BD4NRG
Big Data for Next Generation Energy

D4.6: BD4NRG-PROCESSING/ANALYTICS - First. Technology Release

WP4 – Data Modelling & Open Modular AI-based edge-level Analytics Toolbox

December 2021

www.bd4nrg.eu

Disclaimer

Copyright Message

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grantagreement No 872613

Grant Agreement Number: 872613 Acronym: BD4NRG

Full Title	Big Data for Next Generation Energy		
Topic	DT-ICT-11-2019 Big data solutions for energy		
Funding scheme	H2020- IA: Innovation Action		
Start Date	January 2021	Duration	36
Project URL	http//www.bd4nrg.eu		
Project Coordinator	ENG		
Deliverable	BD4NRG-PROCESSING/ANALYTICS - First. Technology Release		
Work Package	WP4 – Data Modelling & Open Modular AI-based edge-level Analytics Toolbox		
Delivery Month (DoA)	M10	Version	1.0
Actual Delivery Date	28/01/2022		
Nature	Other	Dissemination Level	Public
Lead Beneficiary	ENG		
Authors	Marzia Mammina [ENG], Diego Arnone [ENG], Alessandro Rossi [ENG], Salvatore Cipolla [ENG], Davide Profeta [ENG], Elissaios Sarmas [NTUA], Vaggelis Marinakis [NTUA], Vangelis Spiliotis [NTUA], Vagelis Karakolis [NTUA]		
Quality Reviewer(s):	Elisa Hermann [ATOS], Marco Antonio Bucarelli [ASM]		
Keywords	Big Data, Artificial Intelligence, Machine Learning, Analytics		

Preface

BD4NRG focuses on addressing emerging challenges in big data management for energy sector with an innovative open holistic solution for smart grid-tailored, near real time, energy-specific and AI-based open Big Data Analytics modular framework. The vision is to deliver holistic services for techno-economic optimal management of Electric Power and Energy Systems value chain. Services range from optimal risk assessment for energy efficiency investments planning, to optimized management of grid and non-grid owned assets, improved efficiency, and reliability of electricity networks operation, while at the same time contributing to achieve fair energy prices to the consumers and laying the foundations for an EU-level energy-tailored data sharing economy. The BD4NRG Toolbox will be implemented and validated in real-life pilots in 12 large-scale demo sites across 8 countries for:

- Increasing the efficiency and reliability of the electricity network BD-4-NET
- Optimising the management of assets connected to the grid BD-4-DER

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

- De-risking investments in energy efficiency and increasing the efficiency and comfort of buildings
  BD-4-ENEF

Who We Are

	Participant Name	Short Name	Country Code	Logo
1	ENGINEERING – INGEGNERIA INFORMATICA SPA	**ENG**	IT	
2	NATIONAL TECHNICAL UNIVERSITY OF ATHENS	**NTUA**	GR	
3	RHEINISCH-WESTFAELISCHE TECHNISCHE HOCHSCHULE AACHEN	**RWTH**	DE	
4	EUROPEAN DYNAMICS LUXEMBOURG SA	**ED**	LU	
5	INTERNATIONAL DATA SPACES EV	**IDSA**	DE	
6	EUROPEAN NETWORK OF TRANSMISSION SYSTEM OPERATORS FOR ELECTRICITY AISBL	**ENTSO-E**	BE	
7	PANEPISTIMIO DYTIKIS ATTIKIS	**UNIWA**	GR	
8	ATOS SPAIN SA	**ATOS**	ES	
9	FUNDACION CARTIF	**CARTIF**	ES	
10	UNIVERZA V LJUBLJANI	**UNILJ**	SL	
11	ENEL X SRL	**ENELX**	IT	
12	REN - REDE ELECTRICA NACIONAL SA	**REN**	PT	
13	CENTRO DE INVESTIGACAO EM ENERGIA REN - STATE GRID SA	**RDN**	PT	
14	UNINOVA-INSTITUTO DE DESENVOLVIMENTO DE NOVAS TECNOLOGIASASSOCIACAO	**UNINOVA**	PT	
15	ENERCOUTIM - ASSOCIACAO EMPRESARIALDE ENERGIA SOLAR DE ALCOUTIM	**ENERC**	PT	
16	FIWARE FOUNDATION EV	**FIWARE**	DE	
17	CENTRICA BUSINESS SOLUTIONS BELGIUM	**CENTRICA**	BE	
18	NEDERLANDSE ORGANISATIE VOOR TOEGEPAST NATUURWETENSCHAPPELIJK ONDERZOEK TNO	**TNO**	NL	

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

	Participant Name	Short Name	Country Code	Logo
19	ASM TERNI SPA	**ASM**	IT	
20	VIDES INVESTICIJU FONDS SIA	**LEIF**	LV	
21	COMSENSUS, KOMUNIKACIJE IN SENZORIKA, DOO	**COMSENSUS**	SL	
22	HOLISTIC IKE	**HOLISTIC**	GR	
23	INTERUNIVERSITAIR MICRO-ELECTRONICA CENTRUM	**IMEC**	BE	
24	TERRASIGNA SRL	**TS**	RO	
25	UBIMET GMBH	**UBIMET**	AT	
26	ELEKTRO LJUBLJANA PODJETJE ZADISTRIBUCIJO ELEKTRICNE ENERGIJE D.D.	**EKL**	SL	
27	BORZEN, OPERATER TRGA Z ELEKTRIKO, D.O.O.	**BORZEN**	SL	
28	AJUNTAMIENTO DE SANT CUGAT DEL VALLES	**AJSCV**	ES	
29	ELES, D.O.O., SISTEMSKI OPERATER PRENOSNEGA ELEKTROENERGETSKEGA OMREŽJA	**ELES**	SL	
30	E-LEX - STUDIO LEGALE	**ELEX**	IT	
31	OSMANGAZI ELEKTRIK DAGITIM ANONIM SIRKETI	**OEDAS**	TR	
32	VEOLIA SERVICIOS LECAM SOCIEDAD ANONIMA UNIPERSONAL	**VEOLIA**	ES	
33	STICHTING EG	**EGI**	NL	
34	CINTECH SOLUTIONS LTD	**CN**	CY	
35	EMOTION SRL	**EMOT**	IT	

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Contents

1 Introduction 10

1.1. Purpose of the Document 10
1.2. Relation to other Documents 10
1.1. Acronyms and Abbreviations 10
1.2. Document Structure 11

2 BD4NRG Analytics Tools 12

2.1 Analytic Service Builder Module 12
2.1.1 Model Generation Services 13
2.1.2 Prediction Services 13
2.1.3 Data Processing Analytic Service 14
2.1.4 Startup Guide 14
2.1.5 Declarative and Predictive Data Analytics 17
2.2 Pilot Analytic Services 19
2.2.1 Photovoltaic Production Forecasting Module 19
2.2.2 Load Forecasting Module 22
2.2.3 Water Pumping Systems Load Shifting Module 25
2.2.4 Energy Efficiency Investments Assessment Module 27

3 BD4NRG Data Visualisation and Visual Analytics Tool 31

3.1 Overview 31
3.2 Graphical User Interface 31

4 Conclusions 36

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Figures

Figure 1: fit method 16
Figure 2: predict method 16
Figure 3: process method 16
Figure 4: try/except 17
Figure 5: PV Production Forecasting Module Component Diagram 21
Figure 6: Load Forecasting Module Component Diagram 24
Figure 7: Water Pumping systems Load Shifting Module Component Diagram 26
Figure 8: Energy Efficiency Investments Assessment Module Component Diagram 29
Figure 9: Build query user interface 32
Figure 10: Query Results using Apache Superset for Pilot 8 dataset 33
Figure 11: Build Query, using SQL Lab 33
Figure 12: Time Series Bar Chart - Average Power per day 34
Figure 13: Power consumption per hour of day pie chart 34
Figure 14: Power Consumption per weekday pie chart and customisation options 35
Figure 15: Dashboard for Pilot 8 dataset 35

Tables

Table 1: Acronyms and abbreviations 10
Table 2: PV Production Forecasting RESTFUL interfaces 21
Table 3: Load Forecasting RESTFUL interfaces 24
Table 4: Water Pumping Systems Load Shifting RESTFUL interfaces 26
Table 5: Energy Efficiency Investments Assessment RESTFUL interfaces 29

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Executive Summary

In the IoT era, the high volume of data obtained from various smart grid sources satisfies the characteristics of big data in terms of Volume, Velocity, Variety, Veracity, and Value. These characteristics are challenging when dealing with big data analytics (BDA) along with other major concerns such as security, privacy, etc. There are various potential applications of BDA in the context of smart grids, such as asset management, load management with demand response, detection of outages due to faults and anomalies, load and generation forecast under high unpredictability, and real-time and automatic processing of the electrical consumers' energy consumption, automatic billing, intelligent energy planning and pricing analysis, etc.

BD4NRG aims at delivering an innovative framework that leverages on AI-based cross-sector analytics for integrated and optimised smart energy grid management, based on seamless data-information-knowledge exchange under respective sovereignty and regulatory principles. For this purpose, BD4NRG will release a BDA Energy Toolbox, allowing energy stakeholders to design, develop, and run custom BDA applications by leveraging on reusable tools, including analytics applications and machine learning (ML) models/algorithms, made available by BD4NRG ecosystem members.

In this document, which has to be intended as the accompanying report of the deliverable D4.6 - "BD4NRG-PROCESSING/ANALYTICS - First. Technology Release", a description of the first technology release of BD4NRG tools for BDA and visualization given as well as the user guidelines will be provided.

The first technology release includes the Analytic Service Builder, as part of the Analytics Toolbox, which module allows data scientists, IT operators and in general BDA service providers to create, register into the Toolbox catalogue, and manage BDA applications as microservices. As part of the deliverable D4.6, some services that take advantage of the Analytic Service Builder capabilities have been developed and described in this document. Those services are ML services capable of producing a model and using it for batch predictions.

The first technology also includes the analytics tools implementing the pilot analytic services developed so far in the context of Work Package (WP) 4 - "Data Modelling & Open Modular AI-based edge-level Analytics Toolbox":

- Photovoltaic Production Forecasting - Decision-tree-based models were selected to produce energy forecasts, being regarded as one of the most accurate, flexible, and relatively interpretable types of ML methods based on the literature.
- Load Forecasting - This module adheres the very important issue of load forecasting, and it has been specifically designed to face the consumption problem in small communities such as islands.
- Water Pumping Systems Load Shifting - The purpose of this module is to optimally shift the water pumping stations operating hours to hourly intervals with low expected total load demand, in order to shave peaks.
- Energy Efficiency Investments Assessment - This module exploits ML methods and data analysis techniques in order to provide a data-driven assessment of investments on EE.

