interconnect

interoperable solutions connecting smart homes, buildings and grids

WP3.4 – Middle layers towards user-centric machine learning services pilots

D3.4

Middle layer for machine-learning components towards pilot requirements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant agreement No 857237

DOCUMENT CONTROL PAGE

DOCUMENT	D3.4 – Middle layer for machine-learning components towards pilot requirements
TYPE	Other
DISTRIBUTION LEVEL	Public
DUE DELIVERY DATE	31/12/2021
DATE OF DELIVERY	16/02/2022
VERSION	V1.0
DELIVERABLE RESPONSIBLE	VU Amsterdam
AUTHOR (S)	WP3 Partners
OFFICIAL REVIEWER/s	Georg Jung, Spiros Chadoulos, Carlos Manuel Pereira

DOCUMENT HISTORY

VERSION	AUTHORS	DATE	CONTENT AND CHANGES
0.1	VU Amsterdam	30/10/2021	Template and section outline for a shared redaction with partners
0.2	VU Amsterdam	12/12/2021	Added main content according to section outline
0.3	Inesctec	10/01/2022	Added details on Inesctec work
0.4	VU Amsterdam	21/01/2022	Finalized draft for review
0.5	VU Amsterdam + reviewers	31/01/2022	Updated version version
1.0	VU Amsterdam	16/02/2022	Final version

ACKNOWLEDGEMENTS

NAME	PARTNER
Ronald Siebes Roderick van der Weerdt Victor de Boer Roberto Reda	VU Amsterdam
Carlos Manuel Pereira Ricardo Silva David Emanuel Rua	Inesctec
Georg Jung Chris Caerts	Vito
Spiros Chadoulos	AUEB
Gjalt Loots	TNO

EXECUTIVE SUMMARY

Machine Learning (ML) algorithms play a significant role in InterConnect. Most prominent are the services that do some kind of *forecasting* like predicting energy consumption for (Smart) devices and households in general.

In this deliverable we address two aspects of machine learning algorithms.

Firstly, we look into the specificity of both the *type of predictions* and the *parameter settings*. One important goal of InterConnect is standardization of the interaction between Smart devices and services. Therefore, including the interface to the services that are capable of applying machine learning in a standardized manner benefits the re-use and improve the adoption of them. SAREF and its extensions are the schemas and vocabularies that we use to express the capabilities, interaction details and other aspects of the Smart components and facilitating services. These include measurement parameters and values. Graph Patterns using the SAREF vocabulary allow us to standardize the input and output for the InterConnect services, including those that are based on Machine Learning. This opens the possibility to explore how measurements expressed in SAREF can automatically be *matched* with well known machine learning algorithms that can deal with these specific type of input. For example, a classification task is based on a fixed set of values (e.g. 'open', 'closed') and an estimation task often uses a numerical range (e.g. '24.2 °C'). This *Multi-Modality* in machine learning is currently an active research topic in the international community. In this deliverable we give an overview of the state of the art research in Graph Neural Networks that are able to deal with graphs (of which a SAREF knowledge graph is an instance) and, secondly, how graphs can standardize the parameters, hyper-parameter choices, input and output schemas for the popular ML approaches.

Secondly, we look into the 'black box' aspects of ML approaches. Explainability of (the outcomes of) algorithms is fervently debated topic. In the media we can find various examples where bias in the training data and implementations lead to undesirable results. Users want to know why an algorithm came to a certain result and European Union has expressed the need to address this important aspect through clear guidelines[1]. Through the use of SAREF in InterConnect, we have already an adequate mechanism in place to make knowledge explicit on various parts of the ML process that improve explainability, for example provenance. Having a standardized way to express *who* developed the algorithm, on *which* training data and *where* the privacy details and other aspects can be found contributes to explainability. In this deliverable we show the results of a questionnaire various approaches to model the provenance and other aspects of machine learning algorithms used in InterConnect as ingredients to increase the explainabililty of ML algorithms. More traditional ML approaches such as decision trees have a higher degree of explainability[2], for example one can derive the importance of specific parameters in the input data. ML can be used to derive rules, and these rules are then included into the knowledge base (which reduces the size of the 'black box' in the process hence increasing the explainability).

[1] https://ec.europa.eu/futurium/en/ai-alliance-consultation/guidelines/1.html

[2] https://christophm.github.io/interpretable-ml-book/tree.html

The contributions of this deliverable are as follows:

- *Results of a questionnaire detailing the ML aspects for the services used in the InterConnect pilots.*
- *Demonstrations showing how Jupyter Notebooks can serve as a means to provide explainability for ML components*
- *Results of applying Graph Neural Networks on SAREF serving as a generic middle layer*
- *Extensive description of documenting ML facets for a forecast application as an example of explainability*

In other words, in this deliverable we outline the analysis of the explainability landscape within InterConnect and provide the references to data, implementations and code that are developed towards the goal of exploiting the semantic possibilities of SAREF towards easier integration of Machine Learning algorithms which form the elements of a middle layer between the applications and use cases and the various ML algorithms. We conclude that SAREF allows us to standardize input formats for common ML approaches and that explainability can be increased by selecting algorithms that inherently have these features (e.g. Decision Trees) and by using interactive web environments like Jupyter Notebooks a convenient solution for users is created where step by step the algorithmic procedures can be followed and visualized and forms a implementation guideline for the upcoming pilots.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Machine Learning is a very active and broad field within computer science with the common denominator of algorithms 'learning' something useful from the data they experience. The field evolves rapidly where Deep Learning applications currently make the news headlines also sometimes to expose controversial consequences of applying ML and Artificial Intelligence in domains of health, safety and fundamental human rights. On April 21th 2021, the European Commission adopted a proposal for harmonizing rules for Artificial Intelligence[3] :

"...the Commission puts forward the proposed regulatory framework on Artificial Intelligence with the following specific objectives:

- *ensure that AI systems placed on the Union market and used are safe and respect existing law on fundamental rights and Union values*
- *ensure legal certainty to facilitate investment and innovation in AI*
- *enhance governance and effective enforcement of existing law on fundamental rights and safety requirements applicable to AI systems*
- *facilitate the development of a single market for lawful, safe and trustworthy AI applications and prevent market fragmentation ..."*

Explainable AI recently became a part of the efforts to address the points above: one can only judge as a human if a result of an AI algorithm is acceptable if one knows why it came to its derivations. Within InterConnect we deal with various types of AI technology that have an impact on the human living space. Some examples of where this impact might occur:

- **Safety:** if the output of an AI algorithm causes raising the fire alarm
- **Health:** if the algorithm regulates the CO2 levels in a building
- **Fundamental human rights:** if the algorithm monitors who is in a building and when

To address the topic of explainable AI in this project we will take a couple of steps:

First, we have to analyse where and what type of Machine Learning is being, or going to be used in our project. To that end, we set up a survey that addresses which kind of algorithms are used in the services that are developed in this project, what data is being collected and produced, who can access the data and who monitors the outcomes, etc. The survey questionnaire and the results are described in Section 2.

Second, we address various common Machine Learning algorithms and show how we can exploit parts for explainability. In Section 3 we reference an extensive example of how the internals of two popular

[3] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206

Machine Learning algorithms (Linear Regression and Random Forests) applied on real data sets from the Smart Home domain can provide explainable insights within the InterConnect pilots using the services described in Section 2.

Third, since we strive towards interoperability on a semantic level, with SAREF as a common communication layer, we look into ML algorithms that are designed for graph-based input (e.g. GNNs - Graph Neural Networks[4]). It is important to note the difference in *where* the ML is applied here. Where in section 2 ML is part of the various forecasting services, here ML is applied *directly* on the SAREF knowledge graph. The application of this has various possibilities that are enabled with GNNs, like link prediction, element classification, value prediction and automatic error detection. Section 4 outlines these possibilities and references to the software developed within this project.

Fourth, in section 5 we outline a solution how data expressed in SAREF can be transformed into input data for frequently used ML algorithms.

Fifth, in section 6 we outline some preliminary work done on a hybrid approach of combining the ML services with a rule-based methodology based on the SWRL language.

Section 7 provides a reflexion on the current status of the work and a planning for future work.

[4] Xu, Keyulu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. "How powerful are graph neural networks?." *arXiv preprint arXiv:1810.00826* (2018).

2. ML IN INTERCONNECT SERVICES

2.1. QUESTIONNAIRE

To assess the amount and type of Machine Learning services used in the various InterConnect Pilot services, we conducted a survey among the InterConnect WP 3 participants. The survey was sent in the form of a Google Forms questionnaire in December 2021. The request was for all partners that use any kind of ML to tune service parameters or have an ML algorithm as part of the service functionality to specify the type of data and algorithms used.

The original survey is found here:

https://docs.google.com/forms/d/1EQ9OBkp4w1oHHN5hY6CVHVMDtdGlUhJ1lvDLtrDMhKY/edit#responses

We list the questions below.

