Modelling and simulation infrastructure for smart energy and renewable technologies integration in urban districts

Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (30^{th}cycle)

Modelling and simulation infrastructure for smart energy and renewable technologies integration in urban districts

By

Lorenzo Bottaccioli

******

Supervisor(s):
Prof. Enrico Macii, Supervisor

Doctoral Examination Committee:
Prof. Marco Aiello, Referee, University of Stuttgart
Ana Maria Dumitrescu PhD, Referee, Politehnica University of Bucharest
Federico Viani PhD, University of Trento
Prof. Andrea Acquaviva, Politecnico of Turin
Prof. Andrea Calimera, Politecnico of Turin

Politecnico di Torino
2018

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Lorenzo Bottaccioli
2018

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

I would like to dedicate this thesis to my loved ones.

Acknowledgements

I would like to acknowledge the EDA group and the Lab 4 for their warm welcome and their essential help during this three year of research. In particular, I would like to thank Professor Enrico Macii for giving me the opportunity of undertaking this PhD. Especially, I thank Professor Andrea Acquaviva for his trust and guidance that allow my competences to flourish in the unexplored field of Computer Science. Last, but not least, I would like to express my gratitude to Edoardo Patti for working with me on a daily basis, sharing his knowledge and leading me in this research. Finally, I will like to acknowledge the EEB Cluster on smart communities for funding the research, the Municipality of Turin, the PoliTo Living LAB and IREN for providing the data used in the thesis.

Abstract

The 196 parties attending the conference on climate changes (COP21) in Paris highlighted the need of reducing greenhouse gas emissions [1]. In this regard, in the last years, many countries are providing incentives to promote the deployment of low-carbon and sustainable energy production technologies [2], generation such as Photovoltaic (PV) Systems. The International Energy Agency reports that [3] installation of Renewable Energy Sources (RES), Distributed Generation (DG) and an optimization of consumption with a smart use of energy is required in our cities in order to achieve the goal of reducing green house emissions. ICT technologies, in particular the Internet of Things, enable the possibility of controlling and optimizing consumption [4] hence increasing energy efficiency.

The transition from centralized production system to a distributed generation, that can be based on renewable or on conventional sources, substantially modifies the operation of electricity networks: the direction of power flows in the MV lines and even in high voltage/medium voltage (HV/MV) transformers can be reversed, voltage profiles are modified, fault management is affected [5, 6], etc. For all these reasons, distribution networks need to become *Smart* and new control strategies, algorithms and technologies need to be tested and validated before their implementation and installation in real systems.

In this context, ICT play a crucial role in both planning expansion and monitoring operation of distributed energy sources. The crucial roles of ICT and the emerging Internet-of-Things (IoT) are highlighted by the spread diffusion of heterogeneous and pervasive sensors in our houses, district and cities. IoT devices and sensors allow to collect large amounts of energy related data capable of describing the consumption behaviours of the citizens. Hence, the increasing presence of sensors calls for the development of distributed software infrastructure for exploiting such IoT devices for data management and collection. Furthermore, IoT devices enables the possibility of

monitoring devices and system in order to develop models for the simulation and optimization on energy process.

This Thesis presents a distributed infrastructure, called SMIRSE, for modelling and simulating renewable energy sources and smart policies integration in urban districts. SMIRSE is implemented as a modular infrastructure build with a micro-services approach that exploits Internet of Things communication protocols. This approach enabled interoperability between hardware and software components of the SMIRSE platform and at the mean time its modularity, extendability and scalability. Its modularity allowed the interfacing and integration between dedicated Real-time Grid Simulator, software simulation modules and real-time data in order to model the grid behavior. New modeling and simulation tools for *i)* Solar energy simulation, with a focus on Photovoltaic systems; *ii)* Integration of RES and smart policies with the distribution grid; *iii)* Characterization of thermal performance of Buildings and power consumption prediction; and *iv)* Buildings indoor temperature simulation and monitoring, have been designed, developed and integrated upon the backbone of the microservices-based infrastructure.

The main advantage of the SMIRSE infrastructure is its capability in creating different scenarios for Multi-Energy-System simulation with a minimum effort. Examples of scenarios were SMIRSE can be used are: i) Installation of Renewable Energy Sources, ii) Grid reconfiguration, iii) Demand Response and iv) Demand Side Management. In addition, the proposed infrastructure enables to study the interoperability among different use-cases in a plug-play fashion. Finally, the proposed solution can integrate Smart Metering Architecture to exploit (near-) real-time data collected from the field to co-simulate different smart energy strategies with real information. The main contribution of this study is the design and development of a distributed infrastructure for energy system simulation that exploits state of the art ICT technology. Its worth-nothing to say that such ICT technology have been customized for the purpose of developing energy system co-simulation infrastructure.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms / Abbreviations

API	Application Programming Interface
BHN	Building Heating Network
BIM	Building Information Models
CHP	Combined Heat Power
DG	Dstributed Generation
DHI	Diffuse Horizontal Incident radiation
DIA	Device Integration Adapters
DNI	Direct Normal Incident radiation
DOY	Day of the Year
DR	Demand Response
DSM	Demand Side Management
DSM	Digital Surface Model
DSO	Distribution System Operator
EWMA	Exponentially Weighted Moving Average
GHI	Global Horizontal Incident radiation
GIS	Geographical Information Systems

HDN	Heating Distribution Network
HIL	Hardware in the Loop
HV	High Voltage
HVAC	Heating, Ventilation and Air Conditioning
ICT	Information Communication Technologies
IoT	Internet of Things
JSON	JavaScript Object Notation
KPI	Key Performance Index
LCE	Legates's Coefficient of Efficiency
LV	Low Voltage
MAD	Mean Absolute Difference
MAPE	Mean absolute percentage error
MARS	Multivariate Adaptive Regression Spline
MBD	Mean Bias Difference
MES	Multi Enrgy System
MQTT	Message Queue Telemetry Transport
MV	Medium Voltage
OGC	Open Geospatial Consortium
PD	Peak Detection
PP	Power Prediction
PPBEC	Power Prediction and Building Efficiency Characterizations platform
PV	Photovoltaic
r^2	Coefficient of determination

RES	Renewable Energy Sources
REST	Representational State Transfer
RMSD	Root Mean Square Difference
ROI	Region Of Interest
RTS	Real-time simulator
S	Standard Error of Regression
SIL	Software in the Loop
SIM	System Information Models
SMAPE	Symmetric mean absolute percentage error
SOD	Status Outlier Detection
TCP	Transmission Control Protocol
TMY	Typical Meteorological Year
UDP	User Datagram Protocol
WFS	Web Feature Service
WIA	Willmott’s Index of Agreement
WMS	Web Mapping Service
WPS	Web Processing Service
WSN	Wireless Sensor Network

Chapter 1

Introduction

During the international conference on climate changes (COP21) in 2015, the 196 parties attending the conference in Paris highlighted the need of reducing greenhouse gas emissions [1]. Urbanizations are largely energy-intensive as reported by the UN habitat division, cities consume about 75 % of the global primary energy supply and are responsible for about 50-60 % of the world's total greenhouse gases [12]. Moreover, the majority of the consumed energy is still supplied by fossil fuels (coal, oil and gas). In 2016, more than half of the overall world's population is living in urban areas. Projections states that by 2030, urban areas will host around 60 % of people globally and one third of the population will live in cities with at least half a million inhabitants [13]. On this regard, in the last years, many countries are providing incentives to promote the deployment of low-carbon and sustainable energy production technologies [2], generation such as Photovoltaic (PV) Systems. In order to achieve a reduction of greenhouse gas emission an increasing installation of Renewable Energy Sources (RES), Distributed Generation (DG) and an optimization of consumption with a smart use of energy in our cities [3] are required. ICT technologies, in particular Internet of Things, enables the possibility controlling and optimize consumption [4] hence increasing energy efficiency. These renewable and sustainable generation technologies are often connected to electricity distribution networks in the form of distributed generation (DG), at the low voltage (LV) or medium voltage (MV) level. The introduction of DG, which can be based on renewable or on conventional sources, substantially modifies the operation of electricity networks: the direction of power flows in the MV lines and even in high voltage/medium voltage (HV/MV) transformers can be reversed, voltage profiles are modified, fault management is affected [5, 6], etc. For all these reasons, distribution networks need to become *Smart* and new control strategies, algorithms and technologies need to be tested and validated before their implementation and installation in real systems. Moreover, DSOs (Distribution System Operators) are also enhancing the ICT (Information Communication Technologies) layer used to monitor and control distribution networks. As pointed out by [14], an in-depth simulation and analysis of Multi-Energy-Systems (MES) is required to increase the flexibility of energy systems by integrating different resources for both electric and thermal energy. Furthermore, ICT and MES offer valid options to foster novel services for smart energy management. For example they can foster events of Demand Response (DR) and Demand Side Management (DSM) by integrating buildings equipped with heat pumps, CHP (Combined Heat Power) or HVAC (Heating, Ventilation and Air Conditioning) systems [15].

Thus, specific tools to evaluate resource availability, uncertainty of RES and smart energy policies are required: *i)* to understand their impact on power grids; *ii)* to perform load balancing; *iii)* to perform storage planning and management; *iv)* to perform demand-side management at different scales, from single user up to district or city; *v)* to provide generation profiles to electricity markets (e.g. day-ahead or intra-day market). By analysing the generation loads in fine-grained spatio-temporal domain (e.g. sub-hourly simulations of Photovoltaic systems at district scale), such tools should be able to overcome the current techniques in estimating RES generation and integration that do not perform fine grained spatio-temporal simulation and do not integrate (*near*) real-time environmental data coming form the field.

In this context, ICT play a crucial role in both planning and monitoring of distributed energy sources. The crucial roles of ICT and the emerging Internet-of-Things (IoT) are highlighted by the spread diffusion of heterogeneous and pervasive sensors in our houses, district and cities. IoT devices and sensors allow to collect large amounts of energy related data capable of describing the consumption behaviours of the citizens. Hence, the increasing presence of sensors calls for the development of distributed software infrastructure for exploiting such IoT devices for data management and collection. Furthermore, IoT devices enables the possibility of monitoring devices and system in order to develop models for the simulation and optimization on energy process. For example, electricity consumption data can be used in simulation processes for evaluating: *i)* energy management actions; *ii)* management of electricity distribution networks; *iii)* integration of renewable sources in the city.

This Thesis presents a distributed infrastructure, called SMIRSE, for modelling and simulating renewable energy sources and smart policies integration in urban districts. Strong effort has been done in developing a solution for simulating Photovoltaic systems with good accuracy and for accurate planning of new installations. SMIRSE is able to model and simulate Multi-Energy-Systems in urban context by exploiting a modular architecture, that increases flexibility and scalability. Its modularity allowed the interfacing and integration between dedicated Real-time Grid Simulator, software simulation modules and real-time data in order to model the grid behavior.In order to achieve the challenges presented in Chapter 2 the study is focused in the implementation of a modular infrastructure build with a micro-services approach that exploits Internet of Things communication protocols (see Chapter 4). This approach enabled interoperability between hardware and software components of the SMIRSE platform and at the meantime its modularity, extendability and scalability (see Chapter 5.1.2.2). New modeling and simulation tools for *i)* Solar energy simulation, with a focus on Photovoltaic systems; *ii)* Integration of RES and smart policies with the distribution grid; *iii)* Characterization of thermal performance of Buildings and power consumption prediction; and *iv)* Buildings indoor temperature simulation and monitoring, have been designed, developed and integrated upon the backbone of the microservices-based infrastructure.

The main advantage of the SMIRSE infrastructure is its capability to simulate new systems, which are different from previous tests in terms of input data or type of model, with a minimum effort. Any changes in the input data of the modules (i.e. new consumption profiles; new grid; or new generation scenario) as well as changes in the type or conceptual model of a module (e.g. new prosumer behaviour model; other types of generators, such as PV, wind, fuel cells; and new control and management algorithms) require a minimum effort to be implemented without creating the entire simulation model from scratch. In addition, the proposed infrastructure enables to study the interoperability among different use-cases in a plug-play fashion. Finally, the proposed solution can integrate Smart Metering Architecture to exploit (near-) real-time data collected from the field to co-simulate different smart energy strategies with real information. For example, the SMIRSE has been used to co-simulate the integration of Photovoltaic Systems and battery management within the distribution grid. These are the strengths of the proposed solution: the platform is flexible and open to include, replace, or enhance the modules for any new use cases/scenarios.

The methodology and results presented in the following chapters have been partly published in the papers listed in Appendix A. The rest of the thesis is organized as follows: Chapter 2 presents the motivation and challenges of the research, Chapter 3 presents a review of existing solution for modeling and simulation of RES and smart energy policies, Chapter 4 presents a background on enabling technologies to develop the infrastructure, Chapter 5 presents the SMIRSE platform implementation, Chapter 6 presents the results achieved exploiting the platform and finally Chapter 7 presents the concluding remarks.

Chapter 2

Motivations and Challenges

This research aimed at developing a distributed infrastructure to model and co-simulate renewable energy systems and smart energy policies integration in urban context. SMIRSE combines together different technologies and heterogeneous information to model the energy flows and to simulate the impact of novel control strategies in cities and distribution networks. Exploiting information coming in (near-) real-time from Internet connected devices installed across the city. On these premises, SMIRSE is an infrastructure for simulations as a service that can be used by different stakeholders to build and analyse new energy scenarios for short- and long-term planning activities and for testing and managing the operational status of Multi-Energy-Systems. Examples of scenarios that combine together thermal and electricity trends (load and/or generation) to simulate the energetic behaviour of buildings, districts and cities are: i) Installation of Renewable Energy Sources, ii) Grid reconfiguration, iii) Demand Response and iv) Demand Side Management.

To achieve this purpose, SMIRSE needs to combine novel or already existing modelling and simulation tools together with a real-time grid simulator. At the same time, it needs to correlate heterogeneous information, such as: i) measurements retrieved in (near-) real-time from IoT devices deployed across the city (e.g. information on multi-vector energy trends, weather, indoor temperature in buildings, status of the distribution grids); ii) Building Information Models (BIM), grid models and Geographical Information Systems (GIS); iii) topology of energy distribution networks; and iv) urban cartographies. Integrated systems such as BIM and GIS are need in order to represent the environment of the simulation. In particular BIM are a parametric representation of buildings where all the elements of the building are represented with their physical features. GIS provides fundamental information for the representation of a district such as: Digital Surface Model (DSM), that are used to reconstruct shape of buildings in 3-D; Cadastral maps that contains geometry objects representing the building plant.

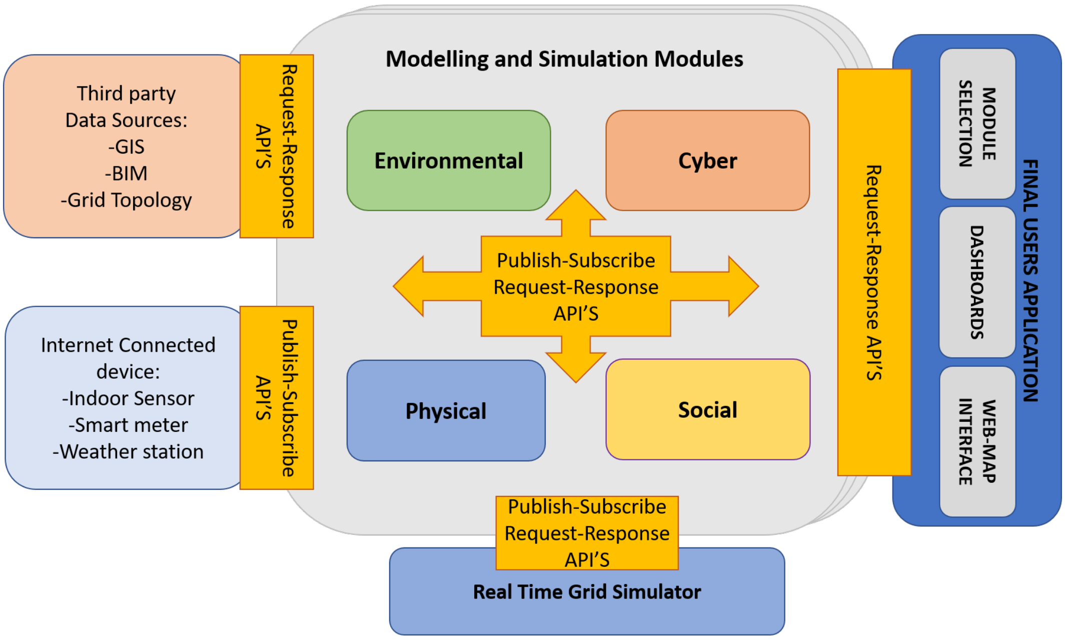


Fig. 2.1 High Level Schema of SMIRSE Platform

To realize the SMIRSE infrastructure, we identified the following key challenges from a literature review [14–18] that needs to be addressed:

i) *Multi-Layer-System*: Smart urban districts are complex systems where different entities cooperate by exchanging heterogeneous information. In this view, Bompard et al. [19] theoretically conceive a multi-layer system with: i) a *Physical layer* for including hardware components to monitor and manage the grid; ii) a *Cyber layer* for defining information flows and managing operations; iii) a *Social layer* for including unpredictable performances due to actors that play in power system scenario (i.e. users, prosumers and system operators); and iv) an *Environment layer* for integrating natural phenomena that affects the previous layers.

ii) *Simulation of Renewable Energy Production*: SMIRSE has to provide modules to simulate the energy production of RES with a fine grained spatio-temporal resolution.

iii) *Simulation of buildings dynamics*: SMIRSE has to provide features for analysing both thermal and electrical dynamics in buildings. For example, modelling and simulating thermal dynamic includes also the analysis of indoor temperature variations related to power consumption. In this view, information on thermal inertia and/or heat storages can be given as input to control policies for shaving demand peaks in district heating networks [20, 21] or for DR and DSM if heating and cooling systems are supplied by electric generators or CHPs.

iv) *Simulation of novel energy management policies*: Novel control policies needs to be evaluated in a realistic environment before being applied in a real-world context. Thus, the effects in terms of energy efficiency, energy optimization, distribution network reliability and economic value can be evaluated in-depth.

v) *Simulation of distribution networks*: SMIRSE must be able to simulate the energy distribution network to provide energy management policies with information on the status of the network itself. For example, from these simulations possible congestions, failures and unbalances can be evaluated in a realistic scenario. For this purpose, simulators like OPAL-RT and RTDS need to be integrated in the infrastructure to perform real-time simulations of the distribution network with microseconds time-steps.

vi) *Simulations with different spatio-temporal resolutions*: SMIRSE has to provide features to simulate energy phenomena with different time and space resolutions. Time resolution ranges from the microseconds, for analysing the operational status of distribution systems, up to years, for planning and refurbishment activities. Whilst, space resolution ranges from the single dwelling up to districts and cities.

vii) *(Near-) real-time integration of real-world information*: Real-world information sent in (near-) real-time by heterogeneous Internet connected devices are needed to develop more accurate event-based models for analysing the operational status of the grid, for developing ad testing more efficient control policies and for planning and refurbishment activities [22].

viii) *Modularity and extendibility in integrating data, models and simulators*: Modularity and extendibility are two main features for Multi-Energy-Systems. In particular, SMIRSE needs to be designed to integrate in a plug-and-play fashion heterogeneous data-sources, models and simulators. This makes the overall infrastructure suitable for simulating different energy scenarios, becoming a general purpose framework for energy simulations in cities. Modularity and extendibility are also two main requirements to allow future extensions with low cost and small architectural impacts.

ix) *Scalability of the infrastructure*: Horizontal and vertical scalability of the infrastructure is another key requirement of SMIRSE. Indeed, it needs to scale up quickly and easily because simulating a city or a district implies the interaction of thousand of concurrent entities. This becomes critical if real-time simulations of power distribution network must be performed.

Figure 2.1 show a high-level overview of the architectural style for building a simulation infrastructure that accomplish to the identified challenges. On the left side of the schema the main inputs of the infrastructure are represented. Thanks to the development of Request-Response and Publish-Subscribe API'S the SMIRSE integrates third party data-sources (BIM, GIS, GRID topology,...) and Internet connected devices (Indoor sensors, Smart meters, Weather Stations). In the core of the infrastructure are placed the Environmental, Physical, Social and Cyber Modules representing a Multi-Layer-Systems. Those modules are able to exchange data among each other thanks to the development of Request-Response and Publish-Subscribe API'S present in the Cyber layer. Such approach enables the composability and modularity of the platform. On the bottom of the modelling and simulation module a Real-time grid simulator is connected to the Simulation and Modelling modules thanks to Request-Response and Publish-Subscribe API's in order to simulate the grid behavior. On the right side the Final Applications are reported. Thanks to Request-Response API'S the user choses the simulation and modelling modules and can see results in forms of Dashboards and Thematics Maps.

Furthermore, this solution is intended to satisfy the needs of different end-users such as:

- ***Single citizen*** can evaluate the economic and environmental savings achievable with the installation of RES; Hence he/she wants to be aware of the avoided CO_2 emission and virtuous actions he/she can perform.
- ***Energy aggregators and Energy Communities*** can use the simulations to schedule consumption of their clients for maximizing self-consumption and minimizing energy bills. In particular ***Energy Communities*** can exploit such infrastructure to perform feasibility studies as proposed in our previous research [23];

- ***Distribution system operators*** (DSO) can take advantage of the proposed solution for network balancing and for planning retrofits and/or extensions of the existing distribution grid;

- ***RES engineers*** can simulate the behaviour of converters with the application of realistic conditions. This simulation helps in dimensioning, validating and optimizing each system before and after installation;

- ***Energy and City planners*** can exploit the infrastructure for evaluating the impacts of Renewable Energy sources installations or for monitoring the performance of existing ones.

Chapter 3

Related Work

In the last years, the study of Multi-Energy-Systems (MES) is becoming crucial to de-carbonize energy production and also to foster a widespread deployment of RES. Geographic Information System (GIS) is considered useful tool to plan the deployment of renewable energy sources, such as solar, wind and biomass systems [24]. Particular emphasis is given to such technology for modelling solar potential in urban environments [25]. To achieve it, there is the need tools for an in-depth analysis and simulation of MES for both electrical and thermal energy [14].

The following sections report first the state of the art solution for co-simulation of MES with a particular focus on Photovoltaic energy simulation. The latter present the innovation of the SMIRSE with respect to MES co-simulation and Photovoltaic energy solutions.

3.1 MES co-simulation and Photovoltaic simulation solutions

DER-CAM [26] is a useful tool for planning and operational analysis of power distribution networks. It aims at providing guidelines for future investment. The input can be given with a resolution up to 5 minutes.

HOMER [27] helps on studying different micro-grid configurations based on hourly input data.

EnergyPLAN [28] is another solution useful for both operational and planning activities. It receive input data up to hourly values. However, none of these solutions provides features for detailed power flow analysis or thermal simulations in buildings. Moreover, they are not flexible in integrating new scenarios in the simulation process and they do not exploit data coming in (near-) real-time from real devices installed across the city.

GRIDSpice [29] is a distributed platform that co-simulate power flows and data communication in smart-grid scenarios. It integrates third-party software like MATPOWER and GridLAB-D to simulate power generation, demand and distribution. It exploit a cloud-based architecture to parallelize the computation of large scale models. Also in this case, GRIDSpice neglects thermal simulations in buildings and does not exploit (near-) real-time information from real devices.

SGsim [30] integrates two tools for simulating both power and communication flows to co-simulate smart grid applications, such as Conservation Voltage Reduction.

DIMOSIM [31] is a platform to perform MES simulations in urban districts. It enables thermal simulations in buildings but it lacks of electrical flows simulations in power grids.

MOSAIK [32–34] is a flexible architecture consisting of four layers for managing control strategies, scenario specifications and simulation models exploiting semantic knowledge. In particular, to perform power-flow analysis in a smart grid scenario, [34] integrates a co-simulation between the software *PowerFactory* with their model (in MATLAB®) for both PV and Load generations.

IDEAS [35] is an open source platform based on Modelica modelling language. It co-simulate Demand Side Management strategies where thermal request of buildings affects power distribution networks.

MESCOS [15] is a co-simulation platform for district energy systems. It simulates Demand Response and Demand Side Management policies by integrating both electrical and thermal loads.

The main limitations of these solutions are summarized as follows: i) they do not integrate (near-) real-time information from real devices; ii) they do not exploit a real-time simulator (e.g. OPAL-RT and RTDS); iii) the integration with other simulation tools is not easy. In addition to that, MOSAIK lacks in simulating thermal behaviors in buildings.

The three co-simulation tools MOSAIK, GRIDspice and SGsim offer a flexible and scalable simulation framework but limited to SIL (Software in the Loop), without the possibility of HIL (Hardware in the Loop). Furthermore, they do not provide real-time features for short-transient phenomena.

HUES [36] platform aims at facilitating the integration of different models for MES analysis. It implements a repository layer that includes all the platform modules whose functionalities are described in a semantic wiki. However, HUES neglects on an interconnection among the platform's modules and lacks on integrating data coming in (near-) real-time from devices installed across the city.

In [37], authors present INSPIRE, a co-simulation framework for evaluating real-time power systems tougher with ICT. It is based on both the IEC 61850 standard and the IEC 61968/61970 common information model. Thus, INSPIRE simulates the interactions between the *Physical* layer and *Cyber* layer.

In [38], authors present a distributed platform for real-time co-simulation of Demand Response events in microgrids. The platform integrates the OPAL-RT simulator and exploits information coming in real-time from real devices. However, the platform does not simulate thermal behaviors in buildings.

Yang et al. [39] propose a co-simulation environment for HIL/SIL (Hardware-in-the-loop/Software-in-the-loop) validations of distributed controls in Smart Grid. They designed the controls by exploiting a model-driven approach using the IEC 61499 standard. Controllers are developed in MATLAB/Simulink® and communicate with power plants models through UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) sockets to build closed-loop models. Authors highlight that a co-simulation approach is extremely useful in the context of distributed automation.

In [40], authors present a specific HIL/SIL co-simulation platform to verify the performance of a volt-VAR optimization engine for smart distribution networks. In its core, it integrates a real-time simulator for distribution networks with physical devices for Measurement & Control. For message exchanges in the platform, they implemented both IEC 61850 MMS and GOOSE protocols.

Another example of multi-layer co-simulation platform for complex-system analysis is presented in previous works [41, 42]. This solution is devoted only to evaluate the impacts of PV distributed generation systems, especially in densely populated urban areas. It implements *Physical*, *Cyber* and *Environmental* layers.

In [43] the authors presents a co-simulation testbed for microgrids. They integrate a RTS with a communication emulator in order to test control algorithms for microgrid.

In the framework of smart grids real-time simulation, ENEL, the biggest Italian utility, is running test and validation projects in some of its smart grids laboratories. POI P3 Smart Grid project is an example of these activities [44], with the purpose of testing new techniques for voltage regulation through SCADA and DMS. All devices and systems in this project communicate only via IEC 61850 standard. Lack of forecasted power-time profiles of the loads and generators was the other limitation of its test set up implementation.

Finally in [45], authors presents a flexible co-simulation environment that exploits an IoT approach. Indeed, information from IoT devices (e.g. smart meters) has feed into a real-time simulator through the publish/subscribe communication paradigm [7].

With respect to Photovoltaic energy simulation the authors in [46] and [47] exploit GIS tools to estimate the yearly Photovoltaic (PV) potential starting from aerial and satellite images. Another approach consists in exploiting *Digital Surface Models* (DSMs) or 3D city models obtained from LiDAR data. DSM represents the earth's surface and includes all objects and buildings on it. For example starting from DSM, Hofierka et al. [48] estimated monthly and yearly solar potential in urban areas using *r.sun* tool [49]. Following this approach, Jakubiec et al. [50] developed a methodology to predict PV electricity gains using LiDAR data combined with DAYSIM simulations. Lukač et al. [51] also used LiDAR data and developed a new algorithm for evaluating solar radiation on rooftops. Finally, Camargo et al. [52] have modelled PV integration into a urban grid by exploiting *r.sun*. Still exploiting LiDAR data, Brito et al. [53] estimated yearly PV potential in Lisbon using ESRI *Solar Analyst* tool. These solutions just perform a time-domain analysis to estimate yearly solar potential. However, to provide more precise PV estimations, both space and time domains must be taken into account [52, 50, 51]. In [25] and [24], authors highlight that the integration of these two domains, with higher time and space resolutions, is needed to better understand spatio-temporal dynamics in energy systems models. This is required to *i)* plan deployment activities; *ii)* evaluate business plans; *iii)* monitor existing plants and *iv)* promote smart energy use.

On this premises, in order to provide simulation tools to wide range of users, GIS solutions have been developed exploiting a Web Service approach. Li et al. [54] have developed a service-oriented environment for sharing geoscience algorithms. They exploited both SOAP (Simple Object Access Protocol) technologies and OGC (Open Geospatial Consortium) standards to make available GRASS-GIS [55] features trough Web Services. *Gwass* [56] is a distributed web-based GIS built on top of the GeoBrain Web Services. This platform exploits a service-oriented architecture to offer an alternative to commercial desktop solution. Literature provides also web-based solutions [57–64] to give PV energy potential information and to foster assessments of environmental and economic benefits as pointed out by Freitas et al. [25]. PVWatts [59] is a web application developed by the National Renewable Energy Laboratory that estimates yearly (Y), monthly (M) and hourly (H) PV production using a *Typical Meteorological Year* (TMY) and a topographic model of $40km^2$. PVGIS [57, 58] is a solar web map that offers information on yearly and monthly PV production in Europe and Africa. It provides also sub-hourly radiation information in clear-sky conditions. To perform this computations, it exploits *r.sun* starting form a DSM with a resolution of 1 *km*. i-Guess [61] is a web based system for urban energy planning in smart cities. It provides maps for yearly solar radiation on rooftops and for yearly PV potential. Mapwell Solar System [60] computes solar radiation and PV potential considering also a TMY. In its core, it exploits the methodology described in [50] to perform simulations. It also provides information on rooftops and Region Of Interest (ROI). I-SCOPE [62] is an integrated platform to give 3-D smart-city services. In particular, it offers a solar map with Yearly and Monthly PV potential. Finally, Brumen et al. [63] developed a web application for PV potential assessments by exploiting *r.sun* starting from DSM. This platform provides data about yearly and monthly PV Potential together with information on rooftops and ROI. Such information on rooftops and ROI are also provided by I-SCOPE and i-GUESS services.

3.2 Innovation of the SMIRSE platform

The presented literature solutions are designed and implemented to address a single use-case or scenario with specific requirements, while the SMIRSE co-simulation platform is designed to be flexible in simulating different use cases and scenarios in power grids. In addition, SMIRSE integrate Internet-of-Things (IoT) communication paradigms and protocols to provide flexibility and scalability in performing HIL and SIL co-simulations. Thus, SMIRSE is ready to enable the communication among next generation devices (e.g. Smart Meters), that will be IoT devices. In a nutshell SMIRSE exploits communication paradigms peculiar of IoT platforms, to implement a modular framework where different power grid scenarios and simulations can be executed. It includes *Physical*, *Cyber* and *Environmental* layers depicted in [19] providing features to perform both HIL and SIL co-simulations. In addition, it integrates Smart Metering Infrastructure (e.g. [65, 66]) to retrieve measurements from real devices deployed along the distribution network.

To highlight the contribution, Table 3.1 reports a comparison of SMIRSE solution with the reviewed smart-grid co-simulation environments. It highlights: i) what are the simulation layers among *Physical*, *Cyber* and *Environmental* that each solution integrates in the simulation environment; ii) what kind of co-simulation is performed; iii) if a Real-time simulator is used; iv) if the solution simulates single or multiple use-cases/scenarios; v) if HIL and/or SIL simulations can be performed; and vi) if the simulation environment interacts with IoT devices.

With respect to Photovoltaic simulation the main limitation of presented solutions consists on overlooking a fine-grained spatio-temporal domain in simulating and modelling energy production and performance of PV systems. Indeed, they are mainly focused on spatial domain by performing yearly or monthly simulations. On the other hand, existing spatio-temporal solutions perform hourly simulations with a low-resolution in spatial domain (e.g. resolution $> 1m$). To provide more accurate estimations, hourly and sub-hourly simulations with fine-grained resolution (e.g. DSM with resolution < 1 *m*) are instead needed. This hight-resolution DSM allows to recognize and exclude encumbrance in rooftops, such as chimneys and dormers. Moreover, such simulations have to take into account real-sky conditions. To do so, they need real weather data (e.g. data from personal weather stations [24]) to compute incident radiation on tilted surface of rooftops and estimate PV performance and energy production.

Table 3.1 Comparison among our co-simulation infrastructure and literature solutions.

Solutions	Integrated layers	Co-Simulation	RTS	Use-case Scenario	HIL/SIL	IoT
DER-CAM [26]	Physical	Up to 5 minute resolution of physical energy-systems.	x	Multiple	x	x
HOMER [27]	Physical	Hourly simulation of micro grid energy-systems.	x	Multiple	x	x
EnergyPLAN [28]	Physical	Hourly simulation of MES energy-systems.	x	Multiple	x	x
GRIDspice [67]	Physical Cyber	Co-simulation of power- and communication-flows in smart-grids.	x	Multiple	SIL	x
SGsim [30]	Physical Cyber	Co-simulation of power- and communication-flows for smart-grids application such as CVR.	x	Multiple	SIL	x
DIMOSIM [31]	Physical	Co-simulation of MES no electrical power flows.	x	Multiple	x	x
MOSAIK [32–34]	Physical Cyber	Co-simulation of power flow and load generation.	x	Multiple	SIL	x
IDEAS [35]	Physical	Co-simulation of demand side management with thermal simulation of buildings.	x	Single	x	x
MESCOS [15]	Physical	Co-simulation of demand side management with thermal simulation of buildings.	x	Single	x	x
HUES [36]	Repository	Repository of simulation models.	x	Multiple	x	x
INSPIRE [37]	Physical Cyber	Real-time co-simulation of Cyber-physical energy-systems.	x	Multiple	x	x
Yang et al. [39]	Physical Cyber	Real-time co-simulation between two simulation environments (MATLAB and Function Block).	x	Single	HIL SIL	x
Manbachi et al. [40]	Physical Cyber	Real-time co-simulation of grid status and volt variation controllers.	✓	Single	HIL SIL	x
Bottaccioli et al. [41, 42]	Physical Cyber Environmental	Real-time co-simulation of PV energy production and grid status.	✓	Single	HIL SIL	x
Hahn et al. [43]	Physical Cyber	Real-time co-simulation of power- and communication-flows to test control algorithms in micro-grids.	✓	Multiple	HIL SIL	x
ENEL [44]	Physical Cyber	Real-time simulation of protection and automation strategies.	✓	Multiple	HIL SIL	x
SMIRSE solution	Physical Cyber Environmental	Real-time co-simulation of smart grid control algorithms and building thermal loads.	✓	Multiple	HIL SIL	✓

Table 3.2 summarize the main features of the relevant literature solutions for Photovoltaic energy simulation with respect to SMIRSE .

Table 3.2 Comparison between the proposed architecture and relevant literature solutions

	Simulation Step			**Sub-hourly Clear-sky simulation**	**Sub-hourly Real-sky simulation**	**Rooftop and/or ROI details**	**Weather Station data integration**	**Distributed and modular architecture**	**REST API**
	Y	**M**	**H**						
SMIRSE	✓	✓	✓	✓	✓	✓	✓	✓	✓
PVWatts [59]	✓	✓	✓						
PVGIS [58]	✓	✓		✓					
i-GUESS [61]	✓								
Mapdwell [60]	✓					✓			
I-SCOPE [62]	✓	✓				✓			
Brumen [63]	✓	✓				✓			

As such, the contribution (presented in Section 5.2.1) with respect to state-of-the-art in solar energy simulation, detailed in Table 3.2, includes the followings innovating aspects: *i)* analysing together spatial and temporal domain in fine-grained resolution; *ii)* providing real-sky sub-hourly simulations, with 15-minutes time intervals; *iii)* integrating real meteorological data gathered from (personal) weather stations.

Summarizing the strength of SMIRSE infrastructure comes from its capability to host new different use cases, input data, or models by means of its modules in a plug-and-play fashion, with a minimum effort to change the overall setup. As an example, several distributed generations (DGs) can be integrated as either independent simulators or standalone physical generators thanks to our IoT-based architecture which is a novel contribution to the co-simulations of electric energy systems with respect to conventional setups.

With respect to literature solutions, SMIRSE is a distributed infrastructure for modelling and simulating renewable energy sources and smart energy policies integration. It aims at overcoming the highlighted limitations and addressing the main challenges identified in Section 2 to evaluate general purpose simulation scenarios. In particular, SMIRSE performs simulations for both thermal and electrical energy systems with different spatio-temporal resolutions. It exploits the OPAL-RT real-time simulator that allows in-depth simulations with microseconds time-steps. It provides features to perform detailed power flow analysis, thermal simulations in buildings and evaluation of RES impacts on the grid. Furthermore, SMIRSE integrates data coming in (near-) real-time from real devices installed across the city. Finally, it eases the interconnection among SMIRSE components and third-party models and simulators in a plug-and-play fashion.

Chapter 4

Background on Enabling Technologies for distributed infrastructures

This Chapter introduces the adopted technologies to enable the development of distributed infrastructure for simulation and modeling renewable energy source and smart policies integration in urban district. The following Sections will present a description of the:

- Microservice approach used in the development of the SMIRSE distributed infrastructure;
- IoT communication protocols that have been used for enabling the integration and the communication between the different modules of the SMIRSE solution
- Open Geospatial Consortium standards for the development of web service for Geographic Information Systems (GIS) data and process

4.1 Microservices

In the last year many companies and researchers are starting to adopt a microservices architectural style in developing their platforms and application. The microservices approach consists on developing software *as a suite of small services, each running* *in its own process and communicating with lightweight mechanisms* [68]. This increases flexibility and maintainability because services are *small, highly decoupled and focus on doing a small task* [69].