The model used by those tools will be later integrated in the Analytics Toolbox.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Finally, as part of the first technology release, the preliminary version of Data Visualisation and Visual Analytics service is described in this document, focusing mostly on the application perspective and the offerings of Apache Superset, as the solution that was selected as the basic visualisation dashboard and reporting tool.

The integrated view of the available tools and ML algorithms developed within the BD4NRG Consortium as a modular toolbox, envisioned in Task 4.7 “Open Modular Analytics Toolbox & Deployment”, will be provided as deliverables D4.7 - “BD4NRG-PROCESSING/ANALYTICS - Second Technology Release” and D4.8 - “BD4NRG-PROCESSING/ANALYTICS – Final Technology Release” and described in the related accompanying reports.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

1 Introduction

1.1. Purpose of the Document

This document is intended as the accompanying report of the deliverable D4.6 "BD4NRG-PROCESSING/ANALYTICS - First. Technology Release" is produced within the WP 4 "Data Modelling & Open Modular AI-based edge-level Analytics Toolbox". The purpose of this document is to present the first technology release of big data processing/analytics implemented in the context of Task 4.3 "Incremental, Distributed & Self-learning Analytics", Task 4.5. "Declarative, Predictive & Prescriptive Real-Time Data Analytics", and Task 4.6 "Visual Analytics, Dashboard and Report Libraries for Energy Domain".

The integrated view of the available tools and ML algorithms developed within the BD4NRG Consortium as a modular toolbox, envisioned in Task 4.7 "Open Modular Analytics Toolbox & Deployment", will be provided as deliverables D4.7 "BD4NRG-PROCESSING/ANALYTICS - Second Technology Release" and D4.8 "BD4NRG-PROCESSING/ANALYTICS – Final Technology Release".

The source code of the BD4NRG software deliverables is stored in the BD4NRG GitLab repository; credentials for accessing the repository can be obtained by filling in the contact form in the BD4NRG website.

1.2. Relation to other Documents

This document relies on the deliverable D2.2 - "BD4NRG Open APIs and Data Assets Functional Specifications", which provides systematic description of the structure and functionalities of the BD4NRG components as part of the system requirements and specifications activities. Additional information on BD4NRG-specific systems is provided by the deliverable D2.5 - "BD4NRG Reference Architecture First Version", which describes a first version of the BD4NRG Reference Architecture that integrates all use cases into one architectural model. Additionally, this document relies on the deliverable D4.1 - "BD4NRG High Performance Data Processing", where BD4NRG approach to BDA and ML has been explained.

1.1. Acronyms and Abbreviations

Table 1: Acronyms and abbreviations

Acronym	Description
AI	Artificial Intelligence
ANN	Artificial Neural Network
BDA	Big Data Analytics
BDAaaS	BDA-as-a-Service
EE	Energy Efficiency

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

k-NNC	KNearest Neighbours Classifier
ML	Machine Learning
PV	Photovoltaic
RES	Renewable Energy Sources
SQL	Structured Query Language
WP	Work Package

1.2. Document Structure

This document is organized as follows. Chapter 2 describes the Analytic Service Builder module of the Analytics Toolbox, which allows the registration of new Big Data Analytics (BDA) services in the catalogue of the Toolbox and the deletion or update of the existing ones, and provides the guidelines for the proper use. Moreover, the software modules implementing the analytic services developed so far in the context of WP4 are described. The models used by those modules will be later registered in the Analytics Toolbox catalogue. In chapter 3 Data Visualisation and Visual Analytics service is presented in detail, focusing mostly on the application perspective and the offerings of Apache Superset, as the solution that was selected as the basic visualisation dashboard and reporting tool. Finally, in Chapter 4 conclusions are given.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

2 BD4NRG Analytics Tools

The main objective of BD4NRG is to deliver an innovative framework which leverages on AI-based cross-sector analytics for integrated and optimised smart energy grid management (including operation and planning), based on seamless data-information-knowledge exchange under respective sovereignty and regulatory principles. To that end, BD4NRG will release a BDA Energy Toolbox, allowing energy stakeholders to design, develop, and run custom BDA applications by leveraging on reusable tools, including analytics applications and machine learning (ML) models/algorithms, made available by BD4NRG ecosystem members. The Analytics Toolbox will allow to combine and make available to end users user-friendly graphical working environment capabilities to support custom selection of local and/or third party assets, including: discovery of heterogeneous data sources formats, a huge variety of analytics ML algorithms/models to be selected among correlation, categorisation, and feature extraction to implement a variety of analytics types (descriptive, predictive, prescriptive), a large variety of presentation modalities (reports, charts, etc.), edge-level available storage and computing resources.

The architecture and the software components of the Analytics Toolbox have been presented in the deliverable D2.2 - *BD4NRG Open APIs and Data Assets Functional Specifications*.

Among the efforts analysed, ALIDA BDA platform[1] has been selected as a starting point for BD4NRG and it will be extended and validated in the energy domain, according to the needs that emerged in the requirements elicitation phase of the project. ALIDA embraces the paradigm of BDA-as-a-service (BDAaaS), which represents the next evolution step of Big Data.

In the following sections, the implementation of the Analytic Service Builder software component of the Analytics Toolbox will be described, as well as the user guidelines for the registration of new BDA services in the Analytics Toolbox catalogue.

Furthermore, the pilot analytic services so far implemented in the context of WP4, whose model will be later registered in the Analytics Toolbox catalogue, are described.

The source code of the Analytics Toolbox will be moved to the BD4NRG GitLab repository.

2.1 Analytic Service Builder Module

The Analytic Service Builder allows the registration of new BDA services in the ALIDA catalogue and the deletion or update of the existing ones. By means of a wizard, it must be possible to allow data scientists, IT operators, and in general BDA service providers to create, register, and manage BDA service as microservices. The aim of this tool is to automate the creation of Docker files, pushing the new images inside a repository and the registration into the BDA service catalogue.

The first version of the Analytic Service Builder consists of a series of Python libraries and examples that helps the developer to generate BDA services for ALIDA catalogue.

[1] ALIDA is a research asset by ENG, aimed to support the design, deployment, execution and monitoring of BDA applications (https://home.ALIDAlab.it/).

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

In particular, such library focuses on how to develop ML services that make use of many commonly used frameworks (e.g., Tensorflow[2], Scikit-learn[3], Pandas[4], etc.) and work on tabular datasets.

The repository is publicly available on Gitlab at https://gitlab.alidalab.it/external/templates-bda-services/tree/master/batch/alida-ml-quickstart.

The repository contains a detailed guide on how to develop for ALIDA while in this section a brief description is presented of what can be achieved using it.

2.1.1 Model Generation Services

ALIDA allows a user to create, among other applications, ML based applications. These applications are made of several small blocks called services. Each service does one small job, such as creating a Model, using it to make predictions, preprocessing some data, etc. Here it is described how to develop a service intended to generate and train a generic ML model.

Model generation through one of the many commonly available frameworks starts typically by training a model on an input dataset and then saving the resulting weights and parameters on a file or folder that represents the resulting model. The process usually also takes a series of inputs, such as hyperparameters and general parameters of the application. The Analytic Service Builder provides all the tools to read such parameters, the dataset (a Pandas dataframe), and to save the model back to ALIDA, so that it is possible to concentrate only on the ML logic. In fact, the only expected output is the path inside the container where the model has been saved.

The dataset is served to the service as a Pandas dataframe[5], a versatile object that can be easily transformed in many other data structures (such as Numpy arrays and matrices) and that is particularly suited for preprocessing.

The basic structure of a model generation service inside the folder is available at https://gitlab.alidalab.it/external/templates-bda-services/tree/master/batch/alida-ml-quickstart/Quickstart%20Empty%20Model.

2.1.2 Prediction Services

While it is possible to create a service that generates a model and uses it straight away, a good practice is to make things as modular as possible in more complex contexts. This helps especially in reusing code and avoiding the so-called boilerplate code; moreover, whilst training may happen once, prediction may happen multiple times on different datasets in a production environment or for testing model's performances. For these reasons, two separate services for a ML model can be found in the examples presented.

[2] https://www.tensorflow.org/
[3] https://scikit-learn.org/stable/
[4] https://pandas.pydata.org/
[5] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

A predictions service receives as input the application parameters, the model (saved as a file or as a folder) and a dataset. The expected output is a dataset containing the predictions, as a Pandas dataframe.

The basic structure of a service that uses a model for prediction inside the folder predict can be found at https://gitlab.alidalab.it/external/templates-bda-services/tree/master/batch/alida-ml-quickstart/Quickstart%20Empty%20Model.

2.1.3 Data Processing Analytic Service

While the heart of the previously described analytics is the model creation and usage, it is often necessary to preprocess the input datasets beforehand. For all the preprocessing operations a data preprocessing service quick start has been conceived as well. These kinds of services expect as input a dataset and the necessary parameters for the preprocessing, and return a dataset.

While the data processing analytics services are here presented as a means of developing preprocessing services, it is possible to create declarative analytics by using them.

The basic structure of a preprocessing service can be found at https://gitlab.alidalab.it/external/templates-bda-services/tree/master/batch/alida-ml-quickstart/Quickstart%20Empty%20Processor.

2.1.4 Startup Guide

In order to realize services on Alida, first make sure that the following prerequisites are satisfied:

- A Unix-like system (Linux/MacOS) or WSL2[6] on Windows.
- Docker engine[7] in order to build and push the dockerised application. Note: if you are using Windows and WSL2, be sure Docker is set up correctly.
- Python[8] (3.6 or later recommended) to locally test and run your application.
- A generic python package manager (e.g., pip[9]).
- (optional) A virtual environment (e.g., venv[10]).

Each service is composed of a docker image. Even though the ultimate goal is to build the service a working docker container, it is useful to test the code locally during the development. For this reason, it is suggested to simulate locally how the service will be used on ALIDA. Indeed, each service example contains a script that can be used to start the service and simulate the passing of the necessary arguments as it happens on the ALIDA platform; checking the resulting models saved on a local folder and the resulting datasets saved files with csv extension.

The services basic structures referenced in the previous paragraphs need to be customised only partially in order to develop a new service. Each example template includes:

[6] https://docs.microsoft.com/en-us/windows/wsl/install
[7] https://www.docker.com/get-started
[8] https://www.python.org/
[9] https://pip.pypa.io/en/stable/getting-started/
[10] https://docs.python.org/3/library/venv.html

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

- A src folder containing the python custom source code of the service.
- A requirements.txt file containing all the necessary requirements for the service to run.
- A Dockerfile defining the way the image has to be built.
- A build.sh script containing instructions on how to launch the docker command to build the image and its name and tag.
- A run.sh script containing all the arguments necessary to test a service locally.
- A generate_json.py python script used to generate a json file, that we call meta-model, used to register the service in the ALIDA platform catalogue.