- **Email**
- **Organisation:** What is the name of your organization?
- **What is the name of the service?** Please look at the matching title here
- **Type of Machine learning algorithm used in the service:** Please provide the type of algorithms that most closely resemble the ones used by the service. For an overview please look here: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/
  - **Regression** (e.g. Linear Regression)
  - **Instance-based** (e.g. k Nearest Neighbour, or Self Organizing Map)
  - **Regularization** (e.g. LASSO)
  - **Decision Tree** (e.g. C4.5, Random Forest)
  - **Bayesian** (e.g. Gaussian Naive Bayes)
  - **Clustering** (e.g. k-means)
  - **Association Rule Learning** (e.g. Apriori, or Eclat)
  - **Artificial Neural Network** (e.g. Perceptron, or Stochastic Gradient Descent)
  - **Deep Learning** (e.g. CNN, RNN, LSTM)
  - **None**
  - **Other:**
- **Source code and license of the Service**
  - The algorithm(s) are part of the functionality of the running service
  - The algorithm(s) are only used to tune the (hyper) parameters for the service
  - Other:
- **Training/test data type**: Please select which of the conditions you have for the service**?**
  - Code is public to everyone and has open-source license
  - Code is public to consortium and has restrictive license

  - Service can be installed by public, source is closed
  - Service can be installed by the consortium during the duration of the project, source is closed
  - Service cannot be copied but the running instance is accessible by the consortium
  - Service is running on private infrastructure and is only accessible granted via the developer
  - Other
- **Training/test data availability**: What type of data, if applicable, is used to train the ML algorithm**?**
  - Synthetic data
  - Real data, containing no user information
  - Real data, with anonymized user information
  - Real data, with privacy sensitive information
  - No training data is needed for the ML model
  - Other
- **Input data sensitivity**: To whom is the data available during the length of the project**?**
  - public for anyone with CC type of license
  - public under certain conditions
  - available only to the consortium under certain conditions
  - not available
  - Other:
- **Input data accessibility**: Does the algorithm when operational receive sensitive data as input during the project**?**
  - data containing sensitive security and safety information (e.g. emergency doors are opened, security camera feed)
  - data containing privacy sensitive data (e.g. body temperature, who is in the building)
  - no sensitive data
  - Other:
- **Input data monitoring**: Who can access and monitor the input data?
- **Potential impact of ML algorithm decisions**: When the service is operational, will data actively (e.g. observing live data stream) and/or passively (e.g in retrospect via log files) monitored on eventual anomalies
  - Actively
  - Passively
  - No
  - Other:
- **Output data verification**: Does the algorithm affect in the decision-making process sensitive areas like health (e.g. reduce CO2 values in a room), (residence) safety (e.g. open emergency doors, raise fire alarm) or fundamental human rights (e.g. do not let this person in the building). Please specify the area and the effect of the automatic decisions/output
- **Output data monitoring**: Are the output data and 'decisions' being monitored actively and/or passively?

  - For some of the (more influential) decisions, there is a 'human in the loop' that need to verify before execution.
  - Data is actively monitored (e.g. building manager keeping an eye on the effects of the decisions)
  - Data is passively monitored (e.g. by regular checks of log files)
  - Other:
- **Output data access**: Who has access to the output data/decisions (e.g. the building manager, service developer, System admin)**?**
- **Explainability**: Please describe in a couple of sentences how you envision to improve the feedback for the human end users about the results of the ML algorithm. For example, for the Random Forests algorithm a visual representation of the decision tree could be shared or a human readable, detailed and specific log report with timestamps to easily trace back some output decisions of interest, etc
- **Pilots**: Please specify in which pilot(s) the service will be used.
- **Graph patterns**: If possible, please copy/paste, or write, the graph-pattern for the SSA of the service
- **Further remarks on privacy**: If applicable, please provide information how you deal with privacy sensitive information. For example, do you have a data protection officer that checks that sensitive information is removed from the trainings data. Or, can you comment where the data is being stored (e.g. is collected locally and processed by machines on the local Intranet).

2.2. SURVEY RESULTS

We received 12 responses to the questionnaire from participants from 7 different project partners (see Figure 1 below).

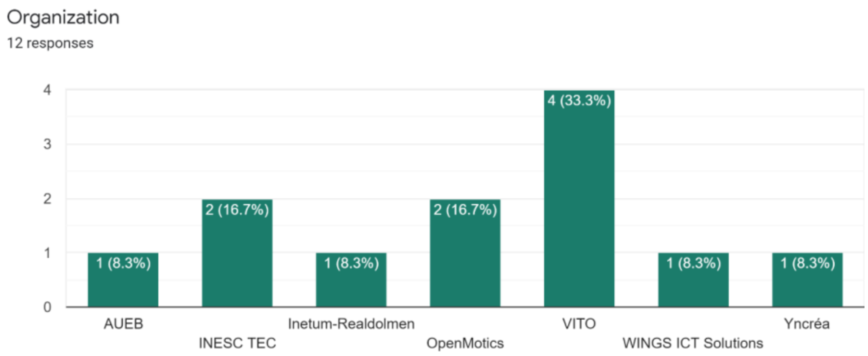


FIGURE 1: NUMBER OF RESPONSES PER ORGANIZATION

Each response corresponds to a single service, with multiple services potentially identified for different pilots/partners. The list of services is found below in Table 1 and corresponds to the list of services on the Inesctec drive[5] . This list shows that various types of services are concerned, with in many cases *forecasting* being a key goal. The results show that in all cases, the ML algorithms used are part of the functionality of the running service, rather than (only) used to tune parameters for the service.

TABLE 1 – THE NAMES OF INTERCONNECT SERVICES INVOLVED IN THIS QUESTIONNAIRE

PowerLimitation	Wind Forecaster	Electricity Demand Forecaster
Conditional Prosumption Forecaster	Prosumption forecaster	Flexibility
WINGS - Predictions - Service Mapping / WINGS - Recommendations - Service Mapping	Flexibility forecaster	INESC - Energy Forecasting
AUEB - AUEB Mobile App	Heat Demand Forecaster	Load Forecaster

Regarding the type of Machine Learning model/method used, Figure 2 below shows that a variety of methods is used by the various services. As the percentages add up to more than 100%, this shows that in some cases more than one method is used. Half of the services use a Regression approach such as Linear Regression, supporting the need for developing reusable solutions for this as introduced in Section 4. Additionally, the majority of services (7) uses some form of Deep Learning, making reusable methods here useful as well (see Section 4).

5 https://drive.inesctec.pt/apps/files/?dir=/InterConnect_Proj/InterConnect%20WP%20Repository%20Nextcloud/WP3%20-%20Semantically%20Interoperable%20Components%2C%20Applications%20and%20Devices%20for%20Smart%20Homes%20and%20Buildings/WP3%20Services%20catalog/Services%20SAREFization&fileid=16403220

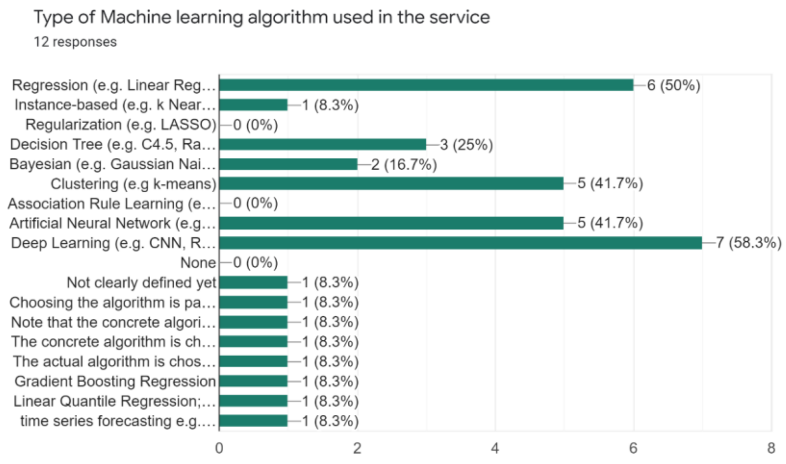


FIGURE 2: TYPE OF ML ALGORITHM USED IN THE SERVICES

Concerning training and test data, in most cases both the models and the data are available under restrictions. The majority of datasets used to train the ML components is accessible within the InterConnect consortium but cannot be shared outside the consortium (See Figure 3, 4).

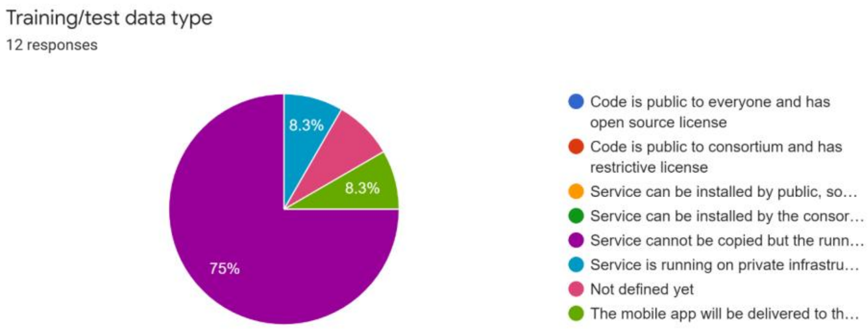


FIGURE 3: TYPE OF TRAINING AND TEST DATA USED IN THE SERVICES

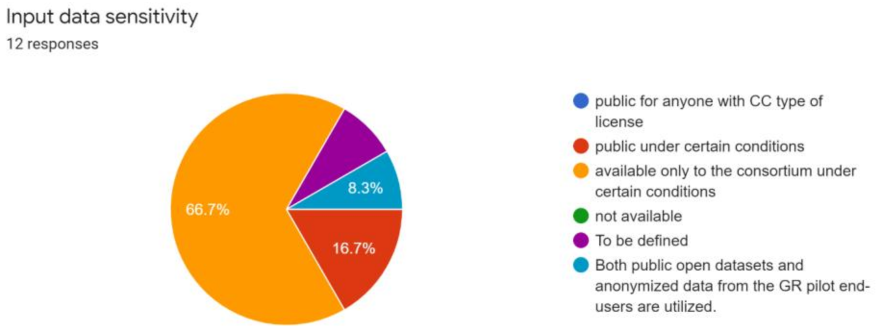


FIGURE 4: SENSITIVITY OF INPUT DATA USED IN SERVICES

In terms of monitoring, the results show a wide variety of methods (see Figure 5). The most prevalent method of monitoring results is by passively monitoring data for example by regularly checking the log files (with 8/12 services indicating to use this method).

