



SOGNO

D3.3

Validation and description of the techniques, interfaces and services for autonomous and self-healing power systems

The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Program, under Grant Agreement no 774613.

Project Name	SOGNO
Contractual Delivery Date:	31.10.2019
Actual Delivery Date:	05.11.2019
Author(s):	RWTH
Workpackage:	WP3– Advanced data analytics techniques for autonomous and self-healing systems
Security:	P
Nature:	R
Version:	V4.0
Total number of pages:	42

Abstract

This document presents the final implementation of the services for autonomous and self-healing power system designed in work package WP3, together with their validation at proof of concept level. The report discusses the procedure and approaches followed for the development of the services and it provides the details associated to the definition of the input/output interfaces.

Keyword list

Autonomous and self-healing power systems, Fault location isolation and service restoration, Load prediction, Generation prediction, Machine learning, Data-driven methods.

Disclaimer

All information provided reflects the status of the SOGNO project at the time of writing and may be subject to change.

Executive Summary

Power grid sustainability and reliability are main targets to be achieved by grid operators in the future smart grid scenario. To this purpose, modern distribution grids need to be endowed with smart management and control functionalities, able to provide autonomous and self-healing features to the system. However, with the increasing penetration of new technologies connected to the system, grid management is becoming more and more complex. This calls for the use of new methodologies and additional intelligence to design smart solutions able to promptly detect possible anomalies and to quickly react by automatically reconfiguring the grid and restoring the safe and reliable conditions.

The goal of WP3 is twofold. First, the objective is to design specific algorithms that are essential to provide the distribution system with autonomous and self-healing features. The first algorithm is the Load Prediction (LP) and Generation Prediction (GP) service, which is used to identify in advance potential contingencies or conditions close to the allowed boundaries, with the goal of triggering adequate countermeasures. The second algorithm is the Fault Location Isolation and Service Restoration (FLISR), which acts in case of fault occurrence to isolate the faulty section of the grid and to automatically restore the power supply in the non-faulty areas. In addition to this, the second goal of WP3 is to investigate the possibility to adopt innovative machine learning techniques for the design of the above-mentioned services. From this perspective, WP3 thus explores the potential associated to new-generation data driven techniques for dealing with the challenges and complexity of today's power systems.

This report provides first an overview of the approaches and techniques adopted for the considered SOGNO services. Then, the proof of concept validation and performance evaluation of the designed algorithms is presented. Last but not least, the interfaces required for the implementation and integration of these services within the SOGNO platform are documented.

Authors

Partner	Name	e-mail
RWTH TI		
	Rana Alhalaseh	rana.alhalaseh@ti.rwth-aachen.de
RWTH ACS		
	Marco Pau	mpau@eonerc.rwth-aachen.de
	Daniele Carta	daniele.carta@eonerc.rwth-aachen.de
GH		
	Khaled Tabbakh	ktabbakh@gridhound.de

Table of Contents

1. Introduction	6
1.1 Related project work	6
1.2 Objective of the report.....	6
1.3 Outline of the Report.....	7
1.4 How to Read this Document	7
2. Services and techniques to achieve autonomous and self-healing power systems.....	8
2.1 Load prediction service	8
2.2 Power generation prediction service.....	10
2.3 Fault location isolation and service restoration.....	14
2.3.1 Fault identification and location detection	15
2.3.2 Fault isolation and service restoration.....	15
2.4 Models related to the load and power generation prediction services	16
2.4.1 Feed-forward deep neural network	16
2.4.2 Long short-term memory (LSTM) network	16
2.5 Models related to the FLISR service.....	16
2.5.1 Feed-forward deep neural network	16
2.5.2 Decision tree	16
2.5.3 Random forest.....	17
3. Performance evaluation	18
3.1 Load prediction service	18
3.2 Power generation prediction service.....	21
3.3 Fault location isolation and service restoration.....	23
4. Documentation of the application programming interface	26
4.1 The Interfaces	27
4.1.1 Measuring Devices with the input from Power Quality Service.....	27
4.1.2 Switches	28
4.1.3 Fault Location, Isolation, and Service Restoration.....	28
4.1.4 Load Prediction	29

4.1.5 Generation Prediction.....	29
5. Conclusion	30
6. List of Tables	31
7. List of Figures	32
8. References.....	33
9. List of Abbreviations	34
ANNEX 1 Examples of the JSON messages	35
A.1.1 Example for the JSON Message from the measuring devices with power quality service.....	35
A.1.2 Example for the JSON message from the switches.....	36
A.1.3 Example for the JSON message from FLISR	37
A.1.4 Example for a CIM file	37
A.1.5 Example for JSON file to enable or disable FLISR	38
A.1.6 Example for the JSON output message from the LP service.....	39
A.1.7 Example for the JSON output message from the GUI to manage the LP service	39
A.1.8 Example for the JSON output message from the DB with the LP necessary input	40
A.1.9 Example for the JSON output message from the GUI to manage the GP service	41
A.1.10 Example for the JSON output message from the DB with the GP necessary input	41

1. Introduction

The project *Service Oriented Grid for the Network of the Future* (SOGNO) is funded by the Work Program H2020-LCE-2017-SGS. It has officially started in January 2018.

1.1 Related project work

This deliverable describes the work done in tasks T3.4 and T3.5 of the work package WP3 as finalization of the work developed in T3.1 and T3.2, whose preliminary results were already presented in the previous deliverable D3.1. More in detail, T3.1 dealt with the investigation of innovative concepts coming from the machine learning field that can be used for the design of autonomous and self-healing services, while T3.2 focused on the design of two specific services, namely the Load and Generation Prediction services (LP and GP, respectively) and the Fault Location Isolation and Service Restoration (FLISR) service. The design of these algorithms has been based on the data-driven approaches deemed more suitable for this purpose, according to the analysis performed in T3.1. Subsequently to the design work, T3.4 carried out the validation of these services at proof of concept level and performed a preliminary evaluation of their performance, based on laboratory simulations. T3.5 defined instead the interfaces needed for the integration of such services in the SOGNO cloud platform in view of their final deployment on the field.

As shown in Figure 1, WP3 and WP2 can be identified as the research WPs in the project, where innovative concepts for the development of the services have been explored resulting in the final design of the SOGNO services. The developed services have been then handed over to WP4 for their integration within the SOGNO cloud platform hosting the services, allowing thus the final deployment in the field trials.

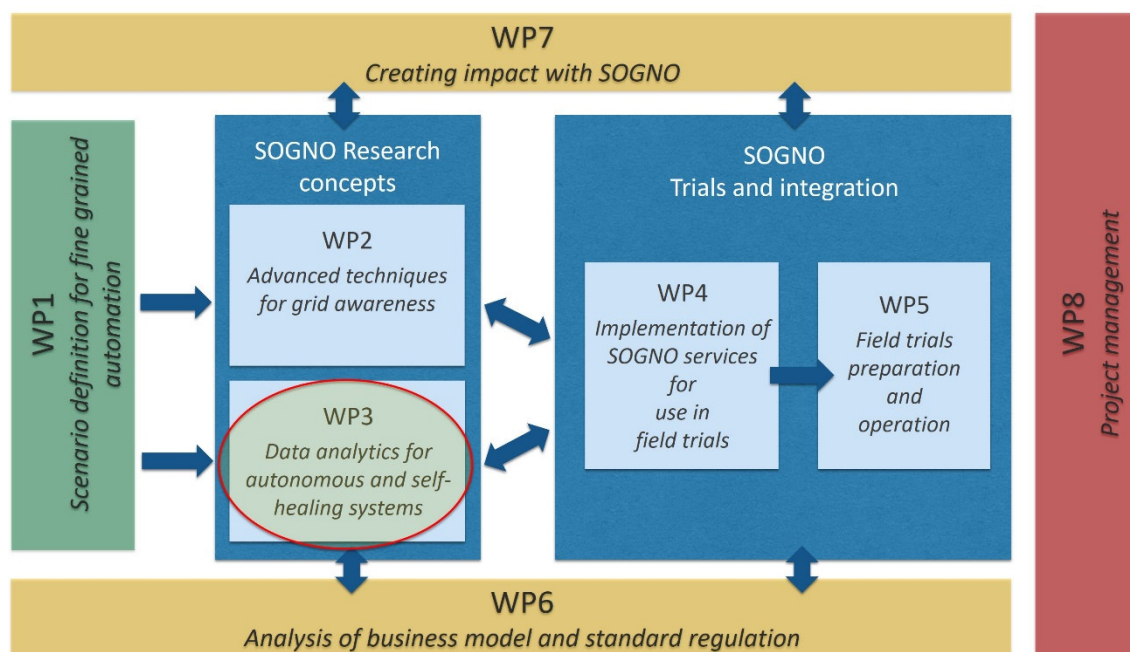


Figure 1: An overview of SOGNO activities.

1.2 Objective of the report

The main objective of this deliverable is to provide the detailed design and implementation behind the autonomous and self-healing power system services, which are part of the work conducted in WP3. The two services implemented within WP3 are the Fault Location Isolation and Service Restoration (FLISR) and Load and Generation Prediction (LP and GP). The concept and definitions of the two services as well as the machine learning based methods and techniques used for their design have been already introduced in a previous deliverable (D3.1) and are here recalled and completed with a more in-depth explanation of the services implementation. The

report then provides a proof of concept validation of the services via tests performed on some benchmark data and scenarios.

1.3 Outline of the Report

The report consists of three main parts covering the service design, the service validation and the description of the service interfaces. The first part (Chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**) introduces the two services designed to provide autonomous and self-healing features to the distribution grid, namely the Load and Generation Prediction and the Fault Location Isolation and Service Restoration (FLISR) services. Chapter 2 provides an update of the previous deliverable D3.1 regarding the details on the data-driven approaches implemented behind the algorithms as well as some additional technical aspects. The second part (Chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**) provides a validation of the developed services by means of simulations and laboratory test procedures. Finally, the last part (Chapter 4) contains the scheme of the interfaces required for integrating the services in the Virtualized Substation.

1.4 How to Read this Document

This report can be read as a standalone document. However, other deliverables can be helpful to get a better view of the concepts advanced in the SOGNO project and to have more details on the autonomous and self-healing grid services. In particular, other deliverables closely related to this one are:

D1.1 – Scenario & architectures for stable & secure grid (M12): This deliverable provides a description of power system scenarios investigated in the project for current and future distribution grids.

D3.1 – Description of new SOGNO techniques for autonomous and self-healing power systems (M10): This deliverable discusses the role of machine learning and other data driven approaches for the possible design of services aimed at providing self-healing features to the grid.

D3.4 – Description of initial interfaces & services for autonomous and self-healing power systems (M12): This deliverable gave a first presentation of the preliminary design of the grid self-healing services and the corresponding interfaces.

2. Services and techniques to achieve autonomous and self-healing power systems

The main focus in this chapter is to provide a detailed and in-depth description of the services implemented in WP3, namely:

- Load and Generation Prediction (LP and GP, respectively)
- Fault Location Isolation and Service Restoration (FLISR)

Machine learning and data analytics algorithms are the main methods employed for the design of these services. These approaches have proved their reliability in several fields and applications, such as classification, regression and pattern recognition problems. Several models can be implemented for the design of the LP, GP and FLISR services. For example, one can consider the load prediction service as a regression problem, where a suitable machine learning function defines the relationship between the input (or trained data) and output (or predicted data). Different machine learning models can have different characteristics, resulting into more or less suitable methods for addressing the problem at hand. In the following, the models (and the underlying concepts behind them) used for the design of the LP, GP and FLISR algorithms are presented.