2.1.4.1 Required Custom Libraries and Datasets

It is possible to install new python libraries by changing directly the requirements.txt file. It is also advised to pin the library version to prevent problems on future builds of the same service. It may also be necessary to change the run.sh file if more arguments are needed or to point to the right datasets while testing locally.

2.1.4.2 The arguments.py File

This file contains information on the non-constant parameters necessary for the service to run (e.g., the number of epochs to train a classifier, the name of the column containing the labels). These variables can be set by the user when designing a workflow on the ALIDA platform. It also contains information about the service needs for input and output datasets and models.

Code-wise, this file must contain an instance of the ALIDA custom implementation of the python class *ArgumentParser*. This implementation supports many of the original functionalities of the official python implementation[11]. Some functionalities have been added in order to allow access to other needed inputs such as models and datasets. Please check the readme file for a detailed explanation.

2.1.4.3 The Custom Class contained in the src Folder

This file contains the heart of your algorithm. It must contain a custom class that implements one of two possible ALIDA abstract classes:

- A bdaservicetemplate.Model: a ML model class containing a fit and a predict abstract methods.
- A bdaservicetemplate.Processor: a generic class containing a processor abstract method. This class can make modifications on the input dataset and transform it to an output dataset.

If you are developing a ML model then you have to implement the first class and its methods.

The *fit* method accepts as input a Pandas dataset and returns the path where the resulting model has been used (Figure 1).

[11] https://docs.python.org/3/library/argparse.html

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

```python
@Service.dataset_to_model
def fit(self, dataset: pd.DataFrame):
    # Training code goes here
    # ...
    # Code to save the model to a file or folder goes here
    # ...
    return model_path
```

Figure 1: fit method

The *predict* method accepts as input a dataset, the path of the model and return a pandas dataframe (Figure 2).

```python
@Service.dataset_and_model_to_dataset
def predict(self, dataset: pd.DataFrame, model_path: str):
    # Loading model code goes here
    # ...
    # Prediction code goes here
    # ...
    return resulting_dataframe
```

Figure 2: predict method

If you are developing a processor, then you have to implement the *bdaservicetemplate.Processor* abstract class. This class contains only an abstract method that receives a pandas dataframe as input and returns a pandas dataframe as output, as it is shown below.

```python
@Service.dataset_to_dataset
def process(self, dataset: pd.DataFrame):
    # Your processing code goes here
    # ...
    return resulting_dataframe
```

Figure 3: process method

This folder contains also a *main.py* file. This file must import the previously described class and it calls its main argument (fit or predict if it is a Model, process if it is a Processor). To be noted that:

- It is not necessary to pass the arguments (models and datasets) to the function, ALIDA libraries do it (even locally).
- The import of a custom class is done inside a try/except (Figure 4). This also allows to monitor import errors while running the service on the cluster.

An example of a *main.py* file is shown below.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

```
from bdaserviceutils import TaskStatus
import traceback

status = TaskStatus()
status.running()

try:
    from custom_class import Custom_Class

    custom_class = Custom_Class()

    custom_class.fit()

    status.completed()

except Exception as exp:
    error_message = 'Task failed: {}'.format(exp)
    status.failed(1, error_message, error_message)
    traceback.print_exc()
```

Figure 4: try/except

2.1.4.4 The generate_json.py File

This script is the key to translate your locally working service into an ALIDA service. As described in the section about the *arguments.py* file, the *ArgumentParser* class has been extended in order to be able to exploit the work done by you defining arguments and parameters to make your code automagically compliant with ALIDA requirements. The generate_json.py file does exactly that, based on the file containing all the input and output parameters, it generates a json file that you can drop on ALIDA and register your service instantly.

2.1.5 Declarative and Predictive Data Analytics

As part of this deliverable, some services that take advantage of the analytic Service Builder capabilities have been developed. Those services are ML services capable of producing a model and using it for batch predictions.

2.1.5.1 ANN Regression Model Fit

This service implements an algorithm capable of generating a ML regression model using an artificial neural network (ANN). In a regression problem, the aim is to predict the output of a continuous value, like a price or a probability. Unlike linear regression models, the use of an ANN allows making predictions exploiting nonlinear relationships between the input features of the dataset.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Tensorflow[2] has been used as a ML framework; Tensorflow is a free and open-source software library for ML that allows to design many neural networks custom architectures, train them, and use them for predictions.

In order to create a model by using this service on the ALIDA platform, model hyperparameters and application parameters have to be provided:

- The portion of the data that has to be used for training the regressor must be specified. If "1" is specified, then all the data can be used for training. This parameter comes handy in case of a small dataset that want to be used for both training and testing purposes.
- It is also possible to set a random seed. This helps to ensure that each time the service is executed, it is always possible to have the same resulting test and train datasets after splitting them. This parameter must be set if consistent results are needed.
- The validation set size must be specified, a portion of the training set used for validation during training.
- The name of column that contains the label.
- The number of epochs needed to train the classifier.

This service includes tests on it. It has been tested on the publicly available *Auto MPG* dataset and demonstrates how to build models to predict the fuel efficiency of the late-1970s and early 1980s automobiles. This dataset includes attributes like cylinders, displacement, horsepower, and weight and contains also the label: the fuel consumption of the car.

The implementation of this service is available at https://gitlab.alidalab.it/external/templates-bda-services/tree/master/batch/alida-ml-quickstart/Quickstart%20Tensorflow/fit .

2.1.5.2 ANN Regression Model Predict

As well as the service capable of creating a model, a service capable of using it for predictions has been developed. It shares many of the parameters of the first service, except the number of epochs and the validation set size. It also needs a parameter to specify what it is going to be the name of the column containing the resulting predictions.

You can create an example workflow by putting first an ANN regression model fit service. Then, the resulting model can be fed to an ANN regression model predict service. Finally, a service that helps evaluating model performances can be placed on the workflow, to calculate accuracy and precision of the newly trained model.

The implementation of this service can be found at https://gitlab.alidalab.it/external/templates-bda-services/tree/master/batch/alida-ml-quickstart/Quickstart%20Tensorflow/predict .

2.1.5.3 KNearest Neighbours Classifier Model Fit

This service implements the KNearest neighbours classifier (k-NNC). The k-NNC is a non-parametric classification method that takes as input k closest training examples in a data set and returns as output a class membership.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

The framework used is Scikit-learn[3], a very commonly used python framework that implements many ML and statistical algorithms.

As stated before, a good practice when developing building blocks for ALIDA workflows is to make small reusable blocks. For this reason, two services have been implemented to train and use the model. The fist service consists of the fitting service: based on an input dataset and input parameters the service produces and saves a model on ALIDA. The parameters are the following:

- The number of neighbors used for training.
- The name of the column containing the prediction.

As for the regressor, local tests are included. The dataset used for the test is the famous Iris dataset, consisting of a multivariate data set containing measurements of flower petals of different iris cultivars.

The implementation of this service is available at https://gitlab.alidalab.it/external/templates-bda-services/tree/master/batch/alida-ml-quickstart/Quickstart%20Scikit-learn/fit .

2.1.5.4 KNearest Neighbours Classifier Model Predict

This service implements a prediction algorithm capable of exploiting the model generated by the previous algorithm. An example workflow to test it can be created by adding in series a k-NNC Model Fit, a k-NNC Model Predict; an evaluation service such as a multiclass-metrics can be added as well.

The implementation of this service is available at https://gitlab.alidalab.it/external/templates-bda-services/tree/master/batch/alida-ml-quickstart/Quickstart%20Scikit-learn/predict .

2.2 Pilot Analytics Services

As anticipated in the introduction of the chapter, the pilot analytic services so far implemented in the context of WP4, whose model will be later registered in the Analytics Toolbox catalogue, are described in the following sub-sections.

2.2.1 Photovoltaic Production Forecasting Module

Renewable energy sources (RES) power production and decision support related with energy management, pricing, and optimisation has been the object of vast research effort. More specifically, for wind and solar power that comprise the most common and cost-efficient types of RES, a significant number of forecasting methods have been developed over the years to forecast RES production. Decision-tree-based models were selected to produce energy forecasts, being regarded as one of the most accurate, flexible, and relatively interpretable types of ML methods based on the literature (Izza et al., 2020)[12]. This family of models is based on recursively partitioning the data space according to some features (independent or explanatory variables) and fitting a simple prediction model within each partition. The leaves of the tree are connected to specific classes which are related to the most suitable target value. Predicted new data can be achieved performing a navigation starting from the root of the tree down to a leaf, based on the outcome of the conducted comparisons along each node of the tree.

[12] Izza, Y., Ignatiev, A., & Marques-Silva, J. (2020). On explaining decision trees. arXiv preprint arXiv:2010.11034.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

This module considers three different implementations of decision trees: (a) the *DecisionTreeRegressor* of the Python Scikit-learn library[13], (b) the *XGBoost*[14] tree boosting system, and (c) the *LightGBM* gradient boosting decision tree algorithm[15]. Note that all three implementations of decision-tree-based models are very popular among the ML and forecasting community. The output of this module can be given by the independent forecasts of each implementation (Scikit-Learn, XGBoost or LightGBM), or the average value of the three forecasts which can be combined in an ensemble model.

More specifically, DecisionTreeRegressor is one of the most commonly used implementations of decision trees, having been applied to various forecasting problems with good results. The model involves several hyper-parameters that are essential for the generated forecasts. XGBoost is an ensemble method based on decision trees that uses a gradient boosting approach to generate robust forecasts, supporting various objective functions, including regression, classification, and ranking. XGBoost has been utilised in order to solve several real-world scale problems, minimizing the necessary number of resources. Similar to DecisionTreeRegressor, XGBoost involves several hyper-parameters which affect the precision of the forecasts. LightGBM is another novel gradient boosting decision tree ML algorithm, enhancing the state-of-the-art performance in many ML tasks in terms of efficiency and accuracy. The algorithm has been tested in various application in the energy sector, such as wind power forecasting and electric load forecasting, among others. LightGBM, like the above-mentioned algorithms, involves several parameters which are crucial for its behavior.

The selection of the model features has not been a trivial procedure. Several datasets have been inspected and the most dominant features in terms of correlation have been incorporated to the applied models. It has been spotted that photovoltaic (PV)-based electricity production is larger in the summer, as well as in April, May, and September, being significantly lower in the winter. Moreover, most time-series of PV production display strong daily seasonality, driven by weather conditions. For instance, due to the daily solar radiation seasonality, electricity production peaks at midday and diminishes in the early morning and night hours. The training dataset must definitely include weather forecasts which can be exploited to accurately predict solar power. Specifically, weather forecasts may refer to solar radiation (measured in W/m^2), temperature (measured in ◦C), cloud coverage (ranging from 0 to 8, with the former value suggesting no coverage and the latter value complete coverage), and wind speed (measured in m/s) predictions made on a weekly basis.

