

There are several ways to handle missing values, the choice of which affects the subsequent results. The simplest would be to replace missing values with the mean, the median, the most commonly occurring value (appropriate also for categorical data), or an interpolated estimate (usually for time-series data). Such substitutions are more appropriate for univariate analysis because they examine only the studied variable, e.g. the assumption behind the mean substitution is to retain the sample mean. Mean imputation however attenuates any correlations involving the variables that are imputed. For multivariate analysis, regression imputation can be performed, in which a regression model is estimated to predict missing values of a variable based on other variables. The model is learnt from available samples and then used to impute values in cases where that variable is missing. The *Least Absolute Shrinkage And Selection Operator* (Lasso) was selected as the regression model for each variable that had missing values. The training phase involves the computation of a vector of weights (β -coefficients) which, when multiplied with the known variables (x), will approximate the variable to be imputed (y). Mathematically it is expressed as a minimization problem

$$\hat{\beta}_{\text{lasso}} = \arg \min_{\beta} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

where X is the covariate matrix and λ a parameter that determines the amount of regularization. The 2nd term in the equation above enforces sparsity of variables (when a coefficient becomes 0, the corresponding variable can be eliminated from the model), and thus results to more generalizable models. A maximum number of 15 coefficients was enforced. The level of regularization (parameter λ) was varied and for each value the *Mean Prediction Squared Error (MSE)* was calculated in a 10-fold cross validation setting. The λ resulting in minimum MSE was finally selected for the prediction model. The training set included all records that didn't have missing values.

For the remaining records, each missing variable was imputed using the corresponding regression model, if the variables it depends on (i.e. the ones with non-zero β coefficients) were available. If the dependencies included missing values, then imputation was not possible and the record could not be used in the subsequent analysis.

3.3 Data from sensors (time series)

Time synchronization: All sensor data were time synchronized to allow multi-channel analysis. The frequency of 25Hz was selected as reference space, since several of the sensor data are sampled at this rate. The recordings of ECG signal were downsampled from 250Hz to 25Hz, whereas other recordings (Breathing Amplitude, Breathing Rate, Heart Rate, R-R intervals, Heart rate variability, Activity classification) were upsampled. Some of the recordings were slightly time-shifted (~15msec) in order to be synchronized with the rest.

4 Data Analysis

4.1 Group-wise histogram analysis of clinical data

Having collected all clinical data from the eCRF platform, the first step to start the data analysis was to summarize and describe the population's behavior. To that end, a univariate analysis was performed across three different bases:

1. Frailty status based analysis
2. Age based analysis
3. Gender based analysis

Regarding the frailty-based analysis, the participants were split in three categories, Frail, Pre-Frail, Non-Frail, according to the Fried categorization that had been performed at the beginning of the study. Similarly, for the gender-based analysis the population was grouped in male and female participants, while for the age-based analysis the participants were divided in three equally separated categories according to the distribution of the variable "birth year".

After splitting the dataset in the aforementioned categories, the univariate analysis was performed across the domains that had been defined by the clinicians in D2.3. The table with the updated domains is shown below:

Table 6: Clinical metrics. Domains investigated by the clinical evaluation.

Items	
Medical Domain (M)	Number of Comorbidities (M)
	Comorbidity's impact (M, P, s, ψ)
	Polymedication (M, p, c)
	Hospitalisations (M)

	Orthostatic hypotension (M, p)
	Visual impairment (M, S, p)
	Hearing impairment (m, S, c)
General Condition Domain (M, ψ)	Unintentional weight loss (M, ψ)
	Self-reported exhaustion (M, p, ψ)
Lifestyle domain (P, M, ψ , s)	Smoking (M, ψ , p, s)
	Alcohol (M, ψ , S)
	Physical Activity (P, M, ψ , s)
Functional capacity domain (M, P, s, c, Ψ)	Basic Activities of Daily living (M, P, s, c, Ψ)
	Instrumental Activities of Daily Living (M, P, s, c, Ψ)
Physical Condition (P, m, c)	Balance (single foot standing) (P, m)
	Gait-related task speed* (P, c) (Timed Get Up and Go test)
	Gait - speed 4 m (P, m)
	Lower limb strength (P, m)
	Grip strength –dynamometer (P, m)
	Low physical activity (P, M, s, ψ)
	Falls (P, m, Ψ)
	Fractures (P, M)
Nutritional domain (M, ψ , c, s)	Too low BMI (M, ψ , p, c, s)
	Too high BMI (M, ψ , P, c, s)
	High waist circumference (M, ψ , P, c, s)
	Lean body mass (M, P, ψ)
	MNA screening and total (when applicable) score (M, ψ , p, c, s)
Cognitive Domain (C, ψ , m, s)	MMSE scores (C, ψ , m)
	MoCA score (C, ψ , m)
	Subjective memory complaint (C, ψ , m, s)
	Natural language analysis (C, ψ)
Psychological Domain (ψ , S, c)	GDS-15*(ψ , S, c)
	Self-rated anxiety (ψ , S, c)

