



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957047.

H2020-LC-SC3-EE-2020-1/LC-SC3-B4E-6-2020

Big data for buildings



Building Information aGGregation, harmonization and analytics platform

Project N° 957047

D5.1 - Initial Description of the BIGG Artificial intelligence toolbox

Responsible:	HELEXIA
Document Reference:	D5.1
Dissemination Level:	Public
Version:	1.0
Date:	07/03/2022

Contributors Table

DOCUMENT SECTION	AUTHOR(S)	CONTRIBUTOR(S)	REVIEWER(S)
Initialization of the document	Pierre Lehanneur (Helexia) Mohamad Hammoud (Helexia)		Eric Pascual, Nicolas Pastorelly (CSTB), Frédéric Wauters (Energis), Vincent Bracke (UGent), Stéphane Leroy (Helexia)
Introduction	Pierre Lehanneur (Helexia) Mohamad Hammoud (Helexia)	Nicolas Pastorelly (CSTB) through information from DS2.2	Eric Pascual, Nicolas Pastorelly (CSTB), Frédéric Wauters (Energis), Vincent Bracke (UGent), Stéphane Leroy (Helexia)
Work Package Tasks	Pierre Lehanneur (Helexia) Mohamad Hammoud (Helexia)	Riccardo Devivo (Energis), Romain Hollanders (Energis)	Eric Pascual, Nicolas Pastorelly (CSTB), Frédéric Wauters (Energis), Vincent Bracke (UGent), Stéphane Leroy (Helexia)
AITB Development	Pierre Lehanneur (Helexia) Mohamad Hammoud (Helexia)	Romain Hollanders (Energis)	Eric Pascual, Nicolas Pastorelly (CSTB), Frédéric Wauters (Energis), Vincent Bracke (UGent), Stéphane Leroy (Helexia)
Business Understanding	Pierre Lehanneur (Helexia) Mohamad Hammoud (Helexia)	Romain Hollanders (Energis), Gerard Mor (CIMNE), Thea Gutmatcher (Inetum), Manu Lahariya (IMEC)	Eric Pascual, Nicolas Pastorelly (CSTB), Frédéric Wauters (Energis), Vincent Bracke (UGent), Stéphane Leroy (Helexia)
Pipeline flowcharts	Pierre Lehanneur (Helexia) Mohamad Hammoud (Helexia)	Romain Hollanders (Energis), Riccardo Devivo (Energis), Gerard Mor (CIMNE), Thea Gutmatcher (Inetum), David Bouret (Inetum) Manu Lahariya (IMEC)	Eric Pascual, Nicolas Pastorelly (CSTB), Frédéric Wauters (Energis), Vincent Bracke (UGent), Stéphane Leroy (Helexia)

Table of contents

I. INTRODUCTION.....	7
I.1. Purpose and structure of the document.....	7
I.2. Interaction with other work packages.....	7
I.3. Scope and audience	9
II. WORK PACKAGES TASKS	10
II.1. Task 5.1 - Provision of data storage infrastructure.....	10
II.2. Task 5.2 - Data analytics tools design and identification of commonalities	10
II.3. Task 5.3 - AI/ML techniques.....	10
II.4. Task 5.4 - AI/ML based service modules	10
II.5. Proposed WP5 approach	11
III. AITB DEVELOPMENT METHODOLOGY	12
III.1. Introduction	12
III.2. Data storage	13
III.3. Business Understanding.....	13
III.3.1. BC 1	14
III.3.2. BC2	15
III.3.3. BC 3	16
III.3.4. BC 4	17
III.3.5. BC 5	18
III.3.6. BC 6:	19
III.4. Pipeline flowcharts description.....	20
III.4.1. Flowchart semantics	20
III.4.2. BC1	20
III.4.3. BC2	25
III.4.4. BC4	27
III.4.5. BC5	28
III.4.6. BC6-UC14.....	30
III.4.7. BC6-UC15.....	31
III.5. Function Blocks and commonalities identification.....	32
III.6. Identified Existing libraries	33
III.6.1. Python libraries	33
III.6.2. R libraries	33
III.7. Consolidated list of Function Blocks.....	35
III.8. Collaborative work management and tools	42
III.8.1. Collaborative code development tool.....	42
IV. PRELIMINARY VERSION OF THE AI TOOLBOX.....	44

IV.1. Data collection and data format	44
IV.2. Data storage	44
IV.3. List of Function Blocks	45
IV.3.1. Data preparation	45
IV.3.2. Data transformation	47
IV.3.3. Modelling	49
IV.3.4. Reinforcement learning techniques	50
IV.4. Code development methodology	51
IV.5. Test and verification process	52
V. STATUS OF IMPLEMENTATION	53
CONCLUSION	54

Table of Figures

Figure 1 – BIGG WP interaction	8
Figure 2 – General workflow – BIGG WP responsibilities	8
Figure 3 - Cross Industry Standard Process for Data Mining	11
Figure 4 - Initial development methodology	13
Figure 5 - BIGG BC1 – Business Understanding	14
Figure 6 – BIGG BC2 – Business Understanding	15
Figure 7 – BIGG BC3 – Business Understanding	16
Figure 8 – BIGG BC4 – Business Understanding	17
Figure 9 – BIGG BC5 – Business Understanding	18
Figure 10 – BIGG BC6 – Business Understanding	19
Figure 11 - Pipeline Flowchart - BC1 – Longitudinal Benchmarking	22
Figure 12 - Pipeline Flowchart - BC1 – Cross Sectional Benchmarking	23
Figure 13 – Pipeline Flowchart - BC1 – ECM Results Benchmarking	24
Figure 14 – Pipeline Flowchart - BC2 - EPC Characterization	26
Figure 15 – Pipeline Flowchart - BC4 - EPCo Management Facilitation	28
Figure 16 - Pipeline Flowchart - BC5 – Comfort Case	29
Figure 17 – Pipeline Flowchart - BC6 – Electrical Flexibility	30
Figure 18 – Pipeline Flowchart - BC6 – Gas Flexibility	31
Figure 19 – List of Function Blocks to be developed	41

Table of Acronyms and Definitions

Acronym	Definition
AI	Artificial Intelligence
AITB	BIGG Artificial Intelligence Toolbox
API	Application Programming Interface
BC	Business Case
BDHF	BIGG Harmonized Format (BHF)
BMS	Buildings Management System (BMS)
CVRMSE	Coefficient of Variation of the root mean square error, CV(RMSE). This basically assess how close you are to the individual data points (such as monthly utility bills)
DR	Demand Response (DR)
DSF	Demand Side Flexibility
EEM (=ECM)	Energy Efficiency Measure (=Energy Conservation Measure)
EPC	Energy Performance Certificate .
EPCo	Energy Performance Contract
ESCO	Energy Service Company
Function	Function in this document refers to a needed operation, transformation, analytic task or modification to be performed on a given set of data.
Function Block	Function block is used in the document to describe a single set of code developed to perform a specific Function identified by the BIGG WP5 team as a singular element. A Function Block is defined by its inputs, its Function and the output it provides.
HVAC	Heating Ventilation and Air Conditioning
INSPIRE	<p>The INSPIRE Directive, establishing an infrastructure for spatial information in Europe to support Community environmental policies, and policies or activities which may have an impact on the environment entered into force in May 2007.</p> <p>INSPIRE is based on the infrastructures for spatial information established and operated by the Member States of the European Union. The Directive addresses 34 spatial data themes needed for environmental applications. See https://inspire.ec.europa.eu/</p>
NMBE	Normalized Mean Bias Error. This assess whether you globally over or under-predict the consumption
Pipeline	A pipeline is generally defined as a linear sequence of processes chained together to perform an instruction. In this document Pipeline refers to one or several Function Blocks assembled and packaged together to perform a larger task. It can be seen as a Function Block made out of other Function Blocks.
R²	R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent

	variable or variables in a regression model. Whereas correlation explains the strength of the relationship between an independent and dependent variable, R-squared explains to what extent the variance of one variable explains the variance of the second variable.
RAF	Reference Architecture Framework
RES	Renewable Energy Source
UC	Use Case. In this document, the various use cases mentioned are taken from the D6.1 and detailed according to a chosen formalism.

I. INTRODUCTION

I.1. Purpose and structure of the document

The objectives of the Work Package 5 are to research, realize, and validate innovative AI-based methods and decision support tools for the analysis of the high quality, anonymized, interoperable building related data collected in the project.

The main objectives are defined by the WP tasks and aim at developing Data analytics tools such as:

- Methods for extracting discriminative data features and validation methodologies for AI-based methods applicable in the project.
- Classification methods by exploiting additional information achieved from public datasets and unlabelled data and possibly obtain explainable results.
- Benchmarking protocols for validating the realized AI-based technologies in cooperation with trial partners.

And AI/ML Techniques such as:

- Regression techniques (linear, polynomial, logistic, random forest, ...)
- Classification techniques (Linear, Nearest Neighbour, Support Vector Machines, Decision Trees, Random Forest, Neural Networks, ...)
- Reinforcement learning Techniques with which the software interacts with a dynamic environment in which it must perform a certain goal. Feedback is provided in terms of rewards as it navigates its problem space.
- Closed Loop Model Predictive control to be used when control strategies are needed.

The AI toolbox (AITB) is the set of computer science tools that is being developed to serve the needs of the different use cases in terms of data analytics and AI/ML techniques..

The purpose of this document is to present the preliminary version of the BIGG Artificial Intelligence (AI) Toolbox.

I.2. Interaction with other work packages

While WP5 focuses on AI techniques, its responsibilities are strongly connected with other technical WP within the project and specifically with WP3 and WP4. The figure below presents the WP5 in the context of the BIGG responsibility matrix.

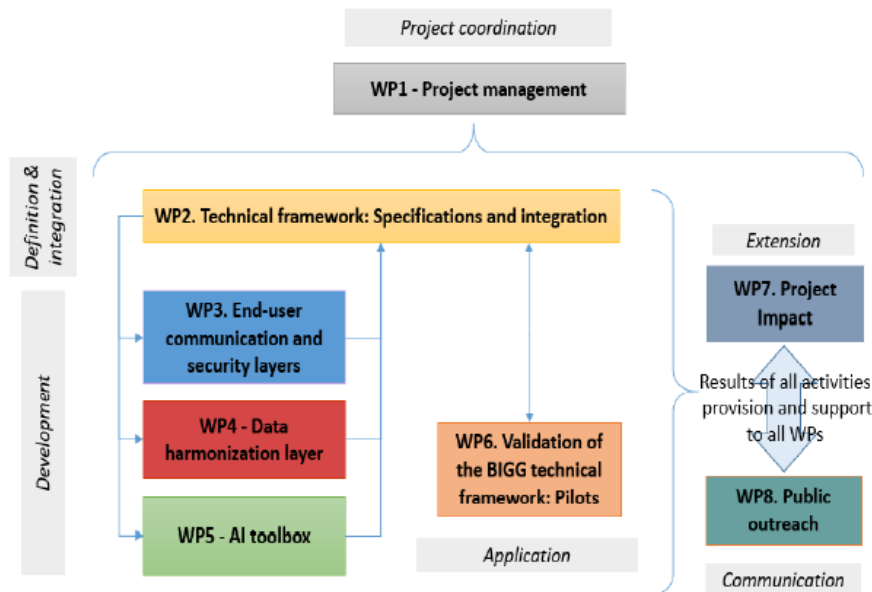


Figure 1 – BIGG WP interaction

This representation is generic and a more specific representation of the interaction between Work Packages was presented in the Reference Architecture Framework designed in deliverable D2.2. The figure below presents it from an implementation standpoint, starting with data collection and going step by step to the final representation of the analytics being performed on the data:

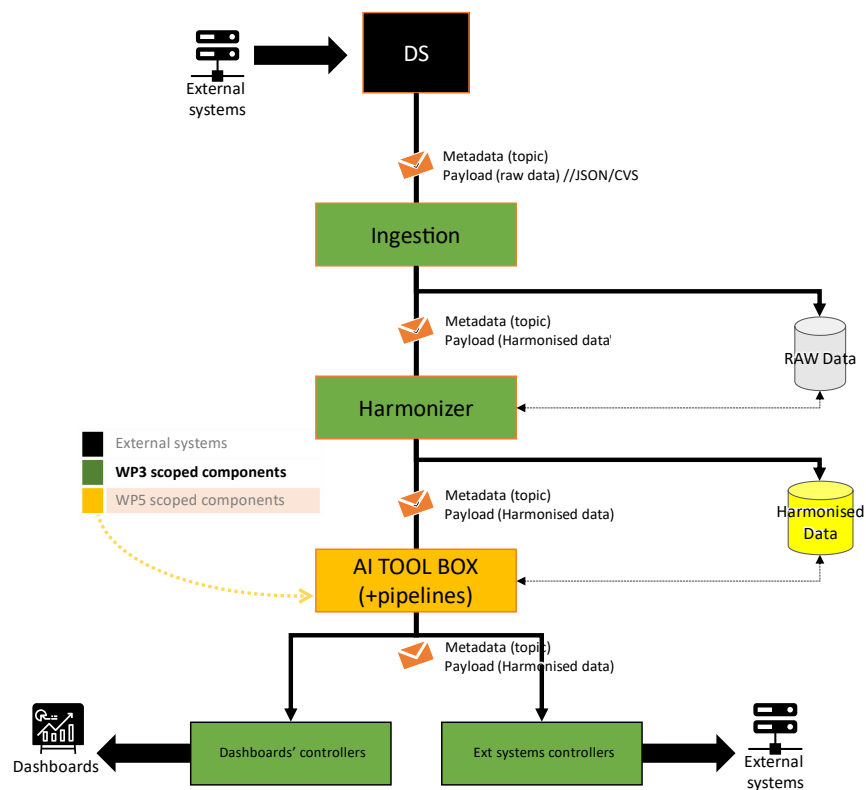


Figure 2 – General workflow – BIGG WP responsibilities

On this schematic, we can identify the responsibility of the WP5 components and how they connect with other components of the BIGG project. This document will focus on these specific WP5 components identified on the orange block stack above and present them in details.

