The eDREAM project is co-founded by the EU's Horizon 2020 innovation programme under grant agreement No 774478

enabling new Demand REsponse Advanced, Market oriented and secure technologies, solutions and business models

DELIVERABLE: D4.5 Specification for Improved Decision Making and DR Optimization toolset V2

Main author: Fathi Abugchem (TU)

Imprint

Specification for Improved Decision Making and DR Optimization toolset V2

Contractual Date of Delivery to the EC:	30.04.2020
Actual Date of Delivery to the EC:	29.04.2020
Author(s):	Fathi Abugchem (TU), Vladimir Vukovic(TU) Napoleon Bezas (CERTH), Paraskevas Koukaras (CERTH), Paschalis Gkaidatzis (CERTH), Giannis E. Gotsos (CERTH), Christina Tsita (CERTH), Dimosthenis Ioannidis (CERTH), Christos Tjortjis (CERTH) Giuseppe Raveduto (ENG)
Participant(s):	Leader: TU Contributing partners: CERTH, ENG
Project:	enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models (eDREAM)
Work package:	Wp4 – Next generation DR Services for Aggregators and Customers
Task:	4.1 – Specification for Improved Decision Making and DR Optimization toolset V2
Confidentiality:	public
Version:	1.0

Legal Disclaimer

The project enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models (eDREAM) has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 774478. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the Innovation and Networks Executive Agency (INEA) or the European Commission (EC). INEA or the EC are not responsible for any use that may be made of the information contained therein.

Copyright

Executive Summary

This document describes deliverable D4.5 related to Task 4.1 and entitled "Specification for Improved Decision-Making and DR Optimization toolset V2". The Success criteria as stated in the DOA is "the usage guidelines of the support system toolkit should be reported".

The report provides an overview of the work carried out to develop the Decision-Making and DR Optimization toolset models. The technical specifications, methods, and techniques used to implement the toolset are detailed.

With reference to the proposed general architecture of the eDREAM project, four main models were developed and implemented, namely the algorithm for trend analysis, the method for calculating the degradation rate of PV systems, the decision-making system, and the optimisation engine. The aim is to develop new solutions for DSO, as well as to improve the decision-making processes of the aggregators and energy retailers taking into account the participation of the virtual power plant (VPP). A VPP is a network of aggregated decentralised medium-scale power generating units traded like one single power plant. These units can be wind turbines, solar photovoltaic systems, and Combined Heat and Power units, as well as flexible power consumers and storage systems. Despite the great advantages of the VPP, the research findings show that the implementation of the VPP poses many challenges. In this deliverable some of these challenges are addressed.

As renewables penetration of the VPP increases, accurate prediction of power production of these systems becomes essential. In Section 2 of this deliverable, a near real-time trend analysis algorithm for short-term forecast of the PV production is proposed. A novel near real-time trending analysis tool based on a modified version of a Slope Statistic Profile method used in other domains has been modified to work for PV systems. The proposed method is capable to detect real-time incidents that occur in the PV production time series. The estimation of incident time point is based on the combination of their linear trend profiles test statistics, computed on a consecutive overlapping data window. In Section 3, a method to estimate the long-term reliability and durability of currently installed PV systems is proposed. Degradation rates must be known in order to predict long-term power delivery. A statistical method, Classical Seasonal Decomposition has been implemented. The method based on extracting the seasonality and trend from the time series which than used to estimate the degradation rate. Estimated degradation rate and deseasonalised data are used to forecast long-term production of the system.

Decision-making and optimisation engine models are described in Section 4. Management of the VPP requires special algorithms to create an optimal schedule of the power generation units. The algorithms should take into account the unpredictable variations of the power generated by renewable energy systems and prosumers power consumption. The idea is that, in order to trade as a single unit, the VPP should be able to ramp up and down generation of the controllable units to balance the fluctuations in the production of renewables and prosumers consumption participating on the VPP. This has been formulated as a customized unit commitment and economic load dispatch optimisation problem. To solve this problem, a method to estimate the upper and lower limits of the total VPP power generation considering PV production and prosumers consumption has been proposed. The method leverages on the output of the eDREAM Electrical Consumption/production Forecast, Flexibility Forecast and Baseline Estimation components. In order to optimally schedule the power generating units, an optimisation engine has been designed based on generic optimisation program GenOpt.

The optimisation engine is a customised GenOpt-based optimization framework designed to solve the formulated optimisation problem. GenOpt is an optimisation program that contains several algorithms which can be used to solve different problems. GenOpt allows customisation of optimisation algorithms and therefore it can be used as an optimisation algorithm development environment. However, guidance about how to select the optimisation algorithms and set their parameters is not integrated in a graphical user interface. Therefore, the user needs to pick and choose optimisation algorithm and also determine their relevance to the optimisation problem based on prior knowledge or external literature.

In this deliverable a novel intuitive user interface and model for the unit commitment and economic load dispatch problem compatible with GenOpt has been developed. The user interface improves optimisation algorithm selection and enhance decision making related to setup of relevant of algorithmic parameters. The developed user interface presents the user with the most relevant optimisation algorithms out of those available in the programme and allows easy modification of algorithmic parameters also generates atomically all the files needed to couple the GenOpt and the model. The developed user interface enables the aggregators to successfully apply GenOpt and use different algorithms to solve the optimisation problem in a user-friendly environment hence no prior coding knowledge is needed. The created algorithm selection framework of the user interface acts as a decision support system to allow the aggregators to take the most relevant and effective approach. Such approach limits the algorithm selection errors and allows the user to pick up most appropriate algorithm for the optimisation.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Scope and objectives of the deliverable

In recent years, transformation of traditional power systems has received a great deal of attention by the power providers over the globe moving from a centralized electricity power generation plants operated by large utilities towards clusters of distributed generation networks. These distributed generation networks are termed as Virtual Power Plant (VPP) and usually consist of mixed types of power generation sources. It usually comprises renewable energy sources, low power generators, storage devices and flexibility of the prosumers participating in the Demand Response programs (DR). The main advantages of the VPP are increased efficiency and reliability, lower cost and reduced environmental impact by integrating renewable energy resources (RES) and DR programs into existing markets. By aggregating a different kind of power generation sources in a VPP, the assets can be forecasted, optimized, and traded like one single power plant. Additionally, individual small plants which does not meet the minimum bid size of the market can be aggregated into a VPP and trade as a large central power plant. However, some complexities arise when optimising the VPP power generation and care should be taken due to the fluctuations in the generation of renewables and variations of the availability of the flexibility offered by DR programs. These uncertainties have a great impact on the power system stability and need to be considered to maintain the security and reliability of the grids while optimising the power generation cost.

The aim of the task T4.1 is to firstly, improve the short-term and long-term electricity production forecasts of the PV/RES systems. Secondly, to develop a decision-making and Optimisation engine toolset.

In particular, the core objective of task T4.1 is to develop a decision-making and optimisation engine toolset that enables the operators (e.g. DSOs and Aggregators) to optimally plan, schedule and make decisions regarding the VPP power generation assets. The toolset should consider during the optimisation process the uncertainties caused by the participation of non-flexible resources (e.g. RES generation and flexibility offered by prosumers) on the load dispatch. The toolset should first identify the optimal VPP balancing reserve based on the uncertainty analysis of the power consumption and production forecasts and then provide optimal schedule of the controllable generation units.

1.2 Structure of the deliverable

The deliverable is structured into five sections in which the specifications, algorithms and results of the optimization toolset modules are described. The report is organized as follows:

- The introductory chapter describes the scope and objectives of the deliverable and contributions of partners.
- Chapter 2 improved short-term forecasting of RES generation. It describes the trend analysis algorithm and how it can be used to improve the forecast.
- Long-term forecasting of PV systems is presented in Chapter 3. This includes the description of the employed method, results, and discussion.
- Chapter 4 presents the specifications of Improved Decision-Making and DR Optimization toolset. This describes the generic structure, decision making method and optimization engine tool.
- Chapter 5 presents the conclusions of the report.

1.3 Contribution of Partners

Partners contributed to this deliverable are CERTH, ENG and TU. CERTH contributed to the chapter 2 short-term forecasting of RES generation. ENG contributed to the external interfacing, implementation of APIs and communication of the developed components. TU contributed to overall organization, writing and editing of the document as well as Chapter 3 development of long-term forecasting of RES generation and Chapter 4 optimisation and decision making.

2 Trend analysis algorithm

This section elaborates on the employed near real-time trend analysis algorithm as well as the conceivement of the method used to improve the forecasting capabilities of this component.

The distributed generation resources can be quite challenging for the power networks which traditionally operate with deterministic inputs. A wide range of methods and techniques has been utilized in order to improve the managing capabilities of generation variability [1] [2] [3]. The current research effort focuses on modelling the uncertainty that weather conditions can generate regarding the possible forecasted and actual values of the PV production that contributes to the grid. This is a great problem for the grid operators since it enhances the weight of uncertainty variables that are imposed. The aggregators and BRPs (Balance Responsible Parties) most of the times are dependent on accurate forecasts in order to maintain their system's reliability after applying optimal DR scenarios and programs. The problem of electricity generation forecasting in micro grids, virtual power plants and community-based VPPs is often considered as a non-linear prediction problem.

It is quite common to refer to short-term forecasting problems with hourly or quarters of hours resolutions while higher time resolutions can be more demanding to handle [4]. Our aim is to develop a system for the aggregators (DSOs) that is able to predict, monitor, schedule, adjust and make decisions regarding the energy used and the energy that is produced in near real-time scale. On a smaller scale we want to monitor and spot faults from the PV production. That way we can improve our modelling and predicting capacity of energy in distributed generation resources while improving our utilization of Demand Response (DR) or Demand Side Management programs. In this section we refer to a novel near real-time trending analysis tool that have been used in other domains [5] but it was modified to work for our use case.

Symbol	Description
H_0	Null hypothesis
H_1	Alternative hypothesis
SSP	Slope Statistical Profile
B_T	Slope variable
t	Statistical test (t-statistic)
a	Significance level of null hypothesis rejection
U_i	Profile of t-statistic

Table 1 Nomenclature for Trend Analysis

Description of the Slope Statistic Profile (SSP) based method for robust malfunction diagnosis [5]

Slope Statistic Profile, denoted hereafter as SSP, is a method that detects the single structural break T from no trend to linear trend assuming a time series Y_t=$d_t + \varepsilon_t, t = 1, \ldots, n$, where d is specified as $d_t = \mu_0 + \beta B_T$. B_T is a variable for the slope change at time T as it depicted below:

$$B_T = \begin{cases} 0, & t \leq T \\ t - T, & t > T \end{cases}$$

The working null hypothesis H_0 is that the time series is stationary, and no trends exist. The alternative hypothesis H_1, which is the desired in our case, is that there is a structural breakpoint *T*, so that β is the coefficient of the linear trend starting at *T*.

The proposed method to detect the structural break in a time series is the use of a standard parametric linear trend test is used in order to find T and is denoted as t-statistic. The t-statistic for the parametric linear trend test is $t = \frac{\beta}{s(\beta)}$, where β is computed through the trend parameters and s(β) is the standard error of β. The null hypothesis of no trend is rejected at the significance level *a* if $|t| \geq t_{w-2,1-a/2}$, where the levels were set at 5% and 20%. The t-statistic is calculated on overlapping sliding data windows of size w with sliding step one along the time series. Thus, we obtain the profile of the t-statistic, denoted as $\{U_i\}$, for $i = 1 + \left[\frac{w}{2}\right], \ldots, n - [\frac{w}{2}]$. The form of this profile depends on time series characteristics, i.e the strength of the autocorrelation, the distribution of the residuals, the strength of the linear trend and the size of the sliding window w.

A first candidate for the breakpoint *T* is the time point at which the profile crosses the threshold line of rejection of the null hypothesis of no trend. The U-profile most probably does not exhibit a sudden change from small magnitudes, in the absence of trend, to large magnitudes, in the presence of trend, but there is *s* smooth transition due to the use of sliding windows with step one.

SSP methodology has the ability to detect all the kinds of changes on the linear trend of a time series. Based on that we modified and used it to perform the diagnosis it offers on PV production data.

2.1 Requirements for the component

Functional requirements (FRs) for the Trend Analysis component:

Component: PV/RES Degradation and Trend Analysis	
Function requirement ID	**Description**
MF01_BR04_UR01_F01	Obtain data for field devices' physical parameters and constraints
MF01_BR04_UR02_FR02	Receive forecasted data for weather conditions from Weather APIs.
MF01_BR04_FR03	Receive historical data for measurements related to generation assets from Decentralized Repository.
MF01_BR04_FR04	Receive historical data for weather conditions.

Table 2 Functional requirements for the Trend Analysis component

In addition, with the Functional requirements described in the table above, the user interacts with the trend analysis view as follows:

- The user has access to the visualization of the results of the trend analysis output.
- The user has access to the data that the trend analysis algorithm uses in order to produce its output.
- The user can choose a datetime range in order to specify the input of time series data to the trend analysis algorithm.
- The users that will have access to this view are the Aggregators and the DSO (excluding Prosumers).

Non-functional requirements (NFR) for the PV and Trend analysis component:

This component has no specific non-functional requirements but it is bound by the requirements that the HMI component has (MF02_BR05_BR10_MF03_BR05_NF0[1-7]) that described in the non-functional requirements of eDREAM deliverable D2.5. Examples of these are the following:

- The specific component should be capable of handling at least 100 simultaneous requests to return the results of the trend analysis algorithm and visualize them without its delays or diminishing performance.
- The component should be portable and not have its normal functionality tampered when Users interact with it from different OS or browsers.

The hardware platform specification:

The recommended specifications of the server should use Intel(R) Core (TM) i5-6600 CPU @ 3.30GHz and 8Gb RAM memory. The operating system of the server is the Windows 10 Pro (for testing purposes). Data are stored into a MariaDB database (version: 10.1.19). The JavaScript runtime environment NodeJS version 12.0.0 is being used during the development along with Sequelize Framework Version: 5.8.5. For the Client-side, any recent Web Browser can be used, (e.g., Chrome with its latest version, Firefox with its latest version, Edge with at least two most recent major versions, IE versions 11,10, 9, IE Mobile version 11, Safari with two most recent major versions, iOS with two most recent major versions, Android versions X 10.0, Pie 9.0, Oreo 8.0, Nougat 7.0).

Regarding the software environment/framework and programming language we are using Angular 8 for the frontend of this component and NodeJS for the backend and the data storage. The current test set for feeding the trend analysis algorithm gets the data from CERTH's smart house PV. PV output energy (in kWh) is measured every 15 minutes. The aforementioned selection of time interval measurement has been done in order to serve the needs of the rest eDREAM tools. Furthermore, such sampling rate has the merit of not burdening heavily neither the communication network nor the eDREAM repository infrastructure, whilst it provides adequate accuracy for the collection of measurements used in the eDREAM forecasting engines. Once the analysis is finalized, the SSP tool returns a list with the results of the analysis. The list contains the real measurements of the PV in the form of time series: $[t_0 \colon 1.5kWh, t_1 \colon 1.65kWh, \dots]$. The output analysis of SSP is the t-statistic, which is applied on every measurement of the PV measurements, in order to find the structural break *T*.

