FP7-SMARTCITIES-2013
Project number: 609132
http://www.e-balance-project.eu/

e-balance

Deliverable D5.2

Detailed specification, implementation and evaluation of energy balancing algorithms

Editor:	Juan Jacobo Peralta (CEMOSA)
Dissemination level: (Confidentiality)	PU
Suggested readers:	Power system, Smart-grids and computation experts
Version:	1.0
Total number of pages:	38
Keywords:	Algorithms, control, energy, grids, electricity, management

Abstract

This document contains the specifications of the energy balancing algorithms designed, implemented and validated to satisfy the requirements of the e-balance stakeholders, which will be deployed in the Bronsbergen demonstration site (the Netherlands) and in the virtual Lab demonstration (IHP premises). These algorithms are based on profile steering, which solves traditional problems that happen when systems and user deal with dynamic prices. In addition, in order to complement the inputs for the algorithms, prediction models for energy consumption and production at neighbourhood and home level have been developed, though its refinement will continue until the final demonstration to improve the whole performance. Finally, a description of the beta version of the GUI for customers is included in this document to illustrate the interaction between users and the balancing mechanisms.

Disclaimer

The information, documentation and figures available in this deliverable are written by the e-balance partners under EC co-financing (project number: 609132) and does not necessarily reflect the view of the European Commission.

Impressum

[Full project title] Balancing energy production and consumption in energy efficient smart neighbourhoods

[Short project title] e-balance

[Number and title of work-package] WP5 Energy Management Platform
[Document title] Detailed specification, implementation and evaluation of energy balancing algorithms
[Editor: Name, company] Juan Jacobo Peralta, CEMOSA

[Work-package leader: Name, company] Marco Gerards, University of Twente

Copyright notice

Executive Summary

This deliverable presents the specification, implementation and validation of the balancing algorithms to satisfy the requirements of the stakeholders related to the energy production, consumption of management in future smart grids. The structure of this deliverable follows closely the energy models described in the context of the project and the e-balance architecture from the required inputs as prediction and users' preferences to the internal calculation and negotiation between management units. The balancing algorithms aim at coordinating all the management units that must control the energy flows in order to make the whole electric grid stable automatically even when many distributed energy sources and storage appliances (electric vehicles and batteries) can be present.

The energy balancing algorithms presented in this document are based on the concept of "power profile steering", which solves the traditional problems of systems or platforms based on dynamic prices. However, this new approach requires accurate consumption and production predictions for all levels, from the consumer to the DSOs. The Chapter 2 depicts the prediction models designed and implemented to make possible the prediction in every management unit, especially in the CMU and LVGMU. The prediction models are based on artificial neural networks (ANN) with an intensive assessment of all the energy consumption and production data collected by Liander company from 80 houses of the Netherlands during 2 years. This amount of data has been especially important to design the neural network structure and select the representative inputs for reducing the difference between the prediction and the real data. Further details are shown graphically in the corresponding chapter.

In Chapter 3, the final user's inputs and outputs are driven through a graphical user interface (GUI) designed on purpose for this project. As energy prediction models, the GUI will provide a friendly interface with the basic information that algorithms requires to operate successfully and to perform the management unit negotiation. The design of the GUI is an on-going activity that is based on social studies, business models and the technical architecture of e-balance. The GUI architecture has been designed following specifically the requirements defined in the deliverable D2.4 concerning user-friendly criteria, security, basic informative graphs and easy information, etc. The beta version of the GUI is currently running in the UMA premises under a web server installed in the HomeWaveControl (by LessWire) and a Beaglebone, and the frontend has been developed using HTML, CSS and JavaScript.

A detailed description of the energy balancing algorithms and the corresponding libraries constitute the Chapter 4 of this document. The steering signal concept is based on a desired profile to each house, which is a vector that indicates the desired power for each 15 minute interval for one day ahead (i.e. a vector of 96 power values). Each house (i.e. the management unit) aims at minimising the Euclidean distance between its own planned profile and the desired profile, and then the negotiation process continue to upper levels both the system architecture and electric grid. This set of algorithms have been design, implemented into C libraries and validated by the University of Twente in simulators.

In Chapter 5, a complementary algorithm called "power limitation algorithm" has been created to support the energy balancing out of normal operation, i.e. the system is approaching to energy supply limitation and the interaction of prosumers' management units is mandatory to keep the entire system operative until restoration tasks can turn the system into normal mode. This algorithm presents a simple logic that disconnect the users' appliances, according to order or priority level established by the user, when the corresponding management unit receives the power limitation signal until the maximum power is below the requested limit. Due to the simplicity of this algorithm, it has been tested using a visual basic routine.

Finally, in Chapter 6, some recommendations are introduced as guidelines for next steps within the project. The prediction models must be refined to guarantee the energy balancing algorithms' outputs are accurate. The GUI is still in a preliminary version (beta) that has to be improved or modified in order to satisfy all the system requirements and gather all the partners' point of view. In addition it is highly recommended to check the interaction of the energy balancing algorithms with the communication platform and security mechanisms when a substantial change of modules happens and to check that all the variables and parameters are measured strictly with the same magnitude, origin, timestamp, etc.

List of authors

Company	Author
CEMOSA	Juan Jacobo Peralta Noemi Jiménez Redondo
UTWE	Marco Gerards
UMA	Jaime Chen Eduardo Cañete Daniel Garrido
IPI	Pawel Kobylinski

Table of Contents

List of Tables

List of Figures

Abbreviations

ANN	Artificial Neural Network
CMU	Consumer Management Unit
CPU	Central Processing Unit
CSS	Cascade Style Sheets
DSM	Demand Side Management
DSO	Distribution System Operator
EV	Electric vehicle
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
JS	Java Script
JSON	Java Script Object Notation
LVGMU	Low Voltage Grid Management Unit
MAE	Mean Absolute Error
MLP	Multi-Layer Perceptron
MVGMU	Medium Voltage Grid Management Unit
PC	Personal Computer
PL-a	Power limitation algorithm
PV	Photovoltaic panel
REST	Representational State Transfer
SNNS	Stuttgart Neural Network Simulator
TCP	Transmission Control Protocol
TLGMU	Top Level Grid Management Unit
TSO	Transmission System Operator
WP	Work Package

1 Introduction

This document presents the description and the implementation details of the energy balancing mechanisms proposed by the project. These mechanisms aim at balancing the production and consumption units within every electric grid level for future smart grids, excepting the transmission grid, in a different way of the traditional electric system as it works nowadays and taking advantage of the smart appliance that allows the remote control thereof. In this way, the distributed production (e.g. PV), storage (e.g. batteries and EV) and the smart appliances are managed to optimise the power flow and to increase in turn the robustness and resilience of the distribution grid, since prosumers' energy flows can be adjusted to achieve minimum costs, the grid capacity can be released to reduce energy losses, and unexpected problems like energy supply limitations can be mitigated with a bottom-up strategy by limiting the energy from prosumers to secondary substations.

Under this smart grid approach, which the e-balance project is dealing with, every management unit must guarantee that the electric energy is balanced downwards according to the technical, legal and business restrictions (see deliverables D2.2, D2.3 and D3.2) and orders from upper management units upwards in the hierarchy. This distributed architecture requires that the energy balancing algorithms should be integrated in the whole system and their modules installed in every management unit, since the processes and routines will run through different units in every calculation time. For this reason, most of the algorithms presented in this document are related to the electric grid and e-balance architecture levels, what constitutes the first step for the integration in the energy platform together with the algorithms from task 5.3 (resilience algorithms).

One of the main issues of the energy balancing algorithms is that they must be supported by a consistent and accurate energy consumption and production models, since they are based on the 24-hour ahead power profile negotiation. The new appliances as PV, electric vehicles, small wind turbines, etc., insert more uncertainty sources within the electric grid that make more complex the prediction of power profiles at neighbour level. Furthermore, when we are dealing with prediction at home level the human behaviour is unpredictable. In order to approach the energy prediction as much as possible to the reality, the partners involved in this activity decided to use a prediction model based on artificial neural networks that, though cannot solve the problem of human behaviour, offers a practical solution for neighbourhood level to allow the system to work automatically. The design and implementation details are described in Chapter 2.

At the prosumers' premises, the project consortium identified previously (D2.4) that the final users will require some means to interact with the new system and make decisions concerning the performance of their appliances in order to obtain the best energy prices and comfort, whilst the better engagement of energy efficiency may increase the environmental awareness. This "wish list" must be facilitated by a graphical user interface (GUI) designed according to social studies, business framework and technical requirements, in order to assure the acceptance of final users and its functionalities are effective to run the energy balancing algorithms. This interface can be installed in smartphones, tablet PC, etc. since it is based on WebServices and the privacy and security system developed in task T5.4. Through this interface, the user will control the energy consumption, will decide the priority of their devices and will be able to compare the energy behaviour with respect to other consumers and has access to other functionalities described in the Chapter 3.

As mentioned above, the baseline of energy balancing algorithms is the negotiation through profile steering. Smart grid studies usually research new technologies and solutions on the basis of dynamic prices. Studies carried out by the University of Twente demonstrate that the profile steering is more effective and avoids additional iterative problems as new energy peaks or the displacement over the time of such effect. In addition, this change of paradigm is totally compatible with the price signals and can run automatically under the e-balance approach. The details and results of these algorithms are depicted in the Chapter 4.

Finally, other complementary mechanisms are required to operate the new system proposed by the project and satisfy requirements from different uses cases (see D2.1). In this case, related to energy balancing, the power limitation algorithm has been designed and implemented in order to manage the power supply limitation scenario, which is a normal situation both for connected electric grids and for micro-grids working on isolation mode. Complete description and details are shown in the Chapter 6.

1.1 Deliverable position in the project

Figure 1 shows the position of this deliverable within the e-balance project. This deliverable is part of work package WP5 (Energy Management Platform). This document provides the more detailed specification of the energy balancing mechanisms based on the energy models depicted in the deliverable D5.1 and on the corresponding prediction models, which are required as inputs thereof. In addition, this document describes the graphical user interface implementation for users based on the results obtained in work package WP2 (Use cases and socio economic aspects) especially regarding business models and social studies.

The algorithms directly influence but also gather from the work packages WP3, WP4 and WP6. The energy balancing algorithms allow implementing one of the basic modules of the energy platform and these participate closely with the energy resilience mechanisms sharing both inputs and outputs to perform required calculation. In summary, the implementation described in this document is one of the first implementation steps for the entire energy platform.

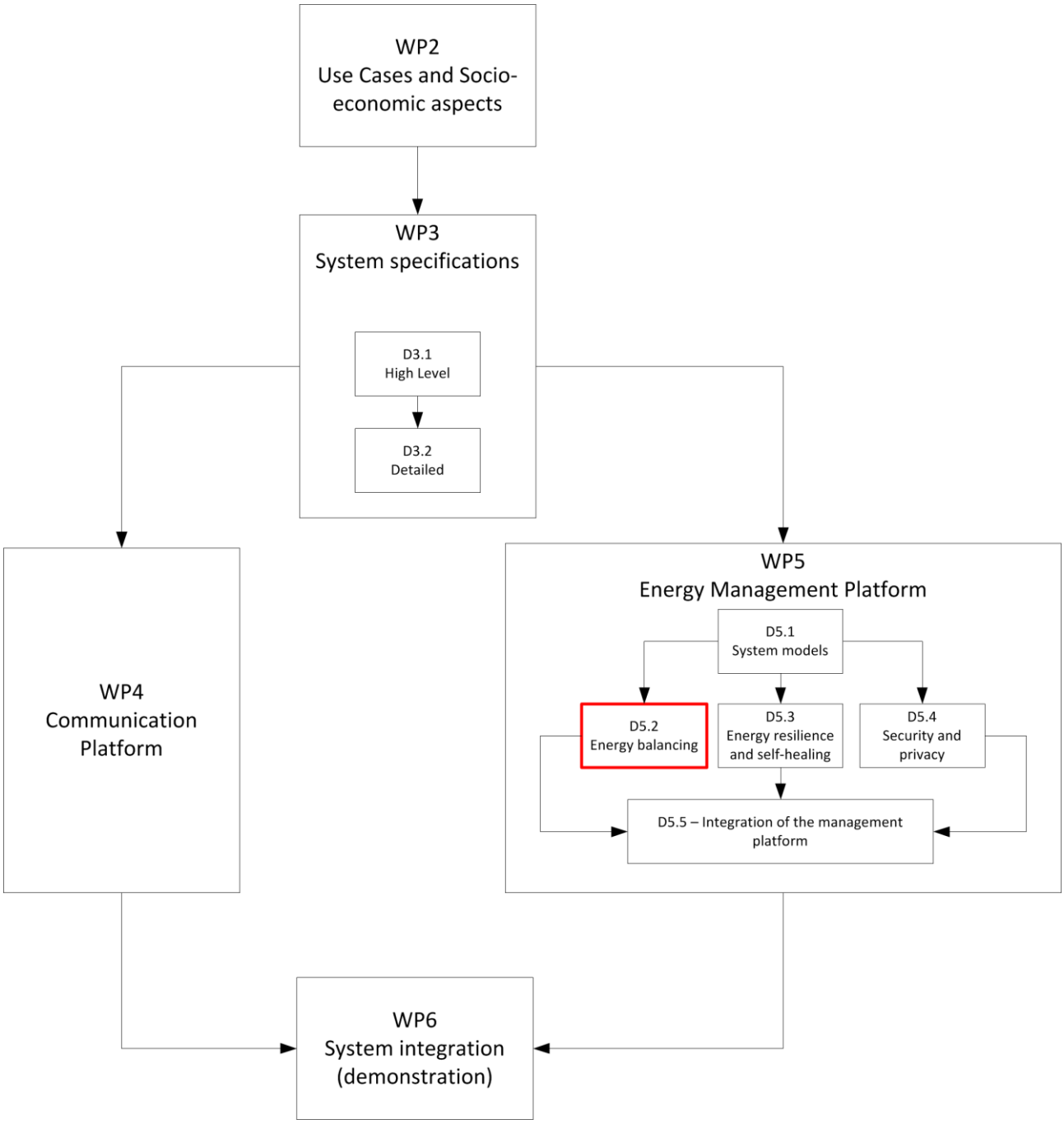

Figure 1: The position of the deliverable D5.2 within the e-balance project work package structure

2 Description, implementation and validation of prediction models.

The e-balance system design involves a module for prediction of energy load 24 hours ahead. We have prepared a fully working prediction model.