I.3. Scope and audience

This document describes what is developed in response to an identification from the business owners of the necessary techniques and tools to solve the analytical challenges faced in their different use cases. This deliverable presents the status after a year in the BIGG project. It will constitute the base of work for the next deliverable of the work package 5 (D5.2: Description of the final version of the AI toolbox) but also in other work packages where interactions between the AITB and other dedicated services are expected with respect to data acquisition, data storage or data harmonization.

The audience for this document is composed of the project partners that want to get an overview of the current progress of the AITB and could want to implement early stage components for their own use cases and the reviewers who need to get a clear understanding of the purpose and achievements of WP5 work to this point.

II. WORK PACKAGES TASKS

The work package 5 distinguishes 4 main tasks to be performed that are geared toward the final design of the AITB:

II.1. Task 5.1 - Provision of data storage infrastructure

This task will deliver the storage framework for the operation and data collection, covering real-time, batch streaming and long-term data processing.

The initial approach to data storage is described in section [III.2. Data storage](#)

II.2. Task 5.2 - Data analytics tools design and identification of commonalities

This task is expected to deliver all the necessary data analytics modules and to support the pre- and post-analysis of data and models, including data quality, detection of trends, model validation, etc. It will be composed of:

- Classical statistical indicators such as P-values, correlations, error criteria, etc;
- Supervised learning techniques such as classification and regression and non-supervised learning such as clustering;
- Graphical tools and others.

II.3. Task 5.3 - AI/ML techniques

Task 5.3 aims at identifying which AI/ML techniques are most relevant to reach the most appropriate solution for each business cases of WP6. Examples of such AI/ML techniques are:

- Regression techniques (linear, polynomial, logistic, random forest, ...)
- Classification techniques (Linear, Nearest Neighbour, Support Vector Machines, Decision Trees, Random Forest, Neural Networks, ...)
- With Reinforcement learning the software interacts with a dynamic environment in which it must perform a certain goal. Feedback is provided in terms of rewards as it navigates its problem space.
- Closed Loop Model Predictive control will be used when control strategies are needed.

II.4. Task 5.4 - AI/ML based service modules

The goal of Task 5.4 is to assemble Service modules combining the work done in T5.1 data sets, T5.2 analytics and T5.3 AI/ML techniques per business use case (WP6) and will be delivered to be executed in the context of WP6

II.5. Proposed WP5 approach

This tasks definition follows closely the general development method for data mining as presented below:

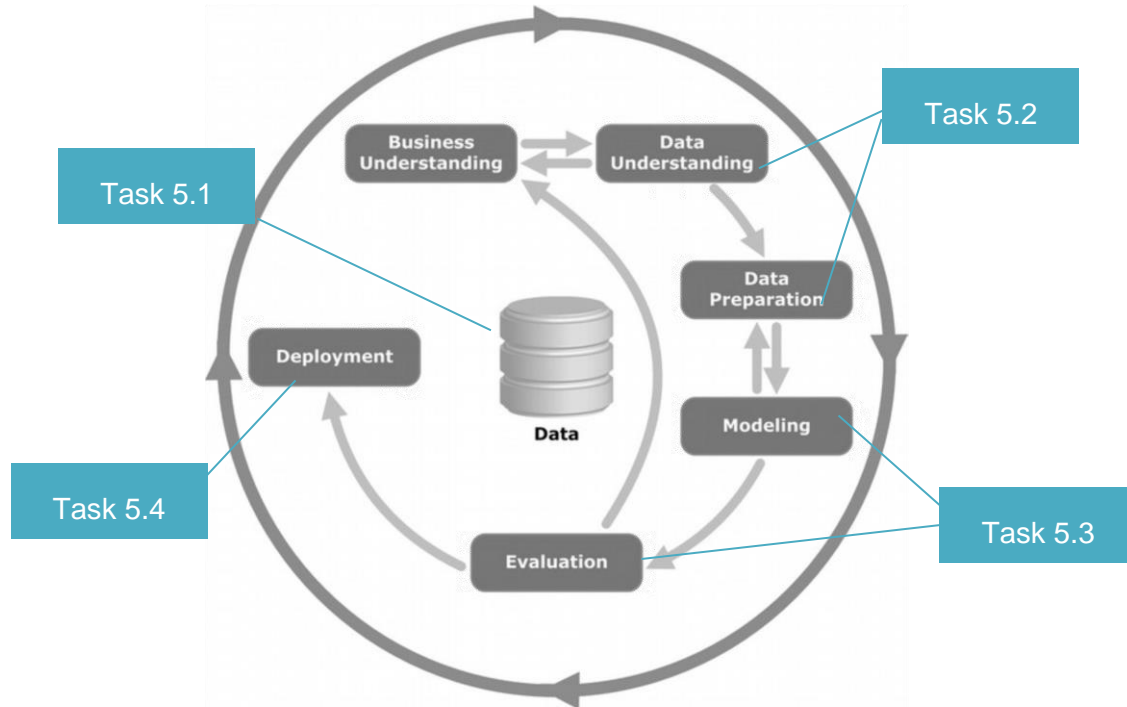


Figure 3 - Cross Industry Standard Process for Data Mining

The WP5 team members chose to follow this decomposition of work. The followed steps are presented in the following section. The methodology adopted to identify needs both in terms of data analytics (Task 5.2) and AI/ML techniques (Task 5.3) was the same. It consisted in the definition and the decomposition of each BIGG Business Case in a granular enough definition to isolate what is referred to in this document as Function Blocks.

III. AITB DEVELOPMENT METHODOLOGY

III.1. Introduction

The following steps were taken to deliver a preliminary version of the AI toolbox that fulfils the needs of all the BIGG business cases (BCs).

The WP5 team adopted a bottom-up approach beginning with a thorough business understanding description and translating it into an identification of necessary analytic needs. The output of that initial step was a comprehensive definition of the business cases from a Function perspective. These statements are presented below in the document section [Business Understanding](#). The goal of that initial step was twofold:

- Align all members of the WP on the expected results of each use case
- Identify which UC would need AI/ML techniques.

The WP5 team then created Flowcharts to illustrate step by step what the underlying challenges were for each business cases and all the individual steps needed to overcome these challenges. These flowcharts are presented in this document in section [Pipeline flowcharts description](#). The main objective of the flowchart is to identify for each single step, an **input**, a **Function** and an **output** which all together describe the **Function Blocks** needed to be developed.

These two preliminary steps allowed to consolidate a list of Function Blocks defined on BC basis. We then identified commonalities that appeared across business cases to minimize the amount of development work needed. All the BIGG business cases are related to building energy efficiency and it results common needs for data preparation, analytics or modelling across business cases. Identifying these commonalities and sorting which items needed to be merged or on the contrary were meant to remain distinct was important. . This step has also been critical to define the best granularity to keep the balance between ease of use and development efficiency.

After the final list of Function Blocks involved was consolidated, an additional screening step was performed to detect items already developed in open-source libraries. One of the main impediments to the use of AI techniques applied to energy efficiency is the difficulty to comprehend the current state of the art and identify which existing libraries can be used as-is or with minor modifications. The result of this step was a list of Function Blocks where items existing in open libraries and items to be developed were identified.

The preliminary version of the AITB is intended to present this library of Function Blocks identified and developed separately to be used in the context of WP6.

From the preliminary AI toolbox created, Function Blocks are to be assembled and packaged together into Pipelines (Task 5.4) The final version of the AITB is intended to enable the use of Pipelines without additional development. To get a viable final product, it will be necessary to give the final user the possibility to use both granular Function Blocks individually, possibly combining them with other existing open source libraries, and pre-established pipelines developed for the specific use cases of the BIGG project .

The general approach to create the AI toolbox can be schematized as follows:

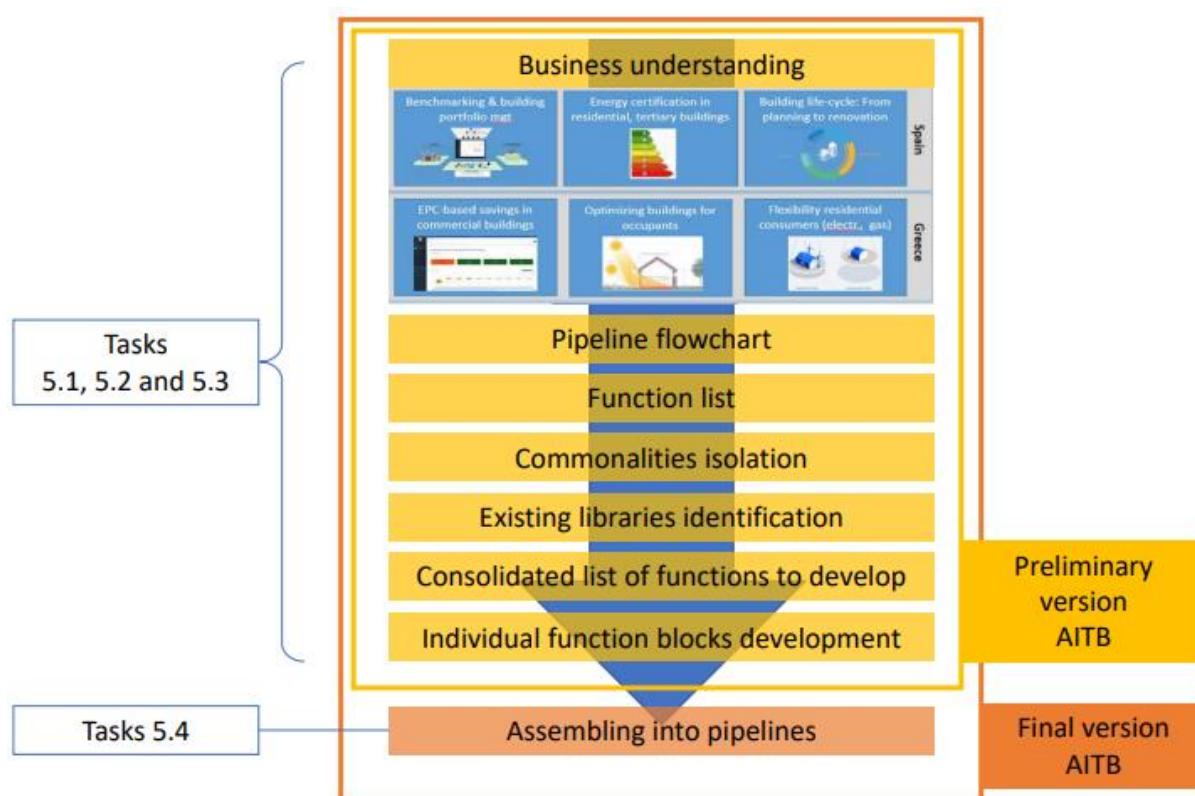


Figure 4 - Initial development methodology

III.2. Data storage

While data storage is a critical step in the data mining process presented above (section: [Work Packages Tasks](#)), it was agreed through discussions with the consortium that data storage could be managed through local storage capacities across the consortium.

All members of the WP5 have existing analytical services operating on real time data stored on their premises and leveraging these existing storage capacities presented a significant advantage in terms of data security management.

The specific storage implementation used in the context of AITB implementation is presented BC by BC in section [Data storage](#). The identification of the storage needs for each business case was performed in a similar way as the identification of analytical and AI/ML technique needs. Storage was identified in the flowcharts presented on section [Pipeline flowcharts description](#).

III.3. Business Understanding

A preliminary step needed to be taken to align all members of WP5 on their understanding of each business case. This work was performed in the early phase of the project and the results are presented below. The goal was to identify what are the expected results, whether or not the BC would the AITB and if so, what types of modules would be needed and finally identify what was the level of expertise within the WP team to help develop the necessary modules.

III.3.1. BC 1

Business Case		Use Case		Dataset	Business understanding	AI tasks (Knowledge Representation, Machine Learning, Reinforcement Learning)	Existing use and Knowledge
					please give your understanding of what this BC is trying to accomplish	please describe what you anticipate this business case will require as AI tools.	Consortium expertise
1	Benchmarking and energy efficiency tracking in public buildings	1	Benchmarking and monitoring of energy consumption	1,3,4,5,7	<p>Main stake holders: government or large organizations managing a large quantity of buildings</p> <p>Main features targeted:</p> <ul style="list-style-type: none"> - Building classification (clarify if part of WP5 or input from other WP) - Building benchmarking (Horizontal and vertical) - ECM classification (clarify if part of WP5 or input from other WP) - ECM benchmarking (Horizontal and vertical) - Consumption trends and alerts about anomalies - Assess weather dependency <p>Use:</p> <ul style="list-style-type: none"> - Create a dashboard presenting a large quantity of building according to their performances and identify which building needs renovation according to specific performance criteria. - Dashboard to display ECMs by type and quickly assess their impact on buildings - Display ECM impact by building type 	<p>Data preparation</p> <ul style="list-style-type: none"> - Time series consumption data - Time series weather data - Threshold building characteristics <p>Extraction from building and ECM data</p> <p>Vertical benchmarking (benchmark a building with himself in time)</p> <ul style="list-style-type: none"> - Vertical modelling (consumption segmentation, anomaly detection, consumption forecasting, consumption KPI's, ...) <p>[Penalized multi-regression models (PML)]</p> <ul style="list-style-type: none"> - Daily load curves Clustering vertical (detect occupancy profile patterns) - Normalization (daily consumption, z-norm, cleaning process) - Base line model using (weather data, calendar features, clustering vertical, occupancy patterns,) - ECM Estimation (Evaluate savings of ECM using Vertical modellings) <p>Horizontal benchmarking (benchmark a building with others)</p> <ul style="list-style-type: none"> - Building characteristics Clustering (detect similar buildings in terms of building characteristics or consumptions KPI's (vertical models)) - Classification modelling (to achieve the theoretical energy consumption of the reference building (extracted from similar buildings)) <p>Mixed outputs:</p> <ul style="list-style-type: none"> - ECM evaluation (some kind of recommendation based on building clustering, consumptions KPI's, ECM characteristics) 	Large knowledge within the consortium CIMNE ++ Energis ++ Intuicy ++
		2	Energy Conservation Measures (ECM) registration and evaluation	1,3,4,5,7			