Krylovskiy et al. [70] have reported the main characteristic of the microservices approach for distributed infrastructures for smart-city and smart-grid scenarios, which are described in the followings:

- **Componentization via Services.** Thanks to the approach is possible to have system composed by single services that are independently, replaceable, upgradeable, and deployable. There are no function calls that use in-memory but the components of the system interacts with web-serivce calls.
- **Organization around Business Capabilities.** As reported by [19] Organization have an important impact on system design. Microservices architecture stimulates organization on business capabilities with respect to a standard layer approach. As reported by [68], microservices approaches are deployed with cross-functional teams by preventing the ”logic everywhere” soiled architectures.
- **Smart endpoints and dump pipes.** The use lightweight communication protocols in the massage exchange process keeps their domain logic internal. With respect to monolithic approach where the communication mechanism provides sophisticated functionality for message transformation and choreography, the business logic in microservices architecture always remains in the endpoints the services.
- **Decentralized Governance.** Due to the fact that the services are independently deployable in the microservices approach there is less need of a centralized governance in standards and technology platforms. The development of the whole systems do not relay on a single technology for each service. This provides the possibility of choosing the right technology for each job.
- **Decentralized Data Management.** Microservices approach provides the possibility of using a decentralized data management system for each service. Hence there is no need of a unique data management system but every service has its own database that fulfills in the best way the job.

- **Evolutionary Design.** In the microservices design the decomposition is used to enable frequent and controlled changes in the systems. This can affects in two aspects: *i)* the possibility of having independently deployable and replaceable components allows to reimplemented services without affecting the rest of the system and *ii)* the changes that might happen inside a services are limited because the service provides as small task.

Newman [69] summarizes the benefits of microservices design in the following:

- **Heterogeneity in the system technology**: Decentralized governance and data management allows coexistence of different technologies used by different components in the system.
- **Resilience**: the components of the system have clear boundaries giving the possibility to easily isolate failures and gradually degrade the system functionality, as well as update and deploy individual services independently.
- **Scalability**: The microservice design pattern allows to easy scale the single components of an infrastructure without scaling up the whole system.
- **Composability**: due to the componentization in the microservice approach is possible to create new system capabilities by composing and re-using existing services.

Hence, microservices are not really a enabling technology but more an architectural approach that eases the implementation of a flexible, scalable and distributed infrastructure.

In Figure 4.1 Daya et al. [9] reports the main difference between microservices and monolithic approaches.

4.2 IoT communication Protocols

The communication protocols used in the development of the SMIRSE distributed infrastructure are based on common IoT communication protocols. Request/response and Publish/subscribe communication paradigms have been adopted. In particular REST (Representational State Transfer) architectural style, in the form of web services, and the MQTT protocol are used.

Table 4.1 Comparison between microservices and monolithic approach [9]

Category	Monolithic architecture	Microservices architecture
Code	A single code base for the entire application.	Multiple code bases. Each microservice has its own code base.
Understandability	Often confusing and hard to maintain.	Much better readability and much easier to maintain.
Deployment	Complex deployments with maintenance windows and scheduled downtimes.	Simple deployment as each microservice can be deployed individually, with minimal if not zero downtime.
Language	Typically entirely developed in one programming language	Each microservice can be developed in a different programming language.
Scaling	Requires you to scale the entire application even though bottlenecks are localized.	Enables you to scale bottlenecked services without scaling the entire application.

4.2.1 Representational State Transfer architectural style

Representational State Transfer (REST) *is a coordinated set of architectural constraints that attempts to minimize latency and network communication, while at the same time maximizing the independence and scalability of component implementations* [71]. Fielding in its PhD dissertation [72] starts describing the REST architectural style by defying the constraints of the REST approach.

- **Client-Server**: The main constrain is the separation of concerns between client and server. By separating user interface concerns from data storage consolers the portability of the user interface is increased. Thus increases scalability at the same time due to the simplification of the server components.

- **Stateless**: The second constrain is a stateless communication between Client-Server interaction. Each request from client to server contains all the information needed by the server to understand the request. This constrains increases Visibility because there is no need to look beyond a single request datum. Scalability is achieved because there is no need to store state between requests and the server can easily and quickly free resources.

- **Cache**: A cache constrain is added in order to reduce client-server interaction. Thus increases scalability and performance.
- **Layered system**: Each component of the system cannot see beyond the immediate layer with which they are interacting. Hence, a client cannot tell if he is directly connected to the end server or to an intermediate. This increases the scalability of the system by enabling load balancing of services across multiple networks and processors.
- **Uniform Interface**: This constrain is the main characteristic of REST architecture. By providing a Uniform interface the system architecture is simplified and the visibility of interaction is improved. Uniform interface is achieved by defining the following interface constraints: *i)*identification of resources; *ii)*manipulation of resources through representations; *iii)*self-descriptive messages; and *iv)*hypermedia as the engine of application state.

In REST architecture the components communicate by transferring a representation of the data in a format matching one of an evolving set of standard data types. The format is then selected dynamically based on the capabilities or desires of the recipient and the nature of the data.

In order to encapsulate the activities of accessing resources and transferring resource representations REST takes advantage of connector types. Thus, connectors provide an abstract interface for component communication. This increases the simplicity thanks to a clean separation of concerns and by hiding the implementation of resources and communication mechanisms. Thanks to the generality of the interface new implementation can replace existing ones without impacting the other components of the system.

4.2.2 Publish/Subscribe communication approach

Distributed infrastructure, as the own presented in this Thesis, involves a large amount of simulation models, sensors end services calling for a dynamic communication schema. *The publish/subscribe interaction paradigm provides subscribers with the ability to express their interest in an event or a pattern of events, in order to be notified subsequently of any event, generated by a publisher, that matches their registered interest [7].*

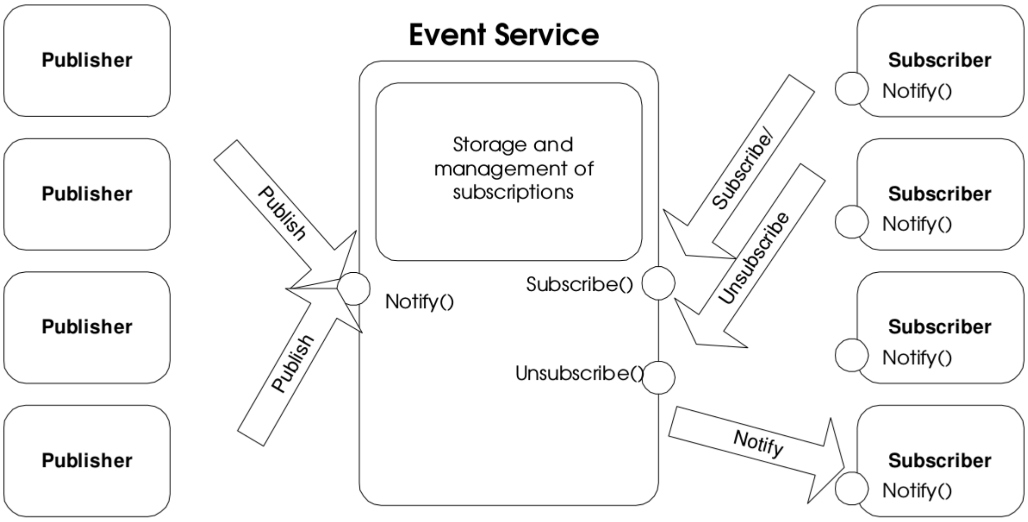


Fig. 4.1 A simple object publish/subscribe [7]

Figure 4.1 shows a basic system model for publish/subscribe interaction. The model relies on an event notification service providing storage and management for subscriptions and efficient delivery of events. The *Event Service* acts as a neutral mediator between publishers, that produces events, and subscribers, that consumes events. A subscriber registers its interest in a topic of events by calling a ***subscribe()*** operation to the *Event Service*, with no knowledge on the sources of these events. On the other side the publisher has no information on how is consuming its informations.

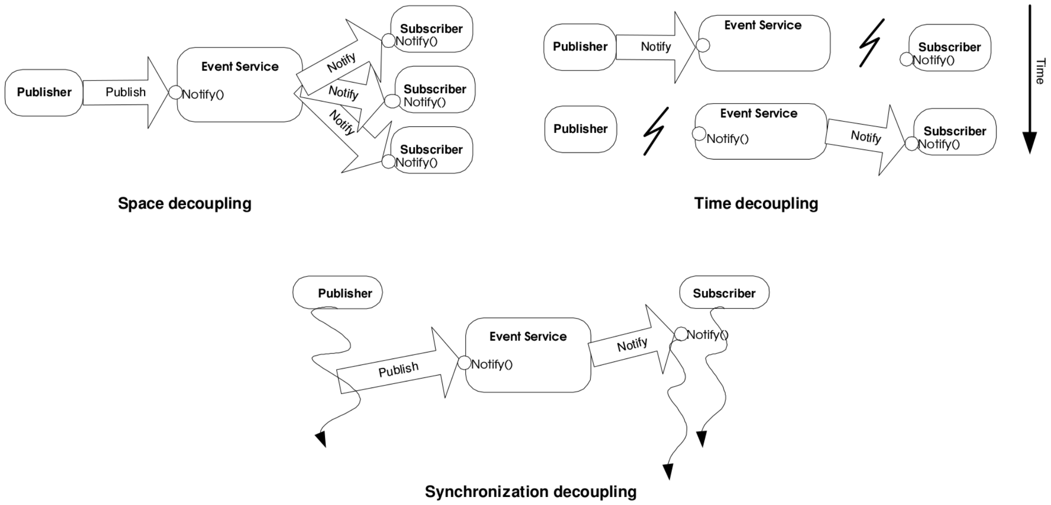


Fig. 4.2 Space, time and synchronization decoupling with the publish/subscribe paradigm [7]

This communication schema provides the possibility of decoupling in three dimension increasing the scalability, as stated by [7] and shown in Figure 4.2:

- Time decoupling: Publisher and subscriber cannot participate to the interaction at the same time. For example a publisher can send a message meanwhile the subscriber is disconnected. The subscriber can get notified by the event after it gets reconnected meanwhile the publisher went off line.
- Space decoupling: Publisher and subscriber do not need to know each other.
- Synchronization decoupling: messages and exchanges between publishers and subscribers can be asynchronously.

4.3 Geographic Information Systems and Open Geospatial Consortium Services

GIS are crucial in the development of simulation and modeling of smart and renewable urban energy system for a variety of reasons [24]. Firstly, GIS provides a heterogeneous information for describing the environment of the analyzed district. In particular, GIS offer information on population distribution, buildings presence and characteristics, localization of sensors (such as Weather Data) and Digital Elevation Models for reconstructing the area morphology. Furthermore, GIS allows to perform accurate simulations in the territory for planning and evaluating the power production from renewable and distributed energy sources. Finally, GIS allows to build thematic maps that are crucial in presenting and visualizing results for planners and decision makers.

The Open Geospatial Consortium (OGC) [73] specifies the interfaces for publishing and performing geospatial processes over the web. In particular, it implements:

- Web Processing Service (WPS): It provides a standard for geospatial processing services describing how inputs and outputs can be structured. Furthermore, WPS defines standards for the execution of a process and for the process output from and to a client request. With this standard any geospatial process can be "wrapped" with a standard interface and integrated into existing workflows. WPS supports short and fast computational tasks and long and time consuming process exploiting asynchronous processing. Moreover, the WPS standard provides an interoperable description of processing functions by defining a general process model that is designed to support process cataloging and discovery in a distributed environment.

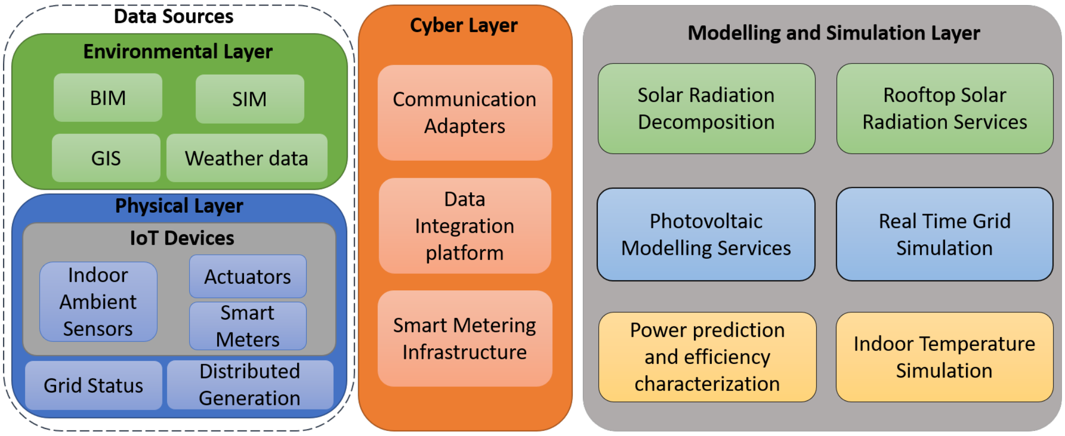
- Web Feature Service (WFS): specifies a standard for services that provides access and operations to GIS features abstracting from the underlying data store. Possible operations enabled by WFS are: *i) Discovery operations* provide the capability of the services and its application schema that defines the feature types of the service. *ii) Query operations* provides the possibility of retrieving features and/or values from the underlying data store based upon constraints, defined by the client, on feature properties. *iii) Locking operations* provides the access to features for the purpose of modifying or deleting features. *iv) Transaction operations* allow features to be created, changed, replaced and deleted from the underlying data store. *v) Stored query operations* allow clients to create, drop, list and described parameterized query expressions that are stored by the server and can be repeatedly invoked using different parameter values.

- Web mapping Service (WMS): standardizes a simple HTTP interface for retrieving GIS maps from one or more distributed geospatial databases. With a request WMS defines the geographic layer(s) and area of interest to be processed. The response provides one or more GIS maps (returned as JPEG, PNG, etc) that can be displayed in a browser application.

Chapter 5

The SMIRSE software infrastructure

In this chapter the SMIRSE distributed infrastructure for modeling and co-simulation of Smart and Renewable Energy Systems in urban districts (see Figure 5.1) is presented.



Data Sources
Environmental Layer
BIM
SIM
GIS
Weather data
Physical Layer
IoT Devices
Indoor Ambient Sensors
Actuators
Smart Meters
Grid Status
Distributed Generation
Cyber Layer
Communication Adapters
Data Integration platform
Smart Metering Infrastructure
Modelling and Simulation Layer
Solar Radiation Decomposition
Rooftop Solar Radiation Services
Photovoltaic Modelling Services
Real Time Grid Simulation
Power prediction and efficiency characterization
Indoor Temperature Simulation

Fig. 5.1 Schema of SMIRSE Platform

This infrastructure exploits the microservice design pattern [68, 69], described is Chapter 4, to increase both scalability and extendibility of the system, and to ease its maintenance. Indeed, the microservice approach defines software architecture as a set of loosely coupled and collaborating services. Thus, the solution is flexible in modelling and co-simulating different energy flows in a single solution made of different interoperable components or modules that can be deployed in a plug-and-play fashion. Moreover, the communication between the different components exploits both the request/response (e.g. REST web services [74]) and publish/subscribe [7] (e.g. MQTT protocol [75]) communication paradigms.

The proposed infrastructure consists of four layers. From left to right in Figure 5.1, both *Environmental and Physical Layers* includes the heterogeneous data-sources needed by the different components in the system. The *Cyber Layer* enables the communication among the different modules in the four layers by exploiting either the request/response or publish/subscribe [7] communication paradigms. The *Modelling and Simulation Layer* consists of different components that simulate energy phenomena.

5.1 SMIRSE Layers

5.1.1 Data sources

The proposed solution integrates heterogeneous data-sources needed by the simulation components. In particular, they have been grouped in two layers, *Environmental Layer* and *Physical Layer* (see Figure 5.1).

The **Environmental layer** integrates all the information needed to describe a city. Among the others this layer includes:

i) *Geographical Information Systems* (GIS) integrate georeferenced information about the different entities (e.g. devices, buildings and pipelines) in cities. It also includes cartographies cadastral maps and Digital Elevation Models.

ii) *Building Information Models* (BIM) are parametric 3-Dimensional models, where each model describes a building, both structurally and semantically.

iii) *System Information Models* (SIM) describe size and structure of energy distribution networks. SIM is built by exploiting parametric and topological data.

iv) *Weather Data* are retrieved by third party services, such as [76]. This information is georeferenced and collected by personal weather stations deployed in cities.

On the other hand, the **Physical layer** integrates data coming from physical systems and Internet connected devices in (near-) real-time. Among the others this layer includes:

i) Measurements of energy production from *Distributed Generation*.

ii) Status of *Distribution Grid* that are needed to simulate energy flows and evaluate the integration of Renewable Energy Sources (RES). Thus, information sampled by devices monitoring the energy distribution network.

iii) Information sent by *IoT devices*, such as *Ambient sensors*, multi-vector *Smart Meters* (i.e. electricity, gas, heating and water) and *Actuators*.

Furthermore, the proposed infrastructure foresees the integration with real devices exploiting two different approaches: i) *Hardware in the Loop* (HIL) and ii) *"Smart Metering Infrastructure in-the-Loop"*.

HIL refers to the simulations in which a real device (e.g. relay and PV panel) is connected to a virtual environment instead of real system (e.g. simulated grid).

For example smart meters can be considered as IoT devices, the proposed architecture interconnects them with the real time simulator (as the grid virtual model) based on an IoT approach, which is different from the conventional off-the-shelf solutions used for HIL. Including *"Smart Metering Infrastructure in-the-Loop"*, gives the possibility to integrate in the simulation framework third-party smart metering infrastructure, such as [65, 66]. As further explained in in Section 5.1.2.2, this integration can be achieved thanks to the *Communication adapter*. In this view, each Internet-connected smart meter sends measurements to its smart metering infrastructure that can forward them to the *Real-Time Simulator* exploiting either MQTT or REST adapters.

5.1.2 Cyber layer

The **Cyber layer** is in charge of enabling data exchange among the different components of the SMIRSE infrastructure. It exploits both the synchronous and asynchronous communication paradigms presented in Section 4. SMIRSE adopts both request/response and publish/subscribe [7] approaches, respectively. Request/response allows a fast bidirectional communication to send/access information to/from different components of SMIRSE (either hardware or software), using, for instance, REST Web Services [74]. Whilst, publish/subscribe is complementary to request/response and allows (near-) real-time data transmission. Publish/subscribe removes the interdependencies between producer and consumer of information. This allows developers in creating distributed software components that are independent from data-sources and can react in (near-) real-time to certain events. Thus, publish/subscribe enables a data-driven and event-based communication that also increases the scalability of the system as pointed out in [77]. The proposed solution adopt MQTT protocol [75], which is an implementation of publish/subscribe.

As shown in Figure 5.1, the **Cyber layer** consists of three main modules the *Communication Adapter*, the *Data Integration Platform* and *Smart Metering Infrastructure*. The Communication Adapter,further presented in Section 5.1.2.2 enables the interoperability across the heterogeneous devices in the Physical Layer. Whilst, the *Data Integration Platform* integrates third party software and platforms in the Environmental Layer. Both act as a bridge between the components of infrastructure and the underlying technologies, either hardware or software. In this view, each technology needs a specific *Communication Adapter* or a *Data Integration Platform* to provide common and unified interfaces to access low-level functionalities through REST Web Services and/or MQTT. Thus, both *Communication Adapter* and *Data Integration Platform* are key component to access each low-level technology transparently. Finally, SMIRSE provides features to integrate also third party *Smart Metering Infrastructure*, such as [78], that makes available historical data collected from real distribution networks and post-processed information output of its services.

5.1.2.1 Data integration platform

The Cyber Layer integrates a *Data integration platform* that provides components specifically designed for accessing and managing information, coming from heterogeneous data sources. The data integration plaform exploits a JSON-based RESTful APIs and in this way creates a virtual District Information Model correlating data from different data-sources.

The platform trough a Resource Catalog provides a constantly updated registry of endpoints for the available resources. The Resource Catalog provides flexibility to the whole infrastructure because new ICT systems can be transparently added, removed or replaced in the system. Furthermore, the platform trough the use of an Historical Datastore API, that is implemented for each integrated ICT system, provides historical data to the simulation and modeling exposing API to access it.

In order to provide Semantic interoperability the platform takes advantage of of Semantic Web technologies to annotate and interlink integrated data sources and query them using the semantic attributes. This feature is implemented by the Semantic Metastore service, which builds on the off-the-shelf Semantic Web technologies such as Apache Jena [27] to provide higher-level, developer-friendly APIs for populating and querying the semantic knowledge base through REST Web Services.

The District Information Model (DIM) created by the *Data integration platform* is composed by different entities from the *Environmental* and *Physical layer* shown in Figure 5.1.

In particular, the following specific data sources, presented in Section 5.1.1 are integrated to get a comprehensive view of a district:

- Building Information Models;
- Geographical Information Systems;
- System Information Models (SIMs [12])
- Weather data
- Grid Status

The *Data integration platform* is heavily based on RESTful Web Services and distributed deployment. The use of RESTful Web Services provides a uniform interface to each component of the system. In particular, the same response format is returned, independently from the actual queried data source. This also means that, if a component technology (e.g. a DBMS) changes, the client application receives the same response as a result of the same query.

The platform provides a RESTful Web Service interface to the underlying databases. It interacts with the Data layer by using SQL queries and preparing JSON results.

5.1.2.2 Communication Adapter

In the Cyber layer are integrated two type of *Communication adapters*: one is in charge of connecting and enabling the communication of heterogeneous devices. The second is in charge of enabling the communication between the different modules of the modeling and simulation layer.

Hence, the first *Communication adapter* is related to the *Smart Metering Infrastructure*, that is in charge of retrieving measurements from heterogeneous devices, either wired and wireless, that exploit different communication protocols (e.g., IEEE 802.11, ZigBee, PLC or 6LowPan). The interoperability among these technologies is achieved trough through *Communication adapters* called *Device Integration Adapters* (DIA). Hence, this layer integrates several IoT devices with traditional and industrial communication technologies by abstracting every communication language in a unique data format. They have been developed following a methodology described in [79]. Then measurements are sent to the cloud infrastructure exploiting MQTT. The other *Communication adapter* (see Figure 5.2) is in charge of enabling data exchange among the *Real-time simulator* (RTS) and other infrastructure modules. It implements two communication paradigms presented in Section 4: i) *publish/subscribe* [7] based on MQTT (Message Queuing Telemetry Transport) protocol [75] and ii) *request/response* based on REST (Representational State Transfer) [71]. In particular, the publish/subscribe communication model allows the development of loosely-coupled event-based systems. As explicit dependencies between data-producer and data-consumer are removed, each module in the proposed infrastructure can publish data, which can be independently received by a number of subscribers. This also increase the scalability of the whole infrastructure [65].

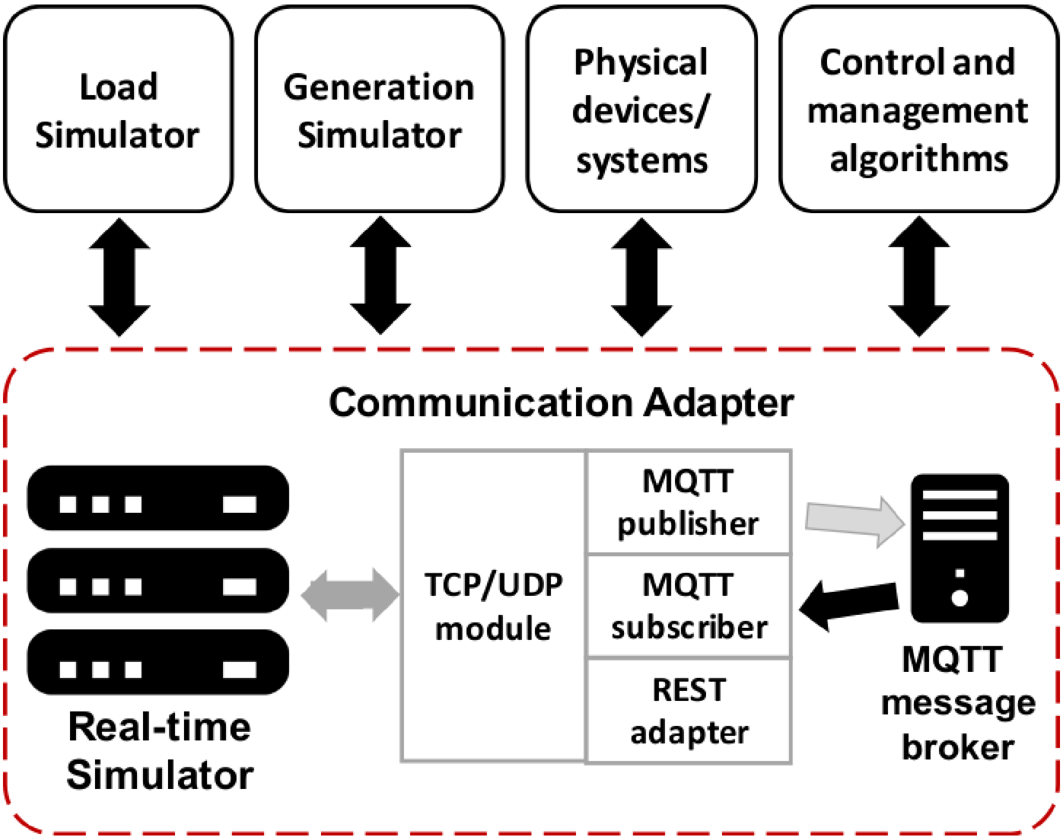


Fig. 5.2 Schema of the communication adapter

In its core, the *Communication adapter* exploits the *TCP/UDP module* to allow a bidirectional communication with the RTS. It exploits either *TCP* or *UDP* protocols to give a full communication compatibility with the real-time simulators in the market (e.g. Opal-RT) that provide software libraries and tools to exploit these two protocols. Thus, depending on the communication requirements needed by the simulation, the Communication adapter can be configured to use either TCP or UDP. Such real-time simulators are in charge of performing hard real-time simulations respecting the predefined time-steps. However, the interaction among different modules of the overall co-simulation for most of the use-cases requires rates of exchanging data in the range of seconds to minutes, which are much higher than the latency of the communication infrastructure. This guarantees that messages are delivered in time. This module implements both server and client functionalities, in that it receives and sends data to the real-time simulation engine. Such data includes either electrical measurement from virtual meters in the grid model (e.g. voltage, current, active and reactive power) or control signals from some control and management algorithms (e.g. voltage or current signals, time-variant load or generation values and switch activation signals).

The Communication adapter provides the *MQTT publisher module* that parses data from RTS into sequence of events before publishing to an *MQTT Message broker*. The Message broker keeps track of all publications and subscriptions and takes care of sending new data to subscribers. Hence, any client application (e.g. Control and managements algorithms) subscribes to the Message broker for receiving an information flow according to its needs. The Communication adapter works also as subscriber. Therefore, it is able to receive commands from remote applications (e.g. Control and managements algorithms and Load Simulator modules) and pushes them to the real-time simulation engine. Finally, the REST adapter is in charge of retrieving information from REST web service, even third party services, and passes the results to RTS, again through the *TCP/UDP module*.

The *Communication adapter* also translates the information from the data-format used by the software components in Figure 5.2 to the data-format suitable for the RTS and vice-versa. These software components exploit JSON that is an open-standard format that uses human-readable text to transmit data objects consisting of key-value pairs. It is the most common data-format used for data exchange in web environments. Whilst, the data-format for RTS must be a vector of numbers. We organized this vector also as a series of key-value pairs. The odd positions in the vector are the numeric keys while the even positions are the associated numeric values. Thus, the *Communication adapter* translates each key-value pair in the JSON into the equivalent numeric key-value pair in the vector and vice-versa.

In this platform, real-time simulation is performed by the digital real time simulator in which the virtual model of the grid is simulated. The overall co-simulation is in near- real-time mode as the data exchange is seldom in real-time due to communication delay of the infrastructure. To ensure synchronization, the real-time simulator is running with a shifted time stamp with respect to the real world wall-clock time.

It is worth noting that the *Communication adapter* allows the integration of the *Real-time simulator* with other smart metering infrastructure (e.g. [65, 66]), where each smart meter is an Internet-of-Things (IoT) device. Hence, data coming from such devices can be used to simulate and test innovative control strategies with (near-) real-time data from the grid. Vice-versa, each simulated grid component is also seen by other modules as an IoT device able to send information and to receive commands.

5.1.2.3 Smart Metering Infrastructure

The SMIRSE platform integrates in its Cyber layer existing *Smart Metering Infrastructure* in order to collect and retrieve measurement coming from heterogeneous IoT devices. In particular the *Flexmeter* [80] infrastructure has been integrated inside the SMIRSE platform. This section will present briefly the *Flexmeter* solution for smart metering.

The *Flexmeter Smart Metering Infrastructure* is build of different software modules that perform: (i) allow bidirectional communication with DIAs, thus with devices; (ii) receive, control and store measurements; (iii) provide REST web services to access data, devices, assets and maintenance operations; (iv) send commands to devices.

The infrastructure exploits a Message Broker in order to enable an asynchronous bidirectional communication with devices, through MQTT [7], to send data in (near-) real-time. The use of this paradigm avoids interdependencies between information producer and consumer. The role of the Message Broker is to direct all events coming from and to the Communication Engine module. The Communication engine manages a bidirectional interaction with DIAs (See Section 5.1.2.2) by storing measurements into the Data Storage and sending commands to devices. The Communication Engine consists of two sub-components: (i) Event Sources and (ii) Command Destinations. The Event Sources are MQTT subscribers that sign up to input topics used by DIAs to publish measurements. Event Sources check the integrity of the incoming message payload and push it into the Inbound Pipeline. Inbound Pipeline is an intelligent buffer that manages network traffic spikes to relieve the database interface from congestion and to ensure the measurements storage. Flexmeter also allows receiving command requests to target devices. Command requests are managed by the Command Destinations. They are MQTT publishers that route command requests to the right devices. Outbound Pipeline, designed as the Inbound Pipeline, prevent spikes in command requests and send the payload content to Command Destinations and then to the target device. The Data Storage module permits to manage the connection with different time-series databases and non-relational databases which are specifically developed for Big Data management. The

Assets Manager is a software module that manages different information regarding people, places and things that are called assets in the Flexmeter platform. Finally, the Device Manager handles the communication be- tween the Asset Manager, the Communication Engine and applications that interact with the Flexmeter platform through the REST API Interface Manager. The REST API Interface Manager defines and provides REST web services that are designed to permit the access to information regarding devices, assets and measurements, and manages these entities in the infrastructure. An authentication is required to request REST web services. Hence, only allowed applications and services can operate with the Flexmeter platform. Through these software components, the abstraction from device communication, data and information storage management and from different low-level hardware functionalities is achieved.

5.1.3 Modelling and Simulation Layer

The **Modelling and Simulation Layer**, in Figure 5.1, consists of different software components to simulate environmental conditions (green boxes), electrical energy (light blue boxes) and thermal energy (yellow boxes).

The *Solar Radiation Decomposition* is a software module that decomposes Global Horizontal radiation (GHI) into Direct Normal Incident radiation (DNI) and Diffuse Horizontal Incident radiation (DHI) by applying mathematical models such as [81]. The inputs are meteorological information retrieved by *Weather Data* module in the Environmental Layer. Often, weather stations sample only GHI. Thus, this module is crucial because DNI and DHI are needed to evaluate the solar heating gains and to simulate incident solar radiation on tilted surfaces (e.g. buildings' rooftops).

Device Integration Adapters

The *Rooftop Solar Radiation* integrates three modules of the infrastructure presented in [82]. It exploits GIS cartographies, real *Weather Data* and results from *Solar Radiation Decomposition* module to simulate incident solar radiation on rooftops. Simulations are done in real-sky conditions with a resolution of 15 minutes. Furthermore, can be used to identify suitable surface for solar energy installations on rooftops. It is able to detect roof encumbrance (e.g. chimneys and dormers) and to estimate their shadowing effects. This module is then used by the *Photovoltaic (PV) Modeling Services*.

The *Photovoltaic Photovoltaic (PV) Modeling Services* module exploits the methodology described in [82–84]. It exploits both *Rooftop Solar Radiation* and *Weather Data* modules to estimate the incident solar radiation and the effects of the air temperature on the efficiency of PV arrays. By exploiting GIS cartographies, it also identifies the best configuration (Foolrplanning) for PV deployment on rooftops and simulates the energy production with a resolution of 15 minutes. Furthermore, the module integrates algorithm for accurate modelling of tension current (I-V) curves of the PV system.

Further details on the implementation of this three services are presented in Section 5.2.1.

The *Real-time Grid Simulator* module integrates a Real-Time Simulators (e.g. Opal-RT or RTDS) as depicted in [85]. It simulates power distribution networks with different time resolutions ranging from microseconds to hours. Exploiting the *Communication adapters* in the *Cyber layer*, it is able: i) to access information from IoT devices deployed across the real distribution network in (near-) real-time and ii) to exchange data with the *Photovoltaic Energy*. The *Real-time Grid Simulator* module enables a more accurate analysis of distribution network when different control strategies are applied. In future, this module can be used to simulate congestions and unbalances, and to evaluate the electrical implication of Demand Response events on distribution networks. Further details on the implementation of this module are presented in Section 5.2.2

The *Indoor Temperature Simulator* follows the methodology described in [22]. It provides tools to simulate and analyse the thermal behaviour of buildings. It combines information about *BIM* and *GIS* together with real *Weather data* and environmental information coming from *IoT Devices* deployed in the corresponding real buildings. This module allows: i) (near-) real-time visualization of energy consumptions in buildings; ii) simulation of indoor temperature trends and iii) evaluations of building performances through energy models. Further details on the implementation of this module are presented in Section 5.2.4

The *Power Prediction and thermal building characterization* provides tools to analyze and predict the power demand of thermal systems in buildings connected to HDN. Furthermore, exploiting the KPIs and the methodology described in [86] the module provides thermal characterization of the buildings. Further details on the implementation of this module are presented in Section 5.2.3.

5.2 Multi Energy System Simulations

5.2.1 Solar energy simulation

This Section presents a distributed and modular software infrastructure that expose REST Web Services [72] to perform solar energy simulations (see Figure 5.3 and Figure 5.4). Figure 5.3 shows the involved modules and data source of the SMIRSE platform for enabling solar energy simulation. Its modularity takes advantage of the microservices approach described in Section 4.1.

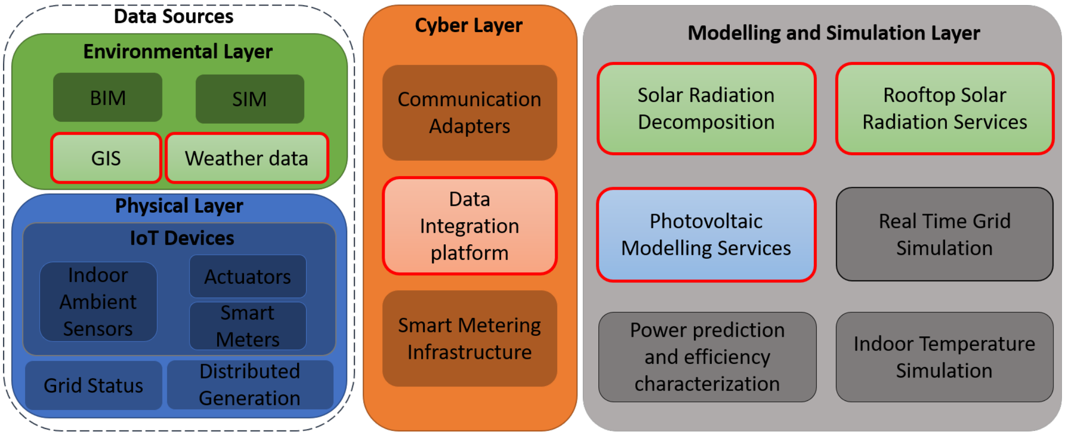

Data Sources
Environmental Layer
BIM
SIM
GIS
Weather data
Physical Layer
IoT Devices
Indoor Ambient Sensors
Actuators
Smart Meters
Grid Status
Distributed Generation
Cyber Layer
Communication Adapters
Data Integration platform
Smart Metering Infrastructure
Modelling and Simulation Layer
Solar Radiation Decomposition
Rooftop Solar Radiation Services
Photovoltaic Modelling Services
Real Time Grid Simulation
Power prediction and efficiency characterization
Indoor Temperature Simulation

Fig. 5.3 Schema of SMIRSE Platform with focus on Solar Energy

The solution needs as main inputs the Digital Surface Models (DSMs) and the Cadastral maps. DSM is a raster image representing terrain elevation with buildings. A DSM with a high-resolution (in the order of sub-meters) permits i) to better recognize encumbrance in rooftops, such as chimneys and dormers, that would not allow deployment of PV panels and ii) to have a better simulation of shadows that will affect the PV energy production. Thus, higher is the DSM resolution, higher is the accuracy of the energy production estimation. Cadastral map is a vector image reporting the square footage of buildings in the area of interest. It also exploits meteorological data coming from third-party services to perform sub-hourly real-sky simulations of solar radiation and PV systems production. The main outputs are information on the size of deployable PV system(s) and the related generation profiles for each building that can be used as input to third-party solutions (e.g. [87–90]).