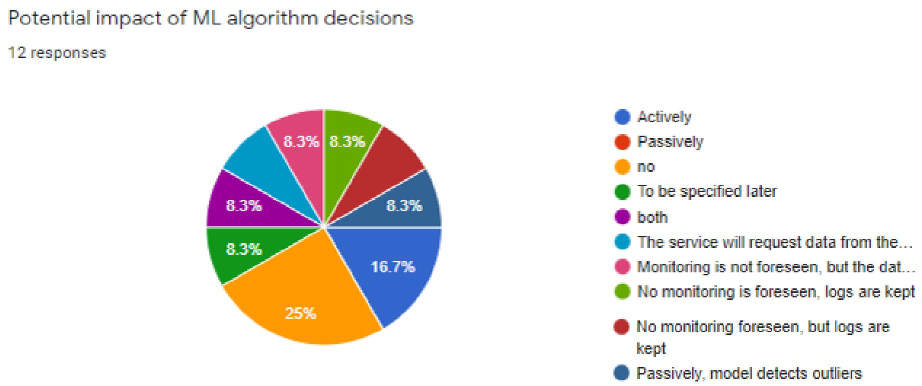


FIGURE 5: METHOD OF MONITORING USED IN THE SERVICES

The complete results of the survey can be found on Google Forms[6]

2.3. ML EXPLAINABILITY IN THE PILOTS

Based on the results of the questionnaire and the communication with the partners during the WP3 meetings an overview can be made which pilots are going to, or have the possibility to use the prototypes detailed in the next sections.

Concretely, in the Dutch pilot data from the EKCO builder[7] will be used to predict sensor measurements from the VideoLab[8] based on the prototype described in Section 3.2.2. Idem for the Greek pilot data. All pilots that either use Linear Regression, Decision Trees and/or Clustering can also use this Jupyter Notebook implementation work as a tool to increase explainability to their stakeholders. The RefriFlex system, described in Section 3.3 is part of the Portugese pilot. In Section 4, two prototypes are described that apply to all pilots since it works directly on *any* SAREF data, which is commonly shared among these pilots.

In Table 2, we visualise the applicability of the methods described in this deliverable to the seven pilots. For each combination of method and pilot we list whether a method: Green cells indicate that a method will definitely be applied in a pilot either due to implementations for the pilot or the use of data from the pilot, yellow that it can be applied to the pilot, and empty cells denote that either a given method is not likely to be applied to that pilot or that the ML implementation approach is not determined.

TABLE 2 - APPLICABILITY OF THE METHODS DESCRIBED IN THIS DELIVERABLE TO SPECIFIC PILOTS

Pilots	Notebook-based interactive	RefriFlex method (S3.3)	Embedding-based method (S4.1)	Feature extraction- based method (S4.2)

6 https://docs.google.com/forms/d/1EQ9OBkp4w1oHHN5hY6CVHVMDtdGlUhJ1lvDLtrDMhKY/edit#responses

7 https://www.builderekco.nl/

8 https://strijp-s.nl/ondernemers/videolab/

	method (S3.2.2)			
Greek pilot				
Dutch pilot				
French pilot				
Portugese pilot				
Belgian pilots				
Italian pilot				
German pilot				

3. DERIVING EXPLAINABILITY FROM MACHINE LEARNING ALGORITHMS

3.1. INTRODUCTION

The effort in the InterConnect project to harmonize and standardize the communication between Smart devices and services, including AI technology as part of the services but also the Knowledge Engines, inherently will influence the life conditions of the people that will use these solutions. Although we all try to improve various aspects like comfort, safety and energy effectiveness, it is paramount that potential negative side effects are actively identified and monitored. AI technology allows us to delegate complex decisions to computer systems. Examples are: "should a fire alarm be raised, and should doors be automatically closed?", "how much CO2 is optimal for the people in this room?", "who is in this room, and should the body temperature of the children be checked?". These examples relate to "Safety, Health and Fundamental Human Rights", which are identified by the European Union as categories with special responsibilities to the AI algorithms involved in the decision-making process[9]. In a complex and large project like InterConnect, where actual devices and AI technology will be implemented in office and home residences, we are obliged to have a look on the potential risks and prepare for the upcoming legislation drafted by the EU.

Next to preparation for the legislation, the *explainability* of AI algorithms is a desired feature in many situations. Some examples:

Error detection A prediction from an AI algorithm about the energy demand for a residential building for the next day can be used by a human trader to buy a certain amount of KWh for that day. When an algorithm predicts a huge spike in energy consumption, it could be the result of an error, but also genuine (e.g. the result of a sudden change in weather). Feedback from the algorithm allows the trader to prevent buying a large amount of energy for the next day that is not needed.

End user understanding When the AI system opens a window and causes draft, the resident might want to know why this sudden drop in comfort is needed. When the system provides feedback that it is needed due to reduce a high level of CO2 in the room and will be closed as soon the level is normal again, the resident can understand the situation and accept the temporal condition.

Procedural improvement feedback Within a complex assembly of various AI components, situations can occur where these components conclude conflicting actions. Take the example above, where the CO_2 level was too high and the window is opened. Another algorithm monitoring the room temperature 'sees' that

[9] https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52021PC0206&from=EN

the room temperature is dropping and concludes that the window must be closed. This might result in the strange behaviour a resident gets into where the window opens and closes every 5 seconds. When the resident reports this strange behaviour, it is important that the problem is identified quickly and resolved effectively and efficiently.

Certification and sales When the explainability is of high quality, the market value of the involved AI components is likely to be much higher, due to the previous three reasons. One can imagine that in the future a higher demand of explainability is to be expected for industrial partners that use an integrated solution like InterConnect for setting up Smart offices and homes. Also next to the upcoming EU regulations, certification organisations like ISO, will require explainability of the AI algorithms in Smart home and office solutions.

In Section 3.2 and 3.3 we provide two implementations of ML systems developed within InterConnect where we explore ways to extract information from the ML workflows that caters explainability.

3.2. THE OPSD DATASET AND JUPYTER NOTEBOOKS

3.2.1. OPSD

The Open Power System Dataset Household data dataset (OPSD)[10] is a large dataset collected over three years from 68 devices from eleven different buildings, three industrial buildings, two public buildings and six residential households located in Konstanz, Germany. For each of those 68 devices it collected the energy consumption on a 15 minute interval.

The advantage of this dataset is that it has collected measurements over a long period of time for multiple buildings, the downside is that it only collects one specific datatype, energy consumption, so it is not a multi modal dataset, making it less representable for the data that will be collected in the pilots.

3.2.2. JUPYTER NOTEBOOKS PROVIDING INPUT FOR EXPLAINABILITY

Jupyter Notebooks[11] are run as a web application for creating and sharing computational documents in the Python programming language. The advantage of having a web interface where a user can monitor step by step the execution flow and visualize intermediate results is gaining insight into the algorithms.

[10] Open Power System Data: Data Package Household Data. Version 2020-04-15 https://data.open-power-system-data.org/household_data/2020-04-15/. (Primary data from various sources, for a complete list see URL). (2020)

[11] https://jupyter.org/

On the Inesctec GitLab[12] repository we developed an extensive example on using two popular ML algorithms (Linear Regression and Random Forests) to forecast the energy consumption of heat pumps based on the OPSD historical data.

The in-line comments in the code together with the Jupyter environment makes the code easy to read even for those not familiar with ML algorithms, hence contributing significantly to the aspect of explainability.

3.3. REFRIFLEX – FLEXIBILITY FORECAST

3.3.1. DESCRIPTION OF THE SUB-SYSTEM

The refrigeration system's flexibility (RefriFlex) forecasting sub-system supports all the functions envisioned for forecasting the flexibility that can be provided by the refrigeration system of 8 supermarkets from SONAE MC, as part of the Portuguese pilot. To do so, the sub-system provides data-driven forecasts, per supermarket, of the refrigeration system's active power consumption time-series. Each store has local controllers enabling three different operation modes, namely "normal", "eco" or "boost".

- "**Normal**" mode refers to the normal operation conditions of the supermarket (e.g., optimal temperature levels).
- "**Eco**" mode is used to decrease power consumption of the supermarket cooling systems during specific time intervals, through an adjustment (increase) in temperature set-points.
- "**Boost**" mode is used to increase the power consumption of the supermarket cooling systems during specific time intervals.

Each supermarket has a default operation mode schedule (here also referred as '*operation mode schedule baseline*') defined by SONAE MC and cold storage flexibility (upwards or downwards) can be obtained by switching from the baseline mode to one of the remaining two alternative modes (with higher or lower power consumption requirements, respectively).

This sub-system is composed by three types of functions, composed by different software modules. Figure 6 provides an overview of each function.

12 https://gitlab.inesctec.pt/interconnect/interconnect-explainability

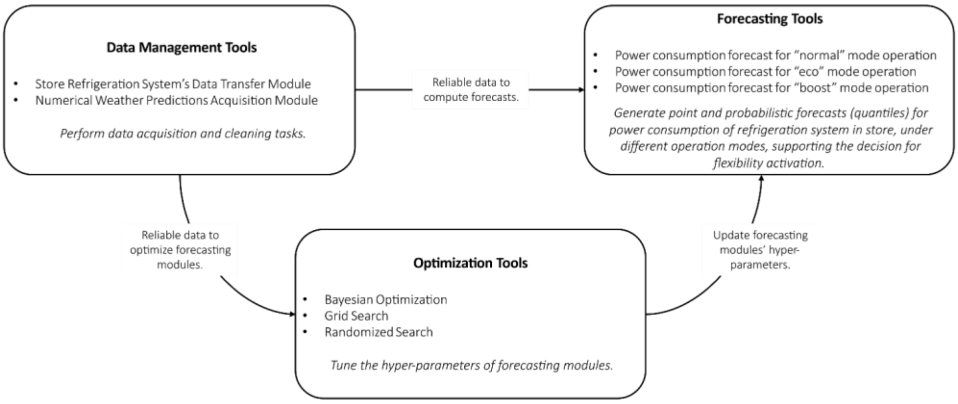


FIGURE 6: SOFTWARE MODULES THAT COMPOSE THE REFRIGERATION SYSTEM FLEXIBILITY FORECASTING SUB-SYSTEM AND RESPECTIVE INTERACTIONS

As depicted in Figure 6, RefriFlex forecasting tool is composed by three forecasting models, designed to forecast not only active power consumption time-series during a "normal" operation mode, but also during "eco" and "boost" modes, with lower or higher power consumption respectively.