2.1 Load prediction service

The purpose of this service is to provide the forecast of load or power consumption at a specific connection point of the grid for future time slots based on the power consumption at past time slots, i.e. based on the historical power consumption. The forecast can refer in general to a particular load (e.g. an industrial or commercial load) or to an aggregation of several downstream loads (e.g. prediction of the consumption at a MV/LV substation supplying several residential customers). The load or power consumption utilized to train the machine learning based model is usually sampled at a certain rate (for example every 1, 5, or 10 minutes, which can be the reporting rate of the measurement devices or the result of an a posteriori aggregation of historical measurement data); as a result, the achievable output is the power consumption predicted with the same reporting rate. Figure 2 shows an example structure of a dataset of power consumption sampled in a one-minute rate.

```
1/1/2007;00:00:00;2.580;0.136;241.970;10.600;0.000;0.000;0.000
1/1/2007;00:01:00;2.552;0.100;241.750;10.400;0.000;0.000;0.000
1/1/2007;00:02:00;2.550;0.100;241.640;10.400;0.000;0.000;0.000
1/1/2007;00:03:00;2.550;0.100;241.710;10.400;0.000;0.000;0.000
1/1/2007;00:04:00;2.554;0.100;241.980;10.400;0.000;0.000;0.000
1/1/2007;00:05:00;2.550;0.100;241.830;10.400;0.000;0.000;0.000
1/1/2007;00:06:00;2.534;0.096;241.070;10.400;0.000;0.000;0.000
1/1/2007;00:07:00;2.484;0.000;241.290;10.200;0.000;0.000;0.000
1/1/2007;00:08:00;2.468;0.000;241.230;10.200;0.000;0.000;0.000
1/1/2007;00:09:00;2.486;0.000;242.180;10.200;0.000;0.000;0.000
1/1/2007;00:10:00;2.492;0.000;242.460;10.200;0.000;0.000;0.000
1/1/2007;00:11:00;2.500;0.000;242.880;10.200;0.000;0.000;0.000
1/1/2007;00:12:00;2.494;0.000;242.570;10.200;0.000;0.000;0.000
1/1/2007;00:13:00;2.492;0.000;242.410;10.200;0.000;0.000;0.000
1/1/2007;00:14:00;2.480;0.000;241.810;10.200;0.000;0.000;0.000
1/1/2007;00:15:00;2.478;0.000;241.730;10.200;0.000;0.000;0.000
```

Figure 2: A sample of input data used for load prediction service.

The one-minute power consumption data shown in Figure 2 is taken from an household electricity consumption dataset in [1]. The first column gives the date, the second column is the particular time instant within the day when the power is measured. The third and fourth columns represent the household global minute-average active and reactive power consumed in kilowatt, respectively. These values are those that are relevant for the training of the LP service and they will be implicitly considered from now onwards (the rest of the columns in Figure 2 are the active energy read via three sub-meters, which are of no use for the designed LP service). As explained earlier, since the available consumed power provided is in a one-minute rate, the predicted power in this case can also be in a one-minute rate. Alternatively, if such time resolution is not needed, the available data can be post-processed to create aggregated profiles with lower granularity. If

aggregated data are provided in input for the training, the same time resolution used for the data aggregation automatically defines the time resolution of LP service forecast.

The first task of the load prediction service is to prepare the data necessary to carry out the prediction. Depending on an “a priori” data analysis, the data can be clustered in different ways in order to improve the load prediction performance. For example, in presence of a dataset covering several years of historical power consumption, it is possible to analyze the possible presence of specific patterns at seasonal level, monthly level, or to simply distinguish working days from holydays and weekends. In general, data clustering can be applied if there is a sufficient amount of data (usually several years of data). In this case, for example, the active and reactive power consumption for one month of the year (or one day of the week) can be used to train a machine learning based method in order to predict these values for the same month in the next year (or for the same day in the next week).

Figure 3 represents, as an example, the block diagram of the load prediction service, when the active and reactive power is classified based on the month of the year (or the day of the week) to create appropriate consumption profiles. Afterwards, the data per profile is used to train the associated machine learning algorithm, which is responsible for providing the forecast of consumed power at future time slots.

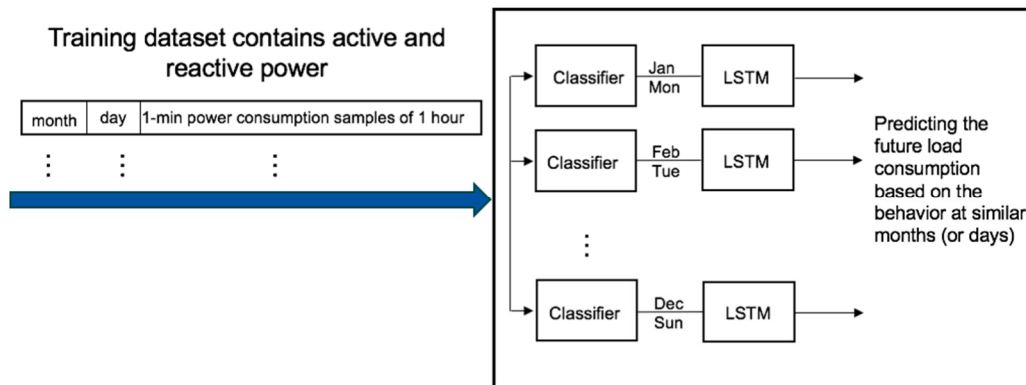


Figure 3: The block diagram of the load prediction service.

When the available set of historical data is not too large (as it is likely to expect in distribution grids with recent installations of the measurement infrastructure) the dataset cannot be clustered since this would lead to have too small datasets to be used for the training of the machine learning algorithm. In such a scenario, all the available data should be used without applying any specific clustering, and the adopted machine learning algorithm has to be able to deal with this scenario and to still provide forecasts as accurate as possible. To this purpose, it is important to adopt deep learning algorithms able to detect possible existing patterns, related to both the long-term and short-term variations of the power profiles. In order to fulfill such a requirement, the machine learning approach chosen for the implementation of the SOGNO LP service is the Long Short Term Memory (LSTM) Deep Neural Network (DNN).

The LSTM neural network is based on a concept of a sliding window to define the inputs and outputs needed for the training, and therefore for the definition of the prediction model. In [2], three different methods have been discussed for similar purposes; i.e., in the context of electrical load prediction. Approaches can for example change depending on the number of features used to determine the forecast (univariate and multivariate, respectively) or on the number of desired outputs from the prediction algorithm (single-step or multi-step prediction, respectively). In the case of the SOGNO LP service, a univariate multi-step prediction algorithm has been used in order to obtain the day-ahead prediction of a specific load with the desired time resolution. The prediction is done in the form of one-step ahead time series fashion, i.e., the quantity at time slot $t+1$ is predicted based on the observations until time slot t . This can be generally described by the following model:

$$\hat{X}_{t+1} = f(X_t, X_{t-1}, \dots, X_{t-n}) + \epsilon \quad (1)$$

where $f(\cdot)$ is the prediction model, $X_t, X_{t-1}, \dots, X_{t-n}$ are the samples of electrical power consumption until time slot t , and \hat{X}_{t+1} is the estimated power at time $t+1$. The term ϵ is the prediction error, which in general will be always present. In the case of day-ahead forecast, the quantities X indicated in the above equation will represent vectors of data associated to all the time steps within the day. For example, if the load forecast is performed with a 15-minutes resolution, each variable X will represent a 96-size vector of the power consumption for the day t . As a consequence, the above-mentioned relationship indicates that several days have to be used for the prediction of the power consumption in the following day.

Based on this concept, several inputs containing the load consumption associated to N days are used, together with the associated power consumption at day $N+1$, to train the LSTM machine learning algorithm. As a result of the training, a DNN model is obtained, which can be used then to perform the load prediction. The specific settings of the LSTM model (such as number of layers for the DNN, number of neurons in each layer, etc.) can be flexibly modified and selected according to the specific scenario where the service has to be implemented. In general, a first phase of tuning of the algorithm with the specific data of the location under analysis is recommended in order to customize the model accordingly and to aim at prediction performance as accurate as possible.

For its own characteristics, the implemented LSTM prediction algorithm can be also easily adapted to perform predictions with different time horizons. While day-ahead forecasting is often the most desired feature for Distribution System Operators (DSOs), additional predictions with shorter time scale (i.e. one hour-ahead) can be also important for DSOs to refine their operation plans. Such a characteristic can be above all important for possible contingency prediction, since the one-hour ahead forecast can rely on more updated information and can be consequently more accurate and reliable. Taking into account this possible need for the DSOs and the possible use of this service for contingency prevention (to obtain preventive self-healing grids), both the day-ahead and the one-hour ahead forecasting has been implemented for the LP SOGNO service. More details on the performance of these algorithms, according to the specific considered scenario, are presented in Section 3 showing the proof of concept validation of the service.

2.2 Power generation prediction service

While the same structure of the LP algorithm can be theoretically adopted also for the GP service, in case of generation based on renewable energy sources additional variables and parameters should be taken into account to improve the forecasting accuracy. In fact, weather and environmental conditions can have a strong correlation with the level of generated power. In the scope of SOGNO, we focus on predicting the power generated in photovoltaic (PV) systems due the solar power in the environment. For this matter, the weather information and the associated variables are essential to this service. In other words, the weather information is the input to the machine learning predictor to provide the amount of generated power due to the renewable energy source. More specifically, the feature vector is the weather information, and the target is the generated power due to the corresponding weather conditions.

To apply the above concept, two main requirements exist: i) to collect a sufficiently large set of weather/environmental data synchronized with the corresponding generation output; ii) to identify the most important weather/environmental parameters to be used as input for the generation prediction. Assuming to have the availability of such data over a large period of time, the feature (weather input) and target (generation output) matrices at the basis of the GP training can be organized as shown in Figure 4.

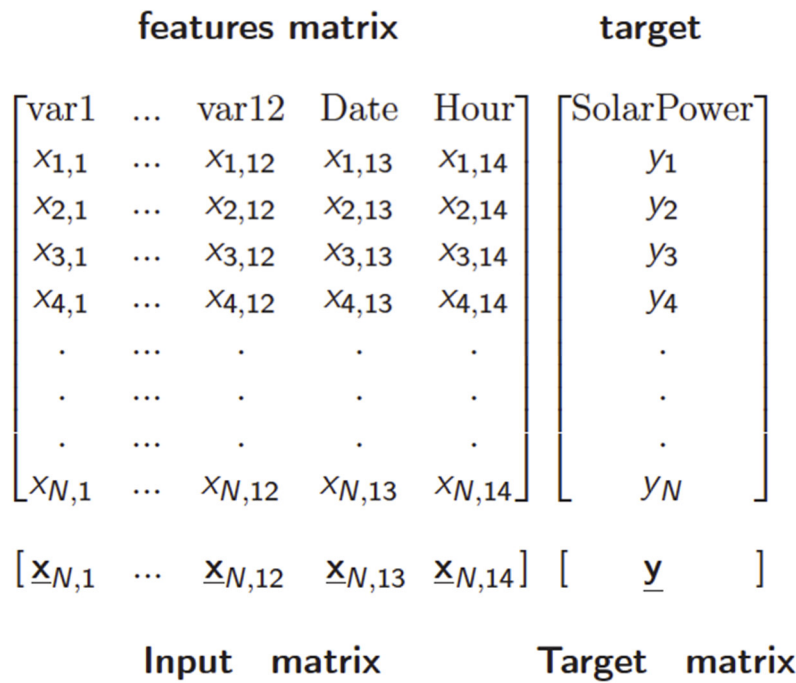


Figure 4: The structure of the raw dataset to be utilized in the power generation prediction service.

In Figure 4, the variables var1, ..., var12 indicate the generic weather variables to generate the corresponding power. N indicates the number of examples in the dataset (set of weather input with the associated generation output available in the historical database). The weather variables can be recorded, as an example, in one-hour rate, along with the corresponding power generated. The preliminary work done for the design of the GP service consists in the identification of the most relevant parameters to be used as input for the training of the machine learning algorithm. To this purpose, in the following, the dataset provided in [3] has been considered, which refers to generation data collected in Australia as part of a forecasting competition. The weather information is obtained from the European Center for Medium-range Weather Forecasts (ECMWF) and it contains historical data from 01.04.2012 until 30.05.2014 collected for several solar power plants located in Australia. Table 1 provides a detailed description of the raw data provided in [3], which have been utilized to test and validate the power GP service.

Table 1: Raw data description.

Weather ID	Weather variable description and unit.
Var1	Total column liquid weather (tclw): vertical integral of cloud liquid water content [Kg m ⁻²].
Var2	Total column ice water (tcw): vertical integral of cloud ice water content [Kg m ⁻²].
Var3	Surface pressure (SP) [Pa].
Var4	Relative humidity at 1000 mbar (r): it is defined with respect to saturation of the mixed phase, i.e. with respect to saturation over ice below -23C and with respect to saturation over water above 0C. In the regime in between a quadratic interpolation is applied [%].
Var5	Total cloud cover (TCC): total cloud cover derived from model levels using the model's overlap assumption (either 0 or 1).