In addition, the solution exploits standard services defined by the Open Geospatial Consortium (OGC) [73], presented in Section 4.3. WPS are used for upload-

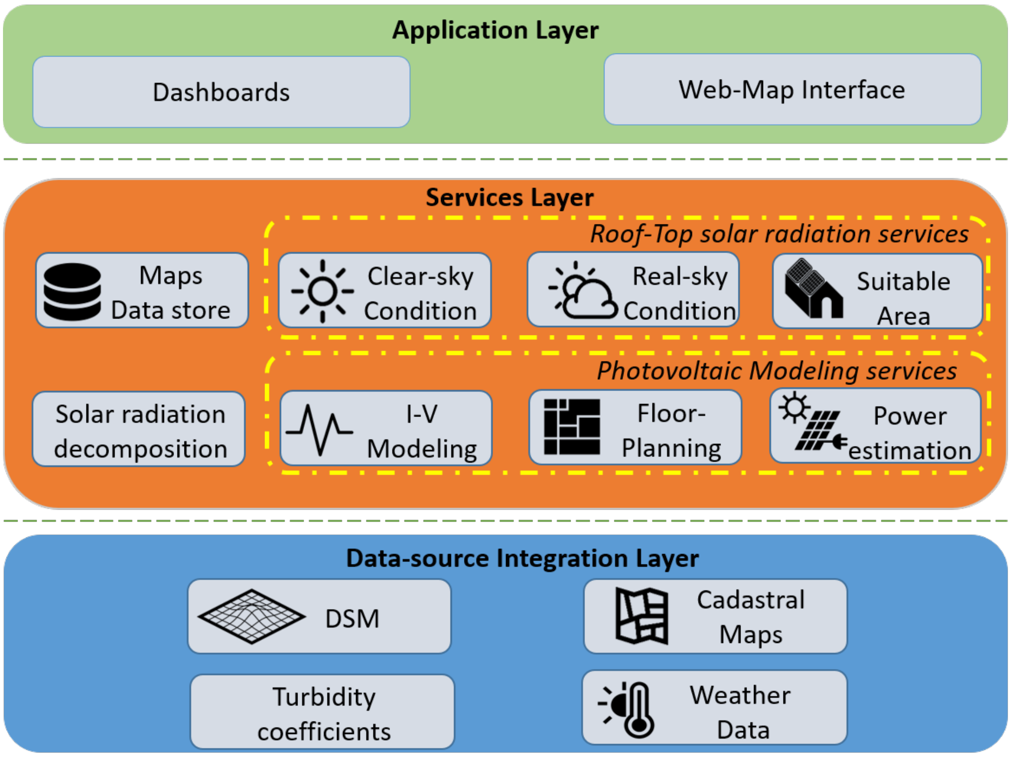


Fig. 5.4 Schema for the proposed software infrastructure to estimate PV energy production

ing both the DSM and the Cadastral map and for executing simulation processes. Indeed, they define rules for standardizing inputs and outputs of a process. For developing WPS services the *PyWPS* implementation of OGC standard has been used [91]. WFS are used for querying and retrieving features about the elements of a polygon-map. WMS helps in visualizing the produced maps through the *Web-Map interface*. As shown in Figure 5.4, the solution is a three-layered distributed infrastructure consisting of: *i) Data-source Integration Layer*, *ii) Services Layer* and *iii) Application Layer*. The rest of this section describes each layer in more detail.

5.2.1.1 Data-source Integration Layer

The *Data-source Integration Layer* (the lowest layer in Figure 5.4) is in charge of integrating in the infrastructure the following heterogeneous information: *i) Digital Surface Model* (DSM); *ii) Cadastral map*; *iii) Linke Turbidity coefficients* [92] and *iv) Weather data*. The *Linke Turbidity coefficients* express the attenuation of solar radiation related to air pollution. This information can be automatically retrieved by third-party services, such as [58, 93], or can be specified by end-users before executing the simulation. Finally, *Weather data* are also retrieved by third party services, such as [76], and collected by personal weather stations deployed in cities. In particular, the needed inputs are solar radiation, ambient temperature and wind speed.

5.2.1.2 Services Layer for real-sky simulations

The *Services Layer* (the middle layer in Figure 5.4) is in charge of *i)* simulating solar radiation in real-sky condition; *ii)* identifying rooftops areas suitable for deploying PV systems and *iii)* evaluating their energy production. It consists of different software modules. It is worth highlighting that each of them exposes REST Web Services. Hence, each module can be invoked by third-party software to retrieve information and simulation results.

5.2.1.2.1 Solar radiation decomposition service Nowadays in our cities, weather stations are pervasively deployed and their information are also provided by third-party services, such as [76]. Normally such stations are equipped with global horizontal solar radiation sensors and do not provide information on direct normal and diffuse horizontal radiation. However, to simulate real-sky solar radiation on a pitched surface, information on direct and diffuse radiation is needed [49]. To overcome this limitation, the *Solar radiation decomposition services* integrates six different solar radiation decomposition techniques in literature [94–96, 11, 97, 98] to compute both direct normal and diffuse horizontal radiation starting from measurements of global horizontal radiation.

These models can be categorized by the number of predictors. In the following we briefly introduce them. Erbs et al. [94], Reindl et al. [97] and Karatasou et al. [11] use only the clearness index k_t as predictor. k_t is the ratio between global radiation and extraterrestrial radiation both on a horizontal plane. Skartveit et al. [98] use k_t, solar altitude and solar zenith angles as predictors. Ruiz-Arias et al. [96] use k_t and the air mass as predictors. In their work, Engerer et al. [95] provide three different models. Our solution implements the second model with the following predictors: k_t, the zenith angle, the time of the day, the clear-sky global radiation on horizontal plane and a variability index, which represents the deviation of the observed k_t value from the clear sky value of the clearness index. The end-user can specify the

Table 5.1 Monthly average days identified by [10]

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Date	17^{th}	16^{th}	16^{th}	15^{th}	15^{th}	11^{st}	17^{th}	16^{th}	15^{th}	15^{th}	14^{th}	10^{th}
DOY	17	47	75	105	135	162	198	228	258	288	318	344

model suitable for the area of interest. Indeed, as reported by [99–101], the accuracy of decomposition models is strongly affected by different latitude, longitude and environmental conditions. Hence, the integration of these decomposition models and third-party meteorological services makes our infrastructure flexible in performing simulation in different geographic locations. The input of this module is a time series of the global horizontal radiation provided by third-party meteorological services in *Data-source Integration Layer*. The output of this process is a JSON with the values of Direct Normal Incident radiation (DNI) and Diffuse Horizontal Incident radiation (DHI) for the requested time interval.

5.2.1.2.2 Rooftop solar radiation services The following three services: *i) Clear-sky condition Service, ii) Suitable surface identification service* and *iii) Real-sky calculation service* have been integrated as unique service in the SMIRSE as reported in Section 5.1.3 and in Figure 5.3.

5.2.1.2.3 Clear-sky condition Service In order to compute clear-sky solar radiation, the infrastructure, exploit *GRASS-GIS* open-source software, which embeds in its core *r.sun* [49]. The *r.sun* tool provides an accurate simulation of solar radiation in urban contexts [102, 25, 48]. The resulting outputs of this module are set of direct and diffuse solar radiation maps in clear-sky condition with 15 minutes time interval. Such maps are stored in the *Maps Data-store* ready to be used by the *Real-sky calculation service*.

It performs such computation considering the monthly average days identified by [10] and reported in Table 5.1 with the related Day of the Year (DOY). DOY is a 1 to 365 non-dimensional sequential index starting from January 1^{st}. For instance: January 17^{th} is day 17, February 16^{th} is day 47, December 31^{st} is day 365.

The module requires as inputs: *i)* the DSM, *ii)* monthly *Linke* turbidity coefficients, *iii)* Slope and Aspect maps, that are produced with *GRASS-GIS r.slope.aspect* tool. Monthly *Linke* turbidity coefficients are retrieved by using third party web services such as [93, 58] or can be specified by end-users as parameters in the execution request. The Slope and Aspect maps represent respectively the inclination and the orientation (expressed in degrees) of each pixel of the DSM. After their calculation, they are stored in the *Maps Data-store*.

5.2.1.2.4 Suitable surface identification service Thanks to the high-resolution DSM given as input to the overall simulation process, this module identifies available surface for deploying solar systems on pitched rooftops excluding, for instance, dormers and chimneys. The *Suitable surface identification service* uses as inputs both Slope and Aspect maps for identifying the suitable area. These maps are retrieved from the *Maps Data-store*. By default, it identifies areas representing tilted rooftops oriented between South-Est and South-west. Commonly, a tilted rooftop has a slope (θ) in-between $10° \leq \ \theta \ \leq 45°$. This range is also suitable to install a PV system. Furthermore, facades with an orientation (γ) in-between $220° \leq \ \gamma \ \leq 320°$ (considering South facing roofs having $\gamma = 270°$) are more exposed to solar radiation. For these reasons, we chose these ranges for θ and γ as the default values for the *Suitable surface identification service*. However, the end-user can give as input new ranges for θ and γ to select the desired suitable surfaces.

The output of this selection is a binary map where pixels with 1 as value represent the available areas. Such map is then filtered with *GRASS-GIS r.neighbors* tool to smooth noise and to remove small areas that are too small for installing a PV system (e.g. areas where deployable PV system are smaller than 1 kW). This resulting map is vectorized and clipped with the cadastral map stored in *Maps Data-store*. The information on the size of area of the resulting polygons is a 2D projection of the real roof surface. To estimate the real surface of polygons, the Formula (5.1) is applied to correct the value of the area with inclination angle of rooftops.

$$S = \frac{S_{2D}}{cos(\theta)} \tag{5.1}$$

where S_{2D} is the 2D area of the polygon and θ is the roof inclination obtained from the slope map, again retrieved from the *Maps Data-store*.

The output of this module is a GeoJSON reporting a number of polygons that represent deployable areas.

5.2.1.2.5 Real-sky calculation service This module produces maps of incident global radiation on pitched rooftops in real-sky conditions. The inputs of this service are values of DNI, DHI, direct and diffuse solar radiation maps retrieved from the *Maps Data-store*. Through the *Data-source Integration layer*, the *Real-sky calculation service* retrieves information on solar radiation from third-party services. Both values of DNI and DHI radiation are required to calculate solar radiation on tilted surface [49]. If meteorological services provide only global horizontal radiation and not its direct and diffuse components, the *Real-sky calculation service* invokes the *Solar radiation decomposition services* to compute such information. Thus, *Real-sky calculation service* uses the values of DNI and DHI for each time interval to calculate the *clear-sky indexes* k_c^b (5.2) and k_c^d (5.3).

$$k_c^b = \frac{DNI_{overcast}}{DNI_{clear-sky}} \tag{5.2}$$

$$k_c^d = \frac{DHI_{overcast}}{DHI_{clear-sky}} \tag{5.3}$$

k_c^b (5.2) represents the ratio between DNI in overcast conditions and DNI in clear-sky conditions. While, k_c^d (5.3) is the ratio between DHI in overcast conditions and DHI in clear-sky conditions. Finally for each time interval, diffuse and direct radiation maps, produced by the *Clear-sky condition Service*, are multiplied by *clear-sky indexes* k_c^b and k_c^d. Then, both maps are summed together to obtain a global incident radiation map in real-sky condition. The resulting output of this process is a set of GeoTIFF images representing the maps with incident global solar radiation in real-sky condition.

5.2.1.2.6 Photovoltaic Modeling service The *Photovoltaic Modeling services*, of the SMIRSE platform presented in Section 5.1.3, offers the possibility of exploiting two different methods for estimating energy production: *i)* Calculation of power production exploiting or NOCT formula [103] or Mattei et al. [104] formula and *ii)* an accurate modeling of I-V curves of the PV modules presented in [84]. Furthermore, the *Photovoltaic Modeling services* integrates and algorithm for foolrplanning [83] to find optimal configuration of PV systems.

$$T_{sol-air} = T_a + kG_t = T_a + \frac{\alpha_{roof}}{h_c} G_t \tag{5.4}$$

$$T_c = \frac{T_{sol-air} + (T_{c,NOCT} - T_{a,NOTC})(\frac{G_t}{G_{t,NOCT}})(1 - \frac{\eta_{mp,STC}(1-\alpha_p T_{c,STC})}{\tau\alpha})}{1 + (T_{c,NOCT} - T_{a,NOTC})(\frac{G_t}{G_{t,NOCT}})(\frac{\alpha_p \eta_{mp,STC}}{\tau\alpha})} \tag{5.5}$$

$$T_c = \frac{U_{PV}(v) T_{sol-air} + G_t[\tau\alpha - \eta_{mp,STC}(1 - \alpha_p T_{c,STC})]}{U_{PV}(v) + \alpha_p \eta_{mp,STC} G_t} \tag{5.6}$$

$$U_{PV}(v) = 26.6 + 2.3v \tag{5.7}$$

$$P_{out} = \eta_{mp,STC}(1 + \alpha_p(T_c - T_{c,STC}))G_t A \tag{5.8}$$

T_a	ambient temperature	α_p	temperature coefficient of maximal power of the solar cells [%/°C]	$T_{c,STC}$	operating cell temperature at standard condition (usually $T_a = 25°C$, $G_t = 800\ W/m^2$)
α_{roof} [%]	convective factor of the roof	$\eta_{mp,STC}$	maximum power point efficiency under standard test conditions (%)	$G_{t,NOCT}$	solar radiation on a tilted surface at NOCT condition ($1000 W/m^2$)
h_c $[W/m^2K]$	radiative loss factor of the roof	τ	solar transmittance of any cover over the PV array (%)	U_{PV}	heat exchange coefficient for the total surface
G_t	solar radiation on a tilted surface	α	solar absorptance of the PV array (%)	A	available surface $[m^2]$
T_c	operating cell temperature	v	wind speed $[m/s]$		
$T_{c,NOCT}$	nominal operating cell temperature	P_{out}	power output $[W]$		

Power production with NOCT or Mattei et al. Formula This module is in charge of estimating PV production for each area identified by *Suitable surface identification service*. The required inputs are *i)* the maps provided by the *Real-sky calculation service* and *ii)* the GeoJSON given by the *Suitable surface identification service* where polygons representing deployable areas are reported. The *Photovoltaic energy estimation service* also uses weather data coming from the *Data source integration layer* in order to estimate the operating cell temperature (T_c), thus the efficiency of the PV system. This module estimates from ambient temperature T_a the sol-air temperature $T_{sol-air}$ which is defined as the ambient Temperature T_a plus the loss factor k, as reported in Formula (5.4).

The use of sol-air temperature to obtain more reliable results in estimating the operating cell temperature is reported by [50, 52]. This module can use two models for estimating the cell temperature. The first [103], denoted as *NOCT*, can be used if wind speed is not provided by the nearest weather station. It is expressed by the Formula (5.5).

The second model [104], denoted as *Mattei*, uses wind speed in order to estimate the operating cell temperature, as reported in the Formula (5.6). *Mattei* model is one of the most reliable with in-situ wind data, as reported by [105].

In Formula (5.5), [10] assumes $\tau\alpha = 0.9$. While in Formula (5.6), [104] assumes $\tau\alpha = 0.81$. Finally, the instant power P_{out} is computed by Formula 5.8.

In its core, the *Photovoltaic estimation service* uses the characteristics of some commercial PV modules as default values (i.e. α_p, $\eta_{mp,STC}$, $T_{c,NOCT}$, $T_{c,STC}$, $G_{t,NOCT}$, α_{roof} and h_c). However before performing the simulation, the end-user can change this parameters depending on the characteristics of the interested PV system. The output of this module is a GeoJSON that provides for each building information on the size of deployable PV system and the related generation profiles for the requested time interval.

Modelling I-V curve This module derives a power model *for an individual PV module*, so that total power extracted by a panel can be adapted to different series/parallel topology. Total power is in fact generally different from the simple sum of the power values of the individual modules, since it is rather voltage (in series) and current (in parallel) that need to be summed up. Therefore, rather than a model of the *power* of a module, models for the extracted *current and voltage of a panel* are needed.

Moreover, the model should be also sensitive to the granularity of the maximum power point tracking (MPPT). Generally speaking two options are available (Figure 5.5): *module-level* MPPT (micro-inverters) or *string-level* MPPT (string inverters). The model depends on this feature because in the former case each module extracts at the MPP (*a*), and therefore only individual maximum power voltages and currents can be tracked. Conversely, when using string inverters (*b*), the MPP is tracked on the resulting I-V curve of the series of modules, which has to be computed from the individual curves.

One last architectural parameter that needs to be considered is the distribution of the bypass diodes in the installation. This is essential because it determines how to compute the aggregate I-V curves of the series string. The most intuitive strategy in which bypass diodes are used around each module [106], as shown in Figure 5.5, is assumed.

Panel Model Using a String-Level Inverter For the description of the methodology, a PV-MF165EB3 module by Mitsubishi as a working example has been used, as this is also the module used for our validation in Section 6.2.6. The methodology has however general validity, since it relies on basic information available in any datasheet.

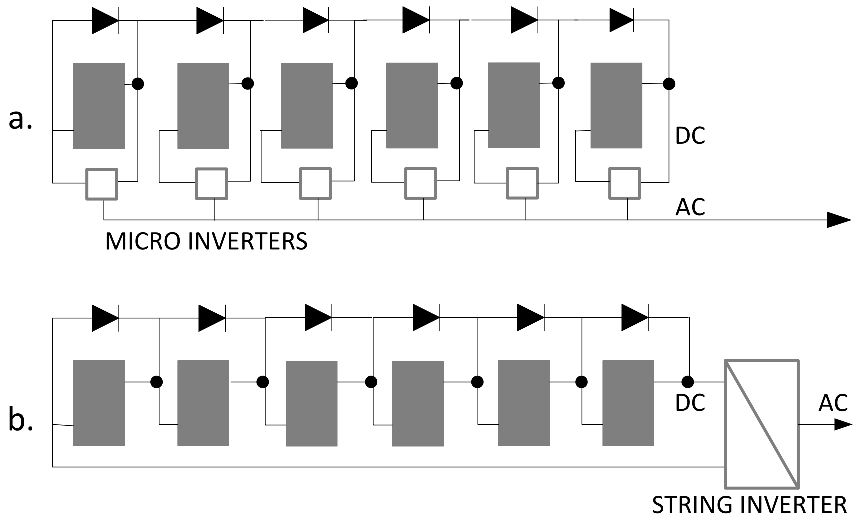


Fig. 5.5 String-level and module-level MPPT architectures.

For the example PV-MF165EB3 module, the datasheet provides the three curves shown in Figure 5.9: from left to right, (1) the I-V curve for different irradiances G, (2) the temperature sensitivity coefficients ($\partial V_{oc}/\partial T$ and $\partial I_{sc}/\partial T$), and (3) the dependence of V_{oc} and I_{sc} of irradiance G. These are very basic information available for almost any PV module.

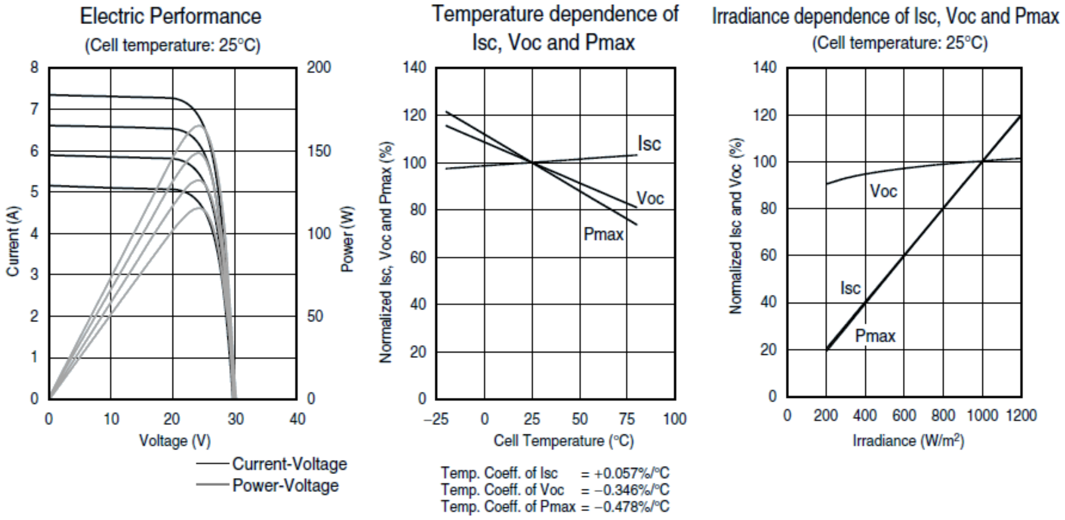


Fig. 5.6 Datasheet information for Mitsubishi's PV-MF165EB3.

The derivation of the model proceeds in two phases, as described hereafter.

Model for an Individual PV Module A model for the output voltage V_{module} and current I_{module} of a single module, as a function of irradiance G and temperature T has been derived.

This phase consists of three main steps:

1. Derivation of the dependence of V_{oc}, I_{sc} on G and T.

Firstly the dependence of V_{oc} and I_{sc} on T and G is derived, by using the center and right plot of Figure 5.9. This simply achieved by digitizing the curves and empirically fitting them using minimization of least-square errors. For the specific case:

$$I_{SC}(G,T) = \alpha \cdot I_{SC,nom}(-0.00055T + 0.9885) \cdot (0.000992G - 0.000344) \tag{5.9}$$

$$V_{OC}(G,T) = V_{OC,nom}(-0.00338T + 1.088) \cdot (-3.069G^{-0.02289} + 3.62) \tag{5.10}$$

In Equation 5.9, the nominal value of I_{sc}, derived from the datasheet, is weighted by an aging factor α. PV panels are indeed subject to an average 0.4%/year degradation rate, that mainly affects current production, while voltage distribution does not change substantially over panel lifetime [107].

It is worth emphasizing that Equations 5.9 and 5.10 do an approximation in considering the effects of G and T as two independent factors.

2. Derivation of the dependence of T on G.

An important aspect to be considered is that T and G are obviously correlated: when irradiance is high, temperature will also be high. Therefore the ambient temperature T_{amb} (available from weather stations) is corrected with a term depending on G, according to the model of [103].

The module temperature $T(G)$ is modeled as:

$$T_{amb} + k \cdot G \tag{5.11}$$

where $k = \frac{\alpha}{h_c} = 0.05 \frac{W}{K \cdot m^2}$ is the ratio of the absorptance of the roof divided by the radiative loss factor of the roof [103]. In Equations 5.9 and 5.10, T will thus be replaced by $T(G)$ (as in Equation 5.11). Notice that relating T and G allows smoothing the approximation contained in Equations 5.9 and 5.10.

3. Derivation of the module I-V curves.

The last step is to derive a function describing the I-V curve for different G's. An equation template that matches the underlying diode equation regulating a PV cell behavior is used:

$$I = I_{SC} - a \cdot (e^{b \cdot V} - 1)$$

The approximation consists in assuming a zero series resistance in the dependence. By imposing then that $I(V_{oc}) = 0$, parameter b can be expressed in terms of a as:

$$b = \frac{1}{V_{OC}} \cdot ln(1 + \frac{I_{SC}}{a})$$

leaving therefore a as the only parameter. Then empirically fit the curves in Matlab and obtain a value of $a = +4.428 \cdot 10^{-5}$. The overall model for the I-V curve is therefore given by the equation:

$$\begin{cases} I(G,T) = I_{SC}(G,T) + 4.428 \cdot 10^{-5} \cdot (e^{b \cdot V} - 1) \\ b = \frac{1}{V_{OC}(G,T)} \cdot ln(|1 + 22,583.56 \cdot I_{SC}(G,T)|) \end{cases} \tag{5.12}$$

Figure 5.7 shows the comparison w.r.t. the datasheet curves using the model of Equation 5.12. The curves are relative to 25^{o}C. The average error of the interpolation is 2.79%.

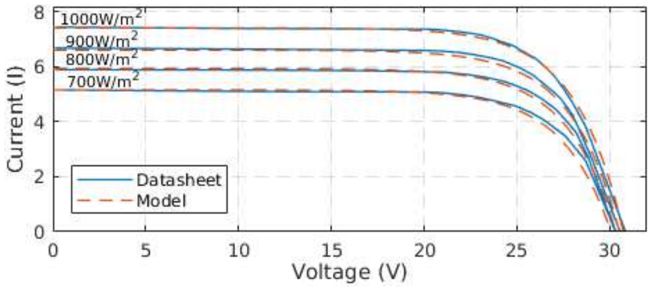


Fig. 5.7 Comparison of the proposed equation-based model (dashed lines) w.r.t. datasheet specifications (solid lines).

Building the Panel Model An expression of the extracted power of a generic series-parallel interconnection of modules must consider (1) the overall series-parallel topology, and (2) the operations of the bypass diodes.

For the sake of simplicity, the procedure by considering a string of two modules will be described. The generalization to the general case of n series modules is straightforward.

Due to the presence of bypass diodes, the weakest of the two modules (i.e., the one with lowest irradiance) does not constrain the current of the strongest one; rather, when the current gets larger than the value that can be produced by the weakest module, the latter gets bypassed and only the strongest module produces power [106]. The total curve is therefore obtained by summing the I-V curves as follows: for each current value in the range $[0, I_{SC,H}]$ the resulting voltage is:

$$\begin{cases} V_{string} = V_L + V_H & \text{if } I < I_{SC,L} \\ V_{string} = V_H - V_d & \text{if } I_{SC,L} < I < I_{SC,H} \end{cases} \tag{5.13}$$

where subscript L and H apply to the low and high irradiance modules, respectively, and $V_d = 0.6V$ is the voltage drop across a forward-biased diode. This yields to the classical I-V curve with multiple "steps", as shown in Figure 5.8.a.

Once this curve has been built, the MPP is extracted as the maximum of the corresponding P-V curve, thus emulating the operation of a string inverter implementing the MPPT.

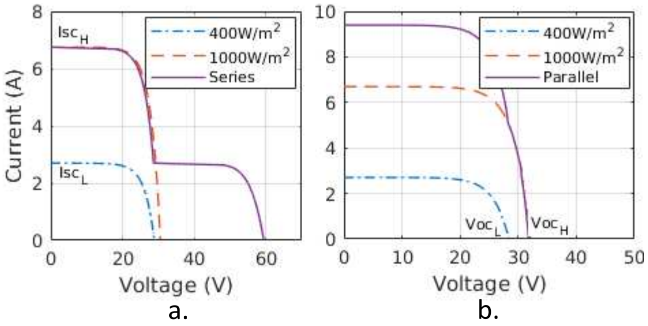


Fig. 5.8 I-V curves of the series (a.) and parallel (.b) connection of two modules with different irradiance with bypass diodes.

Given the I-V curves (and the corresponding MPPs) for the various series strings, are combined by summing the currents. The process in this case is simpler that the series case because there is no diode involved.

Again, the simple case of two strings in parallel for the sake of illustration (Figure 5.8.b) is considered. The resulting parallel curve is obtained, by summing the current of the two modules for each voltage value in the range $[0, V_{OC,H}]$ as follows:

$$\begin{cases} I_{panel} = I_L + I_H & \text{if } V < V_{OC,L} \\ I_{panel} = I_H & \text{if } V_{OC,L} < V < V_{OC,H} \end{cases} \tag{5.14}$$

5.2.1.2.7 Photovoltaic System Floor-planning The objective is to place N PV modules on a given area (not necessarily rectangular). The latter is aligned to a virtual grid whose elements are squares of side s. The sides of the area are integer multiples of s so that it consists of an integer number N_g of grid elements.

The value of s is chosen so that the panel sizes are also an integer multiple of s. Assuming that each panel is identical and has sizes $w \times h$, it have that $w = k_1 \cdot s, h = k_2 \cdot s$. The size of the specific panel used in the analysis is 160×80cm, and it use $s = 20$cm. Notice that since grid points represent possible placement candidates, a smaller s yields more solutions, at the expense of longer computation times.

Are then given a set of measures over time of irradiance $G_i(t)$ and temperature $T_i(t)$, $i = 1, ..., N_g$. The spatial resolution of the irradiance measures determined by the solar data acquisition tool is forced to coincide with the grid granularity, so each grid point has a specific value of G and T. The process of deriving solar data with the desired granularity is provided exploiting the *Real-sky calculation service*.

The problem solved in with this module can be formulated as follows:
Given an available surface, a set of N_T irradiance and temperature measures for each point of a grid, and N PV modules to be placed according to a specific series-parallel topology, find the optimal placement of the panels on the surface that maximizes the energy extracted in the interval $[0, N_T]$.

PV Panel Power Model The analysis needs a power model of an individual single PV module, because the total power extracted by the panel P_{panel} depends on its actual series/parallel topology and is in general different from the sum of the power of the individual modules.

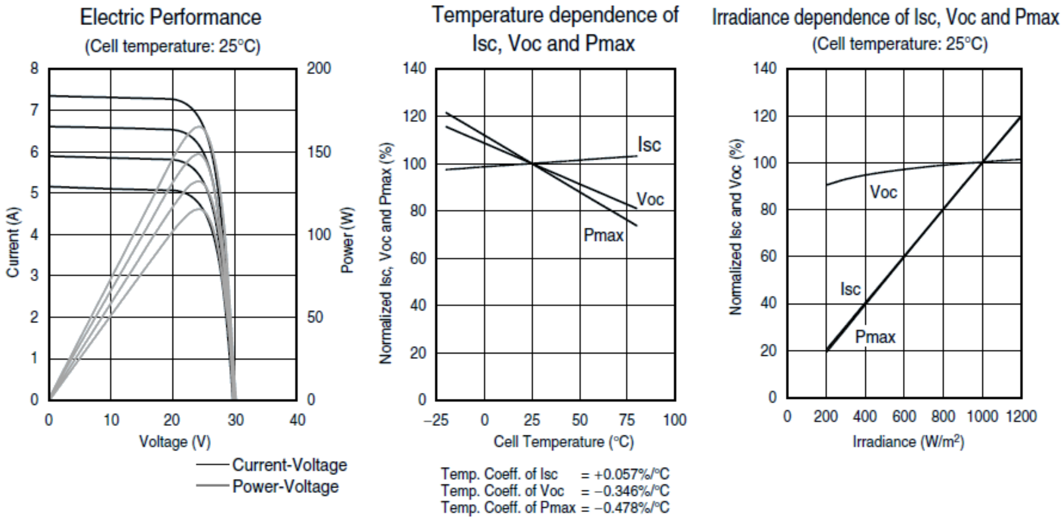


Fig. 5.9 Power characteristics of Mitsubishi's PV-MF165EB3.

Given a $m \times n$ series-parallel interconnection (i.e., n parallel strings each of m modules in series), the total power is obtained as $P_{panel} = V_{panel} \cdot I_{panel}$, where:

$$\begin{cases} V_{panel} & = & \min_{j=1,\dots,n}(\sum_{i=1,\dots,m} V_{module,ij}) \\ I_{panel} & = & \sum_{j=1,\dots,n}(\min_{i=1,\dots,m} I_{module,ij}) \end{cases}$$

and $V_{module,ij}$ and $I_{module,ij}$ are the voltage and current extracted from the i-th module in the j-th string.

The setup, considers a PV-MF165EB3 module by Mitsubishi, for which an empirical model of V_{module} and I_{module} as a function of irradiance G and temperature T from information available in the datasheet (Figure 5.9) is derived. It is assumed that each module extracts the maximum power, i.e., V_{module} and I_{module} are the maximum power voltage and current, and $P_{module} \equiv P_{max}$.

The model derivation is done as follows:

1. Using the rightmost plot of Figure 5.9 the equations expressing the dependence of V_{oc}, I_{sc} and P_{max} are derived with respect to irradiance G. These plots are normalized with respect to reference values (at 25^oC temperature and $G = 1000W/cm^2$) of $V_{oc,ref} = 30.4V, I_{sc,ref} = 7.36A$, and $P_{max,ref} = 165W$, as reported in the datasheet.

2. Using the middle plot, we replace $V_{oc,ref}, I_{sc,ref}$, and $P_{max,ref}$ with functions the express their dependence on temperature, yielding equations for V_{oc}, I_{sc}, and P_{max} that include the dependence on G and T.

3. The above analysis does not consider the important fact that T and G are obviously correlated: when irradiance is high, temperature will also be high. Therefore ambient temperature T is corrected with a term depending on G, according to the model of [103]. The actual module temperature T_{act} is modeled as $T + k \cdot G$, where $k = \frac{\alpha}{h_c}$ is the ratio of the absorptivity of the roof divided by a convective and radiative ($15 \frac{WK}{m^2}$) [50].

4. The last step is to derive V_{module} and I_{module} from V_{oc} and I_{sc}. To this purpose, the fact that (leftmost plot of Figure 5.9), the maximum power voltage of the module is roughly independent of the irradiance and is $\approx 80\%$ (24V) of V_{oc} is exploited. This allows to express V_{module} as a function of G and T. Since the relation betwen I_{module} and I_{sc} is more complex to extract, simply derive it as the ratio of P_{max} and V_{module}.

These processes result into the following equations:

$$
\begin{aligned}
&T_{act} = T + k \cdot G \\
&P_{module}(G,T) = 165 \cdot (1.12 - 0.048 T_{act}) \cdot 10^{-3} G \\
&V_{module}(G,T) = 24 \cdot (1.08 - 0.34 T_{act}) \cdot (0.875 + 0.000125 G) \\
&I_{module}(G,T) = P_{module}(G,T) / V_{module}(G,T)
\end{aligned}
$$

Wiring Overhead Characterization The use of a loose placement of the PV modules incurs in an obvious wiring overhead, causing power loss and cost increase.

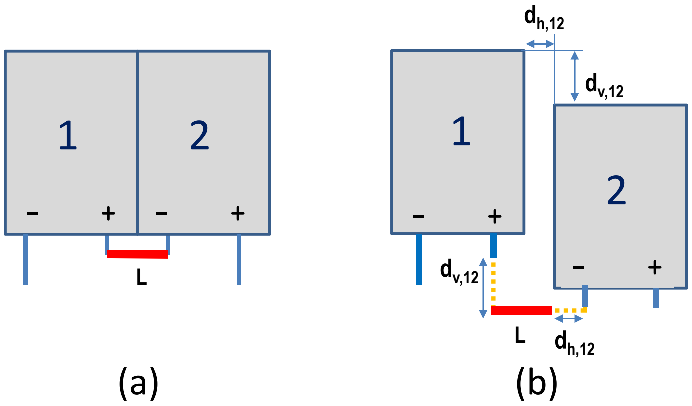


Fig. 5.10 Wiring overhead characterization.

Figure 5.10 exemplifies the problem for the series connection of two modules (1 and 2). Even in a compact placement (Figure 5.10-(a)), some wiring is needed (the thick red connection) between the corresponding terminals. The modules are are distanced vertically by $d_v^{1,2}$ and horizontally by $d_h^{1,2}$ (Figure 5.10-(b)), the extra wiring is simply given by the sum of the two displacements (the dotted orange lines), since we have to subtract the length L of the default connector. Notice that in a real scenario we actually have shorter connections, since we are not forced to route wires along the (x,y) directions.

For the generic serial interconnection of N_s modules, the overhead will simply by given by: $L_{ovh} = \sum_{i=1}^{N_s-1}(d_v^{i,i+1} + d_h^{i,i+1})$. Knowing the current and the unit resistance of the wire used for the connection, we can compute the power drop incurred by the extra cable (RI^2).

For the parallel connection of the strings, conversely, the overhead can be neglected. In fact, typical PV installations with two or more strings do not wire the serial strings “manually”, but do it through a *combiner box* [108], which would be used anyway even for the traditional floorplanning.

Floorplanning Algorithm The calculation of the optimal placement requires an exhaustive enumeration of all possible candidate grid points, which becomes quickly unfeasible even for small areas. The solution space has a worst case size of $O(N_g^N)$; assuming to place $N = 20$ modules on a 100 m^2 surface, a grid of 20 cm implies $N_g = 2500$ candidates points, yielding a space of $O(10^{67})$ solutions. Notice that it is not possible to introduce bounds on the enumeration because the total extracted power can be computed only when all the modules are placed, because it is not possible to sum the power of the individual modules.

For this reason, we devised a simple and efficient greedy approximation to the solution based on a ranking of a “suitability” metric for all the grid points. Given this metric, the algorithm simply allocates modules greedily, by selecting candidate points in decreasing order of suitability.

The suitability metric should distill the temporal traces into a compact signature that synthesizes the distribution of G and T values. The obvious choice of using the average is not a good choice because the typical distributions of irradiance and temperature are strongly skewed towards smaller values, and the average is not a representative value.

As a more aggregate indicator, the k-th *percentile of the distribution* as a compact metric is used, which represents the value below which $k\%$ of the samples fall. Specifically, the 75-th percentile is chosen. Larger values of the percentile identify distributions that are more skewed towards the upper range of the values; therefore, the suitability metric should combine the percentiles of G (favoring larger values since larger G values are beneficial) and T (favoring smaller values, since smaller T values are beneficial).