Figure 5 - BIGG BC1 – Business Understanding

III.3.2. BC2

Business Case		Use Case		Data-set	Business understanding	AI tasks (Knowledge Representation, Machine Learning, Reinforcement Learning)	Existing use and Knowledge
					please give your understanding of what this BC is trying to accomplish	please describe what you anticipate this business case will require as AI tools.	Consortium expertise
2	Energy certification in residential and tertiary buildings	3	Integration of INSPIRE spatial data with Energy Performance Certification (EPC)	2, 7	<p>Mainly aims at harmonization of data across platform (INSPIRE / EPC / EU Level(s))</p> <p>Difficult to see an application in business context see if WP5 is taking care of this or if this is a WP4 task</p> <p>Main actor PA responsible for regional EPC: Business case focus on exploration of results of EPC's. This exploration of results will be: - Check of integrity of data, validation process</p>	<p>Data preparation</p> <ul style="list-style-type: none"> - Time series consumption data - Time series weather data - Threshold building characteristics <p>Energetic territorial characterization</p> <ul style="list-style-type: none"> - GIS analytics crossing Cadastral atom files with EPC-Open data Hub. - Correlation of EPCs, aggregated consumption data, cadastral and socioeconomic data to detect possible relationships between energy and location. - Vertical modelling (consumption segmentation, anomaly detection, consumption forecasting, consumption KPI's, ...) [Penalized multi-regression models (PML)] - Daily load curves Clustering vertical (detect occupancy profile patterns) - Normalization (daily consumption, z-norm, cleaning process) - Base line model using (weather data, calendar features, clustering vertical, occupancy patterns,) - Data transformation EPC Open data (and other data inputs) to EU Framework Level(S) - Extrapolation of EPC assessment for the building stock (possible output Energy Poverty detection,) 	CIMNE ++
		4	Adoption of the sustainability indicators of common EU framework Level(s) in building certification	2, 7			

Figure 6 – BIGG BC2 – Business Understanding

III.3.3. BC 3

Business Case		Use Case		Data-set	Business understanding	AI tasks (Knowledge Representation, Machine Learning, Reinforcement Learning)	Existing use and Knowledge
					please give your understanding of what this BC is trying to accomplish	please describe what you anticipate this business case will require as AI tools.	Consortium expertise
3	Building life-cycle - From planning to renovation	5	Interoperability between BIM, BMS, CMMS and building simulation engines	4,5,6	This BC focuses mainly on data interoperability.	Data preparation - Time series consumption data - Time series of sensors (IOT) data - Time series weather data - Threshold data inputs. Data standardization and data harmonization (data preparation for future AI process) Little need for analytics and AI Techniques	CIMNE ++
		6	Interoperability of BIGG with EEFIG-DEEP	1, 3			
		7	Interoperability between EU Building Stock Observatory (EUBSO) and national/regional Energy Performance Certification (EPC) hubs through BIGG	2, 7			

Figure 7 – BIGG BC3 – Business Understanding

III.3.4. BC 4

Business Case	Use Case	Data-set	Business understanding	AI tasks (Knowledge Representation, Machine Learning, Reinforcement Learning)	Existing use and Knowledge
			please give your understanding of what this BC is trying to accomplish and describe existing tools you can identified as useful	please describe what you anticipate this business case will require as AI tools.	Consortium expertise
4	Energy Performance Contract-based savings in commercial buildings: increase prediction accuracy	8	Assets management to store, view, update all relevant assets such as buildings, contracts, invoices, meters, sub-meters, sensors, equipment, ...	8	Facilitate the implementation, the management and performance tracking of an EPCo. Can be applied very easily to other cases such as local laws (ie French “Décret Tertiaire”).
		9	Actual savings tracking realised by the Energy Conservation Measures (ECMs) undertaken by the ESCO are monitored on a daily/weekly/monthly basis	8	
		10	Energy Performance Contract Management to manage the EPCo life cycle and perform actions (e.g. reporting) according to contractual milestones.	8	
				Time series data processing: - Align time series data with standard time grid - Detect and eliminate outliers Features extraction from time series data - Extract weekly consumption profiles out of consumption data - Detect time patterns such as seasonality Time series Regression modelling	Energis ++

Figure 8 – BIGG BC4 – Business Understanding

III.3.5. BC 5

Business Case		Use Case		Dataset	Business understanding	AI tasks (Knowledge Representation, Machine Learning, Reinforcement Learning)	Existing use and Knowledge
					please give your understanding of what this BC is trying to accomplish	please describe what you anticipate this business case will require as AI tools.	Consortium expertise
5	Optimizing buildings for occupants: Comfort case (commercial and residential)	11	Optimisation using weather forecasts will consider weather forecast 24 hours in advance	8	Main stakeholders: Building / Facility managers	Collect data (large amount) Real time data analysis, rule engine to check forecast VS current configuration 1. ML technique see effect of rules depending on conditions to find best operation sequence. (use existing weather forecast data, for occupancy, use sensor data and ML to model occupancy patterns) 2. Model the forecast reaction of the building	Energis ++ CIMNE: ++ developing an MPC control system to optimize heat pump operation in an international building Heating and cooling (tested under heating season) now working on the heating for MPC aspect (building in Germany) developed in R MPC is black box and already considers a pricing aspect. Eventually it can be adapted with restrictions. Model considers comfort restrictions as well. In effect control over the setpoint of the zone thermostats. Effect on the HP. Achieved 18% cost savings (not energy, idea is consume smarter). They use hourly changing price (gets the price of tomorrow at 12) IMEC: Building RL and Demand Response Main focus on DR --> BC6
		12	Optimisation using occupancy forecasts will add occupancy to the optimization logic	8	ESCOs End users / Building occupants Use forecast of weather,		
		13	Optimisation using price forecasts will add energy prices information on top of the weather and occupancy forecasts	8	occupancy and pricing to improve BMS operation		

Figure 9 – BIGG BC5 – Business Understanding

III.3.6. BC 6:

Business Case		Use Case		Dataset	Business understanding	AI tasks (Knowledge Representation, Machine Learning, Reinforcement Learning)	Existing use and Knowledge
					please give your understanding of what this BC is trying to accomplish	please describe what you anticipate this business case will require as AI tools.	Consortium expertise
6	Flexibility potential of residential consumers on electricity and natural gas	14	Electricity Demand Response	9, 10, 11, 12	Stakeholders: energy supplier (electricity and gas) Aggregator Sell flexibility to the suppliers	ML Techniques for flexibility analysis Data model training and testing Data model identification Parameter optimization Forecast generation	INETUM: ++ Already involved on project Interconnect with similar development work being done on electrical Flexibility IMEC: ++ (on electricity's side) working on the electrical side today DOMX: ++ (on gas) DOMX has already such mechanism in place
		15	Natural Gas Demand Response	9, 10, 11, 12	DOMX wants optimization to happen at the individual user's level. Not at the aggregate level Reward mechanism. Today, they use edge devices (provided by DOMX) the algorithm is run locally algo can live in a library, to be discussed.	RL algorithms to optimize energy (electricity/gas) consumption and minimize suppliers costs	

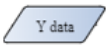

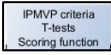

Figure 10 – BIGG BC6 – Business Understanding

III.4. Pipeline flowcharts description

In order to get better illustrations of the process, it was chosen to follow a similar approach as the one implemented in WP2 for the definition of the use cases. A flowchart representation of the business cases involving the AITB was created by the WP members where all necessary functions and methods were displayed. This with the goal of detecting all the commonalities across different use cases.

III.4.1. Flowchart semantics

We defined a common semantics in order to refer to similar concepts the same way across flow charts. Below is the adopted semantics for all flowcharts:

- Color code
 - Blue is related to objects
 - Green is related to actions
- Shapes semantics
 -  Blue diamonds are intermediate objects which are the input and/or output of an action
 -  Blue cylinders are used to designate stored collections of objects
 -  Blue rectangles are used for documentation
 -  Green rectangles are used for the actions

III.4.2. BC1

In BC1, multiple Pipelines were designed for the energy consumption benchmarking of buildings and estimation of Energy Conservation Measures (ECM).

Inputs:

The inputs needed for the computation of these Pipelines are the datasets that have been previously harmonised to the BIGG data model, initially coming from different sources. This process aims to link the raw data sources to a common model that provides clear relationships between building characteristics, locations, metering devices, and other energy-related information about buildings. In this business case, the needed input datasets are:

- Smart-metered electricity time series (at least, hourly frequency) gathered through Datadis, which is the Spanish DSOs data platform.
- Monthly gas or electricity consumption gathered through Gemweb, a company that stores the energy consumption of the Catalan government buildings during the last 4 years for energy auditing purposes.
- Building characteristics obtained from the Spanish Cadaster INSPIRE-harmonised datasets and the GPG repository, which is the Catalan government buildings catalog.

- Local weather data gathered through the Darksky online services. In the case of solar radiation, the information is gathered using the Copernicus Atmospheric Monitoring Service (CAMS).

Outputs:

- Weather and building-size normalized Key Performance Indicators (KPIs) related with energy use (e.g. kWh/m²) at different time spans
- Estimations of energy savings compared to previous periods

As explained above in this deliverable, this business case is divided in two use cases: Benchmarking and monitoring of energy consumption and ECMs registration and evaluation. The benchmarking use case is divided in two main objective:

- Firstly, to assess the energy consumption of a single building using its historical consumption and weather data as input (also known as longitudinal benchmarking). It essentially consists of disaggregating the whole consumption into three components: baseload, heating and cooling. With these components and static information of the building, several KPIs are estimated to assess the energy consumption over time.
- Secondly, an assessment of the energy consumption based on similar buildings is done. This is the so-called cross-sectional benchmarking in literature. Thus, this second Pipeline aims to model the energy-usage KPIs based on building characteristics and weather conditions.