The final output of the forecasting tool consists of a flexibility forecast, which is calculated as the difference between the power consumption under two of the three operation modes (more details available in section 3.3.2.2).

3.3.1.1. DATA MANAGEMENT TOOLS

RefriFlex forecasting models developed depend on two different data sources:

- Numerical Weather Predictions (NWP)
- Store refrigeration system's data, in time series format, namely:
  - cooling system historical average active power consumption measurements
  - historical and future (i.e., scheduled) setpoints per refrigeration store unit (defrosting, illumination, and temperature),
  - historical and future store operation modes ("normal", "eco" and "boost"),
  - historical and scheduled refrigeration unit's state (ON/ OFF).

The following data requirements must therefore be satisfied for the forecasting modules to operate:

- **Weather information**: Access to historical and daily updates of numerical weather predictions (NWP) for the stores' geographical location
  - Specifically, NWP of ambient temperature at 2-meters altitude, for a time horizon of up to 48 hours ahead.
- **Refrigeration system's data:**

- Active power consumption measurements time series include:
  - **Historical average active power consumption measurements**, provided in kW, with a resolution not higher than 15 minutes.
  - **Two time series of power consumption must be provided per supermarket**, one for refrigeration units designed for low-temperature (operating range: [-25°C, -15°C]) and another for units designed for high-temperature (operating range: [-2°C, 12°C]).
- Store (scheduled) setpoints include:
  - **Historical and scheduled defrosting status of each refrigeration unit** in the supermarket, provided as a Boolean (True if the unit is defrosting, False if not), with a resolution not higher than 15 minutes,
  - **Historical and scheduled operation mode of the whole refrigeration system**, provided as a string (one of [“normal”, “eco”, “boost”]), with a resolution not higher than 15 minutes,
  - **Historical and scheduled illumination status of each refrigeration unit** in the supermarket, provided as a Boolean (True if the unit’s light is on, False if it is off), with a resolution not higher than 15 minutes,
  - **Historical and scheduled temperature set-point of each refrigeration unit** in the supermarket, provided in °C, with a resolution not higher than 15 minutes.
- Refrigeration units’ state include:
  - **Historical and scheduled on/off status of each refrigeration unit in the supermarket**, provided as a Boolean (True if the unit is active, False if not), with a resolution not higher than 15 minutes.

Note 1: Regarding scheduled set-points, the user should send future data (e.g., set-points for tomorrow) and past data (set-point activation history).

Note 2: Update of historical data should be performed hourly (important) or once per day (essential).

Early access to historical and scheduled data is paramount for the forecasting models’ selection and initial training. Daily updates (of store data and NWP) are therefore essential for the forecasting capabilities (forecasts cannot be computed without recent data).

At each forecast launch, an assessment phase is performed and, depending on the available data, each flexibility forecast request will return either forecasts created with a primary model (i.e., with an ideal subset of input variables and hyper-parameters), a secondary model (i.e., with a secondary subset of input variables and parameters) or an error message indicating the impossibility of running the forecast.

The data management tools incorporate two different modules that will process all information. The following functionalities are available:

- REST API to be used only by Sensinov’s Building Operating System (BOS) to send store refrigeration system’s raw data and to request flexibility forecasts. The new data is then stored in the platform’s database.

- APIs used to contact external public providers and that will retrieve historical and daily updates of NWP in the form of NETCDF4 files, which are stored in RefriFlex filesystem.

These functionalities are fully described in the following sub-sections.

3.3.1.1.1 DATA TRANSFER MODULE (RESTFUL API)

This module provides an interface for secure and practical communications between external clients and forecasting server databases, where refrigeration system's raw data will be stored. A RESTful API separates the client/user interface from the server and respective data storage components. More details on the REST API description are available in section 3.3.1.4.1.

Figure 7 illustrates the communication between an external agent (in this case, Sensinov client) and the REST API Server of this forecasting platform.

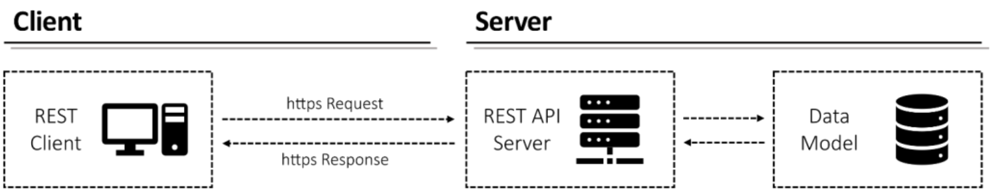


FIGURE 7: DATA TRANSFER COMMUNICATION BETWEEN THE EXTERNAL CLIENTS AND THE REFRIFLEX REST API SERVER. THE REST CLIENT CAN BE ANY RECENT PROGRAMMING LANGUAGE

The communication is always initialized by an external client request. Seven types of requests are available:

- **HTTP POST** to register a user in the platform. Before using any other REST API endpoints, external clients will have to first register in the platform with their own credentials (e.g., email and password).
- **HTTP POST** to login a user in the platform. This will provide a bearer token that must be included in the header of each request to the remaining service endpoints. This token will be used to validate the user and its privileges to access each endpoint and its information.
- **HTTP POST** requests used to send supermarket' raw data to the forecasting platform database. Each request is followed by a predefined REST API server response confirming if the operation was concluded with success or not.
- **HTTP POST** requests to ask for a flexibility forecast computation, for a specific configuration set in the request payload. Each request is followed by a predefined REST API server response confirming if the operation was successfully started and an URL to be later used to retrieve the forecast result.

- **HTTP GET** requests to retrieve the forecast result for each forecast request. If the forecast is already computed the agent should receive a JSON with the forecasted time series, else a response mentioning that the forecast is still being computed will be issued.
- **HTTP GET** requests to retrieve the explainability metrics for the machine learning model per supermarket and per type of model (primary and secondary – please refer to section 0 for more details). The two metrics provided are discussed in section 3.3.1.4.
- **HTTP GET** requests to retrieve measurements, schedules, and statuses' data from the forecasting platform endpoint. ***This method will only be available for the platform admins***.

Data transmitted to the forecasting platform endpoint is first validated and then stored in a PostgreSQL Database storage system. As such, any requests sent to the RefriFlex platform should follow the rules and data structures detailed in **Error! Reference source not found.**.

REST API users should always inspect the request responses provided by RefriFlex platform, confirming if each request was submitted with success or if any problem was detected in the headers, query parameters or JSON payload.

3.3.1.1.2 WEATHER DATA ACQUISITION MODULE (WDAM)

Access to weather data, specifically to temperature forecasts (historical and for the future) is crucial to improve the short-term (i.e., up to 48 hours ahead) forecast quality of the models produced by the RefriFlex modules. The WDAM provides continuous weather data support to these modules by extracting forecasted weather information from the following resource, available in the scope of InterConnect:

- **Meteogalicia** – Meteorological institute of Xunta de Galicia (Spain) that regularly publishes data from the Weather Research and Forecasting Model (WRF), a regional mesoscale NWP model designed for both atmospheric research and operational forecasting applications. The WRF model runs operationally twice a day initialized at 00:00 UTC and 12:00 UTC, the former providing forecasts for a horizon of 96 hours and the later for 84 hours. For this project, the 12 km and 36 km spatial domains are considered.

NWP are extracted for the surroundings of each demonstrator, by providing the coordinates of the respective retail store.

Although relying on a single weather provider, the designated source is reliable, and the availability of good data quality is expected as required by the forecast modules.

3.3.1.2. FORECASTING TOOLS

3.3.1.2.1 Flexibility Forecast Module

Accurate forecasting of electrical energy consumption is a valuable tool for planning the activation of flexible resources for the provision of ancillary services to system operators. The refrigeration systems of supermarkets account for a significant percentage of their total demand and, through a configuration of temperature set-points, it is possible to adjust the electricity consumption of the system making it a potential source of considerable flexibility to be used by the DSO. Given the restrictions at food quality level, SONAE MC has been implementing a safety methodology for remotely changing temperature set-points of the refrigeration system at their supermarkets and adjust power consumption upward/downward, while ensuring minimum quality standards for the products in the supermarket. The methodology consists of three operation modes: a “normal” operation mode, where both temperature setpoints and power consumption do not present any major changes (except during defrosting periods); “eco” mode where power consumption is significantly reduced via an increase of the temperature set-points of the store’s refrigeration units and “boost” mode where power consumption is increased by decreasing the temperature set-points of those units. Each mode results in a central command that adjusts the temperature set-points of every refrigeration unit in the supermarket – while also assuring the compliance with a range of temperature limits that do not impact food quality standards - to indirectly affect the power consumption of the refrigeration system.

This module explores NWP, active power measurements and several historical and scheduled operation set-points (see section 3.3.1.1) to produce up to day-ahead forecasts of active power consumption per supermarket. More specifically, two sets of point forecasts are produced, one for the baseline consumption of the store (i.e., based on the scheduled operation modes) and another set under a different operation mode. By calculating the difference between the two sets of forecasts, for the time horizon considered, one can assess the flexibility that can be provided by changing that scheduled mode of operation. The decision of changing that set-point is, however, outside the scope of this module and of the forecasting tool entirely. The forecasts will be stored in the forecasting platform database and can be accessed by the REST API interface, accessible to any external clients with permissions.