As reported earlier, the working null hypothesis H_0 is an assumption that the timeseries process is stationary (mean, variance, autocorrelation, etc. are all constant over time) therefore, there is no indication that a trend will appear. The values of t-statistic on the graph essentially mark the threshold line of rejection of the null hypothesis (there is no trend). The upper bound# [1-2] and lower bound# [1-2] indicate the candidate breakpoints that point to a trend detection (as shown in Figures 1-4).

2.2 Testing of the Trend Analysis component

MF01_BR04_UR01_F01 (Obtain data for field devices' physical parameters and constraints).
MF01_BR04_UR02_FR02 (Receive forecasted data for weather conditions from Weather APIs).
MF01_BR04_FR03 (Receive historical data for measurements related to generation assets from Decentralized Repository).
MF01_BR04_FR04 (Receive historical data for weather conditions).

Test scenario for Trend Analysis		
Step (actions)	**Obtained result**	**Verdict**
Log in the UI of the component.	-	Passed
Check availability of datetime range for performing the trend analysis.	-	Passed
The user requested the output of the Trend Analysis for 4 different Use Cases. For 1 day, 7 days, 14 days or 30 days respectively.	-	Passed
Establish and test connectivity with field devices and obtain data (in our case the field devices are PVs).	The connection is established and the data is received from the devices fulfilling the requirements of functional requirement MF01_BR04_UR01_F01.	-
Get historical and forecasted weather data from Weather APIs.	Using Weather APIs to obtain historical and forecasted weather data.	Passed
Extract the trend analysis results for the chosen datetime range of Step 3.	Obtain the profile of t-statistic for PV timeseries, in order to find the structural break T.	Passed
Assess whether the requested data were updated according to the user choices of the datetime range pick and Use Case section (1,7,14,30 days) of Step 3.	The visual graph of Trend Analysis results is correctly displayed and matches the user input criteria for the 4 different Use Cases: Figure 1 for 1-day, Figure 2 for 7 days output, Figure 3 for 14 days output or Figure 4 for 30 days output.	We conclude that the component functions are in accordance with its description and updates its output based on the input (selections of the user).

Table 3 Test scenario for Trend Analysis

The following figures present the output of the trend analysis algorithm as it performs under 4 different historic time periods. Figure 1-4 depicts the real values of the PV production in the top graph while the output of the SSP algorithm is shown in the bottom graph.

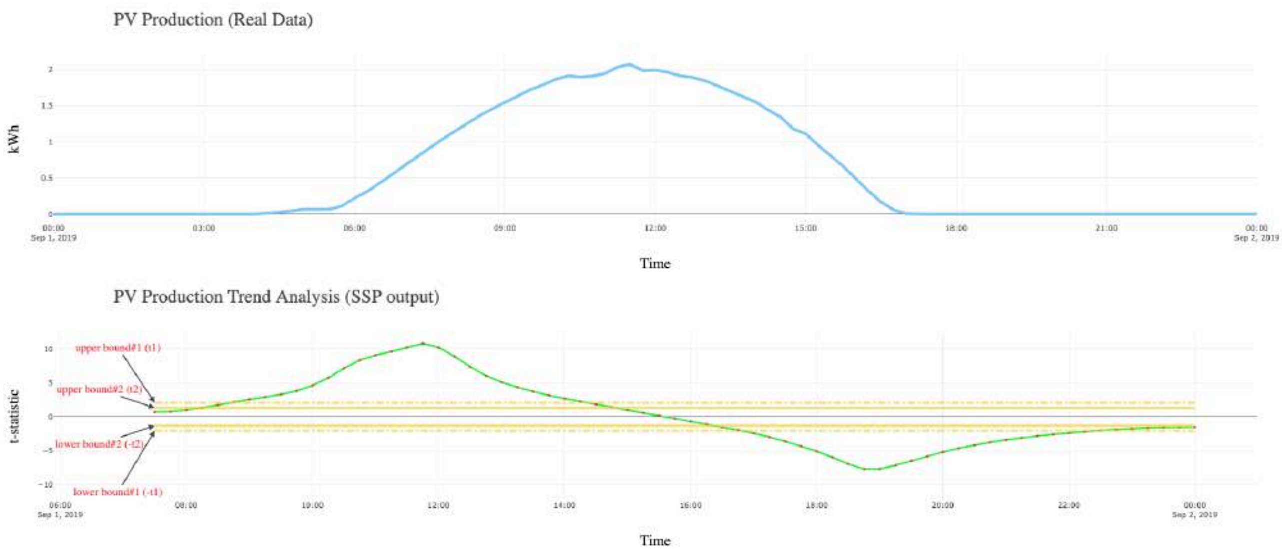


Figure 1 Trend analysis output for 1 day

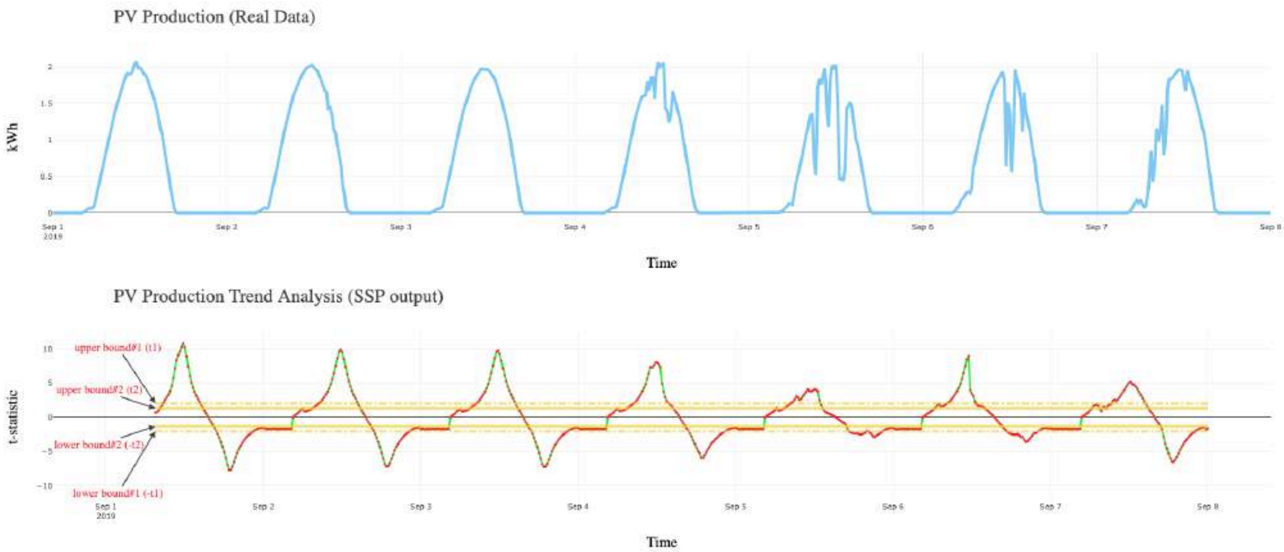


Figure 2 Trend analysis output for 7 days

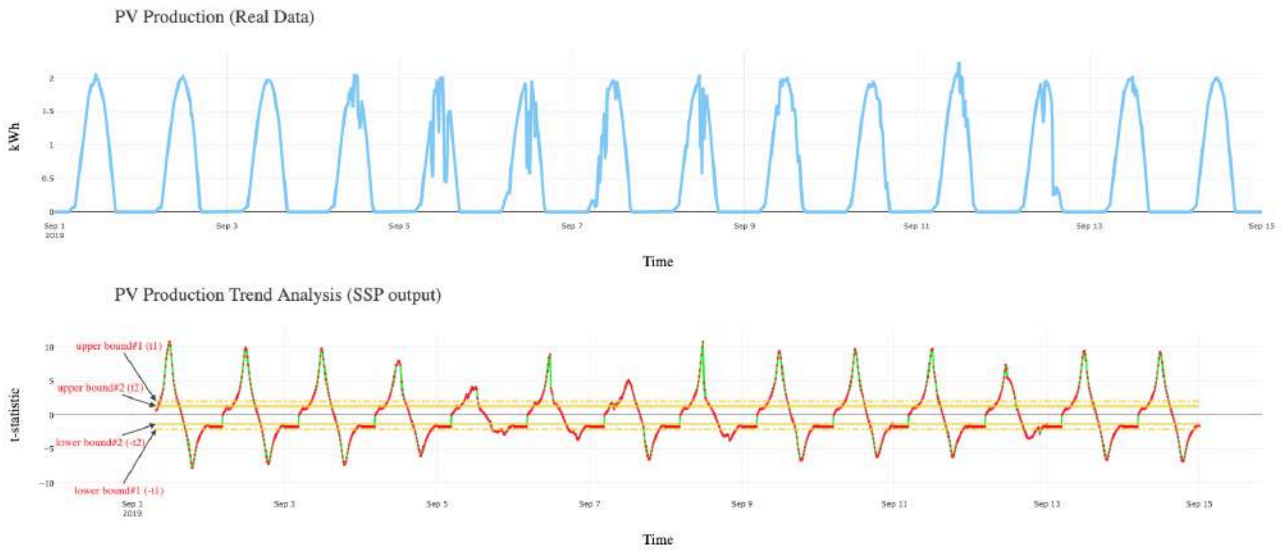


Figure 3 Trend analysis output for 14 days

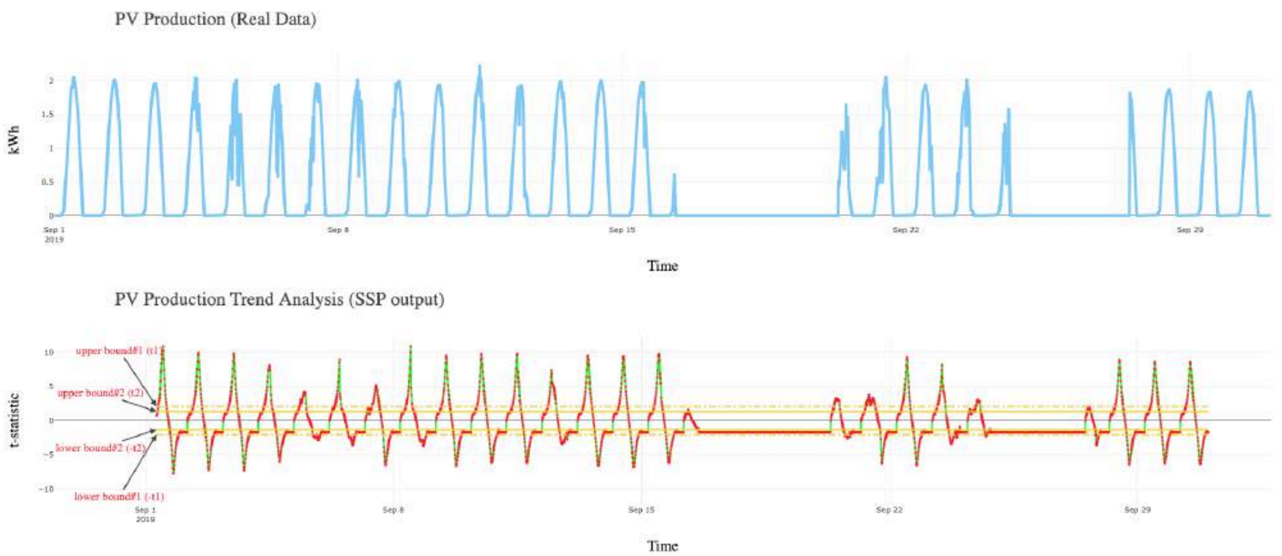


Figure 4 Trend analysis output for 30 days

In the last figure we observe a stable trend line in 2 parts of the trend analysis algorithm output, which is a result of zero production (due to a technical issues) from the PV panels. Therefore, the t-statistic remains stable inside the boundary values of -t1 and -t2 where the null hypothesis is true and not rejected. As a result, there are no changes on the linear trend generation of the timeseries (PV generation), proving that there is no actual change on the PV production.

2.3 Interfacing of the Trend analysis with the eDREAM platform

The final implementation of this component will allow the user to pick a specific time slot (though a datetime picker component-startDate: endDate) and request the trend analysis results in a visualization graph. The component's communication and interaction with other eDREAM platform components is achieved via API, the API specification being:

Methods & Endpoints

Method	Endpoint	Description
GET	/trendAnalysis/{prosumerDeviceId_ID}	Through this interface, other modules may retrieve the trend analysis output for a specific PV system

Table 4 Trend Analysis Methods & Endpoints

Authentication Scheme

The component API uses HTTP basic authentication.

Requests and Examples

GET trendAnalysis example Request URL:

https://virtserver.swaggerhub.com/eng94/aaa/1.0.0/trendAnalysis/12

This assumes that the database is available and running in order to satisfy the request.

GET trendAnalysis example response:


```
{
 "analysis": [
  {
   "number": 0
  }
 ],
 "forecasted": [
  {
   "number": 0
  }
 ],
 "limits": {
  "t1": 0,

  "t2": 0

 },
 "real": [
  {
   "number": 0
```

```
    }
  ]
}
```


The mentioned above description indicates that the output is an object containing the analysis, forecasted, limits and real attributes with specific values.

Expected Responses

1. status = 500 (server error)

   This error happens when the API is called but the server on which it runs is down.

2. status = 401 (not authenticated)

   This error happens when the user who calls the API is not authenticated within the system.

3. status = 403 (unauthorized)

   This error happens when the user who calls the API is authenticated within the system but he does not have the necessary authorization to call it.

4. status = 200 (successful)

The API returns the degradation rate in the form of an object or the trend analysis output

3 Long-term forecast for PV/RES generation

The development of a long-term forecasting model for RES power production requires an accurate estimation of the degradation rate. The term degradation rate (Rd) is defined as a maximum rate of reduction at which RES lose their performance over time and is expressed in %/year. Recent studies suggest that wind farm production degrades by a significant margin each year [6]. For PV devices, long term monitoring and testing in the field has proven that gradual degradation affects the rated power of PV. Most PV manufacturers guarantee that at the end of 25 years of operation a maximum of 20% reduction of the datasheet maximum power production can be observed. They also report that the highest degradation rate for the PV model is 1%/year for the first ten years of use. However, several studies published in the literature showed that these assumptions are not always true. A comprehensive review conducted on [7] proved that the degradation rate of the PV model can be far different from what has been anticipated and it showed that the PV will not offer its theoretical useful lifetime in every operating environment. On-field actual value of degradation rate depends primarily on PV technology and age but also climate type, mounting and geographical location [8]. Cumulative history of exposure to meteorological conditions as a result of the geographical location of the installation might lead to a different Rd even for identical systems. Field studies showed that the PV systems installed on diverse and harsh climates have higher degradation rates [9].