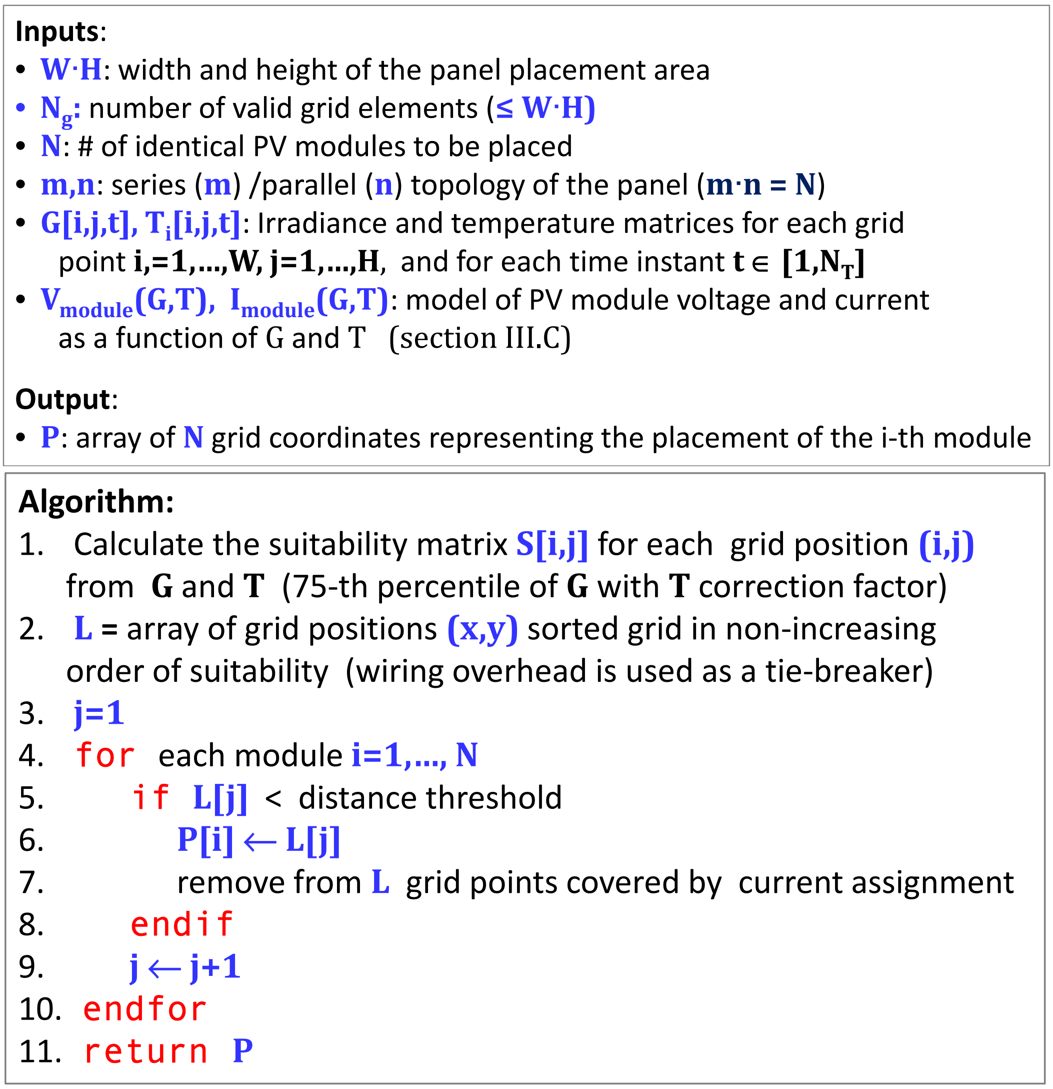
However, mixing two percentiles should be done carefully and with the appropriate weight. From Figure 5.9, is shown that the G affects the output power way more than temperature: over a range of $[200-1000]W/cm^2$, the power changes by 5x, whereas typical T ranges only change power by $\pm 20\%$ at most. Therefore, the suitability metric uses only the 75-percentile of G, and temperature is used as a corrective factor $f(T)$ that tracks the dP_{max}/dT of the middle plot of Figure 5.9. The suitability s_{ij} in each grid point (i,j) is thus obtained as $s_{ij} = p_{75}^{G_{ij}} \cdot f(T)$, where $p_{75}^{G_{ij}}$ is the 75-th percentile of G in the position (i,j).

Figure 5.11 shows a pseudo-code of the algorithm.

As in any greedy algorithm, the steps are relatively straightforward. First (Line 1) the suitability matrix **S** is computed as described above. In Line 2, an array **L** of grid coordinates is calculated, sorted in decreasing order of their value of **S**. In case of identical values of suitability, the distance from the already placed modules is used as a tie-breaker (closer grid points have higher rank). Then iterates (Line 4) over the N modules, in series-first order, i.e., modules belonging to a series string are enumerated before moving to another string.

In Line 5 introduces an important filter on the solutions. Although the wiring overhead is already counted in the sorting, it is just a tie-breaker. It might occasionally be the case that a given $L[j]$ has a high suitability but is quite far apart from the already placed modules. To this purpose only coordinates $L[j]$ that do not exceed a given *distance threshold* are considered; the latter is empirically determined as twice the average distance of the already placed modules.

The i-th module is then placed in the j-th position according to the previously computed ranking (Line 6) . An important point is that, since a module occupies more $k_1 \cdot k_2$ grid points (see Section III.A), all these “covered” points are clearly unusable and must therefore be removed from **L** (Line 7).



Inputs:

- $W{\cdot}H$: width and height of the panel placement area
- N_g: number of valid grid elements ($\leq W{\cdot}H$)
- N: # of identical PV modules to be placed
- m,n: series (m) /parallel (n) topology of the panel ($m{\cdot}n = N$)
- $G[i,j,t]$, $T_i[i,j,t]$: Irradiance and temperature matrices for each grid point $i,=1,\ldots,W$, $j=1,\ldots,H$, and for each time instant $t \in [1,N_T]$
- $V_{module}(G,T)$, $I_{module}(G,T)$: model of PV module voltage and current as a function of G and T (section III.C)

Output:

- P: array of N grid coordinates representing the placement of the i-th module

Algorithm:

```
1.  Calculate the suitability matrix $S[i,j]$ for each grid position $(i,j)$
    from G and T (75-th percentile of G with T correction factor)
2.  $L$ = array of grid positions $(x,y)$ sorted grid in non-increasing
    order of suitability (wiring overhead is used as a tie-breaker)
3.  $j=1$
4.  for each module $i=1,\ldots,N$
5.      if $L[j]$ < distance threshold
6.          $P[i] \leftarrow L[j]$
7.          remove from $L$ grid points covered by current assignment
8.      endif
9.      $j \leftarrow j+1$
10. endfor
11. return $P$
```

Fig. 5.11 Algorithms inputs and pseudo-code.

Then (Line 9) the next coordinate from **L** after the removal of covered points piked. The loop terminates when the *N* panels have been placed.

5.2.1.3 Application Layer

The *Application Layer* represents the highest layer of the proposed infrastructure (see Figure 5.4). It is dedicated to end-user applications, such as *Web-Map displayers* and *Dashboards*, that can provide information about performed simulation across the city with different level of details. In addition at this layer, third-party solutions can retrieve simulation results and estimations on PV energy production to perform further analysis and evaluations.

5.2.2 Real-Time grid simulation

The Real-Time Simulator (RTS) is in charge of reproducing the behaviour of a real electric distribution system, which is modelled as a test bed for testing and validation of new technologies, management algorithms or control strategies, through performing *software in-the-loop* (SIL) or *hardware in-the-loop* (HIL). Figure 5.12 shows the involved modules of the SMIRSE infrastructure in Real-time grid simulation. Real-time simulation can be used for fast simulations, closed-loop testing of measurement, protection and control equipment, and generally all *what-if* analyses.

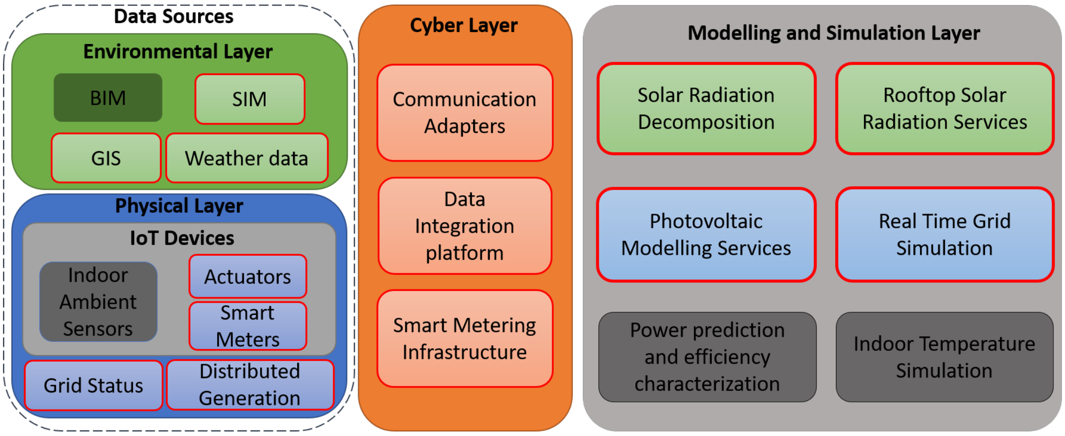

Data Sources
Environmental Layer
BIM
SIM
GIS
Weather data
Physical Layer
IoT Devices
Indoor Ambient Sensors
Actuators
Smart Meters
Grid Status
Distributed Generation
Cyber Layer
Communication Adapters
Data Integration platform
Smart Metering Infrastructure
Modelling and Simulation Layer
Solar Radiation Decomposition
Rooftop Solar Radiation Services
Photovoltaic Modelling Services
Real Time Grid Simulation
Power prediction and efficiency characterization
Indoor Temperature Simulation

Fig. 5.12 Schema of SMIRSE Platform with focus on Real-time Grid Simulation

The main added value of real-time simulation is the possibility of replacing physical devices or systems with virtual models, which eventually reduces costs and enables more complete and continuous testing of the entire system, safely and without interruptions. In SIL, an algorithm or a control strategy can be tested on a real-time model of the system. In this way, validation is carried out safely, and under a variety of different scenarios which may not be all feasible or available in real world (e.g. faulty conditions). In HIL similar tests can be performed, connecting physical devices directly to the simulated environment.

In the proposed architecture, the real-time simulation module is mainly in charge of reproducing the distribution grid behavior from physical infrastructure perspective. In other words, physical components of the grid including generators, transformers, lines, breakers, loads, etc. are modeled in RTS, while the models which generate time-variant inputs for loads, generators, or controllable devices are made and hosted outside. The grid model can be implemented either in RSCAD, in case of using RTDS real-time simulator, or in MATLAB/Simulink, in case of using OPAL-RT real-time simulator with eMEGAsim configuration for EMT analysis.

Real-time simulation is actually simulating a system, which realistically responds to the external stimuli, as the inputs/outputs of the simulation are synchronous with the real world. Section 5.1.2.2 presents the *Communication Adapter*, which is our software solution to enable a bidirectional communication between the real-time simulator and other services and/or devices.

For instance, the requested measurements from the real-time model can be sent out through the *Communication Adapter* to be used by the control and management algorithm development nodes (SIL). In addition, the real-time digital simulator can communicate with real physical devices and system components (e.g. a micro-grid controller, protection automation systems, PV panels and wind turbines) through analog and digital I/O signals (HIL).

In the proposed distributed infrastructure, the RTS models the distribution system, receives inputs from external modules for single components modelling and provides the required measurements or signals to external applications. It actually provides simulation as a service.

In this implementation, an Opal RT OP5600 real-time simulator with 12 cores operating at 3.46 GHz and eMEGAsim configuration aimed for electromagnetic transient (EMT) analysis is used. Electricity grid is modelled in MatLab Simulink using SimPowerSystem (SPS) under SimScape library. An additional library called ARTEMiS provided by Opal-RT is also used to provide fixed-step solver dedicated to complex power systems. The model is compiled using RT-LAB software and executed on the simulator. As introduced before, any application algorithm or control/management strategy is kept outside of the real-time model and interfaced through communication channels.

As the external modules are not necessarily synchronized with the real-time model and they run independently of the real-time system, either at their own time-step or launched by external triggers, Asynchronous Process Application is used. In this case, a low speed serial data acquisition (e.g. with some milliseconds or seconds

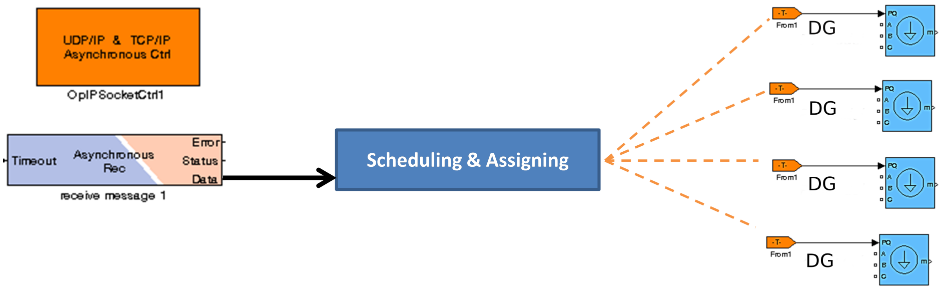


Fig. 5.13 Real time simulator receiving data links to the Distributed Generation over the grid

time step) can be performed in a separate process from the real-time model, where a high speed data exchange is required (e.g. 250 μs time-step).

The asynchronous process requires a C or C++ source code implemented by the user to handle some sort of communication blocks inside the real-time model. The code consists of some predefined functions which are compiled once during the initialization process. The data signals to be sent or received and the communication protocols (i.e. TCP or UDP) to use for communicating with the *Communication Adapter* should be also selected inside the real-time model prior to model execution.

Considering the integrated simulation tool as a realized Software In-the-Loop (SIL), the environment which is the smart grid in this case, is executed on the real time simulator. Grid real time simulation module is responsible to emulate real grid behaviour facing different load or generation profile values, and provides the status of the electrical system in terms of power flows, voltage profile, etc. The signals coming from outside of the real time simulator are controlling or defining modelled prosumers behaviour by updating generation output. During the simulation the grid model requests the necessary values (active and reactive powers) to update modelled generation output, and receives the required data from the PV simulator through appropriate UDP blocks inside the real time model (Fig. 5.13). The grid model can be run for an electromagnetic transient simulation with 50 μs to 250 μs (or phasor simulation with a few milliseconds) time steps, while the new values of generation can be updated every 15 min.

The model is built with the SimPowerSystem (SPS) toolbox of MatLab/Simulink®. The ARTEMiS software from OPAL-RT is used to provide fixed-step solver dedicated to complex power systems. It is an add-in toolbox to SPS enabling hard real-time simulation of power systems. The main modelled components of the distribution grid are a three-phase voltage source in series with an RL branch as an equivalent model for the upstream high voltage (HV) grid connected to a slack bus, three-phase two-winding transformers, three-phase π section lines to model medium voltage (MV) lines and three-phase three-wire dynamic load models with external control of active and reactive powers to model the prosumers.

5.2.3 Power prediction and building efficiency characterization

The power prediction and building efficiency characterizations (PPBEC) platform is a distributed engine providing a variety of software services for energy management systems. Figure 5.14 presents the involved modules of the SMIRSE infrastructure involved in PPBEC simulation.

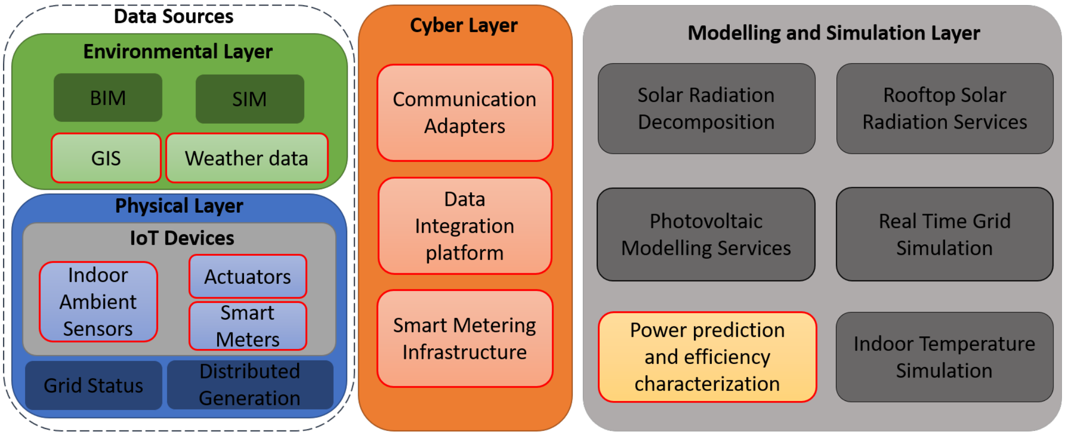

Data Sources
Environmental Layer
BIM
SIM
GIS
Weather data
Physical Layer
IoT Devices
Indoor Ambient Sensors
Actuators
Smart Meters
Grid Status
Distributed Generation
Cyber Layer
Communication Adapters
Data Integration platform
Smart Metering Infrastructure
Modelling and Simulation Layer
Solar Radiation Decomposition
Rooftop Solar Radiation Services
Photovoltaic Modelling Services
Real Time Grid Simulation
Power prediction and efficiency characterization
Indoor Temperature Simulation

Fig. 5.14 Schema of SMIRSE Platform with focus on power prediction and building efficiency characterizations

Figure 5.15 shows the overall architecture of the PPBEC system. Since PPBEC has been designed for the collection, integration, modelling, storage, and analysis of energy-related data, it consists of a four-layered architecture with: (i) a *Data-source Layer*; (ii) a *Middleware Layer*; (iii) a *Storage and Data Analysis Layer* and (iv) a *Application Layer*, briefly described below and detailed in the following sections.

The *Data-source Layer* includes all the source data, such as smart meters and web services, that continuously provide data of interest to the PPBEC system. These heterogeneous data sources provides fine grain data (e.g., roughly every 5 minutes). The *Middleware Layer* enables the interoperability across these heterogeneous data

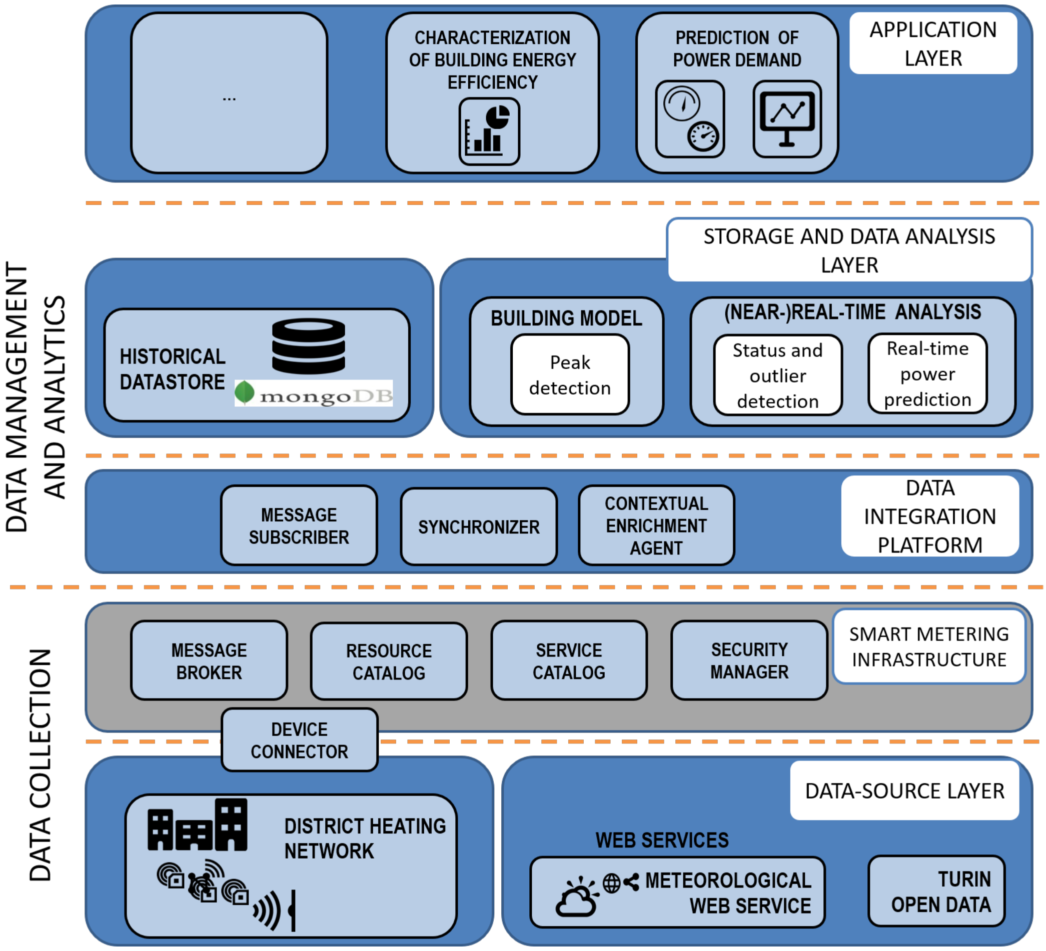

...
CHARACTERIZATION OF BUILDING ENERGY EFFICIENCY
PREDICTION OF POWER DEMAND
APPLICATION LAYER
STORAGE AND DATA ANALYSIS LAYER
HISTORICAL DATASTORE
mongoDB
BUILDING MODEL
Peak detection
(NEAR-)REAL-TIME ANALYSIS
Status and outlier detection
Real-time power prediction
DATA MANAGEMENT AND ANALYTICS
MESSAGE SUBSCRIBER
SYNCHRONIZER
CONTEXTUAL ENRICHMENT AGENT
DATA INTEGRATION PLATFORM
MESSAGE BROKER
RESOURCE CATALOG
SERVICE CATALOG
SECURITY MANAGER
SMART METERING INFRASTRUCTURE
DEVICE CONNECTOR
DATA COLLECTION
DISTRICT HEATING NETWORK
DATA-SOURCE LAYER
WEB SERVICES
METEOROLOGICAL WEB SERVICE
TURIN OPEN DATA

Fig. 5.15 Schema of the PPBEC Platform

sources, by creating a peer-to-peer network in which the communication between peers is trusted and encrypted. Collected data are managed by the *Storage and Analytics Layer* which aims at integrating data and storing them into a non-relational database to effectively support different complex analytics services. A variety of algorithms have been designed, developed, and integrated in PPBEC to support the data analysis task. At the end, the *Application Layer* exploits the knowledge items discovered through the data analysis process to provides useful feedbacks to the different interested users and to suggest ready-to-implement energy efficient strategies.

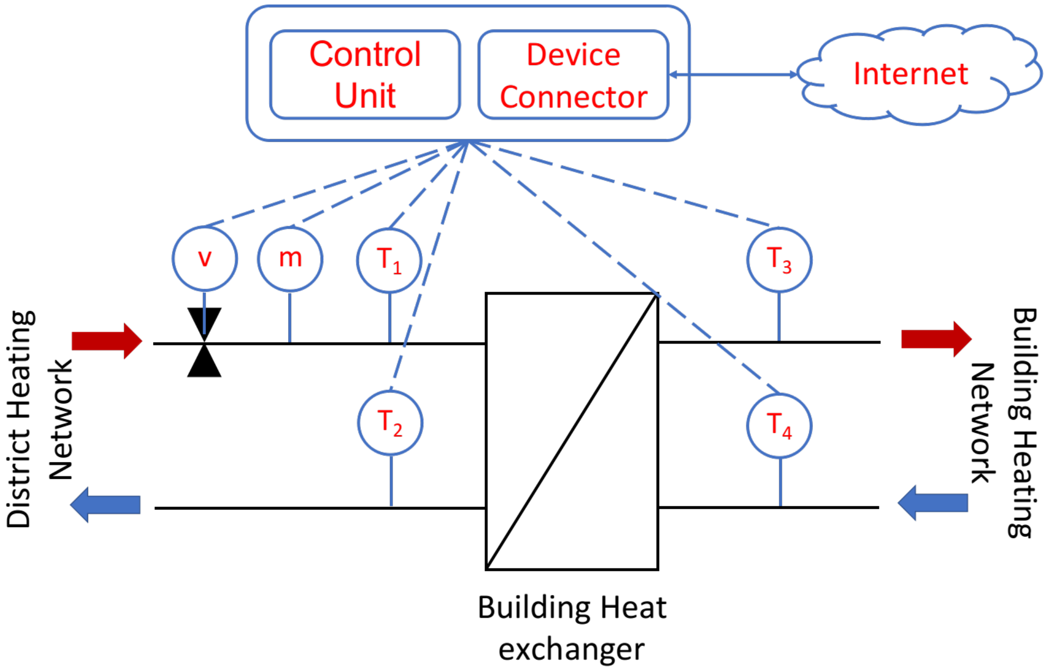


Fig. 5.16 Schema of the Cyber-Physical-System for the Building Heat-Exchanger.

5.2.3.1 The cyber physical system

A Heating Distribution Network (HDN) usually covers a city or a portion of it. The heating energy provider produces the hot water to heat up the buildings with heating power plants in the city. The hot water is sent to the buildings through pipelines of the HDN deployed across the city. Then, each building is equipped with a building heat-exchanger (see Figure 5.16), which connects the Building Heating Network (BHN) to HDN. In particular, the heat-exchanger transfers heat, that can be expressed in power, from HDN to BHN. To monitor the performance and control the heat-exchanger several devices are installed and connected to the Internet. This upgrades the heat-exchanger, a Physical-System, into a Cyber-Physical-System that can be remotely monitored and controlled by exploiting the IoT paradigm. As show in Figure 5.16, the heat-exchanger is equipped with five sensors: *i)* m to provide the water mass flow rate incoming from HDN; *ii)* T_1 to measure the temperature of the incoming hot water flow from HDN to the heat-exchanger; *iii)* T_2 to provide the temperature of the cold water flow that returns to HDN from heat-exchanger; *iv)* T_3 to measure the temperature of the hot water flow from the heat-exchanger to the BHN; and *v)* T_4 to provide the temperature of the cold water flow that returns to the heat-exchanger from BHN. These data are collected by a *Control Unit*, deployed as well, and sent to the proposed platform through the *Device Connector* (see Section 5.2.3.2.1). The *Device Connector* is a software component that enhances the *Control Unit* to provide the heat-exchanger (now a Cyber-Physical-System) with Internet connection. It also integrates the CPS with the rest of the proposed platform described in Section 5.2.3.2 and Section 5.2.3.3. Furthermore, the *Control Unit* exploits m, T_1 and T_2 to compute the power exchanged by the heat-exchanger. Finally, the *Control Unit* manages the valve v increasing or decreasing the mass flow rate, thus the exchanged power. This is needed to remotely control and adjust the power request of the building fulfilling the strategies of the heating energy provider (e.g. changing the schedule of the operating phases to avoid large peak request in the overall network).

5.2.3.2 Data Collection

In a Smart City scenario, one of the main issues concerns the coexistence of several heterogeneous technologies. Moreover, future Smart City systems will deal with Internet-of-Things [109] (IoT) environments, where multiple actors need to access transparently IoT resources and services. Hence, the lack of interoperability among heterogeneous technologies must be addressed. To cope with these issues, a distributed software infrastructure that exploits a middleware approach to integrate different IoT devices and technologies is developed. The aim is providing many services for collecting and managing energy-related data. PPBEC adopts the open-source LinkSmart middleware [110] and extends its features to fulfill the requirements for a Smart City context. Indeed, an IoT middleware for Smart City needs (i) to be highly available, (ii) to scale up rapidly, and (iii) to provide a uniform interface to all deployed technologies.

5.2.3.2.1 Data-source Layer The *Data-source Layer* is the lower layer in PPBEC (see Figure 5.15). It can include different kinds of hardware and software entities that continuously provide various data types of interest to PPBEC. *Hardware entities* correspond to *IoT devices* measuring physical quantities. Instead, *software entities* are *software services* exposing to external clients physical measures collected from third-parties. They allow acquiring data values complementary to those collected through hardware entities, that contribute in the overall characterization of the context under analysis. Web services are an example of software entities that expose interfaces over the Internet allowing clients to send requests and get data using HTTP as transport protocol.

Nevertheless any data source can be integrated in the Data-source Layer, this study is focused on a layer composed as follows. (i) A network of *smart meters* as IoT devices (hardware entities) located in buildings within a HDN to measure building thermal energy values. A single smart meter is located in each building. (ii) *Web services* (software entities) to monitor surrounding conditions when measurements of thermal energy take place. Different web services can be considered to enrich measurements collected through the smart meters. In the solution are selected those exposing meteorological data due to the well-assessed strong correlation between climate conditions and thermal energy consumption.

Each data source in the layer provides the following two types of data: (a) *dynamic data* as monitored measures usually collected roughly every few minutes and potentially exhibiting highly variable values; and (b) *static data* as features describing some time invariant properties of the data source (as the location of the monitoring nodes).

In the PPBEC instance considered in this study, *dynamic data* include measures on *building thermal energy* and *climate conditions* collected roughly every 5 minutes, even if different and variable resolutions can be considered. Thus, a large volume of energy-related data is continuously acquired for each building. Smart meters installed in buildings provide fine-grained data related to building thermal energy (as instantaneous power, cumulative energy consumption, water flow and corresponding temperatures). Meteorological web services (e.g., Weather Underground [111] considered in this study) provide different kinds of meteorological data as temperature, relative humidity, precipitation, wind direction, UV index, solar radiation and atmospheric pressure. Such data are collected through several weather sensors deployed throughout the city.

For each *monitoring node* (building smart meter or weather sensor), *static data* report features characterizing the data source as its *geographical location* (longitude and latitude). Static data also include information characterizing buildings as the *volume* of each building where smart meters are located. This value is used to normalize energy and power values to allow comparison between buildings in terms of consumption per volume unit. When a new data source registers for inclusion in the Data-Source layer, all related static data are acquired, and then stored in the PPBEC data repository.

Measurements collected from the hardware and software entities are also enriched in PPBEC with additional *spatio-temporal information* useful to describe the spatial and temporal distribution of the acquired values (e.g, the spatio-temporal distribution of thermal energy consumption). To this aim, the Data-source layer also includes additional *contextual data sources* such as web services exposing topological data (e.g., Municipality open data portals [112]) or calendar data. More specifically, the geo-coordinates (longitude and latitude) of each monitoring node are mapped to the corresponding neighborhood and city district including that neighborhood. While the geo-referenced location of nodes is given in the hardware/software entities, both the neighborhood and district names corresponding to the geo-referenced location have been added as additional contextual features. They have been retrieved from contextual data sources. Moreover, each measurement time is associated with different blends of time spans as daily time slot (e.g., morning, afternoon, evening, or night), week day, holiday or working day, month, 2-months, or 6-months time periods.

To effectively support the interoperability across heterogeneous IoT devices possibly included in the data-source layer, PPBEC exploits the concept of *Device Connector* (see Figure 5.15). It is a middleware-based component that abstracts a given technology and translates its functionalities into Web Services. The Device Connector enables the communication among heterogeneous devices by allowing developers in exploiting each low-level technology transparently. Thus, it works as a bridge between the Middleware Layer and the underlying technologies or devices in the Data-Source layer.

5.2.3.2.2 Middleware Layer The *Middleware Layer* (see Figure 5.15) is in charge of providing features to discover available resources and services in the Data-source Layer. It creates a network among different entities that can exchange information exploiting two communication paradigms: (i) request/response based on REST [71] and (ii) publish/subscribe [7] based on MQTT protocol [75]. Such features are key characteristics of a software infrastructure dealing with IoT devices. The Middleware Layer includes four software components described below.

The *Message Broker* allows the communication among different entities (both hardware and software) through the publish/subscribe paradigm. This approach supports the development of loosely-coupled event-based systems. Indeed, it removes explicit dependencies between interacting entities (i.e., producer and consumer of the information), thus each entity in the middleware network can publish data and other subscribers can receive it independently. This increases the scalability of the whole system [113]. PPBEC adopts the MQTT communication protocol [75] and delivers data to subscribers as soon as they are measured and published (the delay is negligible).

The *Resource Catalog* registers and provides a list of IoT devices and resources available into the middleware network. It expose JSON-based REST API to automatically access and manage such information. For instance, Device Connectors register their devices and resources, while other middleware-based entities discover such devices and their access protocols.

The *Service Catalog* provides information about available services in the middleware network exposing a JSON-based REST API. It is used by middleware-based entities to discover available services in the network. For instance, it provides the end-points of services such as Resource Catalog and Message Broker.

The *Security Manager* provides features to enable a secure communication among entities in the middleware network. Indeed, it is in charge of authenticating and granting accesses to applications and other middleware-based components. Hence, malicious actors cannot call services in the middleware network and cannot receive any kind of data.

5.2.3.3 Data Management and Analytics

This section presents the *Data Storage and Analysis Layer* of PPBEC that provides different services to address data management and storage as well as analytic tasks.

5.2.3.3.1 Data integration Layer Various data sources can be included in the Data-source layer to allow gathering a rich and heterogeneous data collection, used for feeding the subsequent data analysis phase. To do so, each source can either proactively send its data to the message broker using MQTT communications, that are automatically forwarded to the *Message Subscriber*; or expose REST API to allow the *Contextual Enrichment Agent* collect the required information. For each data source, monitoring nodes may be deployed in different city areas and they may adopt a different timeline in sampling values. Thus, a proper strategy should be devised for the spatio-temporal integration of the acquired measurements.

Currently, in PPBEC power measures collected through the smart meters are enriched with weather data as additional information acquired from external third-party services. The *Synchronizer* module manages the alignment of weather data have with the timestamps of power data, before storing them together in the PPBEC data repository. Specifically, each *power measure* collected for a building is associated with a set of *weather measures* (e.g., temperature, humidity, and pressure) that describe the climate condition when the power measure was collected. Each weather measure (e.g, temperature) is calculated as the weighted mean value of the corresponding measures acquired from N weather stations located near the building. A weight is associated with each weather station based on its proximity to the building. It expresses the relevance of the measure provided by the weather station. The weight is higher for stations closer to the building since they provide a more accurate value on the climate condition at the building proximity. For each weather station, only the closest weather measures in time are considered.

5.2.3.3.2 Data Storage Layer Due to the different kinds of collected data and to easily manage more data types in the future, PPBEC exploits a document-oriented distributed data repository providing rich queries, full indexing, data replication, horizontal scalability and a flexible aggregation framework. Integrated and enriched data are formatted as JSON documents and stored in a NoSQL repository (i.e., MongoDB [114]), which is used as *Historical Datastore*. This collection of historical data is then exploited to create models of the energy consumption for the buildings and for the near real-time data analysis, including building power prediction.

According to the objectives of the data analysis tasks described in Section 5.2.3.3.3, the adoption of the data processing framework Apache Spark [115] upon MongoDB data repository has been evaluated as optimal choice . MongoDB stores data across different nodes (called shards), thus supporting parallel processing by Spark. This distributed architecture provides higher levels of redundancy and availability, which are fundamental when operating in (near-)real-time, and to scale and satisfy the demand of a higher number of read and write operations. Since both Spark and MongoDB adopt a document-oriented data model, they exchange data in a seamless way by making use of the JSON serialization format. This way, Spark jobs are executed directly against the Resilient Distributed Datasets (RDD) created automatically from the MongoDB data repository, without any intermediate data transformation process. Moreover, due to the real time nature of the data analysis, input data sets vary rapidly in time. To improve the performance of the several queries to be executed, MongoDB rich indexing functionalities is exploited in Spark, like secondary indexes and geospatial indexes, that allow to efficiently filter data according to the geospatial coordinates of buildings and nearby weather stations.

5.2.3.3.3 Data Analysis Layer In this study, the PPBEC engine is used to predict the fine grained power level values during the heating cycle of buildings. The data prediction process is structured into three main blocks: (i) data stream processing to support (near-)real-time data analysis, (ii) prediction analysis, and (iii) prediction validation. The main functionalities of the three blocks are briefly presented below and detailed in Sections 5.2.3.3.5,5.2.3.3.6,5.2.3.3.7. **Data stream processing.** Since thermal energy consumption is monitored roughly every 5 minutes in the HDN, a large volume of energy-related data is continuously collected for each building. To efficiently and effectively analyze such large data collection, the PPBEC engine performs the power level prediction task through the data stream analysis over a *sliding time window*, separately for each building. Every time a new measure of power level is collected, one single time window, sliding forward over the data stream of energy-related data, is considered for the prediction task. This window content contains the recent past *energy-related data* for the building heating system, corresponding to thermal *power levels*, along with data about *weather conditions* when those power measures were collected. Consequently, it allows predicting the upcoming value for the building in the near future.

Prediction analysis. This block entails to predict the average future power levels for each building. A prediction model is built for each building separately by considering the energy-related data in the current sliding time window. The building model is then exploited for forecasting the average power level at a given time instant in the near future.

In a HDN, the heating cycle of a building includes two main operational phases: the *OFF-line* phase, when the power exchange is turned off, and the *ON-line* phase, when the power exchange is on. The ON-line phase is then further structured in the alternation of two sub-phases, named the *transient state* and the *steady state*. More in detail, a large exchange of power between building and HDN (*transient state*) interleaves a quasi-constant power exchange between building and HDN (*steady state*).

To deal with this mixed trend and achieve an accurate predicted value, a prediction model composed of three contributions applied in cascade is devised. First, the proposed approach allows the automatic identification of the operational phases described above. Then, it allows forecasting the power level locally at each phase. More specifically, (i) first the *Status and Outlier Detection* (*SOD*) algorithm automatically identifies the operational phases of the heating cycle of a building (Section 5.2.3.3.5). Given a power measurement in a time instant, the SOD algorithm labels the current operation phase as OFF-phase or ON-phase, and this latter case is further categorized as transient or steady state. (ii) Then, the *Peak Detection* (PD) algorithm predicts the peak power value in the transient state (Section 5.2.3.3.6), while (iii) the *Power Prediction* (PP) algorithm predicts the average power profile in the transient state and in the steady state (Section 5.2.3.3.7).

Prediction validation. This block measures the ability of the PPBEC engine to correctly predict the energy consumption values achievable by a building in an upcoming time instant. To this aim PPBEC integrates two metrics named *Mean absolute percentage error* (*MAPE*) and *Symmetric mean absolute percentage error* (*SMAPE*) (See Section 6.1 for further details). Every time a real power level value is received, their values are updated to include the prediction error for the new measure.

