Project no. 691735

REPLICATE PROJECT

Renaissance of Places with Innovative Citizenship And Technology

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

REPLICATE PROJECT

REnaissance of PLaces with Innovative Citizenship And Technology

Project no. 691735

H2020-SCC-2015 Smart Cities and Communities

Innovation Action (IA)

D3.4 Demand Side Platform

Due date of deliverable: 31/01/2021

Actual submission date: 31/01/2021

Start date of project: 01/02/2016 Duration: 60 months

Organization name of lead contractor for this deliverable: TECNALIA

Status (*Draft/Proposal/Accepted/Submitted*): *Submitted*

Project co-funded by the European Commission within the 7th Framework Programme		
Dissemination Level		
PU	Public	X
CO	Confidential, only for members of the consortium (including the Commission Services)	

Editor/Lead beneficiary :	TECNALIA
Internall reviewed by :	NEC and Fomento San Sebastian

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

Index of contents

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

1. EXECUTIVE SUMMARY

The current deliverable covers the implementation and results of the REPLICATE Demand Side Platform (DSP) implemented in the TXOMIN neighbourhood. The implementation of the DSP relies in the data collected by the existing District Heating (DH) SCADA. The District Heating has been developed by Fomento San Sebastian under the Replicate project framework to cover the heating and domestic hot water demand of the neighbourhood being a reference project in the region.

The DSP implementation process had to deal not only with the algorithm development, the integration with the DH SCADA platform and exploitation ready deployment design have been relevant tasks too.

The integration activities analyse the considerations taken in order to help to a generalization and scalability of the implemented algorithms. Generalization and scalability are key factors in seamless replication and transition to exploitation of any IT development. At the end of the document the basics to replicate DSP deployments are described.

The REPLICATE DSP implementation is divided in four main packages, Replicator, Machine Learning (ML), Data-Server and Database. The Machine Learning package (ML package) includes the outcomes of the algorithm in different periods of the year as well as the rationale and theoretical background in each of the methods implemented. The rest of the packages are auxiliary packages that while not being the core of the development are necessary to create a full functional platform.

The REPLICATE DSP is composed by three major functionalities: apartments' energy consumption forecast, HVAC-DHW consumption split and day ahead consumption forecast. These functionalities have been divided in two service groups depending on the targeted user, services for energy users and services for energy producers. Throughout the document, for a better understanding of the background of the implementations, used methods and machine learning techniques are introduced. In order to ease reading and the current document not being a scientific paper, cumbersome mathematical formulations are not included.

One of the final goals of the REPLICATE DSP is to contribute to apartment owners' awareness about their energy usage and potential savings. That ambitious target is addressed with the development of a user-friendly graphical interface that goes beyond the REPLICATE project and it is integrated in the District Heating management platform. The other main objective of the SSP is to provide relevant information to the energy provider in order to accommodate the energy production to the real day ahead needs.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

2. REPLICATE

The REPLICATE project will generate smart city business models, and tailor-made solutions in the areas of energy, transport and ICT starting from the districts: Urumea Riveside (San Sebastián), Novoli (Florence) and Ashley, Easton and Lawrence Hill Neighbourhood (Bristol). In summary there will be pilot actions in energy efficiency, efficient and sustainable transport and integrated infrastructures, being the latter the key elements for the integration and development of cross-sectorial solutions. Three follower cities participate in the project: Essen (Germany), Nilufer (Tutkey) and Lausanne (Switzerland).

Being a demonstration project, the main concept that is on the top of the project is REPLICABILITY: it will be necessary that the project results could be applicable throughout the lighthouse cities and in other cities which want to evolve towards the 'smart city' concept, and could grow of scale too. To assure the large-scale deployment of innovative technologies successfully demonstrated in the lighthouse districts specific studies will be necessary for each of the demonstrated solutions to ensure that they are scalable and can be replicated.

Prior to REPLICATE project San Sebastian, Florence and Bristol have already collaborated in a STEEP project (Systems Thinking for Comprehensive City Efficient Energy Planning) which have allowed to the cities generate Smart City Plans. STEEP project has defined a collaborative and participatory methodology to reach the objective of defining an Action Plan for particular districts of each city.

The main objective of REPLICATE project is the development and validation in three lighthouse cities (San Sebastián – Spain, Florence – Italy and Bristol – UK) of a sustainable City Business Model to enhance the transition process to a smart city in the areas of the energy efficiency, sustainable mobility and ICT/Infrastructure, in order to accelerate the deployment of innovative technologies, organisational and economic solutions to significantly increase resource and energy efficiency, improve the sustainability of urban transport and drastically reduce greenhouse gas emissions in urban areas.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

3. INTRODUCTION

3.1 Relation to Other Project Documents

The development of the REPLICATE DSP is an atomic activity in the sense of that it has no dependency on other developments committed in the project. This limited interaction with other developments minimizes the relation with documents developed in other work packages or tasks.
The documents taken into consideration in the RESPLICATE DSP development are listed in the table below.

Ref.	Title	Description
REPLICATE Grant Agreement signed 240713.pdf	Grant Agreement	Grant Agreement no. 691735
DoA REPLICATE (691735)	REPLICATE Annex 1 – DoA to the GA	Description of the Action
REPLICATE Consortium agreement signed December 2015 (7th December version)	Consortium Agreement	REPLICATE project – Consortium Agreement
REPLICATE Project Management Plan	D1.1 Project Management Plan (v.1) (29/04/2016)	REPLICATE Project Management Plan
REPLICATE District Management Plans	D1.4 District Management Plan San Sebastian D1.5 District Management Plan Florence D1.6 District Management Plan Bristol	REPLICATE District Management Plans
REPLICATE Communication Plan	D11.1 Communication Plan	REPLICATE Communication Plan

Table 1: Projects level documents

These will also be stored on the shared online platform.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

3.2 Abbreviations list

The table below compiles the list of acronyms and abbreviations used in the current document.

GA	Grant Agreement
CA	Consortium Agreement
DoA	Annex I–Description of the Action
EC	European Commission
H2020	Horizon 2020
PC	Project Coordinator
PL	Pilot Leader
PMP	Project Management Plan
TC	Technical Coordinator
WP	Work Package
WPL	Work Package Leader
EMS	Energy Management System
SCADA	Supervisory Control And Data Acquisition
DH	District Heating
DTW	Dynamic Time Warping
HVAC	Heating – Ventilation – Air conditioning
DHW	Domestic Hot Water
SSE	Sum of Squared Errors
IFC	Industrial Foundation Classes
DSP	Demand Side Platform
ML	Machine Learning

Table 2: Abbreviations list

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

4. Integration Platform

The District Heating management platform designed by Fomento San Sebastian (FSS) is based on a complex coupled distributed architecture. The flexibility, scalability as well as the minimum interference among the existing platform components were key factors considered during the design phase. Conceptually two main modules can be identified.

- **Monitoring platform**: Implemented and deployed in the DH SCADA includes all the software and hardware items that collect data from apartments. This platform is the system for the DH control and monitoring of the DH. It is a tool for the energy management of the DH building and the substations of the central buildings that allows providing the qualified energy service to the entire neighbourhood.
- **DSP platform**: Deployed in Tecnalia's DMZ includes all the software modules that target to benchmark Txomin neighborhood apartments behavior and forecast day ahead consumption.

4.1 DSP Architecture Design

The REPLICATE DSP platform is divided in four mayor modules:

- Replicator package: The Replicator module queries the remote SCADA on a daily basis to check if data from new apartments have been uploaded and, in that case, registers them in the local database and configures their polling process.
- ML package: The ML package compiles the set of software bundles that calculate the apartment energy performance as well as the ones that participate in the day ahead predictions calculation.
- Data-Server package: The data-server package covers two functionalities. On the one hand it collects data from the DH platform and rearranges it is a way in which is valid for the algorithms implemented in the ML package. On the other hand, the data-server implements interfaces to allow third parties to read outcome of the ML package algorithms.
- Data Base: Set of web-services and engine used to store data model, ML package outcomes and additional information no available in the Ferrovial SCADA. WS02 and MySQL have been selected to implement data base reading and writing operations and data storage engine. The WSO2 is an open-source framework that facilitates the creation services to read and write from different data sources.

The figure below describes dependencies and relationships among the modules introduced above.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

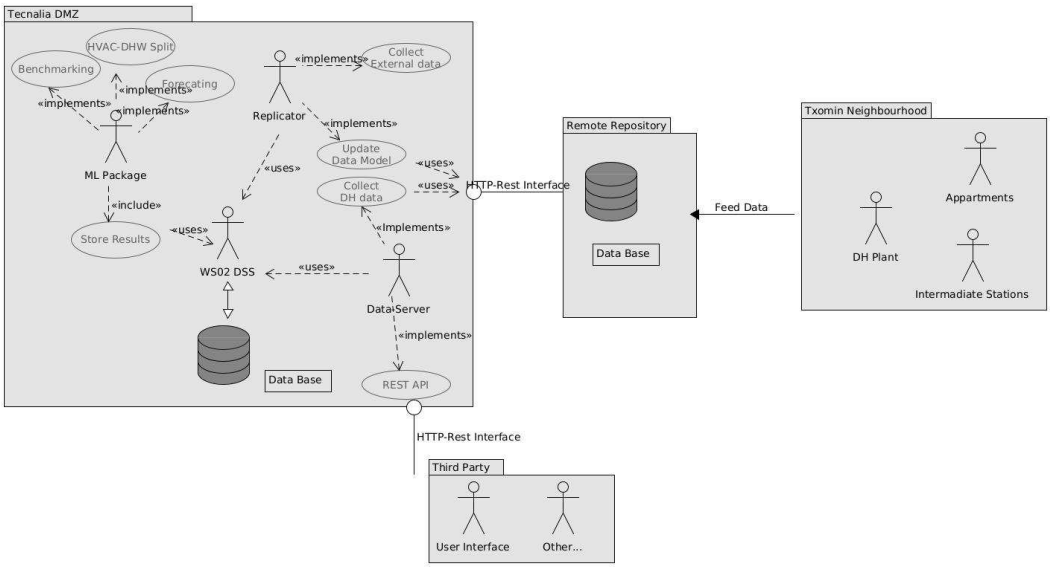


Figure 1:TECNALIA DSP Architecture

4.2 Data Model

The "Holy Grail" for the field of information technology in the architecture, engineering, construction (AEC) and facilities management (FM) industries has been information integration throughout the suite of computer tools used to carry out engineering projects. Vendor-specific approaches have been developed but these are limited in their coverage of potential applications and users.

Industry Alliance for Interoperability (IAI) is a global, industry-based consortium for the AEC/FM industry (IAI 1996, IAI 1998), IAI's goals are to define, publish and promote a specification--called the Industry Foundation Classes (IFCs)--for sharing data throughout the project lifecycle, globally, across disciplines and technical applications (IAI 1998). The IFCs are used to assemble a project model in a neutral computer language that describes building project objects and represents information requirements.

The effectiveness of a data model can be measured considering the range of use cases that it covers keeping an acceptable complexity level. The REPLICATE project has taken the IFC standard as reference to implement the data model in which the DSP will rely.

The transition from data model to database structures may result difficult and sometimes lead to a confusion between the applicationn's data model and the database layout. The database design followed in the REPLICATE project takes as reference the Object Oriented Programing (OOP) concepts described in the IFC standard. Following this approach all the data model entities are considered "objects" with some "attributes", hierarchy among "objects" implemented

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

as "relationships" provides to the data model "relational" structure expected in databases.

In order to create a scalable and flexible hierarchy two abstract entities have been created. Abstract entities can't be instantiated. They are the reference for the rest of the object types. In other words, abstract entities are the reference points to implement further objects and make the data model grow. These are the abstract objects in REPLICATE data model.

- **Object:** Abstraction for every entity in the database.
- **Sensor:** Refers to every single measurement point (magnitude) available no matter it comes from the SCADA or external data source

REPLICATE project identified the following entities as pillar concepts for the data model development.

- **Power Plant**: Refers to the DH plant that provides the heating power.
- **Power Station**: Refers to DH and apartment intermediate metering points
- **Apartment**: Refers to every single apartment
- **Meter**: Refers to the device that makes the thermal readings for each apartment

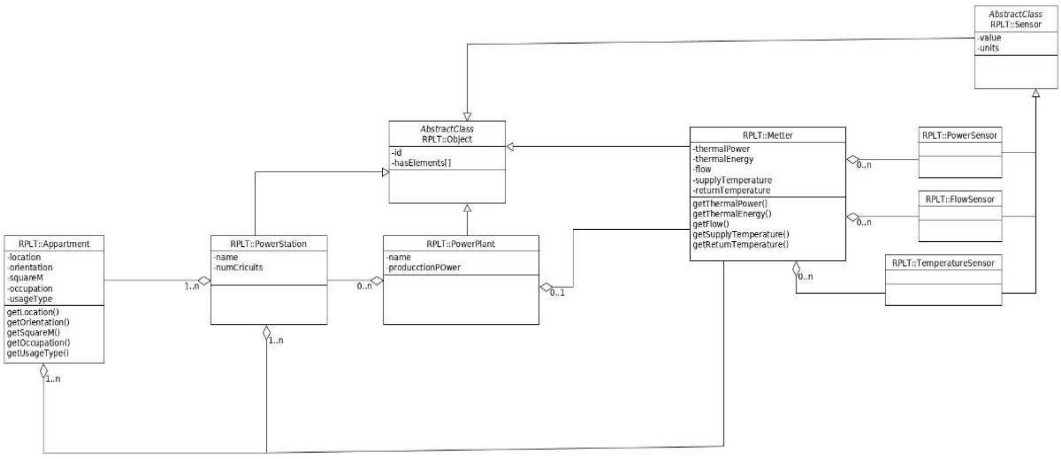


Figure 2: OOP design implemented in REPLICATE

The REPLICATE data model is not only designed to conceptually describe the application domain but to support ML algorithms execution input and output requirements. The modelling of inputs as SENSOR derived entities is straight forward, the extension of the concept to the outputs has lead to define the following entities as output sensors to store the results of the algorithms.