Different methods have been proposed in the literature to estimate the value of Rd for a PV system, these can be classified into two categories; indoor based and field based. Both approaches have their limitations, indoor simulation-based methods are time consuming for large PV systems and inefficient due to the difficulties to simulate in detail the outdoor operating conditions. Furthermore, it requires transporting the model to the testing facility which imposes the risk of failure of the module due to mishandling. On the other hand, the field-based methods require long-term monitoring of the PV performance. It has been found that a minimum of 3-5 years of data is needed to obtain accurate Rd due to seasonality. Reducing the impact of outliers in measured data adds another layer of complexity to these methods. To calculate the Rd, a statistical technique needs to be applied to the time series constructed with a specific PV performance metric derived from these measurements. The goal of the statistical techniques is to calculate the trend of the constructed time series and translate its slop to an annual degradation rate. Several statistical analysis methods were proposed in the literature the major recognized once are (1) Linear Regression (LR), (2) Classical Seasonal Decomposition (CSD), (3) Auto Regressive Integrated Moving Average (ARIMA) and, (4) Year-on-Year (YOY). Through a literature search, the LR is the most commonly used method to calculate the Rd. LR applies ordinary least squares to find a linear line of best fit for the PV performance time series. The statistical model of least squares is of the form $y = mt + b$ where y represents the fitted value, m slop and b intercept of the trend and are the variables being solved for. The value of m and b are determined by minimizing the sum of squared residuals between the fitted line and the time series and the Rd is the slop of the line best fit. This method is very sensitive to outliers and seasonal variations and can result in large uncertainties. In order to overcome the limitation of the LR method, the CSD which is more advanced than LR has been implemented in this deliverable.

3.1 Statistical Method: Classical Seasonal Decomposition CSD

The main concept of CSD is that the long-term trend of PV performance data series consists of three components; trend, seasonality and residual and it is assumed that the seasonal component of the performance is constant from year to year. In CSD method the trend and seasonality are isolated from the time series, seasonal indexes are determined, data is deseasonalized and then the standard LR method is applied to the deseasonalized data to calculate the Rd.

Mathematical representation of CSD is:

$$Y_t = f(S_t, T_t, E_t)$$

where:

Y_t is the time series value (original data) at month t

S_t is the seasonal component at month t

T_t is the trend cycle component at month t

E_t is the residual component at month t

The trend can be extracted by applying 12-month centred moving average on the data series. The trend at time t can be determined as:

$$T_t = \frac{1}{2}\left(\frac{1}{k}\sum_{i=t-m}^{t+m-1} Y_i + \frac{1}{k}\sum_{i=t-m+1}^{t+m} Y_i\right)$$

where:

k is seasonal period and = 12 (because of the number of the months in the year)

m is defined as half width of moving average and $= k/2$

t is the month order on the time series and $t > m$

CSD can be implemented using additive or multiplicative decomposition, selecting which method to use depends on the stability of the seasonal component. If the seasonal variation looks constant; it doesn't change when the time series value increases, the additive model is recommended. The multiplicative model is used if the seasonal variation increases as the magnitude of the time series increases. The stability of the seasonal component can be determined by visually inspecting the time series data.

3.1.1 Additive decomposition

After extracting the trend, the detrended series (seasonality-error component) Se_t is calculated by subtracting the trend from the original data.

$$Se_t = Y_t - T_t$$

To estimate the seasonal component for each month, firstly the unadjusted seasonal index (USI) is calculated by averaging the seasonality-error values for that month. For example, the seasonal component for January is the average of all the seasonality-errors of January values in the data.

$$USI_i = \frac{1}{n} * (Se_i + Se_{i+k} + \cdots + Se_{i+k*(n-1)})$$

Where i represents the months from 1 to 12 and n is the total number of that month on Se_i.

The seasonality is then calculated by adjusting the USI_i. The adjusting of USI_i is done by ensuring that they are add to zero.

$$S_t = USI_i - \frac{\sum_{j=1}^{k} USI_j}{k} \quad where \quad i = t - \left\lfloor\frac{t-1}{k}\right\rfloor * k$$

The remainder E_t is calculated by subtracting the seasonality S_t from the seasonality-error component.

$$E_t = Se_t - S_t$$

Deseasonalized data D_t is produced by subtracting the seasonal component from the original data.

$$D_t = Y_t - S_t$$

3.1.2 Multiplicative decomposition

In this method, the seasonality-error component Se_t is calculated by dividing the measured data by trend.

$$Se_t = \frac{Y_t}{T_t}$$

To estimate the seasonal component, the unadjusted seasonal indexes (USI) is calculated as in equation above then adjusted by ensuring that they are add to k.

$$S_t = USI_i * \frac{k}{\sum_{j=1}^{k} USI_j} \qquad where \quad i = t - \left\lfloor \frac{t-1}{k} \right\rfloor * k$$

The remainder E_t is calculated by dividing the seasonality S_t by the seasonality-error component.

$$E_t = \frac{Se_t}{S_t}$$

Deseasonalized data D_t is produced by dividing the original data values by their adjusted seasonal indexes.

$$D_t = \frac{Y_t}{S_t}$$

3.1.3 Estimation of Rd

Linear regression is applied to the deseasonalized data D_t to develop a fitted trend line which is given by:

$$y = ax + b$$

where a is the slope of the line and b is the intercept value and are given by:

$$a = \frac{n * \sum_{t=1}^{n} D_t . t - \sum_{t=1}^{n} D_t * \sum_{t=1}^{n} t}{n * \sum_{t=1}^{n} {D_t}^2 + \left(\sum_{t=1}^{n} D_t\right)^2}$$

$$b = \frac{\sum_{t=1}^{n} {D_t}^2 * \sum_{t=1}^{n} t - \sum_{t=1}^{n} D_t * \sum_{t=1}^{n} D_t . t}{n * \sum_{t=1}^{n} {D_t}^2 + \left(\sum_{t=1}^{n} D_t\right)^2}$$

The literature includes two definitions of the performance loss ratio, absolute and relative terms. The absolute form represents the absolute estimated performance gain or loss in one-year time, and the quantity:

$$Rd = 12 * a$$

The relative annual rate of degradation is calculated as

$$Rd = \frac{12 * a}{b} * 100$$

3.1.4 Forecast of power generation

In order to forecast the power generation in month t the trend data value is added to their seasonal index for additive decomposition or multiplied by their seasonal index for multiplicative decomposition.

$$\overline{P_t} = a.t + b + S_t$$

$$\overline{P_t} = (a.t + b) * S_t$$

The forecast of the power generation of a year n is calculated by summing up the generation of all months in this year.

$$\overline{yP_n} = \sum_{t=n+(n-1)*11}^{t+11} \hat{P}_t$$

3.2 User guidelines of the component

The user interface of the component is presented in Figure 5. This interface enables the user to input all the information required to perform the calculation of the Rd. The interface consists of three main panels; control panel, algorithm output panel and plotting area.

The *"Load data"* button allows the user to upload the historical power generation data of the PV systems from the CSV file. The file should contain three columns; prosumer asset ID, date and time, and measured power. If there is more than one prosumer in the file the software will split the file and generate a separate data file for each prosumer. The *"Select Prosumer"* allows selection of the prosumer asset and *"Prosumer Method"* to choose the calculation method Additive or Multiplicative. The panel also allows the user to set the number of years the algorithm will forecast and whiter to enable the filter or not. In the algorithm output panel, the Mean Absolute Percentage Error (MAPE) and statistical uncertainty of the testing data are displayed. The algorithm results are saved in CSV file and plotted on the user interface. Two plotting areas are shown on the interface; first plot shows the actual measurements, extracted trend and Regression line. In the second plot the user can chose to plot the actual measurements, data after applying the filter or forecasts.

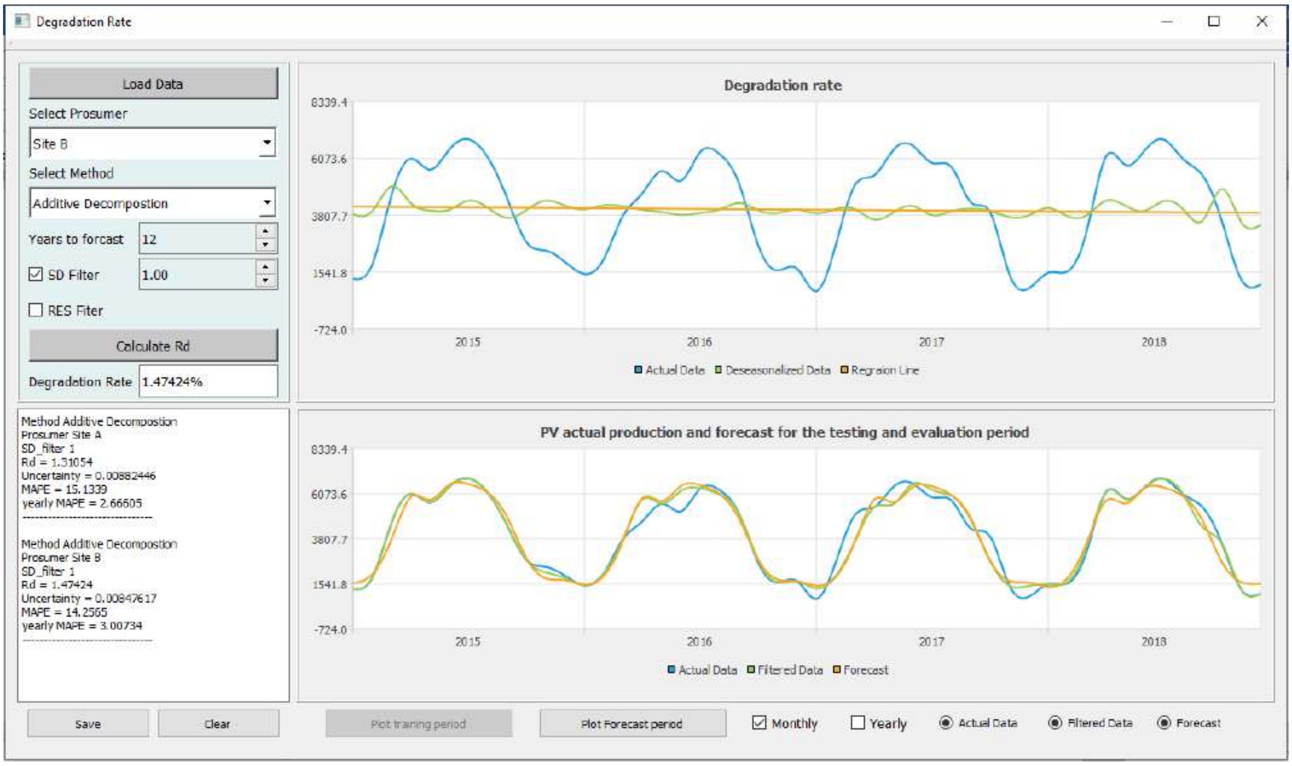


Figure 5 View of the degradation component interface

The degradation rate software is developed and tested on Windows 10 with Microsoft Visual studio 2019 and Microsoft C++ compiler (MSVC). The graphic user interface is developed using Qt 5.12.6. There are no specific hardware requirements for the component, any computer with Intel(R) Core (TM) i5 CPU or equivalent and 8 GB RAM is sufficient.

3.3 Testing and evaluation of the component

The method was applied on four-years data set received from the UK pilot sit (KIWI). The data set contains recorded power generation of PV systems in the period of 01 January 2015 to 31 December 2018 sampled at 15-minute frequency. In order to extract the trend, seasonality and residual, the data is first aggregated into monthly power generation then filtered to remove the outliers. Filtered data is used to estimate the Rd value then the estimated Rd is used to forecast the power generation of the PV system. For the evaluation, two metrics are used.

3.3.1 The statistical uncertainty

The procedure proposed by the Guide to the Expression of uncertainty measurement [10] is used to calculate the statistical uncertainty of the estimated Rd, and given by:

$$U_{Rd} = \sqrt{\left(\left(\frac{12}{b}\right)^2 . u_a^2 + \left(\frac{12a}{b^2}\right)^2 . u_b^2\right)}$$

where u_a and u_b are the variances of the fitting coefficients *a* and *b* of the regression equation and are given by:

$$u_a^2 = \frac{e}{n-2} \frac{1}{\sum_{t=1}^{n}(t_t - \bar{t})^2}$$

$$u_b^2 = \frac{e}{n-2} \left(\frac{1}{n} + \frac{\bar{t}^2}{\sum_{t=1}^{n}(t_t - \bar{t})^2}\right)$$

$$\bar{t} = \frac{\sum_{t=1}^{n} t_t}{n}$$

$$e = \sum_{t=1}^{n} \left(D_t - (a.t + b)\right)^2$$

3.3.2 The Mean Absolute Percentage Error (MAPE)

The MAPE is computed between the actual measurements and the forecasts and given by:

$$MAPE = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{E_i^{actual} - E_i^{forecast}}{E_i^{actual}} \right|$$

where E_i^{actual} is the actual power measurement of the PV system at month i , $E_i^{forecast}$ is the power forecast of the PV system at month i and n is the number of months.

3.3.3 Filtering

Prior applying the CSD method, a filter depending on the standard deviation ranges is applied to the actual data. This step is performed to minimize outliers, noise and to decrease the overall uncertainty in the estimation of Rd. The filter was applied by calculating the average and standard deviation of all similar months on the dataset, the data points which exceeded or fell below specific limits are identified as anomalies. The limits were defended as μ ± k.σ, where the value of k is initially set to one, however, it can be changed by the operator if the uncertainty in the estimation of Rd was high. It should be noted that a tightly defined filter (low value of k) can significantly influence the calculated degradation rates, on the other hand, if the high value of k is used then the filter might not detect the anomalies. The points which are identified as outliers are replaced by estimated values based on the power generation of the PV system at the same time period of the other years.

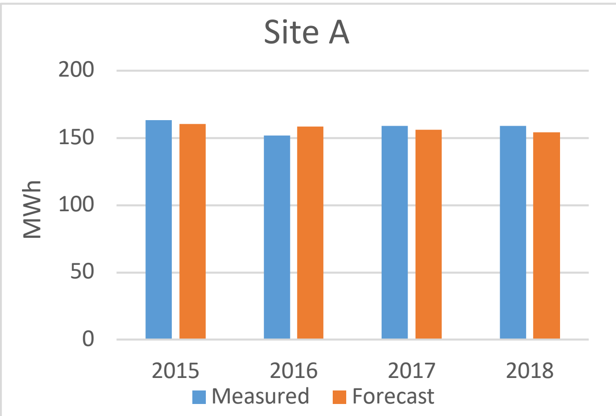
3.3.4 Results

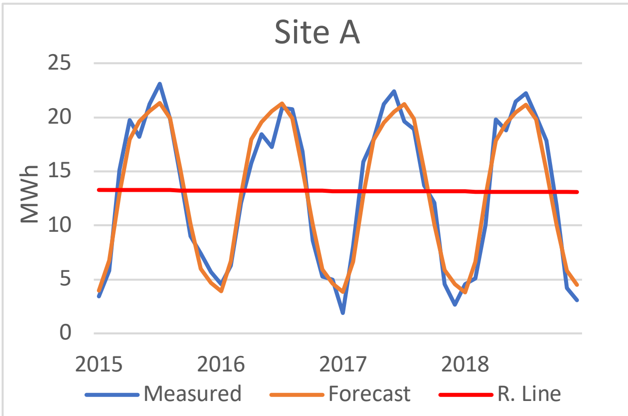
In this section, results for three sites marked as Site A, Site B and Site C will be presented. The estimated Rd, uncertainty and MAPE for monthly as well as yearly power generation forecast using additive decomposition are summarised. The output of the component is presented in Table 5 and Figures 6 and 7.