This module is composed by:

- **Feature Engineering Techniques** – consists of a preliminary step (prior to Machine Learning models’ training or application) that transforms some of the raw store dataset variables into a new subset of features that improve the models forecast quality. This module uses prior knowledge and code from INESC TEC in the energy time series forecasting domain. These new features consist of:
    - Calendar-based features, extracted from the time series datasets timestamps (e.g., day of the week, weekday or weekend, hour of the day and minute of the hour, business day, etc.),

  - Time series lags to model the temporal dependencies of the power measurements time series and the transition moments between store operational mode switches (“boost”,” eco” and ”normal”).
- **Supervised Learning Algorithms** – used to generate point and uncertainty forecasts. We have compared multiple regression models, from linear models such as Linear Regression and Linear Quantile Regression, until more advanced non-linear ensemble-based models such as Random Forecasts and Gradient Boosting Trees (GBT). The best results (e.g., lower forecasting errors during a k-fold cross-validation over the training/ validation dataset) were obtained with GBT, which was then selected as reference algorithm for the flexibility forecasting task conducted by this tool. Note that GBT can also produce an uncertainty forecast, which means that a flexibility forecasting for a given certainty level (or quantile) can be generated, e.g., forecasted available flexibility with a probability of 90%.

As described in section 3.3.1.1, our models depend not only on historical observations but also on future scheduled data for the stores. However, data unavailability periods are common and should be considered when transferring models to an industrial environment. Therefore, two types of forecasting models were envisioned for this module.

1. **Primary models** – Models trained over perfect data availability conditions. That is:
   a. recent historical observations (i.e., until the day prior to the requested forecast period),
   b. future schedule data availability for the interval of dates to predict.
2. **Secondary models** – Models trained over restrict data availability conditions. That is:
   a. historical observations until two days prior to interval of dates to predict,
   b. no scheduled data for the interval of dates to predict.

Figure 8 illustrates the selection process for one of the two model types or an error message in case the minimum data requirements are not fulfilled upon the forecast launch.

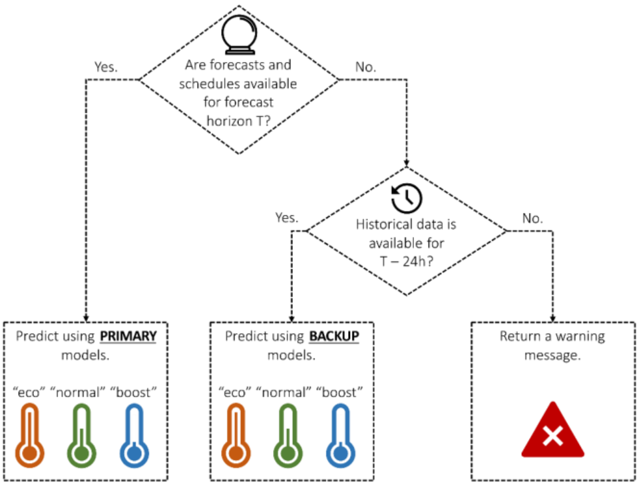


FIGURE 8: FLEXIBILITY FORECAST MODULE'S WORKFLOW DIAGRAM

As depicted by the figure, in the event of periods of data unavailability superior to the ones previously mentioned (secondary models) the forecasting module won't be able to generate new forecasts. In such events, the client requesting forecasts will receive a specific error message, informing that the necessary data requirements cannot be met to create a new forecast.

Per supermarket, there are six pre-trained models (one per operation mode and per temperature group) for each type (primary and secondary) defined above. Each model can generate forecasts, with 15-minutes time resolution, for any time interval in the current and next day. These models will be trained on a pre-scheduled basis (e.g., weekly, or monthly) using the raw data available on the database. The training pipeline consists of:

1. feature engineering applied to the raw data,
2. hyper-parameter tuning using k-fold cross-validation,
3. model fitting,
4. storage of the new model in RefriFlex filesystem, replacing the older model.

3.3.1.3 HYPER-PARAMETER OPTIMIZATION TOOLS

The GBT algorithm used in the flexibility forecast module is characterized by a set of hyper-parameters which cannot be directly inferred from the data during the models' training phase. The definition of proper hyper-parameters can significantly improve the forecast accuracy of each forecasting model. For RefriFlex, two of the most used state-of-the-art models are used, namely:

- **Grid Search**[13]: a brute-force method that searches for the optimal set of hyper-parameters by performing an exhaustive search of all possible combinations.
- **Bayesian Optimization**[14]: a method that attempts to find the maximum value of an unknown function, in as few iterations as possible, by building a constrained global optimization solver upon Bayesian techniques and Gaussian processes.

The algorithms' selection is based on two criteria:

- The necessary exploratory degree: for example, more efficient algorithms such as Bayesian Optimization can be used to find an initial good set of hyper-parameters that can later be fine-tuned using the exhaustive searching algorithm Grid Search.
- The total computational time necessary to optimize all forecasting tools.

3.3.1.4 MODELS EXPLAINABILITY

Traditionally, model explainability is tackled in GBT models by one of two methods: Mean decrease Impurity or MDI (see section 1.11.2.5 of the Ensemble Methods[15]; original article[16]) and Mean Decrease Accuracy or MDA[17] :

- **MDI**: assesses the relative importance of a feature by crediting on one hand the depth of the decision nodes where it is used and on the other the impurity metric for those nodes (i.e., the probability of reaching those nodes). Although requiring small computation times, it suffers from two major biases: the feature importance values are computed on statistics derived from the training set, which can be different for held-out datasets (i.e., new data over which predictions are made); and they favour high cardinality features, i.e., features with many unique values.
- **MDA**: simply described as the decrease in a model score (i.e., capability to better predict) when a single feature value is randomly shuffled. It is only valid when applied over models with good predictability and measures how important a feature is for a particular model. It has a much greater computational burden than MDI but does not suffer from the same flaws[18].

Given the fastness of MDI and the robustness of MDA, both scores are provided by the RefriFlex service, through its REST API. The importance scores provided by MDA can be viewed, with confidence, as the

[13] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[14] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proceedings of the 25th International Conference on Neural Information Processing Systems* - Volume 2, Red Hook, NY, USA, Dec. 2012, pp. 2951–2959.

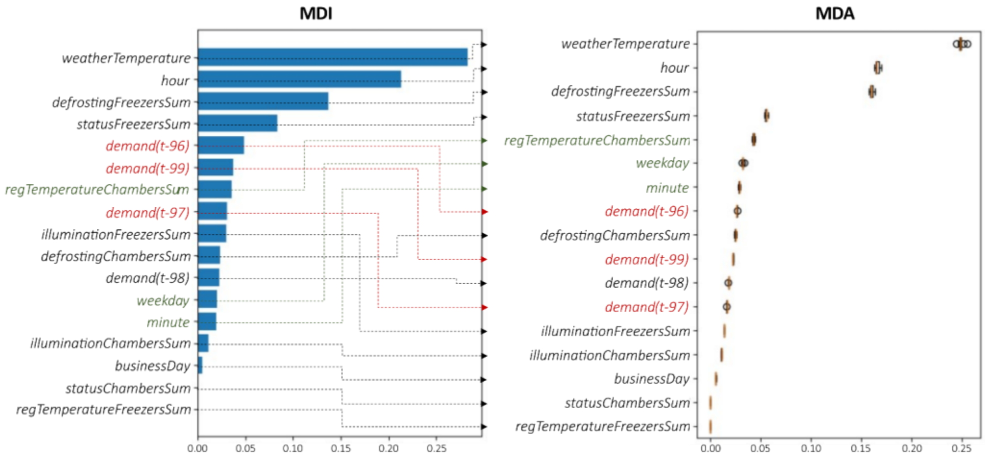
[15] https://scikit-learn.org/stable/modules/ensemble.html#feature-importance

[16] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[17] https://scikit-learn.org/stable/modules/permutation_importance.html#permutation-importance

[18] https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance.html#sphx-glr-auto-examples-inspection-plot-permutation-importance-py

relative importance of a feature for the model's predictability. MDI is provided with a however good purpose: if a feature presents a high MDI value but low MDA value, that feature can be interpreted as possibly having an influence on a particular model's predictability that stems not from its capacity to improve the model's predictability but for other unimportant characteristics (e.g., cardinality). **Error! Reference source not found.** provides an example of both metrics calculated over a specific model, where the differences between both are highlighted. Features "*regTemperatureChambersSum*", "*weekday*" and "*minute*" are features with low cardinality (in the example they have a cardinality of 3, 7 and 4, respectively). Features "*demand(t-96)*", "*demand(t-99)*" and "*demand(t-98)*" consist of float measurements, with a cardinality of over 2600. Although their importance values do not differ significantly from those of features "*regTemperatureChambersSum*", "*weekday*" and "*minute*", it is clear the influence of cardinality on MDI.



businessDay	Business day or weekend.
defrostingChambersSum	Sum of defrosting status of all chambers.
defrostingFreezersSum	Sum of defrosting status of all freezers.
demand(t-96) *demand(t-97)* *demand(t-98)* *demand(t-99)*	Active power measurement lagged variables (Δt=15').
hour	Hour of the day.
illuminationChambersSum	Sum of illumination status of all chambers.
illuminationFreezersSum	Sum of illumination status of all freezers.
minute	Quarter-minute of the hour.
regTemperatureChambersSum	Sum of regulated temperature set-pointof all chambers.
regTemperatureFreezersSum	Sum of regulated temperature set-pointof all freezers.
statusChambersSum	Number of operational chambers.
statusFreezersSum	Number of operational freezers.
weatherTemperature	Temperature forecasts (NWP).
weekday	Day of the week.