As a result, the features of the developed models are presented as follows:

- Temperature
- Cloud Coverage
- Solar Radiation
- Month of the Year
- Hour of the Day
- Last Week Production during the same hour
- Average Production of the last 7 days during the same hour

[13] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html
[14] https://xgboost.readthedocs.io/en/stable/index.html
[15] https://lightgbm.readthedocs.io/en/latest/

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

2.2.1.1 Component Diagram

The component diagram of the PV Production Forecasting Module is shown in the following Figure 5.

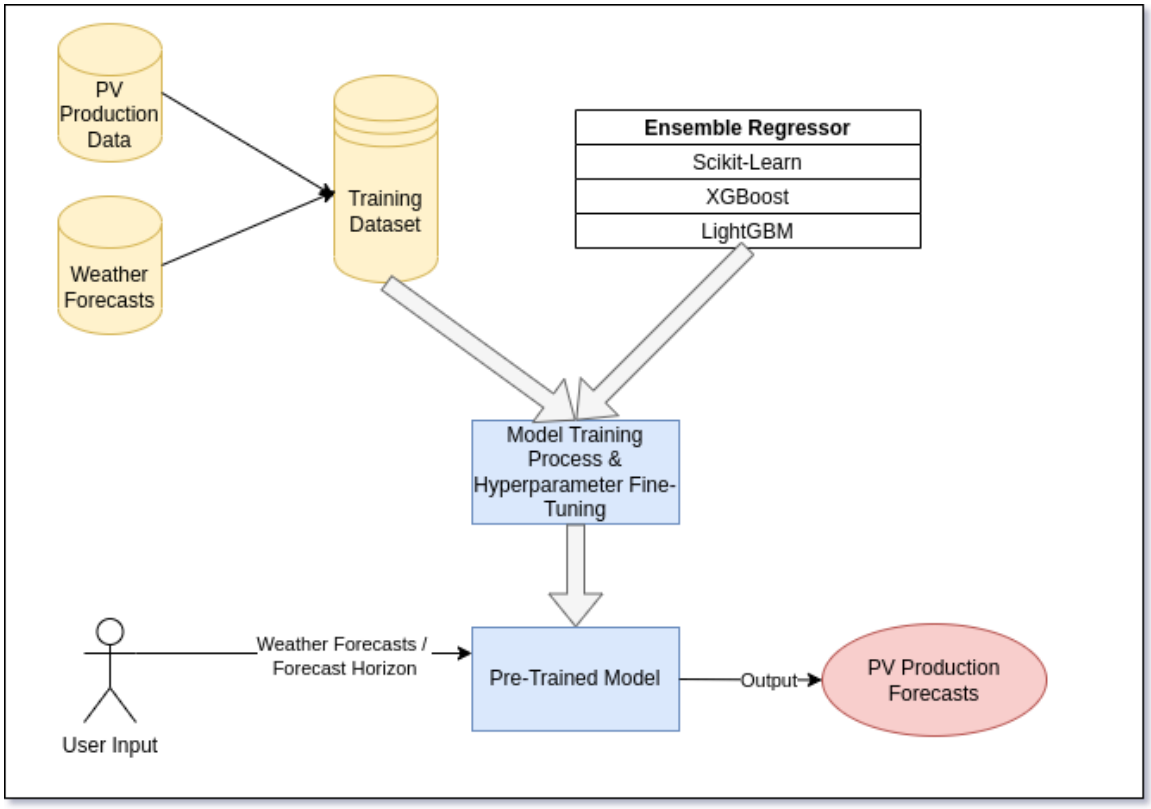


Figure 5: PV Production Forecasting Module Component Diagram

2.2.1.2 Interfaces

Table 2 describes the RESTFUL interfaces exposed by PV Production Forecasting.

Table 2: PV Production Forecasting RESTFUL interfaces

PV Production Forecasting interfaces	
Description	Restful interface which provides hourly PV production forecasts for the horizon given as input. The input features include temperature, cloud coverage, radiation, month, hour, last week's production and last week's average production for each point forecast.
End-point URL	http://0.0.0.0:6000/decision_tree http://0.0.0.0:6000/lighgbm http://0.0.0.0:6000/xgboost http://0.0.0.0:6000/ensemble
Allowed HTTP Methods	POST

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

PV Production Forecasting interfaces	
JSON Header Example	{ "horizon": 3, "data": [{ "Temperature": 35.2, "Cloud Coverage": 7.5, "Radiation": 385, "Month": "June", "Hour": 11, "Last_week_production": 2000, "Last_week_average_production": 3561 }, { "Temperature": 25.4, "Cloud Coverage": 9.5, "Radiation": 485, "Month": "July", "Hour": 12, "Last_week_production": 3000, "Last_week_average_production": 2834 }, { "Temperature": 45.4, "Cloud Coverage": 2.5, "Radiation": 885, "Month": "August", "Hour": 10, "Last_week_production": 4000, "Last_week_average_production": 3834 }] }
Result Example of **/decision_tree**	{ "PV Production": "[2753.20991826 2790.48412262 3715.83377495]", "message": "" }

The source code of the APIs is available at https://gitlab.com/bd4nrg-technology-release/analytics-toolbox/pilot-analytics-services/-/tree/main/Photovoltaic%20Production%20Forecasting%20Module .

2.2.2 Load Forecasting Module

This module adheres the very important issue of load forecasting, and it has been specifically designed to face the consumption problem in small communities such as islands. Load forecasting is critical for effective power system operation and planning. Depending on the forecasting horizon considered, that may span from minutes to years, different types of data and models may be required for designing accurate forecasting solutions (Petropoulos et al., 2021)[16]. In operations management settings, short-term forecasts, spanning from 1 hour to 168 hours ahead, are the most useful ones as they support decisions related with bidding, portfolio optimisation, and tariff design, as well as energy conservation techniques, such as load shifting, peak shaving, energy-storing, and load balancing.

In this module, the same decision trees implementation of the previous module has been exploited. Although decision trees were originally developed for solving classification tasks, they also suit well in regression tasks where the dependent variable takes continuous values. The loss function used during the training process for the implementation of such algorithms is the mean squared error (difference between the actual and the forecasted values). For the implementation of this module the above-mentioned three implementations of gradient boosting and decision trees have been combined: (a) the DecisionTreeRegressor of the Python Scikit-learn library, (b) the XGBoost tree boosting system, and (c) the LightGBM gradient boosting decision tree algorithm. The final forecasts are produced by combining (averaging) the forecasts generated by each of the three models. This strategy has been followed because ensembling has long been considered an effective strategy for improving forecasting accuracy and mitigating model and parameter uncertainty. Alternatively, a method selecting the best model per hourly interval could be implemented, but ensembling has been proven more precise and

[16] Petropoulos, Fotios, et al. "Forecasting: theory and practice." arXiv preprint arXiv:2012.03854 (2020).

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

robust for the purpose of the study.

Although a similar strategy has been employed for the selection of the appropriate models, the problem of load forecasting differs significantly from the PV production forecasting. All the inspected time-series include strong yearly seasonal patterns. More specifically, in the island of Tilos, which is the time-series based on which the model has been trained, the total electricity consumption is higher in the summer and especially in July and August, probably due to the increased tourism activity and air-conditioning use, being significantly lower in the rest of the year. Yet, December and January do display some notable peaks. In addition, the series shows strong daily seasonality, driven by human behaviour. Electricity consumption is also characterised by strong seasonal patterns, observed at both a daily and an annual level. We have observed that auto-regression, i.e., using past electricity consumption observations as features to forecast future ones, is a promising forecasting approach, especially if categorical variables like month, week, weekday, and hour of the day are included among the features to facilitate pattern recognition. Therefore, the aforementioned features (lag 168 and calendar variables) are the only regressors considered by the tree-based model to forecast electricity consumption.

As a result, the features of the developed models are presented as follows:

- Month of the Year
- Hour of the Day
- Day of the Week
- Yesterday's Consumption during the same hour
- Last Week's Consumption during the same hour

2.2.2.1 Component Diagram

The component diagram of the Load Forecasting Module is depicted in the following Figure 6.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

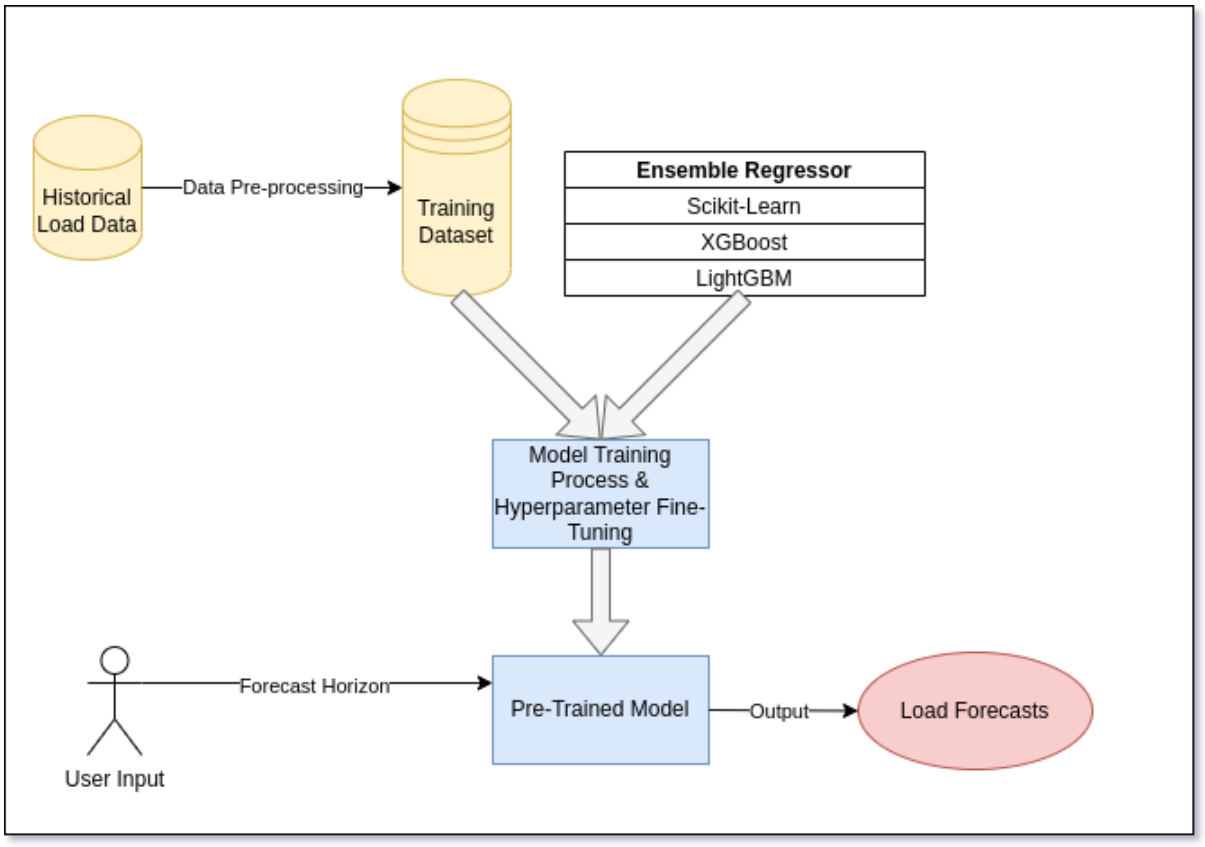


Figure 6: Load Forecasting Module Component Diagram

2.2.2.2 Interfaces

Table 3 describes the RESTFUL interfaces exposed by Load Forecasting.