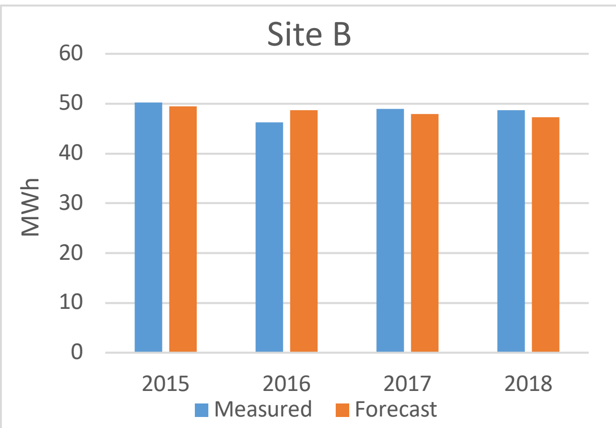
Method	Site	Rd %/y	Uncertainty	MAPE	
				Yearly	Monthly
Additive decomposition	Site A	1.31	± 0.0088	2.666	15.134
	Site B	1.47	± 0.0084	3.007	14.257
	Site C	7.83	± 0.0147	6.489	102.621

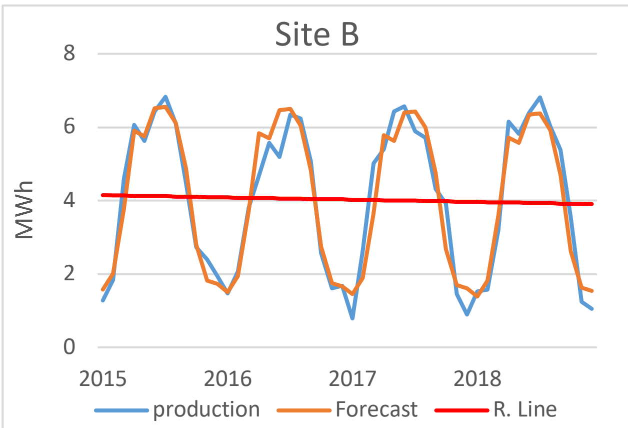
Table 5 The Rd as estimated by the model

A positive Rd implies that the system exhibits a performance loss whereas a negative rate implies that it exhibits performance gain. It is important to note that only metered inverter data is used so the estimated Rd cannot be entirely attributed to the degradation of the PV model because, additional factors such as soiling, shading, and degradation of the inverter can affect the overall performance of the system.









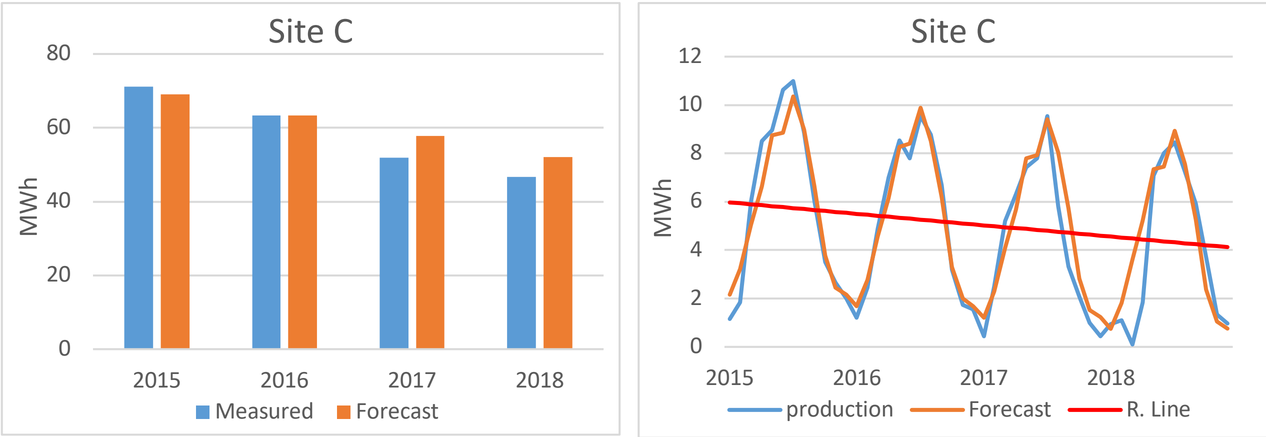


Figure 6 Actual and predicted power generation of the three sites

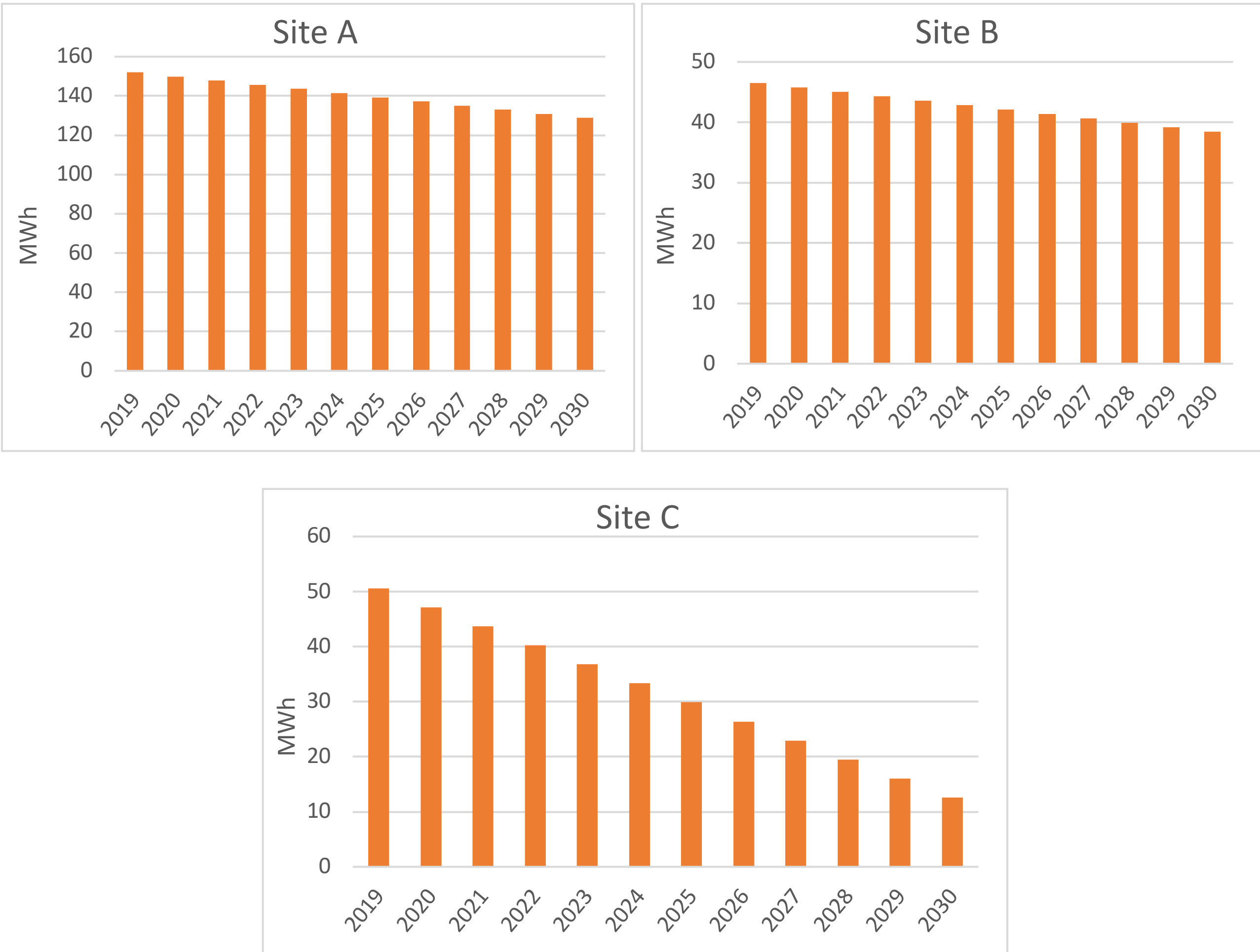


Figure 7 Forecast of power generation of the three Sits for the next 12 years

As can be seen from the Table 5 and Figures 6 and 7, a constant rate of performance degradation is evident. The estimated degradation rate of the system on Site A was 1.3 per year and for Site B was 1.47 per year. However, for Site C the estimated Rd and uncertainty were much higher 7.83 and ±0.0147 respectively. To reduce the uncertainty of the estimated Rd of Site C, the k value of the filter has been set to 0.55, consequently the estimated Rd was reduced to 5.45 and uncertainty to 0.0086 as well as the MAPE reduced to 8.

One way to evaluate the proposed forecasting method is to compare it is results with results obtained from a reference method "naïve forecasting method". Reference methods are therefore very simple forecasting methods and they are usually used as a benchmark for determining whether the values obtained by more complex methods are good or not. If the proposed method can significantly overcome the naïve method, then it is worthwhile developing and implementing the complex algorithm.

As it is expected that the power generation data of the PV systems have a constant seasonal variation and the values are decreasing over time due to the degradation, a simple method (Seasonal- Drift) based on seasonal and drift naïve methods is used.

$$\hat{Y}_{n+h} = Y_{n+h-k} + \left(\frac{Y_n - Y_{12}}{n-12}\right)$$

where n is the number of the months in the training data, k is the seasonal period and h is the number of the month in the forecast period. The forecast is set to be equal to the last observed value from the same season of the year pulse any variation (drift) seen in the historical data. This equivalent to drawing a line between the last observation and the observation of the same month on the first year in the historical data to estimate the drift. The results obtained from both methods for the three sites are listed in the Table 6. The results demonstrated that there is a significant improvement when the CSD method is used.

Site	MAPE	
	Additive decomposition	Seasonal-Drift
Site A	15.134	23.78
Site B	14.257	21.93
Site C	102.621	185.00

Table 6 MAPE for the three sites

3.4 Interfaces with other components of the eDREAM platform

The component's communication and interaction with other eDREAM platform components is achieved via API, the API specification being:

Methods & Endpoints

Method	Endpoint	Description
GET	/degradationRate/{prosumerDeviceId_ID}	Through this interface, other modules may retrieve the degradation rate of a specific PV system

Table 7 PV/RES Degradation Methods & Endpoints

Authentication Scheme

The component API uses HTTP basic authentication.

Requests and Examples

GET Degradation rate example Request URL:

https://virtserver.swaggerhub.com/eng94/aaa/1.0.0/degradationRate/12

This assumes that the database is available and running in order to satisfy the request.

GET Degradation rate example response:

{
    "timestamp": "string",
    "value": 1.5, //%
    "deviceID": "string"
}

The above-mentioned description indicates that the output is an object containing the timestamp, value and device Id attributes with specific values.

Expected Responses

5. status = 500 (server error)
   This error happens when the API is called but the server on which it runs is down.
6. status = 401 (not authenticated)
   This error happens when the user who calls the API is not authenticated within the system.
7. status = 403 (unauthorized)
   This error happens when the user who calls the API is authenticated within the system but he does not have the necessary authorization to call it.
8. status = 200 (successful)
   The API returns the degradation rate in the form of an object or the trend analysis output

4 Decision Making and DR Optimization engine

The main functionality of this tool is to optimise the operation of the VPP, which aggregates various types of distributed energy resources:

- Dispatchable Generators (DG) includes thermal units (using fossil fuels such as coal, oil and gas)
- Renewable Energy Sources (RES) such as wind power plants and photovoltaic units
- Flexible Energy Demand (FDA) the flexibility offered by the prosumers either by increasing or decreasing their consumption

The goal is to manage these resources optimally to help the aggregators to efficiently fulfil the requests from the DSO. The aim of the optimisation is to provide to the aggregators an optimal scheduling and set points of the DG considering the participation of the RES and FDA. The optimisation process will consider the fuel cost of the DG, RESs' power generation forecast and prosumers' flexibility estimation (FDA) which can be offered by DR programs. The use of RES and the FDA contribute to the aggregators' energy supply portfolio diversity and reduce the expand use of fuel-based generators which in turn reduce the environmental impact. However, the uncertainty associated with forecasting the power generated by these resources makes it difficult for aggregators to meet the exact demands of DSO, which may have a significant impact on the overall security and reliability of the grid unless an adequate power ensuring reserve (PER) is set. Additionally, the aggregator might be penalized if he was unable to supply the contracted level of power.

The PER is defined as the instantly extra power generating capacity available to aggregator by being able to increase the power output of the DG units that are already connected to the grid. An adequate PER is crucial to compensate unpredictable imbalance between the actual and forecast power generations and for ensuring that the generation schedule can withstand the uncertainty due to the participation of RES and FDA. For these reasons, the PER should be carefully sized but also ideally minimized to reduce the costs with a satisfying security level. A decision-making system (DMS) has been developed to guide aggregators to set an appropriate amount of the PER, taking into account the uncertainty associated with the forecasting of the RES power generation and load curtailment comes from the FDA. The optimisation engine will then consider the estimated PER level in order to obtain the optimal schedule of the generators along with optimal set points. Assuming that each scheduled (on-line) DG unit can be regulated continuously between its minimum and maximum limits, the fluctuations in the generation offered by RES and FDA can be balanced by ramping up and down the on-line DG units. Figure 8 shows the complete structure of the toolset.

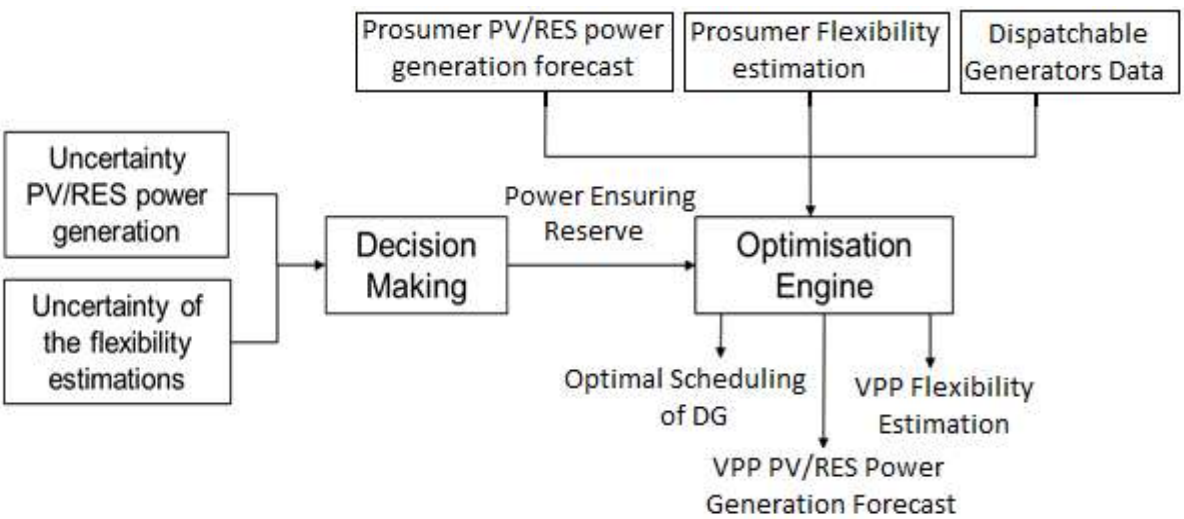


Figure 8 Decision Making and Optimization toolset

The optimization problem can be formulated as a customised unit commitment (UC) and economic load dispatch (ELD) problems. In UC optimisation, the operation schedule of the DG units is determined (which unit is on and which is off) whereas the ELD optimisation determines the optimal set points of the on-line units at every hour interval so as to meet the generation requested from the DSO at a minimum cost under various operating constraints. The problem can be solved by one algorithm where the UC and ELD are modelled as one problem or using the two-phase method in which two different algorithms are used. In the first phase, the UC schedule is determined and in the second phase, the ELD is solved.