2.1 Description of the prediction model

2.1.1 Definition of the prediction problem

A list of energy load related variables was identified:

$\mathbf{E_{fg}}$ denotes energy withdrawn from grid to a household (non-negative)

$\mathbf{E_{tg}}$ denotes energy fed to grid from a household - (non-negative)

$\mathbf{E_{u}}$ denotes energy used by a household (non-negative)

$\mathbf{E_{rn}}$ denotes energy generated in a household from renewable resources (non-negative)

$\mathbf{E_{s}}$ denotes energy fed to a household battery (positive, zero, or negative in case of withdrawal)

$\mathbf{E_{l}}$ denotes energy lost in a household (non-negative)

Based on this identification $\mathbf{E_{balance}}$ variable was defined:

$$\mathbf{E_{balance} = E_{fg} - E_{tg} = E_{u} - E_{rn} + E_{s} + E_{l}} \qquad \textbf{(Eq.1)}$$

$\mathbf{E_{balance}}$ denotes net energy load of a household

In concordance with **Eq.1**:

$\mathbf{E_{balance} > 0}$ denotes net energy consumption of a household

$\mathbf{E_{balance} < 0}$ denotes net energy production of a household

Per analogism, all the variables described above can be easily identified at neighbourhood level. At this level:

$\mathbf{E_{balance}}$ denotes net energy load of a neighbourhood

$\mathbf{E_{balance}}$ is equal to the sum of net energy loads from **n** households constituting a neighbourhood

$\mathbf{E_{balance} > 0}$ denotes net energy consumption of a neighbourhood

$\mathbf{E_{balance} < 0}$ denotes net energy production of a neighbourhood

The $\mathbf{E_{balance}}$ variable, both at household and neighbourhood level was subject to prediction modeling.

To some extent the definition of the prediction problem was predetermined by energy load data available at the moment Høverstad [1]. We have used an Open Data database obtained from Liander company. This database contained a set of time series, each representing $\mathbf{E_{balance}}$ values:

- collected from 75 households
- in 15-minutes intervals
- from 2013-01-01 00:00 to 2013-12-31 23:45
- measured in Wh

Irradiation was available as additional time series variable

- collected in 1 hour intervals
- from 2013-01-01 00:00 to 2013-12-31 23:45
- measured in J/cm^2/h

Thus the prediction model had to be built around 15 minute time intervals. The prediction problem implied estimation of a future (24 hours = 96 of 15-minutes intervals ahead) $\mathbf{E_{balance}}$ value based primarily on several measurements of $\mathbf{E_{balance}}$ values: one located at present interval t_0 and the remaining measurements located backwards in the past. This situation is depicted in Figure 2. The number of the predictor measurements (8) and the number of 15-minutes intervals between consecutive measurements (18) are justified in the subsection 2.1.2.

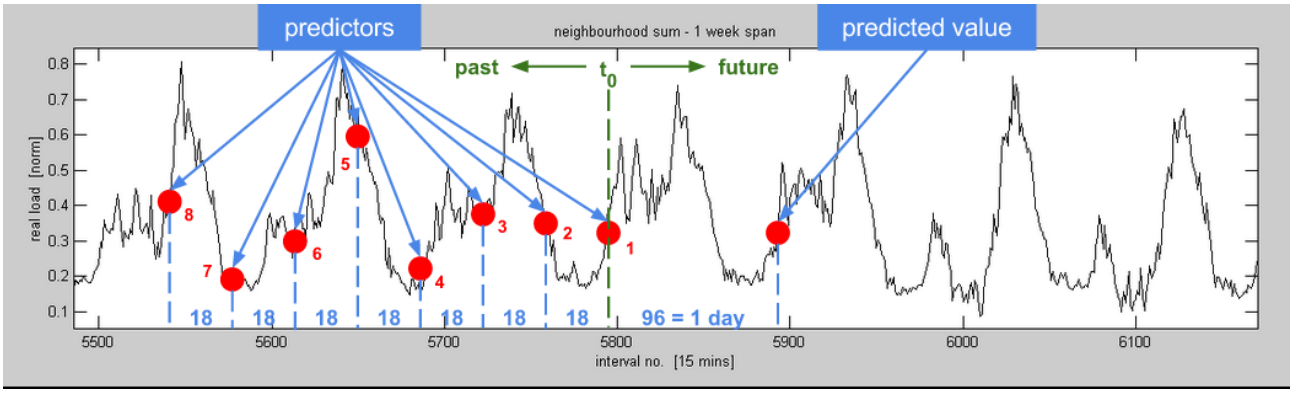

Figure 2: Structure of the energy load prediction problem

2.1.2 Model description

An Artificial Neural Network (ANN) - a Multilayer Perceptron (MLP) was chosen as the tool for the energy load predictions (Bishop [2], Hernández [3], Kandananond [4], Osowski [5]).

Decisions had to be made in relation to number and structure of the ANN inputs. First of all we employed a heuristic to establish how many $\mathbf{E_{balance}}$ measurements to include as predictors of the 24-hour ahead value. The employed heuristic consists of three main steps:

1. Treat the time series as a realisation of a dynamical system.
2. Identify the fractal[1] dimension (**D**) of the dynamical system
3. Set the number of predictor inputs **n = at least 2 * D + 1**

Using correlation integral method we have determined **D = 3** (both at neighbourhood level and on average for the 75 households) and decided for **n = 2 * D + 2 = 8** inputs (Jian-Kai [6] and Theiler [7]).

[1] A fractal is a natural phenomenon or a mathematical set that exhibits a repeating pattern that displays at every scale.

Another heuristic had to be employed in order to establish the number of 15 minutes intervals between consecutives measurements of load values – the so called optimal lag. The heuristic involves two steps (Mackey [8] and Theiler [7]):

1. Determine **n** autocorrelation measures of the original load signal and **n** lagged signals, where the consecutive lags rise from **1** to **n** 15-minutes intervals.
2. Find the lag at which the autocorrelation measure treated as a function of the number of lags has its first minimum.

Using normalized mutual information index as an autocorrelation measure, we have determined **18** 15-minutes intervals as the optimal lag - both at neighbourhood level and on average for the 75 households (Figure 3) and the lower graph takes into account the whole scale of normalized mutual information index.

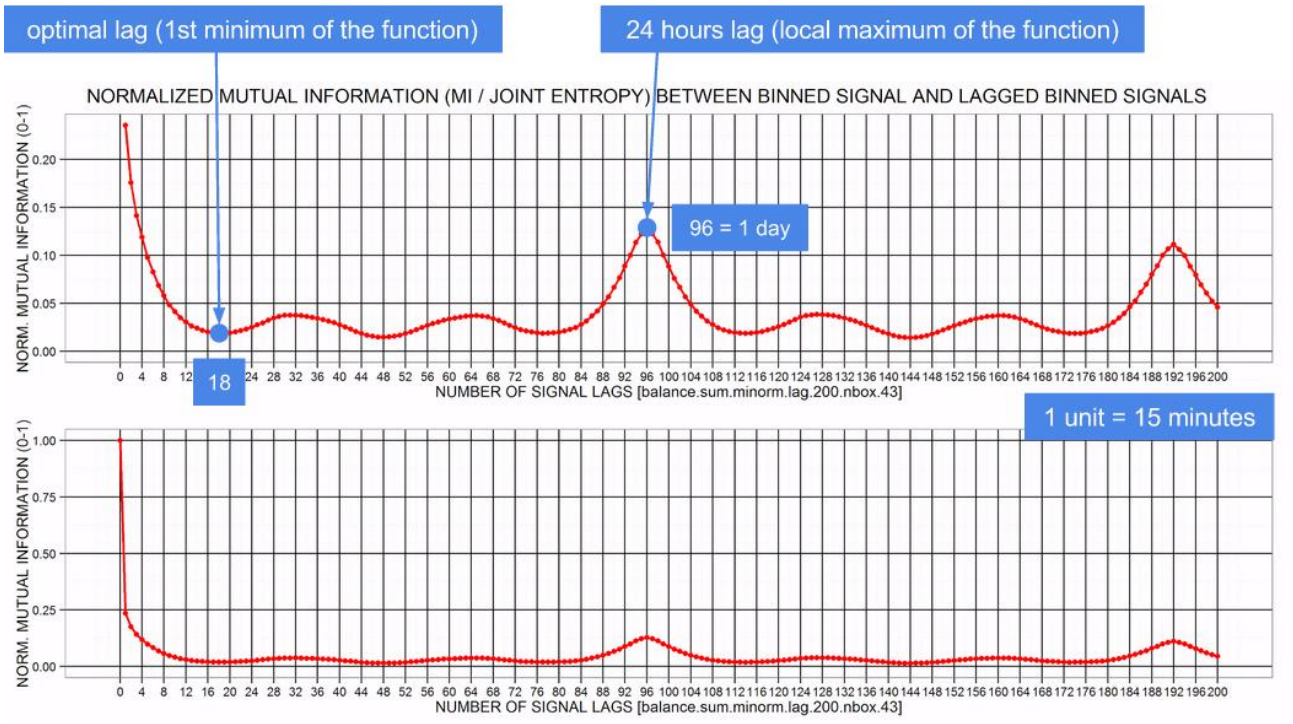

Figure 3: Determination of the optimal lag (the optimal number of 15-minutes intervals between consecutive measurements constituting ANN inputs) at neighbourhood level

Apart from the eight endogenous $\mathbf{E_{balance}}$ inputs (represented by red dots in Figure 4) additional ANN inputs were included into the model to account for exogenous factors such as:

- weather conditions
- natural cycles
- calendar days

Irradiation was the only weather related variable available in the obtained database, thus only one weather related ANN input was designed (represented by a yellow dot in Figure 4). Four inputs were included to account for natural cycles:

- two inputs to reflect the circular character of yearly changes in nature and human activity (represented by green dots in Figure 4)
- two inputs to reflect the circular character of daily changes in nature and human activity (represented by blue dots in Figure 4)

Ten inputs were included to account for the variability of human activity due to calendar days:

- Seven Boolean inputs were designed to pass information about specific weekdays (violet dots in Figure 4 representing inputs for Mondays, Tuesdays, Wednesdays, Thursdays, Fridays, Satrurdays, and Sundays)
- Three Boolean inputs (represented by pink dots in Figure 4) were designed to pass information about public holidays: one input for one-day holidays, one input for first days of two-day holidays, and one input for second days of two-day holidays.

Figure 4 summarises the architecture of the employed Multilayer Perceptron (MLP):

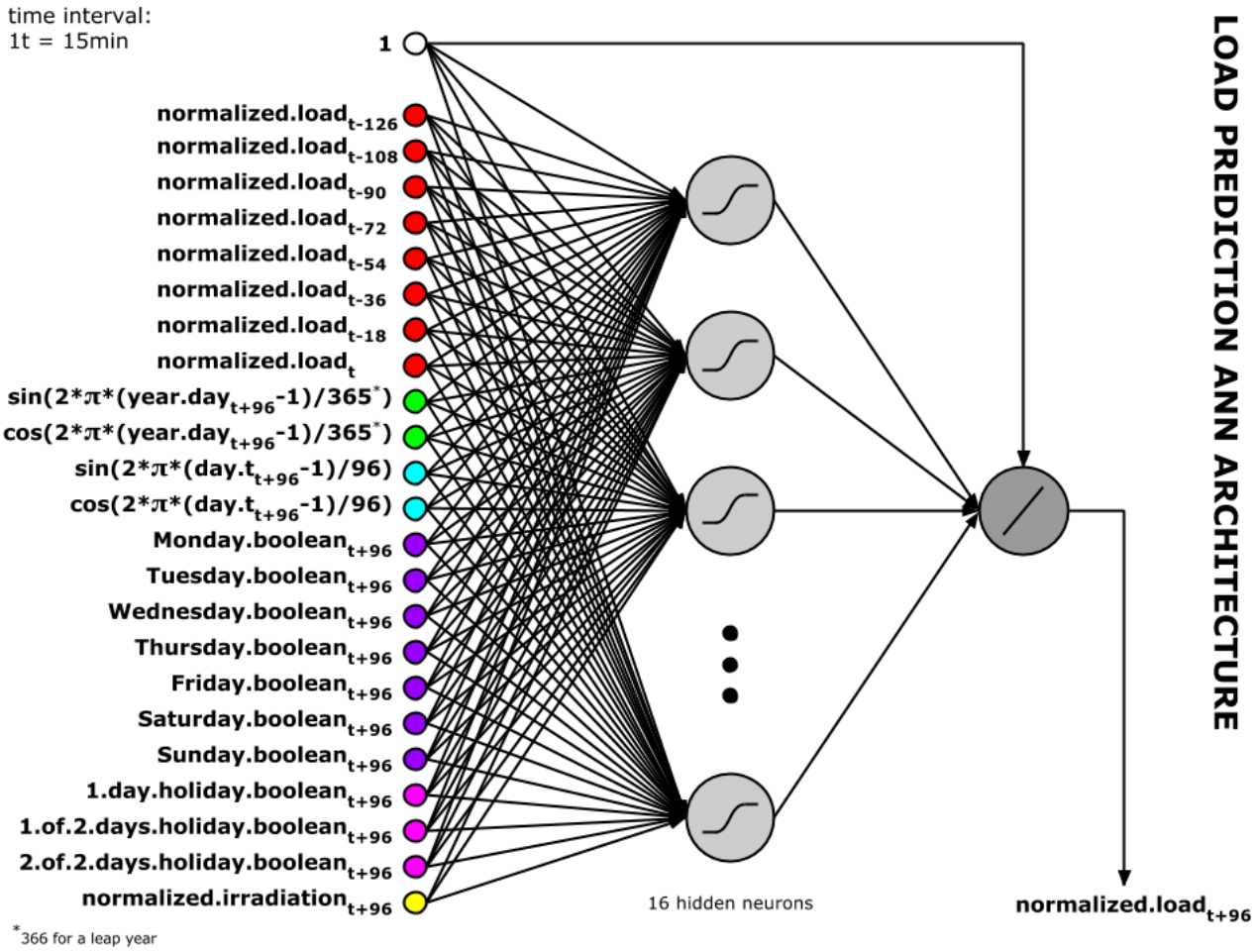

Figure 4: Architecture of the load prediction Artificial Neural Network. The ANN outputs a prediction of a normalised energy load value located 24 hours (96 quarters) ahead of a real load measurement located at any reference time point