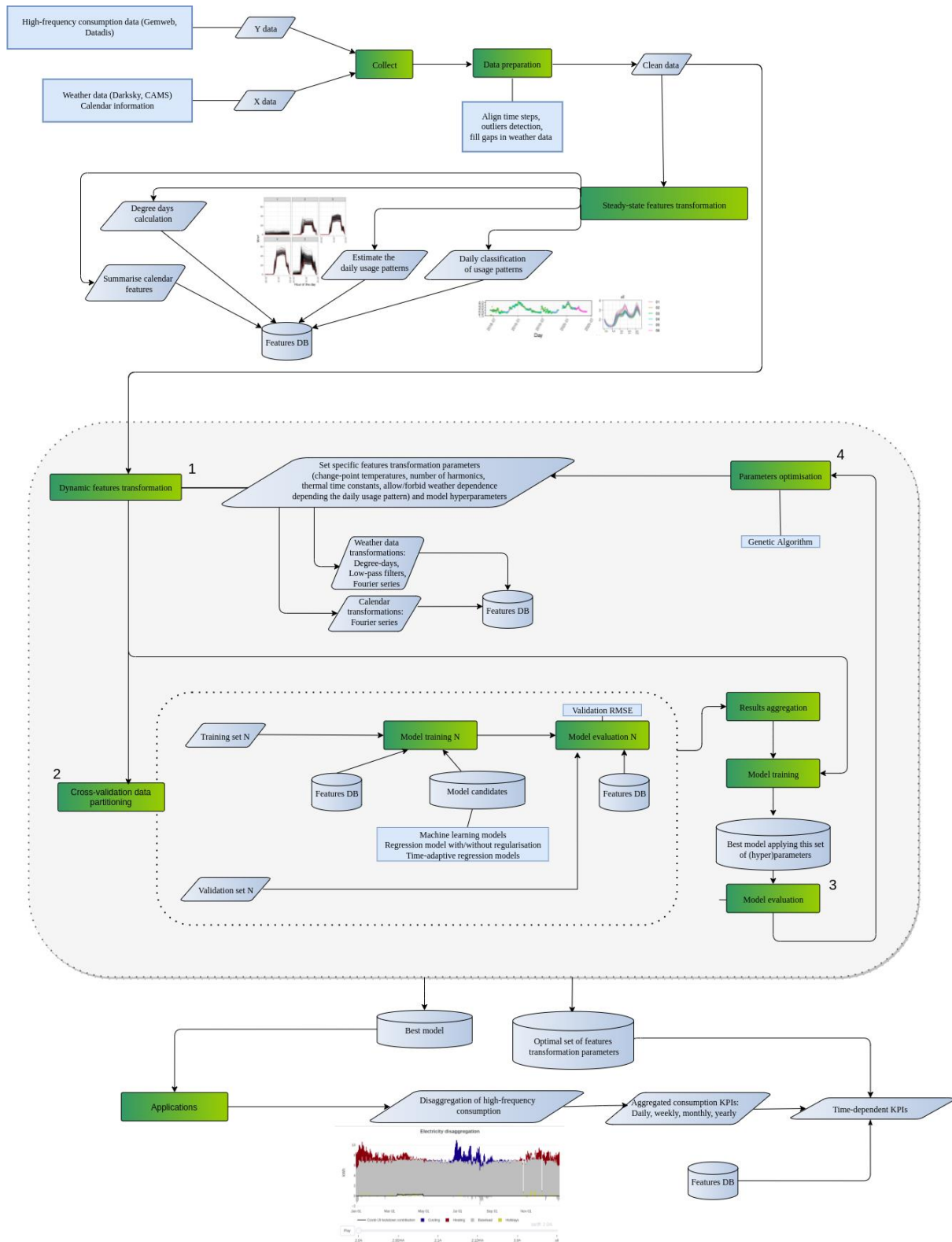


Figure 11 - Pipeline Flowchart - BC1 – Longitudinal Benchmarking

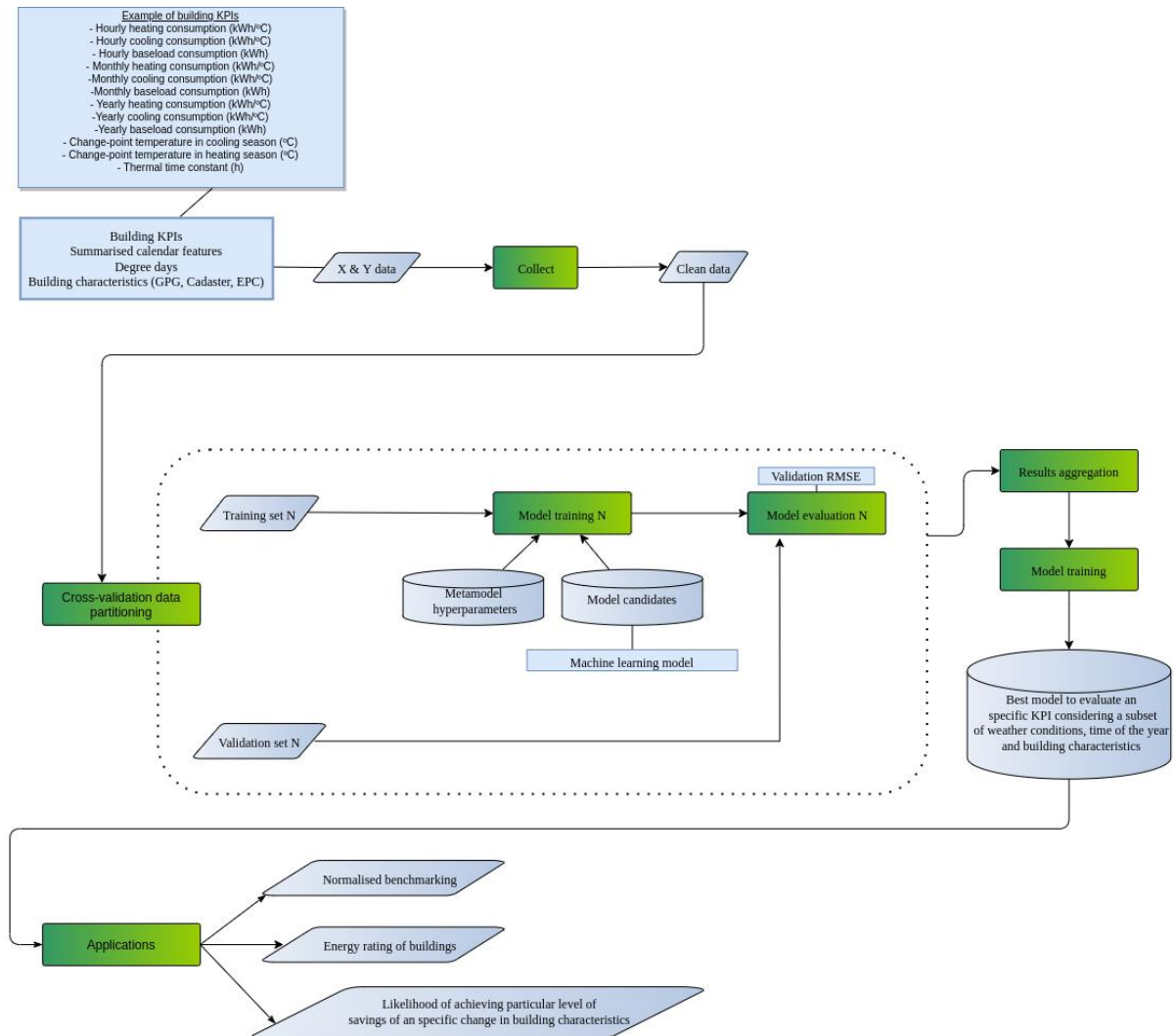


Figure 12 - Pipeline Flowchart - BC1 – Cross Sectional Benchmarking

Afterwards, BC1 integrates a use case to evaluate the energy savings caused by Energy Conservation Measures (ECM) implemented in buildings based on data-driven models and historical consumption time series analysis. In this case, the baseline models used to achieve the objective are similar to those used to disaggregate the general consumption of the building. Essentially, a forecasting of historical data after a certain ECM is implemented based on a model trained with data before the ECM, so-called baseline model. Therefore, the difference in consumption can be assumed to be the energy savings caused by the ECM.

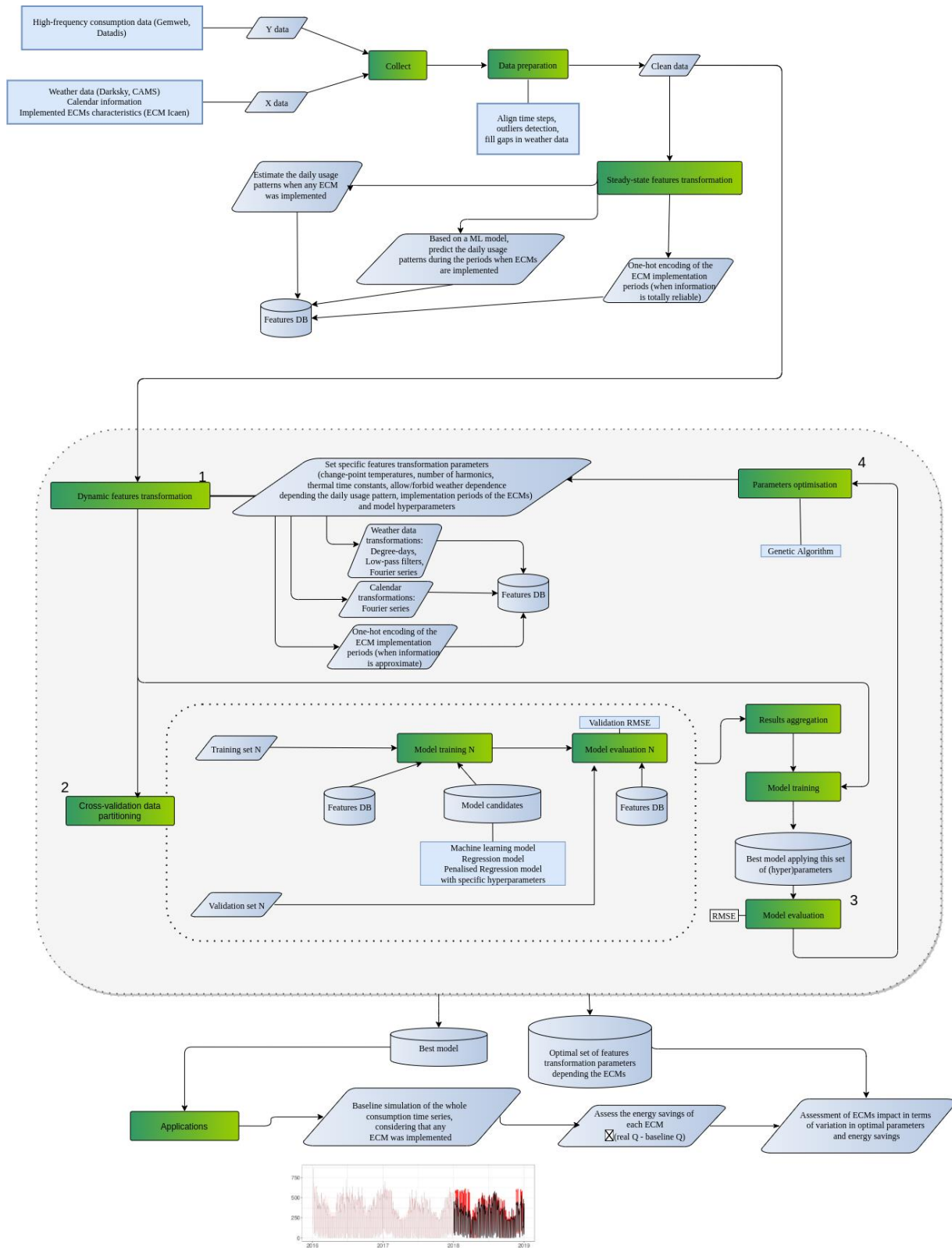


Figure 13 – Pipeline Flowchart - BC1 – ECM Results Benchmarking

III.4.3. BC2

The objective is to characterise the Spanish energy consumption certificates using open data sources such as the Spanish cadaster, building stock characteristics, weather data and aggregated consumption by location. Afterwards, using this model, an estimation of the labelling, energy demand or energy consumption over large geographical areas can be done, helping to boost the knowledge about the territory.

Inputs:

Harmonised datasets to the BIGG data model of aggregated open datasets.

- Energy Performance Certificates (EPC) datasets provided by the ICAEN, which is the Catalan energy institute.
- Aggregated smart-metered electricity time series by postal code gathered through Datadis, which is the Spanish DSOs data platform.
- Aggregated annual gas consumption by municipality provided by ICAEN.
- Aggregated socio-economic data by census tract provided by the Spanish National Statistics Institute (INE).
- Building characteristics obtained from the Spanish Cadaster INSPIRE-harmonised datasets.
- Local weather data gathered through the Darksy online services. In the case of solar radiation, the information is gathered using the Copernicus Atmospheric Monitoring Service (CAMS).

Outputs:

- Drivers of EPC indicators at the Catalan territory
- Territorial assessment of real consumption versus EPC simulations
- Estimation of energy poverty indexes.

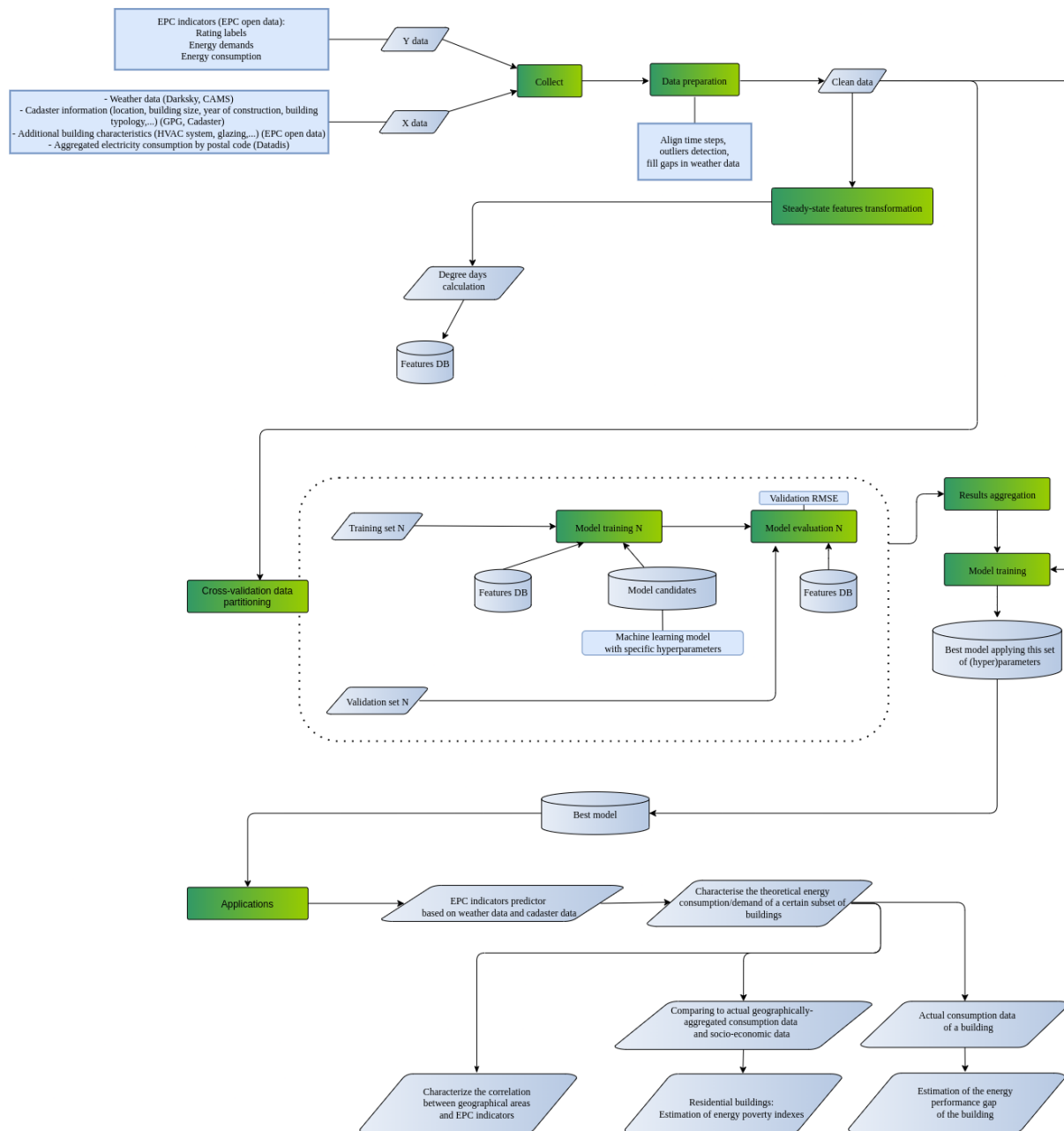


Figure 14 – Pipeline Flowchart - BC2 - EPC Characterization

III.4.4. BC4

The goal of BC4, more specifically UC9, is to estimate the savings from a building retrofit. To this end, we aim to identify a regression model for consumption data (Electricity and Gas), based on weather, occupation and calendar data. The model is trained on a pre-retrofit period and used on a post-retrofit period to evaluate the realised savings. Accuracy is evaluated according to the CVRMSE, NMBE and R^2 , as prescribed by the IPMVP protocol.

This pipeline was applied to the Interamerican building whose data was provided by CORDIA.