Table 3: Load Forecasting RESTFUL interfaces

Load Forecasting interfaces	
Description	Restful interface which provides hourly consumption forecasts for the horizon given as input. The input features include month, day, hour, yesterday's consumption and last week's consumption.
End-point URL	http://0.0.0.0:7000/decision_tree http://0.0.0.0:7000/lighgbm http://0.0.0.0:7000/xgboost http://0.0.0.0:7000/ensemble
Allowed HTTP Methods	POST
JSON Header Example	{ "horizon": 2, "data": [{ "Month": "July", "Hour": 12, "Day": "Friday", "Yesterday Load": 3000, "Last Week Load": 2834 }, { "Month": "August",

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Load Forecasting interfaces	
	"Hour": 15, "Day": "Saturday", "Yesterday Load": 4100, "Last Week Load": 3434 }] }
Result Example of **/decision_tree**	{ "Consumption": "[820.88333333 820.88333333]", "message": "" }

The source code of the APIs is available at https://gitlab.com/bd4nrg-technology-release/analytics-toolbox/pilot-analytics-services/-/tree/main/Load%20Forecasting%20Module .

2.2.3 Water Pumping Systems Load Shifting Module

This module addresses the problem of optimal scheduling shiftable loads. In the specific scenario, the shiftable loads are described by water pumping stations, while external parameters such as Renewable Energy Production and total consumption are taken into consideration. The purpose of the algorithm is to optimally shift the water pumping stations operating hours to hourly intervals with low expected total load demand, in order to shave peaks. Two scenarios are considered for the development of the optimisation algorithm, attempting to simulate both manual and automatic operation.

In the first scenario, the operating plan of the pumping stations is unrestricted, which can be interpreted as a flexibility to start and stop the pumping stations at any hour of the day. The second scenario simulates a constrained operating plan for the pumping stations, which means that the operation of a station should be continuous and it can only stop when the daily required hours are completed. In both scenarios the optimisation algorithm requires the following input:

- The number of available pumping stations (N).
- The number of required operating hours for each pumping station during a day (P Hours).
- The hourly electricity consumption (installed power) of each pumping station (P Cons).
- The expected electricity consumption of the 24-hourly intervals where the water pumping stations will be assigned (T Cons).

The operation scheduling of the pumps is designed targeting to shave the load demand peaks at the island grid level. In this respect, the algorithm selects the pumping station operated at the maximum net electricity load point (i.e., total load minus PV-based electricity production) and it assigns the respective pump to the interval with the minimum hourly load according to the island energy consumption forecasts, as well as the PV production forecasts. After the mapping of the selected pumping station to the selected interval has been completed, the algorithm proceeds with the necessary updates on the variables T Cons, which describes the total consumption of the island, and P Hours, which includes the remaining hours that each pumping station must operate. The whole process is repeated until all operating hours of every pumping station have been mapped to the optimal interval.

In the second scenario, where the operation of the pumps is constrained, the optimisation algorithm is partially altered providing a solution which may not be the optimal in some cases, but it approximates the optimal solution of the problem regarding peak shaving. Firstly, the algorithm selects the pumping station with the maximum hourly consumption, as the algorithm of the first scenario

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

does. The difference in this approach is that the algorithm selects a multi-hour interval, equal to the pumping station daily operating hours, because of the continuous operation constraint. More specifically, the algorithm explores all possible multi-hour intervals aiming to find the hourly interval with the maximum energy consumption in each one. Finally, it maps the pumping station to the multi-hour interval which includes the minimum of these values, attempting to prevent peaks from happening. This procedure resumes until all pumping stations have been mapped to a multi-hour interval. The dynamic allocation of pump stations to hourly intervals ensures that no new peaks will be created throughout the process, because of the algorithm property to update the net load of the selected hourly interval after allocating a new pump to it.

2.2.3.1 Component Diagram

The component diagram of the Water Pumping Systems Load Shifting Module is depicted in the following Figure 7.

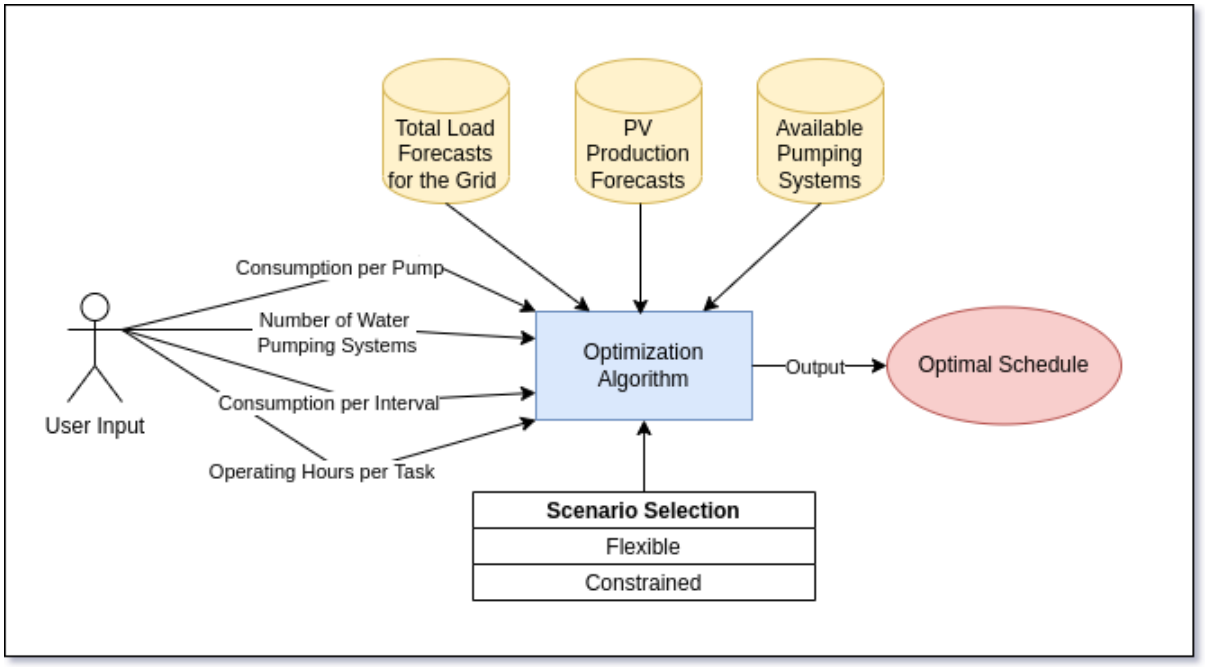


Figure 7: Water Pumping systems Load Shifting Module Component Diagram

2.2.3.2 Interfaces

Table 4 describes the RESTFUL interfaces exposed by Water Pumping Systems Load Shifting.

Table 4: Water Pumping Systems Load Shifting RESTFUL interfaces

Water Pumping Systems Load Shifting interfaces	
Description	Restful interface that provides the proposed operating plan of pumping stations. Input: the initial consumption of the system and for each pump the operating hourly consumption and the operating hours. The output is the proposed plan of the pumping systems. Two scenarios are implemented: restricted and unrestricted.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Water Pumping Systems Load Shifting interfaces	
End-point URL	http://0.0.0.0:8000/restricted http://0.0.0.0:8000/unrestricted
Allowed HTTP Methods	POST
JSON Header Example	{ "hourly_demand": [3429.89, 3604.79, 8658.9, 9178.99, 6951.84, 4396.57, 3727.44, 8401.05, 8257.24, 1537.0, 5151.35, 8351.29, 1739.22, 5346.4, 9501.9, 4811.58, 5148.05, 5807.76, 8903.67, 7208.12, 2145.0, 6589.95, 2441.51, 1607.81], "number_of_pumps": 3, "pumps": { "pump1": { "pump_operating_hours": 6, "pump_consumption": 500.24 }, "pump2": { "pump_operating_hours": 14, "pump_consumption": 5000.41 }, "pump3": { "pump_operating_hours": 2, "pump_consumption": 300.25 } } }
Result Example of **/restricted**	{ "proposed_plan": { "pump1": ["16:00", "17:00", "18:00", "19:00", "20:00", "21:00"], "pump2": ["1:00", "2:00", "3:00", "4:00", "5:00", "6:00", "7:00", "8:00", "9:00", "10:00", "11:00", "12:00", "13:00", "14:00"], "pump3": ["23:00", "24:00"] }, "message": "" }

The source code of the APIs is available at https://gitlab.com/bd4nrg-technology-release/analytics-toolbox/pilot-analytics-services/-/tree/main/Water%20Pumping%20Systems%20Load%20Shifting%20Module .

2.2.4 Energy Efficiency Investments Assessment Module

This module exploits ML methods and data analysis techniques in order to provide a data-driven assessment of Energy Efficiency (EE) investments. The recommendation can be either binary (invest vs. do not invest) or multi-class (e.g., the potential of the investment is low, medium, or high). Based on the existing literature, traditional, statistical classification methods, such as the ordinal logit, the ordinal probit, and the linear discriminant analysis methods along with a limited number of ML methods, such as the k-nearest neighbours and support vector machines have been applied to assess EE investments[17]. In this regard, this module additionally considers the Gaussian naive Bayes, extreme gradient boosted trees, and random forest methods in an attempt to improve the accuracy of the predictions made through methods of higher learning capacity. In addition, the deployed module uses a meta-learner aiming to identify the most accurate classification method for each investment based on its particular characteristics, thus allowing further optimisation and increasing the potential value added by the proposed classification process.