5.2.3.3.4 Data flow to support the prediction task This section describes the data flow for power levels prediction by exploiting data coming from both IoT devices and third-party systems such as web services. IoT devices (i.e., smart meters) are deployed in buildings to monitor the status of the heat-exchangers. As shown in Figure 5.17, they exploit the MQTT protocol [75] to publish energy-related measures as messages with an associated topic. A message includes the power value measured on the heating system of a building, the identifier of the same building and the timestamp of the measurement. Messages are asynchronously collected by the *Message Broker*, using the publish/subscribe mechanism, and distributed to all interested subscribers. Therefore, subscriber nodes are responsible for gathering data notifications about new power measures published by IoT devices to the *Message Broker*.

In the scenario, among the subscribers there are the (near-)real-time algorithms, i.e., *Power Prediction* (PP) and *Status and Outlier Detection* (SOD). Each algorithm independently receives energy-related data, sent by IoT devices, from the Message Broker and retrieves meteorological information from third-party web services through REST interfaces [71]. Furthermore, the algorithms gather data from the *Building Model*, included results from the *Peak Detection* (PD) algorithm, which works with already collected historical data. Finally, the results of (near-)real-time algorithms are stored into the MongoDB *Historical Datastore*.

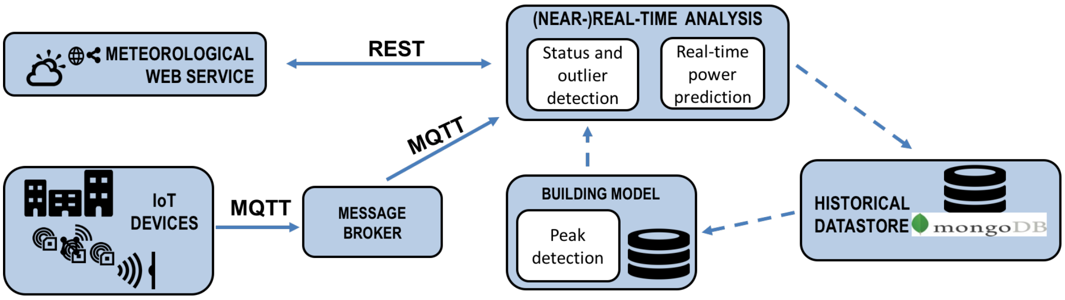

METEOROLOGICAL WEB SERVICE
REST
(NEAR-)REAL-TIME ANALYSIS
Status and outlier detection
Real-time power prediction
MQTT
IoT DEVICES
MQTT
MESSAGE BROKER
BUILDING MODEL
Peak detection
HISTORICAL DATASTORE
mongoDB

Fig. 5.17 Software infrastructure dataflow.

The PP algorithm uses the energy-related measures received from the *Message Broker* to develop the building model for the prediction of future power values. Whenever a new power measure is available for a building, the PP algorithm updates the corresponding model, to use it for predicting the next power values. PP contextually exploits the received measures to calculate the errors of the predictions previously performed for that values, in order to validate the model.

It computes the prediction error based on the expected power values according to the prediction model and the actual power value just received. After data have been processed, they are stored in the *Historical Datastore* together with the produced outcomes.

5.2.3.3.5 Status and Outliers Detection The *Status and Outlier Detection* (SOD) algorithm aims at automatically identifying the current operational phase for the building heating system. SOD also allows to detect abnormal values of the instant power measurements potentially occurred in the steady state.

The operational phases of the heating cycle are the *OFF-line* and *ON-line* phases, with the latter characterized by the alternation of a *transient* and a *steady* state. The *transient* state is characterized by a rapid increase of exchanged power. It usually occurs in the early morning when the heating is turned on. The **steady** state occurs after a *transient* state. It is a relatively constant exchange of power. Each of the above operational phases is characterized by a different amount of power exchange between the building and the HDN. Specifically, the power exchange occurs only during the ON-line phase, while it is absent otherwise.

The SOD algorithm relies on these expected trends in the power exchange to detect the operational phase based on the measured instant power values. Specifically, SOD adopts the *Exponentially Weighted Moving Average* (EWMA), proposed by [116] to filter noise and the effects of dynamic transient for the identification of faulty sensors. In this case EWMA is applied to detect the dynamic transient of the heating cycle and those variations in the steady state that can be filtered similarly to noise in a signal.

First, SOD computes the exponential mean (p_μ) and the corresponding standard deviation (p_δ) values of the instant power over a *sliding window* with one-day size. The day preceding the current day is used for positioning the *sliding window*. This time period allows computing p_μ and p_δ over a significant number of power values, but sampled in time instants not too distant from the current time. Power values in the $p_\mu \pm p_\delta$ range represent the expected power exchange during the steady state for the considered building. This range of power values is used as a reference for the identification of the operational phases. When a new instant power measurement p_{t_i} is acquired at time t_i, SOD assigns a class label describing the current operational phase of the building heating system. The phase categorization process works as follows. The *ON-line* phase is detected when the instant power p_{t_i} is different than zero ($p_{t_i} \neq 0$); otherwise the phase is identified as *OFF-line*. The *steady state* label is assigned when the instant power p_{t_i} is within the reference range of power values (i.e., $(p_\mu - p_\delta) \leq p_{t_i} \leq (p_\mu + p_\delta)$) for at least a minimum amount of time (*transition threshold*). Instead, the *transient state* is labeled when the instant power p_{t_i} is out the reference range of power values ($p_{t_i} < (p_\mu - p_\delta) \wedge p_{t_i} > (p_\mu + p_\delta)$) for more than the transition threshold.

For example, Figure 5.18 plots the instant power measures monitored in one day for a building. The figure also reports the range $p_\mu \pm p_\delta$ computed considering power values collected in the preceding day. Instead, Figure 5.19 shows the status labels assigned by SOD when two consecutive days of power measurements are considered. The assigned labels are equal to 1 for the transient state, to 0.5 for the steady state, and to 0 for the OFF-line phase. To increase readability, the power value reported in the figure has been normalized to the maximum power value. In both days the SOD algorithm identifies one transient phase (around 07:00), followed by one steady state.

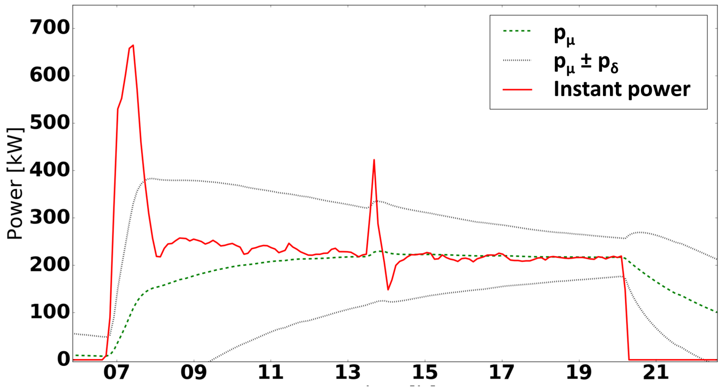


Fig. 5.18 Daily instant power profile against the expected power range during the steady state ($p_\mu \pm p_\delta$)

The SOD algorithm also allows to detect and remove abnormal values in the instant power measurements occurred during the steady state. An abnormal value is an observation that lies outside the expected range of values. It may occur either when a measure does not fit the model under study or when an error in measurement occurs (e.g., caused by faulty sensors). SOD categorizes this abnormal value as an outlier. When the operational phase is the steady state, a single isolated power measure p_{t_i} is categorized as outlier if its value is out of the range characterizing the steady state, i.e., $p_{t_i} < (p_\mu - p_\delta) \wedge p_{t_i} > (p_\mu + p_\delta)$. For example, Figure 5.19 shows an outlier detected during the steady state at around 3:00 p.m. in the second day of monitoring.

5.2.3.3.6 Peak Detection The *Peak Detection* (PD) algorithm aims at (i) *forecasting the peak power value in the transient state* and (ii) *identifying the peak power time instant*, separately for each building.

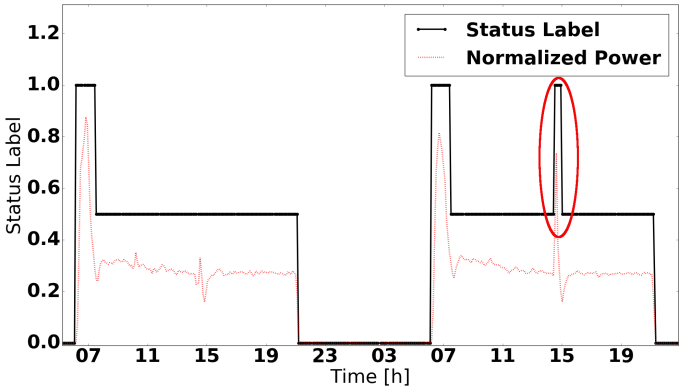


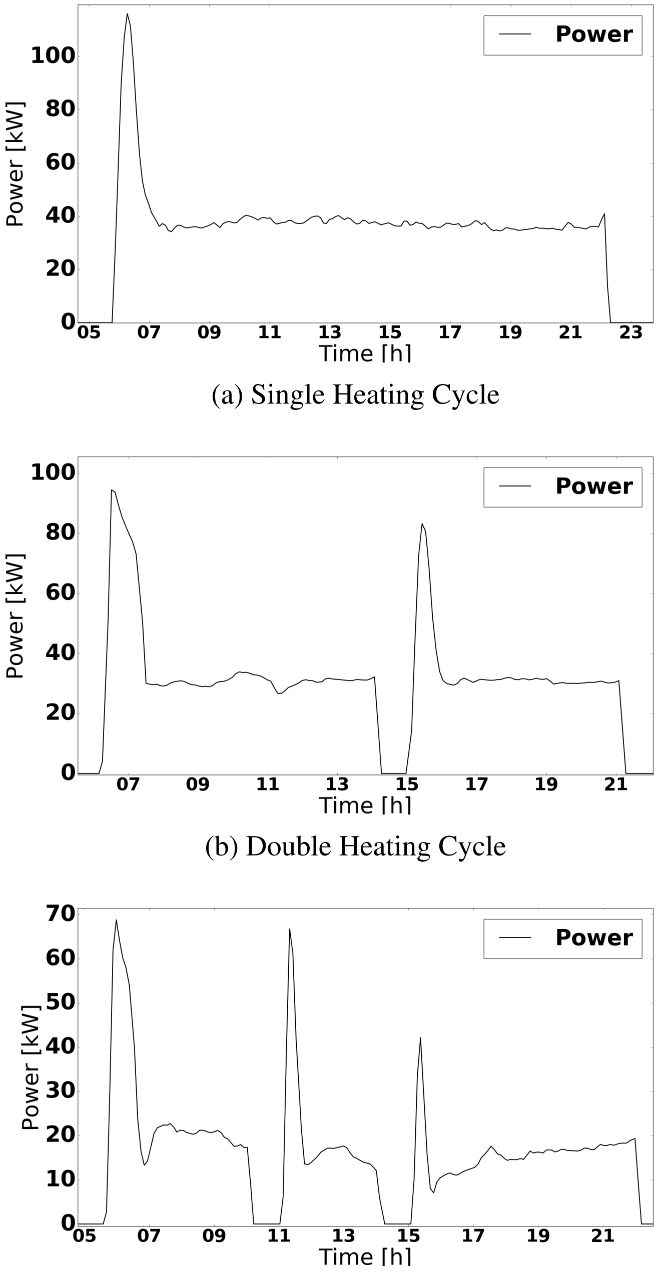
Fig. 5.19 Status detection with outlier value identification

The building heating cycle can have a single daily occurrence or it can be repeated more times per day. Thus, the PD algorithm can be employed only once or more times to forecast the peak power value in each transient state. Through the evaluation of the heating cycle for a large collection of buildings (about 300 buildings), three main building categories based on the number of interleaved heating cycles per day are identified . These categories represent buildings that are daily characterized by a *Single*, *Double* or *Triple Heating Cycle*.

Figure 5.20c reports an example of the daily power profile for the three categories. Note that consecutive heating cycles can show different peak power values in case of variation in the external temperature. The building internal temperature is affected by the values of the external temperature. When the external temperature decreases, the heating system reacts with a higher power exchange to keep the building internal temperature at the desired value of comfort. Thus, when the heating system turns on after an OFF-line phase with a lower external temperature, the heating cycle is characterized by an higher peak power value in the transient state.

To predict the peak power value in the transient state, the PD algorithm hypothesizes a relation between two quantities, named ψ and τ. ψ is the ratio between the peak power value in the transient state and the mean power value in the previous steady state. τ is the mean external temperature value in the previous steady state and OFF-line phase.

To properly model the relationship between the ψ and τ values for any of the three classes of buildings, the PD algorithm relies on the *Multivariate Adaptive Regression Spline* (MARS) [117] approach. MARS is a step-wise linear regression



(a) Single Heating Cycle

(b) Double Heating Cycle

(c) Triple Heating Cycle

Fig. 5.20 Heating Cycles in a day

for fitting variables in distinct intervals by connecting different splines with knots, thus it is suited to model a wide class of non-linear relations between variables. PD exploits the modified version of the MARS model proposed in [118] to predict the energy performance of buildings. PD learns a regression model for each building and for each peak using as training set the data collected in the past days. ψ and τ represent respectively the dependent and independent variables of the regression. Since all the other quantities of ψ and τ are known (from past data), the peak power value of the transient state appearing in ψ is the final target of the prediction.

For predicting the peak power value in the transient state of the first, second, and third heating cycles (named *first, second and third peak value*, respectively) in the target day, PD calculates the ψ and τ values as described below. PD predicts the first peak value for all three building categories (Single, Double, and Triple Heating Cycle), while the second peak value for two categories (Double and Triple Heating Cycle) and the third peak value for a single category (Triple Heating Cycle).

To forecast the first peak value, τ and ψ are computed as follows: τ is calculated as the mean external temperature value during the last steady state and OFF-line phase in the day preceding the target day; ψ is the ratio between the first peak power value (to be forecast) and the mean power in the last steady state of the day preceding the target day.

To forecast the second peak value, τ is the mean external temperature value during the first steady state and OFF-line phase in the target day; ψ is the ratio between the second peak value (to be forecast) and the mean power of the first steady state of the target day.

To forecast the third peak value, τ is the mean external temperature value during the second steady state and OFF-line phase in the target day; ψ is the ratio between the third peak power value (to be forecast) and the mean power in the second steady state of the target day.

The PD algorithm also infers the instant at which the peak power will occur. To this aim, PD computes the mean time where the past peaks have occurred, by considering a sliding window of fixed size preceding the current instant of time.

5.2.3.3.7 Power Prediction with Multiple Regression On the basis of the outcomes of the SOD and PD algorithms, the *Power Prediction* (PP) algorithm exploits the multiple version of the *Linear Regression with Stochastic Gradient Descent* (LR-SGD) [119] to predict the average power levels based on data from the Historical Datastore.

PP defines a *building model* based on a linear dependency between weather data and power level. PP relies on the assumption that the average power exchange for a building heating system at a given time instant is likely to be correlated with the surrounding weather conditions. Moreover, the average power levels are also likely to be temporally correlated with each other [120].

PP traines a multiple linear regression model for each building using historical data on weather conditions and power level. The training set is built using a fixed width sliding window mechanism, so the samples not older than a certain amount of time before the current time instant are included. For collecting samples, is assumed to split the window timeline in slots of the same duration (*slot duration*). Within a time slot, a single sample for each variable (power and weather parameters) is considered, computed as the mean value of the measures taken during the slot. Data sampling is performed for both training and test (i.e., future time slots) datasets.

The LR-SGD algorithm is characterized by a set of input features expressed through a n-dimensional vector x = $[x_1, \dots, x_n] \in \mathbb{R}^n$ and a target variable $y \in \mathbb{R}$ representing the objective of the prediction. The LR-SGD algorithm builds a hypothesis function $h : \mathbb{R}^n \rightarrow \mathbb{R} \mid y = h(\mathrm{x})$ so that, given an input vector x, function $h(\mathrm{x})$ provides a good estimation of the value of y. In the study, features in x correspond to the weather variables (air temperature, humidity, precipitations, wind speed, pressure), while y is the power level. Since power consumption and meteorological values differ in scale and measurement unit, data have been normalized. To preserve the original data distribution without affecting the prediction accuracy, the *Z-Score* standardization technique has been adopted.

The PP algorithm is structured into two phases: (i) *building model learning*, considering a collection of historical values for variables x and y; (ii) *prediction* of the future values of y, using the model generated in the first phase. The two phases are described below.

Model learning. This phase takes as input a training set where each training sample includes both the input vector x of meteorological data values and the corresponding known target variable y. The training set is built using a fixed width *sliding window mechanism*. Given a time instant t_i, the training window includes an ordered sequence of m data samples collected in t_i and in the previous $m-1$ instants t_j $(t_j < t_i)$. If the width of the sliding window (*training window size*) is very short, then almost instantaneous evaluation of the building's consumption is performed.

Instead, a too large time window allows analyzing many data on past building energy performance, but it may introduce noisy information in the prediction analysis. Since the data of training window are sampled in slots, the time interval between two consecutive training samples is fixed (*slot duration*). Given time t_i, is define as prediction time t_p the subsequent instant at which PP predicts the average power consumption. The time gap $\|t_p - t_i\|$ defines the *prediction horizon*.

In a training set of m samples defined over a time window, each sample $s^{(j)}$ is expressed by the pair $(\mathrm{x}^{(j)}, y^{(j)})$. For the LR-SGD algorithm, the hypothesis function $h(\mathrm{x})$ is expressed as follows:

$$h(\mathrm{x}) = w_0 + w_1 \cdot x_1 + \ldots + w_n \cdot x_n \tag{5.15}$$

where $w_1, \ldots, w_n$ are the weights characterizing the relationship between the average power consumption y and meteorological data values in x (i.e., $x_1, \ldots, x_n$), while w_0 is the intercept value. Without lack of generality, by defining x_0=1 Equation 5.15 can be expressed using the following concise expression:

$$\begin{gathered} h(\mathrm{x}) = \sum_{i=0}^{n} w_i \cdot x_i = \mathrm{W}\mathrm{x}^T, \\ \mathrm{W} = [w_0, \ldots, w_n],\ \mathrm{x} = [x_0, \ldots, x_n]. \end{gathered} \tag{5.16}$$

In the training phase, the LR-SGD algorithm learns the values of weights in vector W. The least-squares cost function $J^{(j)}$ in Equation 5.17 is used to measure the distance between the actual value of y and the computed value $h(\mathrm{x})$ for each training sample ($J^{(j)} = y^{(j)} - h(\mathrm{x}^{(j)})$). The overall least-squares cost function on the whole training set is computed as

$$J(\mathrm{W}) = \frac{1}{2} \sum_{j=1}^{m} (J^{(j)})^2 = \frac{1}{2} \sum_{j=1}^{m} (y^{(j)} - h(\mathrm{x}^{(j)}))^2. \tag{5.17}$$

Algorithm 1 reports the process for weight computation in LR-SGD. The algorithm iteratively considers the samples in the training set. It progressively updates the values of weights w_i in W by following the direction of steepest decrease of $J^{(j)}$. The algorithms is driven by two user-specified parameters: the *learning rate* α and the *number of iterations* on the whole training dataset.

ALGORITHM 1: Weights update in Stochastic Gradient Descent

```
for j = 1, ..., m do
    for i = 0, ..., n do
        $w_i := w_i + \alpha \cdot ((y^{(j)}) - h(\mathrm{x}^{(j)})) \cdot x_i^{(j)}$
    end
end
```

Unlike Batch Gradient Descent (BGD), which updates weights after the whole training set is processed, with the Stochastic Gradient Descent (SGD) approach the overall cost function $J(\mathrm{W})$ quickly converges to a value close to the minimum.

Prediction. Once the learning model has been created, it is used to predict the future power level y using the corresponding vectors of known input features x representing meteorological data values $\mathrm{x}^{(j)}, j = m+1, \ldots, +\infty$.

Hence, given the prediction of the weather variables for a future target time ($\hat{\mathrm{x}}^{(j)}$) and the hypothesis function for the model $h(\mathrm{x})$, the estimation of the corresponding power value is calculated as:

$$\hat{y}^{(j)} = h(\hat{\mathrm{x}}^{(j)}) = \sum_{i=0}^{n} w_i \cdot \hat{x}_i^{(j)}. \tag{5.18}$$

The PP algorithm also relies on the outcome of the SOD and PD algorithms. Through SOD, PP can identify when the power prediction is performed for the transient or the steady state. Moreover, since during transient state the power values might not have a clear linear dependence from weather data, PP uses the outcome of PD algorithm to better approximate the transient power profile, through a linear interpolation.

5.2.3.3.8 Building efficiency characterization The *energy signature* is a world wide recognized method for the analysis of building energy consumption. This method was developed in the 80's by American government after the oil crisis, it has been introduced in the European regulatory framework (EN 156036:2008) and was recognized at Italian level in UNI (11300:2008). The energy signature method has been used in many studies to extract the total heat loss coefficient of a building [121–123]. The latter is recognised as an interesting key energy indicator [121, 124] of a building.

Specifically, the *total heat gain* in a building (denoted as Q_{tot}) is expressed as $Q_{tot} = Q_{loss} + Q_{dyn}$ where Q_{loss} represents the ventilation and thermal losses and Q_{dyn} is the heat dynamically stored or released by the building. The term Q_{loss} is expressed as $Q_{loss} = K_{tot} \cdot (T_{in} - T_{ex})$ where T_{in} is the *internal* temperature of the building and T_{ex} is the *external* temperature of the ambient, while K_{tot} is the *total heat loss coefficient* of the building. The term Q_{dyn} takes into account the dynamic of the building. Since Q_{dyn} is related to the thermal inertia of the building, the estimation of its value may be a complex task. Q_{dyn} is expressed as $Q_{dyn} = C \cdot \frac{\delta T}{\delta t}$ where C is the thermal mass of the building, representing the building capability to release or store heat. When the Q_{dyn} value is approximated to zero, the steady-state analysis of the building efficiency can be performed [121, 122]. Specifically, the dynamic contribution Q_{dyn} decreases when energy data are analyzed at coarse granularity (as monthly, weekly) [124]. Instead, these effects are emphasized when finely-grained data are analyzed (as every 15 minutes)[121].

The total heat gain Q_{tot} in a building can also be expressed based on the contribution of four terms as in $Q_{tot} = Q_h + Q_{el} + Q_p + Q_{sun}$ where Q_h is the power supplied by the heating system, while Q_{el}, Q_p, and Q_{sun} represent the heat gains due to electricity usage (Q_{el}), people presence (Q_p) and solar radiation (Q_{sun}), respectively. The influence of random variables (as occupancy, wind, solar gains) and the heat gains due to the electricity usage can be neglected when coarsely-grained data are analyzed [122]. In this case, terms Q_{el}, Q_p, and Q_{sun} can be approximated to zero.

This study focuses on the steady-state analysis of the building efficiency. Consequently, energy data are analyzed at different coarse granularities to neglect both the dynamic contribution Q_{dyn} of and the influence of random variables. Furthermore, K_{tot} estimation is normalized to a single unit of volume, i.e., W/m^3. It follows that the total heat gain Q_{tot} in the building is equal to the power supplied by the heating system per unit of volume and to the ventilation and thermal losses (i.e., $Q_{tot} = Q_h = Q_{loss} = K_{tot} \cdot (T_{in} - T_{ex})$). The linearity of the model has been evaluated as done in [123].

To address the former objective, two indicators have been designed: (i) an intra-building indicator, which addresses the question of abnormal power consumptions given the current conditions with respect to past energy demand in the same conditions; to this aim, the most recent power consumption data for each building is compared to its own historical energy signature, thus identifying changes with respect to previously modeled energy behaviors of the same building; (ii) an inter-building indicator, comparing the building efficiency, given by its energy signature, with respect to nearby and similarly characterized buildings, where similarity takes into account spatial co-location, building size, and usage patterns, e.g., residential or office or public building.

The key intuition behind the designed indicators is based on exploiting the energy signature defined by K_{tot} to compute the expected power consumption for given contextual conditions in a specific time period. Contextual conditions can include any relevant attribute for the specific problem under investigation. In the current implementation, the difference between the outdoor T_{ex} and the indoor T_{in} temperatures, and the specific building characteristics (e.g., position, size, etc.) are considered as the key attributes defining a context. If the given temperatures and time periods are the current ones (e.g., current outdoor temperature, now), and is consider the same building, then the intra-indicator is obtained, whereas using the energy signature of a group of similar buildings, with respect to the one under examination, leads to the inter-building indicator.

Finally, to reach the goal of forecasting the power demand, the same approach can be used, by exploiting the energy signature with a predicted value of outdoor temperature T_{ex} and a fixed value of target indoor temperature T_{in}, with the former obtained by weather forecasts, and repeating the computation for each future time period and each building of interest. Such estimation of future power demand helps district heating providers to better predict the energy demand.

To evaluate and rank building efficiency, its energy signature, defined by its total heat loss coefficient K_{tot}, is exploited. To this aim, the instantaneous power supplied by the heating system per unit of volume (Q_h) is correlated with the difference between the indoor temperature T_{in} and the outdoor temperature T_{ex}. The correlation is based on a linear regression of average power samples per unit of volume, aggregated at different time granularity levels. This process has been designed and developed as a cloud-based service on top of a MongoDB distributed cluster, and it is detailed in the following.

The analysis can be focused by filtering heating power consumption in a given date range t_{period} (e.g., a winter period, a month) and also in specific day time slots of interest t_{slot} (e.g., [5:00p.m-7:00p.m.], [10:00a.m.-7:00p.m.], [10:00a.m.-9:00p.m.]). Hence the time-specific energy signature will be relevant only for those subsets of time periods, both in the characterization and in the prediction applications. Focusing the energy signature by restricting the day and time periods helps in modeling different behaviors such as those in the steady state, in specific seasons, during office hours, etc.

The instantaneous power samples of interest are aggregated by computing the mean value in a given time window, t_{window} (e.g., hourly, daily, weekly). The resulting value indicates the mean power consumption over an hour, a day, or a week. While longer periods are more error-prone due to the large variance of the outdoor temperature T_{ex}, too short periods take into account the thermal inertia of the building, as discussed in Section 5.2.3.3.8.

The application service can estimate K_{tot} by considering any combination of t_{period}, t_{slot}, and t_{window}, which are user-defined parameters. It will be up to the end-user presentation interface to choose the best indicators in any given context.

For each building, the instantaneous power values per unit of volume, and the difference between the indoor T_{in} and the outdoor T_{ex} temperatures are extracted from the MongoDB datawarehouse and aggregated over t_{window}. The result includes all the mean power values per unit of volume and the average difference T_{ex}-T_{in} for each t_{window}.

Given the mean power values (denoted as y) and the mean temperature difference values (denoted as x) it first compute $\sum_{i=1}^{n} x_i$, $\sum_{i=1}^{n} y_i$, $\sum_{i=1}^{n} x_i y_i$. Then the a and b terms of the linear equation $y = a + bx$ is computed as follows.

$$a = \frac{(\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} x_i^2) - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} x_i y_i)}{n(\sum_{i=1}^{n} x_i^2) - (\sum_{i=1}^{n} x_i)^2}$$

$$b = \frac{n(\sum_{i=1}^{n} x_i y_i) - (\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} x_i)}{n(\sum_{i=1}^{n} x_i^2) - (\sum_{i=1}^{n} x_i)^2}$$

The b value corresponds to the total heat loss coefficient of a building, i.e., K_{tot}.

The MapReduce jobs of the estimation of K_{tot} were developed through custom JavaScript functions, and executed using the MongoDB MapReduce framework. For each document, the *map* function emits an object containing the values needed to compute the energy signature equation parameters for the related building. The *reduce* function is in a simple sum over all the records of the same building. Finally, a *finalize* function uses the aggregated results to compute the energy signature equation and returns the K_{tot} estimation for each building. Energy signatures can be aggregated over similar buildings by computing the average K_{tot} among them.

5.2.4 Internal Temperature Simulation

Figure 5.12 shows the involved modules of the SMIRSE infrastructure for Internal Temperature simulation. From the data sources BIM, GIS, Weather and ambient indoor sensors are involved in the simulation process.

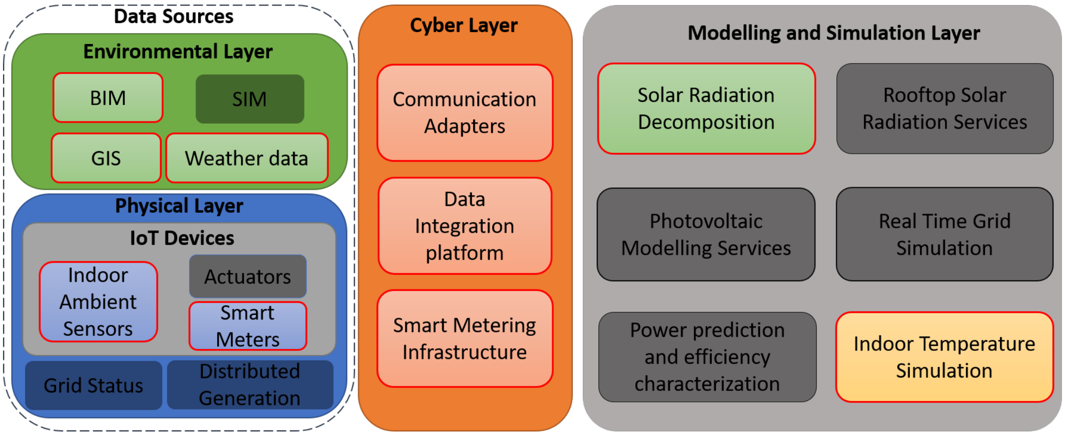

Data Sources
Environmental Layer
BIM
SIM
GIS
Weather data
Physical Layer
IoT Devices
Indoor Ambient Sensors
Actuators
Smart Meters
Grid Status
Distributed Generation
Cyber Layer
Communication Adapters
Data Integration platform
Smart Metering Infrastructure
Modelling and Simulation Layer
Solar Radiation Decomposition
Rooftop Solar Radiation Services
Photovoltaic Modelling Services
Real Time Grid Simulation
Power prediction and efficiency characterization
Indoor Temperature Simulation

Fig. 5.21 Schema of SMIRSE Platform with focus on Indoor Temperature Simulation

Figure 5.22 shows the principal components that are deployed on-site. Buildings are equipped with a: *i)* Wireless Sensor Network (WSN) with low power micro controllers that monitor indoor temperature and air humidity in rooms; *ii)* with a smart-meter that monitors electricity consumption data by communicating with classic Low-Voltage meter exploiting Power-line communication. Both WSN and smart meters are connected to a gateway that sends data to the Cloud, in particular to the *Store Manager*. The interoperation between collected data in the cloud, weather information, BIM and GIS models is given by the data source integration platform (see Section 5.1.2.2) that provides users with: *i)* simulation and modelling of building energy performance; *ii)* building status monitoring and visualization of collected data; *iii)* user awareness.

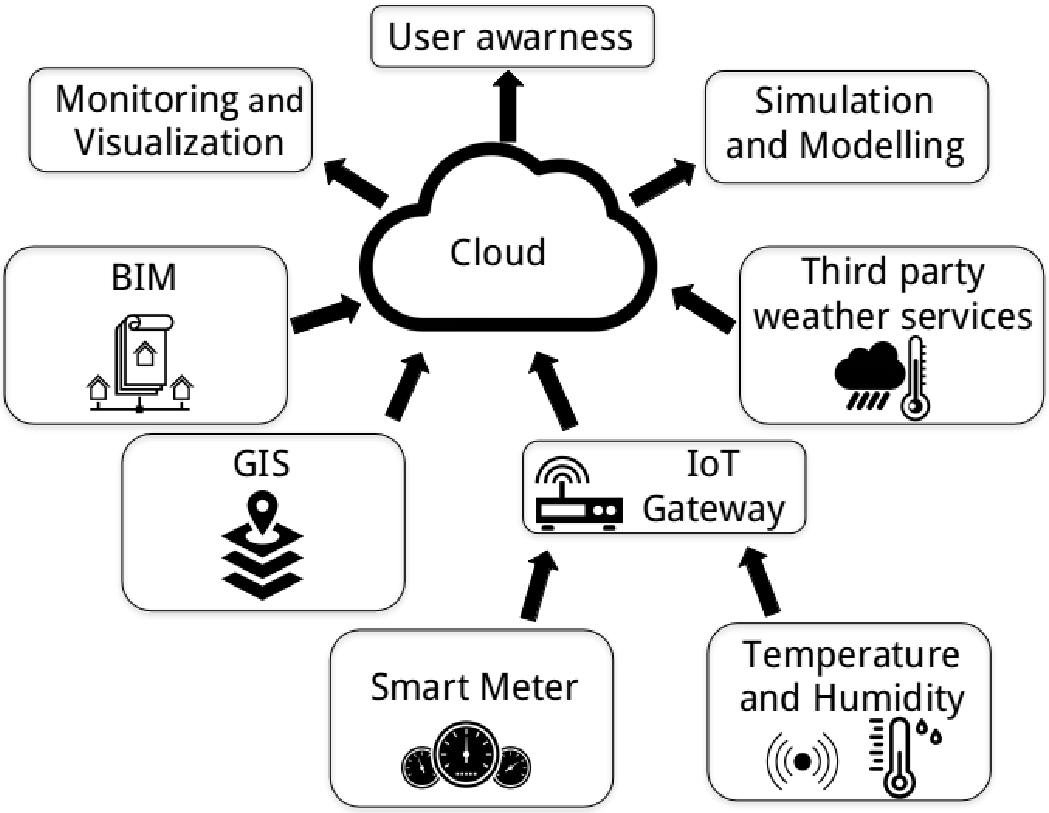


Fig. 5.22 Schema of on-site deployment for the proposed solution.

In this section, the followed methodology to perform energy simulations starting from BIM and correlating IoT data within an integrated process is described. The building energy modelling and monitoring approach is one of the most challenging topic in Smart City scenario. In this context: i) BIM establishes a proper knowledge of the buildings; ii) technical investigations aimed at energy efficiency are required by EU Energy Performance of Buildings Directive [125]; iii) IoT links different domains and provides real data from the field. These factors constitute the key issues for this research development.

To achieve it, BIM models have been developed with *Autodesk Revit 2016* [126] starting from on-site surveys. They include: *i)* accurate building envelope characterizations in terms of correct stratigraphy, thermal and physical properties; *ii)* facility management information (e.g. room type and occupants); *iii)* materials nomenclature standards. Thus, they become a significant repository of graphical and alphanumeric information useful for energy analysis. To properly set the model to perform energy simulations, the BIM needs simplifications by removing excessive details in the

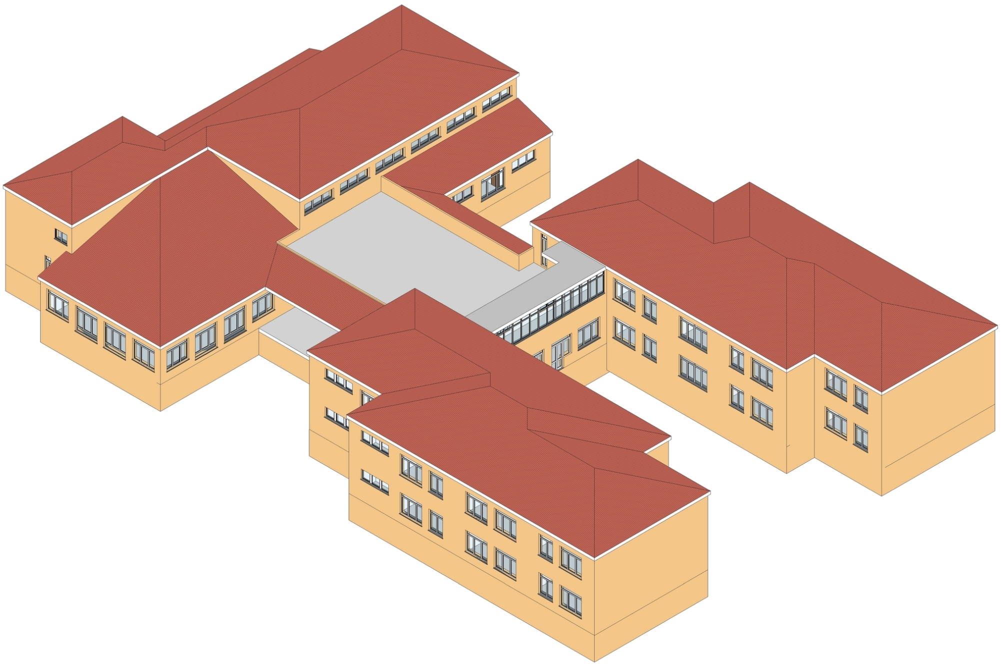


Fig. 5.23 BIM model.

architectural model, such as decorations and staircases (see Figure 5.23). These details are unnecessary and get slow the simulation or can even include inaccuracies in final results. Figure 5.24 shows the *Energy Analysis Model* (EAM) that consists of rooms and analytical surfaces generated from the BIM model and exported by *Revit* in gbXML data-format. Figure 5.25 reports the proposed energy modelling optimization process. The **EAM Simulation Engine** block performs building simulations using EnergyPlus [127]. It needs the following inputs:

- *Geometry and materials* of building components (e.g. stratigraphy and shades) and their thermal and physical properties. These come from BIM models;
- *Real weather data* such as *i)* air dry-bulb temperature, *ii)* solar radiation and *iii)* average air temperature;
- Data retrieved from *Heating Ventilation and Air Conditioning systems* such as *i)* nominal power and flow rate of radiators, *ii)* nominal power and efficiency of boiler, *iii)* climate control unit, *iv)* on/off profile of the heating system;
- *Occupancy* of rooms, including number of users and time-shifts.