Inputs:

- Harmonized timeseries datasets for the total electricity or gas consumption; provided by CORDIA for the Interamerican building
- Harmonized timeseries datasets for weather data, including temperature and irradiation; collected from the weather service Weatherbit
- Training period, namely the pre-retrofit period of the building; provided by CORDIA for the Interamerican building

Outputs:

- Best regression model for the consumption data, provided as a harmonized timeseries dataset
- Confidence intervals for the regression model
- Model evaluation (according to IPMVP protocol): CVRMSE, NMBE, R^2
- Objective function used (combined criteria)
- List of removed outliers
- Training, and validation sets used by the cross-validation

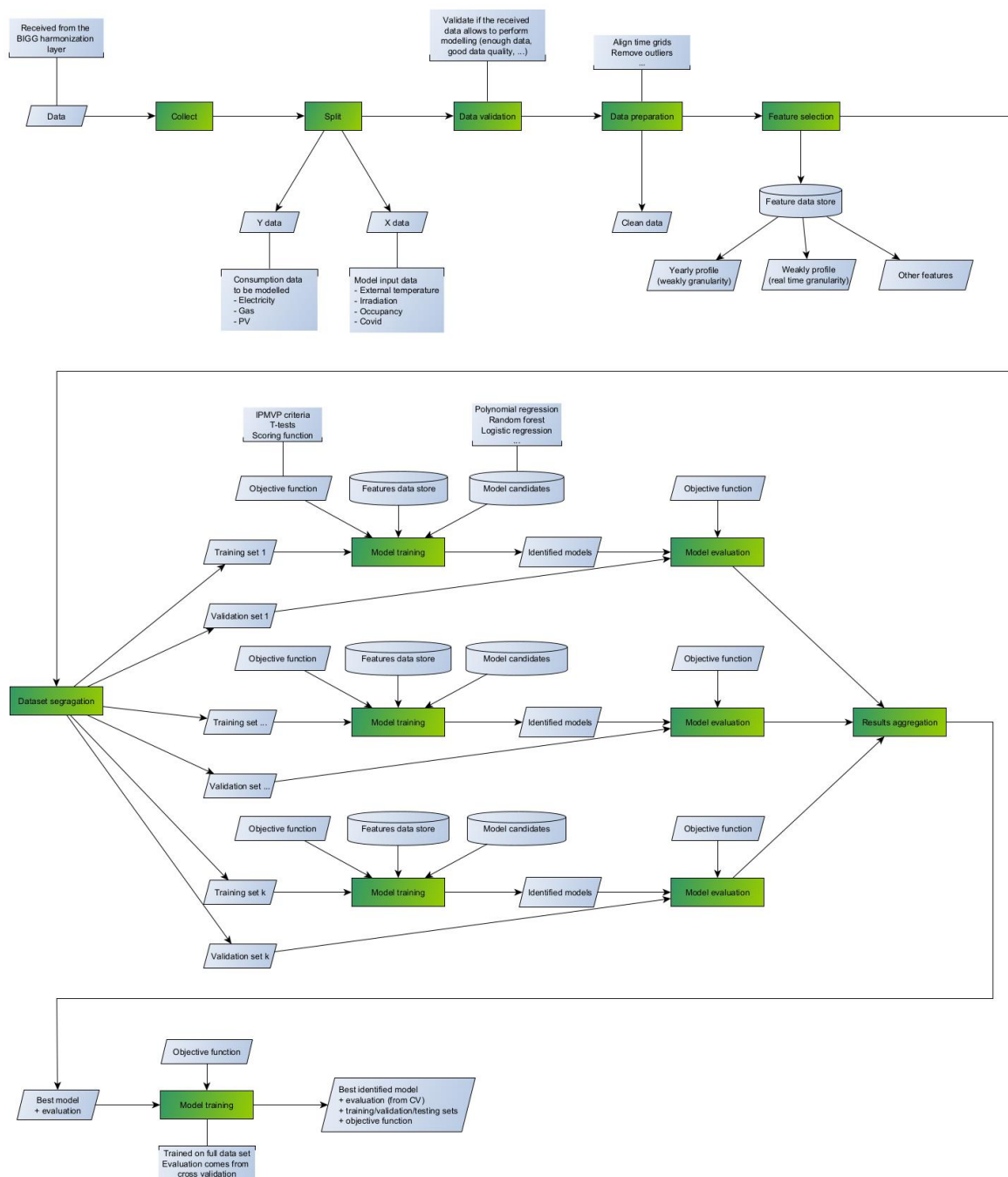


Figure 15 – Pipeline Flowchart - BC4 - EPCo Management Facilitation

III.4.5. BC5

For BC5, team designed pipeline to model the occupancy pattern of the entire building or specific zones based on the data coming from movement sensors. These data, together with calendar data and possibly holidays, can be used to train a model and predict the occupancy for the coming hours. One of the objectives is to support BC5-UC12 and make the predictions available to the rule-based controller to improve the decision-making process. This pipeline would require at least three years of data to have a good predictive model. The movement sensor and holiday data are collected, cleaned up and aligned. After this initial step, calendar components are extracted from the datetime index of the time series, added to the dataset as

new features and transformed to cyclic components. Finally, the best model, between all the model families, is detected using a nested cross-validation procedure and stored in a serialized format. The final model can be loaded when needed to make predictions on new X data.

Inputs:

- Activity Counter Time Series (movement sensor data)
- Time Series of Holidays (optional)

Outputs:

- Predictive Model of the Occupancy

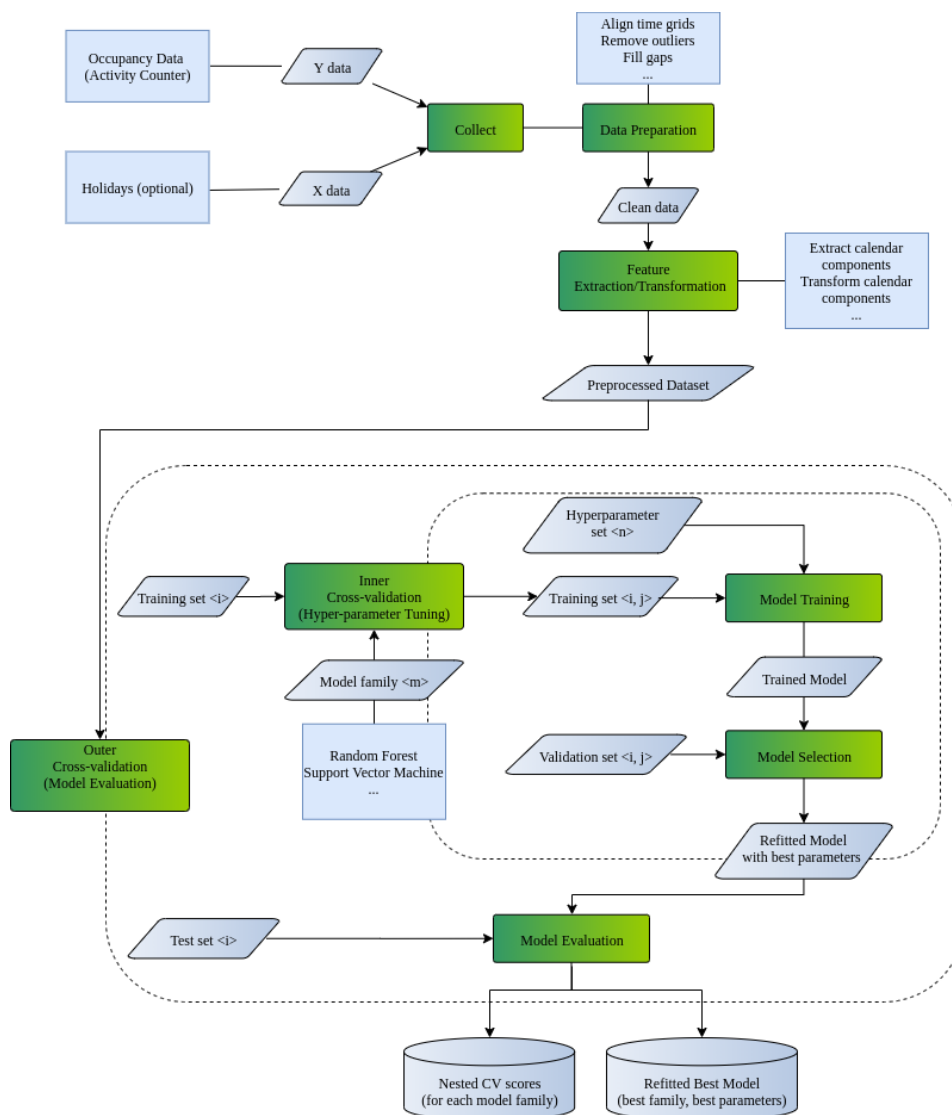


Figure 16 - Pipeline Flowchart - BC5 – Comfort Case

III.4.6. BC6-UC14

For BC6-UC14, we designed a pipeline to predict the next 24h of electrical consumption of a household.

We collect the historical residential data from Heron's API coming from smart meters. These power consumption data, along with holidays and temperature data are imported to train different models. All these data are collected and cleaned up. Secondly, calendar components are extracted from the datetime index of the time series and added as a feature. We then check if there is any correlation between the components (power consumption-day of the week and the weather (temperature) of the same day). With all of that, we train several models to predict the power consumption for a household for the next 24 hours and the best model is selected.

The pipeline below demonstrates the flow from the raw data until the final step which is the forecast.

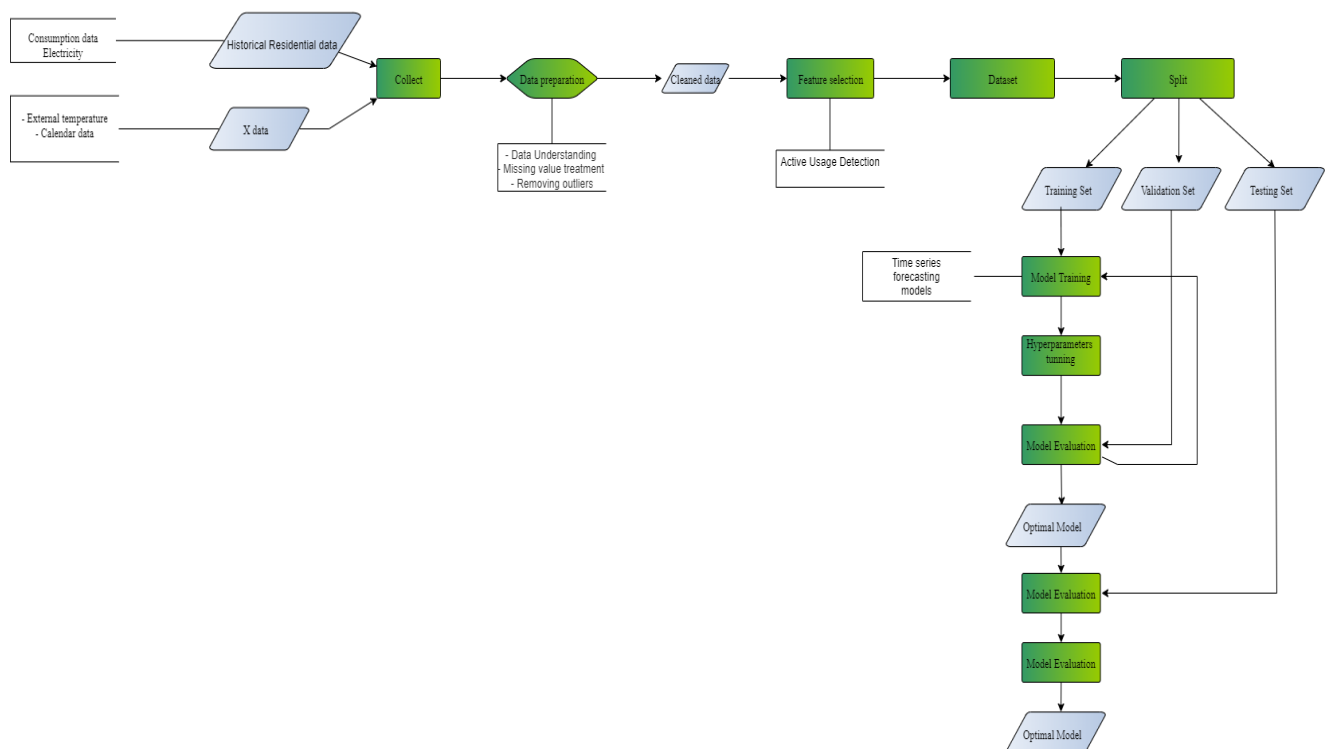


Figure 17 – Pipeline Flowchart - BC6 – Electrical Flexibility

Inputs:

Historical residential power data Temperature data per hour

- Outputs:
- Forecast per hour (Predictive model for the next 24h of power consumption)

III.4.7. BC6-UC15

The goal of BC6-UC15, is to develop a demand response (DR) scheme exploiting gas flexibility in space heating for residential complex. We propose a reinforcement learning (RL) approach to learn this demand response, that results in a controller policy. This RL policy can be learned and validated in an offline setting, feeding from historical and/or simulated data, and deployed in the real world later on. The pipeline below shows the flow of raw data, that is used to learn an RL agent that implements the DR.

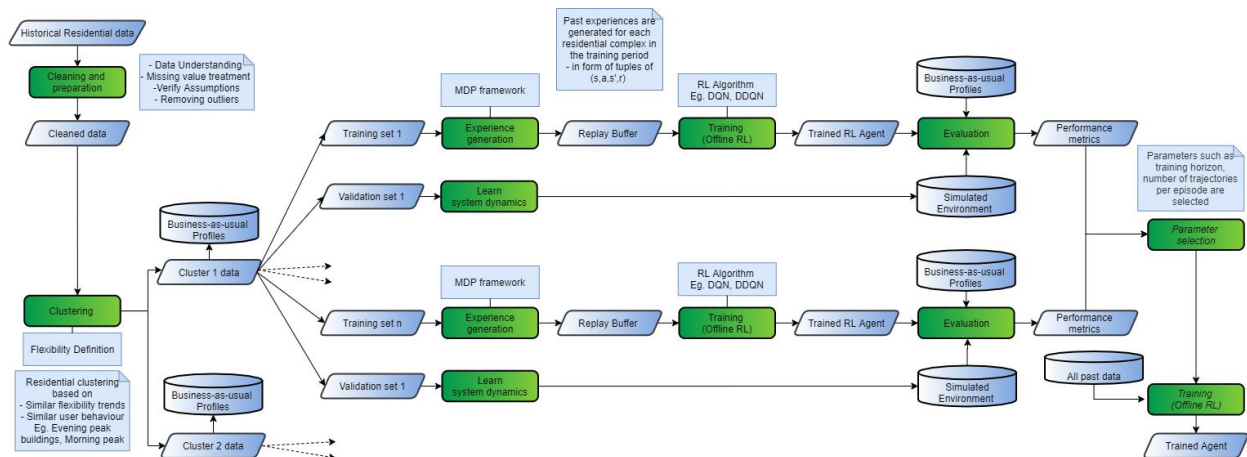


Figure 18 – Pipeline Flowchart - BC6 – Gas Flexibility

Data is transformed for training and validation of the models, and a flexibility parameter – the gas modulation level is defined and included. This transformed dataset is then used to learn a RL based policy.