Var6	10-meter U wind component (10u) [ms ⁻¹].
Var7	10-meter V wind component (10v) [ms ⁻¹].
Var8	2-meter temperature (2T) [K].
Var9	Surface solar rad down (SSRD): accumulated field [Jm ⁻²].
Var10	Surface thermal rad down (STRD): accumulated field [Jm ⁻²].
Var11	Top net solar rad (TSR): net solar radiation at the top of the atmosphere [Jm ⁻²].
Var12	Total precipitation (TP): it is the convective precipitation plus the stratiform precipitation (CP +LSP) [m].

Several profiles have been created based on the most relevant weather conditions. For this matter, feature-selection methods are utilized to identify the most important environmental features to be considered as input to the GP service. Via the feature-selection, the dimensionality of the feature space is reduced by eliminating the redundancy and the non-meaningful information in the dataset. The irrelevant features are eliminated, and the most relevant ones are included in the learning process. In this way, the complexity of the learning process is reduced, and the performance is improved via the selection of the most pertinent information.

Throughout the tests, the mutual information (MI) measure (or score) has been utilized as indicator for the feature-selection. The MI measure was originally introduced in the field of information theory. It measures the mutual dependency between two random variables. Given two random variables A and B , their MI measure $I(A;B)$ is expressed as:

$$I(A;B) = \sum_{a \in A, b \in B} p(a,b) \log \frac{p(a,b)}{p_A(a)p_B(b)}$$

where p_A and p_B denotes the probability density functions (PDFs) of A and B , respectively, and $p(a,b)$ denotes their joint PDF [2]. If $I(A;B) = 0$, this indicates that the two random variables are completely independents (thus, no correlation exists among them). As the PDFs of the dataset are unknown at this point, the approach suggested in [3] has been considered. Via the known number of samples N_A and N_B for the random variables A and B , respectively, their PDFs are estimated by sorting the measurements into k number of clusters based on the k -nearest neighbor distance. Indicating with $z_A(i)$ and $z_B(j)$ the number of measurements in cluster i and j , and with $z_{A,B}(i,j)$ the number of measurements in their intersection, the following holds:

$$p_A(i) \approx \frac{z_A(i)}{N_A}$$

$$p_B(j) \approx \frac{z_B(j)}{N_B}$$

$$p(i,j) \approx \frac{z_{A,B}(i,j)}{N_A + N_B}$$

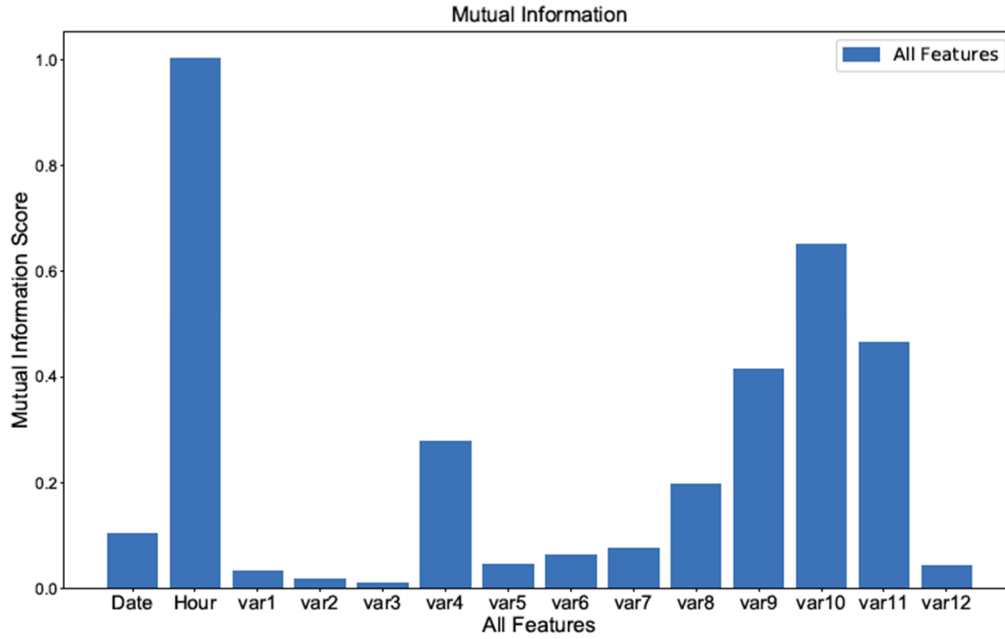


Figure 5: The MI score for the raw data.

Figure 5 presents the MI score resulting for all the features in the dataset; namely the 12 weather variables along with the date and time (in hour). It can be seen that the hour information is the most relevant and important feature to be included in the prediction process, as reasonable to expect since the PV generation is strongly dependent on the particular time of the day. After the time of the day information, the more important features to be considered are the following:

1. The hour information,
2. surface thermal rad down, denoted by var 10,
3. top net solar rad, denoted by var 11,
4. surface solar rad down, denoted by var 9,
5. relative humidity at 1000 mbar, denoted by var 4, and
6. 2-meter temperature denoted by var8.

The prediction process employs these six most relevant features. In other words, these are the most important features that should be utilized to predict the generated power. However, for the sake of simplicity, only the first two most relevant features (except the time of the day information) are utilized in the following to create four weather profiles; namely the surface thermal rad down (var 10), and top net solar (var 11). The other four weather conditions (including the hour information), while not being used for clustering the weather profiles are in any case still utilized for training the machine learning algorithm and then for the prediction. The considered weather profiles are built as follows, depending on the combination of thermal rad down and top net solar:

$$\text{Profile 1: } x_{j,10} \geq \bar{X}_{10} \text{ \& } x_{j,11} \geq \bar{X}_{11}$$

$$\text{Profile 2: } x_{j,10} \geq \bar{X}_{10} \text{ \& } x_{j,11} < \bar{X}_{11}$$

$$\text{Profile 3: } x_{j,10} < \bar{X}_{10} \text{ \& } x_{j,11} \geq \bar{X}_{11}$$

$$\text{Profile 4: } x_{j,10} < \bar{X}_{10} \text{ \& } x_{j,11} < \bar{X}_{11}$$

where $j = \{1, \dots, N\}$. The data for each profile is utilized to train a separate machine learning model. In other words, a separate machine learning model, in our case a deep neural network, is dedicated to each profile. For the prediction purposes, the future weather conditions can be collected from possible local weather provider. The appropriate profile is selected by comparing the values of the surface thermal rad down (var 10), and top net solar (var 11) with the previously described profiles. The block diagram of the service shown in Figure 6. Potentially, the same approach can be tested and implemented for further datasets, and any other weather conditions, also depending on the particular availability of data at the considered location.

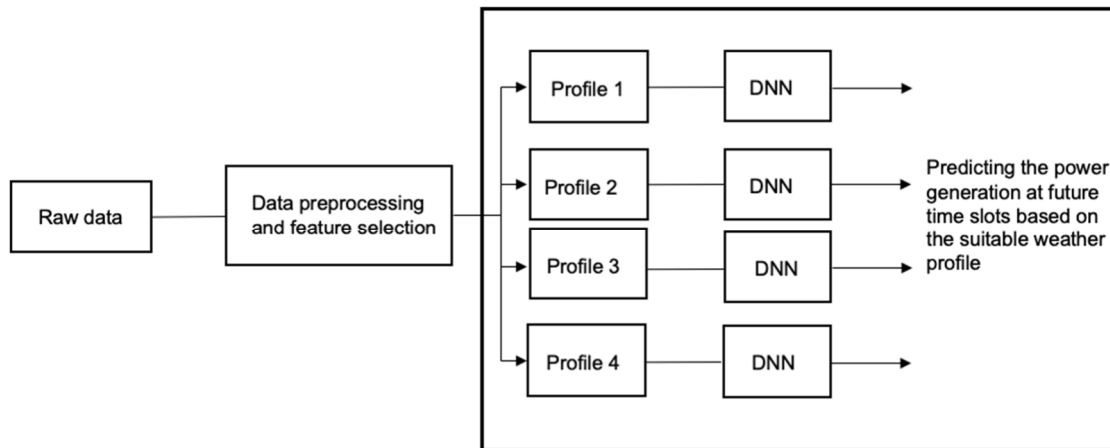


Figure 6: The block diagram of the generation prediction service.

2.3 Fault location isolation and service restoration

The fault location isolation and service restoration (FLISR) is a service providing the automatic localization of the fault and the automatic reconfiguration of the grid to minimize the extension of the outage and, consequently, to minimize also the number of customers affected by the power disruption. Due to these features, FLISR can be seen as the fundamental service providing self-healing features to the grid during emergency conditions and in case of unplanned power supply interruptions.

The FLISR service is composed of two main parts: i) the localization of the point of the grid where the fault occurred, via a dedicated fault location algorithm; ii) the isolation of the fault and restoration of the power supply in the non-faulty zones, through a suitable grid reconfiguration procedure. The most critical part of the overall process is the fault location detection, since from this it depends the possibility to correctly isolate the fault and restore the power supply for the customers connected to non-faulty areas. In the fault location process, the first requirement is to identify the type of fault that occurred (single phase to ground, three-phase to ground, etc.). In fact, depending on the type of fault, different transient conditions occur in the grid and in particular for the three phases of the system. Since in SOGNO the goal is to investigate the potential given by data driven approaches based on machine learning for the design of the service, it is of paramount importance that scenarios leading to completely different conditions are properly detected and classified. Once the type of fault is known, the FLISR service requires information on the transient conditions occurred due to the fault in order to estimate the possible location of the fault. To this purpose, fault scenarios can be created, provided that the needed details on the model of the grid are known, in order to train the algorithm and to generate the model needed for the automatic fault location.

If the fault location is carried out successfully, specific logic can be implemented in order to isolate the faulty area of the grid and to restore the power supply in the rest of the network. To this purpose, the details on the availability of breakers, switches and other possible reconfiguration devices in the grid are needed. The particular set-up of the reconfiguration logic has to be conceived taking into account the specific grid topology and characteristics and, as a matter of fact, needs to be customized for each particular grid scenario.

In the following of this Section, the considered approach to implement the FLISR service is discussed more in detail. In particular, the focus will be mainly on the FLISR part related to the fault location, since this is the module where it can make sense to apply the machine learning theory and because, as previously mentioned, this part of the algorithm is the most critical one for the successful implementation of the FLISR service.

2.3.1 Fault identification and location detection

Among the different possible fault types, the two most common types of fault have been considered in the following to validate the underlying concept behind the implementation of the fault location algorithm: single line-to-ground fault, and three-phase-to-ground fault. Once data from installed sensors are collected, i.e., measurements are read from sensors, the fault is categorized into four cases, or classes: phase A-to-ground (AG), phase B-to-ground (BG), phase C-to-ground (CG), three-phase-to-ground (ABCG). This is done looking at the particular combination of voltage and current measurements at the three phases of the system, knowing that each class of faults exhibits a different behavior at the direct, inverse and homopolar sequences. Figure 7 presents the block diagram of the fault classification approach. A different machine learning based algorithm is then used for each category of fault. Since a large set of historical data on fault-related measurements is usually rarely available for DSOs, the training of the machine learning algorithm has to rely on the set-up of specific simulation scenario (customized for the particular grid under analysis) and on the emulation of the different categories of faults. Through the generation of a large dataset of fault-related data, it is then possible to train the algorithms emulating the acquisition of measurement at grid locations where measurement instruments are available and providing these data, together with the associated location of the fault, in input to the training process. In such a way, a location estimation model can be obtained for each category of fault, which can then be used online to operate when actual faults occur. As easy to imagine, since the input data come from simulated scenarios, the successful implementation of the fault location algorithm and the correct training of the machine learning routine are strictly depending on the availability of accurate models of the grid.

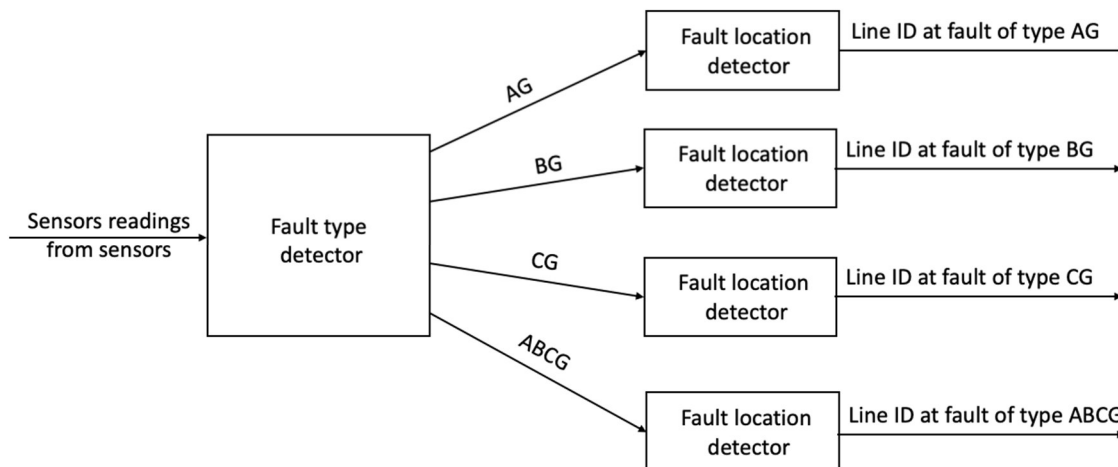


Figure 7: The block diagram of fault location detection within FLISR service.