- XXXX_HVAC_DHW: Label identifies, modelled as sensor, the consumption type for the apartment XXXX in a given period of 5 minutes.
- XXXX_WEEK_CL_M2: Label identifies the cluster in which is grouped apartment XXXX after normalizing the energy consumption using the area of the apartment.
- XXXX_WEEK_CL: Label identifies the cluster in which is grouped apartment XXXX without any

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

normalization factor

- XXXX_SC_FORECAST_KWH: Label identifies forecast consumption for the sub-central XXXX

As already mentioned, it has to be clearly distinguished the data model used by the algorithms and the way in which this data model is stored in the data base. The IFC specification defines mainly objects and a hierarchy of objects, following this concept, the REPLICATE DSP data base design makes abstraction of the OOP design of the data model. In order to explore all the possibilities offered by the IFC standard, entities like "geometry" and "representation", which are not used in the DSP data model, have been included in the database design.

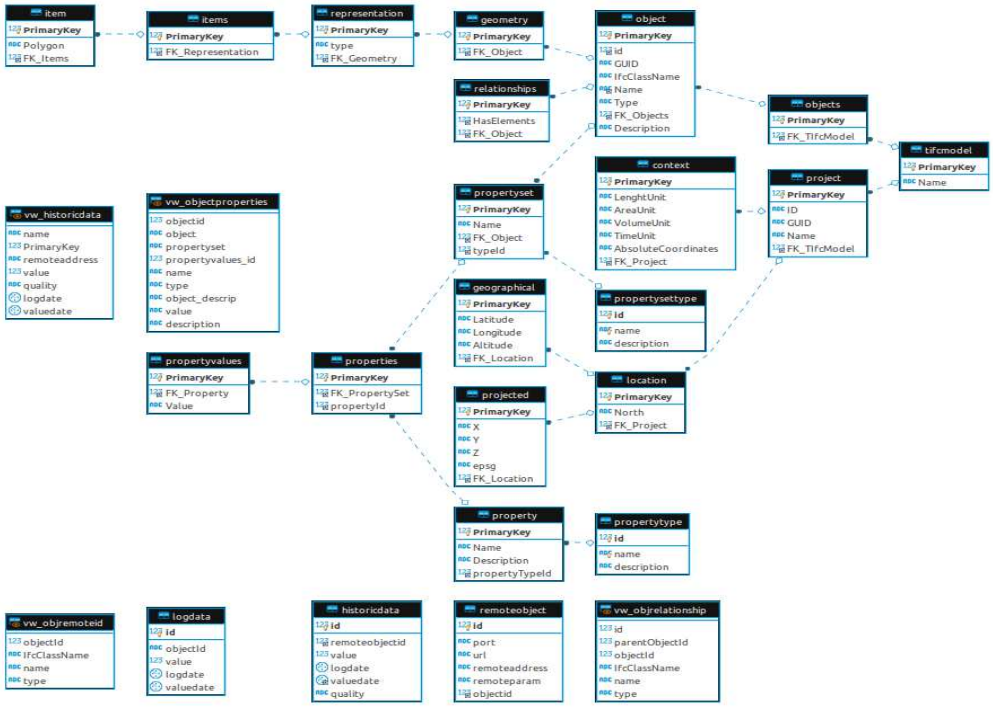


Figure 3: Data base tables and relationships

4.3 Machine Learning Methods and Implementation

The implementation in the REPLICATE DSP can be considered as paradigms of the most popular ML areas:

- **Unsupervised learning**: Mostly used in classification problems, the challenge in the implementation of an unsupervised classification algorithm is to be able to identify the "hidden" patterns in a given dataset. In unsupervised learning, the training set consists of unlabelled inputs, that is, of inputs without any assigned desired output. The formal

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

study of methods and algorithms for grouping, or clustering, objects according to similarities and same characteristics is cluster analysis. It does not use category labels that tag objects with prior identifiers, i.e., class labels. Data clustering distinguishes by the absence of category information. K-means is the most popular and simple clustering algorithm. This Algorithm was published in 1955.

K-means is an unsupervised classification (clustering) algorithm that groups objects into k groups based on their characteristics. The grouping is done by minimizing the sum of distances between each object and the centroid of its group or cluster. Quadratic distance is often used. Time series (as daily consumption) unsupervised classification requires some special consideration as the features to classify (periodic values) have some dependency among them. The implemented consumption benchmarking algorithm used the Dynamic Time Warping (DTW). DTW is a well-known shape-based similarity measure for time series data. Unlike the Euclidean distance function, dynamic time warping breaks the limitation of one-to-one alignment, eventually supports non-equal-length time series. It uses dynamic programming technique to find all possible paths and selects the one that yields a minimum distance between the two time series using a distance matrix, where each element in the matrix is a cumulative distance of the minimum of the three surrounding neighbours.

The figure below describes the difference among the Euclidean matching and the DTW matching. In a short way it is possible to say that DTW metric, in contrast to the Euclidean metric, can consider matching points which are shifted in time.

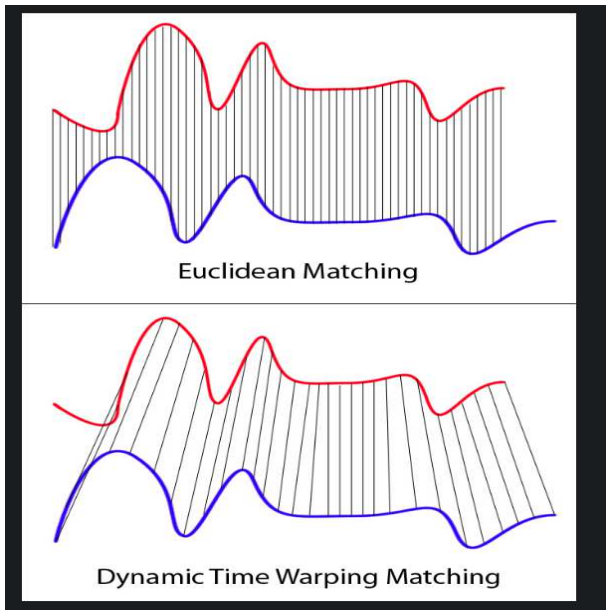


Figure 4: k-Means; Difference between the Euclidean matching and the DTW matching

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

The most relevant considerations to be taken when DTW algorithms are applied are:

- **Step Function**: When there are no differences between the time-series, the warping path coincides with the "diagonal", but as differences between time-series increase, the warping path deviates more from the diagonal line by matching time-axis fluctuations. While DTW finds the optimal alignment of the time-series, sometimes it tends to create an unrealistic correspondence between time-series features by aligning very short features from the one of the series to the long features on the second time-series. In order to avoid such a phenomena, the warping path is subject to constraints on the each step. This constraint implemented as the possible relations between several consecutive points on the warping path.
- **Weighting**: Previously defined measure of distance between time series as the cost a function which essentially is a summation of pairwise distances between corresponding points at time-series X and Y. By adding the weights to the each of the distances based on the step direction we could penalize or favour certain types of point-to point correspondence.

Global Constraints: The computational cost of DTW algorithm applied to time series of N,M length is O(NM) and algorithm requires a storage for two matrices of the size $N \times M$. In order to improve the computational cost and optimize the DTW sensitivity similarly to the step function constraints global constraints must be introduced. Limited wrapping range and diagonal length are the most applied global constraints.

While partitional clustering methods (i.e. K-means) have received more attention in recent literature, hierarchical clustering algorithms represent the traditional choice for performing document clustering, since text collections often contain broad themes that may be naturally sub-divided into more specific topics. Hierarchical algorithms are generally organised into two distinct categories:

- Agglomerative: Begin with each object assigned to a singleton cluster. Apply a bottom-up strategy where, at each step, the most similar pair of clusters are merged.
- Divisive: Begin with a single cluster containing all n objects. Apply a top-down strategy where, at each step, a chosen cluster is split into two sub-cluster

In last years, the increase of the computational power lead to the implementation of new clustering techniques based in kernel methods or spectral clustering.

- Kernel methods involve the transformation of a dataset to a new, possibly high-dimensional space where non-linear relationships between objects maybe more easily identified.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

- Spectral clustering, motivated by work in graph theory, unsupervised feature extraction methods have been developed that employ well-known techniques from linear algebra to analyse the spectral properties of a graph representing a dataset. In practice, this involves constructing a reduced dimensional space existing clustering algorithm may subsequently be applied in the reduced space to uncover the underlying classes in the data.

- **Supervised learning**: In supervised learning, the training set consists of pairs of input and desired output, and the goal is that of learning a mapping between input and output spaces. The learning process in a simple machine learning model is divided into two steps: training and testing. In training process, samples in training data are taken as input in which features are learned by learning algorithm or learner and build the learning model. In the testing process, learning model uses the execution engine to make the prediction for the test or production data. Tagged data is the output of learning model which gives the final prediction or classified data. Most commonly, supervised learning leaves the probability for input undefined, such as an input where the expected output is known. This process provides dataset consisting of features and labels. The main task is to construct an estimator able to predict the label of an object given by the set of features. Then, the learning algorithm receives a set of features as inputs along with the correct outputs and it learns by comparing its actual output with corrected outputs to find errors. It then modifies the model accordingly. The supervised learning tasks are divided into two categories: classification and regression. In classification, the label is discrete, while in regression, the label is continuous.

  Some of the most well-known and traditional supervised learning algorithm are listed below:

  - **Decision tree**: Represents a classifier expressed as a recursive partition of the instance space. The decision tree consists of nodes that form so called root tree, which means that it is a distributed tree with a basic node called root with no incoming edges. All of the other nodes have exactly one incoming edge. The node that has outgoing edges is called internal node or a test node. The rest of the nodes are called leaves. In a decision tree, each test node splits the instance space into two or more sub-spaces according to a certain discrete function of the input values. In the simplest case, each test considers a single attribute, such that the instance space is portioned according to the attribute's value. In case of numeric attributes, the condition refers to a range.
  - **Linear Regression**: The goal of the linear regression, as a part of the family of regression algorithms, is to find relationships and dependencies between

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

variables. It represents a modelling relationship between a continuous scalar and one or more (a D-dimensional vector) explanatory variables (also independent variables, input variables, features, observed data, observations, attributes, dimensions, data point, etc.) denoted X using a linear function.

- **Bayesian**: The Bayesian classification is another method of the supervised learning methods as well as the statistical method for classification. Assumes an underlying probabilistic model and it allows capturing uncertainly about the model in a principled way by determining probabilities of the outcomes. The basic purpose of the Bayesian classification is that it can solve predictive problems

The work done in the REPLICATE DSP implementation uses the two "flavours" of the ML techniques mentioned above. The benchmark between apartments consumption and consumption type discrimination, indeed classification problems, rely in non-supervised methods, on the other hand, the day ahead consumption forecast is handled implemented a predictive model based on supervised regression methods. Each of the methods applied and implementations are described in detail in the corresponding chapter.

The REPLICATE DSP development has covered almost 12 moths of activity. Once data gathering process was implemented the off-line validation phase started. During the off-line validation the outcomes of the algorithms were validated, this phase lasted till a trustful maturity of the implementation was achieved. The off-line validation milestone concluded, the design and implementation of deployable platform started.

Figure 5: Txomin neighborhood in San Sebastián - Donosti

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

The DSP validation has been piloted in the Txomin neighbourhood (San Sebastian – Spain). The neighbourhood is composed by new and refurbished dwellings and is heated (HVAC and DHW) by a district heating plant (DH). The Txomin neighbourhood additionally implements several actions in the energy efficiency, sustainable mobility and ICT/infrastructure fields (retrofitted consolidated homes, electric buses to connect the district with the city centre, high seed connectivity network for public services, a smart street lighting system, etc.).

Following the architecture and data model described in previous chapters, data for 734 apartments and subcentrals from the Txomin neighbourhood have been considered.

The measurements provided by the DH's SCADA for each of the apartments/subcentral (Thermal Energy, Thermal Power, Water Flow, Supply Temperature, Return Temperature) are complemented with the outdoor temperature, apartments area and occupant's information.

From the complete set of apartments and subcentral available at the development of the current document only some of them have been selected as reference to show the outcomes of the REPLICATE DSP:

Apartment ID
5043
5050
5054
5056
5061
5084
5099
5112
Sub-Central ID
5121
5170
7812
11335

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

5. SERVICE FOR ENERGY USERS.

5.1 Group Apartments Consumption Pattern

5.1.1 Description

The apartment energy consumption-based clustering service targets two objectives

- Benchmark consumption pattern: The clustering process will identify the most relevant consumption patterns within the day and for each apartment will deliver its consumption ratio regarding to the rest of the apartments
- Suggest effective consumption pattern: The selection of the effective consumption pattern will suggest for each of the apartments a tentative profile that reduces its energy consumption.

The objectives introduced above are linked since the consumption pattern detection is a preliminary step for later suggest an effective consumption pattern. The benchmark consumption pattern is based on unsupervised machine learning concepts.

5.1.2 Architecture Design

The apartment benchmark flow is quite straight forward, basically is composed by three steps which two are implemented in the ML package and the third one is relative to the results storage.

The main input for the energy benchmark functionality is the apartment energy consumption, the area, other features as the apartment area or occupation are considered as entries for normalization or standardization operations. The result of the functionality are the groups of apartments that present similar consumption pattern.

The designed implementation allows to flexible configuration of the algorithm inputs. The dynamic data structures and normalization/standardization processes allow to use different input features (quantity and value ranges).

The list of features used during the REPLICATE DSP development:

Features
Energy Consumption
Apartment Area

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

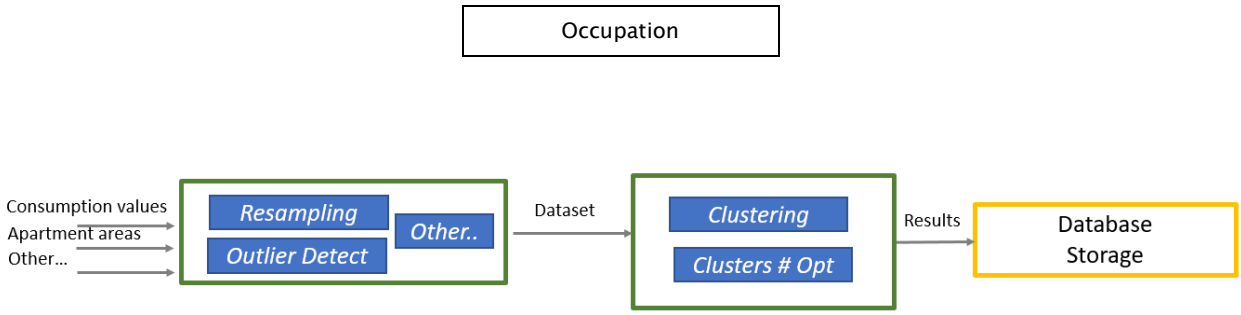


Figure 6: Benchmark flow

The first step of the process deal with the data cleaning and dataset preparation for the clustering operation. In this step the daily 5 minutes-based values are resampled to weekly 1 hour step values, outlier values are removed and suspicious values (i.e., constant values for a large period) are removed. The second step implements the clustering operation, but prior to it the optimum number of clusters to consider is determined. Finally, the results are stored in the data base.