Training the RL policy necessitates the definition of an agent and an environment. A simulator of residential space heating is required to define a virtual environment, which can be used to learn, validate and evaluate the RL DR. Additionally, this model can be integrated with RL based algorithm. In this deliverable, we focus on modelling flexibility, to develop a thermal model of space heating – which is used to define the environment of the RL training paradigm.

Inputs: Historical Residential data collected per household including,

- Timestamp (DD/MM/YY hh:mm:ss),
- Gas consumption/modulation,
- Room temperature/Boiler temperature,
- Boiler set points,
- Outside temperature, and,
- Room temperature set points.

Outputs:

- Sequence of control actions to achieve specified gas consumptions
- RL agent that can be used for demand response
- Trained thermal model of the system can that be used for data simulation

III.5. Function Blocks and commonalities identification

The creation of the flowcharts presented above made it possible to identify commonalities between use cases and eventually led to a list of necessary Function Blocks on a BC basis.

Careful attention needed to be taken during this step. One underlying problem of the flowchart representations is the potential misconception of a given item which may lead to mistakenly combining two Function Blocks appearing to be similar. To solve this problem it was decided to list out all the necessary actions and identify what were the expected inputs, what Function would be applied on it and what were the expected outputs. These elements combined define a Function Block.

This additional step allowed to reach a deeper understanding of each Function Block and to change the development approach. Two Function Blocks presenting similar Functions but different inputs could as an exemple be developped as two different Function Blocks or trigger a redefinition of the Function Block to make it more flexible and allow different types of inputs.

As a result, the final list was composed of unique Function Blocks defining :

- its inputs
- its Functions
- its outputs

To clarify, Function Blocks are sorted in different modules and module blocks. A module is a group of Function Blocks which all refer to a given aspect of data management or analytics and module blocks are dividing modules in 4 main categories:

- Data preparation modules,
Function Blocks linked to the early phase of data management such as data quality checking, outliers detection, time stamp management... This Function Block relates to task 5.2
- Data transformation modules,
Function Blocks involving data classification and secondary dataset management such as calendar elements, weather data elements. This Function Block relates to task 5.2.
- Modelling modules
Function Blocks involving data model generation, assessment and testing. This Function Block relates to task 5.3
- Reinforcement learning modules.
Function Blocks related to the creation and the training of a reinforcement learning agent. Reinforcement learning is a specific type of machine learning. At this stage of development, it was considered the best approach to the BC6. Additional ML modules will likely enter this list in the second phase of the AITB creation process. . This Function Block relates to task 5.3

III.6. Identified Existing libraries

An additional identification step was taken to list existing libraries that would present similar Functions and could then be leveraged to help the BIGG use cases and diminish the needs for specific Function Blocks development.

III.6.1. Python libraries

Along with the Function Blocks to be developed, the AI Toolbox relies on well-known open-source Python libraries, used as a basis to implement other functions to support the BIGG use cases. For example:

Pandas: provides the basic data structures and algorithm to represent, explore and manipulate datasets (pandas.pydata.org).

Numpy: provides data structures and algorithms related to the numerical computation, like arrays, matrices, linear algebra functions, etc. (numpy.org)

Sklearn: used especially for the machine learning modelling part. It provides the implementation of several machine learning algorithms and other tools to support and evaluate the modelling process, like optimization and cross-validation frameworks, performance metrics, etc. (scikit-learn.org/stable/)

Statsmodels: complements the Sklearn library, offering other machine learning algorithms more suited for time-series data and functions for conducting statistical tests and data exploration. (statsmodels.org/stable)

Pycaret: low-code machine learning library in Python that automates machine learning workflows. It is an end-to-end machine learning and model management tool that speeds up the experiment cycle. We have discovered that library from another European project and have implemented it here. (pycaret.org)

Prophet: is a forecasting procedure for time series data. It is implemented in R and Python and released by Facebook. (facebook.github.io/prophet/)

Sarimax: part of Statsmodels library, Sarimax stands for seasonal auto regressive integrated moving average with exogenous factors. It is a model used for forecasting time series data in Python. ([www.statsmodels.org/stable/examples/notebooks/generated/](https://www.statsmodels.org/stable/examples/notebooks/generated/statespace_sarimax_stata.html) statespace_sarimax_stata.html)

PyTorch: an open source machine learning framework based on the Torch library, used for the development of graphical models such as the neural network-based thermal model of space heating. (pytorch.org)

pytorch-lightning: A wrapper library on top of pytorch that is used to streamline the training and evaluation process of PyTorch models. (www.pytorchlightning.ai)

III.6.2. R libraries

In the case of R, multiple external libraries (a.k.a. packages) were used along with biggr implementation, such as:

Lubridate: provides tools that make it easier to parse and manipulate dates. (lubridate.tidyverse.org)

Readr: provides a fast and friendly way to read rectangular data (like 'csv', 'tsv', and 'fwf'). It is designed to flexibly parse many types of data found in the wild, while still cleanly failing when data unexpectedly changes. (readr.tidyverse.org)

Tidyr: is a set of tools to help to create tidy data, where each column is a variable, each row is an observation, and each cell contains a single value. This package contains tools for changing the shape (pivoting) and hierarchy (nesting and 'unnesting') of a dataset, turning deeply nested lists into rectangular data frames ('rectangling'), and extracting values out of string columns. It also includes tools for working with missing values (both implicit and explicit). (tidyr.tidyverse.org)

Zoo: is an S3 class with methods for totally ordered indexed observations. It is mainly aimed at irregular time series of numeric vectors/matrices and factors. The package key design goals are independence of a particular index/date/time class and consistency with ts and base R by providing methods to extend standard generics. (cran.r-project.org/web/packages/zoo/index.html)

Roll: provides a fast and efficient computation of rolling and expanding statistics for time-series data. (cran.r-project.org/web/packages/roll/index.html)

Padr: transforms datetime data into a format ready for analysis. It offers two core functionalities; aggregating data to a higher level interval (thicken) and imputing records where observations were absent (pad). (cran.r-project.org/web/packages/padr/vignettes/padr.html)

Quantreg: provides a framework to estimate and infer models of conditional quantiles. Specifically, Linear and nonlinear parametric and non-parametric (total variation penalised) models for conditional quantiles of a univariate response and several methods for handling censored survival data (cran.r-project.org/web/packages/quantreg/index.html)

Testthat: is a testing framework for R that is easy to learn and use, and integrates with your existing 'workflow' (testthat.r-lib.org).

Kernlab: provides kernel-based machine learning methods for classification, regression, clustering, novelty detection, quantile regression and dimensionality reduction. Among other methods, includes Support Vector Machines, Spectral Clustering, Kernel PCA, Gaussian Processes and a QP solver. (cran.r-project.org/web/packages/kernlab/index.html)

fastDummies: create dummy columns with categorical variables (character or factor types). Much faster than `model.matrix()` integrated in base R package. (cran.r-project.org/web/packages/fastDummies/index.html)

Caret: provides miscellaneous functions for training and plotting classification and regression models. Caret (short for Classification And REgression Training) is a set of functions that attempt to streamline the process for creating predictive models. The package contains tools for: data splitting, pre-processing, feature selection, model tuning using resampling, variable importance estimation, as well as other functionality. (cran.r-project.org/web/packages/caret/index.html)

penalised: is used for fitting possibly high dimensional penalised regression models. The penalty structure can be any combination of an L1 penalty (lasso and fused lasso), an L2 penalty (ridge) and a positivity constraint on the regression coefficients. (cran.r-project.org/web/packages/penalized/index.html)

onlineforecast: is a framework for fitting time-adaptive forecasting models. Provides a way to use forecasts as input to models, e.g. weather forecasts for energy-related forecasting. The models can be fitted recursively and easily set up for updating parameters when new data arrives. (cran.rstudio.com/web/packages/onlineforecast/index.html)

III.7. Consolidated list of Function Blocks

The list of identified Function Blocks extracted from this process, once removed the functionalities already available as open source libraries, is displayed below.

Module	Module block	Function Block	Function Block UID	Data input	Function	Data output
Data preparation	Time stamps alignment	detect_time_step	001	Timeseries, measurementReadingType in {on_change, cumulative, instantaneous}	Detect minimum time step that can be used with the time series Treat calendar features separately (e.g. holidays are daily but must still allow the timeStep to be lower than days)	Time step
		align_time_grid	002	Timeseries, measurementReadingType in {on_change, cumulative, instantaneous}, outputTimeStep, maxMissingTimeSteps	Align data with a regular time grid and detect time gaps	Discrete time series
		clean_ts_integrate	003	Cumulative time series, measurementReadingType in {on_change, cumulative}	Convert cumulative measurements to instantaneous	Instantaneous time series
	Outlier detection	clean_ts_min_max_outliers	004	Time series, min, max	Detect elements of the time series outside the min max range - min / max represents the data range in which you know the data should be. Example for consumption, should be positive and not exceeding the total capacity of all the consumers combined. Not always easy to define.	Logical time series clean time series
		clean_ts_znorm_outliers	005	Time series, znormThreshold, znorm_window	Detect elements of the time series out of the z-normalization transformation threshold	Logical time series clean time series

Module	Module block	Function Block	Function Block UID	Data input	Function	Data output
		detect_ts_calendar_model_outliers	006	Time series, lm_quantileThreshold, holidays calendar	Detect elements of the time series out of the XX% confidence of a linear model based on calendar variables (month, weekday, hour)	Logical time series clean time series
		plot_outliers	007	Time series, outliers logical time series, cleaned time series, showPlot	Plot the data cleaning process of a time series to expose the outliers	Plot *.svg/*.pdf/*.html
		detect_static_min_max_outliers	008	Value, min, max	Detect element outside the min max range (To be discussed _ seems not to be used today)	Logical
		detect_static_reg_exp	009	Value, regexp, possibleValues	Detect if the element fulfill the regular expression or it exists in the list of possible values	Logical
	Missing data management	clean_ts_fill_NA	010	Time series, outliers logical time series, methodFillNA, maxGap, fillMask	Fill the outlier' elements using last or previous value, the lm_calendar model value, interpolation between the gap, uniform interpolation of next value, regression based interpolation of next value, ... fillMask specified time steps that must be filled	Cleaned time series
Data transformation	Profiling	clustering_dlc	013	Consumption and outdoor temperature time series, timeZone, percCons, kmax, nDayparts, normSpecs, inputVars, showPlot	Cluster similar daily load curves based on the load curves itself, calendar variables and outdoor temperature	Clusters detected and the characteristics of them

Module	Module block	Function Block	Function Block UID	Data input	Function	Data output
		classification_dlc	014	Consumption and outdoor temperature time series, args: dlcCentroids, clusteringCentroids, clusteringModCalendar, timeZone, percCons, nDayparts, normSpecs, inputVars, showPlot	Classify daily load curves based on the outputs of a clustering and a new set of data	Classification
		weekly_profile_detection	015	Consumption timeseries, holidays	Derive weekly consumption profile	Weekly consumption profile timeseries
		yearly_profile_detection	016	Consumption timeseries	Detect yearly consumption profile	Yearly consumption profile timeseries
		trend_estimation	017	Consumption timeseries Degree Days timeseries	Detect overall trend (over multiple years), independently from weather impact	Trend consumption timeseries
	Weather	degree_days	018	Outdoor temperature time series, changePointTemperature, mode, outputFreq	Calculate the degree-days with a desired output frequency ("yearly", "monthly", "daily") and considering cooling or heating mode.	Aggregated degree days
		degree_raw	019	Outdoor temperature time series, changePointTemperature, mode, outputFreq="raw"	Calculate the degree-<frequency> depending the frequency of the raw temperature series (e.g. degree-hours) with a desired output frequency ("yearly", "monthly", "daily", "raw") considering cooling or heating mode.	Aggregated Degree raw
	Autoregressive processes	lag_components	21	data: timeSeries parameters: maxLag, featuresNames, predictStep, forceGlobalInputFeatures, forceInitInputFeatures,	This function shift in time a set of features in order to be used in the training and prediction of the models. It is an important step for the multi-step prediction of Autoregressive models, where the estimated output is directly used in the subsequent predictions.	data: timeSeries

Module	Module block	Function Block	Function Block UID	Data input	Function	Data output
				forceInitOutputFeatures , fillInitNAs		
		lpf_ts	022	Time series, smoothingTimeScaleParameter	First-order low pass filter (smoothing) Use cases: - with consumption data, it helps removing artificial fluctuation - with outdoor temperature data, it helps to linearise the relation between consumption and outdoor temperature, as it simplifies the modelling of the thermal inertia of the building	Smooth consumption time series
		get_lpf_smoothing_time_scale	023	timestepsPerHour, timeConstantInHours	Physical transformation of the smoothing time scale parameter Use cases: - for outdoor temperature, timeConstantInHours means the thermal inertial of the envelope of the building in hours	smoothingTimeScaleParameter
	Calendar	calendar_components	024	Time array	Decompose the time in date, day of the year, day of the week, day of the weekend, working day, non-working day, season, month, hour, minute, ...	Matrix of the calendar components
	Fourier series	fs_components	025	Time series, minCycle, maxCycle, nHarmonics	Decompose a cyclic time series (e.g. solar azimuth, solar elevation, calendar features, ...) into the components of sin(x), cos(x) depending the number of harmonics provided	Fourier series components matrix