2.3.2 Fault isolation and service restoration

The routine for the automatic reconfiguration of the grid aimed at isolating the fault and restoring the power supply in non-faulty zones is strictly dependent on the particular topology, characteristics, and availability of reconfiguration devices along the network. Generally speaking, the isolation procedure is implemented by opening the closest switches upstream and downstream the fault (note that the location of the fault is needed as input from the previously described fault location algorithm). The service restoration phase instead consists in closing those switches that would allow to supply back disconnected customers, without triggering again the fault. The particular reconfiguration of the grid also depends on the meshed or radial topology of

the grid, as well as on the number and position of the sectionalizing switches. To this purpose, in the SOGNO FLISR service, dedicated routines are implemented for each test grid (or field trial) according to the particular structure of the network.

2.4 Models related to the load and power generation prediction services

2.4.1 Feed-forward deep neural network

Feed-forward deep neural network (FFDNN), (also known as multilayer perceptron) is a type of artificial neural network. As a supervised machine learning method, DNN has drawn more attention in the last decades due to its outstanding performance in different fields; such as classification, regression, and pattern recognition tasks. The FFDNN are implemented with multiple hidden layers. Each hidden layer contains a certain number of neurons, with certain weights. The weighted input is summed over all layers and transformed through the activation functions through until the output layer [5]. The training process takes place such that the cross-entropy function $\mathcal{L}(f(X, \theta), t)$ is minimized, where $X = [x_1, x_2, \dots, x_N]' \in \mathbb{R}^{N \times M}$ is the training set of $N \in \mathbb{N}$ samples, and $t \in \mathbb{R}^{N \times 1}$ is the predicted target. The observation $x_i \in \mathbb{R}^{M \times 1}$ (with M is the number of features, and $i \in \{1, \dots, N\}$) is the weather conditions in the case of power generation prediction service, or the load consumption for the last one hour in the case of load prediction service. Moreover, the function $f(\cdot)$ represents the input-output relationship of the network. The set $\theta = \{W^{(d)} \mid d = 1, \dots, D\}$ denotes the weights of the D hidden layers, where $W^{(d)} \in \mathbb{R}^{r_d \times r_{d-1}}$ is the weight for the d^{th} layer, which has r_d neurons [3].

2.4.2 Long short-term memory (LSTM) network

Unlike the traditional DNN, recurrent neural network (RNN) can consider sequences of data, such as time series data. A type of RNN is the long short-term memory (LSTM) network, where it has recurring connection to itself which provides the ability to consider previous input (at past time slots) for prediction. LSTM has overcome the vanishing gradient problem, which is found in RNN [6], by introducing a so-called cell state.

Due to its structure, the LSTM network is capable of learning long-term dependencies. The LSTM network consists of a chain of repeated modules of a certain structure, each module is known as a unit or a cell. This structure replaces the usual hidden units or layers found in a regular DNN. A typical unit consists of a cell gate, an input gate, an output gate and a forget gate [7]. The cell gate remembers the values over arbitrary time intervals, and the three other gates control the flow of information in and out of the cell. The forget gate allows the network to learn which information to forget and remove from the cell state. The input gate controls the amount of information to include in the cell state, and the output gate controls the information to be added to the cell state. Detailed information can be found in [6].

2.5 Models related to the FLISR service

2.5.1 Feed-forward deep neural network

The FFDNN is as described in 2.4.1, however with the target value is a discrete set of classes $t \in \mathbb{N}^{N \times 1}$ instead of real numbers as in the case of prediction.

2.5.2 Decision tree

A decision tree (DT) is a non-parametric supervised learning method. It is capable of predicting the target class based on learning from simple decision rules implied from the training dataset [12]. As indicated by the name, a DT classifier has a tree structure with nodes, edges and leaves. Each node in a DT represents a certain feature in the training dataset, whereas each edge or branch coming out of a node represents a certain decision rule applied to split the dataset into subsets. The decision is a binary rule most of the time. The splitting is terminated until the resulting subsets are homogeneous, i.e., all samples belong to the same fault class. The node is then terminated, and a leaf is added representing the class [4].

2.5.3 Random forest

Random forest is an example of an ensemble approach, as in specific bootstrap sampling. The underlying idea is to combine a group of DTs in order to construct a more powerful and more robust learner against over-fitting [8]. The main characteristic of a RF is that it is built based on randomness. Unlike DT, the split in RF is based on the best among a subset of randomly chosen predictions at that node [9] [10]. The main idea behind the randomness is to decorrelate the trees, such that the resulting ensemble scheme has a low variance [11].

3. Performance evaluation

3.1 Load prediction service

The LP service has been tested in different scenarios in order to evaluate the potential performance of the implemented LSTM-based algorithm when dealing with different possible conditions. To this purpose, the energy consumption readings of single households given in [13], which were provided in the framework of a research project in UK have been considered as starting dataset. These data provide smart meter reading with 30-min resolution for 5567 households in London, for one or more years. In the tests presented in the following, different sets of households have been considered, either singularly or by aggregating them, in order to assess the performance of the algorithm when having a different level of customers aggregation. In fact, load profiles become more and more random when moving towards the single-customer level and, as a consequence, clear patterns are more difficult to identify making the load prediction extremely challenging in such a scenario.

As presented in Section 2, given that the load prediction service is a standard time series analysis problem, a LSTM approach has been utilized to implement the LP service. Several models, i.e., hyperparameters, have been tested during the tests. The results presented in the following have been obtained using a set-up that proved to be sufficiently reliable in all the conditions, which consists of a sequential model with four hidden layers, where each hidden layer has 1 ~ 100 LSTM units. The LSTM model also has two dropout layers and one dense layer. Finally, a linear activation function has been considered.

For performance evaluation, the root mean squared error (RMSE) metric has been utilized. For an n number of tested samples, the MSE can be expressed as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}$$

where \hat{Y}_i is the i^{th} (normalized) predicted power, Y_i is the i^{th} (normalized) actual value, while $i = 1, \dots, n$ are the different time steps considered for the day (in the performed tests, 48 time steps are considered for each day, since the input dataset is given with a 30-min time resolution).

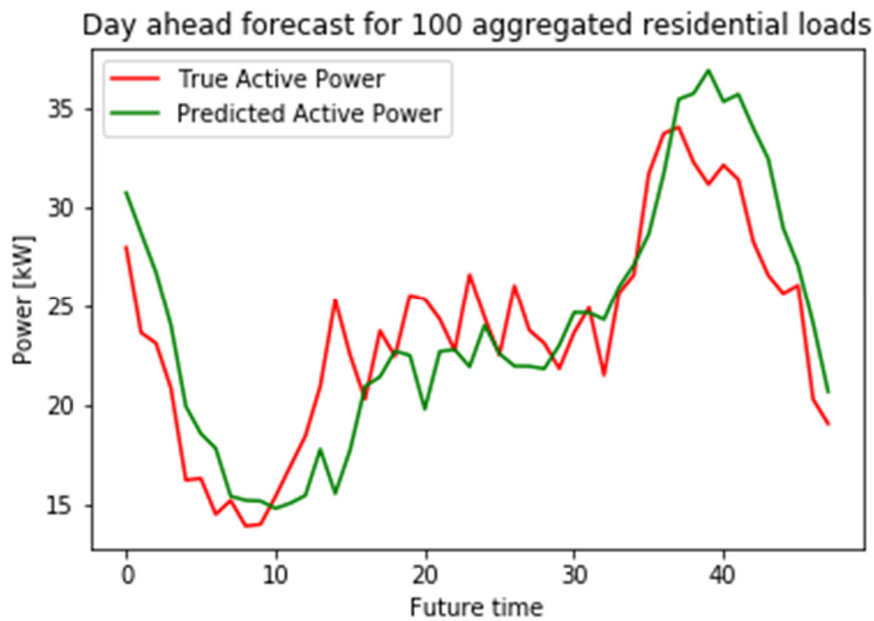


Figure 8: Actual vs day-ahead forecast of active power for 100 residential households, with 100 days in the training set.

Figure 8 presents the actual vs predicted active power in case of day-ahead forecasting when the total load associated to an aggregation of 100 households is considered. Such a forecast has been obtained by training the LSTM model with 100 days of historical data. It is possible to observe that the prediction closely follows the actual pattern of the daily consumption profile: in particular, both the valley during the night and the power peak during the evening are properly tracked, proving that the implemented model is able to correctly identify the existing recurring patterns. Figure 9 shows an analogous test performed on the load resulting from the aggregation of 100 end-users, but this time carried-out using only one month of data (30 days) for training the LSTM model. Also in this case, it is clear that the day-ahead forecast follows in a reliable way the actual profile of power consumption over the day. Such a test suggests that the proposed model can provide satisfying results also if not too many historical data are available in the storage system of the DSO. Therefore, the implementation of the proposed LP service can start being effective already short time after the deployment of the measurement infrastructure.

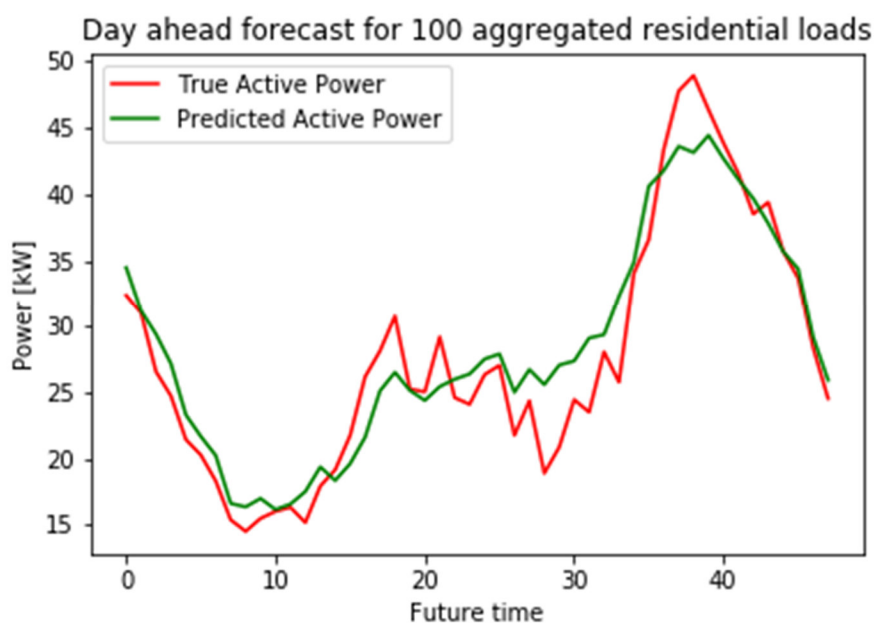


Figure 9: Actual vs day-ahead forecast of active power for 100 residential households, with 30 days in the training set.

Figure 10 presents the day-ahead forecast results for a scenario where the overall load is given by the aggregation of 25 residential customers. The comparison to the actual profile shows that the prediction in this particular case slightly overestimate the power consumption of the aggregated load. Nevertheless, the particular pattern of energy consumption has been again identified and the algorithm gives quite reliable results. Similarly, Figure 11 proves that the LP algorithm can still work well also if the aggregated consumption of only 5 residential customers is monitored. Finally, Figure 12 shows the most extreme case scenario, with the day-ahead forecasting of the energy consumption of a single customer. In this case, it is clear that the highly random behavior of the final customer can affect the performance of the algorithm and in fact the algorithm is not able to track the peak of power consumption. However, even in this challenging scenario, the model is still able to provide a quite accurate estimation of the base power consumed by the customer. The overall results with a varying number of customer forming the overall load consumption, which have been up to now qualitatively evaluated through the power profiles in the figures, can also be evaluated quantitatively through the RMSE index presented in Table 2. Here, it is possible to notice that the RMSE of the normalized prediction errors clearly show that the performance improves for increasing levels of load aggregation. Overall, therefore, the designed LP algorithm is expected to work reliably when implemented in MV grids, where the aggregation of many LV customers is usually present at the secondary substation. At the same time, however, results show that forecasting can be also done down to the LV grid and for small group of end-users, provided that some pattern exists for the aggregated power consumption profile.