5.1.3 Implementation and Results

The energy readings for the Txomin neighbourhood are composed by readings every 5minutes, so granular dataset in which most to the values are not significative (in most cases the energy consumption in two consecutive periods is cero) may lead to inconsistent outcomes. In order to tackle this issue, the energy benchmarking model implemented for Txomin neighbourhood has analysed different aggregations periods for the 24h times series. From the inspected options weekly periods of hourly aggregated data has been selected as most representative grouping pattern.

During the evaluation period in order to benchmark each apartment energy consumption the following data has been produced:

- Overall consumption boundaries.
- Dominant consumption patterns detection

In order to conceptualize the obtained outcomes some representative apartments and days have been selected.

- Date 2019-12-01: The amount of available valid data was for 108 apartments. From the analysed data, it can be extracted that the maximum energy consumption in at least one apartment raised up to 70kWh, but almost the 95% of the apartments hardly consumed

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

the half of it.

Taking into consideration the list of selected apartments the cluster groups show that 4 of them fall in the same group (5053,5056,5084,5099) two of them in another group (5050,5061) and finally the 5054 is grouped it alone.

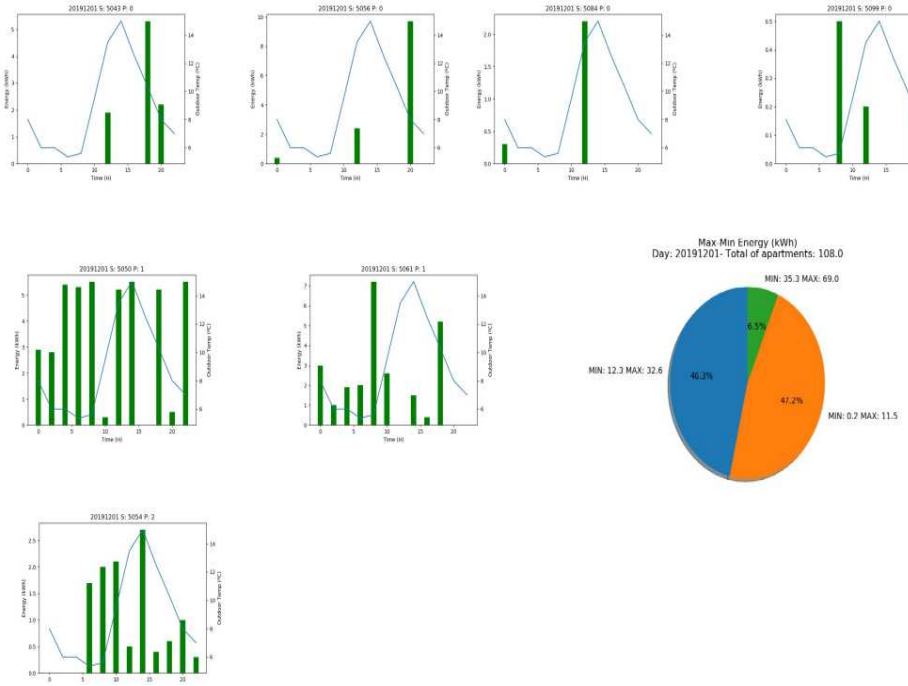


Figure 7: Grouping of Apartments in Terms of Energy Consumption for day 2019-12-01

The benchmarking rank delivers the following information:

APARTMENT	GROUP	ENERGY (kWh)	RELATIVE (%)	ABSOLUTE (%)
5043	0	9.40	81.74	13.62
5050	1	44.10	63.91	63.91
5056	0	12.50	38.34	18.12
5061	1	24.80	76.07	35.94
5099	0	0.90	7.83	1.30

Table 3: Apartments Consumption Information and Clustering Results for day 2019-12-01

Note: The RELATIVE (%) column is calculated among apartments in the same group, GLOBAL (%) is calculated among the whole list of apartments

- Date 2019-12-06: The amount of available valid data was for 151 apartments. From the

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

analyzed data it can be extracted that the maximum energy consumption was around 58 kWh but as for the previous case, almost the 89% of the apartments consumed in worst case only the half of that. Taking into consideration the list of selected apartments, more homogeneous clusters are reached with groups containing 2-2-4 items respectively. The grouping shows how after standardization from those that present scattered behaviour to those that present more homogeneous behaviour.

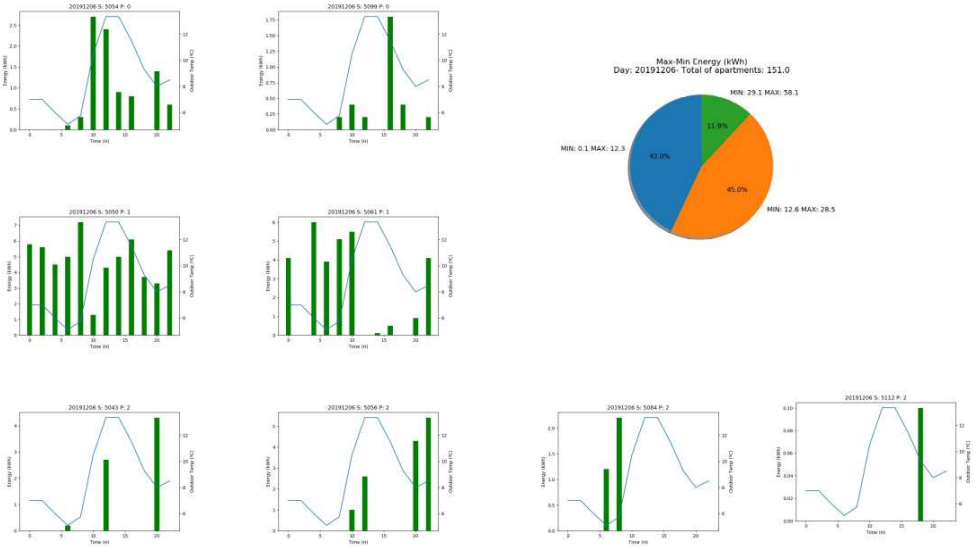


Figure 8: Grouping of Apartments in Terms of Energy Consumption for day 2019-12-06

The benchmarking rank delivers the following information:

APARTMENT	GROUP	ENERGY (kWh)	RELATIVE (%)	ABSOLUTE (%)
5043	0	7.20	58.54	12.39
5050	2	57.20	98.45	98.45
5056	1	13.30	46.67	22.89
5061	2	30.20	51.98	51.98
5099	0	3.20	26.02	5.51
5112	0	0.10	0.81	0.17

Table 4: Apartments Consumption Information and Clustering Results for day 2019-12-06

- Date 2019-12-27: The amount of available valid data was for 393 apartments. From the

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

analysed data it can be extracted that the maximum energy consumption was around 58 kWh but as for the previous case, almost the 84% of the apartments consumed in worst cast the half. Taking into consideration the list of selected apartments, more homogeneous clusters are reached with groups containing 2-2-4 items respectively. The grouping shows how after standardization from those that present scattered behaviour to those that present more homogeneous behaviour.

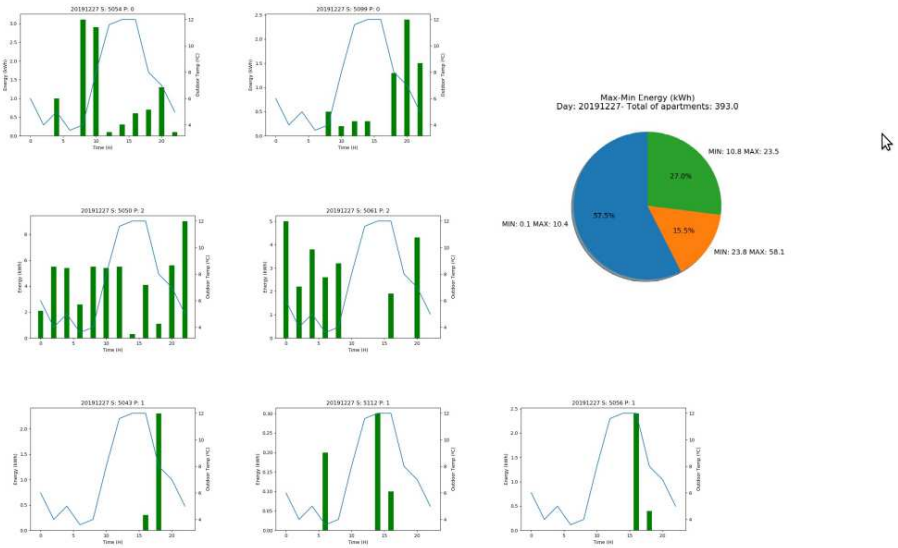


Figure 9: Grouping of Apartments in Terms of Energy Consumption for day 2019-12-27

The benchmarking rank delivers the following information:

APARTMENT	GROUP	ENERGY (kWh)	RELATIVE (%)	ABSOLUTE (%)
5043	0	2.60	25.00	4.48
5050	1	52.10	89.67	89.67
5056	0	2.80	26.92	4.82
5061	2	23.00	97.87	39.59
5099	0	6.50	62.50	11.19
5112	0	0.60	5.77	1.03

Table 5: Apartments Consumption Information and Clustering Results for day 2019-12-27

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

- Date 2020-01-10: The amount of available valid data was for 439 apartments. From the analysed data it can be extracted that the maximum energy consumption was around 70 kWh but as for the previous case, almost the 86% of the apartments

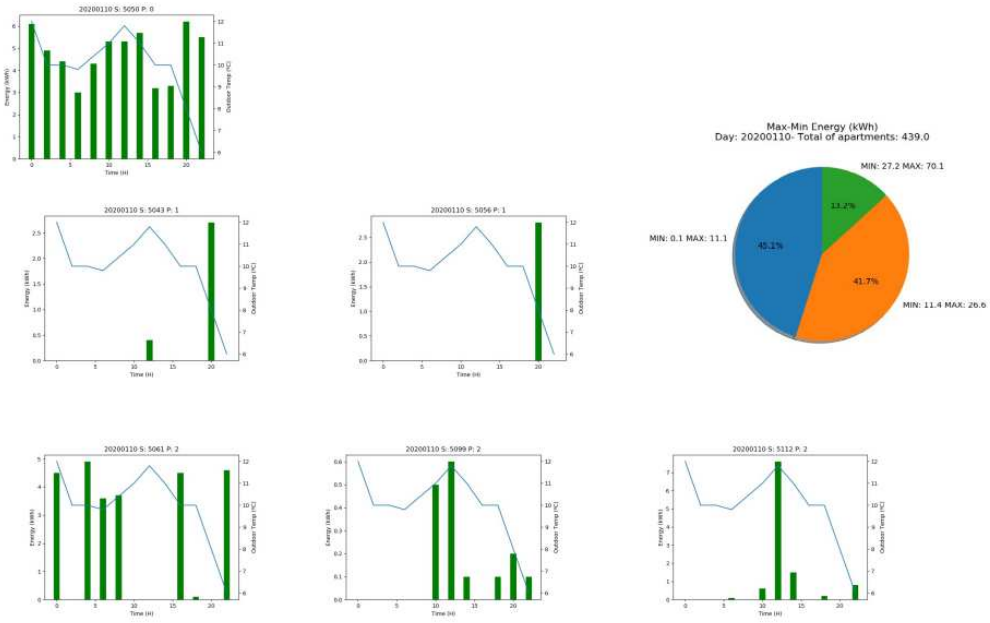


Figure 10: Grouping of Apartments in Terms of Energy Consumption for day 2020-01-10

consumed in worst case the half. Taking into consideration the list of selected apartments, in this case, a kind of erratic behaviour is shown due to the misclassification of apartment not showing scattered or of full-day energy consumption patterns.

The benchmarking rank delivers the following information:

APARTMENT	GROUP	ENERGY (kWh)	RELATIVE (%)	ABSOLUTE (%)
5043	1	3.10	27.93	4.42
5050	0	57.20	81.60	81.60

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

5056	1	2.80	25.23	3.99
5061	2	25.90	97.37	36.95
5099	2	1.60	14.41	2.28
5112	2	10.80	97.30	15.41

Table 6: Apartments Consumption Information and Clustering Results for day 2020-01-10

- Date 2020-01-15 : The amount of available valid data was for 143 apartments. From the analysed data it can be extracted that the maximum energy consumption was around 58 kWh but as for the previous case, almost the 80% of the apartments consumed in worst cast the half.

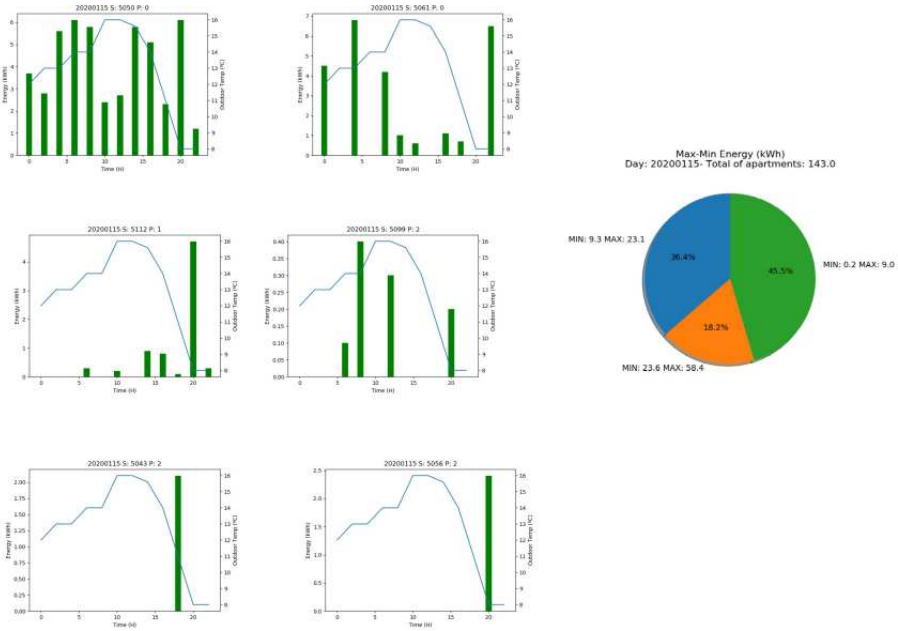


Figure 11: Grouping of Apartments in Terms of Energy Consumption for day 2020-01-15

The benchmarking rank delivers the following information:

APARTMENT	GROUP	ENERGY (kWh)	RELATIVE (%)	ABSOLUTE (%)
5043	2	2.10	23.33	3.60
5050	0	49.60	84.93	84.93

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

5056	2	2.40	26.67	4.11
5061	0	25.40	43.49	43.49
5099	1	1.00	11.11	1.71
5112	1	7.30	81.11	12.50

Table 7: Apartments Consumption Information and Clustering Results for day 2020-01-15

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

5.2 HVAC & DHW Consumption Split

5.2.1 Description

The objective of the functionality is to provide to the apartment owner accurate information about the consumption in HVAC (heating) and DHW (Domestic Hot Water).