Module	Module block	Function Block	Function Block UID	Data input	Function	Data output
Modelling	Model assessment	evaluate model cv_with_tuning	34	model_family: object. X_data: timeSeries. y_data: timeSeries parameter_grid: dict. scoring: string or list of string or None. cv_splitter_outer: Generator. cv_splitter_inner: Generator	This function performs a nested cross-validation (double cross-validation), which includes an internal hyper-parameter tuning, to reduce the bias when combining the two tasks of model selection and generalization error estimation. However, the purpose of this function is not to select the best model instance of a model family but instead to provide a less biased estimate of a tuned model's performance on the dataset.	scores: dict. cv_results: list of dict
	Model Identification	identify_best_model	35	X_data: timeSeries y_data: timeSeries Arguments: cv_splitter_outer: Generator, cv_splitter_inner: Generator, scoring: string or list of string, compare_with: string	This function implements a complete generalized pipeline for supervised learning to find the best model among different model families, each one associated with a specific parameter grid, given an input time series and a scoring function.	best_model_instance: object, best_params: dict, scores: dict, cv_results_final: dict, cv_results_evaluation: dict
	Model Persistence and prediction	serialize_model	40	model_instance: object, model_full_path: string, format: string	This function serializes and saves a model instance, with a given file format, to a specific path on the file system.	file_name: string
		deserialize_and_predict	41	model_full_path: string X_data: timeSeries	This function deserializes a model, inferring the file format from the file name, applies the model on the X_data and returns the predicted values in the form of a time series.	y_data: timeSeries
		test_stationarity_acf_pacf	42	timeseries	Calculating the p-value to estimate the stationarity	Stationarity characteristics
		Split train/test	43	ts: timeSeries test: float (ex: 0.2) or str plot: bool	This function splits the time series into train and test datasets at any given data point.	ts_train: timeSeries ts_test: timeSeries Optionalplot

Module	Module block	Function Block	Function Block UID	Data input	Function	Data output
		param_tuning_sarimax	44	hyperparameters	Selecting a set of optimal hyperparameters for a learning algorithm	Optimal hyperparameters
		param_tuning_prophet	45	hyperparameters	Selecting a set of optimal hyperparameters for a learning algorithm	Optimal hyperparameters
		input_prophet	46	ts_train: timeSeries ts_test: timeSeries	This function adapts the training and testing datasets to match with the requirements of Prophet model.	dtf_train: Dataframe dtf_test: Dataframe
		fit_prophet	47	dtf_train: timeSeries	This function trains and fits a PROPHET model	model: Object
		test_prophet	48	_dtf_test: timeSeries model: Object p: int. freq: str	This function gets the prediction of the prophet model	dtf_forecast: timeSeries
		fit_sarimax	49	hyperparameters, training data, model_function	Training of the sarimax model using a training dataset	Trained sarimax model
		test_sarimax	50	ts_train: timeSeries ts_test: timeSeries exog_test: timeSeries p: int model: Object	This function gets the prediction of the sarimax model.	dtf_test: timeSeries
		evaluate_forecast	51	dtf: timeSeries. plot: bool	This function calculates evaluation metrics for the prediction	dtf_eval: timeSeries. plot
rl_definitions	-	dynamics functions	62	parameters of the space-heating system	Functions that implement the dynamics of space heating -Space-heating grey-box RC model -Gas modulation model Based on these dynamics, we can calculate the next state of the system	Next Room temprature, gas modulation, boiler temperature

Module	Module block	Function Block	Function Block UID	Data input	Function	Data output
	-	phycell	63	dynamics, sequence length	The recurrent unit that implements the dynamics of the system. This is used in the thermal model to sequentially apply physics.	phycell class
		thermal_model	64	phycell, learning_rate, dense_net, data for training (room temperature, boiler temperature, setp points, outside temperature)	the main thermal model class, this class implements the phycell (for physics of the system), and a dense network to estimate the outside disturbances. This model object should have training and testing capabilities. This acts as the environment class training the RL model	model class
		rl_environment	65	thermal model	A wrapper that impl	Environment class
rl_agent_select		rl_agent_select	72	rl_cross_validation, rl_agent, rl_environment, data, search grid	To select hyper parameters from cross validation, and train the final agent based on them	Trained Agent

Figure 19 – List of Function Blocks to be developed

III.8. Collaborative work management and tools

Given the complexity, the wide variety of methods used and given the remote configuration of the team work organization, it was necessary to define a common framework under which the development work would be carried out. The AITB involved implementation of code developed by people from 4 companies in 4 countries. It was chosen to use a web-based open source version management software which would allow multiple people to make changes at the same time on a given document or section of a document. Many tools exist that offer these features (Bitbucket, GitLab, Google Cloud Source Repositories, Phabricator, RhodeCode, ...). Github (github.com) was chosen to host the developed code.

A tool was also needed to test the code and keep track of the different testing phases across the different phases of the project. For this, the WP5 team members chose to work with ML Flow (www.mlflow.org).

Both tools are open source and present very similar levels of functionality when compared to their peers. The final choice was made with a focus on WP members' familiarity with the tool. The point being to minimize training needs.

These two cloud based software tools are presented below:

III.8.1. Collaborative code development tool

III.8.1.a. Github

GitHub provides cloud hosting for software development and version control through Git. In essence, it provides distributed version control and source code management (SCM) functionality, plus other tailored features. Every project has access control and collaboration features including bug tracking, feature requests, task management, continuous integration, and wikis.

All code and documentation repositories of the BIGG project are hosted in GitHub. In particular, a BIGG user account was created containing the R and Python AI-toolbox libraries, so-called biggr and biggpy respectively, and the language-agnostic documentation of the AI toolbox, so-called biggdocs. All repositories are public, so freely available to everybody.

Besides, the GitHub task management functionality was used in the biggdocs repository for project management of the AI-toolbox implementation. A similar management framework will be used for the implementation of the business cases pipelines.

BIGG GitHub account: <https://github.com/bigproject>

Documentation of the AI Toolbox: <https://github.com/bigproject/biggdocs>

Python implementation of the AI Toolbox: <https://github.com/bigproject/biggpy>

R implementation of the AI Toolbox: <https://github.com/bigproject/biggr>

III.8.1.b. Lifecycle code management - ML Flow

It was decided to use MLflow as an open-source framework to manage the entire lifecycle of the AI Toolbox applications. This ML platform is based on four components:

- MLflow Tracking: allows any piece of data science code to be recorded as a run and organized as an experiment. The tracking component keeps trace of two main elements: artifacts, such as figures, serialized models, configuration files, model summaries and machine learning entities, such as model hyper-parameters, performance metrics, and other metadata related to machine learning. Artifacts can be logged in a local or remote artifact store, for example in an Amazon S3 bucket, while

mlflow entities are usually recorded in local or remote databases, such as PostgreSQL. MLflow also allows us to store interactive html figures, that comes in handy when monitoring the pipeline. This way, one can double-check if an intermediate step of the workflow has produced the expected outcome, for example inspect the outlier plot, an autocorrelation plot or the predictions of a model.

- MLflow Projects: allows any data science project to be packaged together with its dependencies, entry points and environment. This way, the code can run on several platforms in a reusable and reproducible way. For example, the environment on which to run the code can be a docker container with the latest version of the AI toolbox already pulled from GitHub and installed.
- MLflow Models: allows to package and deploy machine learning models using standard formats. For example, a model can be deployed as a self-contained Docker image with a REST API endpoint and serve prediction requests. However, models can also be deployed on cloud platforms like Microsoft AzureML or Amazon Sagemaker.
- MLflow Model Registry: is a centralized model store, set of APIs, and UI, to collaboratively manage the full lifecycle of an MLflow Model. Other than storing the model, this component provides also model versioning, model lineage and stage transitions.

All these components together can facilitate and speed up the monitoring, deploying and model versioning phases, without reinventing custom solutions.

IV. PRELIMINARY VERSION OF THE AI TOOLBOX

IV.1. Data collection and data format

While data collection is not an expected service from the AITB, data format had to be taken into consideration. As presented above in the section [Interaction with other work packages](#), the AITB is intended to enable analytics and the implementation of AI techniques over a large variety of data. Data format is a key parameter that had to be taken into consideration. The final version of the AITB is expected to be based on harmonized data inputs and harmonized data outputs. In this case harmonized data refers to data formatted to the **BIGG Standard Data Model 4 Buildings**, as presented in the deliverable: *D4.1 - Description of the preliminary harmonization layer_v1*.

This harmonization of inputs and outputs will be performed at Pipeline level. Whereas the preliminary version of the toolbox presents a list of available Function Blocks that can be used independently, the Harmonization is not expected to be performed at that level of granularity. Each Pipeline will be based on harmonized data inputs and harmonized data outputs but the in between steps involving various Function Blocks are not necessarily involving only harmonized data streams.

IV.2. Data storage

Data storage was identified as the very first task of WP5 (Task 5.1, see section [Task 5.1 - Provision of data storage infrastructure](#)) and the identification of storage needs were done along with the identification of all the necessary Function Blocks for the completion of the business cases.

After extensive discussion about framework and the potential impact on data security management, the BIGG consortium elected to propose an AI toolbox service that would be implemented locally by the users and would leverage their existing storage capacities. The sections below are presenting how the members of the WP involved in the implementation of the BIGG AITB for the trials implemented data storage.

The data used for BC1-BC3 are collected by different external platform companies. These datasets are gathered by means of the provider's API implementation, following the provider's instructions to get the information, or a provided Excel file.

The data gathered are then harmonized following the BIGG ontology and stored in the permanent storage system. During the harmonization process, two different types of data are identified and classified: "Building Information Data" and "Time Series Data".

Using the previous distinction, the data is stored in different databases, in order to speed up the search and later manipulation of the data.

For the Building Information Data, Neo4j has been selected as the graph database due to its proven performance and speed when making requests.

For the Time Series Data, HBase has been selected, as it can handle millions of time series points with a good performance. The data used for BC4 and BC5 is hosted on the Energis.Cloud servers.

The data metrics collected by Interamerican and Vodafone devices are provided by Engie through the Yodiwo platform API. The ingestor is of solicited kind implementing a task that queries Yodiwo at configured pace. The data metrics are stored in raw format into the MongoDB datalake. The data, flowing on a dedicated Kafka topic, are retrieved by the "processor" microservice (the Energis harmonizer) and translated in the ProcessedMessage

common platform language format. Then, these messages are stored into the KairosDB/Cassandra lakeshore and, from there, will be available to the other system components.

During the first stage of the pilots, given the low amount of data that needs to be processed at this stage, the pipeline implementations using the AI toolbox are running locally on personal computers. After further validation, they will be integrated to the Energis.Cloud software and run from their servers.

The data used for the BC6-UC14 is hosted on Heron's servers. Inetum collects the data from an API provided by Heron. After collection, the data is stored locally on Inetum's premises and used to train and evaluate different models for better predictions. So far, the amount of data collected is limited. That's why, for now, we can store the data on personal computers and run locally the AI toolbox.

The data used for BC6-UC15 were collected by DomX (<https://mydomx.eu/>), which is an IoT company that provides smart heating services for domestic households. DomX has established a network of more 50+ households across Greece, which provide opportunities for data collection and demand response. During the development of the toolbox, the dataset provided by DomX is stored locally by imec. This locally stored data is used to train and evaluate the models defined in the AI toolbox. The user can utilize the code provided by AI toolbox to train the models on a locally stored dataset.

IV.3. List of Function Blocks

This section presents a summary of the development work that was carried out after the Function Blocks were identified. The development was done collaboratively on Github and all the related documentation can be accessed directly there on the BiggDocs repository: github.com/biggsproject/biggsdocs.

The section below presents only the definition of the Function for each Function Block. It also identifies where the function is needed in current BC Pipelines. Note that some Function Blocks are intended for future versions of the BC implementations; they have already been introduced for completeness and consistency but are not used in the first version of the pipelines yet.

IV.3.1. Data preparation

IV.3.1.a. Time Stamps Alignment

IV.3.1.a.1. detect_time_step

The function infers, i.e. automatically deduce from the input data, the minimum time step (frequency) that can be used for the input time series, represented with a string alias formatted according to the ISO 8601.

Used in: BC1, BC2, BC4, BC5

IV.3.1.a.2. align_time_grid

The function aligns the frequency of the input time series with the output frequency given as an argument using the specified aggregation function

Used in: BC4, BC5

IV.3.1.a.3. clean_ts_integrate

The function converts a cumulative (counter) or onChange (delta) measurement to instantaneous.

Used in:

IV.3.1.b. Outlier detection

IV.3.1.b.1. detect_ts_min_max_outliers

This function detects elements of a time series outside the allowed range in which you know the data should be. In the case of energy consumption, it should be a positive value and not exceeding the total capacity permitted. Sometimes, this value is not easy to define. Additionally, with the minSeries and maxSeries arguments, this ranges can be set differently along the period.

Used in: BC1, BC2

IV.3.1.b.2. detect_ts_zscore_outliers

This function detects elements of the time series out of a Z-score threshold, applied on the whole time series or a rolling window of predefined width.

Used in: BC1, BC2, BC4, BC5

IV.3.1.b.3. detect_ts_calendar_model_outliers

This function detects elements of the time series out of a confidence threshold based on linear model of the calendar variables (month, weekday, hour). It estimates the outliers of a time series based on a quantile regression model that uses calendar features as input variables. This calendar features, that normally corresponds to common seasonalities, are transformed using Fourier components. However, there are two exceptions of model features that are not transformed using this technique: the intercept, which is a fixed term during all the period, and HOL (holidays) feature, which is a 0-1 depending if the day is holiday or not.

Regarding mandatory features, the intercept is the only one that will be considered even if it is not specified in the calendarFeatures argument. Another interesting point of this argument, is that it allows the interaction between terms. Thus, if we set a HOL*intercept term, a different intercept will be estimated for holidays and non-holidays.

Used in: BC1

IV.3.1.b.4. detect_static_min_max_outliers

This function detects which numerical elements are outside the min-max range. It should be used to filter outliers of static data (e.g. building areas, year of construction, ...)