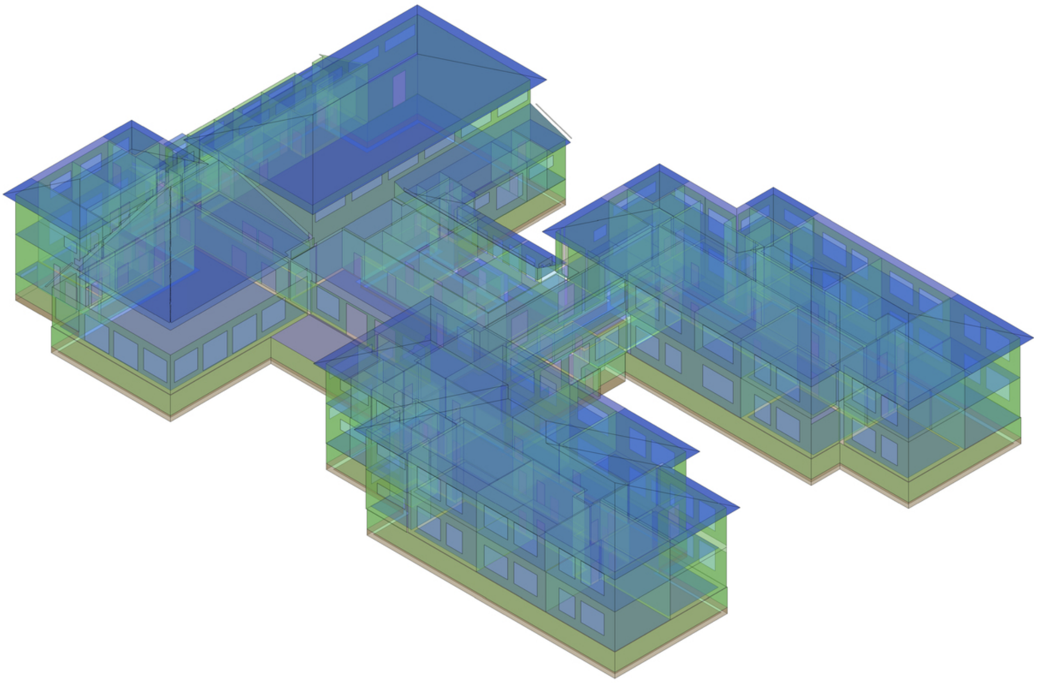


Fig. 5.24 Energy Analysis Model (EAM).

The outputs of the *EAM Simulation Engine* block are radiant, operating and indoor temperature. It also provides the energy consumption profiles of the building. Traditionally energy simulations with EnergyPlus are performed using Typical meteorological year (TMY). TMY is obtained by averaging hourly meteorological measurements collected for 10 years. Thus, it does not represent real weather conditions. As a strong point of the simulations, in the software platform (see Section 5.1.2.2) third-party weather data-source from the nearest weather station are integrated. Hence, real weather information (i.e. solar radiation, outdoor air temperature and humidity) are considered in the simulation process replacing the default TMY.

Indoor air temperature and humidity are sent every 15 minutes by IoT devices and collected in the *Store Manager* of the proposed platform (see Section 5.1.2.2). Such data are needed by the *EAM Validation* block in Figure 5.25 to validate the performed simulations. This validation is done by comparing the results of the *EAM Simulation Engine* with the real measured values coming from the deployed IoT devices. Analysing temperature and consumption trends, factors that may affect the

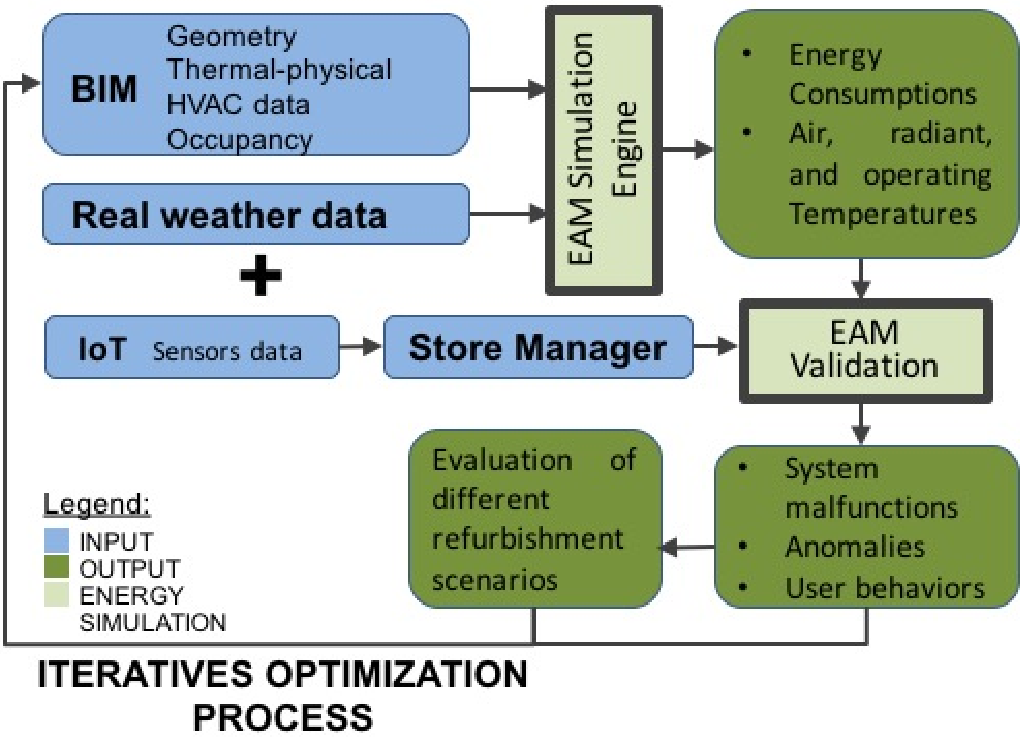


Fig. 5.25 Proposed energy modeling optimization process.

energy model can be identified, such as user behaviors, malfunctions and anomalies in the system. For instance, user-awareness applications can help in minimizing not energy efficient behaviors. Whilst maintenance activities can be planned to monitor and solve identified anomalies (e.g. by comparing measured and simulated data, it is possible to discover irregular trends of real indoor temperatures due to faults in on/off schedules of the heating system or efficiency losses of the building-system).

In addition, BIM models can be used to evaluate different design and/or refurbishment scenarios (e.g. external/internal coat application, fixtures replacement and power peaks regulation). Thus, this updated BIM model is a new input for the energy modelling optimization process. This process is iterative and can help building- and energy-managers in evaluating the best solution for both energy performances and Return of Investment.

Chapter 6

Case studies and Results

The modules of the SMIRSE modelling and simulation infrastructure have been tested mainly in the city of Turin as case study. Turin is a city located in Piedmont, north-west of Italy. It has a total area of 130.01 km^2 and a population of $890'133$ inhabitants with a density of $6'846.65$ $inh./km^2$.

6.1 Metrics for evaluating simulation performance

In this Section we present the metrics used for evaluating the performances of our simulations.

In particular, the following indicators of dispersion are considered. *i)* The Mean Bias Difference (MBD) measures the average squares of errors between predicted and measured values. *ii)* The Root Mean Square Difference ($RMSD$) represents the standard deviation of differences between predicted and observed values. *iii)* The Mean Absolute Difference (MAD) is defined as the average of the absolute difference of two variables X and Y. *iv)* The Coefficient of determination (r^2) indicates the proportion between the variance and the predicted variable. *v)* The Standard Error of Regression (S) expresses how wrong the regression model is on average using the units of the response variable. Small values of S identify a high accuracy of prediction because 95% of predicted values will fall in the range of $\pm 2S$. *vi)* The Mean absolute percentage error ($MAPE$) represents the average difference between the actual and the forecast expressed as a percentage of the actual (or the forecast). *vii)* The Symmetric mean absolute percentage error ($SMAPE$) is a variation on the

MAPE that is calculated using the average of the absolute value of the actual and the absolute value of the forecast in the denominator. These indicators of dispersion, with exception of r^2, are expressed in percentage of mean measured values rather than in absolute units as suggested in [128]. Furthermore, the two following indicators for the overall performance are used for the evaluation. *i)* The Willmott's Index of Agreement (WIA) is the standardized measure of the degree of model prediction error. It varies between 0 and 1. *ii)* The Legates's Coefficient of Efficiency (LCE) is the ratio between the mean square error and the variance in the observed data, subtracted from unity. LCE can vary between $-\infty$ and 1, where 1 represents the perfect model.

6.2 Solar energy simulation

In this section, the experimental results performed exploiting the infrastructure presented in Section 5.2.1 are presented. The solution has been tested in a district of Turin and the case study area described is described in the followings.

The district under analysis is *La Crocetta*, where there is our University campus. *La Crocetta* is located in the city center with an area of about 3.7 km^2 and around 2200 residential buildings. It has been selected because of its buildings, which are heterogeneous in terms of construction type and period. DSM and Cadastral maps for this area have been provided by the city council. DSM has a resolution of 0.25 m^2, which gives the possibility to describe with high accuracy rooftops, highlighting encumbrance like chimneys and dormers. Trough third-party web services, meteorological data (i.e. solar radiation and air temperature) are retrieved by the weather station in our University campus that is located in the middle of the case study district. The weather station collects global horizontal radiation by a *first class* pyranometer that samples every minute. Then, these samples are averaged and provided every 15 minutes. As proposed by [129], we exclude samples of solar radiation with: *i)* the altitude lower than 7° and *ii)* the clearness index lower than 0 and higher than 1. Furthermore, we excluded the measured samples of global horizontal radiation with higher values than in clear-sky condition, again as suggested by [129].

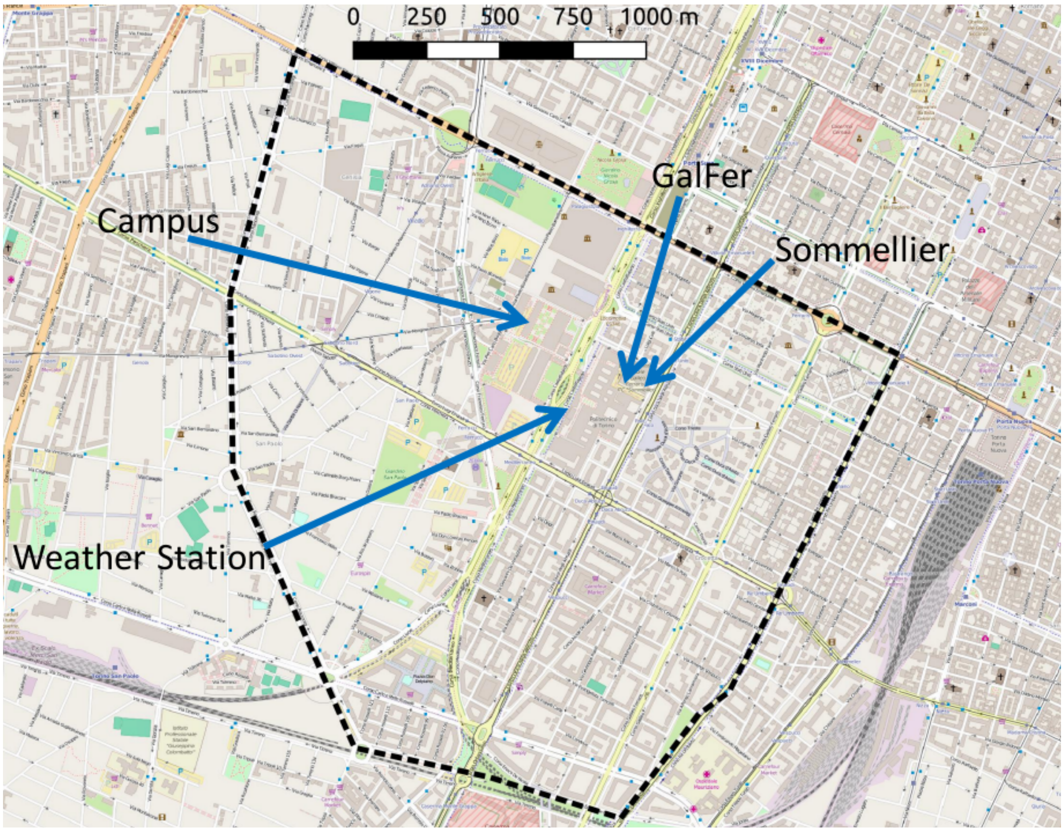


Fig. 6.1 PV systems and Weather station

The performance of simulation are evaluated exploiting the indicators presented in Section 6.1 and reported by Gueymard [128] for evaluating solar radiation models. To validate the solar radiation simulator, we considered three different PV systems and a tilted solar radiometer located in the case study district. Thus, we compared results on PV energy simulation with real energy production data. The first PV system under analysis has been installed in our University campus (*Campus*) in 2008 with an inclination of 26° and an orientation of 23° (considering south 270°). It is a building integrated mono-crystalline system with an efficiency η_{pv} of 20.2% and a nominal power of 15.28 kW. The other two PV systems have been installed in 2004 in two public high schools: *i) Istituto Galileo Ferraris* (*GalFer*) and *ii) Istituto Sommelier* (*Sommelier*). Both PV systems are poly-crystalline with a nominal power of 13.20 kW and 19.80 kW respectively and an efficiency of 13.1%. The $GalFer$ PV system is free-standing with an inclination of 35° and an orientation of 240°. The *Sommelier* PV system is building integrated with an inclination of 20° and an orientation of 240°. Both GalFer and Sommelier have been monitored during the

European project PERSIL [8] with whom we partially share the same case study. PERSIL aims at analysing the energetic performance of PV systems and solar thermal plants. Table 6.1 summarizes the characteristic of these three PV systems. Moreover, to validate the simulations of the *Real-sky calculation service*, we compared our results with measured data collected by a second solar radiometer, different from the one in the weather station. It is a mono-crystalline digital pyranometer and it is installed very closed to the PV system in our University campus. Figure 6.1 shows the case study area reporting the locations for the weather station and the three PV systems.

Table 6.1 PV system characteristics

	Campus	GalFer	Sommelier
Nominal Power [kW]	15.28	13.20	19.80
Module Power [W]	283	165	165
Number of PV modules	54	80	120
Module Efficiency [%]	20.2	13.1	13.1
Module Temp. Coef. [%/°C]	0.38	0.48	0.48
Slope [°]	26	35	20
Aspect [°] (South 270°)	23	240	240
Installation year	2008	2004	2004

First, the results achieved by the integration of the *Solar Radiation Decomposition* and *Real-sky calculation service* are compared with the real data collected by the digital pyranometer in our University campus in order to select the best decomposition model. This test has been performed for all the solar radiation decomposition methods integrated in the proposed infrastructure. After the selection of the best decomposition model the *Photovoltaic energy estimation* service is tested by simulating the generation profiles of the *Campus*, *Sommellier* and *GalFer* PV systems. Finally to highlight the advantages of our solution the comparison of our simulation results with the results obtained in PERSIL [8] is performed with one day time-interval for *Sommellier* and *GalFer*.

6.2.1 Selection of best decomposition model for case study area

In order to select the best decomposition model for the case study area, all the models implemented in the infrastructure have been tested. This because the accuracy of each model is strongly affected by the geographic location (see Section 5.2.1.2.1).

The best model has been selected by comparing solar radiation simulations with measurements sampled by the solar radiometer in our University campus. The tests are performed by simulating solar radiation from the 1^{St} of January 2014 to 31^{St} of December 2015. Figure 6.2 shows the simulations of *GHI* in a spring week for each decomposition model.

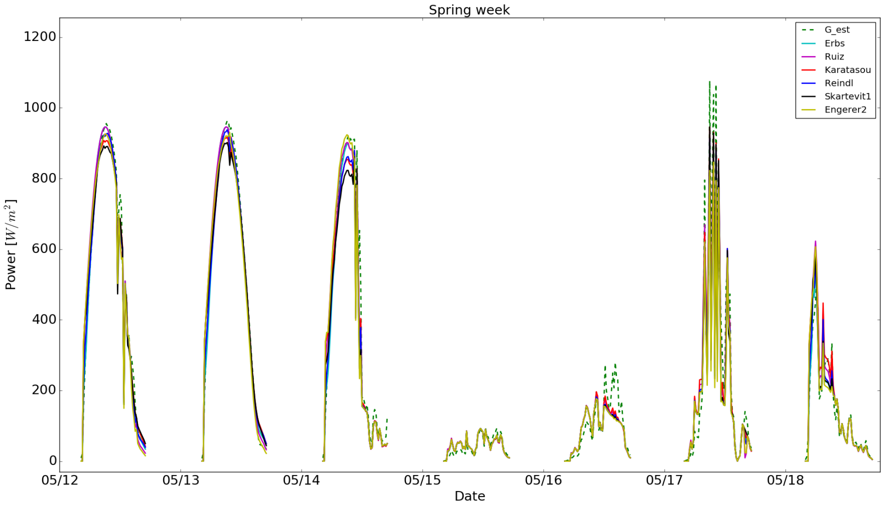


Fig. 6.2 *GHI* simulation for each decomposition model in a spring week

Table 6.2 reports some of the performance indicators presented in Section 6.1 for each decomposition model simulation using time-intervals of 1 hour and 15 minutes respectively.

The performance indicators for hourly simulations show that the best model for the case study area (Turin, Italy) is Karatasou [11]. Indeed, among its performance indicators, this model has the best values with exception of *MBD*, which is -1.88%. Indeed for *MBD* only, Karatasou is worst than the Ruiz-Aris model [96] with the $MBD = -0.68\%$. Considering 15 minutes simulation, Karatasou again has the best performance indicators with exception of *MBD* and *MAD*. Indeed, Ruiz-Aris achieves the best *MBD*, which is -0.5%; while Reindl [97] has the best *MAD* equal to 16.21%.

Table 6.2 highlights that all these decomposition models have better performance in simulation with 1 hour time-interval. This is also confirmed by Gueymard et al. [99], where authors proves that solar decomposition models developed to have the

Table 6.2 Performance indicators for simulations of solar radiation

Model	Time resolution	*LCE*	*MAD* [%]	*MBD* [%]	r^2	*RMSD* [%]	*WIA*
Reindl [97]	1 hour	0.80	15.38	-6.22	0.95	20.93	0.98
Engerer [95]		0.77	17.48	-5.96	0.93	24.16	0.98
Skartveit [98]		0.77	17.46	-8.62	0.93	24.10	0.98
Karatasou [11]		0.80	15.44	-1.88	0.95	20.73	0.98
Ruiz-Arias [96]		0.80	15.31	-0.68	0.94	20.75	0.98
Erbs [94]		0.80	15.47	-6.54	0.94	21.42	0.98
Reindl [97]	15 minutes	0.78	16.21	-6.08	0.93	23.52	0.98
Engerer [95]		0.75	19.09	-5.82	0.90	27.83	0.97
Skartveit [98]		0.75	18.75	-8.55	0.91	26.84	0.97
Karatasou [11]		0.78	16.57	-1.40	0.93	23.42	0.98
Ruiz-Arias [96]		0.78	16.65	-0.5	093	23.62	0.98
Erbs [94]		0.78	16.95	-6.36	0.93	24.35	0.98

best performance with slow variations of k_t (e.g. hourly resolution) cannot provide same performance with faster variations of k_t (e.g. 15 minutes resolution).

Table 6.3 Performance indicators for simulations of solar radiation applying Karatasou model [11]

Period	Weather	*LCE*	*MAD* [%]	*MBD* [%]	r^2	*RMSD* [%]	*WIA*
Winter	Sunny	0.75	14.29	-2.26	0.93	17.94	0.98
	Cloudy	0.75	17.17	-2.31	0.9	26.01	0.97
	Rainy	0.73	25.93	5.25	0.91	37.99	0.97
Spring	Sunny	0.8	13.14	-3.95	0.95	16.99	0.98
	Cloudy	0.79	15.01	-1.13	0.93	21.18	0.98
	Rainy	0.77	18.57	-0.38	0.93	30.72	0.97
Summer	Sunny	0.81	9.5	-4.39	0.96	11.91	0.98
	Cloudy	0.75	16.31	-3.01	0.91	21.79	0.97
	Rainy	0.71	22.6	-3.56	0.87	33.54	0.96
Autumn	Sunny	0.8	9.15	-0.13	0.95	12.06	0.98
	Cloudy	0.72	16.26	-1.66	0.9	21.01	0.96
	Rainy	0.71	24.63	6.73	0.91	32.81	0.99

Table 6.3 reports the performance indicators of solar radiation simulation for season and weather conditions (Sunny, Cloudy and Rainy) using the Karatasou model [11]. Table 6.3 show that best results are achieved in sunny days of summer

and autumn. Indeed for all seasons, *MAD*, *MBD* and *RMSD* have lower values in sunny days. The r^2 is higher or equal to 0.9 for all seasons except for of rainy days in summer. On the other hand, simulations in winter season provide the worst results with respect to the other seasons, in particular during rainy days. More in general, rainy days for the fours seasons do not have the same good performance than in sunny days, with *RMSD* higher than 30%. This is due to the accuracy of the integrated decomposition models in evaluating direct and diffuse components of solar radiation in rainy days. However, the results of 15-minutes simulations in rainy day (worst case) are still acceptable. In further studies the exploitation of sensors to sample direct and diffuse radiation will improve the performance of the simulations.

6.2.2 Evaluation of PV system simulation against measured data

In order to validate the accuracy of the *Photovoltaic energy estimation* in generating power load profiles, simulations for the three PV systems introduced in the case study at the beginning of this Section are performed. Those simulation are compared with the measurements of power generation for the *i) Campus*, *ii) Sommellier* and *iii) GalFer* PV systems. In the simulations an yearly degradation factor of 1% for the efficiency coefficient η has been taken into account for each PV system. In the calculation of the $T_{sol-air}$ temperature , the loss factor k has been set to 0.05 as reported by [52].

Figure 6.3 shows the comparisons among our simulations and measured generation profiles for the *Campus* PV system in three generic days of autumn: *i)* sunny, *ii)* cloudy and *iii)* rainy. It is worth noting that the trends of our results follow the real behaviour of the PV system with a good accuracy.

Table 6.4 reports the results in terms of performance indicators of instant power for the *Campus* PV system. The best results are achieved in sunny days of autumn. Indeed, r^2 is higher than 0.9; *RMSD* and *MAD* are respectively lower than 13% and 10%.

Simulations performed in summer provide the worst results with respect to the other seasons. This is related to the fact that PV system in summer period are more affected by the temperature effect on the efficiency η of the module.

Considering the whole time period (18 months), the accuracy of simulation performance for daily energy production increases with respect of 15 minutes time

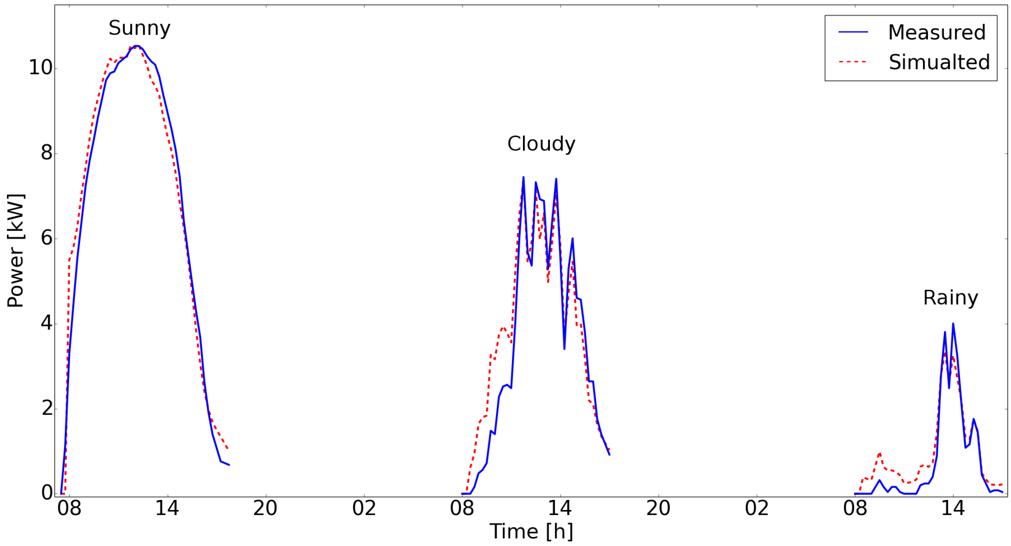


Fig. 6.3 Campus PV system simulation vs measurements

Table 6.4 Campus performance indicators for our solution

Period	Weather	*LCE*	*MAD* [%]	*MBD* [%]	r^2	*RMSD* [%]	*WIA*
Winter	Sunny	0.8	10.72	4.28	0.95	13.67	0.98
	Cloudy	0.77	15.7	6.59	0.93	20	0.98
	Rainy	0.66	33.13	17.5	0.89	41.87	0.96
Spring	Sunny	0.78	12.57	5.82	0.94	16.17	0.98
	Cloudy	0.74	18.27	2.13	0.9	25.29	0.97
	Rainy	0.73	21.69	2.07	0.9	30.72	0.97
Summer	Sunny	0.7	14.3	-0.94	0.89	18.04	0.97
	Cloudy	0.68	19.86	-1.86	0.86	26.02	0.96
	Rainy	0.66	26.2	-1.89	0.82	37.27	0.95
Autumn	Sunny	0.78	9.24	3.93	0.94	12.6	0.98
	Cloudy	0.7	16.91	3.82	0.89	22.09	0.97
	Rainy	0.67	28.85	13.35	0.89	36.94	0.97
18 months period for 15-min simulation	—	0.72	18.85	1.8	0.9	25.21	0.97
18 months period for daily simulation	—	0.82	9.64	7.93	0.96	11.68	0.99

resolution. Daily simulations are computed as the integral of 15-minute simulations; thus, errors tend to be attenuated. The indicator results are as follows: $LCE = 0.82$; $MAD = 9.64\%$; $MBD = 7.93\%$; $r^2 = 0.96$; $RMSD = 11.68\%$ and $WIA = 0.99$.

Considering the overall energy production for the analysed time period, we achieve $MAD = 0.2\%$; $MAD = 0.2\%$ and $RMSD = 1.27\%$.

The other PV systems under analysis, both *Sommelier* and *GalFer*, have been monitored from the 1^{St} of March 2010 to the 22^{Nd} of February 2011 and the data sampling has been done by the PERSIL project consortium [8]. For the *Somelier* system data regarding the 28^{th} of March 2010 where excluded in the simulation due to a wrong collection of samples of instant power. For both system the 15^{th} of December 2010 has been excluded due to a a lack of data of external temperature in the weather data. In Table 6.5 and 6.6 the performance indicators of our simulation compared with measured data for *Sommellier* and *GalFer* respectively are presented. The simulations are performed with different time-scales: *i)* 15 minutes, *ii)* 1 day and *iii)* 1 year. For both *Sommelier* and *Galfer*, the accuracy of performance indicators of daily and yearly simulation increase with respect to 15 minutes simulations. LCE, WIA and r^2 are not calculated for yearly values because they can be applied to series and not to a single value. With respect to 15 minutes simulations, the accuracy of performance indicators for *Sommelier* simulations is in line with the one of *Campus*. Apparently, *GalFer* presents the worst accuracy. However, if the analysis is restricted to the period between 1^{St} of March 2010 and 15^{Th} of May 2010 the results are in line with the other PV systems. This is due to a malfunction of the PV system. Further details one this particular behavior are discussed in next Section 6.2.4.

Table 6.5 Sommellier system performance indicators for our solution

Time-frame	LCE	MAD [%]	MBD [%]	r^2	$RMSD$ [%]	WIA
15 minutes	0.7	21.42	14.38	0.87	28.29	0.97
Daily	0.76	11.64	9.81	0.94	14.63	0.98
Yearly	-	1.6	1.6	-	4.79	-

6.2.3 Comparison with PERSIL simulation methodology

The results obtained for the simulation of *Sommellier* and *GalFer* PV systems are compared with the one obtained in the PERSIL project [8], with whom we partially share the same case study. Figure 6.5 and Figure 6.4 show the real measured daily energy production compared with results of our solution and with results of PERSIL methodology for July 2010. Both Figures point out that PERSIL overestimates daily

Table 6.6 GalFer System performance indicators for our solution

Time-frame	Interval	*LCE*	*MAD* [%]	*MBD* [%]	r^2	*RMSD* [%]	*WIA*
15 minutes	All days	0.55	30.86	24.25	0.76	39.07	0.95
Daily	All days	0.5	14.97	54.25	0.78	28.05	0.95
Yearly	All days	-	2.69	2.69	-	8.08	-
15 minutes	Before 15/05/2010	0.78	15.99	7.94	0.92	23.09	0.981
15 minutes	After 15/05/2010	0.47	36.1	30	0.68	43.93	0.93
Daily	Before 15/05/2010	0.8	9.09	7.94	0.94	11.92	0.98
Daily	After 15/05/2010	0.4	30.55	29.99	0.71	32.71	0.94

energy production with respect to our solution. In particular for the *Sommellier* system PERSIL methodology has a $RMSD = 22.92\%$ and $MAD = 21.77\%$ with respect to the proposed solution. For the *GalFer* system both methodology tend to overestimate daily energy production. The overestimation is related to a malfunction in the *GalFer* system occurred after the 15th of May 2010 as described in Section 6.2.4. By the way for the *GalFer* system our solution performs better as show in Figure 6.4. The PERSIL methodology in this case has a a $RMSD = 23.31\%$ and $MAD = 22.15\%$ with respect to our solution for the moth of July 2010.

This is also highlighted and quantified by comparing the accuracy of performance indicators obtained by PERSIL (Table 6.7) with the one obtained with the proposed solution (Table 6.5 and 6.6). This analysis underlines how the proposed software infrastructure better describes the energy production with respect to PERSIL. Regarding the estimation of daily energy production for *Sommelier*, the solution reduces *RMSD* of 32%, *MBD* of 28.88% and *MAD* of 24.34%. Considering the yearly energy production for the same PV system, the solution decreases *RMSD* of 35.66%, both *MAD* and *MBD* of 35.79%. About the estimation of daily energy production for *GalFer* before the malfunction, the solution reduces *RMSD* of 34.30%, *MAD* of 29.44% and *MBD* of 25.72%. Considering the yearly energy production for *GalFer*, the solution increases the accuracy by reducing the *RMSD* of 42.79%, both *MAD* and *MBD* of 42.76%. With the proposed solution, the accuracy increases because the simulation process takes as inputs an high resolution DSM that allows the possibility of simulating possible shadow. In addition, the proposed infrastructure consider $T_{sol-air}$ to better estimate the temperature for the PV cells. Finally, the daily energy production is computed as the sum of 15 minutes energy simulations. Thus, the effects of temperature on the PV efficiency is evaluated with real and actual values; while PERSIL exploits daily mean value.

Table 6.7 PERSIL Performance Indicator for Sommellier and GalFer PV systems

		LCE	*MAD* [%]	*MBD* [%]	r^2	*RMSD* [%]	*WIA*
GalFer	Daily	-0.16	58.22	56.66	-0.22	66.26	0.8
	Daily (before 15/05/2010)	0.31	30.87	30.87	0.54	34.75	0.91
	Daily (after 15/05/2010)	-32	67.85	65.74	-0.55	76.01	0.78
	Yearly	-	6.29	6.29	-	18.88	-
Sommellier	Daily	0.183	40.3	40.3	0.386	45.6	0.88
	Yearly	-	4.47	4.47	-	13.43	-

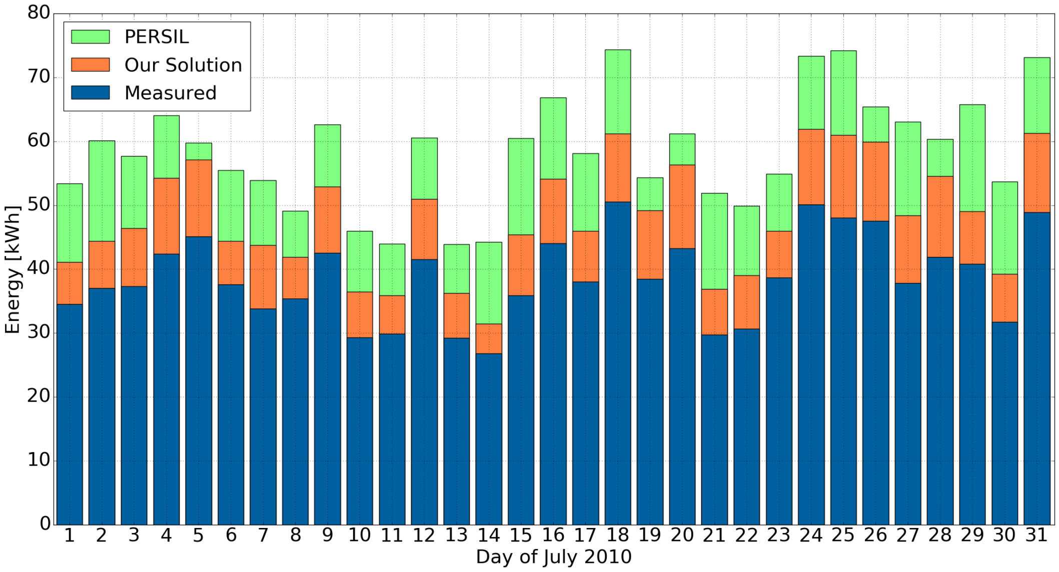


Fig. 6.4 Comparison of daily energy production of our simulation with measured data and PERSIL [8] (Ferraris PV system)

6.2.4 PV system operation assessment

The proposed distributed software infrastructure can help *Energy managers* and *PV system engineers* in monitoring the performance of already deployed PV systems. This can help in planning maintenance activities by identifying systems that are not working properly. During the tests on *GalFer* and after a deep analysis of its generation loads, it was possible to identify a malfunction of the system that affects its performance after the 15$^{\text{Th}}$ of May 2010.

Indeed, this anomaly decreases the efficiency of the modules reducing the maximum peak power. This is pointed out by analysing the peak power in a sunny day before and after 15$^{\text{Th}}$ of May 2010. Until 15$^{\text{Th}}$ of May 2010, the simulation trends follow the real behaviour of the PV system. In particular, the peak power in 17$^{\text{Th}}$

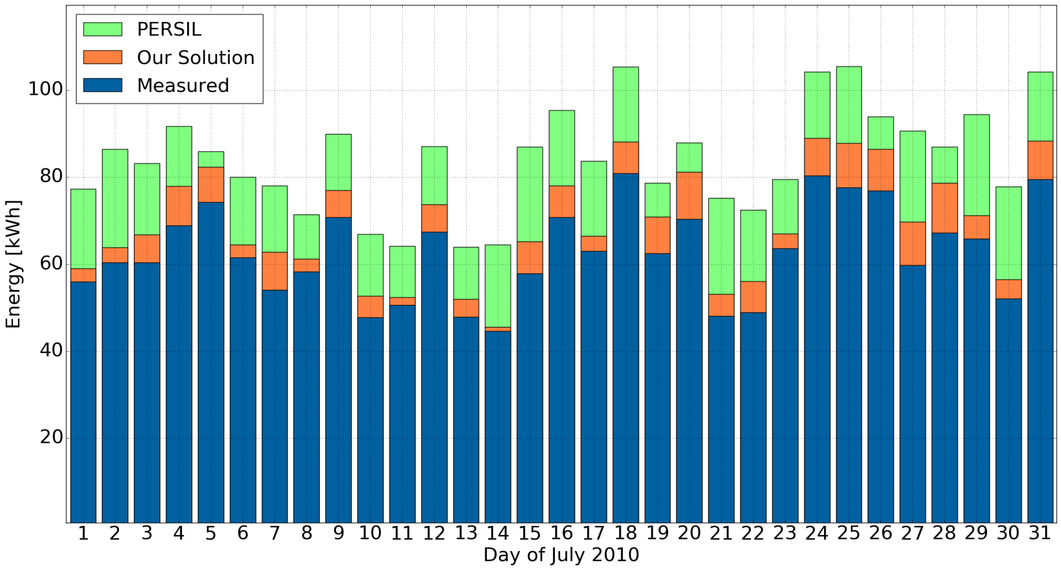


Fig. 6.5 Comparison of daily energy production of our simulation with measured data and PERSIL [8] (Sommelier PV system)

of May 2010 is 6354.77 $[W]$, with a global horizontal radiation of 844.10 $[W/m2]$ in sunny day. Instead, in 20^{Th} of April 2010, still a sunny day, the peak power is 7545.36 $[W]$ with a global horizontal radiation of 810.5 $[W/m2]$. This anomaly is also highlighted in Figure 6.6, where simulation results are compared with real measured values in sunny, cloudy and rainy days before and after 15^{Th} of May 2010. It is worth highlighting that this anomaly is not verified for *Sommellier*, which has almost the same system characteristics and geographic location of *GalFer* as presented in the case study. Indeed, for the *Sommellier* the peak power of the two days are 10027.05 $[W]$ and 9937.60 $[W]$ respectively.