	Natural language analysis (C, Ψ)
Social Domain (S, Ψ , m)	Living conditions (S, Ψ , p, m)
	Leisure activities (S, Ψ , p, m)
	Membership of a club (S, Ψ , p, m)
	Number of visits and social interactions per week (S, Ψ , p)
	Number of telephone calls exchanged per week (S, ψ , m)
	Approximate time spent on phone per week (S, ψ , m)
	Approximate time spent on videoconference per week (S, ψ)
	Number of written messages sent by the participant per week (S, Ψ , m, p)
Environmental Domain (S, P, m)	Subjective suitability of the housing environment according to participant's evaluation (S, P, m)
	Subjective suitability of the housing environment according to investigator's evaluation (S, P, m)
	Number of steps to access house (P, S, m)
Wellness domain (Ψ , S, M, P, c)	Quality of life self-rating (Ψ , S, M, P, c)
	Self-rated health status (M, Ψ)
	Self-assessed change since last year (M, ψ)
	Self-rated anxiety (Ψ , S, M, P, c)
	Self-rated pain (M, P, ψ)
Tags (reflecting impact of each item on each of the aspects of frailty)	
Physical/functional: P dominant, p recessive	
Medical: M dominant, m recessive	
Social: S dominant, s recessive	
Cognitive: C dominant, c recessive	
Psychological: Ψ dominant, ψ recessive	
Abbreviations:	
BMI: Body Mass Index, GDS-15: Geriatric Depression Scale 15 items, MMSE: Mini Mental State Examination, MNA: Mini Nutritional Assessment, MoCA: Montreal Cognitive Assessment.	

For each clinical domain that contained numerical data, box plots were extracted and provided to the clinicians in order for them to evaluate the presence of outliers (a table with the outliers' ID was provided as well). Each clinical center studied the

IDs of the participants that were detected as outliers and either corrected the values that fell outside the norms, or decided that the values were clinically plausible and should no longer be considered as outliers.

Another important part of the univariate analysis was the identification of missing values, which enabled the clinical team to fill in values that either had accidentally been forgotten or were thought to be filled, but for technical reasons the data had not been uploaded to the eCRF platform.

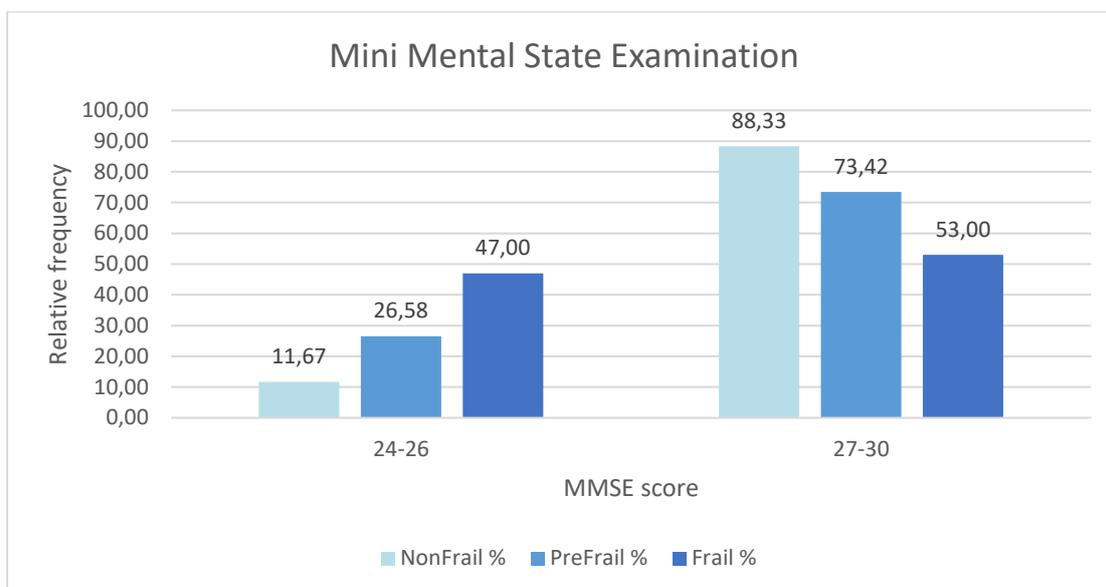
After having “cleaned” the data as mentioned above, some descriptive measures were extracted for each variable of each domain, including the total number of participants of each group, as well as the minimum, maximum and mean value of each numerical variable. Subsequently, we divided the values of the numerical variables into ranges that had mostly been defined by the clinicians in order for the data to be clinically meaningful, while in some cases the values’ split was performed according to percentiles of the data distribution, or even randomly. For ordinal variables there was no need to implement such a procedure, as the values are already divided in categories.

The next step was to count the number of the participants of each group (e.g. Frail, Non-Frail, or Female, Male) whose responses belonged in each of the categories defined by each variable’s ranges. To that end, a table was extracted for each domain’s variable that contained the variable’s ranges and the number of the participants of each group that fell in each range. For visualization purposes, the percentage of the number of participants that fell in each range was calculated, so that it would be easy to compare different groups with different number of participants each.

Two examples of the table described above are shown below, along with the corresponding chart.