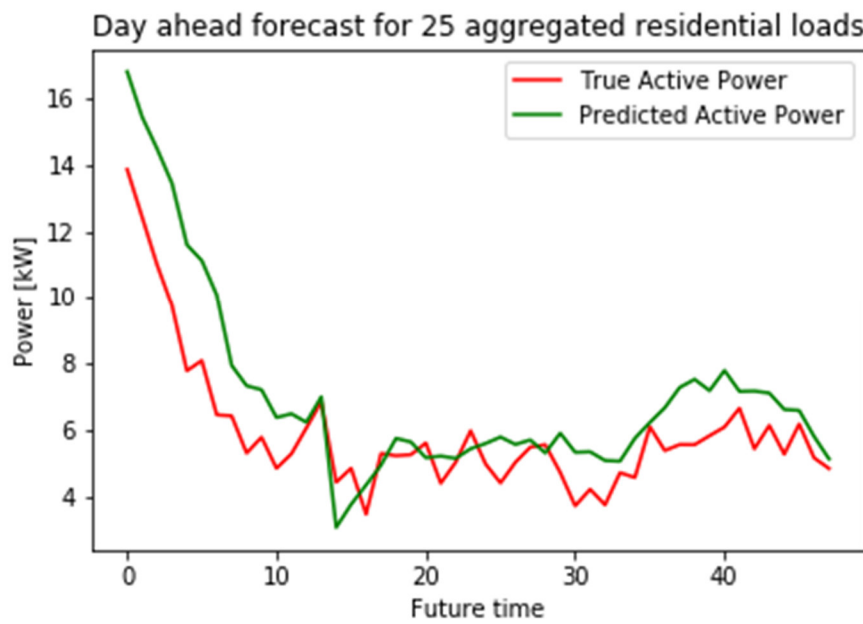


Figure 10: Actual vs day-ahead forecast of active power for 25 residential households, with 100 days in the training set.

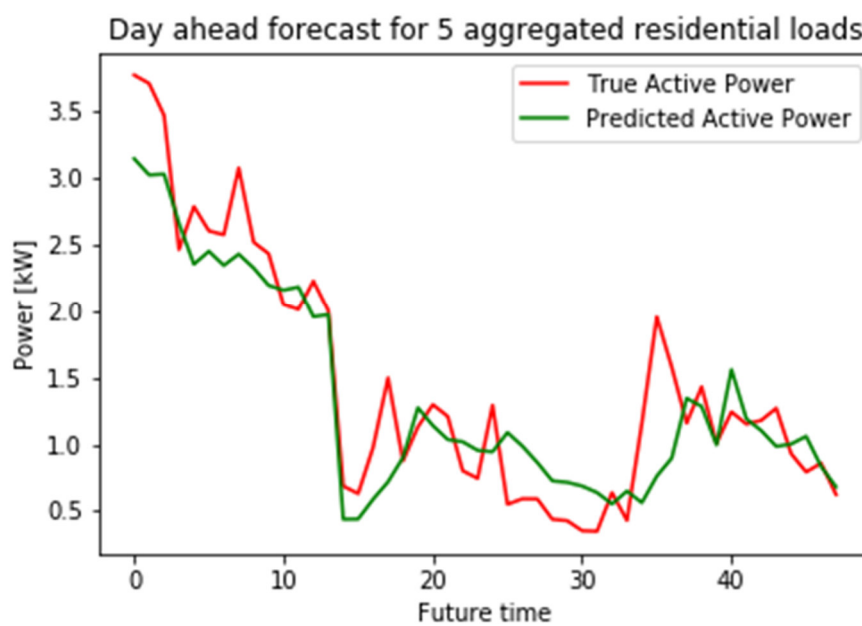


Figure 11: Actual vs day-ahead forecast of active power for 5 residential households, with 100 days in the training set.

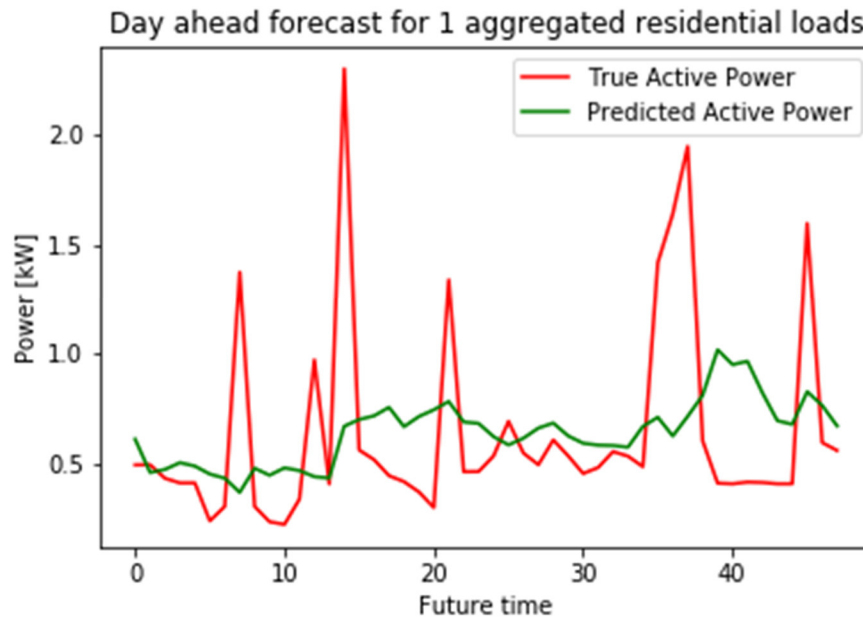


Figure 12: Actual vs day-ahead forecast of active power for a single residential household, with 100 days in the training set.

Table 2: Prediction performance for the LP service.

Scenario	RMSE
Aggregated profile of 100 customers	0.14
Aggregated profile of 50 customers	0.17
Aggregated profile of 25 customers	0.23
Aggregated profile of 10 customers	0.30
Aggregated profile of 5 customers	0.37
Single customer profile	0.64

3.2 Power generation prediction service

For the GP service, since the problem at hand is not expressed in terms of time series analysis, both FFDNN and LSTM prediction models have been tested. For both the models, several hyperparameters have been also tested to come up with the most appropriate settings for the dataset in hand.

The FFDNN has been designed for the four weather profiles with two hidden layers, with each one having between 100 and 250 neurons. A dropout layer is also included in the model. Furthermore, “relu” is the utilized activation function. Finally, the batch size is 100 ~ 200, and number of considered epochs is 40 ~ 100.

On the other hand, the LSTM for the four profiles have been designed with 150 ~ 250 LSTM units. Similar to the DNN, the relu activation function is utilized. The considered batch size is in this case 50 ~ 400, and the number of epochs is 30 ~ 80.

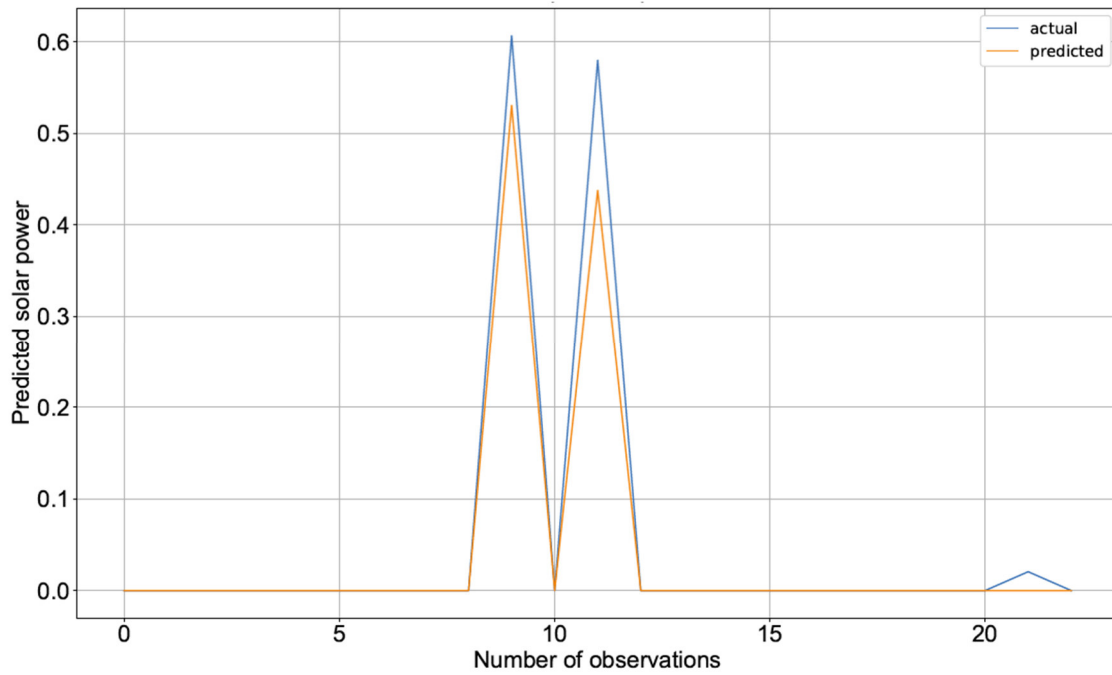


Figure 13: The actual vs the predicted generated power using DNN for weather profile 4.

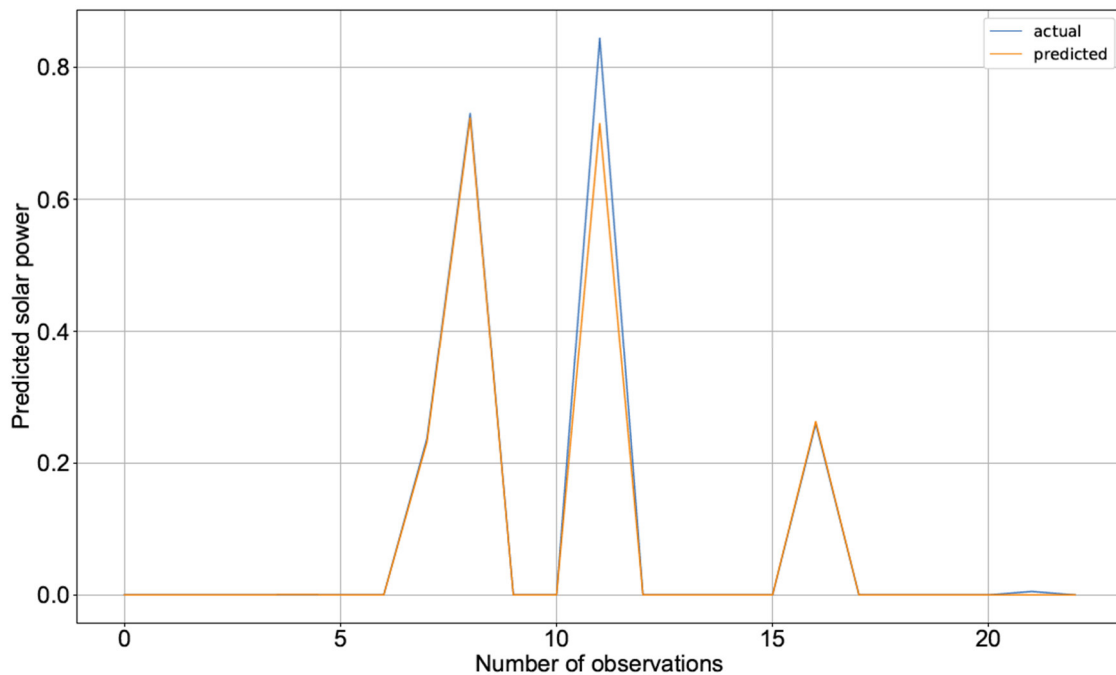


Figure 14: The actual vs the predicted generated power using LSTM for weather profile1.