FIGURE 9: HIERARCHICAL REPRESENTATION OF THE 17 FEATURES USED IN A GBT MODEL EXAMPLE BASED ON THE MDI (LEFT) AND THE MDA (RIGHT)

3.3.2. INTEGRATION WITH INTERCONNECT DIGITAL PLATFORMS

3.3.2.1. DATA EXCHANGE

This system depends on the interoperability and proper coordination of different InterConnect digital platforms from WP4 (DSO Interface from E-REDES, Deliverable D4.2) and WP5 (EcoStruxure Building Operation from Schneider Electric; Building Operating System from Sensinov, Deliverable D5.1). **Error! Reference source not found.** illustrates all the parties and their interactions.

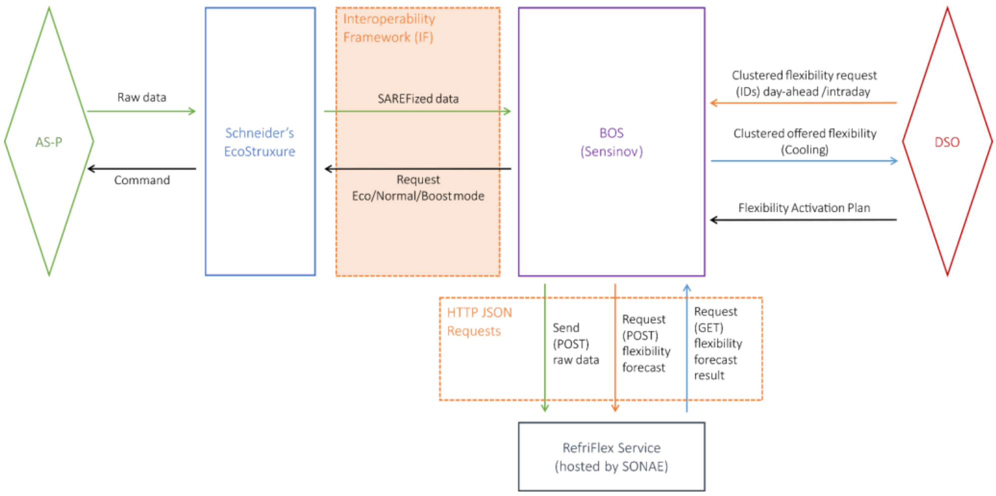


FIGURE 10: SCHEMATIC REPRESENTATION OF THE NECESSARY COMMUNICATION CHANNELS ESTABLISHED WITH ALL RELEVANT PLATFORMS FOR THE PREVISION OF DEMAND-SIDE FLEXIBILITY

DSO Interface (E-REDES): The DSO interface concept was developed as part of InterConnect's smart energy reference architecture, to ensure a standard interaction between DSOs and energy and non-energy marketplaces, ensuring neutral, transparent and secure data access to all market players. The DSO interface will enable the demonstration of InterConnect use cases adopting a standard and replicable interface with the DSO from the different pilots. The main objectives of the platform are the following: (1) Allow universal access of DER, microgrids and energy communities to flexibility and energy markets, considering different flexibility market models (including P2P markets). (2) Accommodate flexibility services designed according to the needs of the DSO. (3) Compliance with GDPR. (4) Cybersecurity. The platform specification can be found in Deliverable D4.2 "Technical Specification of DSO Interface".

The concept was developed to respond to the different InterConnect use cases involving the DSO, considering existing frameworks for flexibility integration (see Chapter 2) and considering other relevant flexibility integration interfaces (see Chapter 3).

EcoStruxure Building Operation (EBO) (Schneider Electric): This digital platform is essentially a Local Building Management System platform and can be applied to control and monitoring HVAC, Lighting, Energy Management, Fire Safety, Security & Access Control and Workplace Management Systems. This digital platform will be used in the demonstration of the Portuguese Pilot. The digital platform EBO consists in a layer of software (Enterprise Central, Enterprise Server) and a layer of hardware (SmartX Controllers). The software layer can be installed locally or hosted in the cloud. Any element of the digital platform EBO, whether software or hardware, provides the same communication protocols. This means that integration with third-party digital platforms can be done through the software or hardware layer. The architecture support standard protocols such as Modbus, Bacnet, LonWorks, KNX, MQTT and Generic Webservices like SOAP, REST and XML in the Southbound and Northbound interfaces and support also data formats like JSON and XML. Also provides data models and ontologies, and the standard implemented in our platform is Haystack and Brick Schema. The implementation of SAREF and integration with the Interoperability Framework (IF) is a functionality that is being developed within the InterConnect project.

Building Operating System - BOS (Sensinov):

Sensinov's Building Operating System (BOS) and interoperability platform help facility managers make better-informed decisions and enforce cross-building policies by providing a single, centralized interface for monitoring and controlling multi-vendor IoT ecosystems.

Within InterConnect, Sensinov will work alongside the Portuguese Demo partners to implement two of the pilot's high-level use cases:

1. **iEMS (integrated Energy Management Systems)**: Sensinov will interface with Schneider's automats to deliver an integrated view for efficiently managing lighting, HVAC, PV, EV Chargers, cold systems, and local metering solution. Benefitting from Sensinov's Hypervision module, the iEMS will provide users with a "helicopter view" of all 12 sites.
2. **HLUC 7 - Flexibility Aggregation of Commercial Buildings**: Sensinov will interface with partners Schneider, E-REDES and INESC TEC to fulfil the requirements of HLUC 7. Schneider and Sensinov will develop an interoperable interface, based on IC's Interoperable Framework[19] to collect data from the local metering solution and cold systems. Sensinov will then interface both with E-REDES and INESC TEC to gather information on the requested/available flexibility for the cold systems during the upcoming period (e.g., day-ahead). Flexibility implementation will be assured by Sensinov, who will calculate the new schedules that will optimize the consumption of the cold systems before sending them to Schneider's automats for implementation. For this HLUC, Sensinov will work on refining its current flexibility feature to provide an additional User Interface that will allow end-users to interact with this feature (e.g., choose cold system devices partic

RefriFlex Service:

For RefriFlex, four main data exchange channels are established and depicted with different colored arrows at Figure 10, namely:

- **Raw data channel**: The local controllers (AS-P) installed on SONAE MC stores enable an easy access to the refrigeration system's raw data, which is first collected by Schneider's EcoStruxure platform. These datasets are SAREFized and transmitted to BOS, via the InterConnect's Interoperability Framework (IF).
- **DSO Flexibility forecast request channel**: The Distribution System Operator (DSO) creates a flexibility request (i.e., as illustrated in Figure 11 and fully described in deliverable D4.2. section

[19] Currently under development.

3.2.2.1.1.1) and sends this information to BOS platform via HTTP requests. BOS will then parse this information and request a flexibility forecast from RefriFlex service, per supermarket store.

- **Flexibility forecast result channel**: Upon requested, RefriFlex forecasting modules compute the flexibility forecast and store the results in the internal database. These results can then be collected by BOS via an HTTP GET request (see section 3.3.1.4.1). Based on the forecasting results, BOS will then send a flexibility offer to the DSO interface.
- **Flexibility activation channel**: Based on the offered flexibility forecasts, the DSO will create a flexibility activation plan. The plan is sent to BOS which sends this information to Schneider's EcoStruxure platform, via InterConnect's IF.

As illustrated by the Figure 10, RefriFlex depends on data from multiple partners, however only BOS will interact directly with this service. The SAREFized raw data BOS received from Schneider's EcoStruxure platform through the InterConnect's IF will then be resent to the RefriFlex service through an HTTP POST request.

Clustered flexibility request
(BOS / DSO)

Field	Example
Request ID	1
Needs	100
Measurement Unit	kW
Direction (Up/Down)	Down
Type of Power (Active/Reactive)	Active
Start time	2021/09/27 10:00
End time	2021/09/27 13:00
Eligible Clients (Meter ID)	[ID Meter1, ID Meter2, …]

Request flexibility forecast
(BOS / RefriFlex service)

Field	Example
Request ID	1
Direction (Up/Down)	Down
Start time	2021/09/27 10:00
End time	2021/09/27 13:00
Eligible Supermarket Store	ID of Supermarket Store

FIGURE 11: FIELDS FROM THE STRUCTURE OF A FLEXIBILITY REQUEST PRESENTED AT DELIVERABLE D4.2 OF INTERCONNECT THAT ARE REQUIRED BY THE REFRIFLEX FORECASTING SERVICE.

3.3.2.2. REFRIFLEX FORECAST OUTPUT

The DSO flexibility needs consist of how much power needs to be increased or decrease during a specific time interval (see Figure 11). These needs will be met on a supermarket level by switching between the available set of store operation modes ("eco", "normal" and "boost").

It is important to underline that the flexibility offers created by RefriFlex result from the activation of different operation modes and not by directly defined specific power consumption setpoints, making these offers non-divisible. Moreover, these flexibility forecasts will be aggregated into a total flexibility offer (i.e., sum of the individual forecast per supermarket) by the BOS platform.

BOS will need to:

- Based on the clustered flexibility requests' eligible clients, select which supermarket stores are eligible and create a flexibility forecast request per store.

RefriFlex will need to:

1. Inspect the DSO flexibility needs request retrieving the flexibility direction (upward/downward), provision time interval and the respective temporal resolution. The right hand side table at Figure 11 shows the necessary fields of the DSO request that need to be parsed and sent to RefriFlex.
2. Assess the scheduled store operation modes baseline (i.e., default modes scheduled by SONAE MC for the next hours/ day-ahead)
3. Evaluate which modes can be activated to provide flexibility through changes in the scheduled baseline mode operation setpoints of the store's refrigeration system.

Figure 12 summarizes the complete chain logic used by the service for defining the necessary modes to forecast depending on the scheduled store operation modes baseline and the flexibility needs requested by the DSO.