1. Frailty-based univariate analysis example for numerical variable

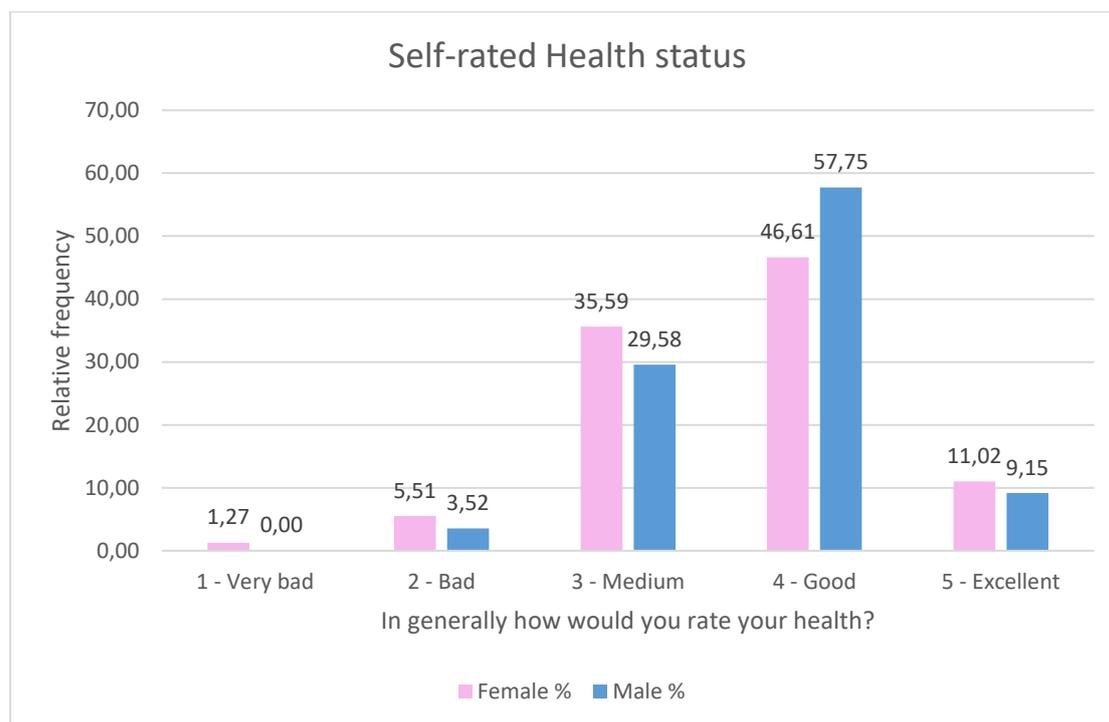
Cognitive Domain						
Mini Mental State Examination						
Ranges	NonFrail	PreFrail	Frail	NonFrail %	PreFrail %	Frail %
24-26	14	42	47	11,67	26,58	47,00
27-30	106	116	53	88,33	73,42	53,00



2. Gender-based univariate analysis example for ordinal variable

Wellness Domain				
Quality of life self-rating				
Row	Female	Male	Female %	Male %
1 - Very bad	3	0	1,27	0,00
2 - Bad	13	5	5,51	3,52
3 - Medium	84	42	35,59	29,58
4 - Good	110	82	46,61	57,75
5 - Excellent	26	13	11,02	9,15

The aforementioned analysis steps and the corresponding results were forwarded to the clinical team for further evaluation of the population. Observing the participants' summarized data from a clinical point of view provides the privilege of deciding in a relatively quick way whether a variable separates the population into groups well or not.



4.2 Multivariate statistical analysis of clinical data

After preprocessing and conversion to numerical data as described in section 3.1, the clinical measurements from eCRF were used for statistical analysis. Specifically, their predictive ability towards the development of a frailty index was examined. Two different frailty indexes (FI) were computed, one aiming to predict the discrete Fried classification score (FI_1) and one trying to estimate a continuous score as a linear combination of the 5 criteria related to Fried classification (FI_2). The ultimate goal is to investigate whether the proposed frailty indexes are more reliable predictors of frailty transition than standard classification scores.

4.2.1 Calculation of FI_1

Lasso linear regression was performed to select a subset of variables and estimate their β coefficients, in order to build a predictive model having the best possible correlation with Fried's score. A set of different values for λ , a parameter which controls the number of retained coefficients and thus the risk for overfitting, was tested and the one with the smallest fitting error was selected.

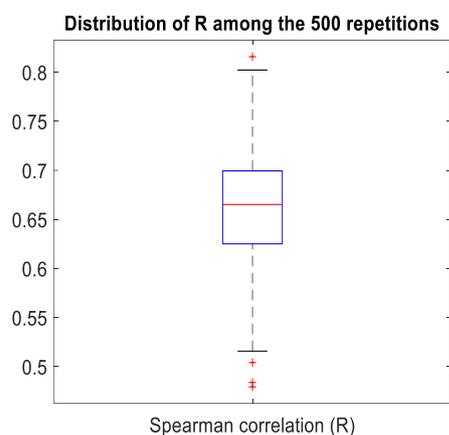
Results

The (15) selected β -coefficients that are used to build the FI_1 score are shown in the following Table next to the corresponding variables and sorted by decreasing significance (absolute value). Positive values mean that an increase of the corresponding variable adds to the frailty index, whereas negative values mean that

an increase of the corresponding variable reduces the frailty index. For example the larger the 'birth_year', the smaller the age and the less frail is expected to be the person. Spearman's rank correlation coefficient (R) between Fried's score and the proposed FI_1 score was 0.726 showing good correlation.

Clinical variables	β (for FI_1 -score)
Intercept (β_0)	1.0420
'alcohol_units'	-0.3963
'bmi_score'	-0.2432
'falls_one_year'	0.1642
'depression_total_score'	-0.1523
'birth_year'	-0.1401
'hospitalization_three_years'	0.1168
'raise_chair_time'	0.0947
'screening_score'	0.0938
'anxiety_perception'	-0.0823
'cognitive_total_score'	0.0786
'stairs_number'	0.0713
'balance_single'	-0.0373
'leisure_out'	-0.0140
'ortho_hypotension'	-0.0125
'smoking'	-0.0077

The previous prediction model was calculated by fitting all the data at once. Subsequently we studied the robustness of variable selection by calculating the predictive model using bootstrapping with 500 repetitions. At each repetition, 70% of the records were randomly selected for Lasso regression and the resulting model was applied on the remaining 30%, i.e. the 70% were the development subset and the 30% the validation subset. The most frequently selected variables among the 500 bootstrap repetitions are shown next (on the right). It can be seen that a different set of variables (compared to regression on the whole training set) had the highest ranks. This might be related to the small training set that was available, or to the presence of features that are correlated. Spearman's correlation between predicted and actual values was also calculated for every repetition and illustrated in the boxplot on the left.