4.1 General architecture of the toolset

Figure 9 presents the general architecture of the decision making and optimisation toolset which consists of three main components; user interactive interface, decision making and optimization engine, and unit commitment and economic load dispatch (UCaELD) problem model.

The user interactive interface is working as a supervisory program that controls the execution of the decision making and optimization programs and sets the configuration of the UCaELD model. The interface provides a user-friendly environment to use GenOpt and creates all the text files which are needed to configure it. The interface allows the user to benefit from the GenOpt and apply any of the implemented optimisation algorithms without a prior coding and interfacing knowledge of the GenOpt program. The selection of the optimization algorithm and the setting of the parameters can be done by simply changing their values on input text edits.

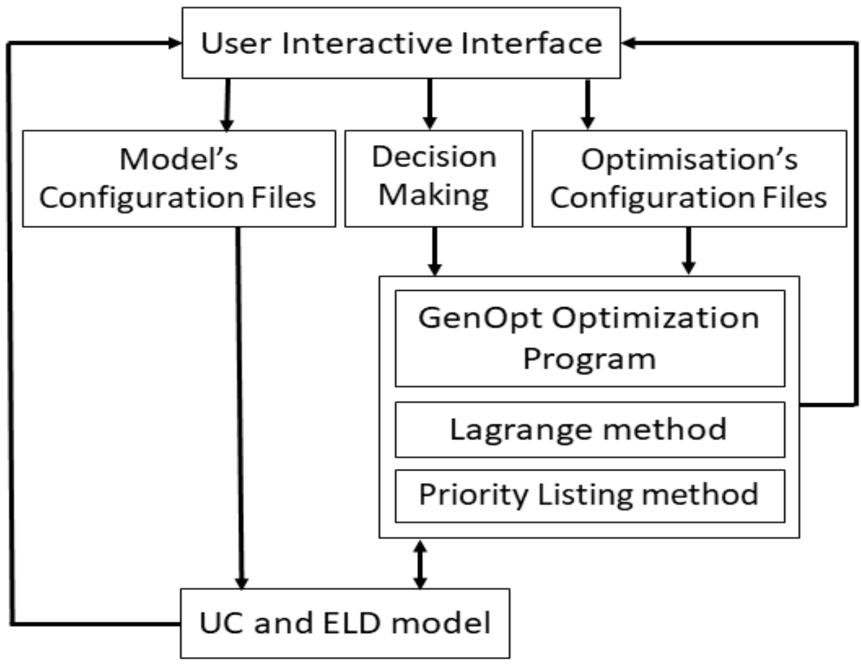


Figure 9 General architecture of the tool

Decision making and optimization engine is developed based on three different optimisation methods; GenOpt, Lagrange (LM) and Priority listing (PL). GenOpt is a generic optimization program developed by Lawrence Berkeley National Laboratory, the University of California for the minimization of a cost function that is evaluated by an external simulation program. Several algorithms are implemented within GenOpt library some of them can be used to solve both the UC and ELD problems while others are able to solve either UC or ELD, the user interactive interface provides guidance to the user on which algorithms can be used. Lagrange method is used to solve the ELD problem while the Priority listing is used to solve the UC problem.

GenOpt algorithms which can solve only one problem can be combined with either Lagrange method or Priority listing to solve both UC and ELD problems.

UC and ELD model (UCaELD) presents the objective function to be minimized in the optimisation process. The unit commitment and economic load dispatch problem of the power system are formulated mathematically considering the fuel cost of power generation. The fuel cost of the dispatchable generators is formulated as a quadratic function of the generated power. Thus, the optimum allocation of active power generation, among the generators, can be calculated for minimum generation cost considering the uncertainty comes from the forecast of PV generation and curtailable loads.

4.2 Decision making system

Two sources of uncertainties are taken in to account by the system, one coming from the forecast error of the PV power generation and the other from the estimation of the flexibility.

Uncertainty of the RES power generation forecast:

Uncertainty of the RES power generation forecasts can be expressed by the calculation of prediction intervals which are expected to contain the forecast points with a given confidence levels. The production forecast of RES systems is done with the Electricity Consumption/Production Forecasting component presented in the eDREAM deliverable 3.1. The forecasts obtained from this component are deterministic, only a single value for the RES power generation in each target hour is provided. Thus, the objective of this section is to present a method to calculate the prediction intervals of the forecast errors for one-day-ahead forecasts of power generation of a single RES system. The proposed method is based on assumptions that for a specific location, at a given time and similar weather data should yield similar forecasts and forecast errors and these errors should belong to the same distribution. Thus, a prediction interval of a specific RES power generation value will be based on the past forecast errors of the same RES for the days with similar input data. Considering that the same method/model used to forecast, similar input data yields similar forecasts. To identify past input data similar to the input data of a target forecast Euclidean distance was used as the similarity parameter. Therefore, to calculate the prediction intervals of a forecast of RES generation for a given hour, the forecast at this hour is compared with the hourly forecasts done in the previous 30 days. From this comparison the forecast errors of n most similar forecasts are retrieved and used in the calculation of the prediction intervals. Calculating this way for a given RES, the prediction intervals will vary according to the output data of the model which in turn depends on input data, weather conditions and target hour of the forecasts. The similarity is calculated as follows:

$$\{S_i(k),\ k\} = \left|\widehat{P_i^{RES}(t)} - \widehat{P_i^{RES}(k)}\right| \qquad \forall \ \ k = t-1, t-2, \dots, t-720$$

where $\widehat{P_i^{RES}(t)}$ is the target forecast and $\widehat{P_i^{RES}}(k)$ is the past forecasts. The $S_i(k)$ are sorted ascendingly and the forecast errors of the first 30 values are used to form a new dataset.

$$d_i = \{e_i(k)\}_{k=1}^{n}$$

where

$$e_i(k) = P_i^{RES}(k) - \widehat{P_i^{RES}}(k)$$

Then a bootstrap method is applied on the dataset d_i in order to estimate the prediction interval of the target forecast hour $\widehat{P_i^{RES}}(t)$. Bootstrapping is a statistical method that based on a random sampling with replacement technique. The method allows us to measure future uncertainty by only using the historical data. A number of B samples are constructed by drawing forecast errors from dataset d_i one at a time and returning them to the dataset after they have been chosen. This allows a given forecast error to be included in a given sample more than once. Doing this, we obtain B number of forecasts which is used to estimate the forecast error and it is upper and lower bounds which is subsequently use to obtain the prediction intervals as shown in Figure 10.

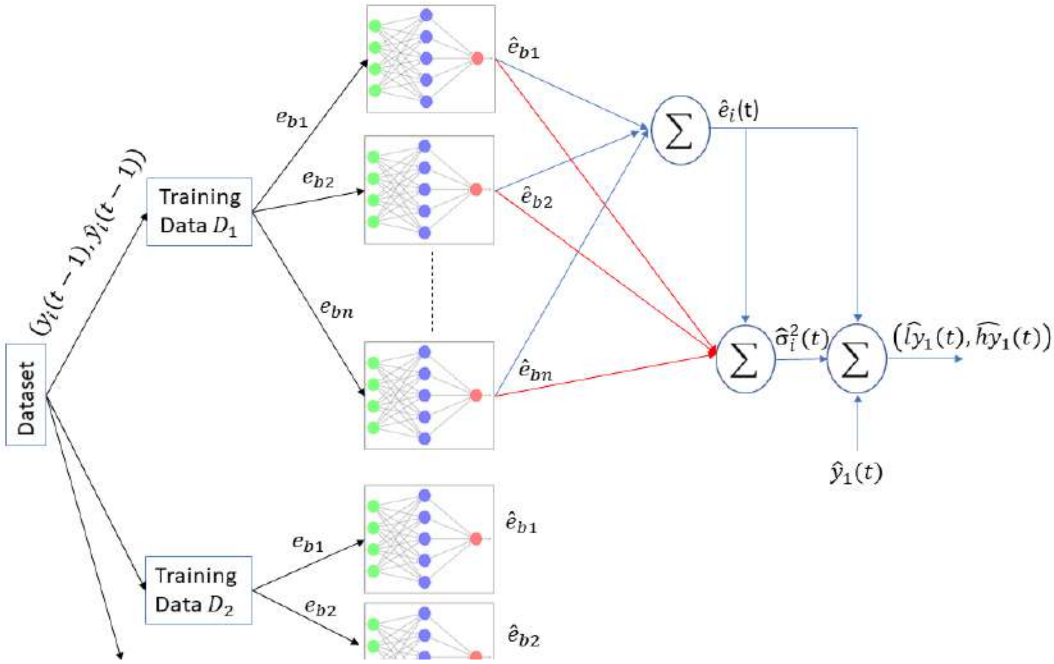


Figure 10 Ensemble of B NN models used by the bootstrap method

The method estimates the forecast error variance $\hat{\sigma}_i^2$, by building B NN models (Figure 10). The forecast error is estimated by averaging the point forecasts of B models

$$\hat{e}_i(t) = \frac{1}{B}\sum_{b=1}^{B} \hat{e}_b$$

where $\hat{e}_b$ is the prediction of the sample generated by the b^{th} bootstrap model and $\hat{e}_i(t)$ is the prediction of the forecast error of i^{th} RES system at hour t. Then, the variance is estimated using the variance of B model outcomes

$$\hat{\sigma}_i^2(t) = \frac{1}{B-1}\sum_{b=1}^{B} \left(\hat{e}_b - \hat{e}_i(t)\right)^2$$

Considering a 95% prediction interval of the forecast error, the upper and lower bounds of the RES power generation forecast is given by

$$\widehat{uH_i^{RES}}(t) = \widehat{P_i^{RES}}(t) + 1.96\,\hat{\sigma}_i$$

$$\widehat{uL_i^{RES}}(t) = \widehat{P_i^{RES}}(t) - 1.96\,\hat{\sigma}_i$$

The aggregated power generation of the RES participating in the VPP is given by

$$\widehat{P_{VPP}^{RES}}(t) = \sum_{i=1}^{v} (\widehat{P_i^{RES}}(t)$$

$$\sigma_{VPP}^{RES}(t) = \sqrt{\sum_{i=1}^{v} \sigma_i^2}$$

Flexible Energy Demand Assets (FDA)

The prediction of the energy flexibility at prosumer level is done by the Flexibility Forecasting component presented in the eDREAM deliverable 3.1. The energy flexibility is defined as the degree in which the prosumer can modify its baseline energy profile either by increasing or decreasing its load. The component predicts the flexibility below the baseline (APC_{below}) and the flexibility above the baseline (APC_{above}).

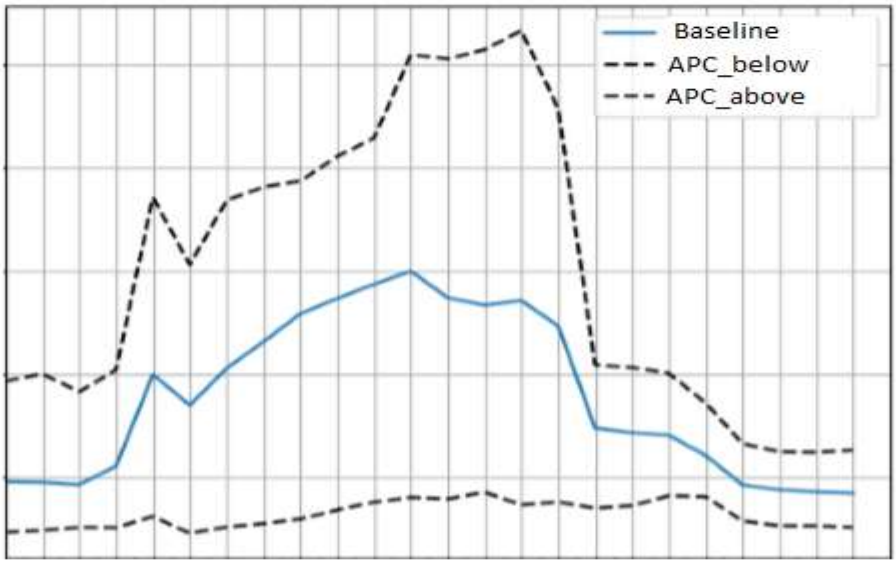


Figure 11 Example of features used for flexibility prediction

As seen from the Figure 11 the predicted consumption $\widehat{D_i^{FDA}}(t)$ of the prosumer *i* at hour *t* can be defined as

$$\widehat{APC_{below}}(t) \leq \widehat{D_i^{FDA}}(t) \leq \widehat{APC_{above}}(t)$$

As the interval was calculated with 90% confident the $\hat{\sigma}_i$ can be obtained by

$$\hat{\sigma}_i = \frac{\widehat{APC_{above}}(t) - \widehat{APC_{below}}(t)}{2 * 1.64}$$

The energy flexibility that can be potentially elicited by the prosumer *i* at hour *t* can be obtained by subtracting the equation above from baseline which is given by

$$\widehat{uL_i^{FDA}}(t) \leq \widehat{f_i^{FDA}}(t) \leq \widehat{uH_i^{FDA}}(t)$$

where

$$\widehat{uH_i^{FDA}}(t) = baseline - \widehat{APC_{below}}(t)$$

$$\widehat{uL_i^{FDA}}(t) = baseline - \widehat{APC_{above}}(t)$$

$$\widehat{f_i^{FDA}}(t) = baseline - \widehat{D_i^{FDA}}(t)$$

This way the flexibility offered by the prosumer will be positive if his consumption is below the baseline (decreases the consumption) and negative if the consumption above the baseline (increases the consumption).

The estimated total energy flexibility that can be potentially elicited by the prosumers participating in a VPP is defined as the sum of their flexibility profiles and given by

$$\widehat{f_{VPP}^{FDA}}(t) = \frac{1}{2}\sum_{i=1}^{f}(\widehat{uL_i^{FDA}}(t) + \widehat{uH_i^{FDA}}(t))$$

$$\sigma_{VPP}^{FDA}(t) = \sqrt{\sum_{i=1}^{f} \sigma_i^2}$$

The total RES and FDA estimation of the VPP

The total of the power generated by the RES and flexibility offered by the FDA participating on the VPP is given by

$$\widehat{P_{VPP}}(t) = \widehat{P_{VPP}^{RES}}(t) + \widehat{f_{VPP}^{FDA}}(t)$$

$$\widehat{\sigma_{VPP}}(t) = \sqrt{\sigma_{VPP}^{RES}(t)^2 + \sigma_{VPP}^{FDA}(t)^2}$$

The PER of a VPP according to a risk level and for each time step can be obtained. With a fixed risk index, the operator can then easily quantify the PER. The value of the PER which ensure a secure operation of the VPP can be calculated by

$$P_{VPP}^{PER}(t) = c * \widehat{\sigma_{VPP}}(t)$$

where the multiplier *c* depends on the selected coverage probability. The value of *c* is set to three which represents 99.73% of the prediction interval. The operator can decide to accept some risk and reduce this value. Two reliability assessment parameters are used to support the operator: the risk index x% and the expected energy not served $EENS$ which can be calculated as:

$$x\%\ (t) = 1 - prop(P_{VPP}^{PER}(t) < R) = \int_0^R pdf(\tau)\, d\tau$$

$$EENS(t) = P_{VPP}^{PER}(t) - prop(P_{VPP}^{PER}(t) < R) * P_{VPP}^{PER}(t)$$

The index x% represents the remaining probability that the predicted power generated by RES and FDA exceeds the actual power delivered (blue part in Figure 12) and EENS measures the magnitude of the power that might not be served.