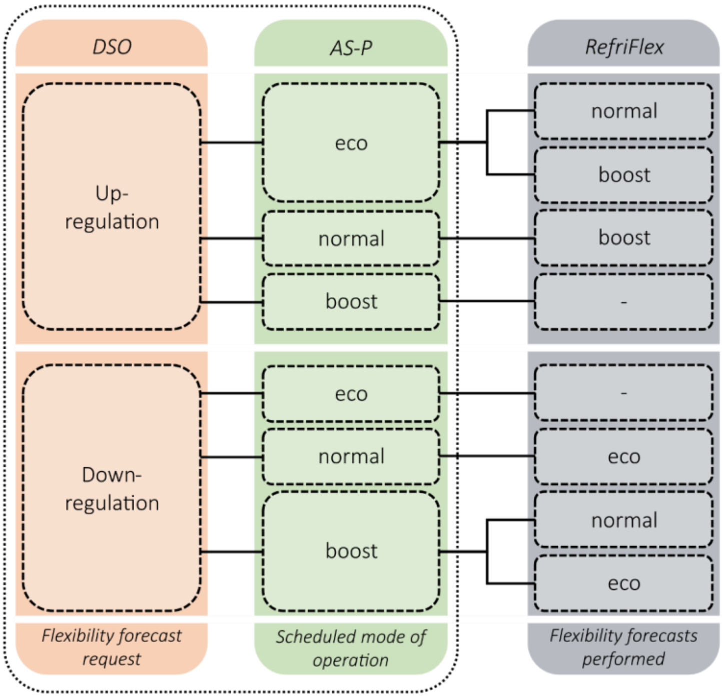


FIGURE 12: CHAIN LOGIC USED BY THE REFRIFLEX SERVICE TO DEFINE WHICH FLEXIBILITY FORECASTS WILL BE PERFORMED. THE COMBINATION OF THE DSO'S FLEXIBILITY FORECAST REQUEST FOR A GIVEN HORIZON T, PLUS THE SCHEDULED MODE OF OPERATION FOR THAT SAME HORIZON, DICTATES WHICH FORECASTS WILL BE PERFORMED BY THE REFRIFLEX SERVICE.

As an example, during periods in which the supermarket scheduled operation mode baseline is set to "eco" mode, only upwards flexibility is available (increase in consumption) through the activation of "normal" or "boost" modes. Therefore, any flexibility request for these time intervals will result in zero downwards flexibility and two forecasts of upwards flexibility (one for each available mode).

For time horizons that encompass more than one scheduled baseline mode of operation, a similar rationale is applied, being forecasts produced for up to two different operation modes ("normal" and "boost" modes for up-regulation and "normal" and "eco" modes for down-regulation). **Error! Reference source not found.** illustrates the subset of forecasts computed by RefriFlex considering a scheduled store operation mode baseline and different flexibility requirements (upwards or downwards).

AS-P

Timestamp	Scheduled mode	If up-regulation..			If down-regulation		
		Forecast for “eco”	Forecast for “normal”	Forecast for “boost”	Forecast for “eco”	Forecast for “normal”	Forecast for “boost”
01:00:00	eco	0	kW	kW	0	0	0
02:00:00	eco	0	kW	kW	0	0	0
03:00:00	normal	0	0	kW	0	kW	0
04:00:00	normal	0	0	kW	0	kW	0
05:00:00	normal	0	0	kW	0	kW	0
06:00:00	boost	0	0	0	kW	kW	0
07:00:00	boost	0	0	0	kW	kW	0
08:00:00	normal	0	0	kW	0	kW	0

RefriFlex Service

FIGURE 13: EXAMPLE OF THE OUTPUTS PROVIDED UNDER A CERTAIN SCHEDULE OF OPERATION FOR AN UPWARD AND DOWNWARD REGULATION REQUEST

Despite having multiple forecasts computed depending on each scenario, only one forecast (for a single mode) will be included in the final response to BOS flexibility result request. The selection of the operational mode to activate and its forecasts is made using the following hierarchical criteria:

1. Evaluate which offer best complies with the flexibility request made by the DSO. RefriFlex prioritizes the offer encompassing the amount of flexibility closest to the one asked by the DSO’s request and that can offer it throughout the majority of the request’s horizon. Picking the example illustrated at **Error! Reference source not found.**, and assuming that the DSO’s flexibility request cannot be entirely met by the store’s offer, an up-regulation request would be met by the forecast for “boost” mode, instead of the “normal” mode, since it offers more flexibility and for most of the request’s horizon.

2. Evaluate which offer has a smaller impact on the store’s baseline of operation. Although expected that a more “aggressive” operation mode such as “eco” or “boost” (depending on the direction of the consumption modulation), incurs in a greater amount of flexibility, some scenarios of operation prior and during its activation may prove the contrary. In the event of obtaining forecasts that equally serve the interests of the DSO (i.e., same amount of flexibility and same duration), then the chosen forecast to be offered will be the one that considers the less abrupt change from the system’s scheduled baseline. This means that if the system was scheduled to operate in “eco”/ ”boost” mode during the flexibility provision horizon, a change to “normal” operation mode would be preferred over a change to “boost”/”eco” mode.

Each forecast result (to be retrieved by BOS) includes:

- **Forecast time series**:

    - Active power that can be reduced or increased at each (hourly) timestamp of the requested period.
- **Forecast metadata**:
    - Forecast launch time [UTC].
    - Forecast active power units [kW].
    - Identifier of the operation mode to be set on a store level, to activate the forecasted flexibility for the requested period [“eco”/ ”normal”/ ”boost”].

With this information, BOS will be able to provide the flexibility offers to the DSO and map the flexibility activation plan (later provided by the DSO) with the store operation mode associated with the flexibility forecast result.

3.3.3. SUB-SYSTEM FUNCTIONS

This section contains a description of the sub-system functions as well as the respective code references in Python programming language. It is organized as follows:

a) Description of each module’s functionalities.
b) Correspondence between each functionality and the respective code reference.

3.3.3.1. DATA MANAGEMENT TOOLS

3.3.1.4.1 DATA TRANSFER MODULE (RESTFUL API)

The REST API server used by this platform was developed in Python3, mainly using a powerful and flexible toolkit for building Web APIs: Django REST Framework library[20].

Each endpoint of the REST API also includes auxiliary functions to validate the JSON requests’ payload of external clients and assure that all the data sent via REST API follows the necessary structures and data types (i.e., as defined in the REST API documentation in **Error! Reference source not found.**) prior to its storage in RefriFlex internal database.

Every interaction with the REST API (sending or requesting data) should respect the respective request structures. Every request is met with a response message, indicating the success of the data transfer process once it is concluded or if any problem was verified at the RefriFlex service endpoint.

The following main functionalities are available to external clients, via HTTP requests to our REST API endpoints.

1. **Account registration** – Method used by external clients to register in the forecasting platform.

[20] https://www.django-rest-framework.org/

2. **Account authentication (login)** – Method used by external clients to login in the forecasting platform. This method will return a bearer token that should be included in the header of the endpoints below.
3. **List stores and store resources** – Methods used by external clients to list all the stores registered in the platform and the identifiers for every store equipment considered by RefriFlex. These identifiers will be used by the client to send raw data to the platform.
4. **Data ingestion** – Methods used by external clients to send raw data into the platform database.
5. **Request flexibility forecasts** – Method used by external clients to request a new flexibility forecast. The DSO flexibility requirements are provided as payload in this request and a forecast ID is returned to the client, to later get the flexibility, forecast results.
6. **Get flexibility forecast results** – Method used by external clients to obtain the flexibility forecast result, for a specific ID (provided in request query parameters), matching a desired time resolution (chosen from 15-, 30- or 60-minutes).
7. **Get models features' explainability metrics** – Method used by external clients to obtain MDI and MDA metrics of the models (primary and secondary) of a particular store (provided in request query parameters).

Table 2 links each of the functionalities to the respective REST API Endpoints.

TABLE 2 - CORRESPONDENCE BETWEEN EACH MODULE FUNCTIONALITY AND REST API ENDPOINT (A COMPLETE ENDPOINT DESCRIPTION IS AVAILABLE ON ERROR! REFERENCE SOURCE NOT FOUND.**).**

Function Type	REST API Endpoint
1. Register	[POST] /account/register
2. Authentication	[POST] /account/token
3. List stores and store resources	[GET] /manager/store [GET] /manager/store-resource
4. Data Ingestion	[POST] /data/store/active-power [POST] /data/store/mode [POST] /data/resource/defrosting [POST] /data/resource/illumination [POST] /data/resource/temperature-setpoints [POST] /data/resource/status
5. Request flexibility forecasts	[POST] /forecast/flexibility/request
5.1 List forecast requests performed to RefriFlex	[GET] /forecast/flexibility/request
6. Get flexibility forecast results	[GET] /forecast/flexibility/result
7. Get explainability metrics	[GET] /forecast/feature-importance

3.3.1.4.2 NUMERICAL WEATHER PREDICTIONS ACQUISITION MODULE

The Weather Data Acquisition Module encompasses different tasks that guarantee continuous and reliable weather data support to the forecasting modules. The following tasks

1. **Weather acquisition** – Retrieve NETCDF4 files from Meteogalicia THREDDS server, with NWP data provided by the WRF model.
2. **Data Inspection** – Inspect NETCDF4 files and identify any corrupted files
3. **Data Extraction** – Extract NWP variables from the weather files, for the surroundings of the demonstrator pilot location.
4. **Data storage** – Save raw data variables the forecasting platform database.

Table 3 shows the correspondence between each functionality and respective Python code reference.

TABLE 3 - NUMERICAL WEATHER PREDICTION (NWP) ACQUISITION FUNCTIONALITIES AND RESPECTIVE PYTHON CODE REFERENCES

Function Type	Python Class Method / Python Function
1. Weather acquisition	MeteogaliciaWeather.collect_daily_meteogalicia_dataset()
2. Data Inspection	MeteogaliciaWeather.check_if_corrupted()
3. Data Extraction	MeteogaliciaWeather.process_netcdf4()
4. Data Storage	MeteogaliciaWeather.upload_to_database()

These processes will run based on scheduled tasks (CRON jobs) for specific hours of the day in order to accommodate the most recent NWP information available on the weather provider.