Clinical variables	Selection frequency
'activity_regular'	1
'birth_year'	0.94
'depression_total_score'	0.888
'raise_chair_time'	0.888
'pain_perception'	0.838
'balance_single'	0.832
'cognitive_total_score'	0.826
'bmi_score'	0.662
'audition'	0.614
'social_visits'	0.598
'mmse_total_score'	0.592
'alcohol_units'	0.5
'leisure_club'	0.46
'health_rate'	0.422
'hospitalization_one_year'	0.406

4.2.2 Calculation of Fl_2

A similar analysis (as above) was performed but this time the variable to be predicted is a continuous frailty index expressed as a linear combination of the 5 criteria related to Fried score:

$$Y = \sum_{k=1}^5 \alpha_k f_k, \text{ subject to } \sum_{k=1}^5 \alpha_k = 1$$

where $f_k \in \{\text{involuntary weight loss, slow walking speed, poor handgrip quality, reported exhaustion, low physical activity}\}$. In this case, we want to estimate jointly the β -coefficients and α -coefficients, such that for every record x_i , $i = 1, \dots, n$, where n is number of records,

$$\sum_{j=0}^m \beta_j x_{ij} \approx \sum_{k=1}^5 \alpha_k f_{ik},$$

$$\text{subject to } \sum_{k=1}^5 \alpha_k = 1$$

and β sparse,

where m is the number of β_j coefficients (number of variables), β_0 is the intercept

and $x_0 = 1$. In a vector form, if $a = \begin{bmatrix} a_1 \\ \vdots \\ a_5 \end{bmatrix}$ and $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}$, we seek for $p = \begin{bmatrix} a \\ \beta \end{bmatrix} \in$

\mathbb{R}^{m+6} . Given a dataset $X_{n \times (m+1)} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nm} \end{bmatrix}$ that should be mapped to

frailty criteria $F = \begin{bmatrix} fx_{11} & \cdots & f_{15} \\ \vdots & \ddots & \vdots \\ f_{n1} & \cdots & f_{n5} \end{bmatrix}$, this problem can be expressed as a generalized

Lasso regression under equality constraints:

$$p = \operatorname{argmin} \frac{1}{2} \|Wp\|_2^2 + \lambda \|Dp\|_1,$$

$$\text{subject to } Ap = 1$$

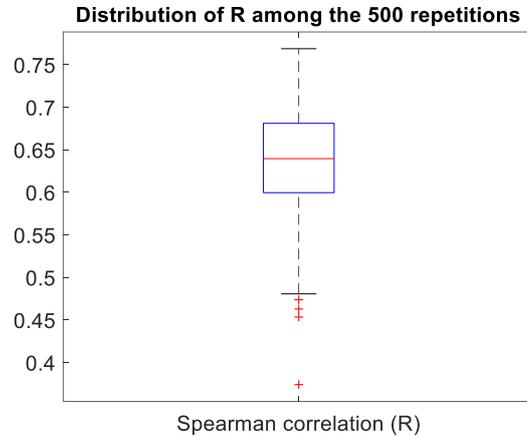
where $W = [F \quad -X] \in \mathbb{R}^{n \times (m+6)}$. $D \in \mathbb{R}^{n \times (m+6)}$ is a penalty matrix used to enforce sparsity constraints only on β , and is $D = [\emptyset \quad I]$, where $I \in \mathbb{R}^{(m+1) \times (m+1)}$ is the identity matrix and $\emptyset \in \mathbb{R}^{(m+1) \times 5}$ is a matrix filled with zeros. Various algorithms to solve the constrained Lasso, including quadratic programming, were investigated following the work in (Gaines2016) and finally the alternating direction method of multipliers (ADMM) was used based on the Gurobi (<http://www.gurobi.com>) solver.

Results

The coefficients (weights) for the calculated prediction models according to the 2nd approach are shown by decreasing significance (absolute value) in the next Table.

Fried's criteria	α (for Y-score)	Clinical variables	β (for FI2-score)
		Intercept (β_0)	-0.1525
'weight_loss'	0.3501	'raise_chair_time'	-0.1830
'low_physical_activity'	0.3129	'activity_regular'	0.1602
'exhaustion_score'	0.1365	'house_suitable_participant'	-0.1236
'gait_speed_slower'	0.1341	'pain_perception'	-0.0748
'grip_strength_abnormal'	0.0664	'depression_total_score'	-0.0243
		'leisure_club'	0.0232

The Spearman correlation was 0.657. Evaluation of the repeatability of the results was again performed by 500 bootstrap repetitions with 70% of the data used for training and 30% for testing. The variation of Spearman's correlation is shown next.



4.3 Analysis of data from sensors (multi-dimensional time series)

The recordings from the vest/strap were analyzed offline for classification of activities of daily living (ADL), whereas in our future work we will focus more on the prediction of a frailty index.