This module provides a grant financing recommendation for future EE projects by learning from a pool

[17] Doukas, Haris, et al. "How Successful are Energy Efficiency Investments? A Comparative Analysis for Classification & Performance Prediction." Computational Economics (2021): 1-20.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

of already implemented projects with similar characteristics. The basis of the proposed approach is a stacking ensemble classification model that builds on five baseline ML classification methods, namely k-Nearest Neighbors, Gaussian Naive Bayes, Extreme Gradient Boosted Trees, Random Forest, and Support Vector Machine. The train set of the classifiers is a set of projects from Latvia that involves information about the renovations performed in a variety of buildings during the last decade. This includes data that was originally available before implementing the renovation (used as features) and data recorder after the investment has been completed (used for labeling).

Specifically, the features of the investment involve specific attributes of the projects that, in our case, are the renovated building age, number of apartments, total heating area, and current energy consumption, the expected CO_2 decrease, and the renovation cost. The output of the baseline classifiers are probabilities that the investment belongs to each discrete class among the following three:

- Class A: The project should be financed.
- Class B: The project should be partially financed.
- Class C: The project should not be financed.

In order to label past investments, the realised utility of each project is computed using a measure of choice. Consequently, the top performing projects are labeled as Class A, the worst performing projects as Class C, while the rest as Class B. Having labeled the investments, the baseline classification methods are used for classifying past projects and their predictions are stored. Using these predictions as input along with the actual labels of the investments and the features originally used by the baseline classifiers, a meta-learner, namely a logistic regression classifier, is trained with the objective to predict which of the five baseline methods is the most appropriate for predicting the class of each project. Given this insight, the "optimal" classifier is used for each investment to derive the probability of each class. These probabilities are finally combined in order to provide a recommendation on the percentage of grant financing for a future investment.

Finally, the proposed model offers recommendations about the percentage of grant financing that future investments should receive. If the predicted class of the meta-learner was used for determining this percentage, then the suggestions of the decision-support system would essentially be discrete numbers, getting one the three possible percentages assigned to each class. For instance, if we assumed that Class A suggested 100% financing, Class B 50% financing, and Class C 0% financing, then all projects would be financed by a factor of $f_A = 1$, $f_B = 0.5$, or $f_C = 0$, respectively. It becomes evident that this approach significantly limits the value added by the methodology, also ignoring the uncertainty around the predictions. For example, assume a scenario where an investment is labeled with a probability of 0.55 to Class A, 0.35 to Class B, and 0.10 to Class C.

The features that are used as input to the classifiers are listed below:

- Investment Cost
- Building Construction Year
- Planned CO_2 Reduction due to the Renovation
- Electricity Consumption before the Actions
- Total Heating Area.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

2.2.4.1 Component Diagram

The component diagram of the Water Pumping Systems Load Shifting Module is depicted in the following Figure 8.

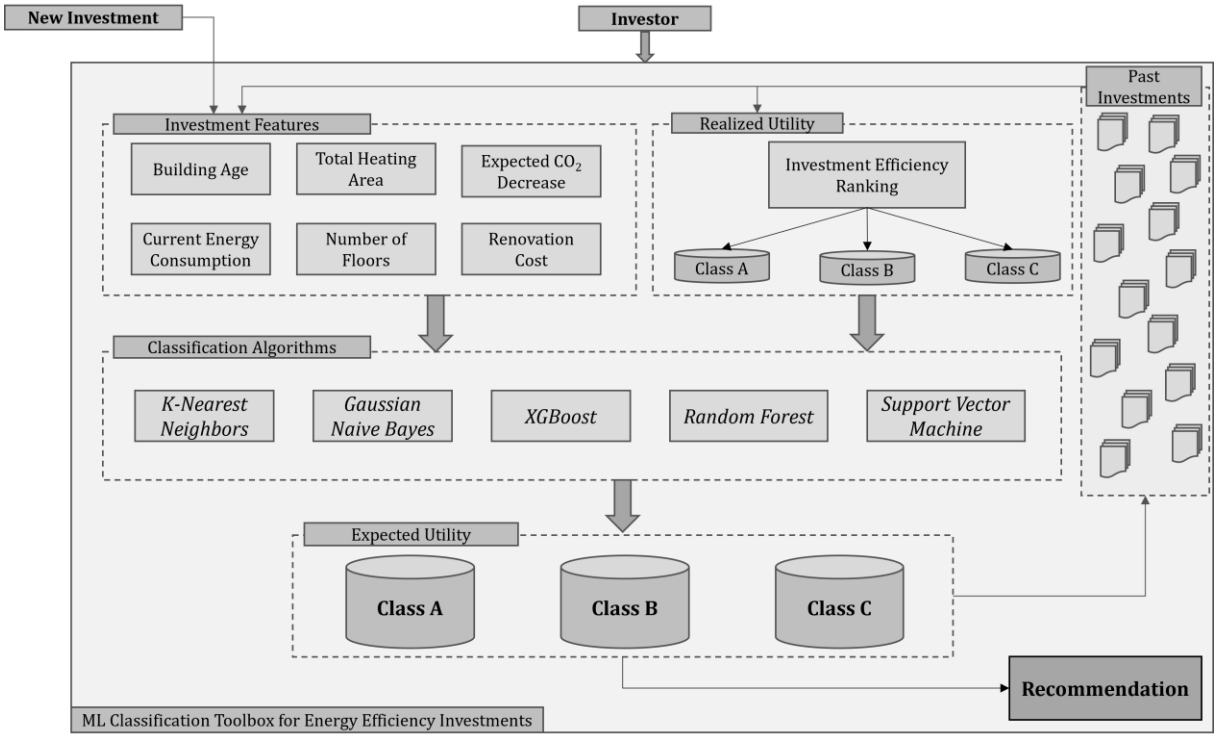


Figure 8: Energy Efficiency Investments Assessment Module Component Diagram

2.2.4.2 Interfaces

Table 5 describes the RESTFUL interfaces exposed by Load Forecasting.

Table 5: Energy Efficiency Investments Assessment RESTFUL interfaces

Energy Efficiency Investments Assessment interfaces	
Description	Restful interface that provides recommendations for EE investments. Input: The investment cost, the building construction year, the planned CO2 reduction, the consumption before the renovation and the total heating area of the building. The output is either the distinct class of the renovation (A, B, or C), or the probabilities of each class. Finally, the /recommendation URL supports a final recommendation on the financing percentage that the investment should take.
End-point URL	http://0.0.0.0:5000/prediction http://0.0.0.0:5000/probabilities http://0.0.0.0:5000/recommendation

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

Energy Efficiency Investments Assessment interfaces	
Allowed HTTP Methods	POST
JSON Header Example	{ "Cost": 367882.94, "Building Year": 1972, "Planned CO2 Reduction": 95.767, "Consumption before": 578.703, "Total Heating Area ": 2834 }
Result Example of **/prediction**	{ "Class Prediction": "1", "message": "" }

The source code of the APIs is available at https://gitlab.com/bd4nrg-technology-release/analytics-toolbox/pilot-analytics-services/-/tree/main/Energy%20Efficiency%20Investments%20Assessment%20Module .

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

3 BD4NRG Data Visualisation and Visual Analytics Tool

In this chapter, the first version of the Data Visualisation and Visual Analytics service is presented in detail, focusing mostly on the application perspective and the offerings of Apache Superset, as the solution that was selected as the basic visualisation dashboard and reporting tool.

3.1 Overview

The Visual Analytics service objective is to provide the interface for interaction between the BD4NRG platform data and the end users that intend to explore, analyse and draw significant inferences through intuitive and interactive data visualisations. It enables the exploration and analysis of different datasets by providing several utilities to query and visualise the requested data. Moreover, it provides the ability for the end-users to build reports and dashboards that combine different visualisations, text and external information that they plan to insert.

The main technology that was selected for this service is Apache Superset[18], which is a modern and powerful data exploration and visualisation platform. Specifically, Superset is fast, lightweight, intuitive, and loaded with options that make it easy for users of all skill sets to explore and visualise their data, from simple line charts to highly detailed geospatial charts, as reported also on Superset official website. In more detail, it was selected as it provides quick and easy integration with several databases, such as Structured Query Language (SQL) based data sources through SQLAlchemy[19], including no sql ones, and enables visualisations and queries to data, without writing any code. Moreover, it is lightweight and highly scalable, leveraging the power of existing data infrastructures without requiring yet another ingestion layer.

What is missing from Apache Superset in the context of Task 4.6 *Visual Analytics, Dashboard and Report Libraries for Energy Domain* is the utility of building custom services except visualisations, and using other services of the BD4NRG platform. This functionality will be developed independently providing the users with an environment that will enable writing and execution of their own code, and using BD4NRG resources. Of course, this functionality has not been developed yet, and is envisioned to be reported in the next technological releases of the BD4NRG services.

3.2 Graphical User Interface

As stated in the previous section, Apache Superset enables the connection with different databases, as well as an interface for building queries without writing any code. The provided interface is shown in the following figure. Specifically, a connection with the database that contains the data provided by Pilot 8 has been established. The aforementioned connection with the database is enabled through Presto[20] distributed SQL query engine, which is the main technology used in the Integrated Querying mechanism (Task 3.7 *Integrated Querying of Streaming Data and Data at Rest*). In the following Figure

[18] https://superset.apache.org/

[19] https://www.sqlalchemy.org/

[20] https://prestodb.io/

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

9, the user selects all columns of the queried dataset (tilos_input), with no aggregations and with the limit of 100,000 rows.