Figure 13 and Figure 14 present a sample of DNN and LSTM performance, respectively. A summary of the performance for all profiles is shown in Table 3 and Table 4. Coherently with what described in Section 2, input data used for the model training and for the following prediction include, in addition to the time information, data about different irradiance level, about the temperature and humidity. With this set of input data, the GP is able to accurately track the generated power also in presence of intermittent weather conditions, as it can happen during cloudy days. However, it is worth noting that often the information on the irradiance condition can be unavailable, or in any case it will be given with a certain degree of uncertainty. This of course

can have a clear impact on the performance of the GP service, which can degrade coherently with the goodness of the irradiance information. In case of lack of irradiance measurements (or predictions), other weather and environmental inputs can be used, as shown in Table 1. However, the impact of these parameters would be much lower, since their correlation and consistency with respect to the generated power is relatively lower. As a consequence, again the GP performance could be affected and could degrade due to this.

Table 3: The prediction performance (measured in MSE) for the power generation prediction service using DNN.

Generation scenario	MSE
Profile 1	0.008281
Profile 2	0.013618
Profile 3	0.002284
Profile 4	0.005285

Table 4: The prediction performance (measured in MSE) for the power generation prediction service using the LSTM network.

Generation scenario	MSE
Profile 1	0.011088
Profile 2	0.012701
Profile 3	0.005241
Profile 4	0.015725

3.3 Fault location isolation and service restoration

Performed tests aim at verifying the possibility to use machine learning concepts for the design of the fault location module of the FLISR service and at evaluating the accuracy and goodness of the results achievable via data-driven approaches.

To this purpose, tests have been performed on the sample IEEE 13-bus grid, which is a benchmark grid often used for fault simulations and fault-related test cases [14]. While being a relatively small system, this grid allows having the proof of concept validation of the machine learning-based fault location routine and to evaluate its capability of properly identifying the faults occurring at different branches or nodes. In the implemented simulation set-up, the grid has been modelled following the state-of-the-art approaches for the emulation of the fault conditions. In particular, the same assumptions considered in the reference documentation have been done regarding the modelling of the equivalent system impedance, of the distribution transformer and of the faulted branch/node, in order to reproduce the same short circuit results provided in [14]. Furthermore, the overall grid has been modelled using the three-phase distribution system models for unbalanced grids, in order to be able to emulate any kind of fault. Figure 16 shows a simple schematic, extracted from [14], to show the main components taken into account for the overall distribution grid model.

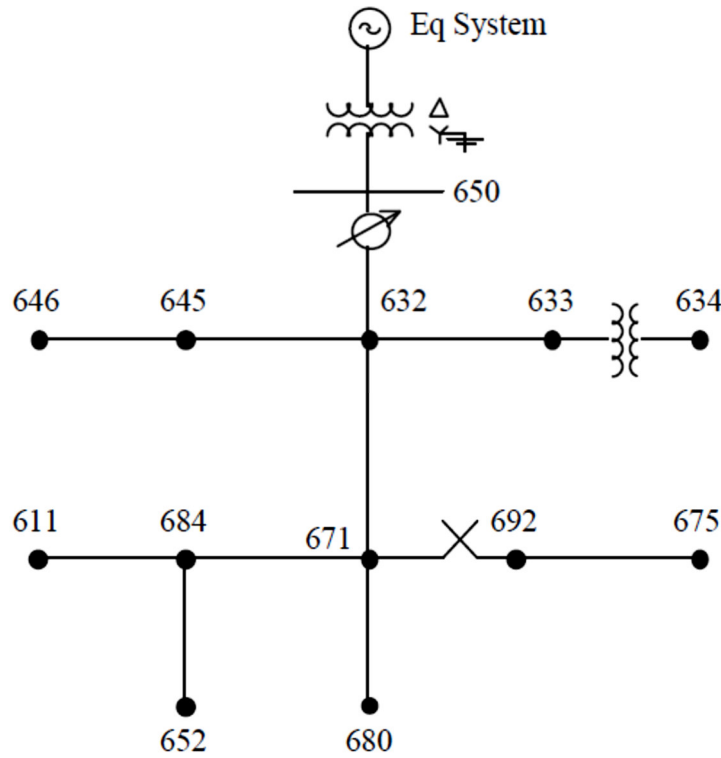


Figure 15: IEEE 13-bus test grid [14].

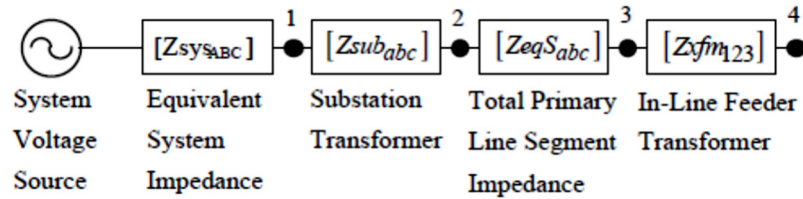


Figure 16: components models for the fault scenarios simulation [14].

Given the presented simulation set-up, the different types of single-phase to ground faults and three-phase to ground faults have been simulated in order to create the datasets for the training of the fault location algorithm. In particular, for each possible faulty branch, and every type of fault, 5000 Monte Carlo simulations have been performed. This means that for each type of fault, and each branch in which was possible to test that specific type of fault, a set of 5000 measurements has been created for the training of the machine learning algorithm. The set of 5000 tests related to the same branch have been carried out to randomly extract the position of the fault along the branch. Considering that the grid also has single-phase laterals, a total of 160000 tests has been generated in total, distributed among the different fault categories in the following way:

- 30000 for three-phase-to-ground faults (6 possible faulty branches);
- 40000 for phase A-to-ground faults (8 possible faulty branches);
- 40000 for phase B-to-ground faults (8 possible faulty branches);
- 50000 for phase C-to-ground faults (10 possible faulty branches).

For each one of the created fault scenarios, all the measurements of voltage and current in the “during fault” conditions have been saved, in order to emulate the possible presence of different measurement configurations in the grid, with a different penetration of measurement devices. The

datasets created in this way for each fault category have been then used for the training of the machine learning model.

After the machine learning algorithm training, to evaluate the performance of the developed fault location algorithm, the F_1 score has been used as key performance indicator. Such an index is a measure that gives the accuracy of a classification method in a range from 0 to 1, where F_1 score=0 represents a 0% accuracy, and F_1 score= 1 represents a 100% accuracy. More specifically, the performance of the system shown in Figure 7 has been finally tested. The dataset has two labels to train the two detectors in Figure 7: fault type and fault location labels. The fault type is used to train the fault type detector, which is considered as a first stage. As previously mentioned, the considered fault types belong to one of the following three categories: AG, BG, CG, and ABCG, which indicates single line phase A to ground, single line phase B to ground, single line phase C to ground, and three-phase to ground, respectively. The fault location or, in other words, the faulty line or segment in the grid, is used as a label to train the fault location detector. At the end, the final detection output indicates the fault type, and the fault location (segment or line). The following strings are examples of the output given by the overall fault location algorithm:

'AG:8_10' 'AG:8_9' 'AG:8_9' ... 'ABCG:4_5' 'ABCG:3_8' 'ABCG:3_4',

where the starting label indicates the type of detected fault, while the x_y label indicates the terminal nodes of the specific branch of the grid identified as possibly faulty. Tests performed with measurement devices located in nodes 632 and 671 and 684 (in all the branching nodes) gave a corresponding F_1 score= 0.9999. Possible degradation of the results could instead occur when only few measurements are available, due to the possibility of having similar fault transient conditions for faults occurring in different branches, but with a similar overall short circuit impedance. This issue is known in the literature (often referred to as multiple fault location problem) and it can be overcome with a proper placement of the measurement instrumentation.

For the purposes of the FLISR implementation, however, it is worth highlighting that the main requirement is only the identification of the area where the fault is, since this is the information needed for the automatic coordination of the isolation and service restoration steps. The precise identification of the branch can be a plus, since it would allow the DSO to send the crew directly at the location where the fault occurred (saving time for the fault search and reducing in this way the outage time for the faulty zone), but a rough estimation is in general still a good compromise considering the very low penetration of measurements in the distribution grid and the simultaneous very large size (from a topological perspective, in terms of sections and nodes) of these networks.

4. Documentation of the application programming interface

These interfaces describe the interaction of the various services implemented within the scope of the EU project SOGNO. The fundamental architecture pattern is a combination of microservices interacting via an MQTT broker. This pattern is used in information technology as it enables complex software instances to communicate with each other via defined endpoints independently of their implementation.

The services are mostly decoupled from each other and process a clearly defined task. This pattern enables a modular structure of the entire system, so that a failure of one or more services does not necessarily mean a failure of the entire system. In addition, new or existing services can be directly injected, deleted or updated without any significant impact on the rest of the system.

Figure 17 describes how the publish/subscribe mechanism between the services should be understood. As an example, let us assume that Service B is dependent on the input from Service A. Service A will publish its results on a specific predefined topic. The broker will forward the results from Service A to Service B, because Service B has subscribed to the topic before. The direction of the red components should be read as follows: Service A offers an interface that publishes its results in JSON format on Topic X. This JSON message is subscribed to by the B service. Once Service B has received these results, it can proceed with the data as its implementation requires.

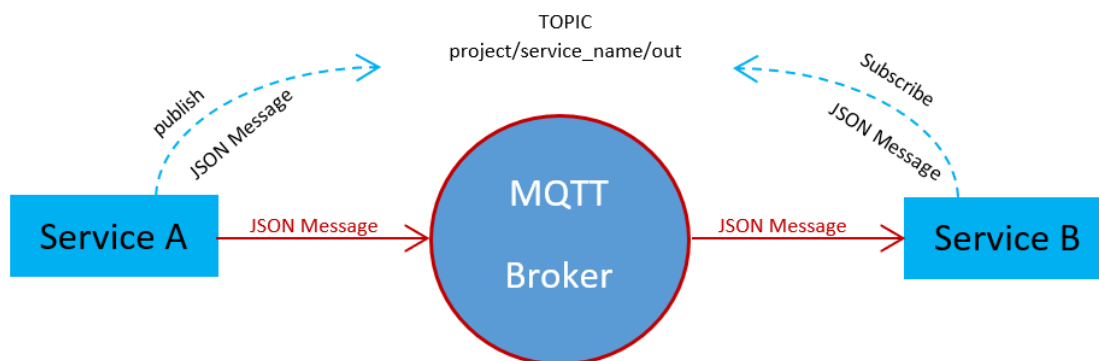


Figure 17: Publish/subscribe mechanism.

There are services that publish different results on different topics. The topic naming follows a certain logic. This consists of the project name followed by the service name and then the direction separated by a slash, which separates the topic levels. If necessary, the results types will be appended to the end of the name as shown in the Table 5.

Table 5: Logic of topic naming.

LOGIC	EXAMPLE
PROJECT_NAME / SERVICE / DIRECTION [/ RESULT_TYP]	sogno / se / out
	sogno / gui / out / flisr
	sogno / gui / out / topology

The usage of topic level separators allows to work with placeholders. While the character “+” stands for any content of a single layer, the character “#” can be used to select any content in layers afterwards. These are provided by the MQTT specification. For example, a service can be notified to all outgoing messages from any service with this simple topic name:

sogno / + / out / #

Figure 18 shows an overall view of the interaction of the services developed in this project from a logical point of view and illustrates the dependencies between the different services.

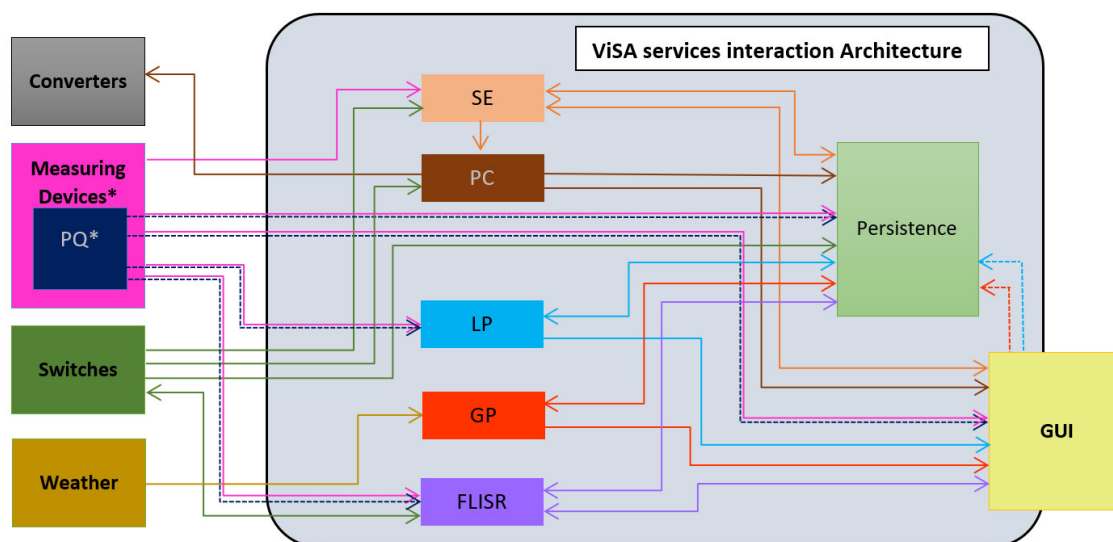


Figure 18: ViSA service interaction architecture.