Eight load measurement inputs (red dots) as well as the output were designed for normalised $\mathbf{E_{balance}}$ values. We decided for a normalisation formula which leaves the 0 value unchanged, positive $\mathbf{E_{balance}}$ values denote net energy consumption and negative $\mathbf{E_{balance}}$ values denotes net energy production.

The formula beneath defines the normalisation:

$$\mathbf{E_{balance}^{NORM}(t) = E_{balance}(t) / max(|E_{balance}|)} \qquad \mathbf{(Eq.2)}$$

The irradiation input was designed for analogically normalised values.

The finite MLP architecture included one hidden layer of 16 neurons. This number of the hidden neurons was established experimentally in the model testing phase.

2.2 Implementation of the prediction model

A software implementation of the described prediction model was conducted in order to test and tune it.

2.2.1 Software implementation

Tools used for the implantation of the prediction model included:

- R programming language (R Core Team [9])
- R package RSNNS for neural networks (Bergmeir [10])
- several other R packages used for data pre-processing, signal analysis, results analysis and results visualisation
- our own R code wrapped over the RSNNS package
- our own R code for data pre-processing, signal analysis, results analysis and results visualisation
- MATLAB (MathWorks [11])
- MATLAB Neural Network Toolbox (MathWorks [11])
- our own MATLAB code wrapped over the Neural Network Toolbox
- our own MATLAB code for data pre-processing, signal analysis, results analysis and results visualisation

The RSNNS R package mentioned above provides an R interface to the Stuttgart Neural Network Simulator – SNNS (Zell et al [12]). The SNNS is an established C library for designing, training, testing and implementing Artificial Neural Networks. It was the SNNS kernel that constituted the very engine of our prediction model implementation, MATLAB being an additional environment used mainly for cross-tool check of the results obtained via the R interface. Our decision to employ the SNNS kernel was dictated not only by its acclaimed quality, but also by the possibility to integrate the same C kernel into the actual prediction modules. Thus the model implementation can be viewed as a fully working prototype of the e-balance prediction module.

2.2.2 In-depth parameters of the ANN implementation

Several in-depth characteristics of the implementation are listed beneath:

- hyperbolic tangent activation function was chosen for hidden layer neurons
- linear activation function was chosen for the output neuron
- error backpropagation learning algorithm was chosen for the ANN training
- second order Levenberg-Marquard method was chosen for error backpropagation algorithm

2.3 Validation of the prediction model

Mean Absolute Error (MAE) of model fit and Mean Absolute Error of prediction were used as Key Perfomance Indices for the prediction model specified above. The employed formula (**Eq.2**) for normalisation of the $\mathbf{E_{balance}}$ energy load values implies a natural interpretation of the MAE in our case: it is the average absolute difference between real and fitted/predicted values denoted as a fraction (or a percentage if multiplied by 100%) of the maximal absolute real load.

2.3.1 Model Fit Analysis

Figure 5 illustrates detailed distribution of fit MAE for ANN trained with whole year data at the neighbourhood level. The average fit MAE was no larger than 0.044 (4.4%). The very bottom row illustrates the margin distribution of fit MAE as function of time in day. The very left column illustrates the margin distribution of fit MAE as function of day in week. The bootm-left tile informs on the average MAE index. The average fit MAE was no larger than 0.044 (4.4%).

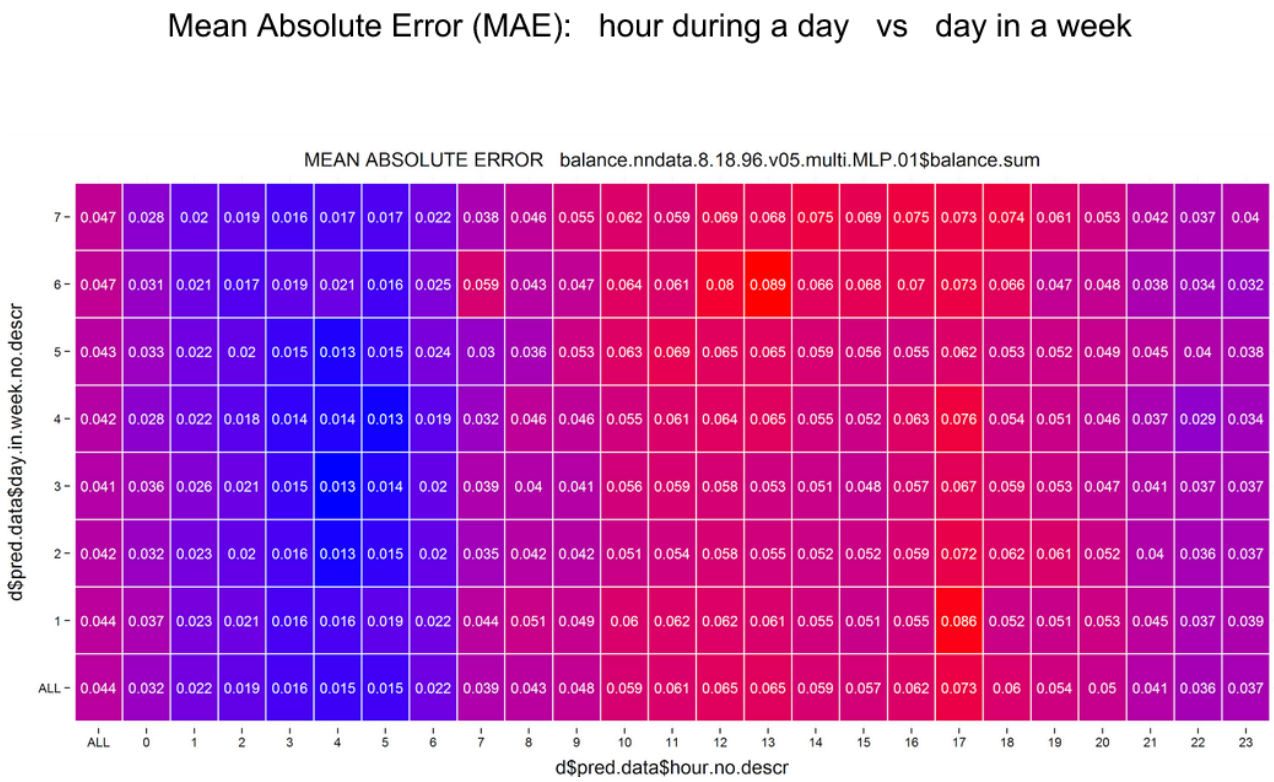

Figure 5: Distribution of fit MAE for ANN trained with whole year data at the neighbourhood level as a function of time in day and day in week

Figure 6 illustrates detailed distribution of (inverted) fit error as a fuction of the normalised energy load - for ANN trained with whole year data at the neighbourhood level. As can be seen the fit error distributes evenly throghout the majority of energy load values, however the ANN tends to slightly overestimate the output for low energy load values and slightly underestimate the output for high energy load values.

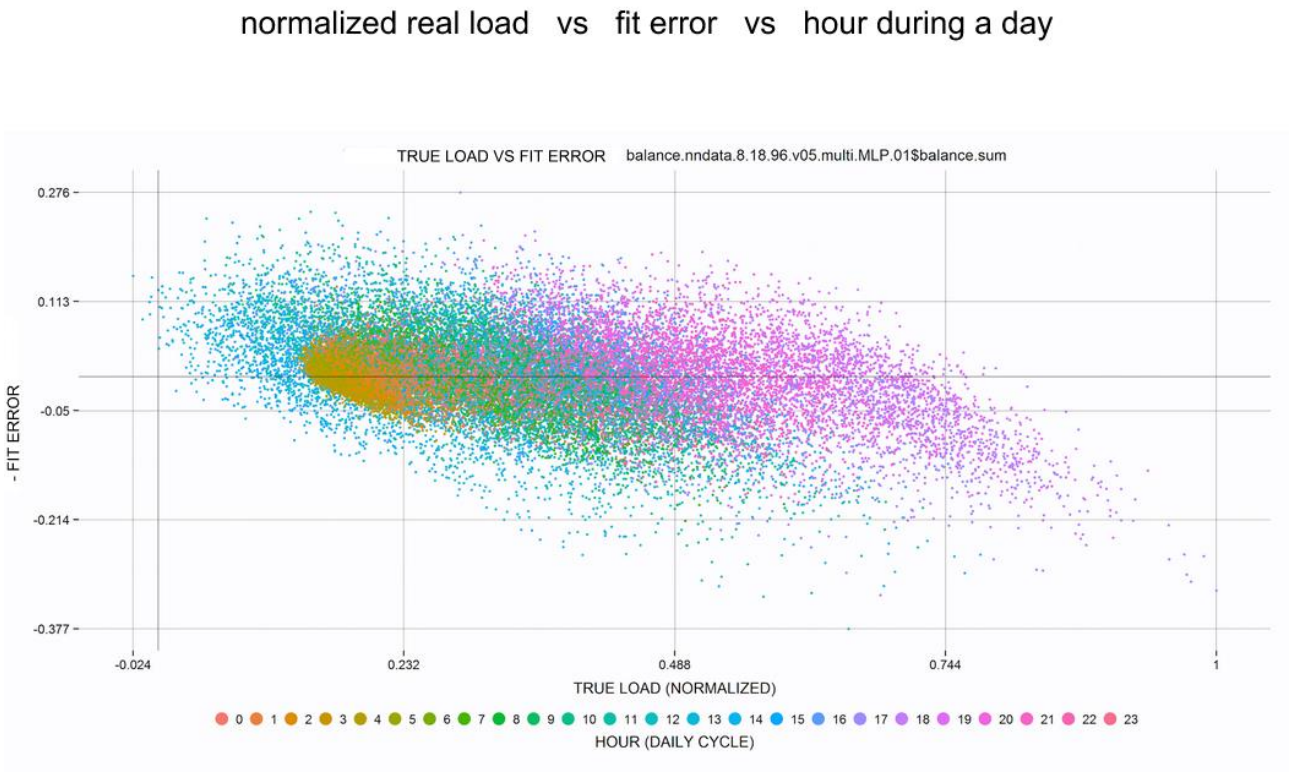

Figure 6: Distribution of (inverted) fit error for ANN trained with whole year data at the neighbourhood level as a function of the normalised energy load

Figure 7 illustrates detailed distribution of fit MAE for ANN trained with whole year data obtained from a typical energy-generating household. The average fit MAE was no larger than 0.076 (7.6%). The very bottom row illustrates the margin distribution of fit MAE as function of time in day. The very left column illustrates the margin distribution of fit MAE as function of day in week. The bottom-left tile informs on the average MAE index. The average fit MAE was no larger than 0.076 (7.6%).