4.3.1 Activity Classification

In this section, we present our work on human motion identification from multi-parametric sensor data acquired from accelerometers and gyroscopes using a large number of time-domain and frequency domain features in order to be used as part of an end-to-end system from accelerometers and gyroscopes using a large number of time-domain and frequency domain features in order to be used as part of an end-to-end system for sensing and predicting treatment of frailty and associated comorbidities using advanced personalized models and advanced interventions. The proposed methodology was evaluated in multiparametric data from 30 subjects. The evaluation of the multiclass SMO (Sequential Minimal Optimization) classification algorithm showed that a mean accuracy of 96% was achieved. Feature ranking investigation and evaluation of the classification models using subsets of features were performed and revealed the most significant features for the classification task. The use of the most discriminative features ($N = 550$) achieved accuracy equal to the accuracy achieved when all features are used.

The aim of this work is to analyze the data collected from the FrailSafe participants during the home visits, and accurately identify the activity they are performing. By annotating the collected data and combining other physiological measurements, we are able to provide meaningful clinical parameters towards the medical partners of

FrailSafe, such as average heart rate while walking. These parameters are being stored in the Virtual Patient Model, with which the clinicians can interact in order to monitor the state of the older persons and design the interventions.

Architecture

The proposed classification methodology can be used as a core module in order to discriminate the detected motions to six basic activities: walking, walking-upstairs, walking-downstairs, sitting, standing and laying. The block diagram of the overall workflow for learning the activity classifiers is illustrated in Figure 6.

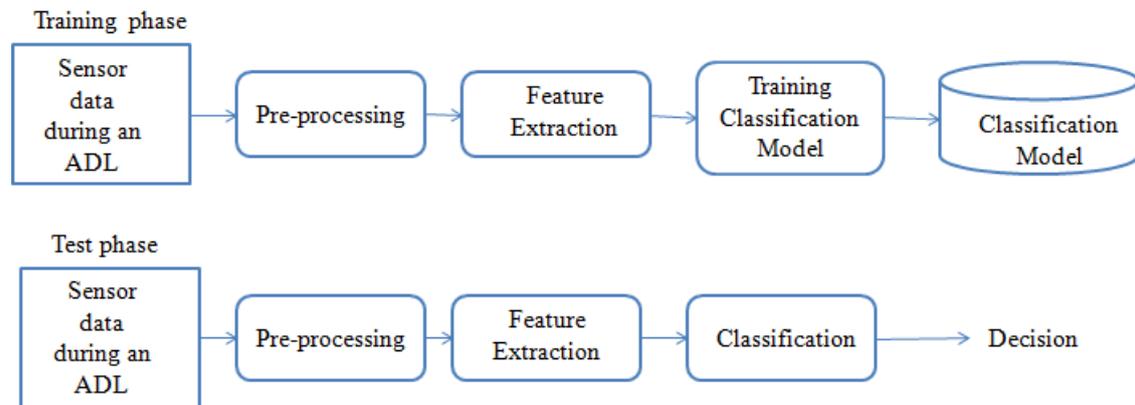


Figure 6: Human Motion Identification Module Architecture

The multi-parametric sensor (accelerometer and gyroscope) data are pre-processed as in (Anguita et al., 2012; Reyes-Ortiz et al., 2013) by applying noise filters and then sampled in fixed-width sliding windows $W_i, 1 \leq i \leq I$ (frames) of 2.56 sec and 50% overlap. The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cut-off frequency was used. From each frame, a vector of feature $s V_i \in R^k, k = |F_T| + |F_F|$ was obtained by calculating variables from the time $F_T^i \in R^{|F_T|}$ and frequency domain $F_F^i \in R^{|F_F|}$.

The extracted time domain and frequency domain features are concatenated to a single feature vector as a representative signature for each frame. Details on the type of extracted features are provided in the following section. All frames are used as input to FRAILSAFE's human motion identification module which classifies basic activities of daily living (ADLs) in order to obtain some preliminary evaluation results for the proposed architecture. In this module, a model for multiclass classification between six basic ADLs (walking, walking-upstairs, walking-downstairs, sitting, standing and laying), which has been previously built in a training phase, is used in order to label the frames. Each frame is classified independently.

During the training phase of the classification architecture, frames with known class labels (labeled manually) are used to train the multiclass classification model. During the test phase, the unknown multi-parametric sensor signals are pre-processed and parameterized with similar setup as in the training phase. Each extracted feature vector is provided as input to the trained classifier.

Feature Extraction and Classification

Initially, the sensor signals (accelerometer and gyroscope) were pre-processed as proposed in (Anguita et al., 2012, Reyes-Ortiz et al., 2013) in order to proceed with feature extraction. The features selected for this analysis are those proposed in (Anguita et al., 2012, Reyes-Ortiz et al., 2013) which come from the accelerometer and gyroscope 3-axial raw signals denoted as tAcc-XYZ and tGyro-XYZ with prefix 't' used to denote time. The sampling frequency of these time domain signals was 50 Hz. In order to remove noise Anguita et al. performed low pass filtering using a median filter and a 3rd order low pass Butterworth filter with a corner frequency of 20 Hz. Then, in order to separate the acceleration signal into body and gravity acceleration signals denoted as tBodyAcc-XYZ and tGravityAcc-XYZ, they used another low pass Butterworth filter with a corner frequency of 0.3 Hz.

Subsequently, Jerk signals denoted as tBodyAccJerk-XYZ and tBodyGyroJerk-XYZ were obtained by the time derivation of the body linear acceleration and angular velocity. Also they used the Euclidean norm to calculate the magnitude of these three-dimensional signals yielding the following signals: tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag and tBodyGyroJerkMag.