3.3.1.4.3 FLEXIBILITY FORECAST

This tool is responsible for creating the final flexibility forecasts. Different stages are considered to assure that the minimum data availability requirements are fulfilled, to process raw data and to call the pre-trained forecasting models in order to generate forecasts for the different store operation modes. The following functionalities are available.

1. **Load historical data** – Perform queries to the platform's database and gather recent (or full) historical datasets from all the raw data variables available.
2. **Verify minimum data requirements** – Inspect the loaded dataset and check if the minimum data requirements are fulfilled. If not, break the pipeline here and inform the client that the necessary data requirements are not fulfilled.
3. **Perform feature engineering** – Apply feature engineering to the loaded dataset and check which forecasting models should be used (primary or secondary – see section 0).
4. **Select store operation modes to predict** – Select which store operation modes should be considered in the forecasting process ("eco"/"normal"/"boost" – see section 3.3.2.2) and load pre-trained models for each mode.

5. **Train forecasting models** – This functionality is not triggered by external client requests but should be called by predefined scheduled tasks to train the forecasting models with the most recent data. Hyper-parameter optimization is included in the training process.
6. **Create flexibility forecasts** – Computes individual forecasts per operation mode and calculates final flexibility forecasts.
7. **Store forecasts** – Stores forecast result (for every model) in the database, to be later accessible via REST API.
8. **Generate feature explainability metrics** – This functionality calculates and retrieves the MDI and MDA metrics for the models (primary and secondary) used when forecasting flexibility for a given store.

Table 4 show the correspondence between each functionality and respective Python code reference.

TABLE 4 - FLEXIBILITY POWER FORECASTING SUB-MODULES FUNCTIONALITIES AND RESPECTIVE PYTHON CODE REFERENCES.

Function Type	Python Class Method / Python Function
1. Load historical data	RefriFlex.load_dataset()
2. Verify available minimum data requirements	RefriFlex.verify_available_data()
3. Feature engineering	RefriFlex.feature_engineering()
4. Select operation modes	RefriFlex.select_op_mode()
5. Train forecasting models	RefriFlex.train_model()
6. Create flexibility forecasts	RefriFlex.forecast()
7. Store forecasts	RefriFlex.store_forecasts()
8. Generate explainability metrics	RefriFlex.feature_importance()

3.3.4. IMPLEMENTATION DETAILS

3.3.4.1. TECHNOLOGY READINESS LEVEL

This RefriFlex component results from the combination of methods and software developed in previous projects in energy analytics (e.g., CognitiveLoad platform for time series modelling and forecasting) with new modelling developments and communication interfaces specifically designed for InterConnect. The forecasting models were validated and demonstrated using historical real data from SONAE MC stores refrigeration systems. As such, current TRL for this software is 4 “technology validated in laboratory” and the expected TRL is up to 7 “system prototype demonstration in operational environment” after the Pilot demonstration in Portugal.

3.3.4.2. SOFTWARE DETAILS

The software source code is written in Python Programming Language (version 3.9). Table 5 presents a list of the most relevant libraries, together with a brief description and identification of the software license.

TABLE 5 - FLEXIBILITY POWER FORECASTING SUB-MODULES FUNCTIONALITIES AND RESPECTIVE PYTHON CODE REFERENCES.

Library	Description	URL	License
netCDF4	netcdf4-python is a Python interface to the netCDF C library, most common library to manipulate nc4 files (weather data).	http://unidata.github.io/netcdf4-python/	Unlicense
siphon	Collection of Python utilities for retrieving atmospheric and oceanic data from remote sources.	https://github.com/Unidata/siphon	BSD
joblib	joblib is a set of tools to provide lightweight pipelining in Python. Used for parallel computing, logging and tracing.	https://pypi.org/project/joblib/	BSD
scipy	Open-source software for mathematics, science, and engineering.	https://www.scipy.org/	BSD
numpy	Fundamental package for scientific computing with Python.	http://www.numpy.org/	BSD
pandas	Python library providing high-performance, easy-to-use data structures and data analysis.	https://pandas.pydata.org/	BSD
scikit-learn	Machine-learning library for Python.	http://scikit-learn.org/	BSD
statsmodels	Python library that integrates different statistical models used for forecast and statistical data exploration.	https://www.statsmodels.org/	BSD
bayesian-optimization	Constrained global optimization package built upon Bayesian inference and Gaussian process.	https://github.com/fmfn/BayesianOptimization	MIT
sphinx	Documentation generator.	http://www.sphinx-doc.org/en/master/	BSD

Django-rest-framework	Django REST framework is a powerful and flexible toolkit for building Web APIs.	https://www.django-rest-framework.org/	Encode OSS Ltd.
loguru	Logging library for Python.	https://loguru.readthedocs.io/en/stable/index.html	MIT

4. SAREF TO ML INPUT

Our goal is to create forecasters by learning from IoT data where all the IoT data will be available through the Knowledge Engine as a SAREF knowledge graph. However, most forecasters are unable to learn from linked data, instead they require numerical input or numerical representations of the input. This section will present two different methods to transform the SAREF knowledge graph into data that is interpretable by ML methods that are commonly used to create forecasters. Links to examples are included that show how the methods could be implemented.

The most straightforward approach would be to query the knowledge graph for all the numerical measurements in the SAREF knowledge graph, resulting in a collection of data points with all the measurements made by all the devices available (this is detailed in Section 4.2). But this approach would lose out on all the additional information that is available in the SAREF knowledge graph, such as: in which room is this measurement made, what values are measured by other devices in this same room (or house, or building), what was the measured value two timesteps before this measurement, etc. Section 4.1 describes methods that create numerical representations for each node in a knowledge graph, that are able to represent the additional information included in the graph.

4.1. EMBEDDINGS ON SAREF DATA

The idea of a numerical representation, or an embedding, for a node is taken from the world of Natural Language Processing, where you have the literal word, and the semantic meaning of that word. Because the context of a word is relevant to the meaning of a specific word they created numerical representations for every word that is based on the words that are occur "near" that word in sentences in (large) texts used for training. This results in a multi-dimensional embedding vector, with similar words having vectors that are closer together to each other and dissimilar words having vectors that are more apart.

Using this approach on knowledge graphs, where the knowledge graph functions as the text, and *walks* through the text function as the sentences, where each word is a node from the graph and the next "word" is a node connected to the current node.

In the following repository: https://gitlab.inesctec.pt/interconnect/SAREF-RDF2vec-into-MLP an example can be found (including the Jupyter notebook), that demonstrates how such an approach would take the SAREF knowledge graph, create embeddings for relevant nodes, and use this embedding to train a Multi-Layered Perceptron (MLP) model. The MLP model can then be used as a forecaster.

In the future it will be interesting to revisit this method when we have more data from more different devices to see whether differences between them can be learned, since these are more likely to have distinguishable different neighbourhoods.

4.2. FEATURE EXTRACTION ON SAREF DATA

Feature extraction uses the numerical values that are available in the SAREF knowledge graph. We query the SAREF knowledge graph to retrieve all the measurement values from specific devices (e.g. all the devices in one home or one room) which are collected in a large table, with columns for each device and with each row collecting the measurements made for each specific timestamp.

This process benefits from the SAREFization process, since all the data from the different devices is collected and standardized within the knowledge graph, so we can reliably query the graph instead of having to combine the data from the different devices / datasets manually.

Our example of how feature extraction can be implemented, in combination with a MLP forecaster can be found here: https://gitlab.inesctec.pt/interconnect/SAREF-Feature-Extraction-into-MLP

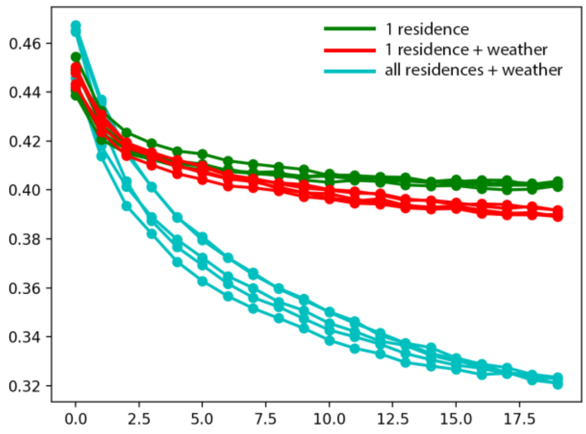


FIGURE 14: GRAPH VISUALIZING THE LEARNING PROCESS OF THE MLP, WITH THE NUMBER OF EPOCHS ON THE X-AXIS, AND MAE ON THE Y-AXIS

Using that code we created Figure 14 that displays the loss curves of training a forecaster that predicts the energy consumption from the heat pump in residence 1 from the OPSD data of 1 residence, all the residences, and with additional weather information. After every epoch the mean average error (MAE) decreases, showing that with more information, weather data or data from more residences, the descent is quicker and deeper, meaning that the model generalizes better.

5. CONCLUDING REMARKS

Commented [SC(1]: I think the conclusion needs to be bigger so that someone can read this section and get a rich summary of what the whole document is about.

This document provides an outline of, and references to the various pieces of work that were conducted as part of this deliverable. Machine Learning plays an important role in many of the services and in the pilots they are deployed. Since SAREF allows us to have a uniform representation of the data that forms the input and output for the ML algorithms, we explored various aspects like automated transformation into common data input formats and even applying ML directly on the graph itself. Next to this, we show the possibilities on gaining *explainability* that increases the transparency on the output of the algorithms which are key in situations related where human health, safety and fundamental human rights are involved. We hope that the coherency between wide variety of contributions became clear in this accompanying written work. Future work lies in applying the solutions on more Interconnect datasets and services with the intent to form a guideline accompanied with tools to gain insight and contribute to explainability for the ML algorithms within the further development of the pilots and InterConnect as a whole.