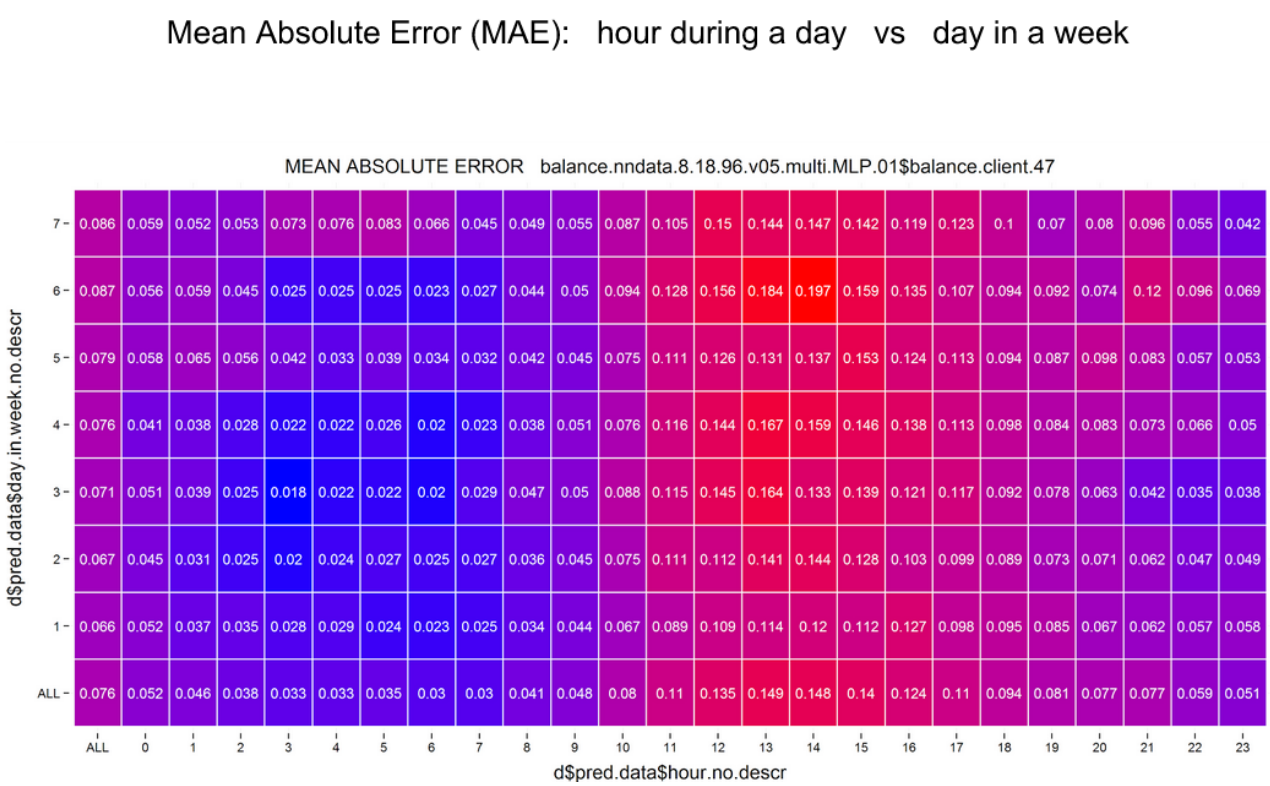

Figure 7: Distribution of fit MAE for ANN trained with whole year data obtained from a typical energy generating household - as a function of time in day and day in week

Figure 8 illustrates detailed distribution of (inverted) fit error as a function of the normalised energy load - for ANN trained with whole year data obtained for the same household. On the net consumption (right) side of the graph, especially during late afternoon and evening hours, a nearly linear relation between the load and fit error is visible – the ANN overestimates slightly the output for low energy load values, yet heavily underestimates the output for rare high energy load spikes. We atribute this to the unpredictability of the precise timing of individual human actions, for example turning on anf off a vacuum cleaner, a hair dryer or an electric kettle. At the household level the ANN treats rare and sudden energy consumption spikes as noise. On the net production (left) side of the graph a tendency to underestimate high energy generation (overestimate highly negative load) as well as overestimate the low generation (underestimate slightly negative load) can be spotted.

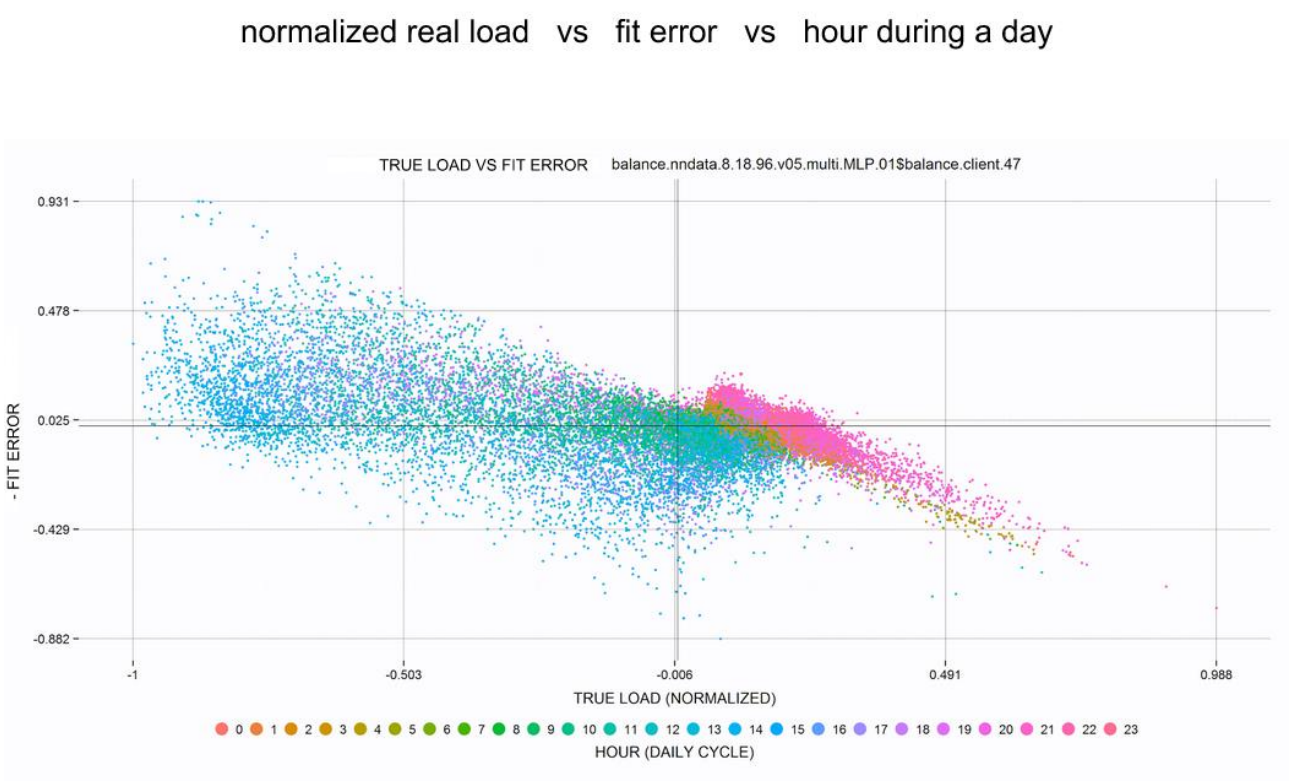

Figure 8: Distribution of (inverted) fit error for ANN trained with whole year data at the neighbourhood level as a function of the normalised energy load

2.3.2 Prediction Test

Figure 9 illustrates detailed distribution of prediction MAE for ANN trained at the neighbourhood level with randomly chosen one week data and then tested with next week data. The average fit MAE was no larger than 0.052 (5.2%). The very bottom row illustrates the margin distribution of fit MAE as function of time in day. The very left column illustrates the margin distribution of fit MAE as function of day in week. The bottom-left tile informs on the average MAE index. The average fit MAE was no larger than 0.052 (5.2%).

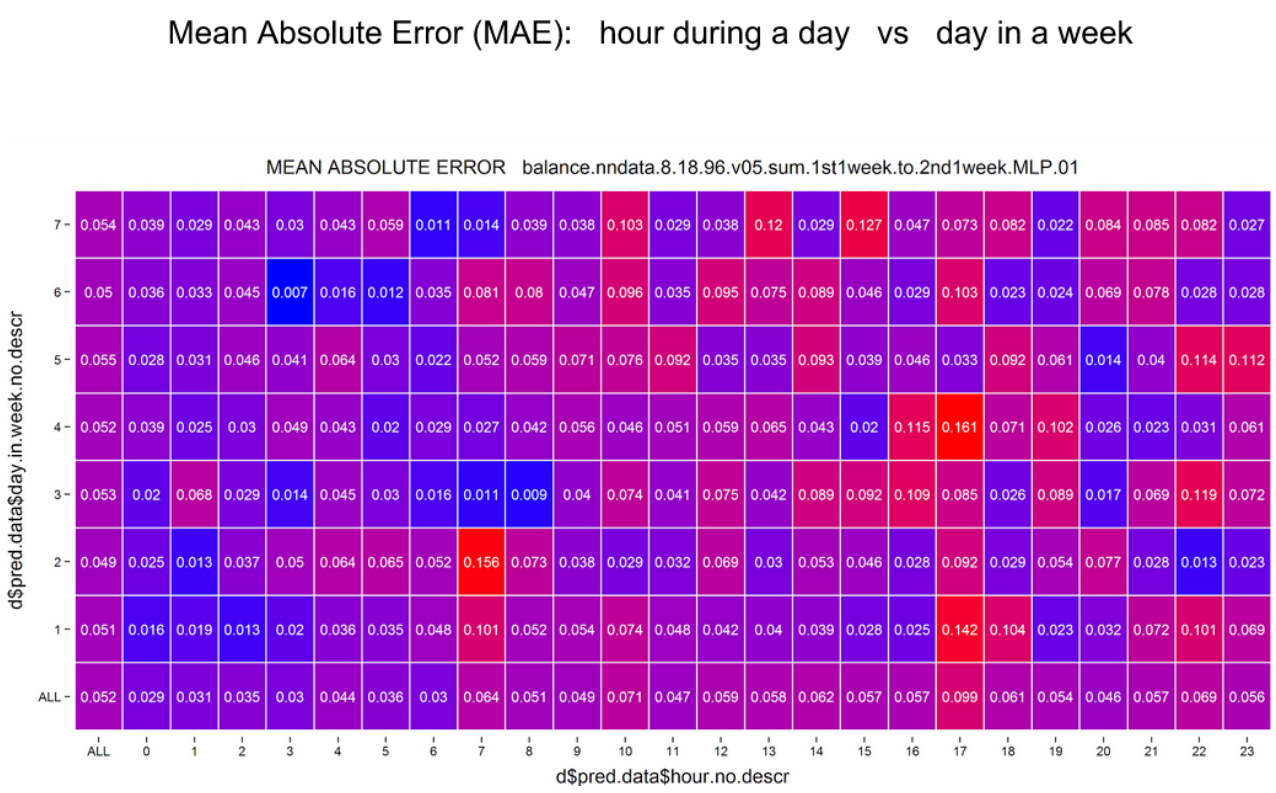

Figure 9: Distribution of prediction MAE for ANN trained at the neighbourhood level with randomly chosen one week data and then tested with next week data - as a function of time in day and day in week

Figure 10 illustrates detailed distribution of (inverted) prediction error as a function of the normalised energy load - for ANN at the neighbourhood level with randomly chosen one week data and then tested with next week data. The prediction error distributes evenly throughout the majority of energy load values, showing only a slight tendency to underestimate the output for high energy load values.

normalized real load vs prediction error vs hour during a day

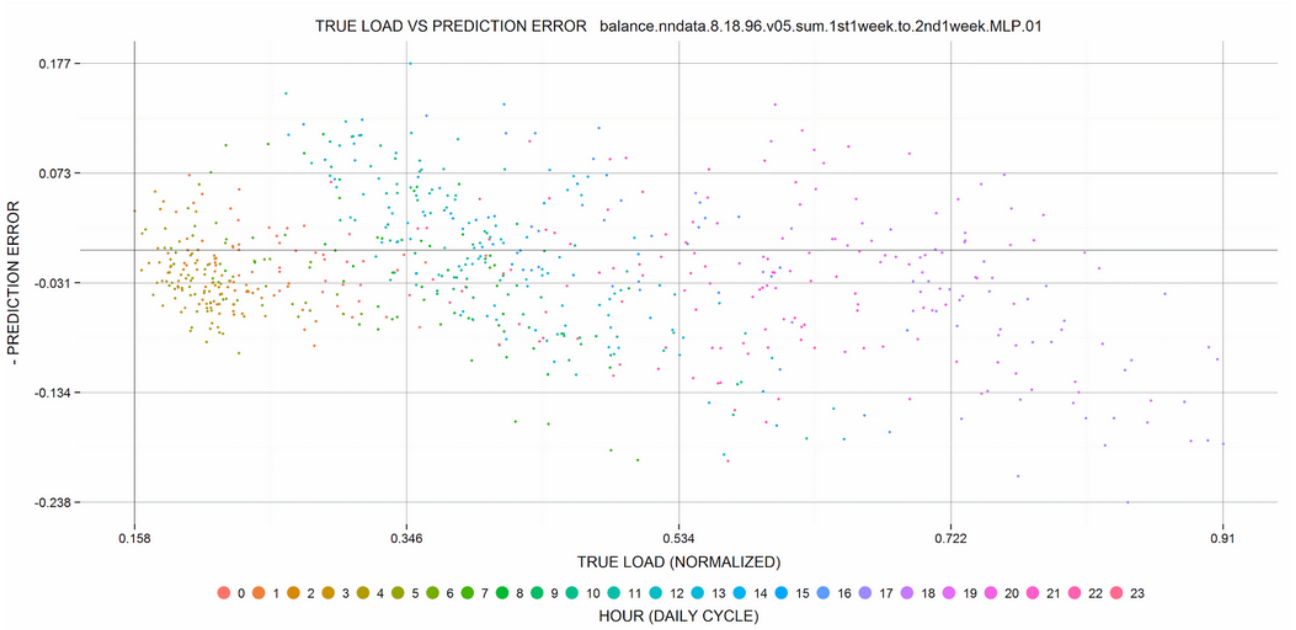

Figure 10: Distribution of (inverted) fit error for ANN trained at the neighbourhood level with randomly chosen one week data and then tested with next week data as a function of the normalised energy load

3 Description and implementation of the GUI for prosumers

All the functionality provided by e-balance must be presented to different users with different requirements and needs. This has been achieved by means of the GUI presented in this section. The GUI shows information generated or handled by the e-balance system and allow users to interact and control it depending on the user credentials. This section details the motivation and objectives of the GUI (Section 3.1) and the general architecture of the GUI (Section 3.2). The description of the GUI and the implementation and security details are presented in Section 3.3 and 3.4.