The thermal heat delivery metering deployed at Txomin neighbourhood is not able to split between HVAC or DHW thermal consumption. In this context, the used unsupervised classification is the machine learning technique, that as said before, tries to find "hidden" data patterns. This technique has been applied to discriminate between both type of consumptions.

In contrast with benchmarking functionality, which was to deal with time series, the HVAC-DHW split functionality deals with static data. With no time dependency in the training features, K-Means algorithm application with Euclidean distance has been selected to implement the current functionality.

Among the set of measures retrieved from the SCADA, the features selected to model the HVAC-DHW split are:

- **Event duration**: The duration in minutes of periods in which the water flow is not 0. It is assumed that such event implies thermal consumption in the apartment.
- **Water Flow**: Refers to the flow in the meter given in m3/h
- **Temperature Drop**: Refers to the supply-return temperature drop in meter during the event duration

Distance algorithms like K-Means, KNN or SVM are most affected by the range of features. This is because behind the scenes, they are using distances between data points to determine their similarity. Since both the features have different scales, there is a chance that higher weightage is given to features with higher magnitude. This will impact the performance of the machine learning algorithm and obviously, we do not want our algorithm to be biased towards one feature.

Formally, standardization is a scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

The K-Means algorithm works iteratively to assign each "point" (the rows of our input set form a coordinate) one of the "K" groups based on their characteristics. They are grouped based on the similarity of their features (the columns).

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

The "centroids" of each group will be "coordinates" of each of the K sets that will be used to label new samples.

Labels for the training dataset. Each tag belonging to one of the K groups formed.

When using k-means clustering, it is necessary some way to determine whether they are using the right number of clusters.

One method to validate the number of clusters is the Elbow Method. The idea of the Elbow Method is to run k-means clustering on the dataset for a range of values of k (say, k from 1 to 10) and for each value of k calculate the sum of squared errors (SSE). The SSE tends to decrease toward 0 as we increase k. The goal will be to choose a small value of k that still has a low SSE. The elbow method usually represents the value where we start to have diminished returns by increasing k.

The figure below describes the typical shape of the Elbow Method applied to a dataset. In the example below 3 would have been selected as optimal k value for clustering for being the value where the curve takes more concavity.

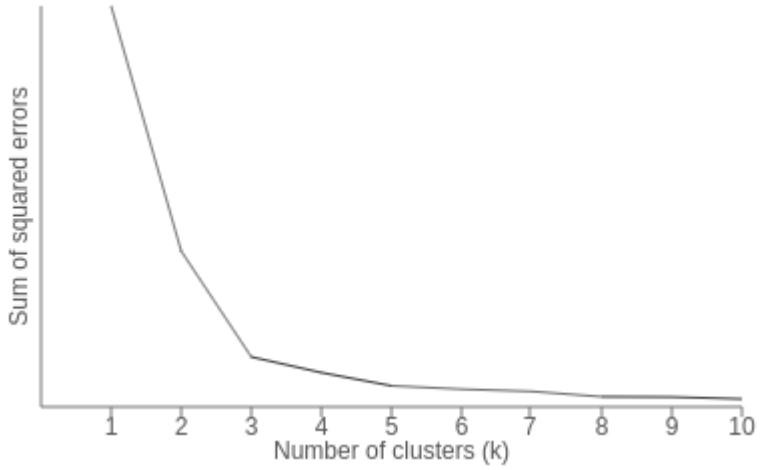


Figure 12: Typical shape of the Elbow Method

In our problem, we slightly differ from the Elbow approach described before. In the HVAC-DHW split case it is know that there are two labels (groups). The apartment is using the heating or the DHW, so it makes sense to force the algorithm to create two groups.

First, a study of the data has been carried out. Analyzing the behavior of the temperature drop, two trends can be observed. On the one hand, the measures that have been taken periodically over time, indicating that the system is switched on longer, take high values at first (due to the turning on of the system) but end up taking lower values (with green color in Figure 13). On the

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

other hand, high point values are also observed, which have a shorter duration than the previous ones, generally ranging from 5 to 15 minutes (with red color in Figure 13).

The developed algorithm would be capable of, using data analytics techniques, differentiate the HVAC system, measures that, with the system on for a longer time, draws a curve similar to an exponential decreasing curve; from the DHW point measurements, measurements with higher values and less system ignition time.

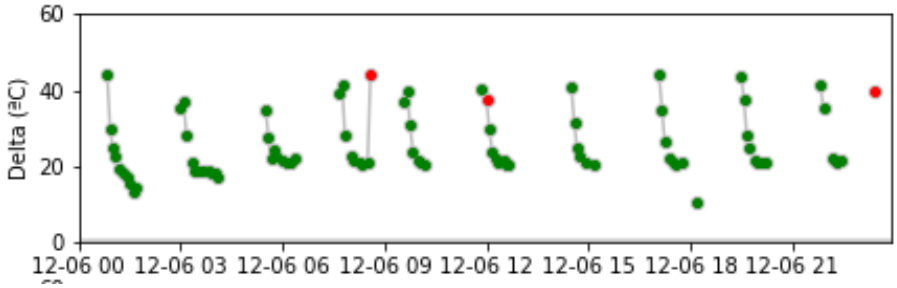


Figure 13: Measured Data Behaviour in 24 hours

5.2.2 Architecture Design

The HVAC-DHW consumption split functionality has a complex flow with dependencies with the benchmarking functionality. The HVAC-DHW split analysis takes the clustering information from the results calculated by the benchmarking operation. The goal for the HVAC-DHW split, as already described in other chapters, is just to determine which fraction of the consumption belongs to HVAC and which part belongs to the DHW. Nevertheless the studies and data analysis that can be concluded after this operation are tightly linked to the benchmarking results.

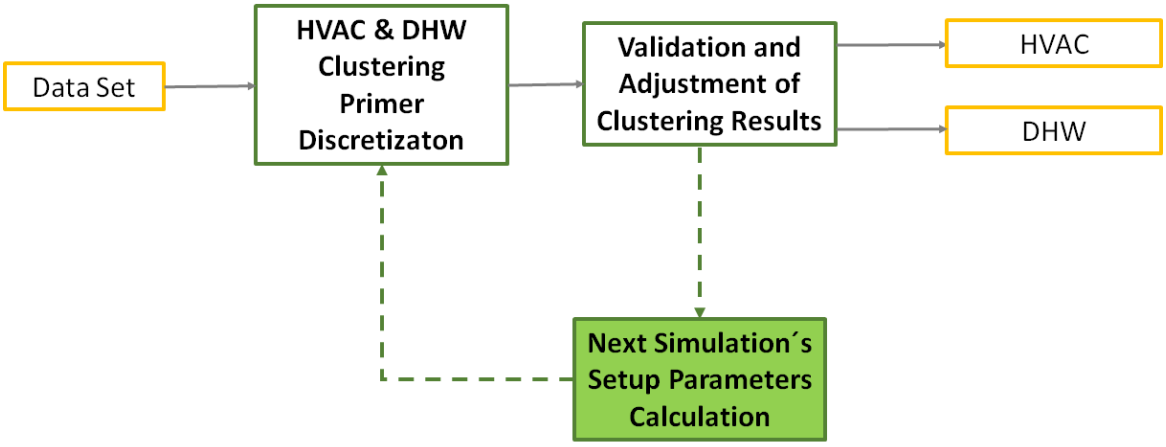


Figure 14: HVAC & DHW Consumption Split Data Architecture

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

Once the dataset is ready, the HVAC-DHW clustering is done, in this process to each period of 5 minutes the corresponding label is assigned. The flow, the supply and return temperatures measured by the meter are the inputs for that process. Once the labelling for each period is determined, the consumption groups derived from the benchmarking operation are considered and the analysis is extended. The benchmarking analysis is extended to include not only the overall consumption profile but to provide additional detailed information.

The HVAC-DHW split functionality is designed with a feedback loop in order to assimilate the seasonal (most relevant in winter or summer) fluctuations that flow and supply-return temperatures may have.

5.2.3 Implementation and Results

The implemented HVAC-DHW split functionality results, grouped by apartment, are described below. The apartments taken as reference are the same as for the benchmarking functionality and in order to summarize it is possible to say that two major behaviours have been detected.

- Slight overlapping groups:
- Hard overlapping groups.

Apartment 5043 & 5056:

The figure below describes the values obtained for apartments 5043 and 5056. In both cases there exists overlapping but in general the histograms describe relatively well-defined groups (HVAC and DHW). The lower flow and lower supply temperature drop would clearly represent the usage of HVAC (green), as the higher flow and temperature drop would describe DHW usage (red). It is important to notice that there are some HVAC system measures that also would have high flow and temperature drop due to system power on.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

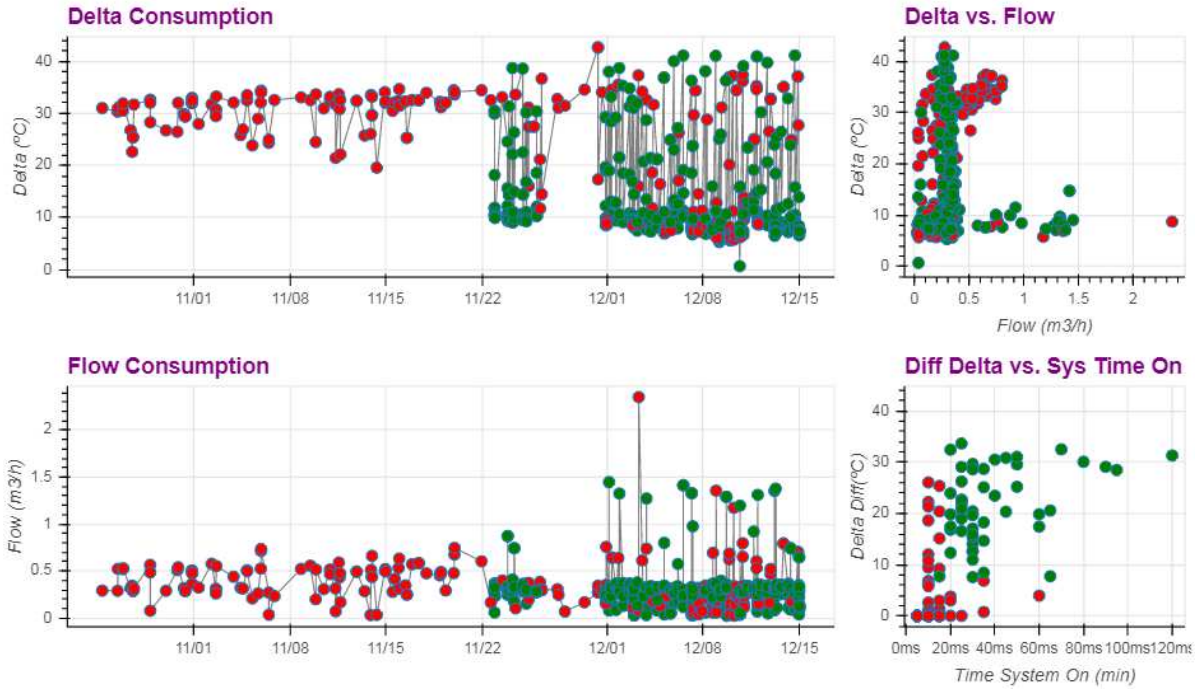


Figure 15: HVAC and DHW Clustering Results for 5043 Apartment

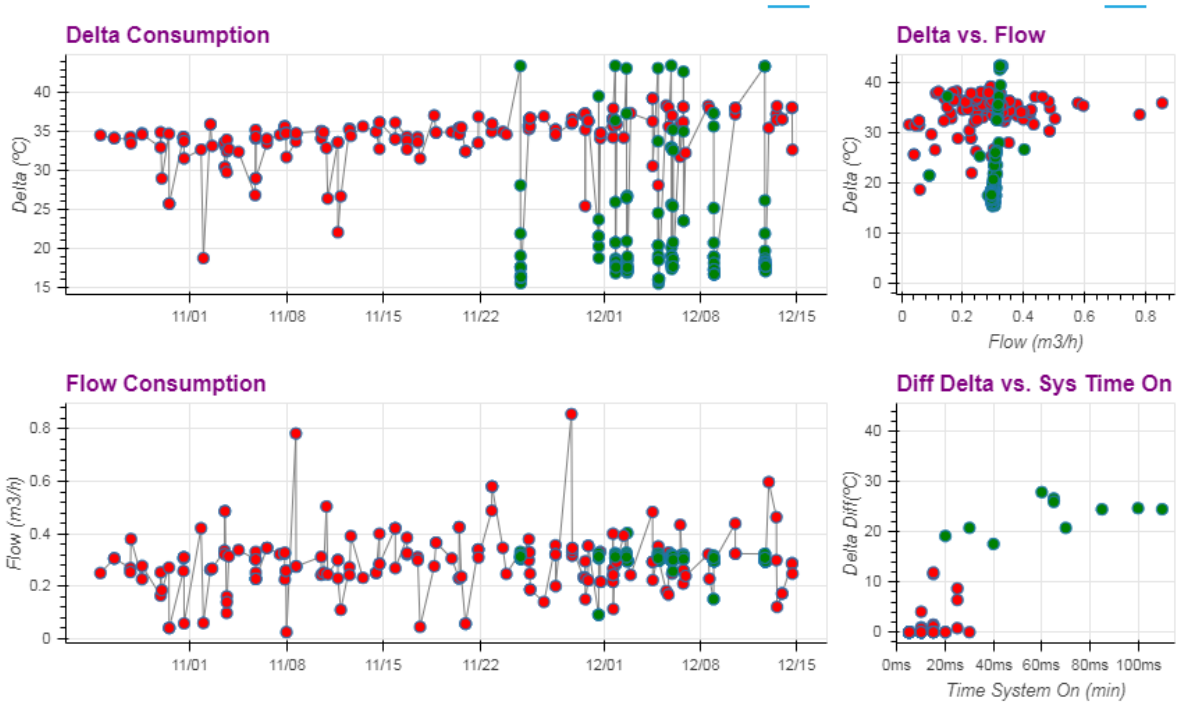


Figure 16: HVAC and DHW Clustering Results for 5056 Apartment

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

In next image it can be seen how the usage of HVAC and DHW is distributed along week 2020-12-17. In intense red and green colours, it is represented the 5056 consumption and in lighter red and green colours the total amount of energy used by all the station in the cluster in which apartment 5056 has been placed for that week.