* The output of the power quality service is included in the message of the measuring devices.

4.1 The Interfaces

This section describes the endpoints for the corresponding services in this deliverable, namely FLISR, LP and GP. The remaining services are part of other working packages and will be presented in the deliverables D2.3 and D4.3.

Basically, each service in this project will have two interfaces that will register a certain number of topics for publishing or subscribing. A complete list of the interfaces contained in the corresponding components were part of deliverable D2.2 and are described there individually.

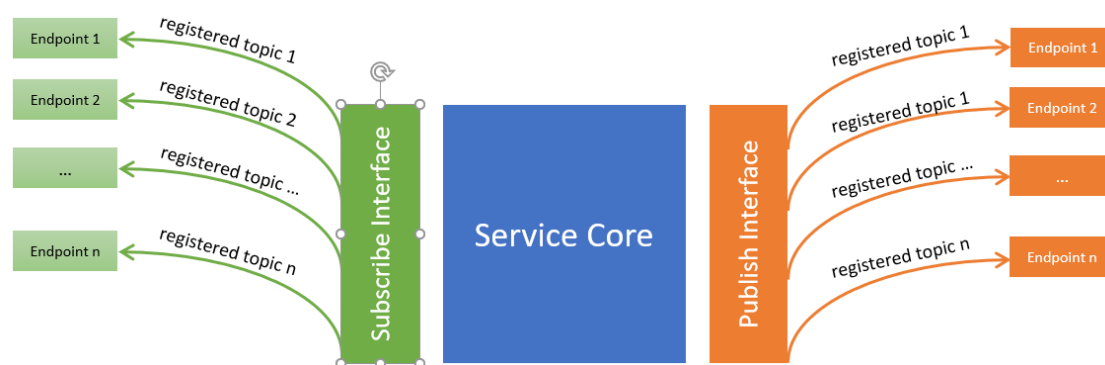


Figure 19: A service with its publish and subscribe endpoints

4.1.1 Measuring Devices with the input from Power Quality Service

Topic	Mechanism	Description	Payload
sogno/md/out*	publish	The endpoint on which the APMUs and the PQ service publish their measurements	In the annex

* The measuring device message contains the results of the power quality service.

4.1.2 Switches

Topic	Mechanism	Description	Payload
sogno/switch/out	publish	The endpoint on which the switches publish their operational states.	In the annex
sogno/flisr/out	subscribe	The endpoint subscribed by the switches to receive the commands from the FLISR service	In the annex

4.1.3 Fault Location, Isolation, and Service Restoration

Topic	Mechanism	Description	Payload
sogno/flisr/out	publish	The endpoint on which the FLISR service publishes its results	In the annex
sogno/md/out	subscribe	The endpoint subscribed by the FLISR service to receive the measurement data from the APMU devices	In the annex
sogno/switch/out	subscribe	The endpoint subscribed by the FLISR service to receive the operational states of the switches	In the annex
sogno/db/out/topology	subscribe	The endpoint subscribed by the FLISR service to get the topology as CIM file from the persistence service when the system is restarted, or new topologies have been imported in the GUI	In the annex
sogno/gui/out/flisr	subscribe	The endpoint subscribed by the FLISR service to enable or disable the functionality of FLISR for one or more grids	In the annex

Since the CIM files can be very large, they will be sent compressed by the publisher using the python “zlib” library. The subscriber will decompress the message and then process it further. A small piece of python code for reading and compressing the CIM file could look like this:

```
import zlib

compressed_cim_file = None

with open('path/to/cim/cim.xml', 'rb') as cim_file:
    cim_file = cim_file.read()
    compressed_cim_file = zlib.compress(cim_file, zlib.Z_BEST_COMPRESSION)
```

The subscriber receives the compressed message as the payload of a message. It is assumed that the payload was stored in a variable called `compressed_cim_file`. The following code can be used to decompress the CIM file.

```
import zlib

decompressed_cim_file = zlib.decompress(compressed_cim_file)
```

4.1.4 Load Prediction

Topic	Mechanism	Description	Payload
sogno/lp/out	publish	The endpoint on which the load prediction service publishes its results	In the annex
sogno/md/out	subscribe	The endpoint subscribed by load prediction service to receive the measurement data	In the annex
sogno/db/out/lp	subscribe	The endpoint subscribed by load prediction service to receive the measurement data for a specific time period for one or more nodes or lines	In the annex
sogno/gui/out/lp	subscribe	The endpoint subscribed by load prediction to receive a list of selected nodes or lines for service processing	In the annex

4.1.5 Generation Prediction

Topic	Mechanism	Description	Payload
sogno/gp/out	publish	The endpoint on which the generation prediction service publishes its results	In the annex
sogno/weather/out	subscribe	The endpoint subscribed by generation prediction service to receive the weather data	In the annex
sogno/db/out/gp	subscribe	The endpoint subscribed by load prediction service to receive the measurement data for a specific time period for one or more nodes or lines	In the annex
sogno/gui/out/gp	subscribe	The endpoint subscribed by generation prediction to receive a list of selected nodes or lines for service processing	In the annex

5. Conclusion

The focus behind work package WP3 has been to design services responsible of achieving autonomous and self-healing power systems, using innovative data driven and machine learning approaches. This deliverable, in combination with the already presented D3.1, provided an overview of the concepts and services responsible of achieving this goal. A detailed description of the implementation and design of these services, namely, the fault location isolation and service restoration (FLISR), as well as the load prediction (LP) and generation prediction (GP) services has been given, together with the presentation of the underlying concepts related to machine learning considered for their design. Machine learning data driven techniques, and deep neural networks are the main methods utilized to design these services. All the services have been tested to prove the correct design and operation of the algorithms and to validate them at proof of concept level. Openly available datasets or simple benchmark test grids have been used at this stage, since the goal was mainly to prove the validity and correctness of the implemented approaches and to assess possible strengths and weaknesses in relatively small test cases. Further work to prove the performance of the designed services, both in laboratory environment and in the field trials, will be presented at the end of the project through the Deliverables related to the trial implementation and results.

6. List of Tables

Table 1: Raw data description..... 11

Table 2: Prediction performance for the LP service. 21

Table 3: The prediction performance (measured in MSE) for the power generation prediction service using DNN..... 23

Table 4: The prediction performance (measured in MSE) for the power generation prediction service using the LSTM network. 23

Table 5: Logic of topic naming. 26

7. List of Figures

Figure 1: An overview of SOGNO activities.	6
Figure 2: A sample of input data used for load prediction service.	8
Figure 3: The block diagram of the load prediction service.	9
Figure 4: The structure of the raw dataset to be utilized in the power generation prediction service.	11
Figure 5: The MI score for the raw data.	13
Figure 6: The block diagram of the generation prediction service.	14
Figure 7: The block diagram of fault location detection within FLISR service.	15
Figure 8: Actual vs day-ahead forecast of active power for 100 residential households, with 100 days in the training set.	18
Figure 9: Actual vs day-ahead forecast of active power for 100 residential households, with 30 days in the training set.	19
Figure 10: Actual vs day-ahead forecast of active power for 25 residential households, with 100 days in the training set.	20
Figure 11: Actual vs day-ahead forecast of active power for 5 residential households, with 100 days in the training set.	20
Figure 12: Actual vs day-ahead forecast of active power for a single residential household, with 100 days in the training set.	21
Figure 13: The actual vs the predicted generated power using DNN for weather profile 4.	22
Figure 14: The actual vs the predicted generated power using LSTM for weather profile1.	22
Figure 15: IEEE 13-bus test grid [14].	24
Figure 16: components models for the fault scenarios simulation [14].	24
Figure 17: Publish/subscribe mechanism.	26
Figure 18: ViSA service interaction architecture.	27
Figure 19: A service with its publish and subscribe endpoints.....	27

8. References

- [1] G. Herbrich, „UCI Repository,“ [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption>.
- [2] J. Zheng, C. Xu, Z. Zhang und X. Li, „Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network,“ in *51st Annual Conference on Information Sciences and Systems (CISS)*, 2017.
- [3] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli und R. J. Hyndman, „Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond,“ *International Journal of Forecasting*, pp. 896-913, 2016.
- [4] R. Alhalaseh, H. A. Tokel, S. Chakraborty, G. Alirezai und R. Mathar, „Feature-Selection based PMU Placement for Detection of Faults in Power Grids,“ in *International Telecommunication Networks and Applications Conference (ITNAC)*, Australia, 2018.
- [5] A. Kraskov, H. Stoeckbauer und P. Grassberger, „Estimating mutual information,“ *Physical review. E, Statistical, nonlinear, and soft matter physics*, 2004.
- [6] K. P. Murphy, *Machine learning: a probabilistic perspective*, The MIT Press, 2012.
- [7] I. Goodfellow, Y. Bengio und A. Courville, *Deep learning*, The MIT Press, 2016.
- [8] „Cohah's Blog,“ 27. 8. 2015. [Online]. Available: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- [9] L. Breiman, „Random forests,“ *Machine learning, Volume 45*, pp. 5-32, 10 2011.
- [10] S. S. Baboo und E. Iyyapparaj, „A classification and analysis of pulmonary nodules in CT images using random forest,“ in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, 2018.
- [11] A. Liaw und M. Wiener, „Classification and regression by random forest,“ *R News*, pp. 18-22, 2002.
- [12] T. Hastie, R. Tibshirani und J. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*, Springer Series in Statistics, 2001.
- [13] SmartMeter EnergyConsumption Data in London Households, available online at: <https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households>
- [14] W. H. Kersting, G. Shirek, "Short circuit analysis of IEEE Test Feeders," *PES T&D 2012*, Orlando, FL, 2012, pp. 1-9

9. List of Abbreviations

WP3	Work package 3 in SOGNO project
FLISR	Fault location isolation and service restoration
LP	Load prediction
GP	Generation prediction
PV	Photovoltaic
LSTM	Long short-term memory
RNN	Recurrent neural network
DT	Decision tree classifier
RF	Random forest classifier
DNN	Deep neural network
FFDNN	Feed-forward DNN

ANNEX 1 Examples of the JSON messages

A.1.1 Example for the JSON Message from the measuring devices with power quality service

```
{
  "identifier": "863921030120813-1",
  "readings": [
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_1",
      "measurand": "activepower",
      "data": "-0.010"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_2",
      "measurand": "activepower",
      "data": "0.001"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_3",
      "measurand": "activepower",
      "data": "-0.001"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_1",
      "measurand": "apparentpower",
      "data": "-0.010"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_2",
      "measurand": "apparentpower",
      "data": "0.000"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_3",
      "measurand": "apparentpower",
      "data": "0.037"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_1",
      "measurand": "current",
      "data": "0.000"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_2",
      "measurand": "current",
      "data": "0.000"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_3",
      "measurand": "current",
      "data": "0.095"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_N",
      "measurand": "current",
      "data": "0.308"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_1",
      "measurand": "frequency",
      "data": "50.010"
    }
  ]
}
```

```

    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_2",
      "measurand": "frequency",
      "data": "50.000"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_3",
      "measurand": "frequency",
      "data": "50.010"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_1",
      "measurand": "powerfactor",
      "data": "-1.000"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_2",
      "measurand": "powerfactor",
      "data": "-1.000"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_3",
      "measurand": "powerfactor",
      "data": "0.053"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_1",
      "measurand": "reactivepower",
      "data": "0.000"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_2",
      "measurand": "reactivepower",
      "data": "0.00"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_3",
      "measurand": "reactivepower",
      "data": "0.037"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_1",
      "measurand": "volt",
      "data": "228.720"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_2",
      "measurand": "volt",
      "data": "229.719"
    },
    {
      "timestamp": "2019-04-08T08:44:01",
      "source": "chan_3",
      "measurand": "volt",
      "data": "229.300"
    }
  ]
}

```

A.1.2 Example for the JSON message from the switches

```

{
  "version": "1.0",
  "identifier": "756294756273",
  "type": "switch_input",
  "timestamp": "2019-04-08T08:44:01",