3.1 GUI motivation and objectives

Despite being presented in WP5, the GUI is used to monitor and control all the different processes of the e-balance system, i.e. communication, monitoring of sensors, energy management, etc. The GUI identifies authorized users in the e-balance system and allows them to verify and control different parts of it. In this document the GUI implemented in the CMU (Customer Management Unit), which runs the e-balance energy management application at prosumer level, is presented. The GUI does not only monitor the status of the e-balance energy management application but also has the following goals:

- Monitor the status of the grid from a user perspective
- Monitor energy usage and show historical data
- Monitor device consumption and priority in case of energy supply limitations
- Show weather information
- Show user and contract related information
- Alert users when warnings of alarms are detected
- Manage administration and configuration options for network operators
- Manage security and restrict access to unauthorized users

The main objective of the GUI is to offer all these functionalities in an attractive and easy-to-use way, taking into account that it will be used by different user profiles with different requirements, e.g. administrators operators, normal users, etc.

In addition, the GUI has been designed according to requirements collected in the deliverable D2.4 (stakeholders' requirements) and in turn based on the first social studies outputs of the entire WP2 activities. These constitute **33** functional, non-functional and constraints that are shown in Table 1.

Table 1: Prosumer's requirements for the GUI

Id[2]	Description
1.1	Specify the set points of the selected strategy through some GUI with a "setup wizard". This setup wizard has to be friendly.
1.2	The GUI shows all the available information of the customers' system: production, storage capacity, current consumption, energy prices, and historic consumption and production when the customers start the application. This set of information changes automatically as devices are removed or as new devices are added.
1.3	The GUI supports the customer while entering the strategy and rejects every incoherent strategy that may became unprofitable or dangerous (incoherent strategy has to be defined during the e-balance project).
1.5	Customers have a user friendly GUI available.
1.6	The GUI shows the information about the time, the day, weather conditions, and weather forecast.
1.7	The GUI shall show the possibilities for the customer strategy definition: show all active devices that produce, store and consume energy that can be taken into account while specifying the strategy as well as the grid and market specific parameters that are available and can influence the strategy execution, like dynamic energy price, option to give up control in case of grid level failure, etc. All dynamic parameters like obligations due to dynamic production and/or consumption contracts or agreements as well as options that are defined by the contract between the customer and supplier/DSO/Aggregator shall be visible in the GUI.

[2] The requirements Id number is related to the sequence defined in the deliverable D2.4 to facilitate the reference thereof. The first number is related to the use case and the second one is the requirements' order.

Id[2]	Description
1.8	Depending on the available devices and control parameters provided by the grid (from the energy supplier or aggregator) customers must be able to specify, using the GUI, their strategy that controls the use of energy produced by them.
2.1	The GUI shall allow the customer to specify the energy consumption priorities of his devices. These priorities define the order in which the devices may be switched off as the available energy level drops, or switched on, in the opposite case.
2.2	The GUI shall allow the customer to specify the use of different power modes of the devices that may be applied by the customer management unit instead of switching the device off in order to reduce the energy consumption in case of power source limitations.
2.5	The customer shall be able to allow (via the GUI) an aggregator to modify his energy consumption priorities. The changes are limited to the frame defined by the customer. This setting may be also connected with negotiating the benefits for the customer.
2.6	Specify/change consumption priorities and limitations for each appliance at home through some GUI with a "setup wizard".
3.1	The GUI shall provide the customer with information on current energy production limits and compare those with the current production.
3.2	The GUI shall provide the customer with current power quality parameters and shall allow comparing these with thresholds defined by the energy supplier (or aggregator).
3.3	The GUI shall allow the customer to define a strategy depending on the available means, i.e., reactions in case the production exceeds the limit (e.g., limit the production, consume the surplus energy or store it in the local storage)
3.4	The GUI shall allow the customer to define a strategy in case the power quality is below the defined threshold (e.g., switching off the generators that may cause the quality degradation)
3.5	The customer shall have the possibility (via the GUI) to allow an aggregator to control the energy production. The degree of freedom for the aggregator shall be definable by the customer.
4.2	The GUI shall allow the customer to define a strategy that defines the way the mutual agreement on consumed and produced energy is realized. For instance, defining a safety margin or reaction in case of deviation, i.e., controlling production, storage and consumption in order to fit into the defined numbers.
4.4	The GUI shall allow the customer to define that an aggregator controls the energy production and consumption limits agreement procedure on his behalf. The customer is able to define the degree of freedom the aggregator enjoys with that respect.
4.7	GUI shows information on the agreed contract and current consumption and production compared to the forecast.
4.8	GUI warns of the possibility of contract violation.
4.9	GUI shows calculated cost of contract violation.
4.10	Customer is able to report in simple way (through GUI) change of production capabilities (e.g. due to short-term maintenance of micro generation).
5.6	The GUI shall allow the customer to define the strategy on the usage of the energy storage. The strategy can be defined depending on the available parameters from the grid and the market, like energy price and depending on the devices available for the customer, like production and consumption.
5.7	The GUI shall present the current state of the energy storage, like the capacity, the structure of the energy nature, the charging level and mode - charging or discharging.
5.12	The GUI shall allow the customer to give the control of the energy storage to an aggregator. It shall also allow the customer to define the degree of freedom for the aggregator.
6.8	The GUI shall allow the customer to define the strategy for charging and/or discharging the electrical vehicle.
7.1	The system shall provide a diversity of GUI for different stakeholders and with different sets of functionalities.
7.2	The interfaces to access the system have to be defined, in order to keep the implementation of the different GUI simple.
7.3	The access to the data via the GUI interface is protected against unauthorized access.
7.4	The GUI shall provide self-feedback to the customer.
8.2	The GUI shall allow the customer to specify the restrictions that apply for his personal data. The customer may define policies for all data kinds that can be exchanged between the customer and grid domains.
8.3	The GUI allows negotiating the data handling policies.
8.4	The resolution of the sampling of the customer consumption (and production) should not reveal any customer private details when the data leaves the customer domain.

Id[2]	Description
12.2	The resolution of the sampling of the customer consumption (and production) must not reveal more customer private details that defined by the customer when the data leaves the customer domain.

These requirements have been the baseline to design and implement the GUI architecture whose details are explained in the following section.

3.2 GUI architecture

The e-balance solution is a heterogeneous system that entails many different devices, monitoring sensors, different communication networks and different participating stakeholders. This complex scenario imposes a set of key requirements:

1. The GUI must be platform independent because it is thought to be executed in different hardware devices, such as embedded devices e.g., a Beaglebone or powerful PCs e.g., a server acting as a MVGMU.
2. The GUI should be easily accessible. Users access the system from remote places such as remote control centres, prosumer premises, companies, etc.
3. The GUI should be able to control and monitor the state of the e-balance system.
4. The GUI should only allow access to authorized users according to their credentials.
5. The GUI should be dynamic. It should be able to show up-to-date information.

As depicted in deliverable 4.3 of the e-balance project the communication architecture of the prototype will be based on web services. All information and control messages will be exchanged between the main management units using web services over HTTP. Communication based on this technology is platform independent and can work over ubiquitous TCP/IP networks. This communication architecture provides a flexible way of implementing GUIs that use this information. Only web service consumption is required to obtain and to generate request from/to the e-balance system. In other words, the e-balance system is not tied to a specific type of GUI (such as a PC application). Figure 11 depicts the architecture of the e-balance system from the point of view of the GUI. Although only one GUI prototype is presented in this document the architecture presented allows different GUIs to consume data and generate requests from/to the e-balance system.

Figure 11: GUI and e-balance services architecture

In order to make the GUI available to a wider set of devices and to cope with requirements 1 and 2 the GUI has been implemented as a web application. These applications can run over a wide set of hardware devices and can be accessed from a conventional web browser from remote places. A web server has been installed in both the CMU (HomeWaveControl device by LessWire) and the LVGMU (Beaglebone black) to host the web application that provides the GUI. Both devices are shown in Figure 12. Finally, it is interesting to highlight that although the GUI and the e-balance middleware have been deployed in the same management units for both the CMU and LVGMU in the e-balance demonstrator the architecture presented does not enforce this.

Figure 12: Devices running the energy management algorithms and hosting the GUI web app. Left: HomeWaveControl. Right: Beaglebone black

3.3 GUI description

As explained in the previous section, the GUI has been implemented as a web application that runs on the management units. The GUI obtains the information from the deployed e-balance web services and shows them to the users. The GUI is composed of a set of screens that are used to control different parts of the e-balance system. The different screens will be shown only to the corresponding user according to their role and permissions. For example, administration screen will only be shown to administration users.

The following set of figures shows the aspect and functionality provided by the e-balance GUI. The current screens and the set of actions that can be carried out from the GUI will likely change in future iterations of the project to include new functionalities or update existing ones. The first screen a user would interact with is the login window depicted in Figure 13. The set of screens available once you log in will depend on the credentials provided. For example, Figure 14 and Figure 15 show the same screen for an administrator and a normal user respectively. The screen is composed of a series of widgets and shows a summary of the current state of the e-balance system. Let us note, for example, that the admin has access to an administration menu (Figure 14) and the normal user does not (Figure 15).

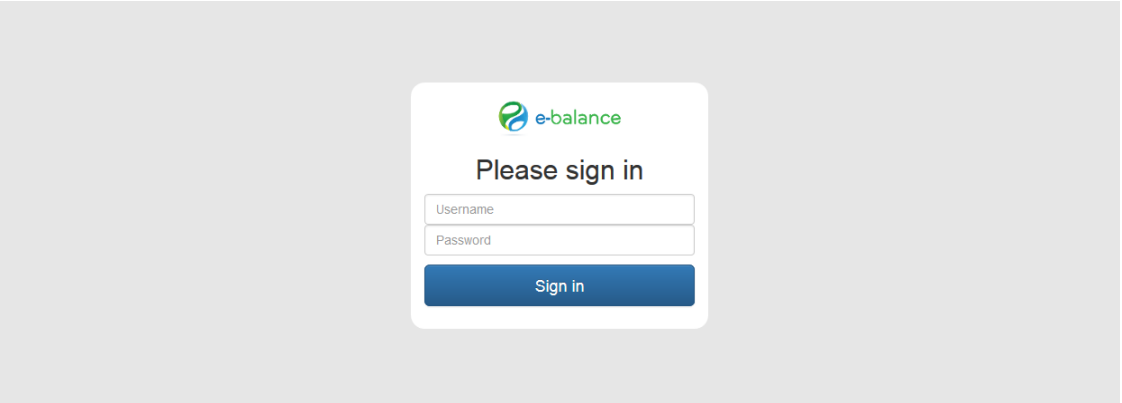

Figure 13: Login screen

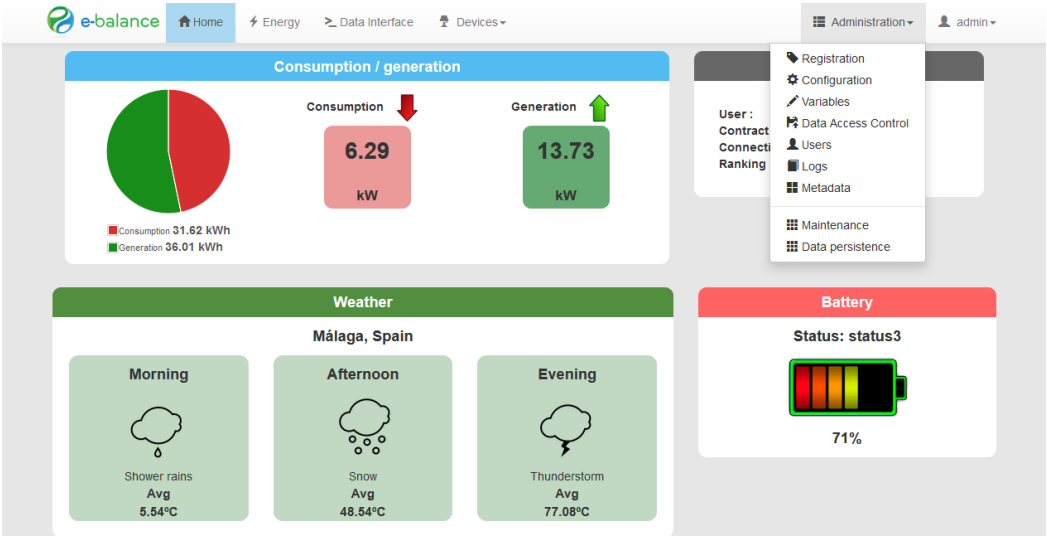

Figure 14: Summary screen for an administrator user

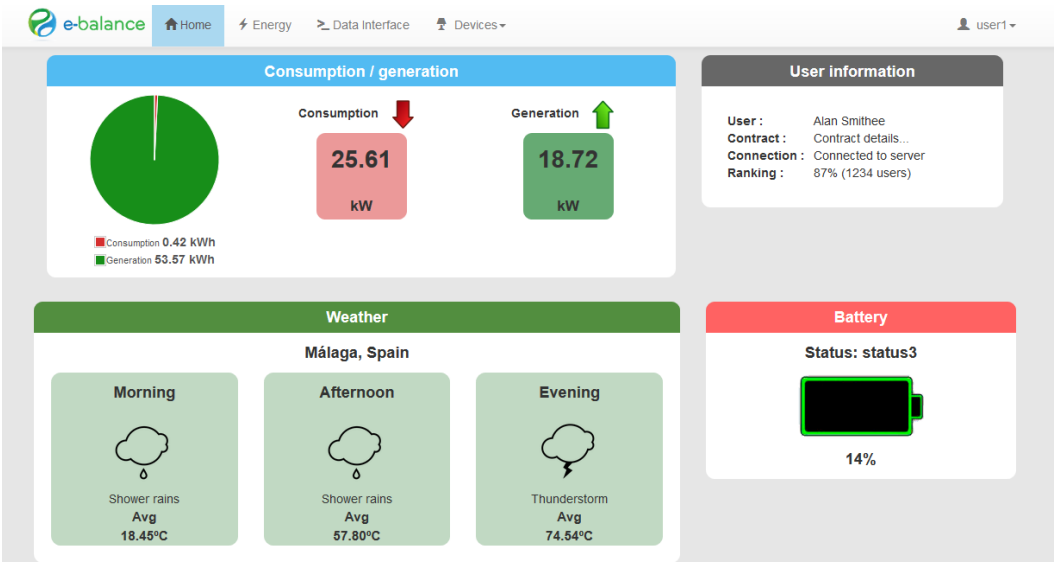

Figure 15: Summary screen for a default user

Energy consumption/production monitoring and control is one of the main goals of the e-balance system. From the "energy" screen shown in Figure 16 users can control and query historical energy consumption/production. The information collected by the system is shown as a series of graphs that give

users an idea on the amount of energy that is being consumed and produced. Also the state of storage devices is monitored.