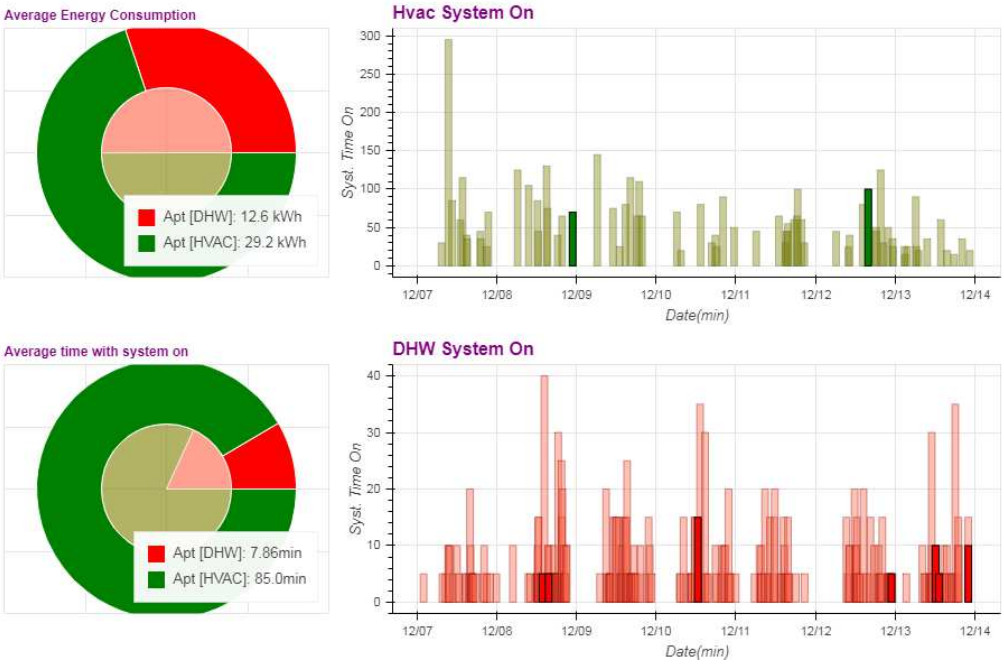


Figure 17: 5056 Apartment Consumption Comparation to Clusters Value for week 2020-12-17

Apartment 5050 & 5061:

Apartments 5050 and 5061 are a clear example of "hard" overlapping groups. In these examples it is possible to identify the relevance of the "event duration" feature. In the Flow vs Temperature Drop chart is clear that considering only that features many of HVAC events would have been classified as DHW events.

As in the previous case, there are some high valued flow and temperature drop measured marked as green, belonging to the ignition of the heating system. this behaviour is clearly shown in the Figure 19, where it is shown the algorithm results for 5050 apartments for different days.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

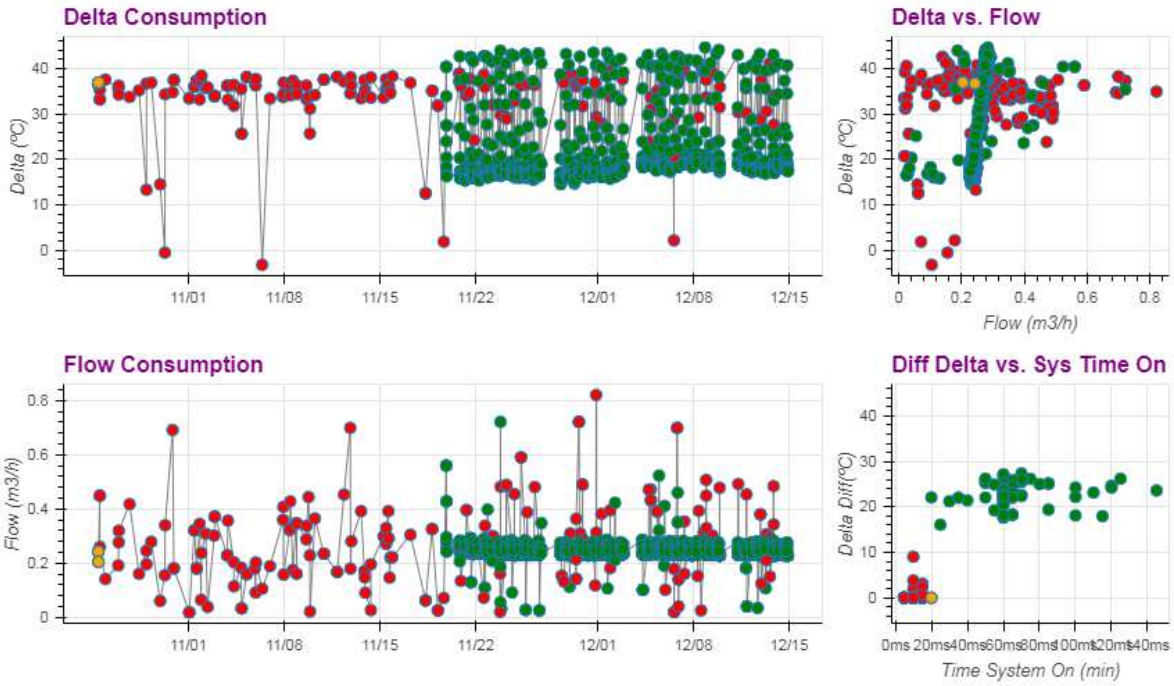


Figure 18: HVAC and DHW Clustering Results for 5050 Apartment

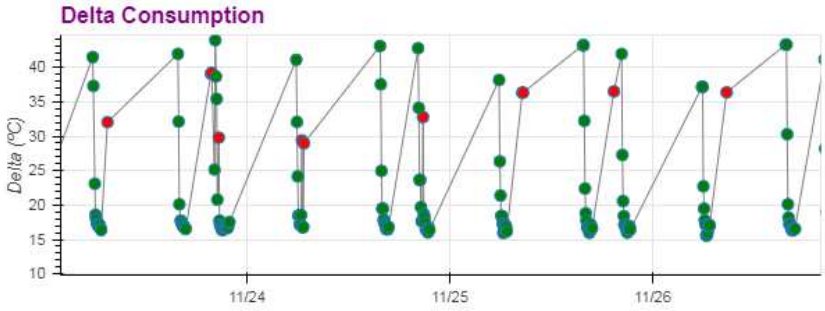


Figure 19: 5050 Apartment Clustered Measured for Different Days

In **¡Error! No se encuentra el origen de la referencia.** it can be seen how the usage of HVAC and DHW is distributed along week 2020-12-17. In intense red and green colours, it is represented the 5050 consumption and in lighter red and green colours the total amount of energy used by all the stations in the cluster in which apartment 5050 has been placed for that week

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

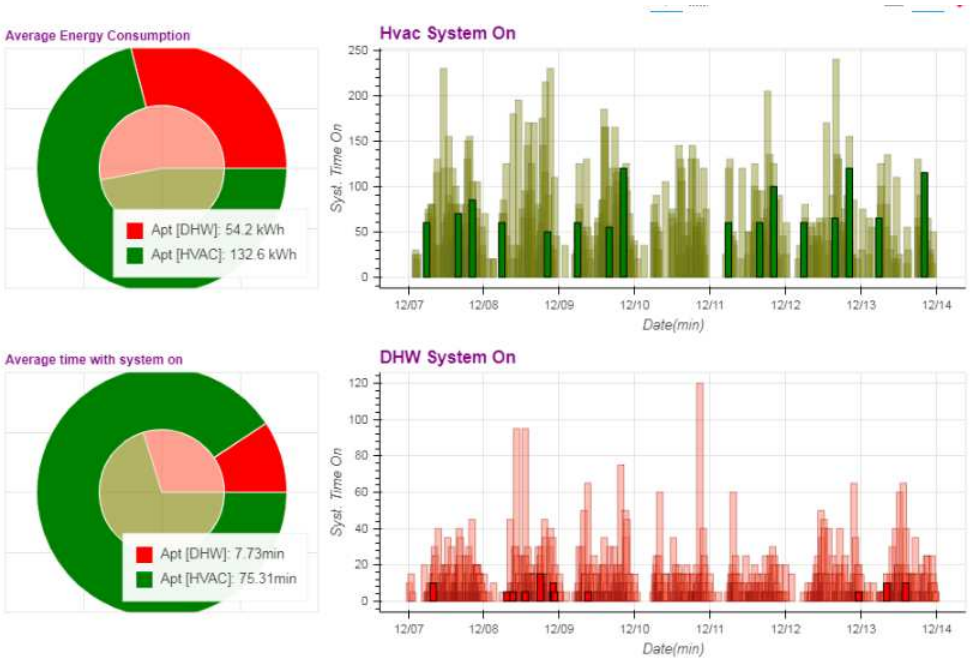


Figure 20: 5050 Apartment Consumption Comparation to Clusters Value for week 2020-12-17

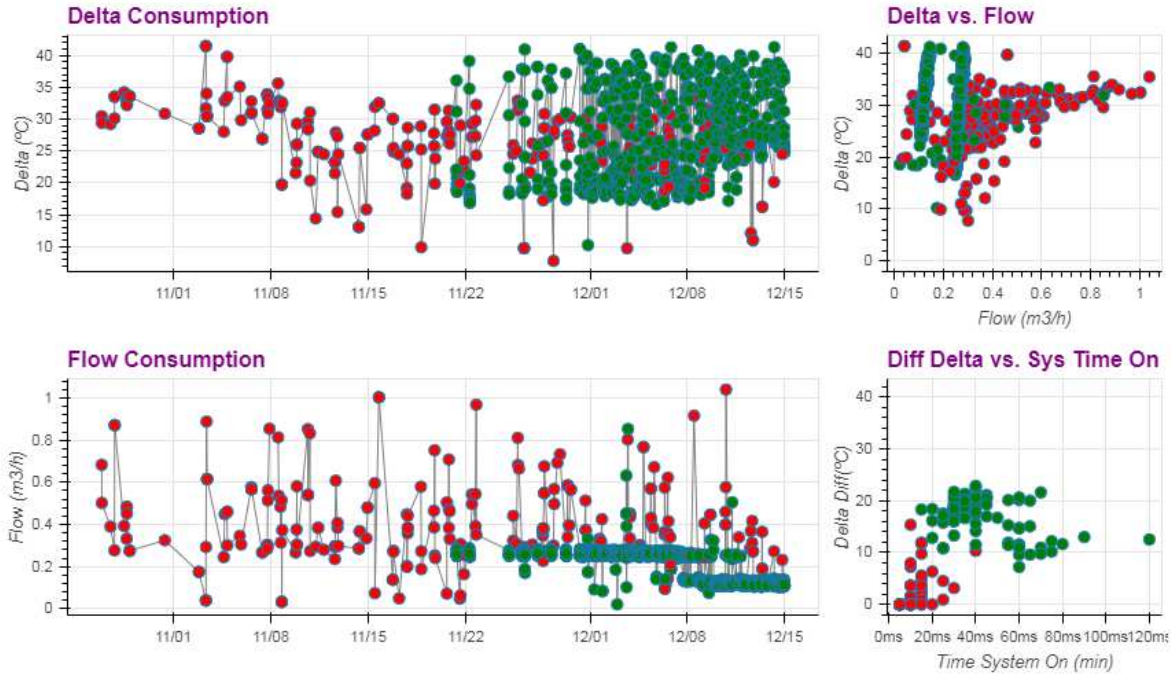


Figure 21: HVAC and DHW Clustering Results for 5061 Apartment

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

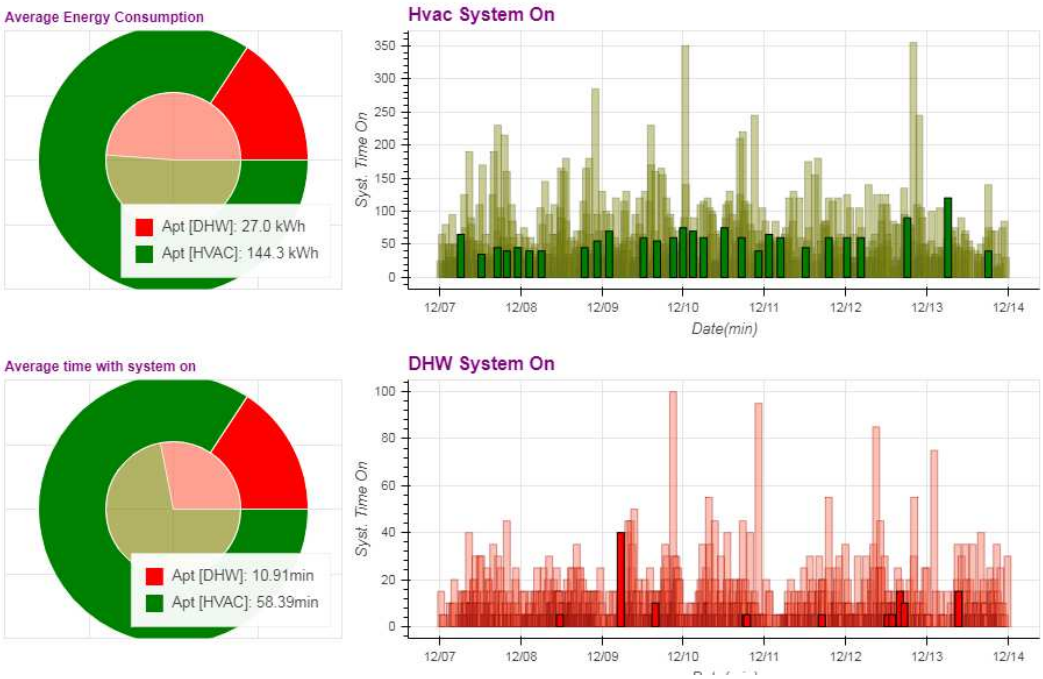


Figure 22: 5061 Apartment Consumption Comparation to Clusters Value for week 2020-12-17

Taking into consideration the outcome of the HVAC-DHW split algorithm is possible to generate additional views that describe the hourly aggregated energy consumption detailing which part belongs to HVAC and which one to DHW. Two examples are included below.