Finally, a Fast Fourier Transform (FFT) was applied to signals tBodyAcc-XYZ, tBodyAccJerk-XYZ, tBodyGyro-XYZ, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag producing fBodyAcc-XYZ, fBodyAccJerk-XYZ, fBodyGyro-XYZ, fBodyAccJerkMag, fBodyGyroMag, fBodyGyroJerkMag. Here, the prefix 'f' was used to indicate frequency domain signals.

These signals were used to estimate variables of the feature vector for each pattern: '-XYZ' is used to denote 3-axial signals in the X, Y and Z directions. The aforementioned signals which were produced by processing accordingly the initial sensor recordings are tabulated in Table 7.

Table 7: Pre-processed Signals

Signals
tBodyAcc-XYZ
tGravityAcc-XYZ
tBodyAccJerk-XYZ
tBodyGyro-XYZ
tBodyGyroJerk-XYZ

tBodyAccMag
tGravityAccMag
tBodyAccJerkMag
tBodyGyroMag
tBodyGyroJerkMag
fBodyAcc-XYZ
fBodyAccJerk-XYZ
fBodyGyro-XYZ
fBodyAccMag
fBodyAccJerkMag
fBodyGyroMag
fBodyGyroJerkMag

The set of features that were extracted from these signals are those proposed by Anguita et al. including the mean value, the standard deviation, the median absolute deviation, the largest value in array, the smallest value in array, the signal magnitude area, the energy measure as the sum of the squares divided by the number of values, the interquartile range, the signal entropy, the autoregression coefficients with Burg order equal to 4, the correlation coefficient between two signals, the index of the frequency component with largest magnitude, the weighted average of the frequency components to obtain a mean frequency, the skewness of the frequency domain signal, the kurtosis of the frequency domain signal, the energy of a frequency interval within the 64 bins of the FFT of each window and the angle between two vectors.

Additional vectors were obtained by averaging the signals in a signal window sample. These are used on the angle variable (Table 8).

Table 8: Additional Signals

Additional Sigansls
gravityMean
tBodyAccMean
tBodyAccJerkMean
tBodyGyroMean
tBodyGyroJerkMean

In conclusion, for each record a 561- feature vector with the aforementioned time and frequency domain variables was provided.

The computed feature vectors were used to train a classification model. In order to evaluate the ability of the above features to discriminate between ADLs we examined the SMO (Keerthi et al., 2001; Platt et al., 1998) with RBF kernel classification algorithm, which was implemented by the WEKA machine learning toolkit (Hall et al. 2009). SMO algorithm is an implementation of Support Vector Machines provided by the WEKA toolkit. Here we selected SMO for the classification since SVMs are used mostly in relevant literature.

During the test phase, the sensor signals were pre-processed and parameterized as during training. The SMO classification model was used to label each of the activities. Evaluation was performed in a subject dependent cross-validation setting.

In a further step, we examined the discriminative ability of the extracted features for the human motion identification. The ReliefF algorithm (Kononko, 1994) (which is an extension of an earlier algorithm called Relief (Kira and Rendell, 1992)) was used for estimating the importance of each feature in multiclass classification. In the ReliefF algorithm the weight of any given feature decreases if the squared Euclidean distance of that feature to nearby instances of the same class is more than the distance to nearby instances of the other class. ReliefF is considered one of the most successful feature ranking algorithms due to its simplicity and effectiveness (Dietterich, 1997; Sun and Li, 2006; Sun and Wu, 2008) (only linear time in the number of given features and training samples is required), noise tolerance and robustness in detecting relevant features effectively, even when these features are highly dependent on other features (Dietterich, 1997; Kononko, 1997). Furthermore, ReliefF avoids any exhaustive or heuristic search compared with conventional wrapper methods and usually performs better compared to filter methods due to the performance feedback of a nonlinear classifier when searching for useful features (Sun and Wu, 2008).

In this work, ranking is performed on the whole dataset including all frames from all subjects. We examined the performance of the method, in terms of accuracy for different number of N-best features (N =10, 20, 30, ... 560), with respect to the ReliefF feature ranking algorithm.

Classification Results

Test dataset: The previously described classification methodology was evaluated on multi-parametric data from the UCI HAR Dataset (Anguita et al., 2013). The dataset consists of accelerometer and gyroscope recordings from 30 volunteers within an age bracket of 19-48 years when performing six activities (walking, walking-upstairs, walking-downstairs sitting, standing, laying). For the experiments, each person wore a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz were captured. The data were labelled manually using the corresponding video recordings which were captured during the experiments. Since the evaluation here was held using a subject dependent cross-validation setting,

data were initially merged in a single dataset and then split in 30 datasets, one for each subject.

Results

The classification methodology presented was evaluated using the classification algorithm and the cross-validation scheme described in Section 4. The classification performance was evaluated in terms of accuracy

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

where TP denotes the true positives, TN the true negatives, FP the false positives and FN the false negatives. The results of the method using all features are shown on Table 9.

As can be seen in Table 9, the overall highest accuracy of the proposed methodology for human motion identification is 100% for subjects 1 and 29 and the lowest accuracy 85.76% was obtained for the 9th subject. However, the mean accuracy is relatively high, 95,84%. The mean (across all subjects) confusion matrix is shown in Table 10. As can be seen, all ADLs except sitting and standing are nearly perfectly discriminated from the others with only a few false dismissals or false alarms (2 to 5). The misclassification of some sitting and standing instances are probably owed to the similarity of these ADLs.