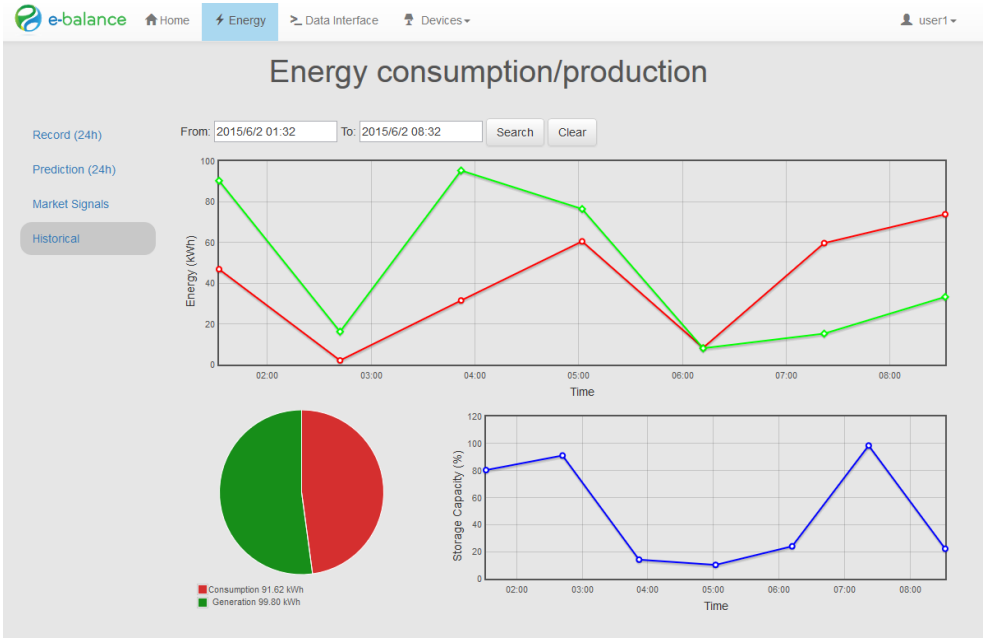

Figure 16: Energy consumption/production monitoring (including historical data)

The e-balance system optimizes the use of the smart grid based on different variables such as the number of devices connected. In case of energy supply limit the system decides to disconnect certain electrical devices to achieve a graceful degradation, that is, to avoid general outages by limiting energy supply to those devices that have the more critical needs. The more critical electrical appliances for each user are controlled by means of the screen shown in Figure 17. This screen lists the smart appliances registered in the system in the customer premises and allows the user to prioritize these devices. Only in case of energy supply limitation they will be disconnected to the extent possible based on the user preferences.

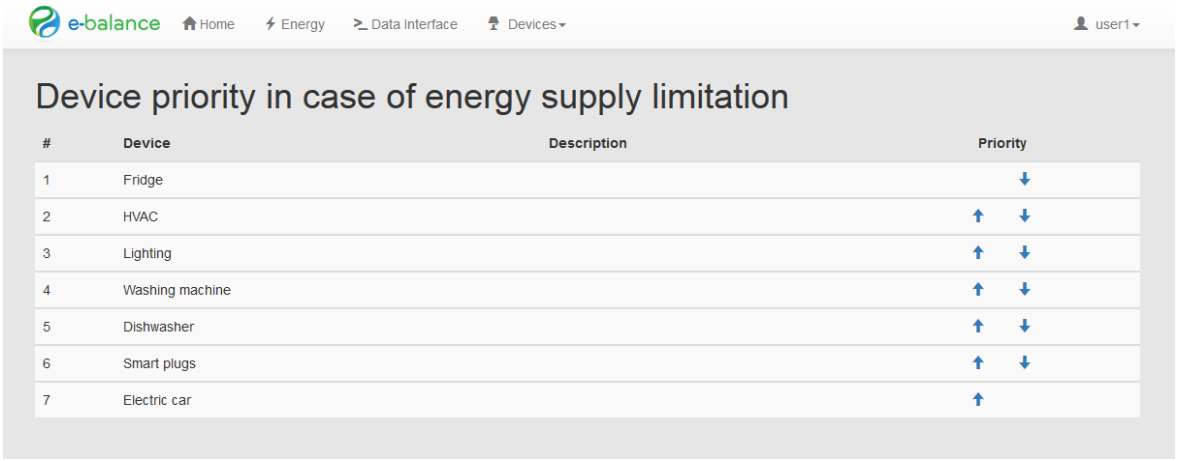

Figure 17: Device priority in case of energy supply limitation screen

Finally, there is an additional number of screens that lets the user configure other functionalities such as configuration of electrical appliances, weather forecast services, storage monitoring, etc.

3.4 Implementation and technical details

3.4.1 Implementation architecture and technologies

As mentioned in Section 3.2 the GUI prototype for the e-balance demonstrator has been implemented as a web application. This GUI implementation allows seamless access from any computer with a conventional web browser. Figure 18 shows the implementation architecture and how it communicates with the e-balance services.

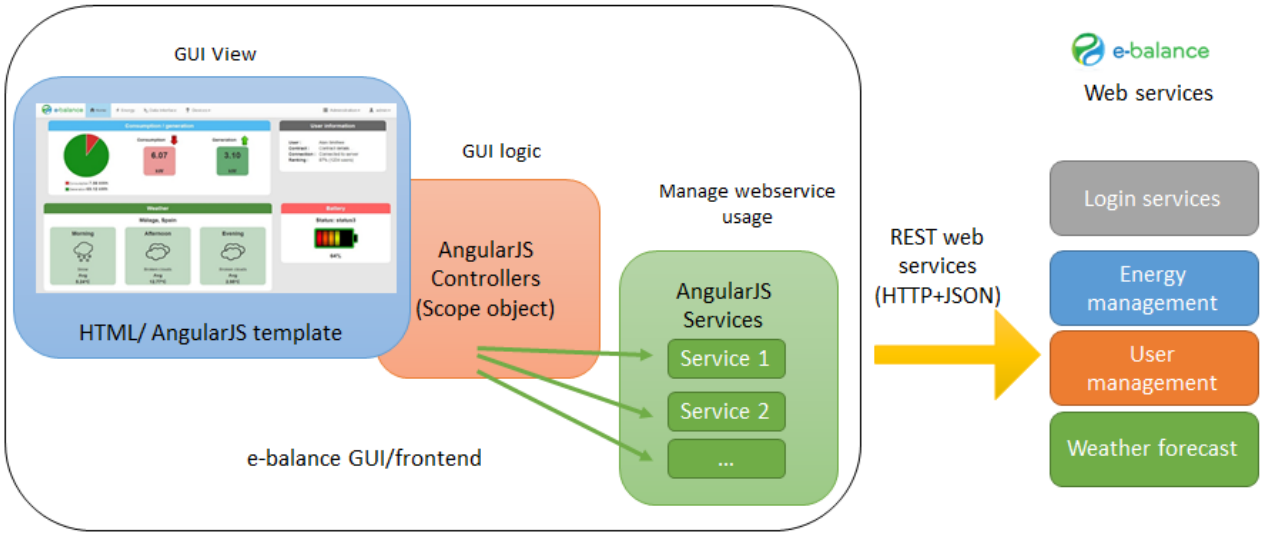

Figure 18: GUI implementation architecture and technologies

All the GUI frontend has been developed using HTML, CSS and JavaScript. The GUI requires dynamism to show up-to-date information based on the data provided by a set of e-balance services. In order to simplify the development of this web application a web application framework called AngularJS has been used. This framework is implemented in JavaScript and simplifies the development of dynamic web applications by providing functionality to communicate with external services, separate program logic from view and additional libraries to help the developer. The view is implemented with plain HTML and AngularJS templates. These templates contain information that is controlled and generated by different AngularJS controllers. These controllers contain the logic of the web application and among other tasks are in charge of consuming e-balance web services with the help of AngularJS services. The information modified in the controller is automatically refreshed in the HTML view.

The GUI was uploaded to a public web server in order to test its functionality by all partners involved in the project. Once a first version of the e-balance communication platform was available the GUI was ported together with the communication middleware to a Beaglebone device. A public IP has been set up so that the Beaglebone with the communication platform and the GUI are accessible by all partners in order to update its functionality according to the requirements of the project or to detect and solve bugs. The GUI functionality is expected to grow as the project develops, but, so far, the GUI is stable and is currently running on the final device that will be used in the demonstrator (Beaglebone black).

3.4.2 Security

As described in D4.3 all services provided by e-balance have a predefined security policy which is used to block access to unauthorized users. In order to execute any e-balance web service a user needs the corresponding credentials. Credentials are handled by the e-balance system via the e-balance web services. This means that the GUI is just a mere facade to the underlying e-balance services and more specifically to the underlying security system. From the login screen the e-balance login service is consumed and it will give access to a set of services depending on the credentials provided.

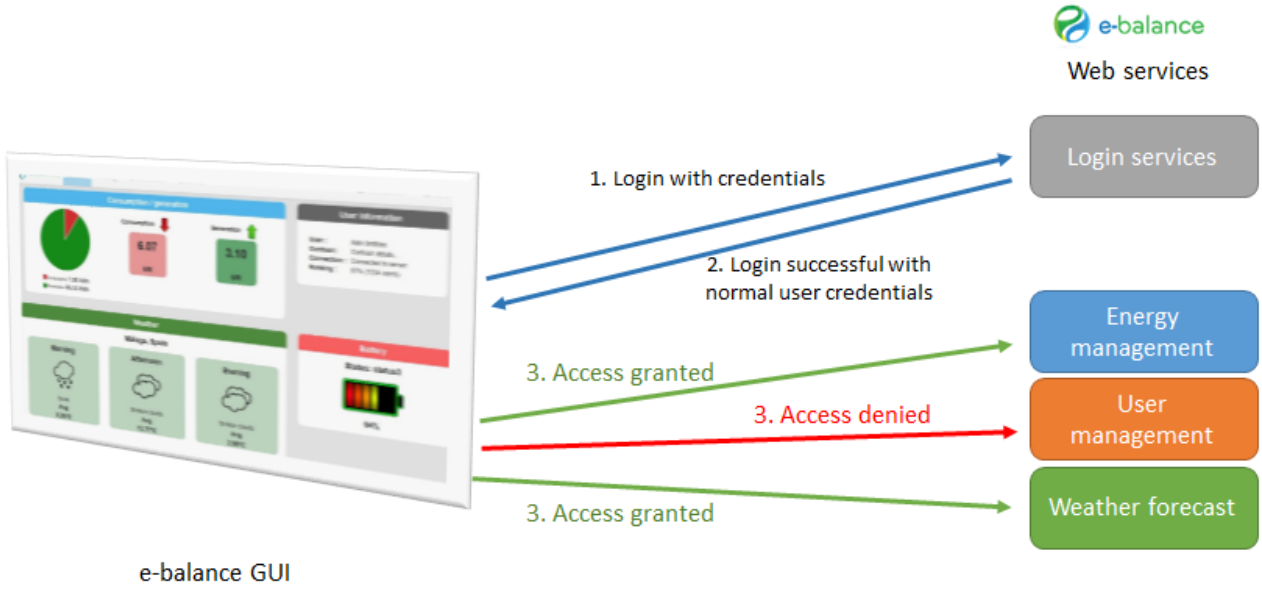

Figure 19: Login process

An example is depicted in Figure 19. A user logs in in steps 1 and 2. The login service acknowledges that the credentials provided correspond to that of a normal user. This means that the user can access some services such as the weather forecast but cannot access other ones such as the user management services. This security description is only the basic level to control users' credentials since other advanced features and mechanisms concerning security and privacy are specifically assessed in deliverables D4.3 and D5.4.