- Apartment 5050: The following figure shows the hourly added energy consumption for each day of December 2019, for 5050 apartment. In green, the amount due to HVAC system consumption and in red the amount due to DHW system consumption. In this case, the consumption is clearly higher than the previous cases. The results indicate that the HVAC system has been used for long daily periods almost every day of the month. On the other hand, measures identified as DHW have also been collected.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

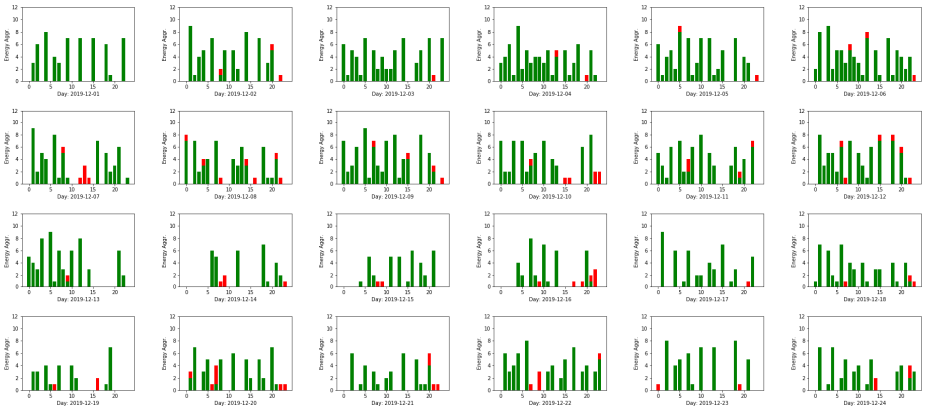


Figure 23: Hourly Added Energy per Day for HVAC (green) and DHW (red) for 5050 Apartment

- Apartment 5061: The following figure shows the hourly added energy consumption for each day of December 2019, for apartment 5061. In green, the amount due to HVAC system consumption and in red the amount due to DHW system consumption. In this case, there is also a high consumption of HVAC. Such different HVAC consumption values for the same month days for different apartments may be due to the orientation and their occupancy.

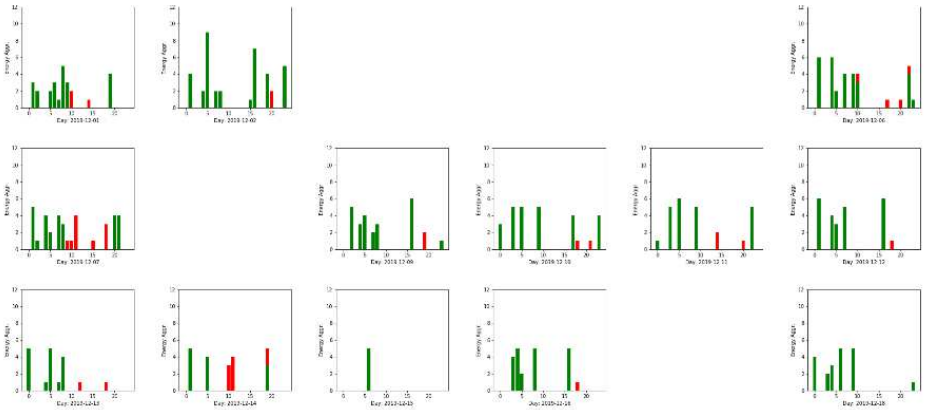


Figure 24: Hourly Added Energy per day for HVAC (green) and DHW (red) for 5056 Apartment

Note: Charts generated only for days with valid data. Missing chart means that there was not valid data for that day.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

As mentioned above the HVAC-DHW split functionality produces basis for deep analysis of the consumption profiles and gives added value to the benchmark functionality. The following set of figures describe the detailed analysis that can be provided combining the benchmark and HVAC-DHW functionalities.

For next study, the data was recovered from 668 different apartments and there were grouped for each week, according to their consumption. In other words, depending on the weekly consumption, the different apartments have been grouped into 3 different groups. Therefore, each apartment had been grouped with the apartments with a consumption like theirs. It should be remembered that the groups are not fixed, so a new aggrupation would be made each week.

- **Date 2020-10-26**: Next image shows week starting in date 2020-10-26 cluster aggrupation analysis. In this case the third aggrupation, purple coloured one, had a notorious higher amount of energy consumption. Also, even if in all cases the DHW consumption if higher than HVAC consumption, the apartments of this groups have also more HVAC consumption than in the other cases. Studying the Total Energy Consumption in time graph it is shown that the three different clusters have different consumption profiles, this can be due to the different apartment orientation, size or having different working hours.

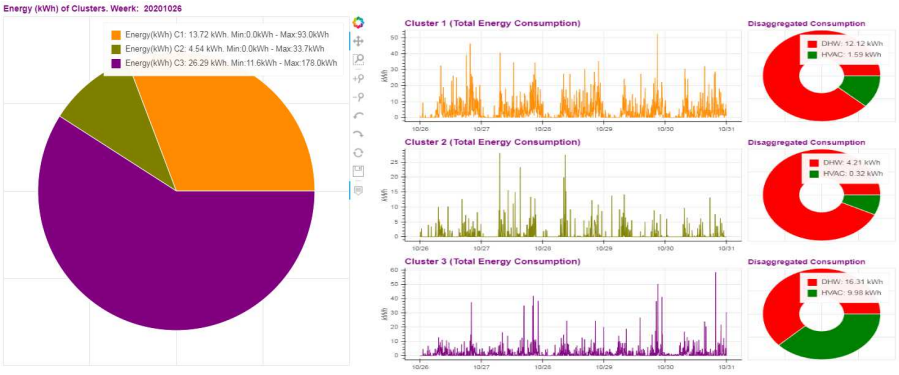


Figure 25: Disaggregation of Energy Consumption for day 2020-10-26

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

- **Date 2020-11-09**: For week starting in date 2020-11-09, the aggrupation with higher amount of consumption is the second one, green coloured one. Studying the Total Energy Consumption in time graph it is shown that the three different clusters have different consumption profiles. Also, even if in all cases the DHW consumption if higher that HVAC consumption, the apartments of this groups have also more HVAC consumption than in the other cases. It is shown that, the higher the HVAC consumption, the group consumption profile would have a plateau and more longer system switched on times

Figure 26: Cluster detail 2020-11-09

- **Date 2020-11-16**: For week starting in date 2020-11-16, the aggrupation with higher amount of consumption is the third one, purple coloured one. Also, even if in all cases the DHW consumption if higher that HVAC consumption, the apartments of this groups have also more HVAC consumption than in the other cases. Studying the Total Energy Consumption in time graph it is shown that the three different clusters have different consumption profiles. The third group, being the one with higher HVAC energy consumption is also the one which has the system turned on for longer time.

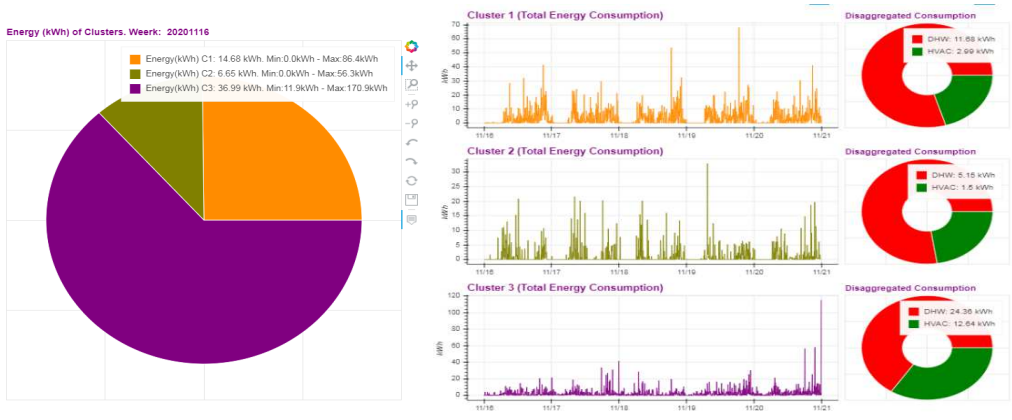


Figure 27:Cluster detail 2020-11-16

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

- Date 2020-11-30: For week 2020-11-30, the aggrupation with higher amount of consumption is the third one, purple coloured one. The closer to winter, the higher energy consumption due to HVAC. So, the system would be switched on for longer times in the three cases, which translates into a graph with a flatter profile but more continuous over time. Studying the Total Energy Consumption in time graph it is shown that the three different clusters have different consumption profiles. But in this case, comparing to previous cases, it is clearly shown that HVAC consumption has been increased in all three aggrupation.

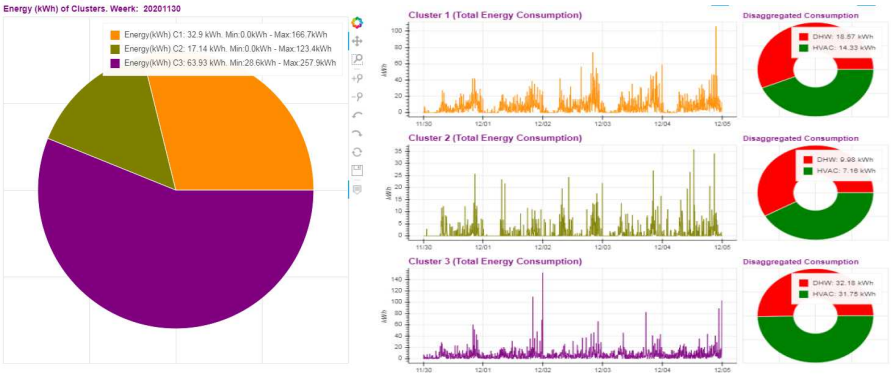


Figure 28:Cluster detail 2020-11-30

6. SERVICE FOR ENERGY PROVIDERS

The set of functionalities implemented in the current scope have as objective to provide relevant information to the energy provider in order to accommodate the energy production to the real day ahead needs.

6.1 Aggregated day ahead energy consumption forecast

6.1.1 Description

The objective of the day ahead energy consumption forecast is to predict the hourly aggregated day ahead consumption for the Txomin neighbourhood.

The implemented algorithm is divided in two steps:

- Consumption Pattern discovery: The consumption patterns identification is focused on

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

identifying similar behaviour days that help to create more accurate training models.

- Create predictive models: The predictive model

Peak detection is a common task in time-series analysis and signal processing. Standard approaches to peak detection include using smoothing and then fitting a known function to the time-series; and matching a known peak shape to the time-series. Another common approach to peak-trough detection is to detect zero-crossings (local maxima) in the differences (slope sign change) between a point and its neighbours. However, this detects all peaks-troughs, whether strong or not.

The combination of peaks detection and shape factors of daily consumption profiles has been used in REPLICATE DSP to identify the different day types regarding to their consumption patterns.

Regression has been widely studied from the statistics field, which provides different approaches to this problem: linear and generalized linear regression, least and partial least squares regression (LS and PLS), least absolute shrinkage and selection operator (LASSO) etc. Furthermore, several methods arising from the field of machine learning were designed to be universal function approximators, so they can be applied both for classification and regression: neural networks, support vector machines, regression trees and rules, bagging and boosting ensembles, random forests and others.

The predictive model implemented in the REPLICATE DSP follows the multilinear auto-regressive (AR) approach. A general description of the autoregressive models could be given by saying that these models explain, partially at least, the values of a variable or set of variables, based on the past values of this variable or set of variables.

The implemented regression model considers the outdoor temperature as exogenous variable. An exogenous variable is one whose value is determined outside the model and is imposed on the model. In other words, variables that affect a model without being affected by it. In order to include in the model the relevance of the period of the day, the forecasting is implemented as the sequential forecast of 24 values. In other words, for each one hour period of the day its own model is created on the fly.

The training data set for each period of the day has the following structure.

Date	Feature	Feature	Feature	Result
day -1	Energy $_{ti-2}$	Energy $_{ti-1}$	OutdoorT_{i-1}	Energy T_i

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

day -2	Energy $_{ti-2}$	Energy $_{ti-1}$	OutdoorT_{i-1}	Energy T_i
day - ..	Energy $_{ti-2}$	Energy $_{ti-1}$	OutdoorT_{i-1}	Energy T_i
day -n	Energy $_{ti-2}$	Energy $_{ti-1}$	OutdoorT_{i-1}	Energy T_i

Considering the 24h ahead weather forecast and the seed values of the previous valid two time steps the 24h forecast is triggered, in order to minimize the impact in the deviation of the seed values, the process is repeated every hour, in this way unexpected behaviours may be assimilated by the forecasting process.

6.1.2 Architecture Design

The forecasting process is divided in two sequential steps. The first of the steps identifies consumption patterns and identified different day types regarding to their consumption profile. The second step creates models for each of the day types identified before and then produces the forecast using the appropriate model.

The forecasting process is implemented as an autoregressive method, in this context the proper selection of training days, with comparable behaviour is a key factor. In datasets with very different behaviours extreme values could be considered as outliers and taken out of the model. This could lead over or underestimate measured magnitudes.

The figure below describes the both steps mentioned above. ML package implements the boxes highlighted in green.

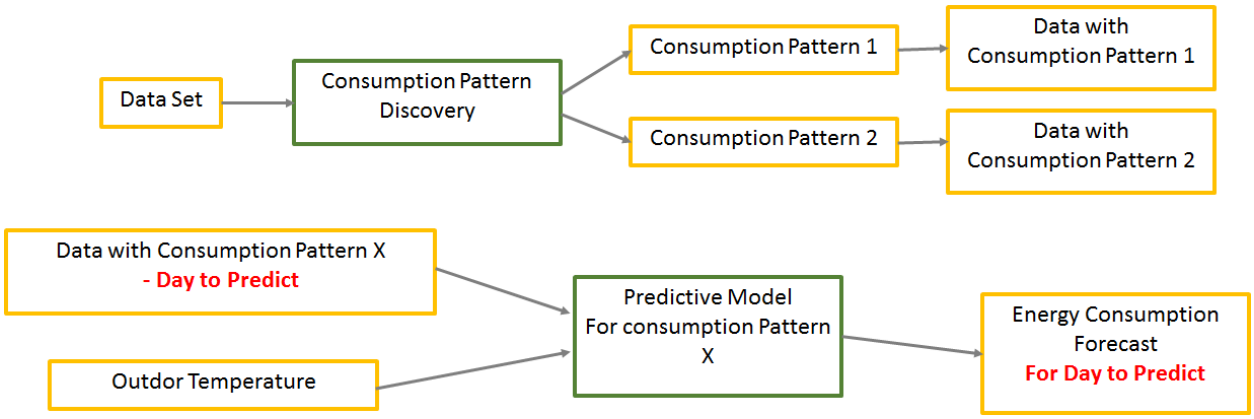


Figure 29: Data Architecture for Consumption Pattern Discovery

The day type identification process in the Txomin neighbourhood delivered well defined

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

patterns for week or weekend days.