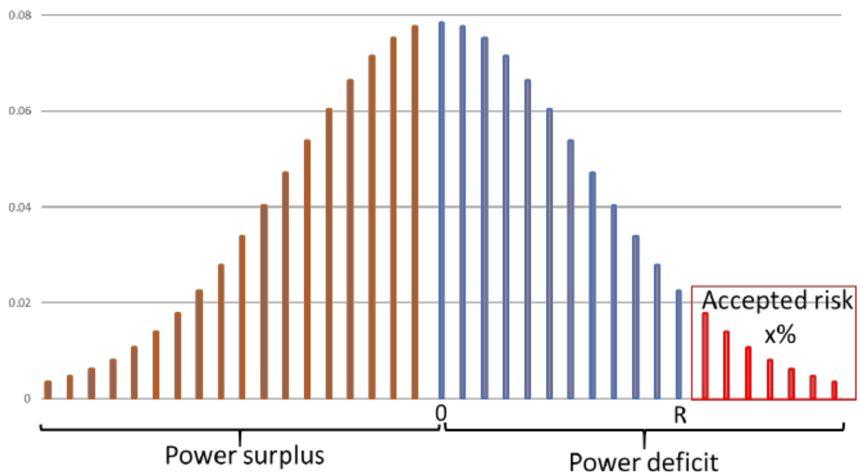


Figure 12 Calculation of PER requirements with x% index, at time step t

4.3 VPP and DR services optimization engine

In this section, the general structure, implementation, requirements, and interfacing of the optimization engine are described.

The optimization problem is formulated as a customized UC and ELD problem in which the DGs are used as flexible resources. The fuel cost, generating constraints of the DGs and PER are taken into consideration for solving the problem. The curtailable loads of the FDA and power generation of the RES are considered as non-flexible resources and their operation cost is taken as zero. The idea is that the online DGs can be ramped up and down to balance any fluctuations in the of the electrical power offered by RES and FDA.

The cost of the fuel per unit power output of the DG various significantly with its power output. A typical generation fuel cost characteristic is shown in Figure 13 where P_{min} is the output level below which it is infeasible to operate the unit and P_{max} is the maximum output power.

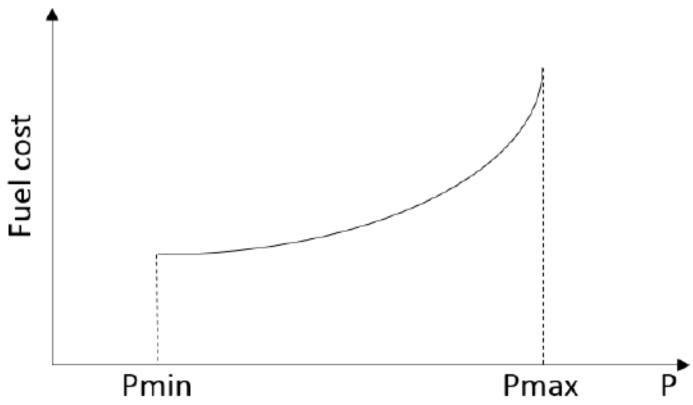


Figure 13 DG fuel cost characteristics

The generation fuel cost of any generator unit i can be modelled with a quadratic function of the real power output as:

$$F_i(P_i(t)) = a_i + b_i.P_i(t) + c_i.P_i(t)^2$$

where a_i, b_i and c_i are the cost coefficients of ith generator. The $P_i(t)$ is the power output of the generator i at time t.

The operating cost of the DG unit is an accumulation of the generation fuel cost, start-up cost and shut down cost. Generally, shut down cost is constant for each unit and the start-up cost depends on boiler temperature of a thermal unit and can be specified as:

$$ST_i = \begin{Bmatrix} HST_i & if & Toff_i \leq Tcold_i + Tdown_i \\ CST_i & if & Toff_i > Tcold_i + Tdown_i \end{Bmatrix}$$

where HST_i and CST_i are hot and cold start-up cost respectively. $Toff$ is the duration of hours in which the unit i has been offline. $Tdown_i$ is the minimum down time and $Tcold_i$ is the cold start hour.

The total operating cost (OC) of N DG units over the scheduled time period T is calculated by accumulating the operating cost of all committed units and can be given by:

$$OC = \sum_{t=1}^{T}\sum_{i=1}^{N} F(P_i(t)).I_i(t) + ST_i.X_i(t) + SD_i.Y_i(t)$$

where

$I_i(t)$ is the control variable of the unit i at hour t (ON/OFF)

SD_i is the shutdown cost of unit i

$X_i(t) = I_i(t).\left(1 - I_i(t-1)\right)$

$Y_i(t) = I_i(t-1)\,(1 - I_i(t))$

The main objective of the optimization is to minimize the OC subject to the constraints listed below:

1- Power balance constraint: The accumulation of power generated by all the committed DG units, RES and FDA must satisfy the DSO request at that particular interval and is characterized as:

$$\sum_{i=1}^{N} P_i(t).I_i(t) = P_D(t) - P_{VPP}(t)$$

where $P_D(t)$ is denoted as the total power that the aggregator promised the DSO to deliver. $P_{VPP}(t)$ is the total forecast power generated by the RESs and the total estimated curtailable loads offered by FDAs.

2- Generation limit constraint: For system stability and safe operation of each generating unit, the power output must not exceed its upper limit nor drop below a specified lower limit as follows:

$$P_i^{min} \leq P_i(t) \leq P_i^{max}$$

3- Power ensuring reserve constraint: Adequate PER should be required for stable and reliable operation. The PER constraint can be designed as:

$$\sum_{i=1}^{N} I_i(t)\, P_i^{PER}(t) = P_{VPP}^{PER}(t)$$

$$P_i^{PER}(t) = P_i^{max} - P_i(t)$$

where $P_i^{PER}(t)$ is the power ensuring reserve for the unit i at time t.

4.3.1 Priority Listing method

The PL method is proposed to solve the UC problem. The priority orders (PO) of the DG units are assigned on the basis of their full load average production cost which is defined as the cost per unit of power when the unit is at its maximum capacity.

$$PO_i = \frac{F_i(P_i^{max})}{P_i^{max}}$$

The units are arranged in ascending order according to their PO_i, the DG with lowest PO_i will have the highest priority to commit. The units are committed in order until the condition (29) is met:

$$\forall t \quad \sum_{i=1}^{N} P_i^{max}.I_i(t) \geq P_D(t) + P_{VPP}^{PER}(t)$$

The maximum power that can be generated by committed DG units is greater than or equal to the requested power plus the estimated balancing reserve power.

4.3.2 Lagrange method

The LM or lambda iteration method is one of the most popular traditional optimisation methods used to solve the ELD problem. The idea is to transfer the constrained optimisation problem into unconstrained using Lagrange multipliers, which can be expressed as:

$$L(x, \lambda) = f(x) + \sum \lambda_k \, g_k(x)$$

where $f(x)$ is the objective function OC, λ_k is Lagrange multiplier, and $g_k(x)$ the constrain and given by

$$g_k(P_i(t), I_i(t)) = P_D(t) - P_{VPP}(t) - \sum_{i=1}^{N} P_i(t).I_i(t)$$

Through iterative procedure, the method tends to find the optimal value of the multiplier, and subsequently find the minimum value of the objective function. The setting of the initial value of the Lagrange multiplier and updating it is significant to the optimality of the solution. The initial value of λ_k is computed as

$$\lambda_{initial} = \frac{P_D(t) - P_{VPP}(t) + \sum_{i=1}^{N} \frac{b_i}{2 * c_i}}{\sum_i^N \frac{1}{2 * c_i}}$$

The setpoints of the DG units at iteration k can be calculated as

$$P_i(t)_k = \frac{\lambda_k - b_i}{2 * c_i}$$

To enforce the upper and lower generation limits of the DG units the following expressions are applied

$$if \quad P_i(t)_k > P_i^{max} \quad \text{then} \quad P_i(t)_k = P_i^{max}$$

$$if \quad P_i(t)_k < P_i^{min} \quad \text{then} \quad P_i(t)_k = P_i^{min}$$

The iteration will stop if

$$\left| P_D(t) - P_{VPP}(t) - \sum_{i=1}^{N} P_i(t)_k \right| \leq \Delta_k$$

Otherwise update λ_k and start next iteration

$$\lambda_{k+1} = \lambda_k + \frac{\lambda_k - \lambda_{k-1}}{\sum_{i=1}^{N} P_i(t)_k - \sum_{i=1}^{N} P_i(t)_{k-1}} * \left| P_D(t) - P_{VPP}(t) - \sum_{i=1}^{N} P_i(t)_k \right|$$

Iteration procedure

1. Set initial values for $\lambda_{initial}$
2. Repeat
   2.1 For i = 1 to N
      2.1.1. Calculate $P_i(t)_k$
   2.2 Calculate ε
   2.3 Update λ_k
3. Until | ε | <= Δ_k

4.3.3 GenOpt optimisation program

GenOpt, a generic optimization program, has been developed to find optimal values of the design parameters (independent variables) that minimizes a user-supplied cost function, using a user-selected optimization algorithm. GenOpt has a library of various optimization algorithms that can be used to solve the UC and ELD problems in different ways. The GenOpt algorithms can be classified according to the problems they are able to solve into three sets. First set is the algorithms which are able to solve problems whose design parameters are continuous variables. Continuous variables can take on any value on the real line, possibly between lower and upper bounds. Second set is the algorithms which can solve problems with discrete variables. Discrete variables can take on only integer values. Third set of algorithms are which can solve problems whose design parameters are both continuous and discrete variables.

With optimisation engine, GenOpt continuous variables algorithms are used to solve the ELD problem, while discrete variables algorithms are used to solve UC problem. Algorithms that can solve problems with both continuous and discrete variables are used to solve UC and ELD as a one optimisation problem. The optimisation algorithms included in the GenOpt library are:

- Generalized Pattern Search algorithms (the Hooke-Jeeves and the Coordinate Search algorithm), which can be run using multiple starting points
- Particle Swarm Optimization algorithms (for continuous and/or discrete independent variables), with inertia weight or constriction coefficient and velocity clamping, and with a modification that constricts the continuous independent variables to a mesh to reduce computation time
- A hybrid global optimization algorithm that uses Particle Swarm Optimization for the global optimization, and Hooke-Jeeves for the local optimization
- Discrete Armijo Gradient algorithm
- Nelder and Mead's Simplex algorithm
- Golden Section and Fibonacci algorithms for one-dimensional minimization

Optimisation is the procedure which aims to find the optimal solutions to the objective function by modifying a set of independent variables. The optimization objective is to minimize the operating cost OC. The independent parameters of the UC are the control variables of the DG units $I_i(t)$ and the independent parameters of the ELD are the setpoints of the DG units $P_i(t)$. The setpoints parameters are constrained by a lower and upper bounds generation limits of the DG units using box constraints.

$$\forall t \quad -\infty \le P_i^{min} \le P_i(t) < P_i^{max} \le \infty \; for\; i \in \{1, \dots, n\}$$

The other constraints are enforced using penalty functions. Penalty functions add a positive term to the cost function if a constraint is violated and implement as:

$$\bar{f}(t) = OC(t) + \mu\left(\sum_{i=1}^{g} g_i(t) + \sum_{i=1}^{h} h_i(t)\right)$$

where μ is a monotonically increasing positive weighting factor and have to satisfy:

$$0 < \mu_0 < \cdots < \mu_i < \mu_{i+1}$$

$g(t)$ and $h(t)$ are given by:

$$g(t) = max\left(0, P_D(t) - P_{VPP}(t) - \sum_{i=1}^{N} P_i(t).I_i(t)\right)^2$$

$$h(t) = \left(\sum_{i=1}^{N} I_i(t)\big(P_i^{max} - P_i(t)\big) - P_{VPP}^{PER}(t)\right)^2$$

The optimization algorithm is then applied to the new function $\bar{f}(t)$.

4.4 User interactive interface

The key module of the optimisation tool is the user interactive interface, which interacts with both the UCaELD model and optimisation programs (GenOpt, PL and LM). The main objective of the user interactive interface is to create an algorithm selection framework that acts as a decision support system to help the operator to take the main advantages of the GenOpt program. The developed user interface benefits operators, including who does not have prior coding knowledge to successfully apply GenOpt, since the selection, settings, and parameters of the algorithms can be controlled and amended in a user-friendly environment. The interactive interface provides a flexible and extensible interface between simulation model and iterative optimisation programs. Such an approach helps the user to focus on the optimisation problem rather than the interfacing procedures and creating the initialisation, configuration, input, and output text files. It is worth mentioning that these files need to be amended whenever the algorithm or its parameters have been amended. This allows the user to try and test different optimisation algorithms and different parameters with almost no additional effort and gives better understanding of the model behaviour. Screen shoot of the developed user interface is shown in Figure 15.

The core idea of the user interactive interface is that at the beginning of each optimisation, the operator selects the optimisation program/programs which will be used for the optimisation, the available options are:

1- ***GenOpt - GenOpt*** in this option the GenOpt will be used to solve both the UC and ELD
2- ***PL - GenOpt*** in this option the PL is used to solve UC and GenOpt to solve ELD
3- ***GenOpt - LM*** in this option the GenOpt is used to solve UC and LM to solve ELD
4- ***PL - LM*** in this option the PL is used to solve UC and LM to solve ELD

Based on the option selected, the user interactive interface will provide the user with a list of the relevant optimisation algorithms out of those available in the selected program and allows easy modification of algorithm parameters. After the user selects the preferred optimization algorithm form this list recommended values of the algorithm parameters are displayed which then can be amended. The data needed by the decision making, optimisation engine and model are uploaded. This data includes generator data, historical and forecast of PV generation data, baseline and flexibility of the prosumers. Subsequently, the user interactive interface creates the model configuration file, optimisation program files and copy the data files to relevant folders. When GenOpt is used the configuration, initialisation, command and templet files are automatically generated.

When the start button is pressed, user interactive interface calls the optimisation program to start the optimisation. During the optimisation, the optimisation program generates the input parameters and sends it to the UCaELD model input file and then the optimisation program launches the model to evaluate the cost function. The optimisation program reads the cost function from the UCaELD output file once it has been evaluated. Based on the value of the output of the model the optimisation algorithm will determine the input parameters for the next run. The process is repeated iteratively and, in each iteration, the optimisation algorithm generates a new set of input parameters to the UCaELD model until a minimum of the cost function is found. At the end of the search, the results are stored on CSV files and displayed graphically on the user interface.