4 Description of energy balancing algorithms

Electric grids are dimensioned to accommodate expected peak loads. Existing grids were often planned decades ago and not designed for a high penetration of large loads, such as electrical vehicles (EVs) and heat pumps, and for large decentralized generation by, e.g., PV. As these devices increase the peak load of the consumption or generate peaks in production, measures such as peak shaving are an interesting alternative to costly grid reinforcements. Reducing peaks has additional advantages, such as: reduction of transport losses, power quality improvements and reducing the need for peak generators. Siano [13] surveys many reasons for peak shaving and goals of Demand Side Management (DSM) in general. Many of the current approaches only aim at peak shaving at the transformer, and do not consider power quality problems within the network. In contrast, the e-balance DSM approach prevents peaks in each hierarchical level of the electricity grid, and therefore resolves power quality problems at each level.

This chapter presents the core ideas behind the e-balance balancing algorithms. For further details, we refer to our publications on this subject [14][15]. These papers contain an extensive motivation, full description of the algorithms and evaluation of the results.

4.1 Steering signal

Within most DSM approaches, an aggregator can steer a group of houses towards a given objective using steering signals. These steering signals are often in the form of time varying prices for a unit of energy. This approach, referred to as dynamic pricing, is a well-accepted and popular approach in the research literature and in practice (e.g., [15][16][17][19]). The response of a house to such a signal consists of shifting its consumption from where it is relatively expensive to where it is relatively cheap. When all households receive the same price signal (referred to as uniform pricing), all houses tend to shift their loads to the periods where the energy is cheapest. McKenna and Keane [16] show that, under several uniform pricing schemes, the peaks are hardly reduced and only shifted in time. By using control algorithms that shift loads on behalf of customers, these problems may even become larger. The problem increases even further in future scenarios where customers have a lot of PV, EVs and heat pumps installed. Hence, **uniform pricing is not a proper approach for peak shaving.**

Whereas price based steering indicates in which intervals the consumption / production should be increased or decreased as much as possible, we argue that it is desirable to use a steering signal that precisely indicates what the goal is to avoid only shifting the peaks. Such steering signals can be (deviations from) a desired power profile, which is used by the approach presented in this chapter. We refer to this approach as profile steering.

In this approach we send (the difference with respect to) a desired profile to each house. This is a vector that indicates the desired power for each 15 minute interval for one day ahead (i.e., a vector of 96 power values). Each house (management unit) aims at minimising the Euclidean distance between its own planned profile and the desired profile. Because the planning problem is NP-hard (see [15]), a heuristic algorithm is required. The next section describes such a heuristic: an iterative algorithm that uses profile steering to obtain a balanced electric grid.

4.2 Profile steering algorithm

The electric grid has a hierarchical (fractal-like) structure. Within the e-balance architecture, management units are installed in each hierarchical level: top level (TLGMU), medium voltage level (MVGMU), low voltage level (LVGMU) and the customer level (CMU). This section presents an algorithm that exploits this hierarchy and is the same for each of these management units. For further details, see deliverable D3.2.

To initialize, each CMU is asked to obtain a power profile (24 hours ahead) that is as flat as possible. This makes sure that it is locally as flat as possible. For this, the algorithm considers the predicted power profile of the uncontrollable loads and the available flexibility. The algorithm is used to correct for peaks that occur at neighbourhood level, by requesting neighbouring houses or groups of houses to compensate for these peaks.

To configure the algorithm, at the top level a desired power profile is set by the operator. For example, a flat power profile can be set to obtain balanced electric grid. In the first step of the algorithm, the TLGMU requests the initially planned power profile (24-hour ahead) from the MVGMUs and sums them up to obtain a planning at the top level. The difference between the planned profile and the desired profile, called the difference profile, is sent to the MVGMUs. Each MVGMU tries to follow this profile, and how much closer it can get its own profile to the desired profile (the compensated distance) to the TLGMU as a single value. The TLGMU selects the best improvement, asks the respective MVGMU to change its behaviour to follow its suggestion and submit the planned changes. After the TLGMU incorporates these changes in the top level planned profile, it repeats this process until the top level planned profile is close enough to the desired profile, or until no significant improvements are made. Note that the TLGMU only accepts a single MVGMU to make a change, to avoid overcompensation.

The profile steering approach is explained for the TLGMU, but can be implemented at each hierarchical level. For example, the MVGMU updates its desired profile using requests from the TLGMU, and it requests similar changes from the LVGMUs it controls. The profile steering algorithm is also implemented in the LVGMU and the CMU.

The profile steering algorithm is formally given in Figure 1. For full details regarding this algorithm, the design choices, and an evaluation we refer to our paper on this subject [15].

Request each appliance $m \in \{1, \dots, M\}$ to minimize $\|\vec{x}_m\|_2$

$\vec{x} := \sum_{m=1}^{M} \vec{x}_m$ {Total household consumption}

repeat

$\vec{d} := \vec{x} - \vec{p}$ {Difference vector}

for $m \in \{1, \dots, M\}$ **do**

$\vec{p}_m = \vec{x}_m - \vec{d}$

For appliance m, find a planning $\vec{\hat{x}}_m$ that minimises $\left\|\vec{\hat{x}}_m - \vec{p}_m\right\|_2$

$e_m = \|\vec{x}_m - \vec{p}_m\|_2 - \left\|\vec{\hat{x}}_m - \vec{p}_m\right\|_2$ {Relevant flexibility of appliance m}

end for

Find the appliance m with the highest contribution e_m

$\vec{x} := \vec{x} - \vec{x}_m + \vec{\hat{x}}_m$ {Update the total consumption}

$\vec{x}_m = \vec{\hat{x}}_m$ {Update the profile of the appliance m}

Until $e_m < \varepsilon$ {Repeat as long as there is sufficient progress}

Figure 20: Profile steering algorithm

The profile steering algorithm in the CMU requests devices to follow a desired profile. To accomplish this, specialized device planning algorithms are required that follow the profile while respecting device constraints. This is discussed in the next section

4.3 Device planning

Devices cannot produce any desired profile, but should consider device constraints. For example, a time-shiftable device (e.g. a washing machine) should respect the configured deadline and can (commonly) only be shifted in time, its power usage cannot be influenced. The sections below shortly describe the device planning for several different classes of devices.

4.3.1 Time-shiftable devices

Time-shiftable devices (e.g. washing machines, dryers, dish washers) should be planned in a time window that is configured by the user. The planning algorithm should find a starting time t that, when the device is started at this time, minimizes the distance between the desired device profile and the planned profile that follows from the start time t.

The planning algorithm is easy to obtain: the algorithm calculates the profile obtained for each starting time t, and for each of these profiles it determines the distance to the desired profile. The profile with the shortest distance is optimal, it is chosen as the planned profile, and this distance is reported to the profile steering algorithm that is implemented in the CMU.

4.3.2 Electric vehicles

The charging power and time of electric vehicles can be chosen by the electric vehicle device planning algorithm. There are a few restrictions, namely: for each interval the planned power x_n should be in the permitted power range [P_{min}, P_{max}], the vehicle is charged in the intervals a,...,d (where a is the earliest allowed charging interval and d is the latest), and the required charge C is obtained within this interval (i.e. $\Sigma\ x_n = C$). The optimisation objective is to minimise the distance between the planned power vector x and the desired power vector p. For the vector x in the optimization problem below, we only consider the intervals a,…,d and require that the elements of x are zero outside this charging interval.

This can be stated as the following optimization problem:

$$\min_x \sum_n (x_n - p_n)^2$$

S.T.

$$P_{min} \leq x_n \leq P_{max}, \forall n$$

$$\sum_n x_n = C$$

Figure 21: Electric vehicle optimisation problem

In our research we found an algorithm that solves this problem (and several variations, for example taking electric prices into account) in O(N log N) time. Due to this low complexity (with low constant), we can find the optimal solution in milliseconds on a device with a limited CPU and memory such as the CMU. We describe the details about this algorithm in [14].

4.3.3 Batteries

Batteries provide a lot of flexibility to follow a desired profile. Similar to the electric vehicle, the power is constrained (but can be negative). Furthermore, the state of charge must remain positive and below its capacity M. We did not need this constraint explicitly for the electric vehicle, because we did not consider vehicle to grid in that case (note, our battery model allows to model an electrical vehicle with vehicle to grid).

The optimisation problem can be stated as follows.

$$\min_x \sum_n (x_n - p_n)^2$$

S.T.

$$P_{min} \leq x_n \leq P_{max}, \forall n$$

$$0 \leq \sum_{n=1}^{k} x_n \leq M, \forall k$$

Figure 22: Battery optimisation problem

To solve this optimisation problem, a slightly altered version of the electrical vehicle planning is used. This problem can be solved in $O(N^2 \log N)$ time.

4.4 Relation with use cases

The algorithms described in the previous sections cover several of the e-balance use cases. The relation to the use cases are briefly discussed below. For details on these use cases, we refer to D2.1 for a description, D3.2 for the system architecture details (management units) and to D6.1 for the relation with the demonstrators.

Use case 1 (Strategy-driven decision on the use of produced energy) **and 5** (Strategy-driven decision on the usage of grid-connected DER) are inputs for planning the power profile optimisation regarding production sources.

Use case 3 (distributed generation balancing and resilience): The profile steering algorithm and the respective device planning algorithms optimise how the generation is used, and keeps it balanced.

Use case 4 (energy consumption and production agreement/contract): The profile steering algorithm negotiates the flexibility of the household on behalf of the customer with the party interested in this flexibility. The user can configure the restrictions that should be respected. These restrictions can be strategies (see D5.1) or appliance settings such as deadlines as were described in the previous section.

Use case 9 (intelligent home appliance energy consumption balancing) and **Use case 10** (additional sensors for appliance energy consumption balancing): These use cases closely match what is described in the previous two sections.

Use case 11 (Microgrid energy balancing): Our balancing algorithms can assist at making the grid more balanced.

The balancing algorithms presented in this chapter can also increase the resilience of the grid. This topic is explored in D5.3.

4.5 Results

The algorithm from this section was verified in simulation for a neighbourhood of 121 houses with steerable EVs, PV panels, batteries and steerable washing machines. For details on this study we refer to D5.3. Figure 23 compares the e-balance "Profile Steering" algorithm against no control and the state-of-the-art algorithm. Clearly, the algorithm is capable of balancing the power within the network. Moreover, our algorithm is better at keeping the voltages within required bounds and balancing the power at each node in the network. For more details on results regarding power quality and energy resilience, we refer to D5.3.

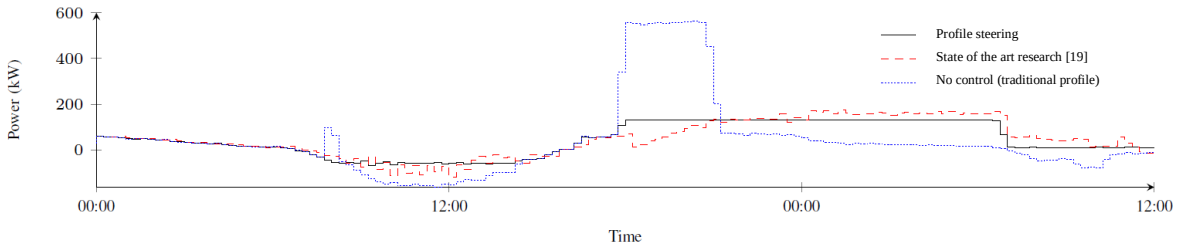

Figure 23: Power at the transformer (including losses) in optimal uniform pricing case study [15]

4.6 Implementation and validation details

4.6.1 Balancing algorithms

The balancing algorithms are implemented as a set of libraries written in C. During the integration (T5.5), these libraries will be integrated with the Middleware, GUI, etc.

libdsmdata

The first library, called libdsmdata, does all the data handling related to profiles, optimisation problems, etc. As operations on profiles are common within the planning algorithms, libdsmdata provided an efficient and yet general implementation of all common routines.

For each profile, four options are available:

- Each interval has a flexible length, the data values are integers
- Each interval has a flexible length, the data values are doubles
- Each interval has a fixed length (e.g. 15 minutes), the data values are integers
- Each interval has a fixed length (e.g. 15 minutes), the data values are doubles

The library contains routines to deal with all types of profiles in a convenient way. For example, functions to allocate, convert, free, iterate profiles are available. Also routines to convert the time base, take time slices and perform (unary/binary) operations on profiles are offered to make developing planning algorithms as straightforward as possible are offered by the library.

The libdsmdata library offers datatypes to describe optimisation problems specific to devices. For example, the optimisation problems from section 4.3 can be described and altered using this library.

This library is extensively tested using unit tests. Unit tests have been implemented for the corner cases for each of the functions.

libdsmplanning

The second library, libdsmplanning, contains a set of planning routines for device planning and fleet planning. For the first class, the device planning, tailored algorithms are registered in the library. For example, the library contains (a.o.) algorithms to plan time-shiftable appliances, electrical vehicles, batteries, etc. These algorithms are subdivided into two classes, namely time-shiftable (ts) or controllable-load (cl). The latter class can be used to describe devices such as: electric vehicles, batteries, heat pumps, read rods, CHPs, etc.

The user of the library requires no knowledge of the internal working of the algorithm. Instead, she can describe the device restrictions using the data structures from libdsmdata, and ask the libdsmplanning_ts_find_algorithm and libdsmplanning_cl_find_algorithm functions, to find an algorithm within a given class.

To control a set of appliances or management units, so-called fleet planners are used. These can be easily added or replaced such that other algorithms can be integrated in the future. At the moment, only the profile steering algorithm is implemented.

This library is extensively tested using unit tests. Unit tests have been implemented for the corner cases for each of the functions and planning algorithms and the results have been positive, what allows following with the integration of the entire energy platform in task T5.5 and the whole e-balance system in WP6.