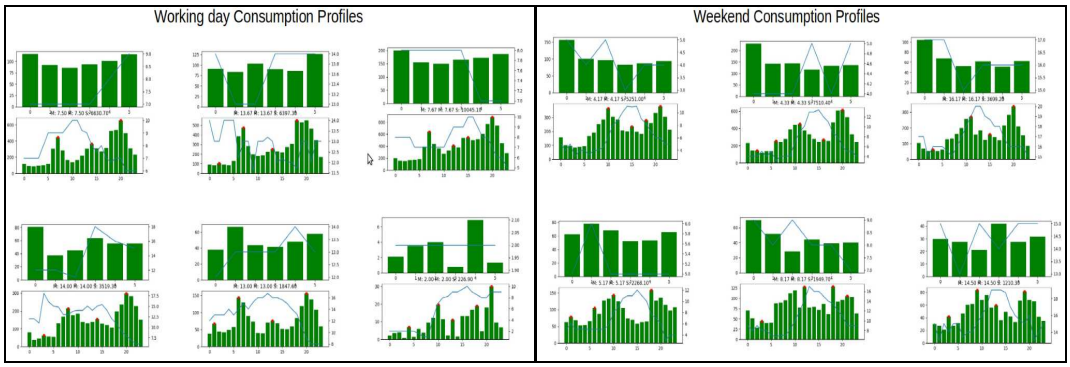


Figure 30: 24h (even rows) and 6h (odd rows) consumption profiles. In blue outdoor temperature.

During working days, there are clear consumption peaks at 7am and 8pm approximately, these peaks, weekend days are shifted to 10am approximately. Another interesting feature is the consumption drop, during working days the consumption drop after the early morning peak is abrupt, during weekends the consumption drop is, in most cases, smooth.

From the picture above it can be concluded that make sense to split the training and forecasting processes in working days and weekend days

6.1.3 Implementation and Results

The variability that individual homes present makes that the forecasting process applied to single homes losses relevance. On the other hand, the forecasting process applied to sub-centrals, intermediate points between the destring heating plant and individual homes, can help to identify, at group of apartment level, different needs in regarding to the type of construction, average age of the population or other characteristics.

The day ahead forecasting process has been implemented as a batch process that triggers day ahead forecast every mid-night. The forecasting process it is not as a sliding window forecasting, this means that its execution of the process tunes the outcomes for the rest of the day, not for the next 24h ahead.

The accuracy of the results has been measured using the MAPE and RMSE values.

The MAPE (Mean Absolute Percent Error) measures the size of the error in percentage terms. Most people focus primarily on the MAPE when assessing forecast accuracy because people are comfortable thinking in percentage terms, making the MAPE easy to interpret. The MAPE is scale sensitive and should not be used when working with low-volume data. Notice that because "Actual" is in the denominator of the equation, the MAPE is undefined when Actual

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

demand is zero. Furthermore, when the Actual value is not zero, but quite small, the MAPE will often take on extreme values. This scale sensitivity renders the MAPE close to worthless as an error measure for low-volume data.

In order to add more detail about the accuracy of the forecasts the Root Mean Square Error (RMSE) value has been added. RMSE is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results.

The MAPE and RMSE values for each forecasting have been included. MAPE value refers to the

- **Date 29-10-2020**: The forecast for the sub-central 5017 shows how the forecasting method matches almost all the peaks of the real consumption. From the shape of the figure mainly three aspects may be highlighted.
    - Early morning "table": Between 6am and 8 am the chart shows a flat profile, this is quite rare, most of the times the consumption shows a peak as it shows at dinner time.
    - Spiky noon: At noon and in the afternoon some consumption spikes appear. This behaviour is stronger some days and have clear impact in the accuracy of overall forecast.
    - Dinner peak: The "dinner" peak is the most regular behaviour. It clearly shows the home arriving time for the apartment owners. At this point it worth to remember the location of the pilot, north of Spain, where families regular dinner time in later than the rest of EU.

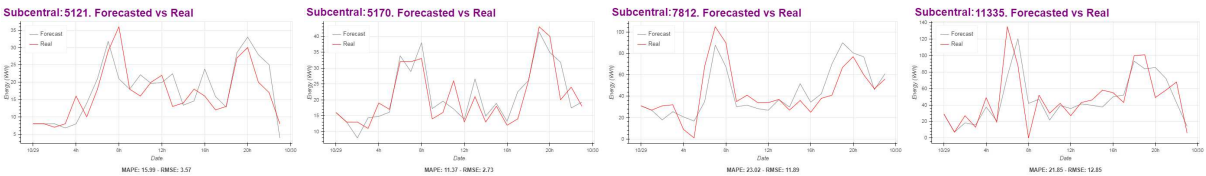


Figure 31: Energy Consumption Forecast for some Sub-Centrals

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

7. DEPLOYMENT

7.1 Description

REPLICATE's high level entities introduced in chapter #1 from the implementation point of view are divided in four software packages

- Replicate Forecastor:
- Replicate Replicator
- Replicate Data-server (fetch mode):
- Replicate Data-server (request mode):

The software and hardware requirements to implement and deploy the mentioned software packages are:

- Software requirements: The REPLICATE DSP implementation has been validated using Ubuntu 18.04 environment, as development environment Ubuntu 20.04 has been tested. Nevertheless, for deployment and production Ubuntu 18.04 is recommended. License free tools have been selected for DSP implementation, Java 1.8 (Replicator) and Python 3.7 (Forecastor & Data-server) have been the programming languages used during the development.
- Hardware requirements: The algorithms implemented in the DSP have a quite big computation cost in terms of calculations size, power and parallelization, in this context the minimum requirements for deployment environment are:
    - CPU: Minimum 4 CPU platform, better if 8 CPU platforms
    - RAM: Minimum 8GB RAM available, better if 16 RAM
- Third party tools: To complete the DSP functionalities, database engine and data base management tools are required, in both cases license free tools have been selected.
    - MySQL 5.7: The worldwide well known MySQL data base engine has been selected to store data entities and models required by the DSP as well as to store the its outcomes.
    - WS02 Data Server Service: In order to standardize and export and data base

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

access API the WS02 Data Server Service (DSS) has been selected. The DSS enables the implementation of HTTP-Rest interfaces to allow third parties to access the DSP results in a regular way.

The forecastor and the data-server packages implement the following folder tree for deployment and development.

- **images**: The images folder stores pictures and charts that are created during the algorithms execution and that may help to validate or evaluate their execution. The pictures and charts stored in the images folder are not considered as outcome or results of the algorithms
- **results**: The results folder stores the result files (csv, pictures, text,..) that can be considered as outcome of the algorithm as well as the log files generated during its execution.
- **src**: Main folder for implemented source code. The code folder may contain sub-folders. The *code* folder stores the configuration file too.
- **repo**: The repo folder stores the files (csv, pictures, text), that are valid inputs for the algorithms. It includes two sub-folders, *raw* and *processed. Processed* folder contains those files that being inputs for the algorithms can be considered as cooked files derives from files in raw folder.

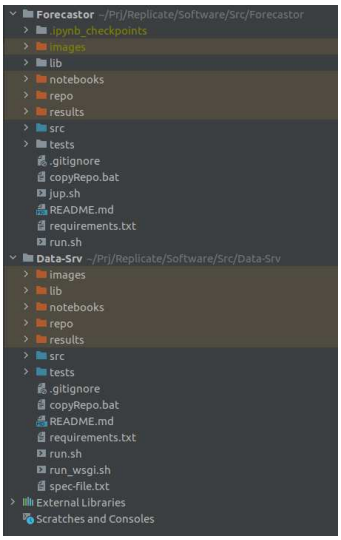


Figure 32: Development and Deployment tree

The following Tecnalia's repositories have been enabled during the development phase.

- https://git.code.tecnalia.com/DigitalEnergy/replicate/forecastor.git

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

- https://git.code.tecnalia.com/DigitalEnergy/replicate/replicator.git
- https://git.code.tecnalia.com/DigitalEnergy/replicate/repdata-srv.git

7.2 Architecture Design

7.2.1 Replicate DSP ML-Package

Forecastor package implements the benchmarking and forecasting algorithms included in the DSP. The algorithms, already described in previous chapters, are executed in sequential way, by means of a configuration file the execution of one, or some, of them can be deactivated. In the same way, the outcomes produced by their execution can be tuned adjusting some values in the configuration file too.

- Benchmark flow: The apartment benchmark functionality creates clusters that group apartments taking into consideration their hourly aggregated consumption. It is relevant to mention that the main inputs (but the weather forecast) are read as datafiles stored in *repo/processed* folder.

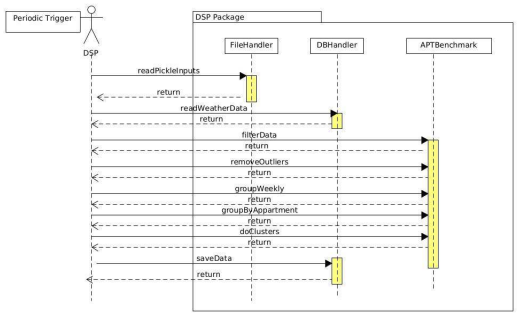


Figure 33:Apartments benchmark flow

The most relevant configuration file entries for benchmark module:

- **ENERGY_SAMPLE_TIME**: Defines the sample time be considered to aggregate energy values (Default value 1H)
- **ENERGY_SAMPLE_TIME_VALUES**: Linked to the previous one the values must be consistent (Default value 24)
- **OUTDOOR_ENERGY_SAMPLE_TIME**: Defines the minimum outdoor weather data frequency given in minutes (Default value 5)
- **CLUSTER_SAMPLE_TIME**: Defines the aggregation period for the clustering process (Default value 4H)
- **CLUSTER_SAMPLE_TIME_VALUES**: Linked to the previous one, the values must be consistent (Default value 6H)

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

- **IS_DMZ**: Defines if the algorithm is being executed in cloud server or in a desktop computer. When used in desktop some functionalities as disabled. (Default value 1)
- **OUTDOOR_FROM_DB**: Defines if the weather data is read from database or it is available as input file in the repository folders (Default value 1H)
- **IS_PLOT**: Defines if intermediate plots are required (Default value 0)
- **SAVE_PATTERS_DB**: Used to execute dry-runs, enables or disables data base storage of outcomes (Default value 1)
- **DO_WEEKLY_CLUSTERS**: Enables de execution of the benchmark functionality without any normalization (Default value 1)
- **DO_WEEKLY_CLUSTERS_M2**: Enables the execution of the benchmark functionality, taking into consideration apartments area for normalization. (Default value 1)

- Forecasting flow: The day ahead energy consumption forecast is designed to predict the consumption at group of apartments level. As for the benchmark the main inputs are read from input files. The historic weather data, required for model training, and the day ahead forecast, required to predict, are read from the database.

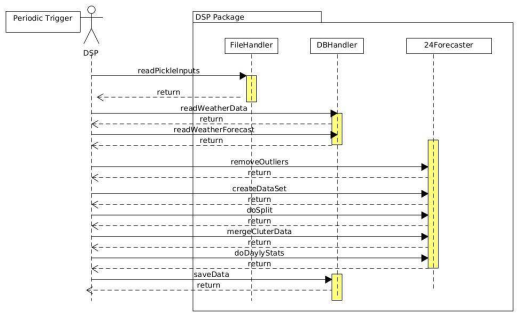


Figure 34: Day ahead forecast

The most relevant configuration file entries for day ahead forecasting module.

- **ENERGY_SAMPLE_TIME**: Defines the sample time be considered to aggregate energy values (Default value 1H)
- **ENERGY_SAMPLE_TIME_VALUES**: Linked to the previous one the values must be consistent (Default value 24)
- **OUTDOOR_ENERGY_SAMPLE_TIME**: Defines the minimum outdoor weather data frequency given in minutes (Default value 5)

- **IS_DMZ**: Defines if the algorithm is being executed in cloud server or in a desktop computer. When used in desktop some functionalities as disabled. (Default value 1)
- **OUTDOOR_FROM_DB**: Defines if the weather data is read from database or it is available as input file in the repository folders (Default value 1H)
- **IS_PLOT**: Defines if intermediate plots are required (Default value 0)
- **SAVE_FORECAST_DB**: Used to execute dry-runs, enables or disables data base storage of outcomes (Default value 1)
- **DO_DAY_AHEAD_ENERGY_FORECAST_SC**: Enables de execution of the forecasting functionality (Default value 1)

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

- HVAC-DHW Split flow: The HVAC-DHW consumption split is the third of the functionalities implemented in the DSP ML-Package. Its behavior is similar to the modules described above, but as additional input it handles the outcomes of the clustering processed for its comparisons.