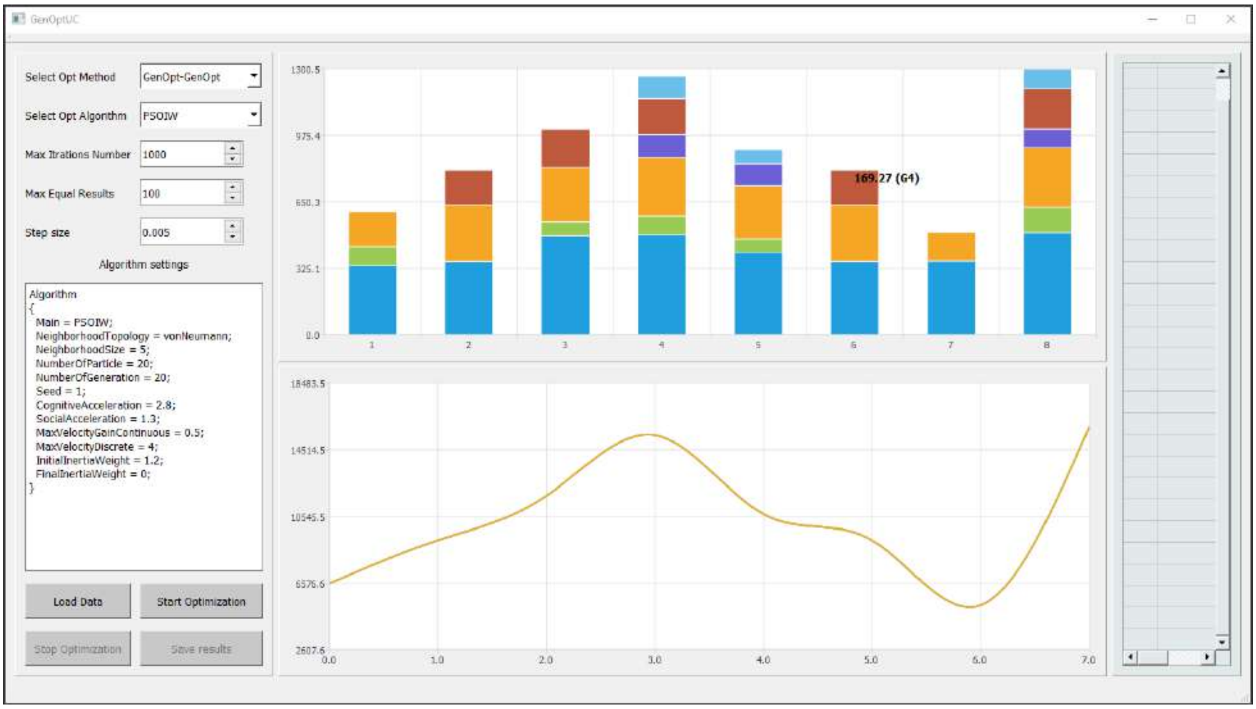


Figure 14 User interactive interface

The user interface can be described as a simple and complete interface to couple GenOpt optimization program with a simulation program as well as with other optimisation algorithms which are not included in the GenOpt library such as PL and LM. The user interface is presented in Figure 15. This simple interface enables the user to input all the information required to make the optimization. The default values of all control parameters are provided and they can be easily changed. The optimisation outputs are saved into output files for later processing, comparison, and evaluation. The interface consists of four main panels; selection panel for selecting the optimisation program and algorithm, algorithm control panel to adjust the algorithm parameters, DG panel to display the setpoints of all units stacked at each time slot, and cost panel displaying the total cost at each time slot. In addition to a few push buttons to load data required for the optimisation, start and stop the optimisation.

4.5 Integration procedure of the optimisation tool

The proposed optimization framework can be divided into three main layers: the lowest layer includes the UCaELD model and its configuration file, input and output files and data files. The optimization engine layer which includes the optimisation programs (GenOpt, PL and LM) and it is initialisation and configuration files. The optimisation settings and configuration layer which consists of the graphic user interface. The detailed structure and how data are exchanged between these layers for GenOpt program is shown in Figure 16.

The optimisation engine provides the user with a range of methods to address the problem of UC and ELD optimisation. The PL, LM and the GenOpt algorithms are integrated in such a way that the users can quickly and flexibly find solutions that are closer to Pareto optimum. These methods are briefly described in this section, detailed explanations of the algorithms can be found in the GenOpt manual [11].

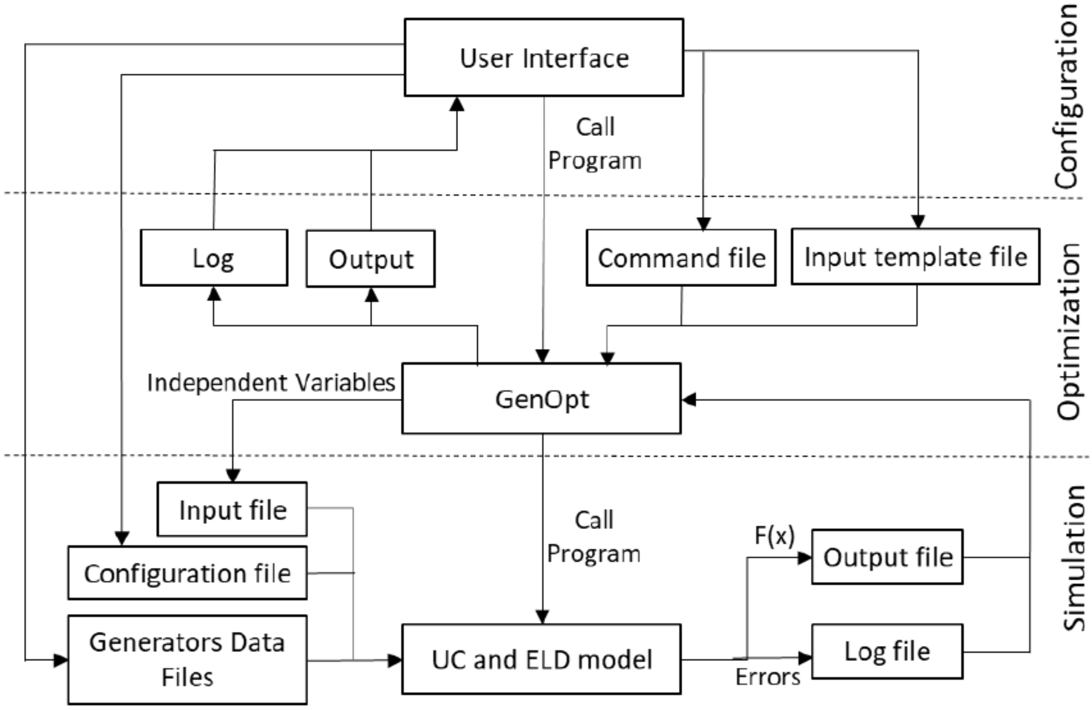


Figure 15 Optimisation tool framework

PL and GenOpt or ELD

In this configuration the PL method is used to evaluate the optimal schedule of the DG units and LM or any of the listed below GenOpt algorithms can be used to calculate the optimal set points of this units.

1- GPSCoordinateSearch presents the implementation of the Coordinate Search algorithm with adaptive precision function evaluations using the Model Generalized Pattern Search algorithm (GPS)
2- GPSHookeJeeve presents the implementation of the Hooke-Jeeves algorithm with adaptive precision function evaluations using the Model Generalized Pattern Search algorithm
3- Both algorithms GPSCoordinateSearch and GPSHookeJeeve can be run using multiple initial points. Using multiple initial points increases the chance of finding the global minimum and decreases the risk of not finding a minimum. To invoke the algorithms with multiple initial points the required parameters are made ready by the tool under a separate names GPSCoordinateSearchMS and GPSHookeJeeveMS
4- DiscreteArmijoGradient presents the implementation of the Discrete Armijo Gradient algorithm
5- PSOIW presents the implementation of the Model Particle Swarm Optimization (PSO) with Inertia Weight
6- PSOCC presents the implementation of the PSO Optimization with Constriction Coefficient
7- PSOCCMesh presents the implementation of the PSO Optimization on a Mesh
8- GPSPSOCCHJ hybrid Algorithm presents the implementation of the PSO and GPS algorithms
9- NelderMeadONeill presents the implementation of the Simplex algorithm

The interactions procedure between the PL algorithm and the other models of the toolset is shown in Figure 16. The PL model through the user interactive interface receives the preferred amount of the ensuring power reserve from DMS, total RES power generation, and total flexibility available. It also receives the generators data and the requested power to be generated by the VPP. Accordingly, the PL generates and sends the schedule of the units to the interface which will send it to the LM or one of the GenOpt algorithms listed above to calculate the optimal setpoints.

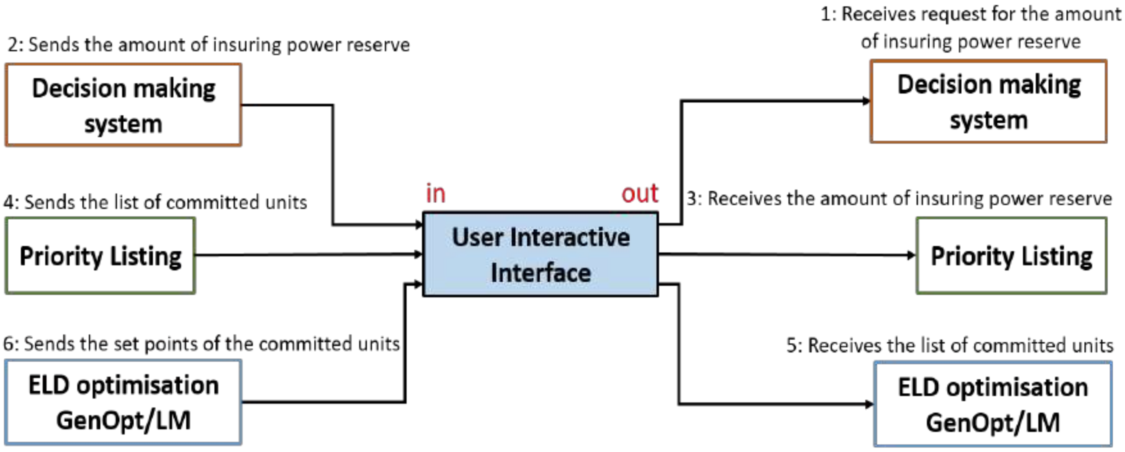


Figure 16 Interactions procedure of the PL method

GenOpt or PL and LM

When this option is selected, PL or one of the GenOpt algorithms are used to evaluate the optimal schedule of the DG units and to calculate the optimal set points of this units the LM is used. The GenOpt algorithms which can be used are:

1- PSOIW presents the implementation of the PSO with Inertia Weight
2- PSOCC presents the implementation of the PSO Optimization with Constriction Coefficient

The interactions procedure between the optimisation programs and other models of the toolset is shown in Figure 17. The user interactive interface sends to either PL or one of the GenOpt algorithms (according to the user selection) the desired amount of ensuring power reserve, total RES generation and total available flexibility obtained from the DMS to determine the optimal schedule of the DG units. The generated schedule is then sent to the LM to calculate the optimal setpoint of the selected DG units.

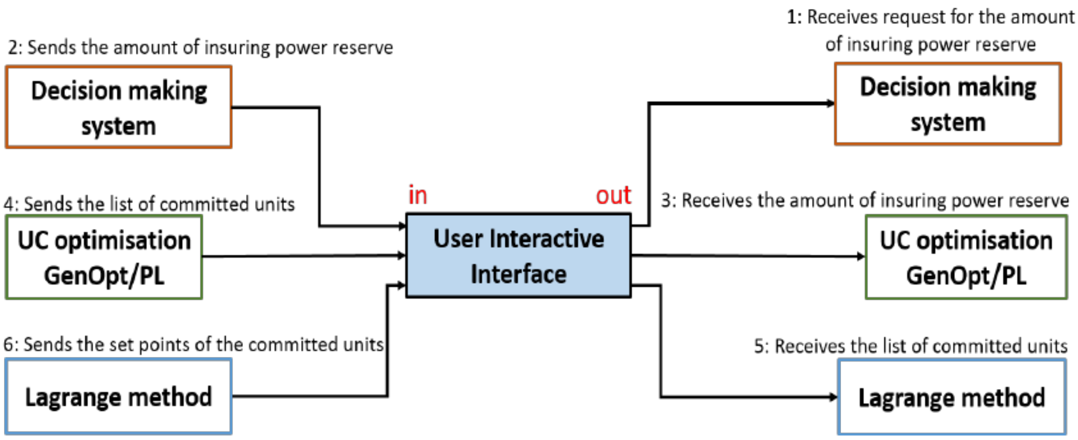


Figure 17 Interactions procedure of the LM method

GenOpt

The GenOpt program can be used to solve both UC and ELD problems. In this case the UC and ELD are addressed as one optimisation problem. Only the algorithms which can deal with continuous and integer dependent variables can be used, these algorithms are listed below:

1- PSOIW presents the implementation of the PSO Optimization with Inertia Weight

2- PSOCC presents the implementation of the PSO Optimization with Constriction Coefficient

3- PSOCCMesh presents the implementation of the PSO Optimization on a Mesh

4- GPSPSOCCHJ hybrid Algorithm presents the implementation of the PSO and GPS algorithms

4.6 Requirements for the component

To run the optimization toolset, GenOpt optimization program version 3.1.1 should be installed. GenOpt can be downloaded from https://simulationresearch.lbl.gov/GO/download.html. The requirements, guidance and installation instructions are also described in this link. The GenOpt version 3.1.1 is the current release when the optimisation toolset was developed, other versions have not been tested. GenOpt, needs to have Java Runtime Environment (JRE) v1.8.0 or higher version installed which can be downloaded from https://www.java.com/en/download/ or other sites.

In order to verify that GenOpt works, you can double-click on the executable GenOpt.jar contained in the GenOpt folder and try one of the included examples. If this is completed, you can create a UCandELD working directory in the GenOpt folder and copy to it the optimisation toolset files as shown in Figure 18. To run the optimisation toolset, you can double-click the executable UCandELD.exe contained in the UCandELD folder. The optimisation toolset software is developed and tested on Windows 10 with Microsoft Visual studio 2019 and Microsoft C++ compiler (MSVC). The graphic user interface is developed using Qt 5.12.6. In terms of hardware requirements, multi-core computer such as Core i7 or Intel Xeon are recommended, since GenOpt runs multiple simulation in parallel to minimise processing time.