5 Complementary algorithm: Power limitation algorithm

In order to complete the use cases' requirements and carry out the programmed activities in the final demonstrator sites, it is necessary to develop complementary algorithms and routines to control and manage all the functionalities together. The most important detected and related directly to e-balance algorithms is the use case #2 (Energy consumption priorities in case of energy delivery limitations) according to the page 96 of the e-balance deliverable D6.1 (Specification of the demonstrators). The power or energy supply is an additional restriction that can appear when unexpected malfunctioning of production units or increment energy consumption happen. This issue is usually managed by the DSO and through the TSO and balance responsible party through an iterative process that tries to avoid blackouts and requires more energy from backup production units in a few seconds. However, the complexity of a fractal-like system, such e-balance should assure such issue is solved faster to increase the promised robustness and resilience of the potential smart-grid.

The following algorithm is called "Power limitation algorithm (PL-a)" and we can find previous references and assessments as Pipattanasomporn [20], where the power limitation in a demand response scenario is evaluated and programmed with simple algorithms. The details of design, implementation and validation are described in the following section.

5.1 Description of the power limitation algorithm (PL-a)

The power limitation algorithm (PL-a) runs through the management units LVGMU and CMU and the latter deals with the appliance disconnection to keep the power below the maximum power calculated by the LVGMU. The description of the algorithm is shown as follows and as flow chart in the Figure 24:

Preconditions (LVGMU)

Detection of inevitable energy supply limitation (u:= units; p:= prosumers):

$$\sum_{u} \text{energy production}(u) < \sum_{p} \text{energy demand}(p)$$

Quantification of energy limitations:

$$\text{energy limitation} = \sum_{u} \text{energy production(u)} - \sum_{p} \text{energy demand(p)}$$

Generation of individual energy limitation based on the whole "energy limitation" value

Comment: The generation of individual power/energy limitations runs through the profile steering algorithm

Algorithm

Request each CMU $\boldsymbol{\mu \in \{1,\ldots,M_H\}}$ to change to priority level and the maximum power allowed $\mathbf{P_{max}}$

Request each appliance $\mathbf{m \in \{1,\ldots,M\}}$ to maximum power allowed $\mathbf{P_{max}}$

CMU calculates new power profile

If $\Sigma_M\, Power(\mu) > P_{max}$ **then**

$PL = PL_{higher}$

Comment: PL is the current priority level and PL_{higher} is the appliance with the lowest priority (first one to be disconnected)

Repeat

Request appliance $\mu(PL)$ disconnection

Update current priority level: $PL = PL - 1$

CMU calculates new power profile

Until $PL = 1$ or $\Sigma_M\, Power(\mu) < P_{max}$

CMU sends current Power profile

Else *"No limitation is necessary by this CMU"*

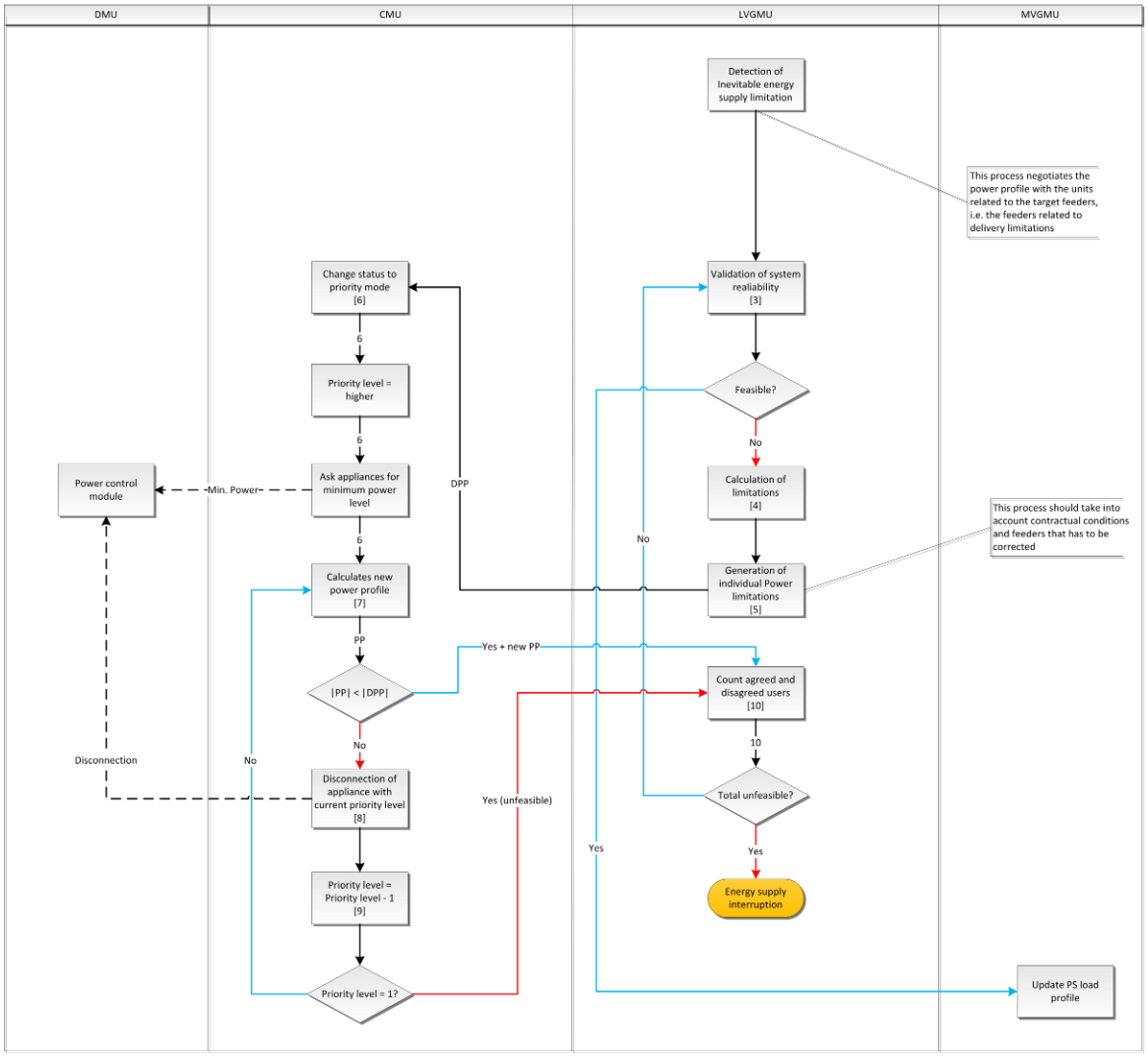

Figure 24: Flow chart of the Power limitation algorithm (PL-a)

Due to the simplicity of this algorithm, it has been implemented and validated in a visual basic environment to check the routine flow is right in all the conditions and no further specific test for validation are expected. In addition, the real implementation will be carried out during the integration task (T5.5) in the selected computer language of the middleware and management units when the number of smart devices, which can participate in a tentative disconnection due to power supply limitation, will be decided in next steps. Therefore, the real validation and demonstration is moved into the integration task (T5.5) and demonstration work package (WP6) stages respectively.

6 Conclusions and Next steps. Integration in the Energy Platform

The activities carried out in the task T5.2 and described in this document have permitted to obtain a list of conclusions and recommendations for next steps of the e-balance project, which are commented as follows:

1. The prediction models have shown very positive results at the neighbourhood level, though the human behaviour prediction constitutes still a big challenge. Even when we dispose a complete data base of energy consumption and production, as we have accessed in this study, there are many uncertainty sources that struggle obtaining accurate results. Although the data processing has been very complete and exhaustive, we expect to continue with the refinement of these predictions models during the project demonstrator in order to obtain better results both at neighbourhood and home level. The prediction models should be integrated in the corresponding module of the energy platform and they will provide the required inputs for the energy balancing algorithms in the CMU and LVGMU. The difference between the predictions at neighbourhood and home levels will be one of the aims during the integration and demonstration steps in WP6, since the potential conclusions and outputs can consolidate the reliability of the proposed e-balance services like fraud and fault detections due to the detected discrepancies of demand and consumption. The comparison between predictions and measures at different levels in the grid opens new chances to complete and improve the management of the e-balance platform.
2. As mentioned in the chapter 3, the GUI development is still in a beta stage and is open to receive modifications from partners and the suggestions collected from the futures surveys of final users in the demonstrators. However, though most of the requirements have just been implemented, the representation of results and the interaction with the user is still pending of the detected improvements with the aim of satisfying all the requirements: use cases, control and balancing algorithms, users, etc. This activity will continue through task 5.5 and the activities of WP6.
3. Concerning the energy balancing algorithms, these have been widely tested by the University of Twente team and we expect the integration thereof will not present problems with the middleware, the communication platform and the security/privacy mechanisms. However, as the risk of incompatibilities is inherent for every integration process, it is suggested to execute the models when a substantial change in the platform happens in order to mitigate iterative problems. In addition, during the integration and demonstration stage, the implementation of the PL-a will be done with a selected computing language compatible with the system and when the smart appliances to be included in the demo have been defined, since this depends on the specifications of installed devices. Finally, the virtual demo in the IHP premises should take into account such functionalities, both to optimise the consumption/production profiles, using the profile steering algorithms and to check the PL-a is able to fix the power supply limitations.
4. All the inputs and outputs of the presented algorithms, models and the GUI should be taken into account especially within the activities T5.3 (resilience mechanisms) and T5.4 (privacy and security mechanisms) in order to guarantee all the implemented mechanisms are using the same information regarding magnitude, origin, timestamp, etc. Although this issue has been considered in the communication platform (WP4) and system architecture (WP3), it is important to emphasize whatever deviation in measures regarding one of these aspects can be very difficult to detect after the integration, so it is mandatory to check the shared information among the different modules.

References

[1] Høverstad B. A., Tidemann A., Langseth H. (2013). Effects of data cleansing on load prediction algorithms. 2013 IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG).

[2] Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Cralendon Press, Oxford.

[3] Hernandez L., Baladrón C., Aguiar J. M., Carro B., Sanchez-Esguevillas A. J., Lloret J. (2013). Short-Term Load Forecasting for Microgrids Based on Artificial Neural Networks. Energies, Vol. 6, No. 3.

[4] Kandananond. K. (2011). Forecasting Electricity Demand in Thailand with an Artificial Neural Network Approach. Energies, Vol. 4, No. 8.

[5] Osowski Stanisław (2013). Sieci neuronowe do przetwarzania informacji. Oficyna Wydawnicza Politechniki Warszawskiej.

[6] Jian-Kai, L., Cattani, C., Wan-Qing, S. (2015). Power Load Prediction Based on Fractal Theory. Advances in Mathematical Physics, Vol. 2015.

[7] Theiler, J. (1990). Estimating the Fractal Dimension of Chaotic Time Series. The Lincoln Laboratory Journal, Vol. 3, No. 1

[8] MacKey, D. J. C. (2003). Information Theory, Inference, and Learning Algorithms. Cambridge University Press.

[9] R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

[10] Bergmeir C., Benitez J.M. (2012). Neural Networks in R Using the Stuttgart Neural Network Simulator: RSNNS. Journal of Statistical Software, Vol. 46, No.7.

[11] MathWorks, Inc. (2002). MATLAB and Neural Network Toolbox Version 4. Natick, Massachusetts.

[12] Zell A., Mache N., Hübner R., Mamier G., Vogt M., Schmalzl M., Herrmann K.-U. (1994). SNNS (Stuttgart Neural Network Simulator). In: Neural Network Simulation Environments, Vol. 254.

[13] P. Siano, "Demand response and smart grids–a survey," Renewable and Sustainable Energy Reviews, vol. 30, pp. 461–478, 2014.

[14] T. van der Klauw, M. E. T. Gerards, G. J. M. Smit, and J. L. Hurink, "Optimal scheduling of electrical vehicle charging under two types of steering signals," in IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Istanbul, Turkey. USA: IEEE Power & Energy Society, October 2014, pp. 1 – 6.

[15] M. E. T. Gerards, H. A. Toersche, G. Hoogsteen, T. van der Klauw, J. L. Hurink, and G. J. M. Smit, Demand side management using profile steering," in Proceedings of the IEEE PES PowerTech 2015, Eindhoven, the Netherlands, June 2015, p. 6, (in press).

[16] K. McKenna and A. Keane, "Discrete elastic residential load response under variable pricing schemes," in IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Istanbul, Turkey. USA: IEEE Power & Energy Society, Oct 2014, pp. 1–6.

[17] A. Molderink, V. Bakker, M. G. C. Bosman, J. L. Hurink, and G. J. M. Smit, "Management and control of domestic smart grid technology," IEEE transactions on Smart Grid, vol. 1, no. 2, pp. 109–119, September 2010.

[18] A. Faruqui, D. Harris, and R. Hledik, "Unlocking the €53 billion savings from smart meters in the EU: How increasing the adoption of dynamic tariffs could make or break the EUs smart grid investment," Energy Policy, vol. 38, no. 10, pp. 6222–6231, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0301421510004738

[19] S. Bu, F. Yu, and P. Liu, "Dynamic pricing for demand-side management in the smart grid," in Online Conference on Green Communications (GreenCom), 2011 IEEE, Sept 2011, pp. 47–51

[20] Pipattanasomporn, M.; Kuzlu, M.; Rahman, S., "An Algorithm for Intelligent Home Energy Management and Demand Response Analysis," Smart Grid, IEEE Transactions on , vol.3, no.4, pp.2166,2173, Dec. 2012