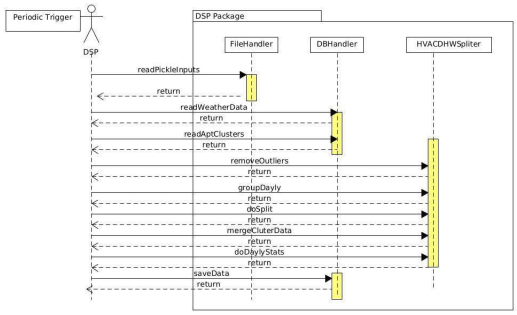


Figure 35:HVAC-DHW Split

The most relevant configuration file entries for HVAC-DHW split module:

- **ENERGY_SAMPLE_TIME**: Defines the sample time be considered to aggregate energy values (Default value 1H)
- **ENERGY_SAMPLE_TIME_VALUES**: Linked to the previous one the values must be consistent (Default value 24)
- **OUTDOOR_ENERGY_SAMPLE_TIME**: Defines the minimum outdoor weather data frequency given in minutes (Default value 5)
- **CLUSTER_SAMPLE_TIME**: Defines the aggregation period for the clustering process (Default value 4H)
- **CLUSTER_SAMPLE_TIME_VALUES**: Linked to the previous one, the values must be consistent (Default value 6H)

- **IS_DMZ**: Defines if the algorithm is being executed in cloud server or in a desktop computer. When used in desktop some functionalities as disabled. (Default value 1)
- **OUTDOOR_FROM_DB**: Defines if the weather data is read from database or it is available as input file in the repository folders (Default value 1H)
- **IS_PLOT**: Defines if intermediate plots are required (Default value 0)
- **IS_PLOT_WFA**: Defines if additional detailed plots of water flow analysis execution are required
- **SAVE_PATTERS_DB**: Used to execute dry-runs, enables or disables data base storage of outcomes (Default value 1)
- **DO_WATERFLOW_ANALYSIS**: Enables de execution of the hvac-dhw split and analysis (Default value 1)

7.2.2 Replicate Replicator

The Replicator package implements the collection process for all the data not directly linked to the execution of the Forecastor package, data related to the data model, and for the data that are retrieved from data external data sources, weather forecast data.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

The continuous update of the data model used by the Forecastor is relevant in order to keep the number of apartments and stations considered in its execution aligned to the apartments and stations registered in the DH management platform. The Replicator enables the execution of this replication process as an unattended process.

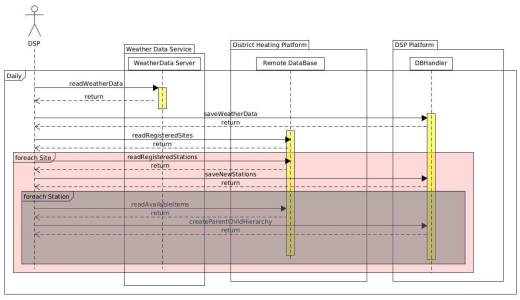


Figure 36:Data model replication flow

The ***meteostat*** has been selected as weather data service. The ***meteostat*** service offers a fee-free access which provides an appropriate reliability to implement REPLICATE's ML package algorithms. The meteostat services provides the following weather information:

- utc time: Time stamp
- time_local Local time stamp
- Outdoor temperature (°C)
- Dew point (°C)
- Relative humidity in percent (%)
- One hour precipitation (mm)
- Snow depth (mm)
- Wind direction in degrees (°)
- Average wind speed (km/h)
- One hour sunshine total in minutes (m)

For the REPLICATE DSP implementation only the local time, UTC time and outdoor temperature are relevant.

7.2.3 Replicate Data-Server (fetch mode)

The Data-Server package operating in its ***fetch*** mode retrieves for each of the stations

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

registered in the DH platform the thermal energy, thermal power, supply and return temperatures as the flow. Once the data is fetched, appropriate datasets are created in pickle files so its usage for other modules is prepared. The fetch mode is triggered periodically.

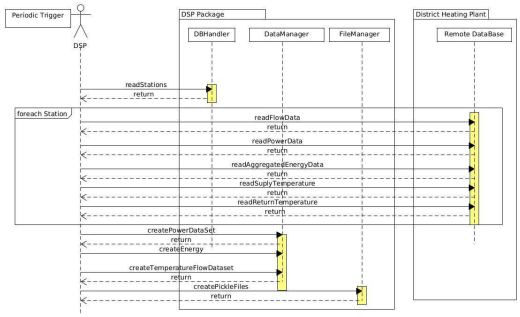


Figure 37: Data-server flow

Figure 38: Data-Server fetch mode.

7.2.4 Replicate Data-server (request mode):

In its request mode the Data-Server is ready to receive third party request to read results already stored in the data-base or to trigger a remote data fetching. In order to speed up the visualization processes, the data server does not stored data in raw but it creates meaningful structures.

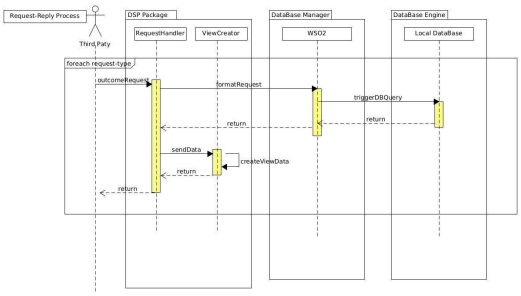


Figure 39: Data-server flow (request mode)

7.3 Public Interfaces

REPLICATE DSP development implements public interfaces to allow to third party developments read the outcomes of the algorithms. REPLICATE's public interface even it is implemented following the well-known HTTP-Rest definition, document and develop functionalities easy to

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

adopt by third parties is always a challenge.

In order to overcome challenges in collaborative projects the use of Swagger tools (Editor and Hub) has been decided. Swagger tools are specifically designed to create, share, collaborate, version, test, and publish REST-APIs and documentation in ways that don't require extensive customization or time.

During the REPLICATE DSP development, main challenges solved by using Swagger tools have been:

- **Versions**: Swagger not only allows to save API specification but also save different versions of the working specification used. As a result, it is possible to test new content by adding a new version, return to any version and publish or unpublish version. When a new publish happens, the published version becomes Read Only.

- **Export to HTML**: Among Swagger's many options for generating client and SDK files is an HTML option. Swagger allows to export the API spec as a static HTML file in one of two styles: HTML or HTML2. The HTML export is a more basic output than HTML2.
- **Mocking Servers**: Another feature of Swagger is the ability to create mock API servers and simulate responses that let users get a sense of how the API works.

The REPLICATE DSP functionalities have been covered implementing four different calls:

Base URL:

- **getWEEKCLClusters**: The call retrieves the data released by the benchmark functionality without normalization.

Method	GET
URL	http://150.241.40.157:5000/v1/replicate/fss/{token}/WEEKCL/object/measurements/{station}?fecha_desde=&fecha_hasta=
token	Id to grant access to the API
station	Required apartment id in the DH platform
fecha_desde	Start date for the query
fecha_hasta	End date for the query

- **getWEEKCLClustersM2**: The call retrieves the data released by the benchmark functionality normalized considering apartment area.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

Method	GET
URL	http://150.241.40.157:5000/v1/replicate/fss/{token}/WEEKCLM2/object/measurements/{station}?fecha_desde=&fecha_hasta=
token	Id to grant access to the API
station	Required apartment id in the DH platform
fecha_desde	Start date for the query
fecha_hasta	End date for the query

- **getHVACDHWSplit**: The call retrieves the data released by the HVAC-DHW split functionality.

Method	GET
URL	http://150.241.40.157:5000/v1/replicate/fss/{token}/HVAC_DHW/object/measurements/{station}?fecha_desde=&fecha_hasta=
token	Id to grant access to the API
station	Required apartment id in the DH platform
fecha_desde	Start date for the query
fecha_hasta	End date for the query

- **getForecastSC**: The call retrieves the data released by the day ahead energy consumption forecasting functionality.

Method	GET
URL	http://150.241.40.157:5000/v1/replicate/fss/{token}/FORECASTSC/object/measurements/{station}?fecha_desde=&fecha_hasta=
token	Id to grant access to the API
station	Required apartment id in the DH platform
fecha_desde	Start date for the query
fecha_hasta	End date for the query

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

8. USER INTERFACE – LOOK AND FEEL

User awareness is one of the main goals for the REPLICATE DSP implementation, in that context a friendly user web interface has been developed. The user interface development, indeed, has been a real test of probe of the overall DH platform (DH SCADA + REPLICATE DSP) design. The UI implementation has been done by a company not involved in the REPLICATE project, in this context the flexibility and scalability of the DH platform to interface with third parties has been successfully verified and validated.

The set of pictures below describe some views example of the UI developed for end-users.

The HVAC-DHW split results and data analysis stacked bar charts have been developed. Each section of the bar represents either the HVAC or DHW consumption.

The user interface developed for end-users, apartment owners, allows the following options for all the implemented functionalities:

- Request period selection: The user can request the period of data to be shown in the interface, daily, weekly and monthly are the available options
- Aggregation period selection: For the request period the user can select the aggregation period, every 5, 15 and 60 minutes are the available options.

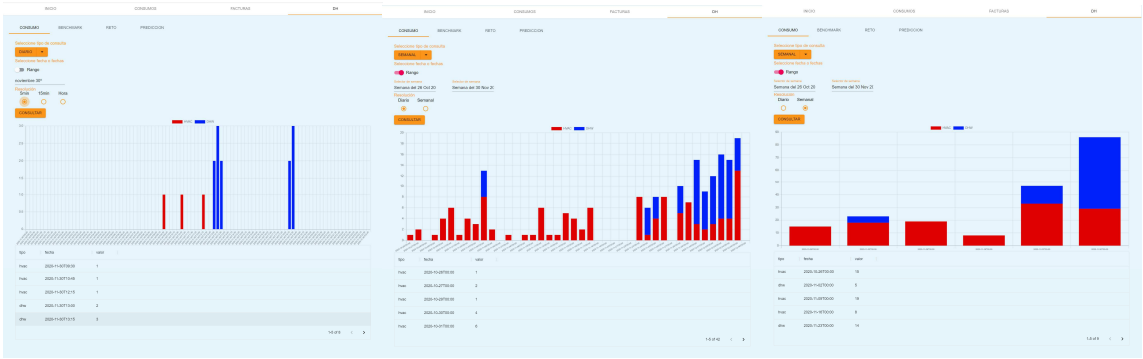


Figure 40:UI example for HVAC-DHW split

The set of images below describe the view implemented for the benchmarking functionality Benchmarking views include the same information already described in chapter #4.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

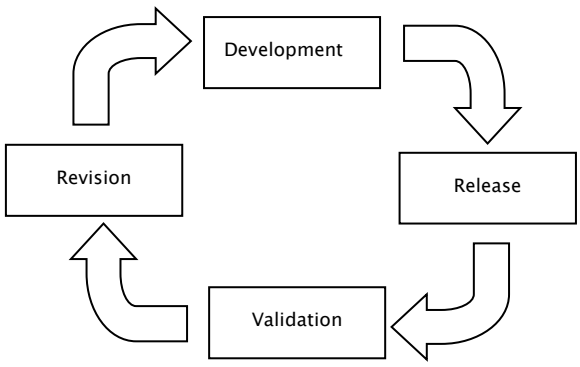
Figure 41: Benchmarking UI

The rest of the functionalities described in the current document follow the same concept of period requesting and data grouping. For each of them the information included is the already described in the corresponding chapter. It is important to mention that the UI, even being designed as a multilanguage application, so far, is only available in Spanish.

9. LESSONS LEARNT

During the REPLICATE DSP implementation several lessons have been learnt, most of them related to ML algorithms development and tuning to building and apartment domains but also in the field of agile developments methods for multi-party applications.

The "sprint" based development planning delivers continuous releases that enables parallel development and validation phases. From the development point of view each of the functionalities in the REPLICATE DSP has been planned as a sprint following the next sequence:



This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

Regarding to the ML implementation the most relevant lesson is that in consumption patterns in which little deviations may represent 100% of accuracy loss, as individual apartment forecasting is, the power of ML is lost.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

10. INNOVATIONS, IMPACTS AND SCALABILITY

10.1 Innovation solution

The innovation provided by the REPLICATE DSP comes from the perspective of the new services that the usage of ML is able to generate. The implementation of the REPLICATE DSP did not require the installation of any additional sensor or metering device, in other words, the usage of regular sensors used by the building heating industry is able to provide advanced monitoring and awareness services. Nevertheless, the adoption of new sensors may help in the overall accuracy of the solution or algorithms calibration and training phases

10.2 Social impacts

Awareness in the energy usage and the behavioural change that it may lead are direct social impacts of the REPLICATE DSP implementation. Little by little citizens understanding of the global impact of individual actions may be achieved with developments as REPLICATE DSP is.

10.3 Environmental impacts

Linked to the social impacts, behavioural changes in the way in which energy consumption is understood and handled has direct impact in our environment.

10.4 Replication and scalability potential

The REPLICATE DSP, as already has been described in the current document, is a development that fits DH plants management standards in monitoring and third party interoperability. The implementation of the REPLICATE DSP has not involved additional or customization work in the DH plant management framework. The seamless integration with the DH plant management framework probes that the REPLICATE DSP could be easily adopted by other DH plant deployments.

Chapter 7 describes basics for replication process, constraints and pitfalls.

10.5 Economic feasibility

The implantation of the REPLICATE DSP, as already mentioned, does not require additional

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

sensor but those needed in regular DH deployments, in this context, all the commissioning, maintenance and repair operations do not need additional budget prevision. The almost null budget burden in additional hardware or sensing stuff makes the REPLICATE DSP a very attractive solution to be adopted by the DH managers.

On the other hand, the costs of a new software development, deployment, maintenance and training have to be considered. These costs if the exploitation is foreseen in for single neighbourhood may be considerable high but as it happens with any software development, as much as deployment cases increases the cost effectiveness is increased too.

10.6 Impact on SME´s

TECNALIA, as research center is not really interested in full exploitation of its developments, the technology transfer is the final purpose of any activity in research center. In this context during the REPLICATE DSP preliminary pillars and meetings about how to articulate this transfer has been held

10.7 Other

N.A.

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 691735

11. CONCLUSIONS

The implementation of the REPLICATE DSP has been a challenge that delivered some interesting outcomes, not only as result of the implemented methods but also in the domain of the solution feasibility and deployment constraints.

The REPLICATE DSP development phase, conducted as mentioned above as a continuous development – validation process, which led to highlight in which the relevance of not having an additional hardware burden. The building management industry is used to apply mature, off-the-shelf metering equipment, the need of additional sensors which would have require, commissioning or installation training would have been a constraint for the REPLICATE DSP generalization. However, in order to solve the lack of some sensors, the monitoring data already provided by the DH has allowed the DSP to develop the required algorithms for estimating the additional needed data. Future DH developments should consider the need of increasing the investment in sensors that might be necessary to further develop DSP solutions with additional recommendations to citizens and also to work on consumptions predictions. From the point of view of the applied ML methodology and results, the first thing to say is that the traditional ML methods perform quite well in the context of REPLICATE DSP. The use of approaches as deep-learning, real time streaming or other state of art research areas in ML is not necessary to tackle challenges as REPLICATE DSP is. The outcomes of the REPLICATE DSP show how ML methods may increase the awareness and detail of the energy consumption and in parallel identify scenarios which require maintenance actions due to unexpected flow or temperature values that at the end deliver excessive consumption or lack of comfort for end users.