6.2.5 Floor Planning

6.2.5.1 Experimental Setup

The algorithm for PV system Floor Planning has been applied on roofs of three industrial buildings, shown in Figure 6.7-(a). They are lean-to roofs of approximately 49m x 12m, facing S/S-W with inclination of 26°. The colored areas of Figure 6.7-(a) highlight the identified suitable areas. The figure highlights that some parts of the roofs are discarded, due to the presence of encumbrances - this is especially evident

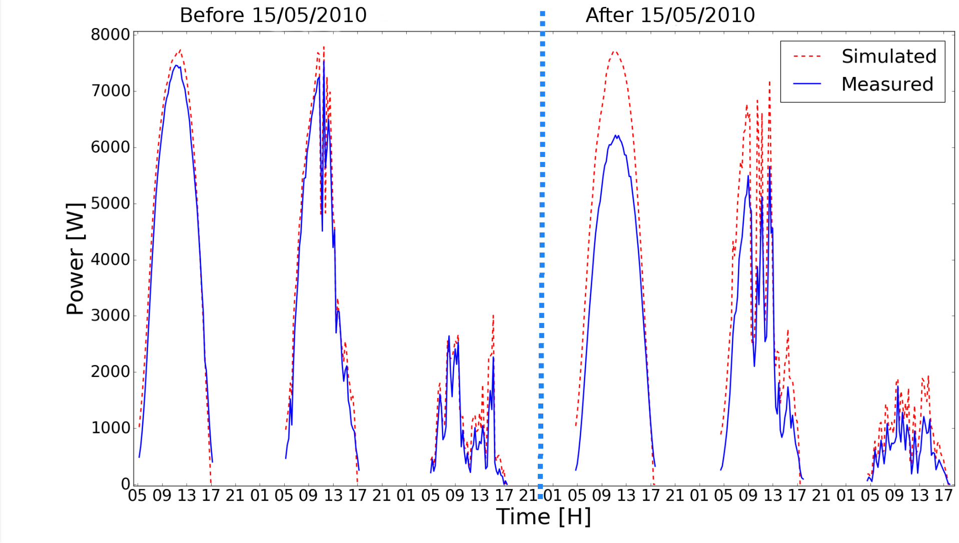


Fig. 6.6 Comparison of GalFer generation loads for Sunny, Cloud and Rain day before and after 15^{Th} of May 2010

for roof 1, where pipes occupy a large space. The suitable area is then aligned to the 20cm grid. The key feature of the roofs are reported in Table 6.8.

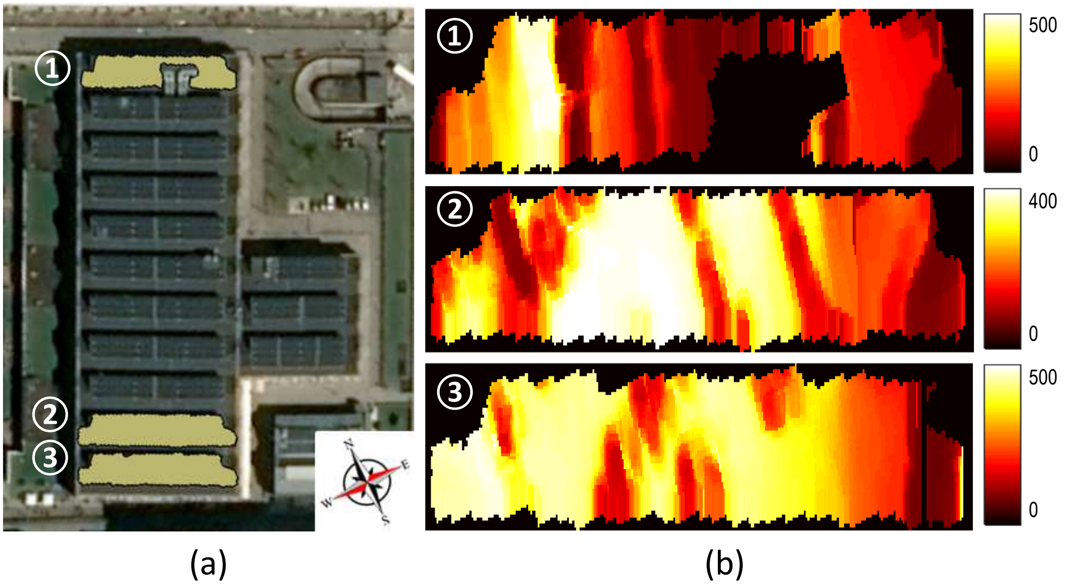


Fig. 6.7 Roofs used for the experimental analysis (a), and corresponding irradiance distributions (b).

The solar radiation data coming from the *Real-sky calculation service* has then been used to derive the evolution of irradiance and temperature over time for the roofs. Despite of the geographical proximity, the roofs have quite different irradiance

Table 6.8 Characteristics of each roof, and power production of the proposed PV floorplanning algorithm with respect to traditional placements.

Roof	WxL	N_g	N	Traditional	Proposed	
				MWh	MWh	%
Roof 1	287x51	9,416	16	3.430	4.094	+19.37
			32	6.729	7.499	+11.44
Roof 2	298x51	11,892	16	2.971	3.619	+21.85
			32	5.941	7.404	+23.63
Roof 3	298x52	11,672	16	2.957	3.642	+23.16
			32	5.746	7.405	+28.86

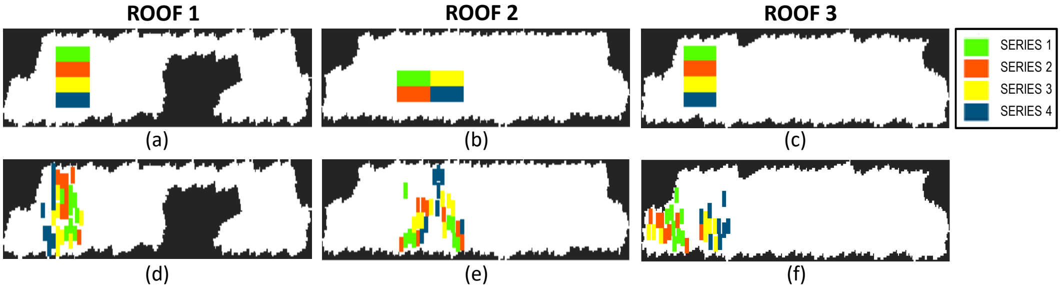


Fig. 6.8 Traditional PV panel placements (a-c) and placements resulting from the PV floorplanning algorithm (d-f). Colored rectangles represent panel positions, and panels of the same color are connected in series.

distributions. Figure 6.7-(b) shows the 75th percentile of irradiance distribution (brighter colors represent a larger irradiation). All roofs tend to have the least irradiated grid elements on their right-hand side, as an effect of roof orientation. In general, irradiance is quite non-uniform, and the variance is quite heterogeneous over the different roofs; this is especially evident for Roof 1, as the pipes tend to lower irradiance over a wide area of the roof.Notice that this map does not include the effect of temperature.

6.2.5.2 Simulation Results

The PV floorplanning algorithm is run twice on the three roofs to place $N = 16$ and $N = 32$ panels. The panels are always organized with series of 8 panels ($m = 8$). The execution time of the placement algorithm is proportional to the number of valid grid elements and to the number of panels to be placed, and required less than 120 s under all configurations on an Intel 8-core i7 server with 15.4GB of RAM. Due to the large number of grid elements (almost 12,000 for roofs 2 and 3), it is not possible to compare our results against an exhaustive algorithm.

Figure 6.8 compares the loose placements generated by our algorithm against traditional “compact” placements. Colored rectangles represent panels, with panels of the same color belonging to the same series string. Due to space constraints, the figure reports only the experiments run for $N = 32$.

The compact placements (a-c) are placed in the most irradiated area of the roof; notice that these placements are determined using accurate spatio-temporal irradiance information that are not normally available to installators. Therefore, we are comparing our solution to a particularly “good” reference.

The placements resulting from our floorplanning algorithm are shown in Figure 6.8-(d-f). They clearly tend to be placed nearby the traditional placements (e.g., compare (a) and (d)), yet the are sparser, since they try to exploit fine-grain differences in the distribution of irradiance and temperature. This is clearly visible for example in the triangular shape of the placement in (e), that matches the irradiance variation in that region of Roof 2 (Figure 6.7-(b)).

Table 6.8 clearly shows that our PV floorplanning can significantly improve the energy production on a yearly basis, with improvements that range from 11% to 28%. Obviously, the magnitude of the benefit is proportional to the available space; this explains the smaller improvements for Roof 1, which has fewer valid grid points than the other roofs.

It is possible also to notice how more irradiated roofs improve the benefit of a customized placement; Roof 1 has a sensibly smaller benefit from the placement than the other two, as a result of a clearly visible lower average irradiance (Figure 6.7-(b)).

The sensible improvements obtained by the placement, however, are not just due to a positioning of modules that matches irradiance and temperature. The placement is also *topology-aware*; by enumerating modules in series-first fashion, it guarantees that he bottleneck effect in a series string due to a “weak” module (which determines the current of the entire string) cannot occur. This effect is visible in Roof 1: the traditional and modified placements occupy more or less the same portion of the roof, and therefore they are subject to similar G and T conditions; however, the energy extracted in two placements differ by 11.4%, as mostly due by avoiding the “weak” module issue.

6.2.5.3 Overhead Assessment

For the calculation of the wiring overhead, an AWG 10 cable with resistive loss of $\approx 7m\Omega/m$, and a approximate cost of $1\$/m$ is assumed to be used. As a conservative calculation of the overhead, assume a 4A current in a series string (corresponding to an irradiance of $600W/cm^2$); the power would be $RI^2 \approx 0.11W/m$ for each meter of extra cable in the string, i.e., $\approx 0.5kW/m$ of energy in one year (assuming 50% of the time at zero current for dark periods). If is multiply this number for 8 strings, and comparing it to the figures of Table 6.8, the overhead is approximately only $0.05\%/m$. The wiring overhead is in the order of 20 meters for the worst-case solutions, so both power and cost overhead are not an issue.

The placement algorithm does not directly include power overhead, if not indirectly by restricting the greedy choices.

6.2.6 I-V Modelling

The model has been tested against real data obtained from an installed PV array on the roof of Sommelier and Galfer, described in the case study at the beginin of the Section 6.2. The array consists of the parallel connection of 4 strings in parallel, each consisting of 10 modules in series, with bypass diodes around each module. The MPPT with the inverter is placed after the parallel connection of the 4 strings, and therefore the MPP is extracted on the global curve of the entire array. The analysis covers one year, from March 2010 to February 2011, for which both environmental data and power production traces were available.

The resulting traces have been compared to the actual power extracted *before the inverter*, as returned by the measurements on the actual PV installation. Figure 6.9 provides a graphical comparison of the traces (from top to bottom, total power, voltage and current of the panel). For space constraints, the analysis is restricted to 12 days (from August 8th to August 21st). The plot highlights that the proposed model (solid lines) follows quite well the experimental measurements (dashed lines). The model slightly overestimates the current and the total power; this is likely due to a conservative assumption on the aging of the modules, which derived from the literature [107] as this information was not available for the PV installation.

In order to assess accuracy, the proposed model is compared with the results obtained with the NOCT method presented in Section 5.2.1 and with the one proposed in [8]. Table 6.9 reports four indicators for each model: (i) the Root Mean Square Difference (*RMSD*), (ii) the coefficient of determination R^2 (i.e., the proportion between the variance and the predicted variable), (iii) the Willmott's index of agreement (*WIA*, measure of prediction errors), and (iv) the Legate's coefficient of efficiency (*LCE*, ratio between the mean square error and the variance of observed data) [130]. The former two are percentages that measure dispersion, and thus the lowest the better. The latter two are indicators of performance, for which a higher value indicates a better model. Table 6.9 highlights that the proposed model outperforms both the NOCT forumla and the one proposed by [8] on all indicators.

The high level of accuracy is additionally confirmed by the analysis of the yearly power production. The estimated power production is 6.695MWh, very close to the *measured* production of the actual PV system (6.708MWh). The very low error rate (<0.2%) highlights that, despite of local fluctuations, the proposed model adheres to the actual PV system behavior. This is even more meaningful when considering that the proposed model requires as inputs only the datasheet of the adopted PV panels and the system topology. This high level of accuracy, together with the computation speed (26.6s for model construction and 259.7s for one year long simulation) prove the effectiveness of the proposed model in the context of autonomous building design.

Table 6.9 Statistical performance indicators of the proposed model w.r.t. NOCT Formula and [8].

MODEL	RMSD	R^2	WIA	LCE
Proposed	22.20%	0.906	0.975	0.751
NOCT Formula	28.29%	0.870	0.970	0.700
[8]	45.60%	0.386	0.880	0.183

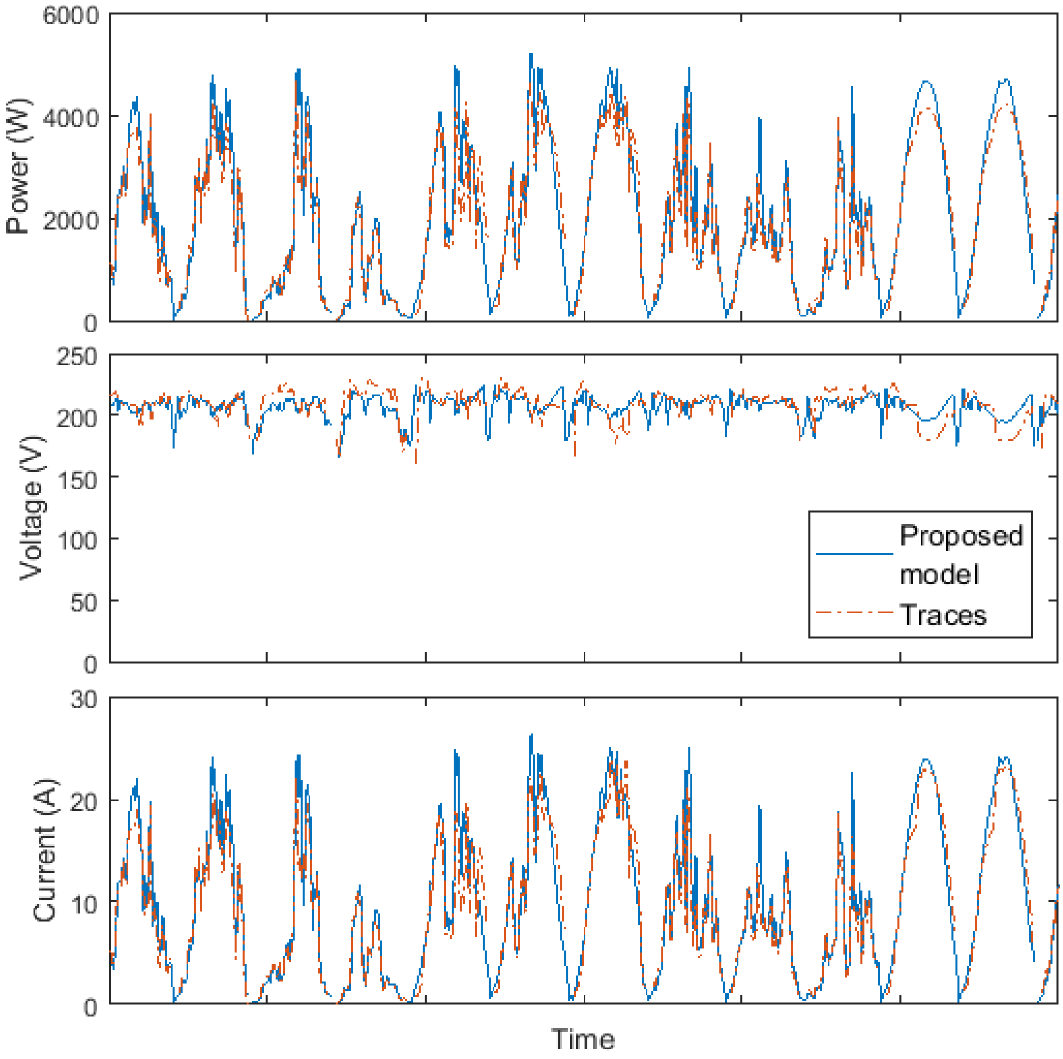


Fig. 6.9 Comparison of the proposed model (solid lines) w.r.t. the experimental traces (dashed lines).

6.3 Renewable energy and Smart policies grid integration

In this section the results obtained with the integration of the PV simulator with the real-time simulator presented in Section 5.2.2 are reported. In particular the integrated solution is used to assess and discuss the impacts of rooftop PV penetration and distributed storage on grid operation status from different perspectives. The advantages of this integration can be seen both in planning and operation phase.

In phase of planning and network reinforcement, the tool can be used to assess the impacts of different levels of PV penetration on the existing grid. The test conducted in the case study area will show that problems can involve transformers capacity in case of high penetration of PV in low voltage grids. As another example of planning, homogeneous distribution of PV generation with respect to the level of demand should be taken into account in order to evaluate the levels of self-consumption and self-sufficiency. In the planning phase neglecting the analysis of homogeneous distribution of PV generation with respect to the level of demand can introduce challenges in the network voltage control. Based on these studies, existing network can be refurbished through investment, or new regulation can be adjusted to meet the requirement of the system (e.g. when new regulations are being made to provide incentives to install PV arrays on the rooftop, different areas may get different incentives).

In the operation phase, the co-simulation tool proves how the PV simulator can provide in time and quite sufficient information about generation to support low voltage system state estimation [131]. Since there are too many low voltage connected PV panels in the grid, there are a lot of challenges in the estimation of generation data due to the need to install new suitable smart meters, acquiring a huge number of measurements, retrieving so much data so frequently (e.g. every 15 minutes), and fast data processing. The performed simulation will show how system operator can forecast and monitor substation power flow by using the data generated by the PV simulator.

The district of *La Crocetta* in Turin has been chose as case study for testing the co-simulation environment.

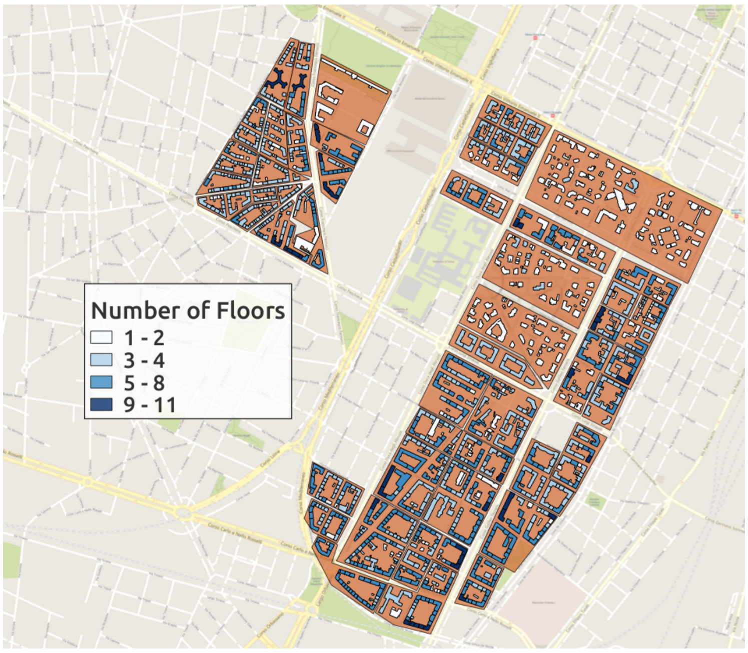


Fig. 6.10 Buildings in substations

This area counts 2198 residential buildings connected to 43 MV/LV substations. Each substation serves an area whose extension depends on the number of connected households, as reported in Fig. 6.10. To evaluate the integration of PV systems in the district, a summer sunny and a summer cloudy day have been simulated. During summer days, in Italy, the energy consumption of residential users is lower than in winter days [132] because residential households do not usually have air conditioning systems. On the other hand, during winter season, heating systems circulation pumps run almost all day long. In addition, sunny days in summer produce more electricity from PV systems and this can be a critical situation for distribution grids. Fig. 6.11 shows the daily energy consumption for each substation (consumption data were obtained through measurements in MV/LV substations). Energy consumption is not proportional to the area served by the sub-station, but rather to the number of households. Unfortunately, information about DNI and DHI are not available. Hence, the *Solar radiation decomposition* module (see Fig. 5.1) is used and the values of GHI radiation are retrieved, via web-services, from a weather station located in the center of the district. Finally, high quality mono-crystalline Si PV modules with efficiency $\eta_{pv} = 20.4\%$ in standard test conditions (STC) and thermal coefficient of maximum power $\gamma_{pv} = -0.38\%/^{\circ}C$ have been considered.

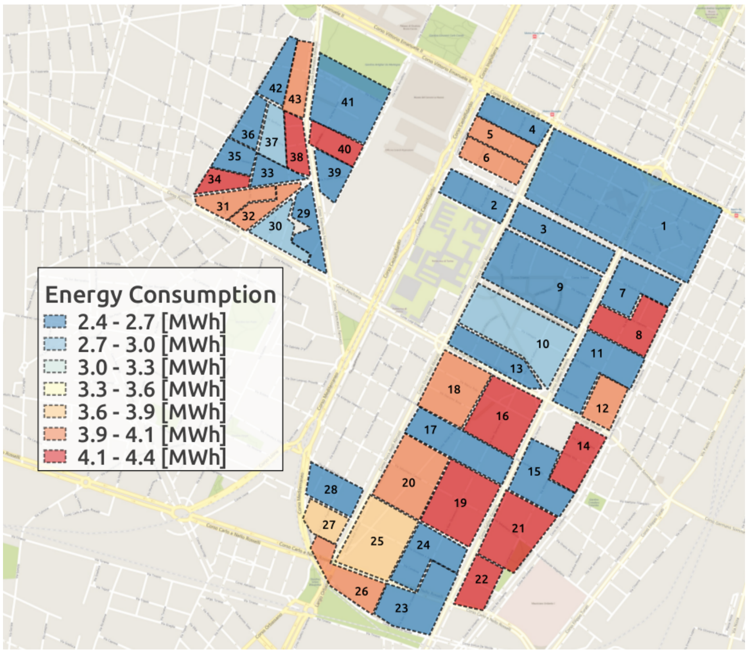
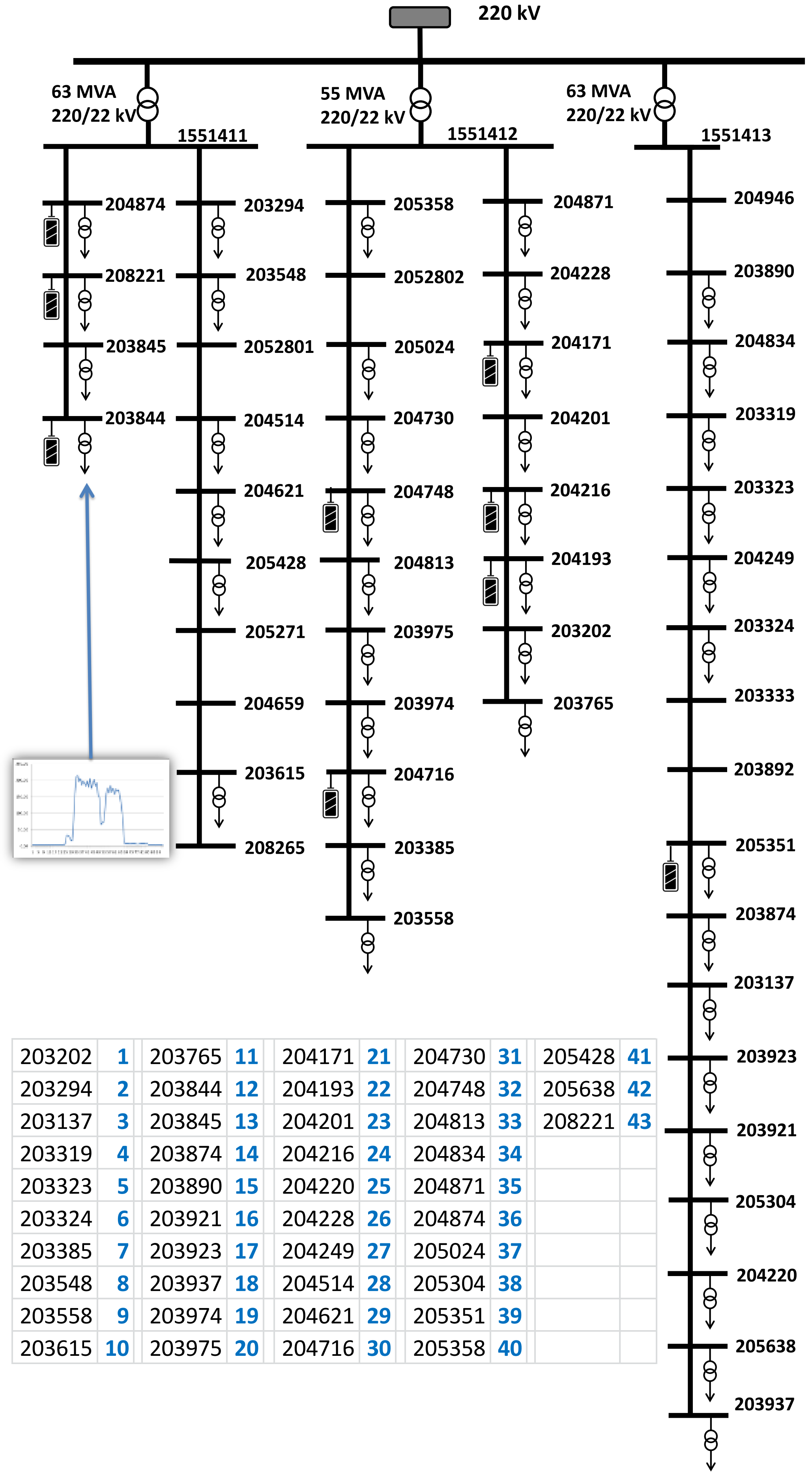


Fig. 6.11 Substations Electricity consumption

The electricity grid is a medium voltage (MV) network with five feeders derived from three 22kV busbars of a 220/22 kV primary substation. Each of the busbars is energized by a 220/22 kV transformer. The total length of MV lines, mostly constituted by underground cables, is around 39 km (Fig. 6.12). There are 49 MV/LV substations out of which 43 are supplying loads (mainly 2200 residential buildings). The MV/LV transformers are characterized by voltage ratio of 22 kV/400 V and a nominal power of 400 kVA, 250 kVA or 160 kVA.



203202	1	203765	11	204171	21	204730	31	205428	41
203294	2	203844	12	204193	22	204748	32	205638	42
203137	3	203845	13	204201	23	204813	33	208221	43
203319	4	203874	14	204216	24	204834	34		
203323	5	203890	15	204220	25	204871	35		
203324	6	203921	16	204228	26	204874	36		
203385	7	203923	17	204249	27	205024	37		
203548	8	203937	18	204514	28	205304	38		
203558	9	203974	19	204621	29	205351	39		
203615	10	203975	20	204716	30	205358	40		

Fig. 6.12 Grid topology

To evaluate the integration of PV systems and distributed storage in the district, a summer day scenario (in July) with a quite high PV production has been simulated. During summer days, in Italy, the energy consumption of residential users is lower than in winter days because residential households do not usually have air conditioning systems [132]. In addition, sunny days in summer produce more electricity from PV systems and this can be a critical situation for distribution grids.

The simulation process identified 944 areas in the building rooftops, equivalent to 71595.53 m^2, suitable for deploying PV systems with a maximum nominal power potential equal to 14.21 MW. During the sunny day, the peak power production is around 3.77 MW and the energy production is equal to 28.41 MWh. On the other hand, during the cloudy day, it is around 3.94 MW with an energy production of 16.95 MWh. The peak power production does not reach the nominal peak power in either of the two cases as shown in Figure 6.13. The higher peak power reached in the cloudy day can be explained through the phenomenon of irradiance spikes caused by broken clouds (ISBC) [133] or by a lower temperature of the PV arrays.

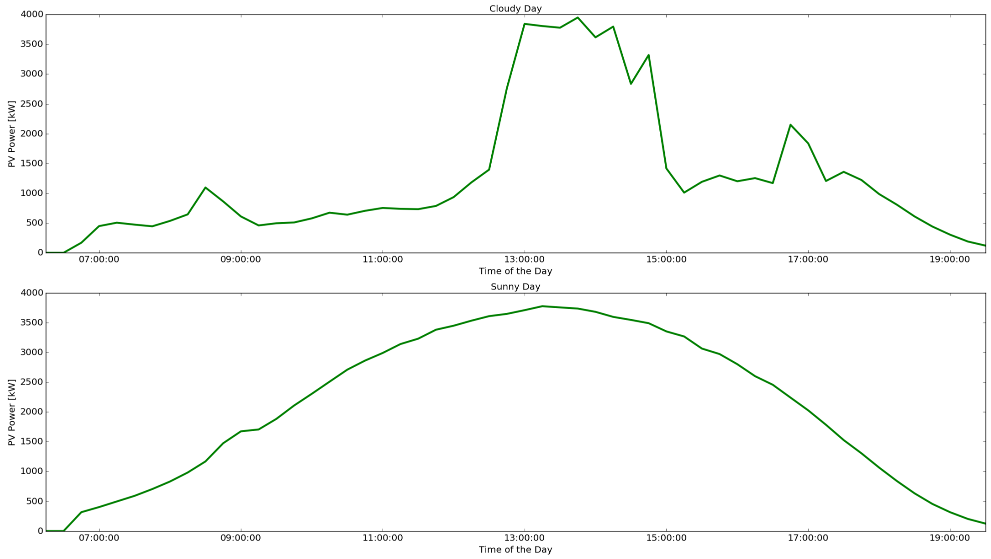


Fig. 6.13 Grid Total generation in a Cloudy and Sunny day

We discuss in the next paragrphs the results in the two above-mentioned phases (i.e. planning and operation) for Self-consumption and Self-sufficiency evaluation, voltage control, transformers capacity and substation power flow.

6.3.1 Self-consumption and Self-sufficiency evaluation

The distribution of power and energy production for each MV substation area during a sunny day in summer is shown in Fig. 6.14 and Fig. 6.15 respectively. As the simulation process considers also the shadows of surrounding buildings and vegetations, rooftops areas with high power production potential can have an energy production impact lower than areas with low power production potential (see Fig. 6.14 and Fig. 6.15). In 6.17 and 6.16 the levels of self-sufficiency and self-consumption are reported. 6.17 shows that the sub-station with less production has less than 10% of self sufficiency. However, sub-stations $17, 18, 23, 24$ and 25 have a level of self-sufficiency over 39%. Figure 6.16 revels that almost every sub-station absorbs all the produced energy from PV system. Only sub-stations $17, 18, 23, 24$ and 25 feed the MV distribution grid and are the one with the highest level of self-sufficiency.

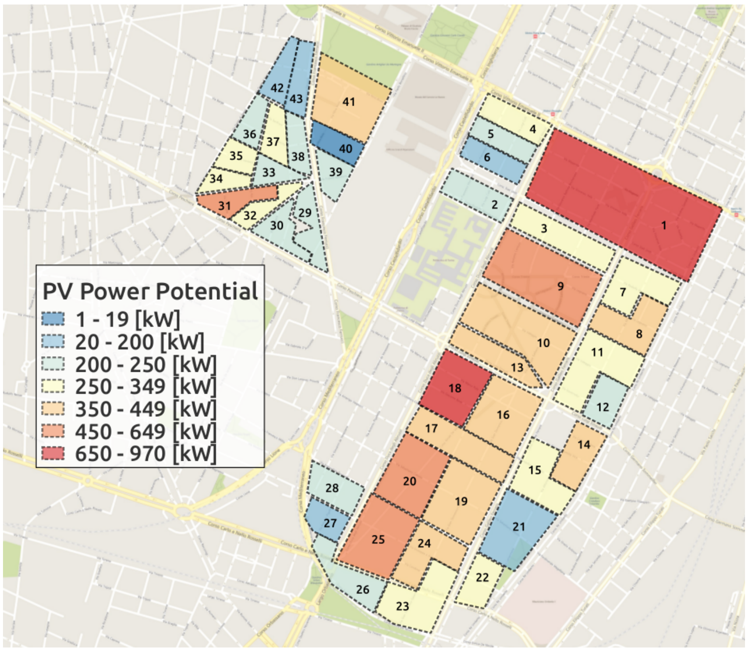


Fig. 6.14 PV potential for each substation

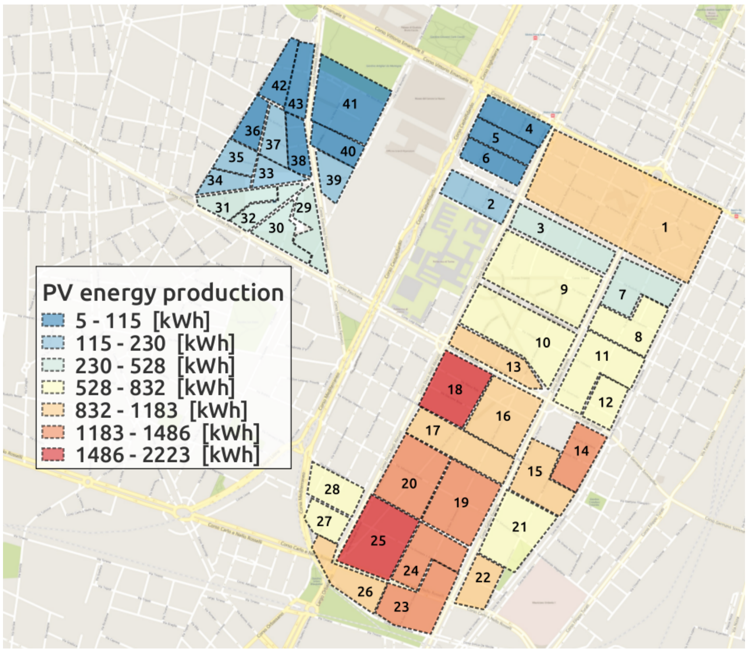


Fig. 6.15 PV production in a sunny day for each substation

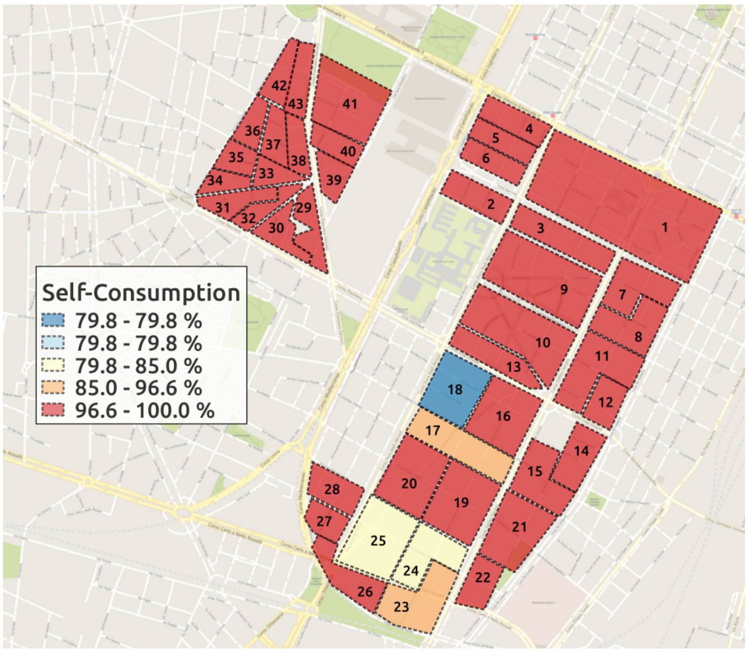


Fig. 6.16 Level of self consumption for each substation

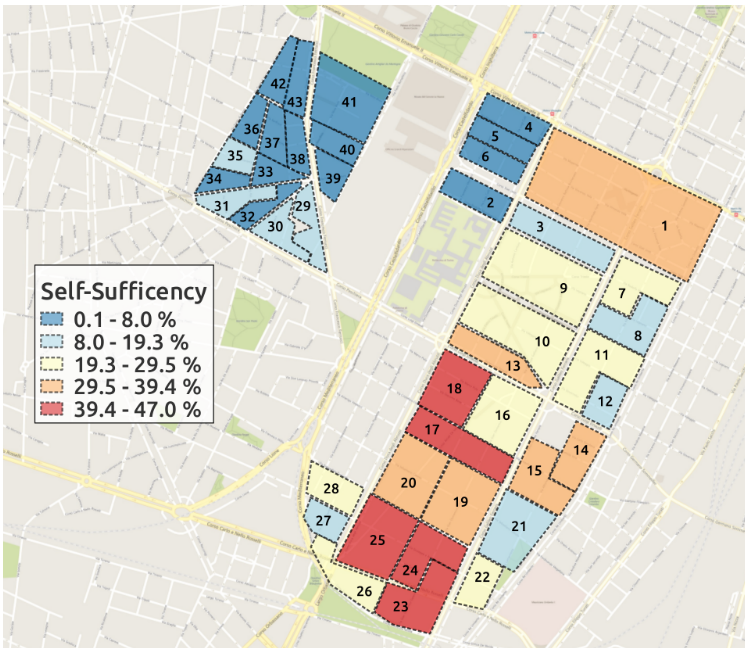


Fig. 6.17 Level of self-sufficiency for each substation

6.3.2 Voltage Profile

In distribution systems, tap changers of the HV/MV transformers at the primary substations would try to keep the voltage at the MV bus-bars at a certain level, by measuring and monitoring transformer current. When a transformer feeds several feeders, characterized by different PV penetrations with respect to peak loads, the voltage profiles at the secondary substations on the different feeders follow different profiles. This means that monitoring and regulating the voltage at the beginning of feeders is not necessarily sufficient for keeping voltages of all the substations of all the lines in the desired range. For example, in feeders where generation is higher, in some substations voltages may be above the admissible limit. In this case, the first feeder from the left is characterized by low demand, while its PV generated power is more or less the same as the others. In the second feeder, MV/LV transformers have greater sizes (400 kVA and 250 kVA), and consumption is higher than in the left one in which smaller transformers (160 kVA) are installed. We run the co-simulation for a summer scenario of a sunny day. Fig. 6.18 and Fig. 6.19 shows aggregated generation and consumption at substaions 7, 9, 21, and 24, and Fig. 6.20 shows the net active power injected into the downstream LV grids connected to these subsations.

Substations 21 and 24 with high self-sufficiency are connected to one feeder, and substations 7 and 9 with lower self-sufficiency are connected to the other feeder. According to the voltage profiles of these sub-stations (Fig. 6.21), any changes in the level of transformer voltage to correct over/under voltage in one feeder would result in more deviation in the other feeder.