Table 9: Subject Dependent Human Motion Identification Accuracy

Subject	Accuracy
1	100%
2	93,71%
3	97,36%
4	93,69%
5	86,09%
6	92,92%
7	93,83%
8	98,93%
9	85,76%
10	95,92%
11	98,42%
12	97,50%
13	95,41%
14	95,67%
15	96,65%

16	94,81%
17	90,76%
18	96,70%
17	98,33%
20	96,33%
21	98,53%
22	99,38%
23	96,77%
24	98,95%
25	94,87%
26	96,94%
27	99,73%
28	93,72%
29	100,00%
30	97,65%

Table 10: Mean across subjects confusion matrix. Rows represent the actual class and columns the predicted class.

	Standing	Sitting	Laying	Walking	Downstairs	Upstairs
Standing	1795	110	0	0	0	1
Sitting	289	1485	2	0	0	1
Laying	0	0	1944	0	0	0
Walking	0	0	0	1718	2	2
Downstairs	0	0	0	5	1397	4
Upstairs	0	0	0	0	0	1544

In a further step, we applied feature ranking on the whole dataset (consisting of all available subjects) using the ReliefF algorithm as described earlier. The performance of the classification, in terms of accuracy, for different number of N-best features (N =10, 20, 30 ..., 560) for the SMO algorithm is shown in Figure 7. As can be seen in this figure, the highest classification accuracy is achieved when a large subset of discriminative features is used. Specifically, the highest accuracy is achieved for a subset of 550 best features with a percentage of 97% which is equal to the accuracy achieved when all features are used. It seems that the size and the variability of the dataset is relatively large requiring a feature vector of high dimensionality to accurately discriminate between the six classes. However, with only 40 features a high accuracy equal to 90% can be achieved.

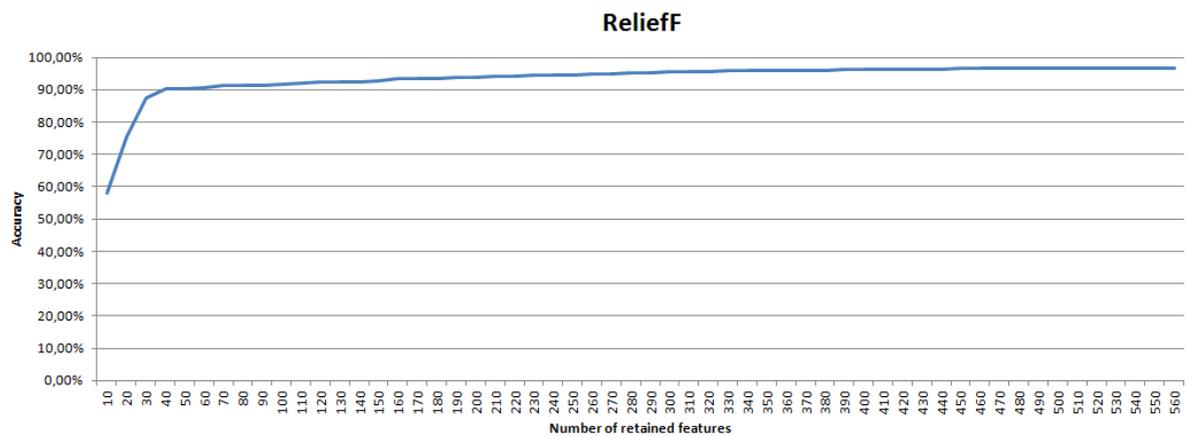


Figure 7: Classification Accuracy for different subsets of N-best features (N=10,20,..., 550).

Table 11 shows the 40 best features according to the ReliefF ranking algorithm. Although it is best to use a high dimensional feature vector to achieve higher classification accuracy, feature selection still is important in cases where a light human motion identification module is needed such as in FrailSafe.

Table 11: ReliefF Feature Ranking

Ranking	Feature
1	tGravityAcc_energy_X
2	fBodyAccJerk_entropy_X
3	fBodyAcc_entropy_X
4	fBodyAccJerk_entropy_Y
5	tBodyAccJerkMag_entropy
6	angle(X_gravityMean)
7	tGravityAcc_min_X
8	tGravityAcc_mean_X
9	tBodyAccJerk_entropy_X
10	tGravityAcc_max_X
11	fBodyBodyAccJerkMag_entropy
12	tBodyAcc_max_X
13	tBodyAccJerk_entropy_Y
14	fBodyAccMag_entropy
15	fBodyAcc_entropy_Y
16	fBodyAccJerk_entropy_Z
17	tBodyAccJerk_entropy_Z
18	tBodyGyroJerkMag_entropy

17	tGravityAcc_energy_Y
20	tBodyAccMag_entropy
21	tGravityAccMag_entropy
22	tGravityAcc_mean_Y
23	tBodyGyroJerk_entropy_Z
24	tGravityAcc_max_Y
25	fBodyAcc_entropy_Z
26	tGravityAcc_entropy_Z
27	tGravityAcc_min_Y
28	fBodyGyro_entropy_X
29	fBodyGyro_entropy_X
30	tBodyGyroJerk_entropy_X
31	fBodyAcc_mad_X
32	fBodyAcc_std_X
33	tBodyAcc_std_Y
34	fBodyAcc_mad_Y
35	tBodyAcc_std_X
36	fBodyBodyGyroJerkMag_entropy
37	tBodyAcc_mad_Y
38	fBodyGyro_entropy_Y
39	tBodyGyroMag_entropy
40	fBodyAcc_std_Y

Although direct comparison with other studies performed on the same dataset is not feasible due to different problem identification (subject dependent classification studied here versus subject independent classification studied on previous works) and different validation protocols followed as well (cross validation used here instead of 70% and 30% train and test sets respectively on the literature), the proposed method for the subject independent classification slightly improves the classification accuracy from 94% and 96% achieved in (Reiss et al., 2013) and (Romera-Paredes et al., 2013, Kastner et al., 2013) respectively to 97%. This improvement is significant since it can be achieved with fewer features, providing the means for lighter human motion identification.