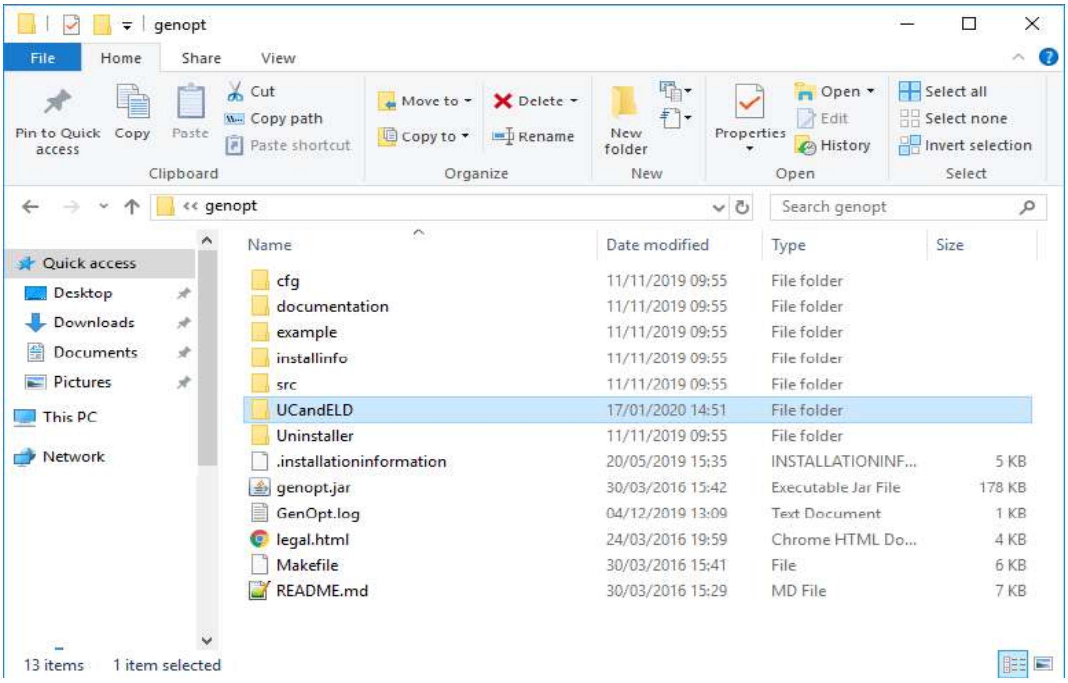


Figure 18 Optimisation toolset folder

To start the optimisation, the data listed below must be available:

- List of the prosumers participating on the VPP
- The data related to the DG units which includes: cost coefficients, start-up and shut down cost, and generation limits
- Past forecast and historical PV generation data of the prosumer's asset
- Flexibility of the prosumers participating on the VPP

4.7 Optimisation results

The optimisation engine is tested on 6 and 40 DG units test systems data. The results of different optimisation methods of the two systems are tabulated below.

In this test, the Particle Swarm algorithm with inertia weight (PSOIW) is used for PL-GenOpt and GenOpt-LM methods whereas the hybrid Algorithm (GPSPSOCCHJ) that presents the implementation of the PSO and GPS algorithms is used for GenOpt-GenOpt. The full capacity of the 40 DG test units is 12,722 MW and for 6 test units is 1476 MW. The power and cost reported in the Table 8 and 9 are on MW and $/MW respectively.

40 units system:

t	Demand	PER	PL-LM	PL-GenOpt	GenOpt-LM	GenOpt-GenOpt
0	400	0	3326.09	3326.1	3420.21	3804.55
1	560	0	4436.81	4436.84	4675.24	5387.82
2	690	0	5524.89	5526.16	5886.47	6664.26
3	800	45	6401.05	6419.15	6780.45	7934.05
4	1100	66	9009.25	9074.59	9483.18	10840.3
5	2300	80	20550.9	20760	20741.9	23778.6
6	4900	100	45931.4	46109.9	46160.1	50439.5
7	5633	150	53251.2	53354.2	53567.2	58315.9
8	7350	300	71587.2	71830.6	71893.5	81256.6
9	8800	366	87629.3	88144.9	88596.8	98617.6
10	9500	700	98587.3	100531	100113	107942
11	10500	600	113683	116707	116040	126113
12	12000	500	139824	154699	139824	166885
13	11100	340	123029	125338	123029	129094
14	9500	200	97132.1	98140.1	98075.2	105187
15	6714	180	64836.5	65185.5	65016.7	72958.2
16	7350	780	71833.6	72376.8	73452.7	80430.1
17	5633	730	53932.1	54262.3	54218.6	58315.9
18	3267	270	29883.4	30121.3	30268.5	32152.2
19	1370	197	13551.6	11636.7	12028.1	13056
20	884	56	7104.03	7148.54	7680.01	8486.68
21	542	0	4310.49	4310.51	4634.64	5261.54
22	476	0	3850.28	3850.33	4116.97	4357.97
23	302	0	2460.66	2460.73	2562.49	2709.02
	111671	**5660**	**1131666**	**1155750**	**1142265**	**1259988**

Table 8 Cost for 40 units test system

As seen from Table 8, there are virtually no differences between PL-LM and PL-GenOpt when power generation demand is low compared to full capacity of the DG units. For example, the outputs of both methods at 400 MW power demand are equal. However, as the power demand approaching higher levels, the differences in cost becomes more apparent. When the power demand was at 11100 MW, the cost calculated by PL-GenOpt increased by 1.87%.

When comparing the PL-LM with GenOpt-LM, the differences are non-significant when the demand for power generation is high compared with full capacity of the DG units, but as the demand for power generation reaches lower levels, the differences in cost become evident. As an example, the outputs of both methods at 11100 MW power demand are identical, but when the power demand decreases to 400 MW, the cost calculated by GenOpt-LM increased by 2.82%.

An increase in cost was observed in all methods as the PER increased. The output for the PL-LM, PL-GenOpt and GenOpt-LM increased by 1.26%, 1.67% and 1.2% respectively when the PER increased from 150 at t=7 to 730 at t=17 for the same power demand. A similar remark was also observed at t=8 and t=16.

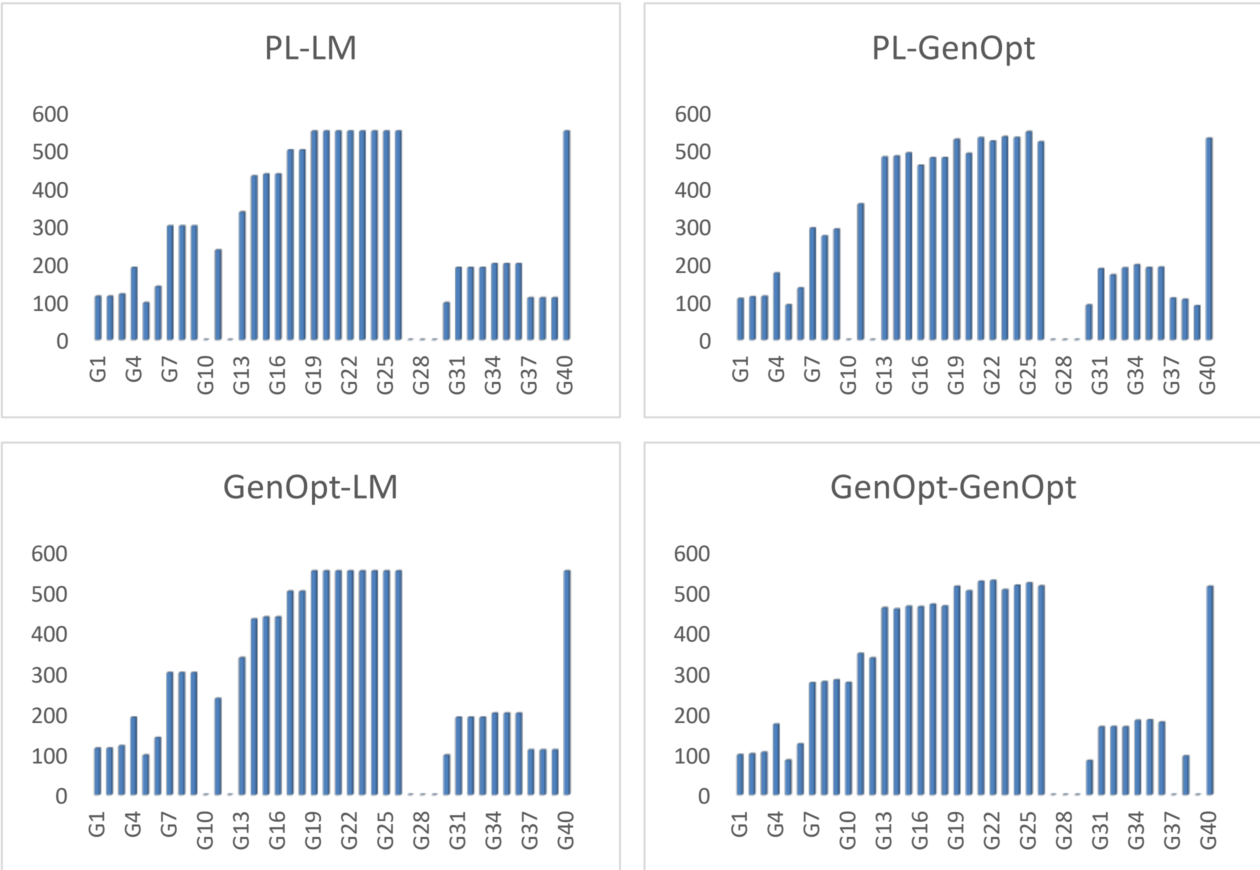


Figure 19 Dispatch schedule for the power demand =11100 MW

6 units system:

As can be seen from Table 9 the output of all methods is nearly equal at all power demand levels. Slight increase was observed when both UC and ELD were solved with the GenOpt. Table 10 shows the dispatchable schedule of the PL-LM method which is identical to the schedule obtained by PL-GenOpt and GenOpt-LM methods.

t	Demand	PER	PL-LM	PL-GenOpt	GenOpt-LM	GenOpt-GenOpt
1	177	30	1698.3	1698.3	1698.3	1698.3
2	507	70	5318.69	5318.71	5318.71	5366.33
3	650	150	7065	7065.05	7065.05	7232.81

4	800	100	8978.65	8980.55	8980.55	8982.6
5	989	250	11567.7	11570.1	11570.1	11594.3
6	1320	380	16035.9	16037	16037	16044.3
7	1450	20	17802.6	17803.1	17803.1	17803.3
8	939	255	10812.6	10817.9	10817.9	10979.9
9	776	250	8784.62	8785.76	8785.76	8793.02
10	1190	110	14171	14172.1	14172.1	14327.1
11	1320	50	16035.9	16036.3	16036.3	16038.2
12	355	80	3639.04	3639.09	3639.09	3748.38
	10473	**1745**	**121910**	**121923.96**	**121923.96**	**122608.54**

Table 9 Cost for 6 units test system

Table 10 shown that generator G1 is the lowest cost unit to run, while generator G6 is the highest also the generators tend to operate as close to its constraints, and only the generators that capable of generating the demand are committed. For instance, at t=1 the demand was 177 only the G1 is committed as it is the lowest to run and the maximum power that can generate is 500. At t=2 the demand was 650 the G1 and G3 were committed they were the lowest to run and the maximum power that they can generate is greater than 650.

t	G1	G2	G3	G4	G5	G6	Demand	Cost
1	177	0	0	0	0	0	177	1698.3
2	332.062	0	174.937	0	0	0	507	5318.69
3	412.5	0	237.5	0	0	0	650	7065
4	414.07	147.209	238.721	0	0	0	800	8978.65
5	401.018	137.592	228.569	89.6804	132.141	0	989	11567.7
6	457.882	179.492	272.797	133.908	181.897	94.0233	1320	16035.9
7	485.088	199.539	293.957	151	201	119.416	1450	17802.6
8	412.724	146.218	237.674	0	142.384	0	939	10812.6
9	364.636	110.785	200.273	0	100.307	0	776	8784.62
10	449.249	173.131	266.083	127.194	174.343	0	1190	14171
11	457.882	179.492	272.797	133.908	181.897	94.0233	1320	16035.9
12	246.562	0	108.437	0	0	0	355	3639.04

Table 10 Dispatch schedule for 6 units test system

In these experiments, different methods based on the GenOpt optimisation program to solve the UC and ELD problems have been tested. The findings obtained showed that all methods provide similar results for problem dealing with six units and similar size. For larger problems, the PL-LM method provided the best results and less computational time.

5 Conclusions

The report outlines the specifications and architecture of the developed Improved Decision-Making and DR Optimization toolset. A variety of optimization algorithms/strategies have been developed to support optimization needs of stakeholders (aggregators and DSOs). Three components have been developed and implemented; namely, short-term forecast of the PV production, long-term forecast of the PV production and optimisation and decision-making toolset. To improve the PV systems' short-term forecast, a novel near real-time trend analysis algorithm was developed using a Slope Statistic Profile method. To calculate the degradation rate and provide long-term output forecast of the PV system, a statistical method based on Classical Seasonal Decomposition algorithm was implemented. To develop Improved Decision-Making and DR Optimization toolset an optimisation engine based on GenOpt program has been developed and implemented. The engine provide access to several advanced optimisation algorithms and mechanisms.

The optimisation engine is a customised GenOpt-based optimization framework designed to solve the UC and ELD problem. A novel intuitive user interface compatible with GenOpt has been developed. The user interface provides a guidance on how to select the optimisation algorithms and to set their parameters in a user-friendly environment. The developed algorithm selection framework of the user interface acts as a decision support system to allow the aggregators to take the most advantages of the GenOpt program. Additionally, PL to solve UC and LM to solve ELD problems have been integrated with the user interface. The interface can be used as an optimisation algorithm development environment where more methods and algorithms can be implemented, tested and evaluated. With minimum effort the interface can be generalised and used to solve any other problems which can be modelled and coded as executable that reads its input from text files and writes its output to text files.

References

1. J. DeCesaro, K. Porter and M. Milligan,"Wind energy and power system operations: A review of wind integration studies to date" Electr. J, vol. 22 pp. 34-43 2009.
2. L. E. Jones,"Strategies and Decision Support Systems for Integrating Variable Energy Resources in Control Centres for Reliable Grid Operations" Washington DC USA: Alstom Grid Inc. 2011.
3. Alizadeh, M. I., Moghaddam, M. P., Amjady, N., Siano, P., and Sheikh-El-Eslami, M. K.,"Flexibility in future power systems with high renewable penetration: A review Renewable and Sustainable Energy" Reviews, 57, 1186-1193 2016.
4. Orwig, K. D., Ahlstrom, M. L., Banunarayanan, V., Sharp, J., Wilczak, J. M., Freedman, J. and Hodge, B. M., "Recent trends in variable generation forecasting and its value to the power system. IEEE Transactions on Sustainable Energy", 6(3), 924-933 2015.
5. T. Vafeiadis, S. Krinidis, C. Ziogou, D. Ioannidis, S. Voutetakis, D. Tzovaras, "Robust malfunction diagnosis in process industry time series", 14th IEEE International Conference on Industrial Informatics (INDIN'16), Futurescope, Poitiers, France, 18-21 July 2016, pp. 111-116.
6. Iain S., Richard G., "How does wind farm performance decline with age?", Renewable Energy, Volume 66, 2014, Pages 775-786, ISSN 0960-1481, https://doi.org/10.1016/j.renene.2013.10.041.
7. Jordan D.C., Kurtz S.R.,"Photovoltaic degradation rates-an analytical review", Prog Photovoltaic Res Appl 2013;21:12–29.
8. Jordan D.C., Kurtz, S.R., Vansant, K., Newmiller J., "Compendium of photovoltaic degradation rates", Prog. Photovoltaic. Res. Appl. 2016; 24:978–989.
9. Georgitsiotia T., Pearsall N., Forbes I., Pillai G., "A combined model for PV system lifetime energy prediction and annual energy assessment", Solar Energy 183 (2019) 738–744
10. Evaluation of Measurement Data—Guide to the Expression of Uncertainty in Measurement, International Bureau of Weights and Measures, BIPMJCGM-100:2008, Sevres, FR, Guide, 2008.
11. Michael Wetter "GenOpt Generic Optimization Program User Manual", Simulation Research Group, Lawrence Berkeley National Laboratory, June 2008.