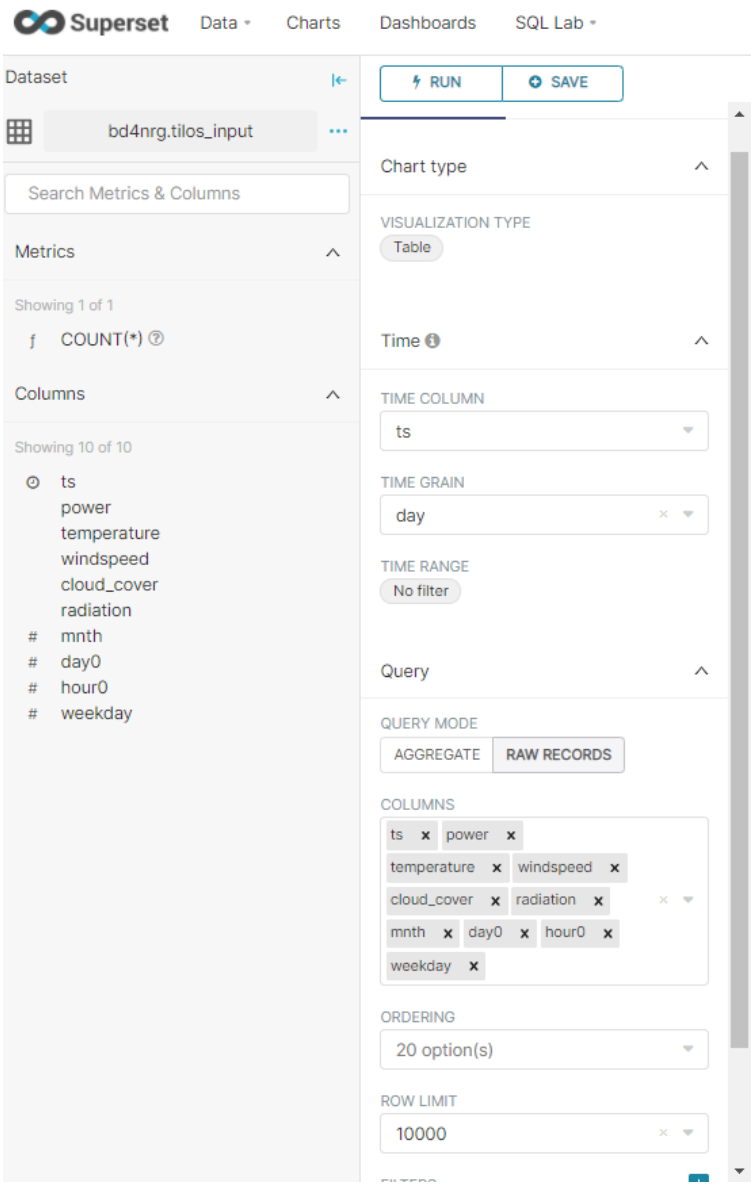
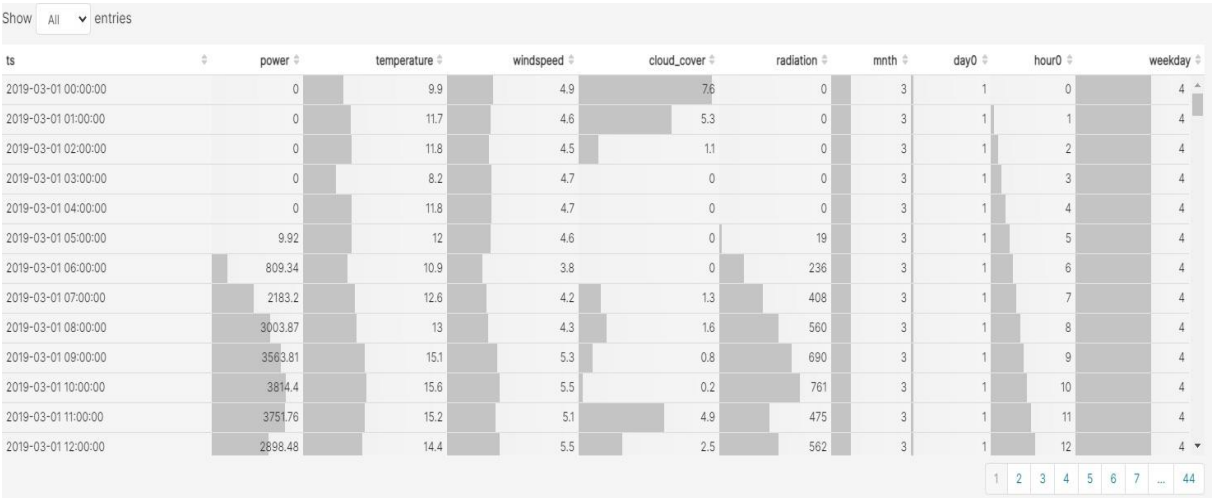


Figure 9: Build query user interface

In Figure 10 below, the result of the created query is presented in brief.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.



Show All entries

ts	power	temperature	windspeed	cloud_cover	radiation	mnth	day0	hour0	weekday
2019-03-01 00:00:00	0	9.9	4.9	7.6	0	3	1	0	4
2019-03-01 01:00:00	0	11.7	4.6	5.3	0	3	1	1	4
2019-03-01 02:00:00	0	11.8	4.5	1.1	0	3	1	2	4
2019-03-01 03:00:00	0	8.2	4.7	0	0	3	1	3	4
2019-03-01 04:00:00	0	11.8	4.7	0	0	3	1	4	4
2019-03-01 05:00:00	9.92	12	4.6	0	19	3	1	5	4
2019-03-01 06:00:00	809.34	10.9	3.8	0	236	3	1	6	4
2019-03-01 07:00:00	2183.2	12.6	4.2	1.3	408	3	1	7	4
2019-03-01 08:00:00	3003.87	13	4.3	1.6	560	3	1	8	4
2019-03-01 09:00:00	3563.81	15.1	5.3	0.8	690	3	1	9	4
2019-03-01 10:00:00	3814.4	15.6	5.5	0.2	761	3	1	10	4
2019-03-01 11:00:00	3751.76	15.2	5.1	4.9	475	3	1	11	4
2019-03-01 12:00:00	2898.48	14.4	5.5	2.5	562	3	1	12	4

1 2 3 4 5 6 7 ... 44

Figure 10: Query Results using Apache Superset for Pilot 8 dataset

Except building queries using the provided interface, a user can create custom queries by writing SQL code. This option is provided through the SQL Lab tab of Apache Superset, as illustrated in the following Figure 11.

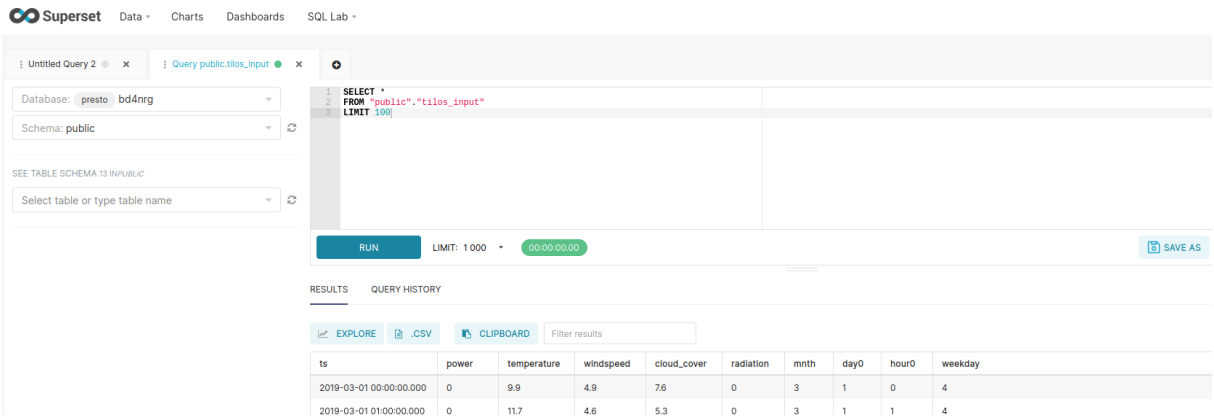


Figure 11: Build Query, using SQL Lab

Of course, the user can create visualisations on the provided datasets, as well. For example, in Figure 12 below, the user builds a time series bar chart that illustrate the average power consumed per day. It is worth mentioning that the provided visualisation is interactive, meaning that when the user hovers over a specific bar (day), a tooltip appears, that explains the result in more detail.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

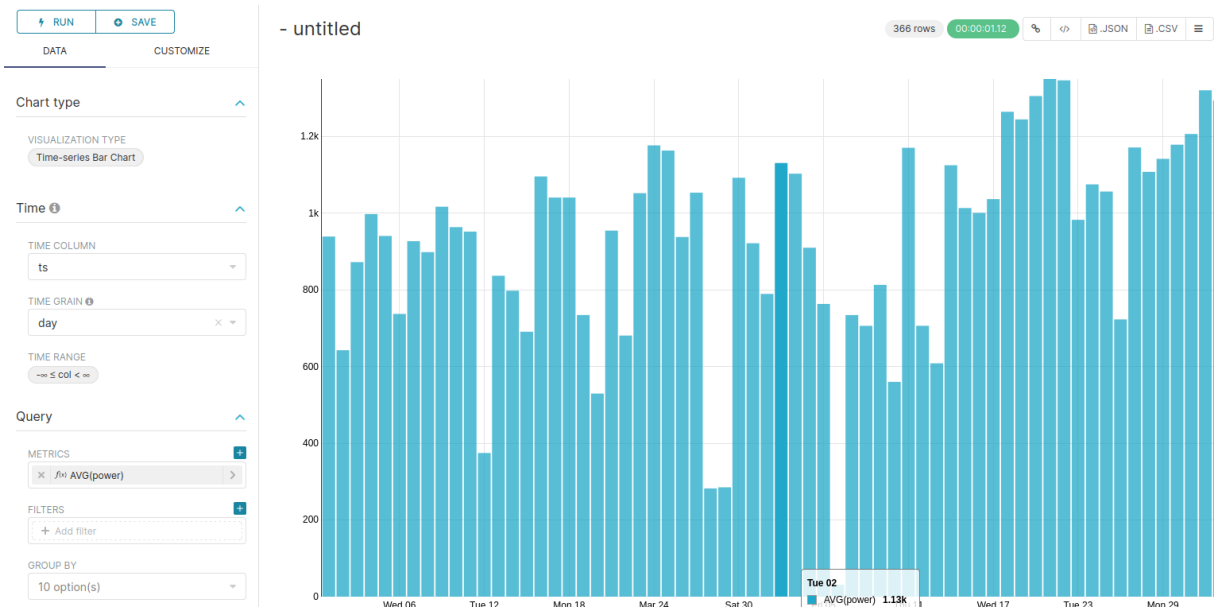


Figure 12: Time Series Bar Chart - Average Power per day

Furthermore, the user can choose from a wide variety of visualisations and tailor them according to his needs. For instance, in the following Figure 13, the donut chart shows the average consumption per month. The months are presented in order of power consumption, with months that pose higher power consumption, to appear before lower consumption months.