4.3.2 Towards prediction of frailty

Extraction of predefined features

Using the Activity Classification algorithm which we developed, we are able to analyze our stored data and annotate them accordingly. Then using this annotation

we are able to provide summaries of physiological parameters of participants towards the clinicians. Currently we are providing daily summaries of average/min/max values for Heart Rate and Respiration Rate for each participant during these activities: Sitting/standing, Lying, Walking, Walking Upstairs, Walking Downstairs.

Deep learning for feature extraction and prediction

In our future work, we plan to investigate deep learning techniques for an in-depth analysis of the time series data and for the seamless extraction of a features' hierarchy that will be linked through a deep neural network to a frailty index. The prediction model will be used to provide a frailty indicator during the recordings without the need for a thorough clinical evaluation. Also we will examine the change of this frailty indicator over the evaluation period and assess its temporal consistency. The change of frailty indicators calculated from clinical data has shown to be inconsistent over time manifested as unexpected frailty recession (e.g. transition from frail to non-frail) in a large percentage of subjects (Xue2011). Our hypothesis is that non-subjective reproducible measurements from sensor data are unbiased and could lead to more stable and reliable markers.

Finally we will correlate our proposed frailty index with measurements from upcoming clinical evaluations and see whether a current index has any predictive ability for near-coming events and changes in the participants' health status, which are not observed by the clinical scores.

Proposed method

Previously, existing methods for classification or modeling generally relied on the usage of domain specific features normally selected by human experts. Finding the best features was the subject of a lot of research and the performance of the classifier was heavily dependent on their quality. The advantage of deep neural networks and especially convolutional neural networks (CNNs) is that they can learn such features by themselves, reducing the need for human experts. A recent review on deep learning techniques for time series analysis can be found in (Gamboa2017). Building upon our previous work on deep CNNs (Zacharaki2017), we aim to implement a deep architecture that employs convolution and pooling operations to capture the salient patterns of the multi-channel time series data at different time scales. The architecture will be similar to the one proposed by Yang et al. (Yang2015), which showed very competitive performance. In this architecture, the convolution and pooling filters in the CNN are applied along the temporal dimension for each sensor, and all the feature maps for different sensors are unified as a common input for the neural network classifier. A main difference however in this approach and our problem is that in the former classification (e.g. for activity) is performed for each temporal window, whereas in our case a single decision (prediction of frailty) has to be made for the whole recording. Thus, a fusion operation is required at the end of the architecture. An intrinsic challenge lies in the

non-uniformity of the data in respect to duration, activities performed during the recordings, as well as the low quality of some signals due to their acquisition in a real-life home environment and not in a controlled experimental setting.

References

- (Anguita2013) Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L., 2013. Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine. IWAAL 2012 International Workshop of Ambient Assisted Living.
- (Dietterich1997) Dietterich, T. G., 1997. Machine learning research: Four current directions. *Artificial Intelligence Magazines*, vol. 18, pp. 97–136.
- (Gaines2016) Gaines, Brian R., and Hua Zhou. "Algorithms for Fitting the Constrained Lasso." arXiv preprint arXiv:1611.01511 (2016).
- (Gamboa2017) Gamboa, John Cristian Borges. "Deep Learning for Time-Series Analysis." arXiv preprint arXiv:1701.01887 (2017).
- (Hall2009) Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann P., Witten, I. H., 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, vol. 11.
- (Keerthi2001) Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., Murthy, K.R.K., 2001. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, vol. 13, pp. 637-649
- (Kira1992) Kira, K., Rendell, L. A., 1992. A practical approach to feature selection. *Proc. 9th Int. Conf. Mach. Learn.*, pp. 249 – 256.
- (Kononenko1994) Kononenko, I., 1994. Estimating attributes: Analysis and extension of RELIEF. *Proc. Euro. Conf. Mach. Learn.*, vol. 784, pp. 171– 182.
- (Kononenko1997) Kononenko, I., Simec, E., Robnik-Sikonja, M., 1997. Overcoming the Myopic of Inductive Learning Algorithms with RELIEF-F. *Applied Intelligence*
- (Platt1998) Platt, J., 1998. Fast Training of Support Vector Machines using Sequential Minimal Optimization. *Advances in Kernel Methods - Support Vector Learning*.
- (Reyes-Ortiz2013) Reyes-Ortiz, J.L., Ghio, A., Parra, X., Anguita, D., Cabestany, J., Catala, A., 2013. Human Activity and Motion Disorder Recognition: Towards Smarter Interactive Cognitive Environments. *ESANN 2013 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.
- (Sun2006) Sun, Y., Li, J. 2006. Iterative RELIEF for feature weighting. *Proc. 21st Int. Conf. Mach. Learn.*, pp. 913–920.
- (Sun2008) Sun, Y., Wu, D., 2008. A RELIEF based feature extraction algorithm. In *Proceedings of SIAM International Conference on Data Mining*.
- (Xue2011) Xue, Qian-Li. "The frailty syndrome: definition and natural history." *Clinics in geriatric medicine* 27.1 (2011): 1-15.
- (Yang2015) Yang, Jianbo, et al. "Deep convolutional neural networks on multichannel time series for human activity recognition." *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015.

(Zacharaki2017) Zacharaki, Evangelia I. "Prediction of protein function using a deep convolutional neural network ensemble." PeerJ Preprints 5 (2017): e2778v1.