Used in: BC1, BC2

IV.3.1.b.5. detect_static_reg_exp

This function detects which string element satisfy the regular expression. To test regular expressions configured in the regExpValues argument, you should use the web application <https://regexpr.com/>

Used in: BC1, BC2

IV.3.1.c. Missing Data Management

IV.3.1.c.1. *fill_ts_na*

The function imputates values to Not Available (NA) elements of a time series, based on the outliers estimation made the functions implemented in Outlier Detection module block of this library. It requires the previous usage of the Outlier Detection functions. An interpretation of the maxGap and a resample of the fillMask time step is done, considering the actual time step of the data time series. Actual methods to fill the NA elements are quite simple, but in future more complex implementation of this imputation could be integrated.

Used in:

IV.3.2. Data transformation

IV.3.2.a. Profiling

IV.3.2.a.1. *clustering_dlc*

The function clusters similar daily load curves based on the load curves themselves, calendar variables and outdoor temperature. Spectral clustering is used to infer the unknown daily load curve patterns. The minimum frequency allowed of the arguments consumption and temperature to cluster daily load curves is hourly.

Used in: BC1

IV.3.2.a.2. *classification_dlc*

The function classifies daily load curves based on the outputs of a clustering or a labelled dataset and a new set of data. The minimum frequency allowed of the arguments consumption and temperature to cluster daily load curves is hourly.

Used in: BC1

IV.3.2.a.3. *weekly_profile_detection*

The function returns the weekly profile of the input time series.

Used in:

IV.3.2.a.4. *yearly_profile_detection*

The function returns the yearly profile of the input time series.

Used in:

IV.3.2.b. Calendar

IV.3.2.b.1. *add_calendar_components*

The function decomposes the time into many features (e.g. date, day of the year, day of the week, day of the weekend, working day, non-working day, season, month, hour, minute). The transformation must be done considering the local time zone. Typically, the features generated by this function are used as model inputs for modelling the user behaviour seasonalities of energy consumption.

Used in: BC1, BC2, BC4, BC5, BC6

IV.3.2.b.2. trigonometric_encode_calendar_components

This function returns a transformer that encodes all the calendar components added to the input data into sin and cosine trigonometric cyclic components. This type of encoding greatly boosts the predictive capabilities of some models.

Used in: BC4, BC5

IV.3.2.c. Weather

IV.3.2.c.1. degree_days

The function calculates the degree-days with the desired output frequency and considers cooling or heating mode.

Used in: BC1, BC2

IV.3.2.c.2. degree_raw

The function calculates the difference between outdoor temperature and a base temperature without considering the frequency of the original data.

Used in: BC1

IV.3.2.d. Autoregressive processes

IV.3.2.d.1. lag_components

The function shifts in time a set of features for model training and prediction. It is an essential step for the multi-step prediction of Autoregressive models, where the estimated output is directly used in the subsequent predictions.

Used in: BC1

IV.3.2.d.2. lpf_ts

This function computes the first-order low pass filter for smoothing a time series. This function can be used in different cases: 1 - Consumption time series, it helps remove artificial fluctuation; 2 - Outdoor temperature time series, it helps to linearise the relation between consumption and outdoor temperature, as it simplifies the modelling of the thermal inertia of the building; 3 - Wind speed time series, it helps to smooth the wind speed data; 4 - Solar radiation time series, it helps to linearise the relation between consumption and solar radiation, as it simplifies the modelling of the solar gains of the building.

Used in: BC1

IV.3.2.d.3. get_lpf_smoothing_time_scale

The function calculates the smoothing time scale parameter of the first-order low pass filter over an input variable, considering a specific time constant in hours.

Used in: BC1

IV.3.2.e. Fourier Series

IV.3.2.e.1. fs_components

The function obtains the components of the Fourier Series in sine-cosine form. It helps linearise the relationship of a seasonal input time series (e.g. solar azimuth, solar elevation, calendar features) to some output (e.g. energy consumption, indoor temperatures).

Essentially, it decomposes a cyclic time series into a set of sine-cosine components used as inputs for modelling some output, where each of the components linearly depends on the output.

Used in: BC1

IV.3.3. Modelling

IV.3.3.a. Cross Validation

IV.3.3.a.1. BlockingTimeSeriesSplit

This class is a splitter performing a special type of time series partitioning to be used in the cross-validation framework. Differently from TimeSeriesSplit, this method will generate disjoint partitions of the dataset in each iteration.

Used in:

IV.3.3.b. Model Assessment

IV.3.3.b.1. evaluate_model_cv_with_tuning

This function performs a nested cross-validation (double cross-validation), which includes an internal hyper-parameter tuning, to reduce the bias when combining the two tasks of model selection and generalization error estimation. However, the purpose of this function is not to select the best model instance of a model family but instead to provide a less biased estimate of a tuned model's performance on the dataset.

Used in: BC4, BC5

IV.3.3.c. Model Identification

IV.3.3.c.1. identify_best_model

This function implements a complete generalized pipeline for supervised learning to find the best model among different model families, each one associated with a specific parameter grid, given an input time series and a scoring function.

Used in: BC1, BC2, BC4, BC5, BC6

IV.3.3.d. Model Persistence and Prediction

IV.3.3.d.1. serialize_model

This function serializes and saves a model instance, with a given file format, to a specific path on the file system.

Used in: BC1, BC2, BC4, BC5

IV.3.3.d.2. deserialize_and_predict

This function deserializes a model, inferring the file format from the file name, applies the model on the X_data and returns the predicted values in the form of a time series.

Used in: BC1, BC2, BC4, BC5

IV.3.4. Reinforcement learning techniques

For reinforcement learning DR, we first develop a ‘thermal model’ of the space heating system, which is used as a simulator to learn the RL policy. Below, we document the functions in BIGG toolbox about the proposed thermal model.

Used in: BC6

IV.3.4.a. Thermal Model

IV.3.4.a.1. *thermalmodel*

Description: *thermalmodel* class, can be used to create a *thermalmodel* object. This object can be trained, validated and tested using data

Used in: BC6

IV.3.4.a.2. *DenseNet*

Description: A function to create a DenseNet neural network with given layers.

Used in: BC6

IV.3.4.b. Physics Cell

IV.3.4.b.1. *PhyCell*

Description: This is *PhyCell* class. An object of this class is used as a main recurrent unit in the thermal model. A *forward()* method is implemented in the class, which takes the current hidden state *zt* and current inputs *xt*, and returns the next hidden state and output (*xt+1*, *zt+1*). The class also has *set_param()* and *set_param_grad()* methods to set the values of parameters of *PhyCell* and if they should be optimized. A *weight_loss()* methods returns the value of loss calculated for weights of the cell

Used in: BC6

IV.3.4.b.2. *PhyCell.forward()*

Description: forward method for the *PhyCell* class.

Used in: BC6

IV.3.4.b.3. *PhyCell.set_param()*

Description: method to set parameters of the *PhyCell* class

Used in: BC6

IV.3.4.b.4. *PhyCell.set_param_grad()*

Description: method to set gradient of the parameters of *PhyCell* class

Used in: BC6

IV.3.4.b.5. *PhyCell.get_param()*

Description: Returns the dictionary of the parameters of the current instance of the *PhyCell* object.

Used in: BC6

IV.3.4.c. Dynamics

IV.3.4.c.1. *RoomT_next*

Description: This is the calculation for room temperature for next time step. This is based on the space heating model.

Used in: BC6

IV.3.4.c.2. *BuildingT_next*

Description: This is the calculation for building temperature (temperature of the thermal mass of the building) for next time step. This is based on the space heating model.

Used in: BC6

IV.3.4.c.3. *BoilerInletT_next*

Description: This is the calculation for boiler outlet temperature for next time step. This is based on a decay/growth model for the boiler temperature, where a1 and a2 are the variables that control the rate of change of boiler temperature.

Used in: BC6

IV.3.4.c.4. *BoilerOutletT_next*

Description: This is the calculation for boiler outlet temperature for next time step. This is based on a decay/growth model for the boiler temperature, where a1 and a2 are the variables that control the rate of change of boiler temperature.

Used in: BC6

IV.4. Code development methodology

The AI Toolbox has been developed following the coding style conventions and common best practices.

For Python, it was decided to be compliant with the following guidelines:

- PEP8 (<https://www.python.org/dev/peps/pep-0008/>)
- PEP257 (<https://www.python.org/dev/peps/pep-0257/>).

The first PEP defines general coding conventions for Python to improve the code readability, make it consistent across different libraries and projects and ease collaboration between developers. The second PEP addresses one aspect of coding conventions: docstrings. A docstring is a string literal that is used to document a segment of code and occurs as the first statement in a module, function, class, or method definition.

For example, the docstring of a function usually begins with a brief description of the functions, the parameters and the return values:

```
def serialize_model(model_instance, model_full_path, model_format='joblib'):
    """
    This function serializes and saves a model instance, according to the specified file format,
    to the specified full path on the file system.

    :param model_instance: Model instance which has been already fitted on X data.
    :param model_full_path: String identifying the full path (not relative and with no file extensions) of the file
        where the model should be saved. The extension will be added by the function based on the format chosen.
    :param model_format: Format of the model to serialize and persist.
        By default it will use the 'joblib' pickle format.
    :return: String identifying the filename in which the data is stored. The function will add the extension
        according to the format chosen.
    """
```

IV.5. Test and verification process

The testing framework used to test the Python code is “unittest”, which is based on the main concepts of test case, test suite and test fixtures: <https://docs.python.org/3/library/unittest.html>. It was decided to write a test suite, an aggregation of tests executed together, for each module of the AI Toolbox. Each function of the toolbox can be tested versus different sets of inputs, whenever possible, to check if its response matches the expected one. This is the general concept of a test case. A test fixture is some code that prepares the test environment for entire test suite, such as importing data, connecting and creating databases or directories and implements clean-up actions at the end. Usually, the purpose of writing tests is to make sure that some code works as expected in different environments and that new code added to the library does not break the other functionalities.

The other tool we decided to use for the test and verification process is called “tox”: <https://tox.wiki/en/latest/>. It is a command line tool to check that some package or library can be installed and tested successfully in multiple environments, such as using different Python interpreters. Tox will create one virtual environment for each python interpreter specified in a configuration file, install the package or library in that environment and run all the tests. This is particularly useful to make sure that the AI Toolbox works correctly with different versions of Python.

To provide further support to the documentation process, we created and included for most of the functions in the AI Toolbox a Jupyter notebook which serves as a guideline on how to use them: https://github.com/bigqproject/bigqpy/tree/main/ai_toolbox/notebooks.

Some of the datasets imported in the Jupyter notebooks and in the tests are “toy datasets” already integrated in other libraries as submodules, such as “sklearn.datasets”. The dataset used for testing models developed for UC 15 is a residential heating dataset, including past temperature and gas consumptions. The data is handled and models are trained using submodules and functionality provided by “pytorch_lightning”.

In the case of R, the “testthat” library was used to run tests every time the library is compiled, mainly checking that each function is providing an expected response to a given set of inputs. The implementation is quite similar to the ‘unittest’ used in Python library, and has the same objectives. Regarding the user tutorials implemented in R, first of all, multiple datasets containing electricity consumption and weather data of six different buildings are included inside the biggr package. These datasets are used in the vignettes (R notebooks in HTML format), which become part of the library, showing the potentialities of analysing harmonised building energy consumption data with the presented AI Toolbox. Each of these tutorials proposes an implementation of a certain BC Pipeline.

V. STATUS OF IMPLEMENTATION

As presented in the section [AITB Development methodology](#), after the initial development phase of the itemized AITB with all Function Blocks presented separately, the AITB needs an additional development step where Function Blocks are assembled into Pipelines. One Pipeline will be an assembly of several Function Blocks arranged in a single element where the inputs and outputs are mapped on one BIGG Business case.

At this stage, this additional step of Pipeline Development is initiated and the associated results will be presented in the [D6.2: Detailed description of pilots technical assets: ICT tools and accessibility to data sets](#) which will focus on the implementation of the AITB to the different use cases of the BIGG project.

Today the development of these pipelines is done using Jupyter Notebooks and R Notebooks. Each notebook integrates Function Blocks from the BIGG AITB but also other items from existing libraries identified in the section [Identified Existing libraries](#).

Eventually, the final version of the toolbox will present these Pipelines as individual Function Blocks that can be used individually and separately without any needs for additional development.

CONCLUSION

A year after the BIGG project has started, the development stage of the AI toolbox is well aligned with the expected schedule. The necessary storage needs (Task 5.1), analytical tools (task 5.2) and AI/ML modules (Task 5.3) were identified, described, adapted from existing means and libraries when possible or developed when needed.

Connecting with the objectives defined for the WP5 and specifically the objectives for the development of the AI Toolbox, the current status of development is a toolset based on itemized Function Blocks fully operational that can be assembled to match the specific needs of each business case. The Function Blocks were developed collaboratively on an opensource software tool and can be accessed by everyone and implemented for various uses.

The development phase was very dedicated to solving the specific needs of the BIGG business cases. The development followed a bottom up approach where the definition of needs was done based on each BC understanding and specifically focused on providing answers to the challenges presented by these BC. Each Function Block identified in this early problem definition phase was then described with three main parameters : the Inputs, the Function and the outputs. The preliminary toolbox is composed of this list of Function Blocks. Although the preliminary toolbox development was based on addressing specifically the challenges of BIGG, the final product allows the use of the Function Blocks for different use cases.

The task 5.4 will lead to the final version of the AI toolbox which will be an assembly of results from the task 5.1, 5.2 and 5.3. There is still considerable work to be done to first put this initial version of the AI toolbox to test and then to assemble all the Function Blocks into Pipelines. The final version of the toolbox is expected to feature these pipelines and as such they will need to be packaged, inputs and outputs defined at pipeline level and Function Block parameters set to optimal values to best address each BC.

This Pipeline creation has already been initiated and early results will be demonstrated in the context of WP6. The initial version of the toolbox was designed with a focus on maximizing the reuse of blocks across BCs. It is anticipated that this focus will be highly valuable in the immediate next step and the work to be done on task 5.4.