```

```

    "phase": "a",
    "action": "closed",
    "location": [
      {
        "node1": "863921030120813-1",
        "node2": "863921030120746-1"
      }
    ]
  }
}

```

A.1.3 Example for the JSON message from FLISR

```

{
  "version": "1.0",
  "type": "flisr_result",
  "timestamp": "2019-04-08T08:44:01",
  "switches": [
    {
      "identifier": "756294784691",
      "cmd": "opening/closing"
    }
  ]
}

```

A.1.4 Example for a CIM file

```

<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cim="http://iec.ch/TC57/2012/CIM-schema-cim16#"
  xmlns:md="http://iec.ch/TC57/61970-552/ModelDescription/1#"
  xmlns:entsoe="http://entsoe.eu/Secretariat/ProfileExtension/2#"
  xmlns:neplan="http://www.neplan.ch#">
  <md:FullModel rdf:about="urn:uuid:f2ff6631-14b2-42d2-8fdb-a2edc910aefe">
    <md:Model.created>2018-11-06T16:34:52</md:Model.created>
    <neplan:Model.createdBy>NEPLAN by Busarello + Cott + Partner AG,
    Switzerland</neplan:Model.createdBy>
    <md:Model.scenarioTime>2018-11-06T16:34:50</md:Model.scenarioTime>
    <md:Model.version>7</md:Model.version>
    <md:Model.DependentOn rdf:resource="urn:uuid:25276efc-54c7-450a-87d6-6f58ae3470a9"/>
    <md:Model.description>NEPLAN</md:Model.description>
    <md:Model.modelingAuthoritySet>FULL</md:Model.modelingAuthoritySet>
    <md:Model.profile>http://iec.ch/TC57/61970-456/Topology/3</md:Model.profile>
  </md:FullModel>
  <cim:NameType rdf:ID="_180dea0a-f4a7-4fec-90f4-fbb189409882">
    <cim:NameType.name>description</cim:NameType.name>
  </cim:NameType>
  <cim:TopologicalNode rdf:ID="N0">
    <cim:IdentifiedObject.name>N0</cim:IdentifiedObject.name>
    <cim:TopologicalNode.BaseVoltage
      rdf:resource="#_fd4aebc6-81ee-42dd-bea2-e3368f769175"/>
    <cim:TopologicalNode.ConnectivityNodeContainer
      rdf:resource="#_53a66d62-e48c-4548-bdb0-8d46f596c259"/>
  </cim:TopologicalNode>

  <cim:Terminal rdf:about="#E-132">
    <cim:Terminal.connected>true</cim:Terminal.connected>
    <cim:Terminal.TopologicalNode rdf:resource="#N4"/>
  </cim:Terminal>

  <cim:NameType rdf:ID="_3e875c98-dd64-4d7d-af2b-be41ca499dd6">
    <cim:NameType.name>description</cim:NameType.name>
  </cim:NameType>
  <cim:BaseVoltage rdf:ID="_0a92d9bb-71b6-4256-aa94-54fde8700496">
    <cim:BaseVoltage.nominalVoltage>20.00</cim:BaseVoltage.nominalVoltage>
  </cim:BaseVoltage>

  <cim:VoltageLevel rdf:ID="_c9026ac7-114f-42db-ad42-c62d5f6dd04e">
    <cim:IdentifiedObject.name>98</cim:IdentifiedObject.name>
    <cim:VoltageLevel.Substation rdf:resource="#_3FA2-2E-F1-00000000000000003938"/>
    <cim:VoltageLevel.BaseVoltage rdf:resource="#_0a92d9bb-71b6-4256-aa94-54fde8700496"/>
  </cim:VoltageLevel>

  <cim:Substation rdf:ID="_3FA2-2E-F1-0000000000000000323130">
    <cim:IdentifiedObject.name>N2</cim:IdentifiedObject.name>

```

```

    <cim:Substation.Region rdf:resource="#_f86cebef-3021-4a41-895e-b0d480beee45"/>
  </cim:Substation>

  <cim:GeographicalRegion rdf:ID="_452d9bf4-88a4-497a-93ce-cf500fc681e5">
    <cim:IdentifiedObject.name>Area 1</cim:IdentifiedObject.name>
  </cim:GeographicalRegion>

  <cim:SubGeographicalRegion rdf:ID="_f86cebef-3021-4a41-895e-b0d480beee45">
    <cim:IdentifiedObject.name>Zone 1</cim:IdentifiedObject.name>
    <cim:SubGeographicalRegion.Region rdf:resource="#_452d9bf4-88a4-497a-93ce-cf500fc681e5"/>
  </cim:SubGeographicalRegion>

  <cim:OperationalLimitType rdf:ID="_bcae0a62-8109-4523-b661-4bc89de7c05c">
    <cim:IdentifiedObject.name>PATL</cim:IdentifiedObject.name>

  <cim:OperationalLimitType.acceptableDuration>45000</cim:OperationalLimitType.acceptableDuration>

    <cim:OperationalLimitType.direction rdf:resource="http://iec.ch/TC57/2012/CIM-schema-cim16#OperationalLimitDirectionKind.absoluteValue"/>
  </cim:OperationalLimitType>

  <cim:EnergyConsumer rdf:ID="LOAD-I-7">
    <cim:IdentifiedObject.name>LOAD-I-7</cim:IdentifiedObject.name>
    <cim:EnergyConsumer.p>0.076500</cim:EnergyConsumer.p>
    <cim:EnergyConsumer.q>0.047410</cim:EnergyConsumer.q>
    <cim:Equipment.aggregate>false</cim:Equipment.aggregate>
    <cim:Equipment.EquipmentContainer rdf:resource="#_bad2e7c9-5aa8-4017-af2c-5ef5dcf65cf9"/>
  </cim:EnergyConsumer>

  <cim:Terminal rdf:ID="E-337">
    <cim:IdentifiedObject.name>LOAD-I-7_0</cim:IdentifiedObject.name>
    <cim:Terminal.sequenceNumber>1</cim:Terminal.sequenceNumber>
    <cim:Terminal.ConductingEquipment rdf:resource="#LOAD-I-7"/>
  </cim:Terminal>

  <cim:SvVoltage rdf:ID="N0_5B20">
    <cim:SvVoltage.v>110.000000</cim:SvVoltage.v>
    <cim:SvVoltage.angle>-0e+000</cim:SvVoltage.angle>
    <cim:SvVoltage.TopologicalNode rdf:resource="#N0"/>
  </cim:SvVoltage>

  <cim:SvPowerFlow rdf:ID="LOAD-I-7-sv">
    <cim:SvPowerFlow.p>0.076500</cim:SvPowerFlow.p>
    <cim:SvPowerFlow.q>0.047410</cim:SvPowerFlow.q>
    <cim:SvPowerFlow.Terminal rdf:resource="#E-337"/>
  </cim:SvPowerFlow>

  <cim:Diagram rdf:ID="_9a844459-cf44-48c4-b5ae-e56d7446f017">
    <cim:IdentifiedObject.name>Diagram 0</cim:IdentifiedObject.name>
    <cim:Diagram.orientation rdf:resource="http://iec.ch/TC57/2010/CIM-schema-cim16#OrientationKind.negative"/>
  </cim:Diagram>

  <cim:DiagramObject rdf:ID="_450e8d17-a49d-476b-ae6e-ef263e5e02b5">
    <cim:DiagramObject.rotation>0</cim:DiagramObject.rotation>
    <cim:DiagramObject.IdentifiedObject rdf:resource="#E-1229744368"/>
    <cim:DiagramObject.Diagram rdf:resource="#_9a844459-cf44-48c4-b5ae-e56d7446f017"/>
  </cim:DiagramObject>

  <cim:DiagramObjectPoint rdf:ID="_b5988043-8ea2-42e5-a9af-0c5400027158">
    <cim:DiagramObjectPoint.sequenceNumber>1</cim:DiagramObjectPoint.sequenceNumber>
    <cim:DiagramObjectPoint.DiagramObject rdf:resource="#_450e8d17-a49d-476b-ae6e-ef263e5e02b5"/>
    <cim:DiagramObjectPoint.xPosition>120.000000</cim:DiagramObjectPoint.xPosition>
    <cim:DiagramObjectPoint.yPosition>80.000000</cim:DiagramObjectPoint.yPosition>
  </cim:DiagramObjectPoint>
</rdf:RDF>

```

A.1.5 Example for JSON file to enable or disable FLISR

```

{
  "version": "1.0",
  "type": "gui_flisr",
  "timestamp": "2019-04-08T08:44:01",
  "user": "xyz",
  "cmd": "enable/disable"
}

```

```
}
```

A.1.6 Example for the JSON output message from the LP service

```
{
  "version": "1.0",
  "type": "lp_out",
  "nodes": [
    {
      "node_id": "123456",
      "values": [
        {
          "timestamp": "2019-04-08T08:44:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 123.3
        },
        {
          "timestamp": "2019-04-08T08:44:01",
          "phase": "a",
          "measurand": "reactivepower",
          "data": 123.3
        },
        {
          "timestamp": "2019-04-08T08:45:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 125.3
        },
        {
          "timestamp": "2019-04-08T08:45:01",
          "phase": "a",
          "measurand": "reactivepower",
          "data": 125.3
        }
      ]
    },
    {
      "node_id": "987654",
      "values": [
        {
          "timestamp": "2019-04-08T08:44:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 123.3
        },
        {
          "timestamp": "2019-04-08T08:44:01",
          "phase": "a",
          "measurand": "reactivepower",
          "data": 123.3
        },
        {
          "timestamp": "2019-04-08T08:45:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 125.3
        },
        {
          "timestamp": "2019-04-08T08:45:01",
          "phase": "a",
          "measurand": "reactivepower",
          "data": 125.3
        }
      ]
    }
  ]
}
```

A.1.7 Example for the JSON output message from the GUI to manage the LP service

```
{
  "version": "1.0",
  "type": "lp_request",
```

```

    "from": "2019-04-08T08:00:00",
    "to": "2019-04-09T08:00:00",
    "node_id": [
        "863921030120813-1",
        "863921030120746-1"
    ]
}

```

A.1.8 Example for the JSON output message from the DB with the LP necessary input

```

{
  "version": "1.0",
  "type": "db_lp",
  "nodes": [
    {
      "node_id": "123456",
      "values": [
        {
          "timestamp": "2019-04-08T08:44:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 123.3
        },
        {
          "timestamp": "2019-04-08T08:44:01",
          "phase": "a",
          "measurand": "reactivepower",
          "data": 123.3
        },
        {
          "timestamp": "2019-04-08T08:45:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 125.3
        },
        {
          "timestamp": "2019-04-08T08:45:01",
          "phase": "a",
          "measurand": "reactivepower",
          "data": 125.3
        }
      ]
    },
    {
      "node_id": "987654",
      "values": [
        {
          "timestamp": "2019-04-08T08:44:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 123.3
        },
        {
          "timestamp": "2019-04-08T08:44:01",
          "phase": "a",
          "measurand": "reactivepower",
          "data": 123.3
        },
        {
          "timestamp": "2019-04-08T08:45:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 125.3
        },
        {
          "timestamp": "2019-04-08T08:45:01",
          "phase": "a",
          "measurand": "reactivepower",
          "data": 125.3
        }
      ]
    }
  ]
}

```


A.1.9 Example for the JSON output message from the GUI to manage the GP service

```
{
  "version": "1.0",
  "type": "gp_request",
  "from": "2019-04-08T08:00:00",
  "to": "2019-04-09T08:00:00",
  "node_id": [
    "863921030120813-1",
    "863921030120746-1"
  ]
}
```

A.1.10 Example for the JSON output message from the DB with the GP necessary input

```
{
  "version": "1.0",
  "type": "db_gp",
  "nodes": [
    {
      "node_id": "123456",
      "values": [
        {
          "timestamp": "2019-04-08T08:44:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 123.3
        },
        {
          "timestamp": "2019-04-08T08:45:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 123.4
        },
        {
          "timestamp": "2019-04-08T08:46:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 123.5
        }
      ]
    },
    {
      "node_id": "987654",
      "values": [
        {
          "timestamp": "2019-04-08T08:44:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 123.3
        },
        {
          "timestamp": "2019-04-08T08:45:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 123.4
        },
        {
          "timestamp": "2019-04-08T08:46:01",
          "phase": "a",
          "measurand": "activepower",
          "data": 123.5
        }
      ]
    }
  ]
}
```