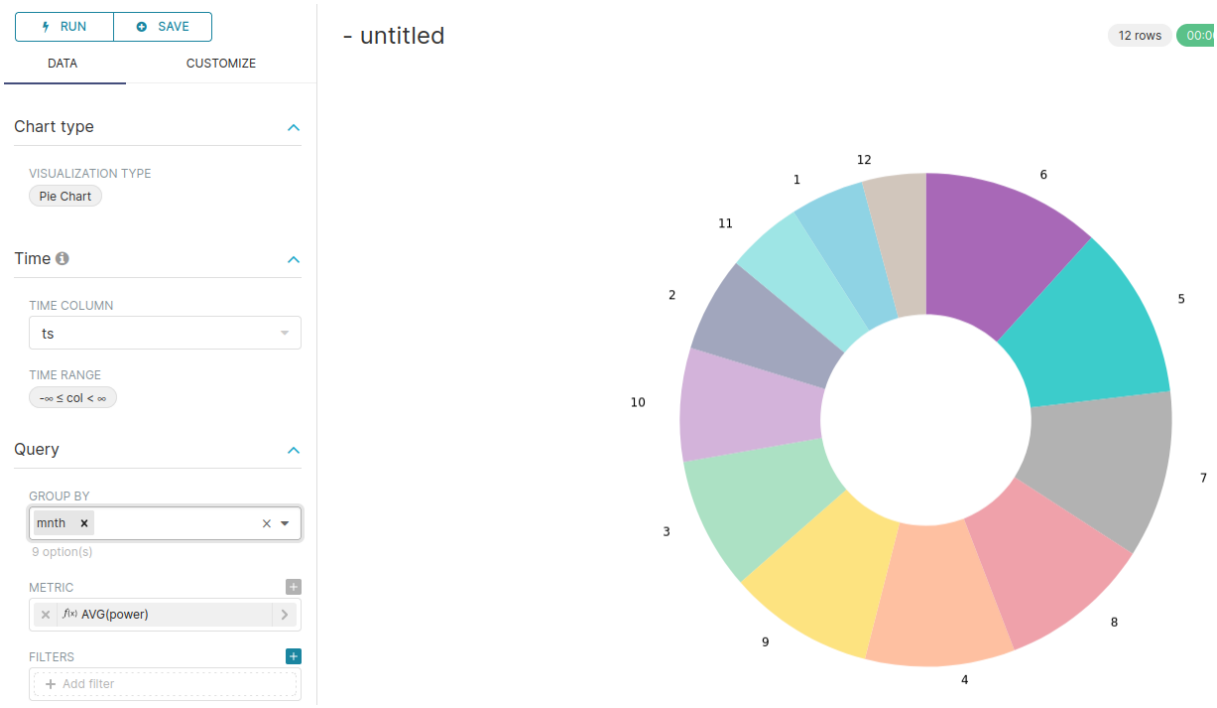


Figure 13: Power consumption per hour of day pie chart

Of course, the user can also customise the visualisations he/she creates, in terms of colors, shapes and more. In the following Figure 14, a donut chart with the consumption per day of week is presented alongside the available customisation options provided to the user.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

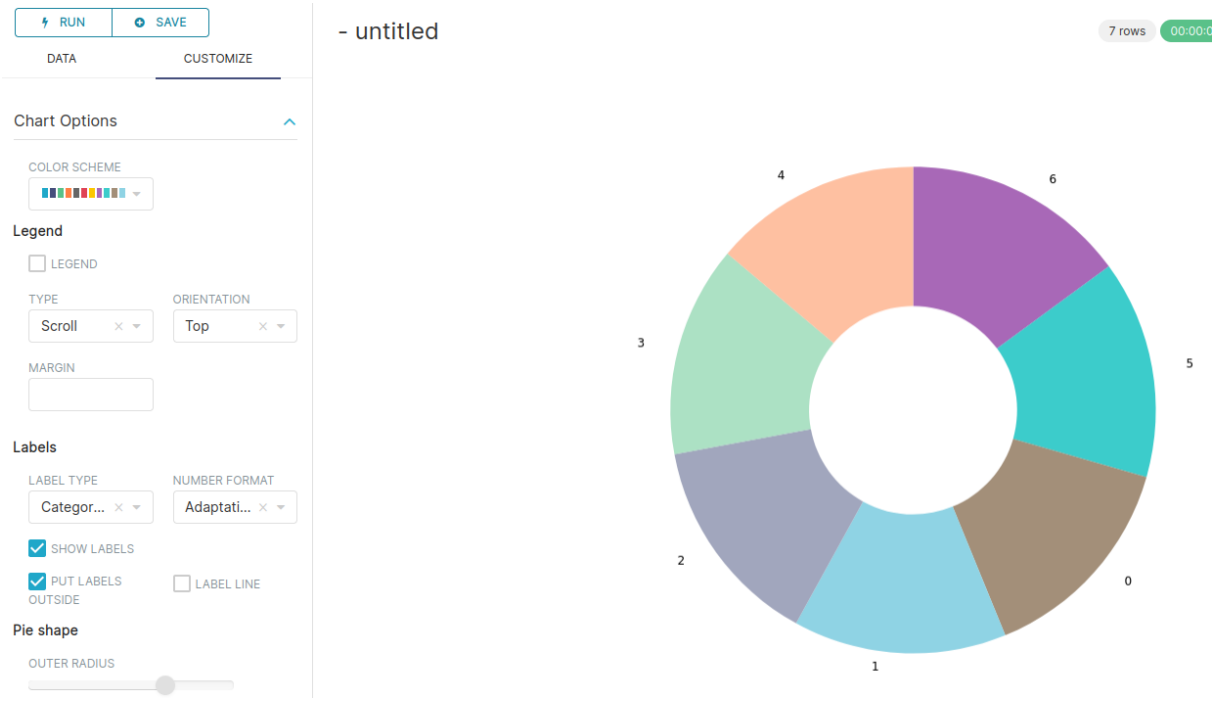


Figure 14: Power Consumption per weekday pie chart and customisation options

Last but not least, a user can create dashboards that combine several different visualisations. An example is shown in the following Figure 15. Specifically, the following dashboards, shows the time series data on a 24-hour resolution for consumed power, temperature, windspeed, cloud cover and radiation. Moreover, the bar charts, showcase the power consumption per weekday and per hour accordingly, on a descending order of consumed power.

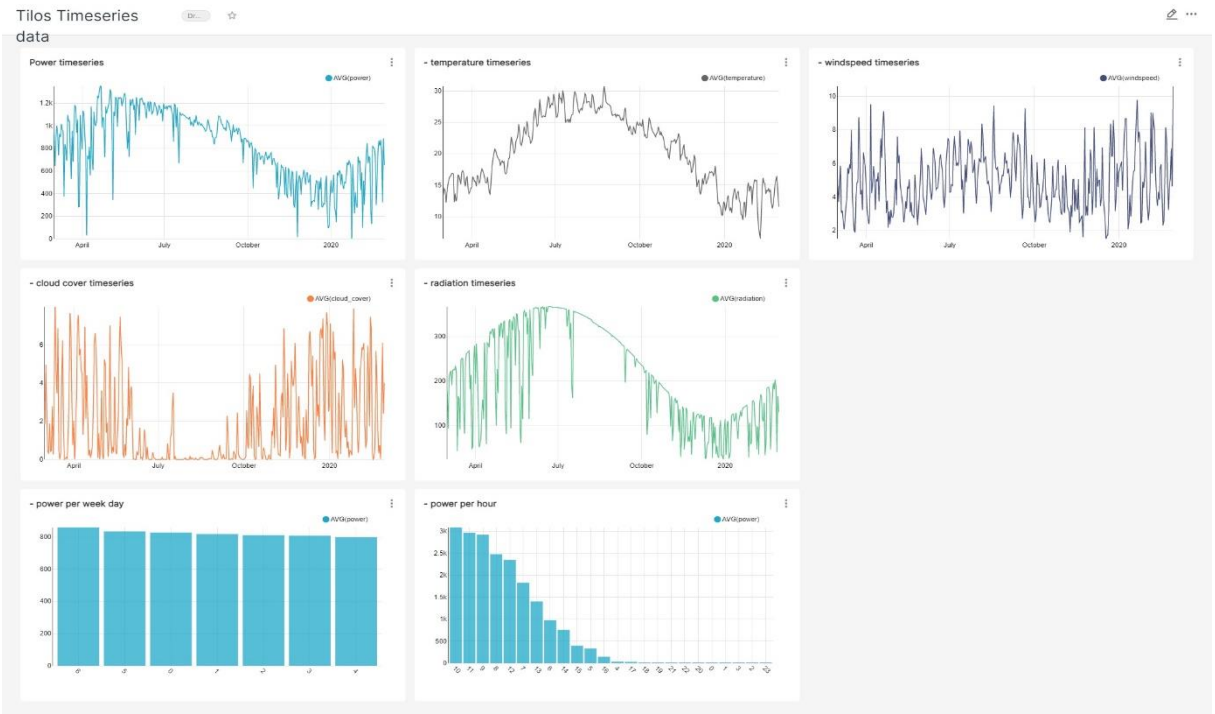


Figure 15: Dashboard for Pilot 8 dataset

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.

4 Conclusions

In this document, which has to be intended as the accompanying report of the deliverable D4.6 "BD4NRG-PROCESSING/ANALYTICS - First. Technology Release", a description of the first technology release of BD4NRG tools for BDA and visualization has been given.

As part of the Analytic Toolbox, which relies on the ALIDA platform[1], the Analytic Service Builder module has been described. This module allows the registration of new BDA services in the Toolbox catalogue and the deletion or update of the existing ones. The main functionalities of the module are described and a startup guide is provided to allow data scientists, IT operators, and in general BDA service providers to create, register, and manage BDA service as microservices: the creation of Docker files is automated, new images are pushed inside a repository, and the BDA service is registered into the catalogue.

As part of the deliverable D4.6, some services that take advantage of the Analytic Service Builder capabilities have been developed and described in this document. Those services are ML services capable of producing a model and using it for batch predictions.

Analytics tools implementing the pilot analytic services developed so far in WP4 are described and specifically:

- Photovoltaic Production Forecasting
- Load Forecasting
- Water Pumping Systems Load Shifting
- Energy Efficiency Investments Assessment.

The model used by those tools will be later integrated in the Analytics Toolbox.

Data Visualisation and Visual Analytics service has been presented in detail, focusing mostly on the application perspective and the offerings of Apache Superset, as the solution that was selected as the basic visualisation dashboard and reporting tool.

The integrated view of the available tools and ML algorithms developed within the BD4NRG Consortium as a modular toolbox, envisioned in Task 4.7 "Open Modular Analytics Toolbox & Deployment", will be provided as deliverables D4.7 "BD4NRG-PROCESSING/ANALYTICS - Second Technology Release" and D4.8 "BD4NRG-PROCESSING/ANALYTICS – Final Technology Release" and described in the related accompanying reports.

The source code of the BD4NRG software deliverables is stored in the BD4NRG GitLab repository; credentials for accessing the repository can be obtained by filling in the contact form in the BD4NRG website.

BD4NRG project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 872613.