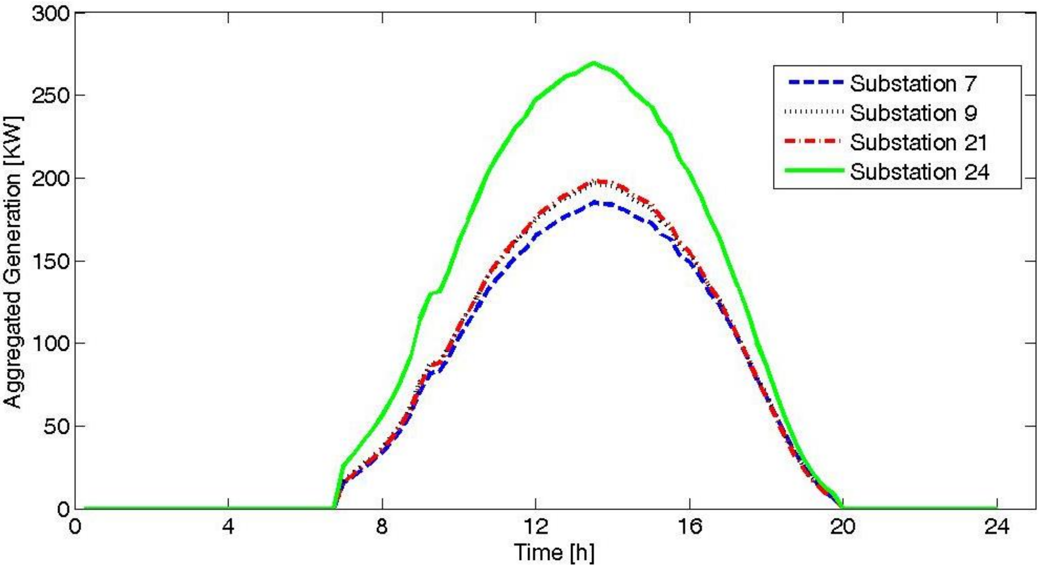


Fig. 6.18 PV Generation Profiles for substation 7, 9, 21, 34

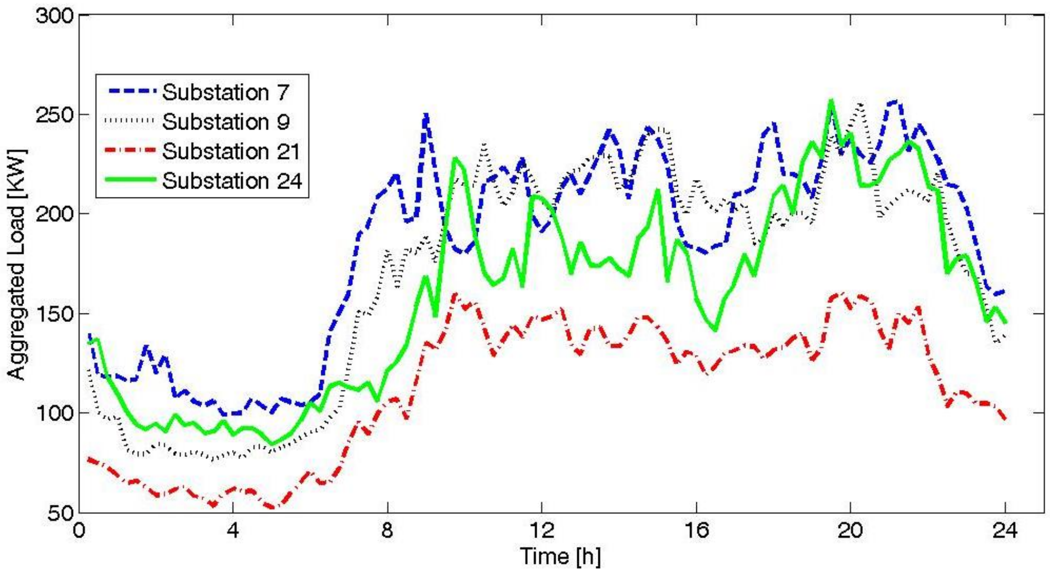


Fig. 6.19 Load Profile for substation 7, 9, 21, 34

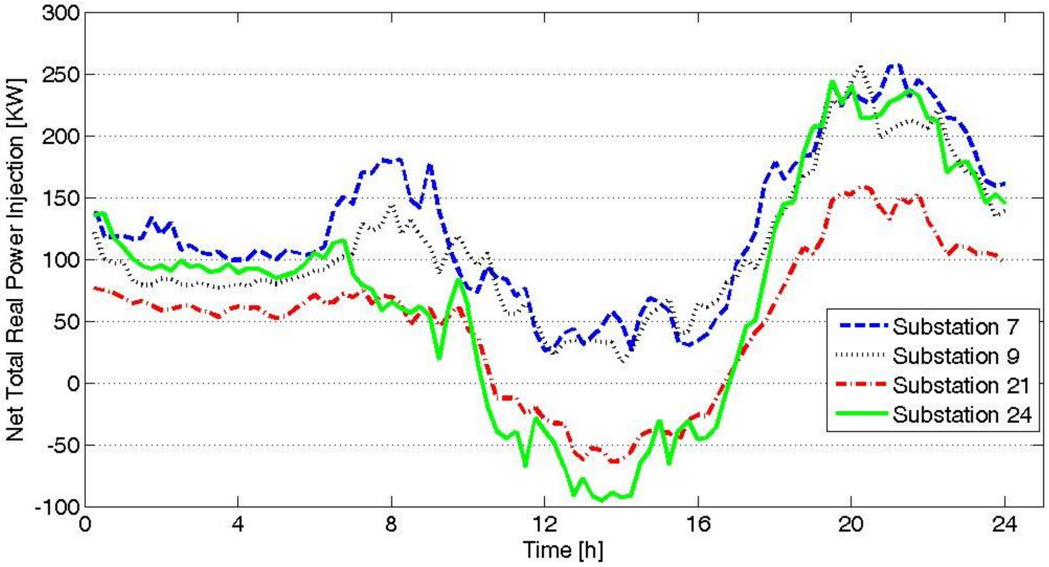


Fig. 6.20 Net consumption for substation 7, 9, 21, 34

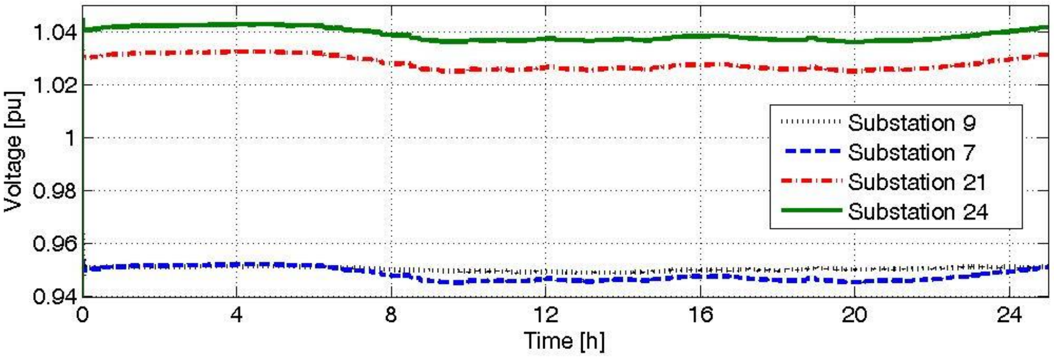


Fig. 6.21 Voltage profiles for substation 7, 9, 21, 34

6.3.3 Transformers capacity

The MV/LV transformers at the secondary substations are modelled based on the existing transformers in the real network, which were installed without considering the new PV generation capacity. Maximum aggregated generated power of each substation on the study day is shown in Fig. 6.22.

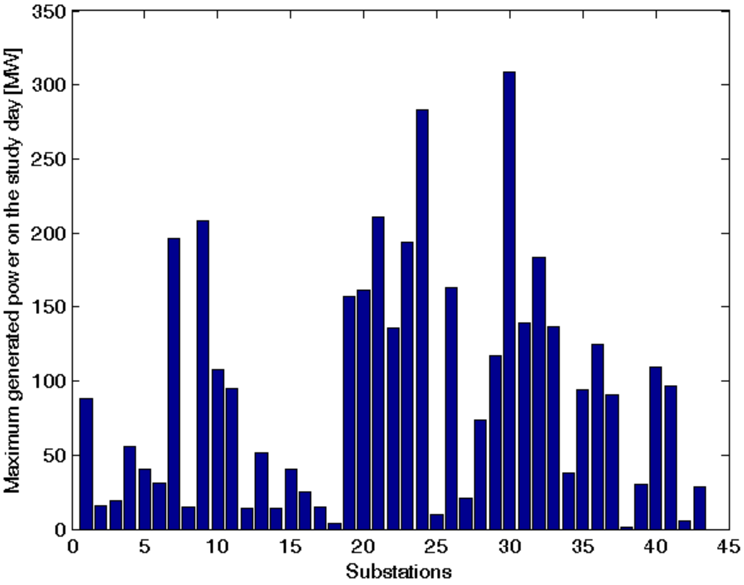


Fig. 6.22 Maximum PV generation of each MV/LV substation in a Sunny day

Transformer capacity is based on maximum apparent power in [kVA] and can be considered for both power absorption and injection (the red continuous lines in Fig. 6.23). All values of power generation and consumption are calculated for every 15 min of the study day, therefore there would be 96 snapshots of the systems status. To reach the worst scenario, the maximum net consumption (subtracting local generation from local consumption) of each substation during the study day are considered, indicated with large green bars in Fig. 6.23. As shown with narrow dark bars, the maximum load consumption of all substations is within the transformer capacity range, while integrating PV generation would cause violations in 2 substations (24 and 30 in Fig. 6.23). The maximum net consumption in these 2 substations exceeds the transformers capacity due to high amount of PV generation and low consumption. This highlights the fact that in cases where local generation is much more than local demand, either the installation of PV arrays should be reduced or grid infrastructures in terms of transformers (and also cables/lines) should be enhanced to tolerate reverse power injection from substations to the grid.

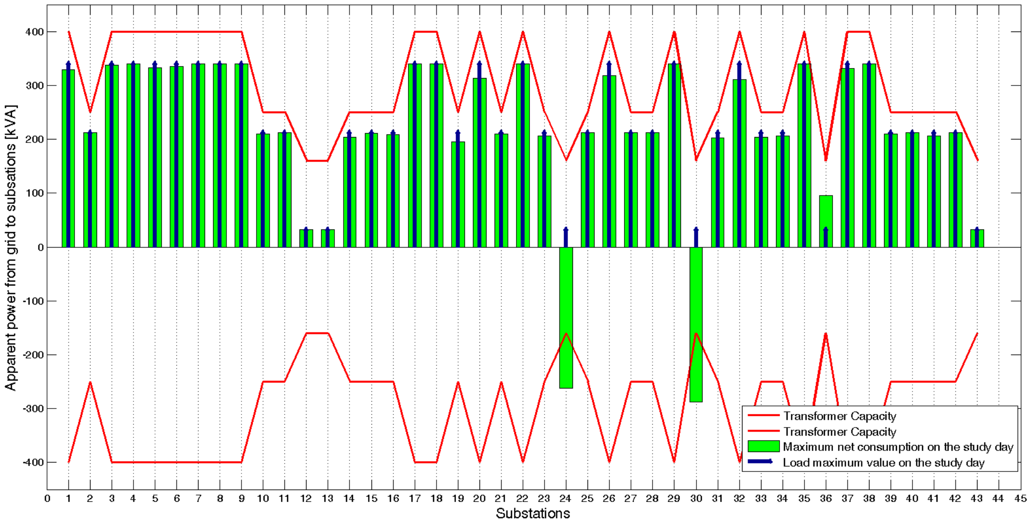


Fig. 6.23 Network results

6.3.4 Power Flow

In this case study, a scenario in which almost all residential buildings install PV arrays on their rooftops is analyzed, which reach their peak power on a sunny day of

July. The PV generation module does not consider grid constraints and introduces PV generation with the highest possible penetration, considering the available surface areas on the rooftops. In this case, penetration level is 51%, where PV penetration level is defined as total peak PV generation divided by total peak load apparent power. Total PV generation profile in this area in a cloudy day during July is shown in Fig. 6.24 with a green solid curve. An effect of high PV penetration on such a cloudy day can be observed when generation profiles change dramatically within a few seconds. If a huge generation drop occurs exactly in the ascending period of consumption profile, in high PV penetration scenarios, a large power deficit will be experienced. The blue dashed curve shows the result of power flow at the primary substation in terms of total active power injected to the distribution system; the steep descents means a rapid demand reduction in the distribution system (1.7 MW), and the steep rises imply a fast demand increase in the distribution system. Thanks to the PV simulator, Distribution Management System (DMS) can perform some analysis in advance (e.g. 15 minutes earlier) to prevent these steeps, for example by demand side management through flexible loads, or PV curtailment.

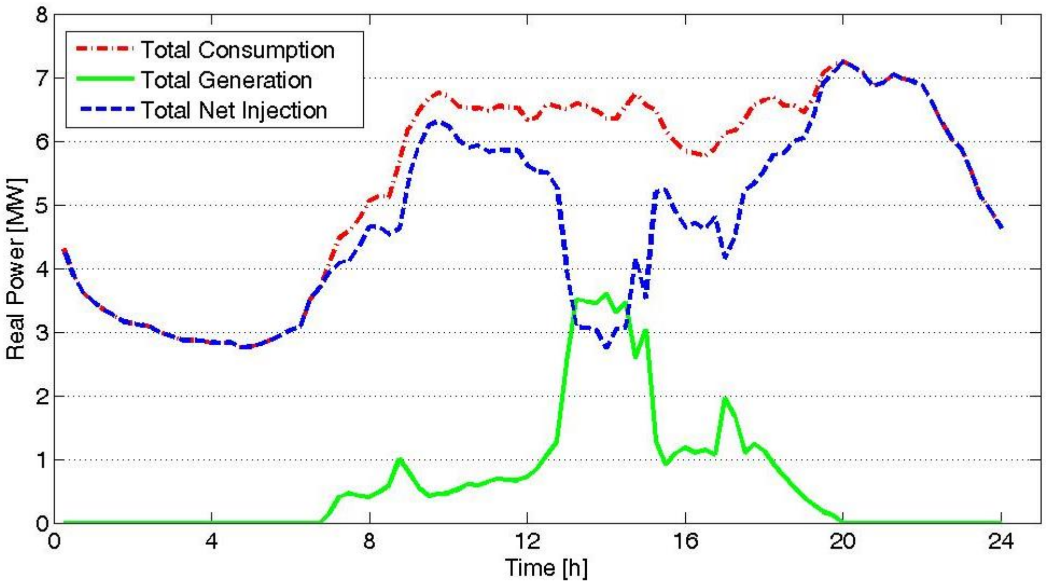


Fig. 6.24 Total load, generation, and net real power

The added-value of this integrated framework is the possibility of concurrently taking into accounts both real-like PV generation behavior from one side, and grid behavior and constraints on the other side, highlighting the fact that it can support DMS during operation mode.

6.3.5 Distributed Battery Management

In this paragraph the usefulness and functionality of the proposed distributed infrastructure for developing and testing distributed battery management when PV panels are installed on rooftops in the whole district area is shown. In this case, choosing storage batteries with the right capacity, location, and charge/discharge strategy in not only beneficial but also a need. The added storage is constituted, at each substation, by lead-acid batteries with a total capacity of 3000 Ah. The maximum power that each storage can exchange with the grid is 50 kW, working between the limits of 10% and 100% SOC.

In 9 substations, the combination of daily load and generation profiles provides the possibility of that the net consumption in some hours become negative. Batteries are added on the LV side of the transformers in these 9 substations (positions are shown in Figure 6.12 with aggregated batteries) and a control strategy to manage their charge/discharge profiles as been tested.

The local controller reads the total amount of active power exchange in MV/LV transformers and calculates the required amount of active power to be injected or absorbed by batteries connected on the LV side of the secondary substation. The transformer power measurement is periodically (every 15 minutes) sent to the battery controller module through MQTT protocol in the co-simulation infrastructure, and the battery power absorption or injection amount (i.e. charge or discharge respectively) is periodically pushed into the battery model inside the grid again through MQTT. In the grid simulator, batteries are modelled as dynamic loads with either positive (consumption mode) or negative (generation mode) power.

The modelled batteries are distributed stationary bulk storages connected to the LV side of the secondary substations, however the same platform can be used to simulate V2G by only changing battery models and battery control strategies.

The control strategy is to compensate the power difference between the substation net consumption and zero as the reference value of an ideal micro-grid. Nominal power of the batteries, nominal capacity, efficiency and minimum-maximum state-of-charge (SOC) of the batteries are taken into account to calculate the right amount of charge/discharge power.

In order to create a soft transition between the battery on/off states, when the batteries reach the minimum and maximum SOC limits, the controller creates a ramp to proportionally decrease rate of charge or discharge. In this way the occurrence of a high step on the total net consumption of the substation is prevented.

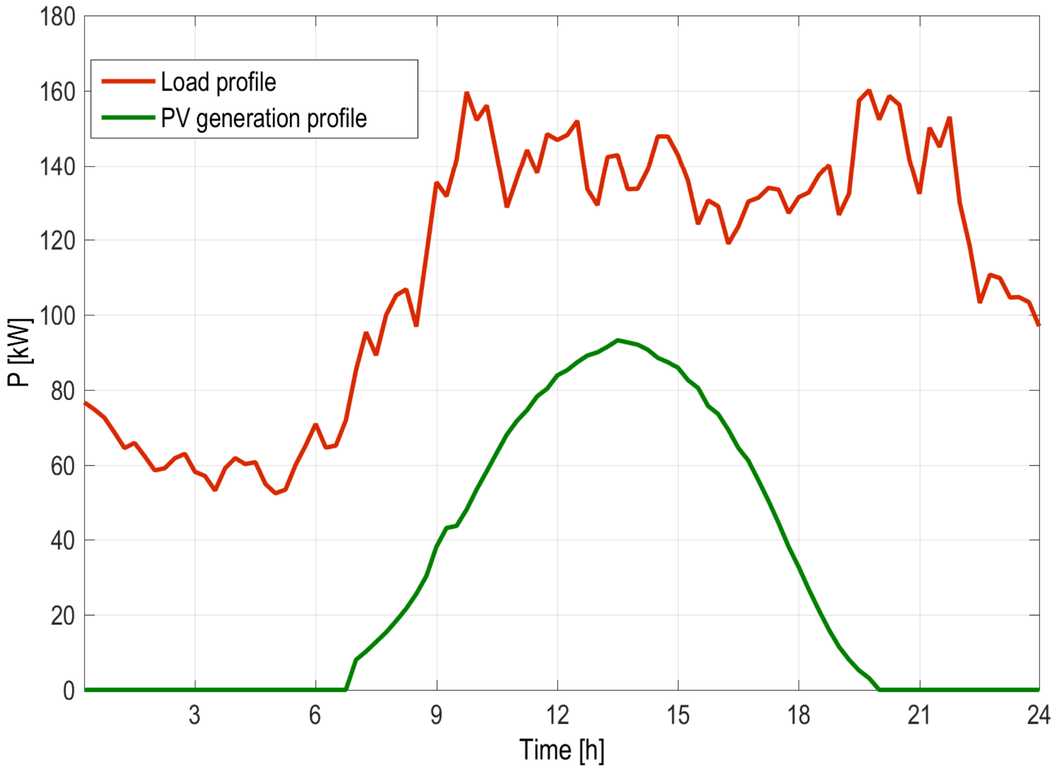


Fig. 6.25 Aggregated load and generation profiles in substation 204171 before battery installation

Based on the control strategy explained the battery behaviour in terms of power exchange with the grid can be captured as a time-variant profile. For the same substation presented in Figure 6.25, in Figure 6.26 and Figure 6.27 the net consumption with and without storage and the storage SOC respectively are presented. In the simulation the storage initial SOC is 50%. It initially discharges, as the net consumption in the substation is positive. When generation becomes bigger then load, the storage system charges trying to follow exactly the net consumption. At a certain point the residual power becomes negative as the maximum power of the storage is exceeded and because the SOC reaches 100%. The smooth charging/discharging profile produced by the strategy under test can be observed in Figure 6.27. The areas between the two curves in Figure 6.26 represent the energy exchanged between storage and grid during the charge/discharge cycles: as batteries efficiency is kept into account in the model, discharge areas are smaller.

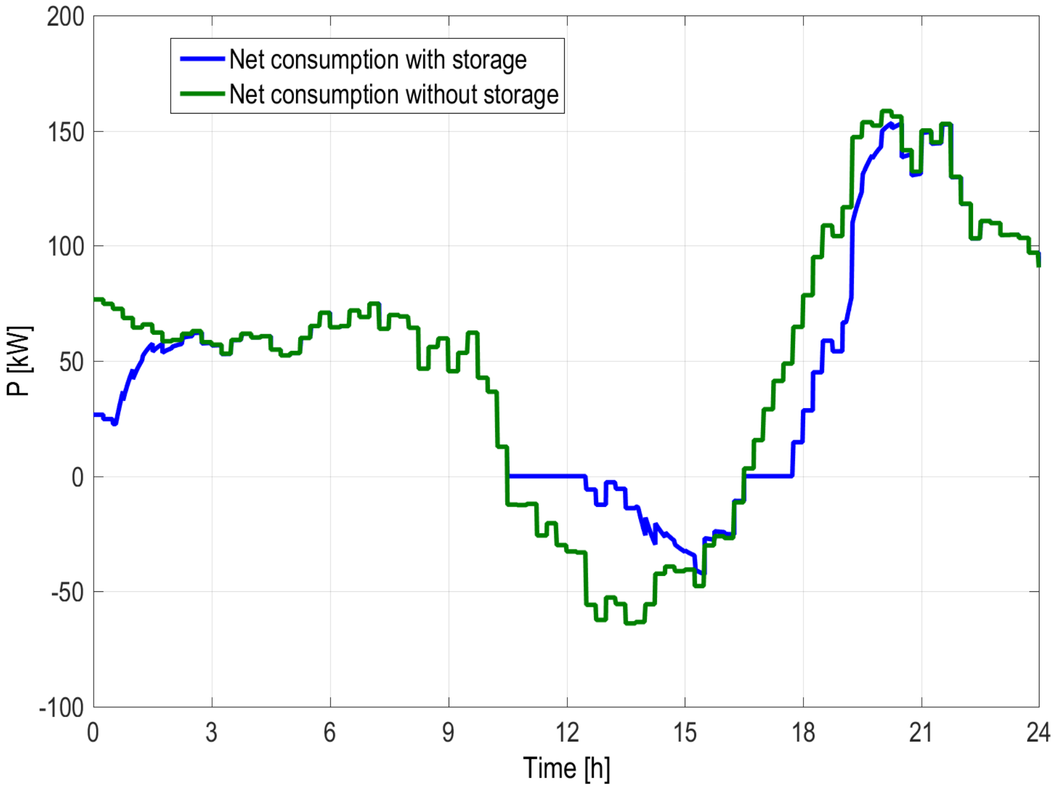


Fig. 6.26 Net consumption power with and without storage in substation 204171

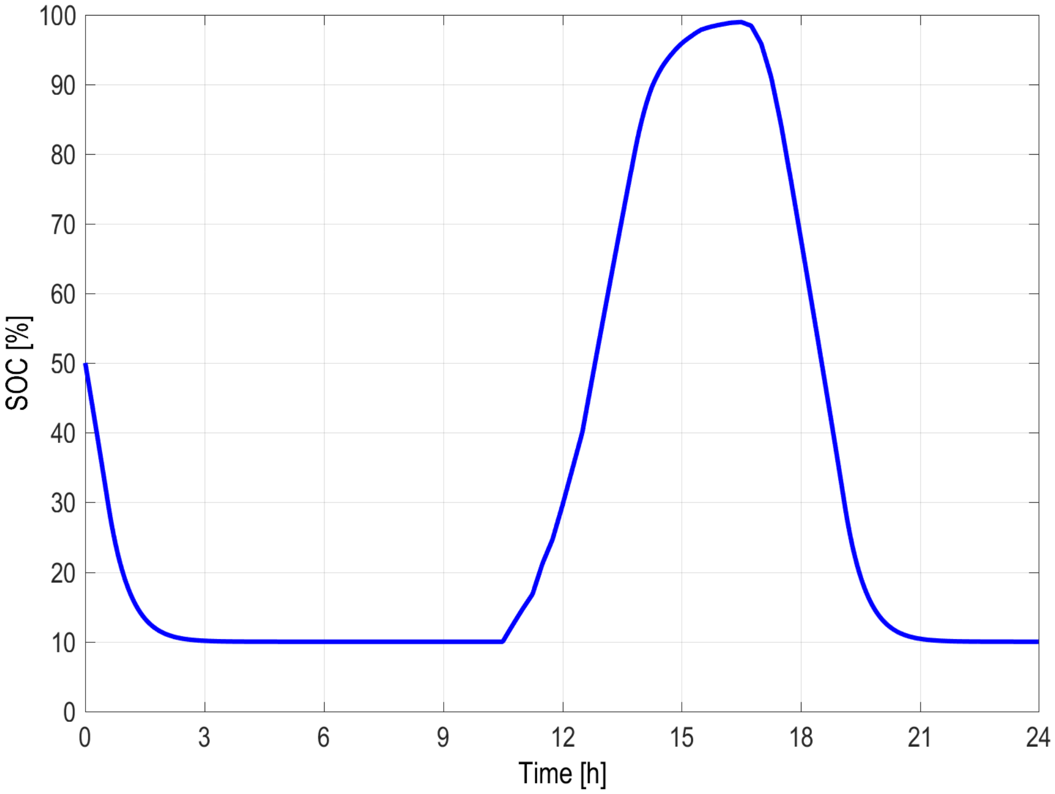


Fig. 6.27 State Of Charge profile of storage in substation 204171

Similarly to what is presented in Figure 6.25, Figure 6.26 and Figure 6.27, in all the substations where storage is integrated, the net consumption curves are smoothed: during the night, when the substation net consumption is positive, batteries discharge until their SOC reaches the minimum level. The storages start charging again when there is energy surplus in the substation or, in other words, the net consumption becomes negative.

Thanks to the network model running in the real-time simulator, it is possible to analyse the impact of the storage control strategy not only on the single MV/LV substations, but also on the overall distribution grid. The profile of the net demand of the primary substation is shown in Figure 6.28.

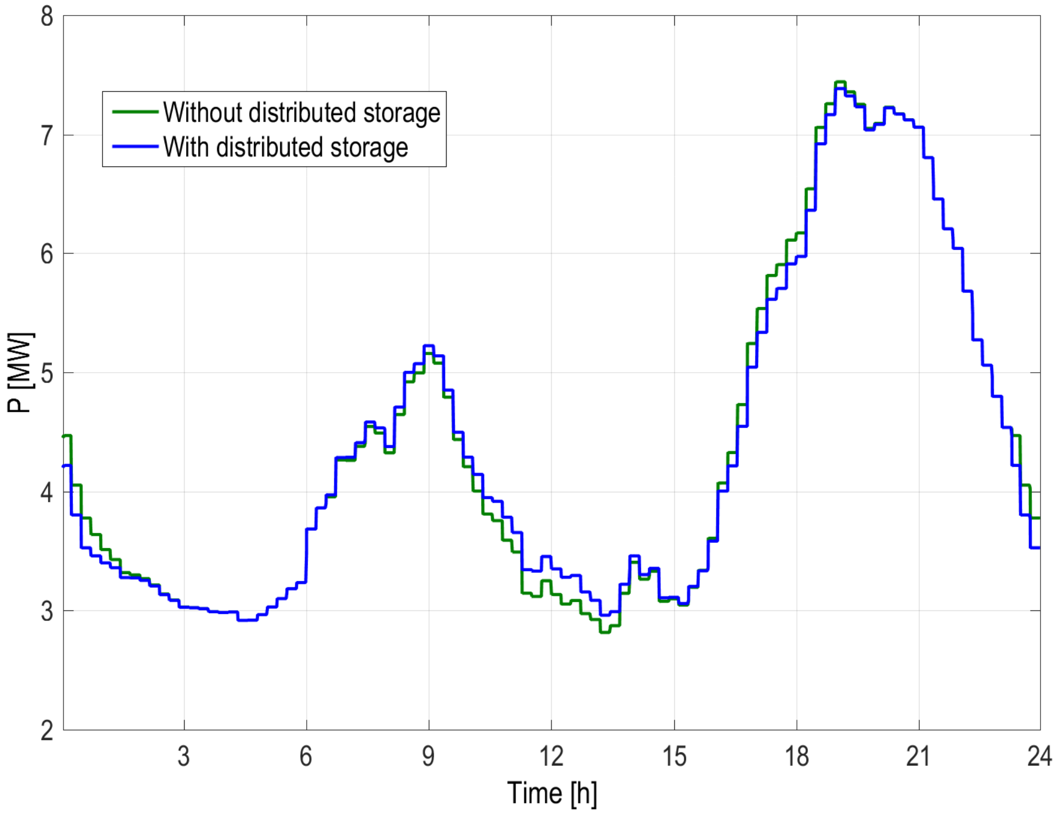


Fig. 6.28 Net demand profile of primary substation with and without storage

6.4 Power Prediction and building efficiency characterization

As a case study, energy-related data collected in a real world system in Turin are analyzed, where more than 50% of buildings are served by the heating distribution network. To monitor thermal energy consumption, gateway boxes have been installed in the monitored buildings. Each gateway includes a GPRS modem with an embedded programmable ARM CPU. An ad-hoc software has been developed to execute different activities: sensor management, GPRS communication, remote software update, data collection scheduling and collected data sending to a remote server.

Each gateway is responsible for the management of all the sensors deployed in its building. Thermal energy consumption is measured under different aspects, such as instantaneous power, cumulative energy consumption, water flow and corresponding temperatures. Furthermore, gateways also collect indoor and outdoor temperatures and the status of the heating system.

A cloud architecture is used for storing and processing all the monitored data. There are about 4,000 monitored buildings, each generating about 2,000 data frames per day. Thus, a growing base of at least 8 million data frames per day needs to be managed and analyzed. The gateways send the data frame to the cloud architecture, where a firewall first authenticates the data sender and then assigns each data frame to one of four dispatchers to guarantee the system reliability. Each dispatcher delivers the frame to a cluster of computers including different processing servers where data are stored in an HDFS distributed file system. The dispatcher is able to recognize if the process server has stored the frame correctly and in that case it sends the ACK to the gateway which can send the next data frame.

The meteorological data are collected from the Weather Underground web service [111]. Data from three different weather stations are collected to estimate the weather conditions nearby each building.

In our study we thoroughly evaluated PPBEC by considering a small cluster of 12 buildings with different Heating Cycle, (see Section 5.2.3.3.6): (i) 5 buildings with a Single Heating Cycle, (ii) 2 buildings with a Double Heating Cycle and (iii) 5 buildings with a Triple Heating Cycle.

6.4.1 Energy signature analysis and classification

To evaluate the quality of the linear regression that estimates K_{tot}, the R-squared value presented in Section 6.1 is exploited.

Figures 6.29 and 6.30 show the energy signature of a random building in the Turin area. The chosen t_{window} values are 24 hours and 7 days, hence the analysis considers the daily mean power values per unit of volume with respect to the daily mean outdoor temperature[1]. The analysis has been performed by considering as t_{period} the latest full Italian heating season (at the time of writing), from October 15^{th},

[1] In most residential buildings, the indoor temperature is not monitored through a sensor network, hence in the analysis a fixed value of $20°C$ is considered, since it is the typical value set by local regulations. Being fixed T_{in}, the charts report T_{ex} only.

2013 to April 14th, 2014. To focus the analysis on the steady state, t_{slot} has been set to the time range from 5 to 9 pm.

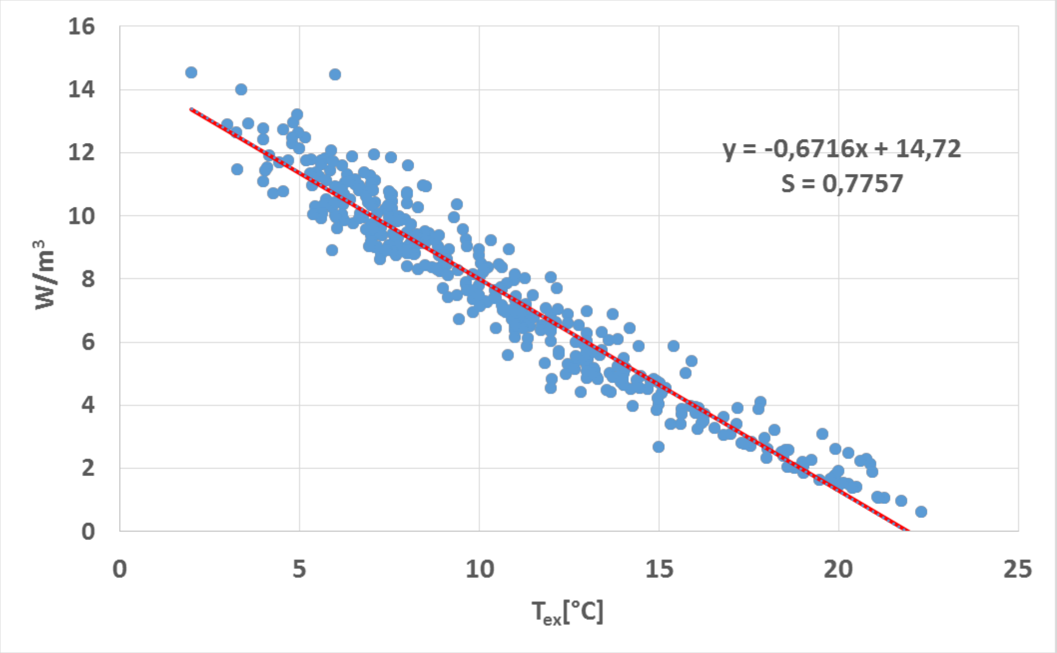


Fig. 6.29 Residential building, scatter plot of daily power consumption per unit of volume (W/m^3) with respect to T_{ex} ($°C$).

Figure 6.29 focuses on the daily t_{window} scatter plot and its resulting regression (red line) to estimate K_{tot}. A low S value of 0.78 is obtained, whereas the estimated value of K_{tot} is 0.67, which is a good result in terms of energy performance, as shown see later in the experiments.

Figure 6.30 shows the linear regression by aggregating mean power values over daily t_{window} for the considered building (dotted line). Figure 6.30 also shows the energy efficiency of the considered building (dotted line) with respect to the energy signature of (i) the most efficient building (dashed line) and (ii) the average power profile (solid line), considering all buildings in the corresponding district. Such comparative information allows to rank the buildings within districts, immediately putting in perspective the initial value of K_{tot}=0.67: even if it is a generally good value in terms of energy efficiency, as shown later, being better than its district average, the best performing building in the same district is far better. In such a situation, an end-user can consider to adopt energy-aware structural improvements to reach and improve over the best performing building.

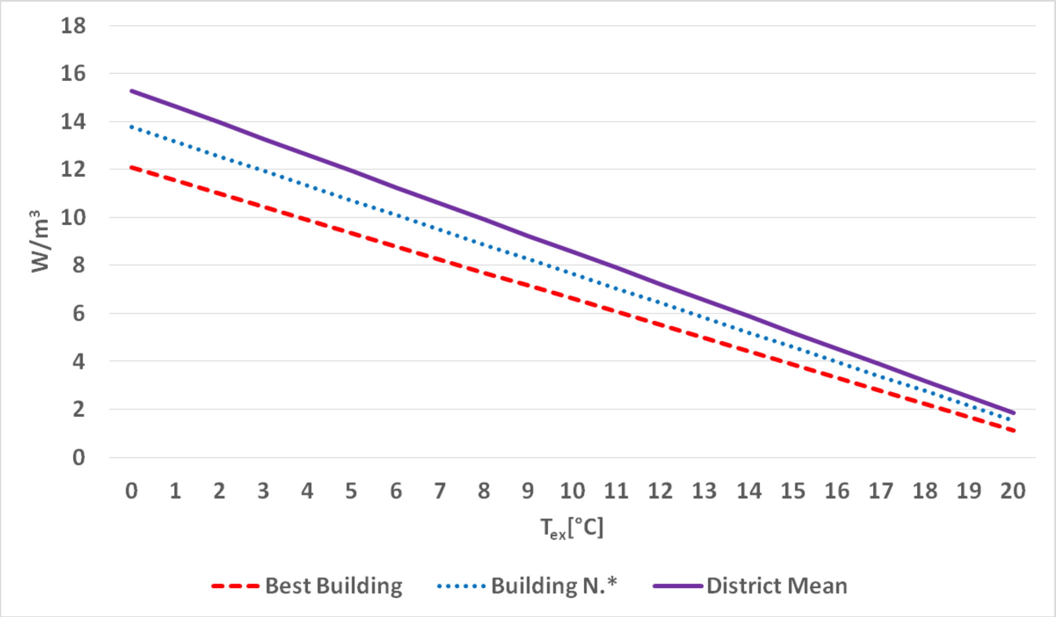


Fig. 6.30 Residential building, linear regression of daily power consumption per unit of volume (W/m^3) with respect to T_{ex} ($°C$).

Figure 6.31 shows the daily power consumption per unit of volume of a Turin school building, where a sensor network has been deployed for real-time monitoring of indoor temperature. The analysis has been performed by considering t_{period} from the beginning of the heating season 2013-2014 in Turin, on October 15th, 2013, to the latest data available at the time of writing, February 28th, 2015. Since many indoor temperature sensors are deployed in different rooms of the school building, the instantaneous indoor temperature T_{in} has been computed as the mean temperature value for each timestamp. The analysis has been performed by considering values in t_{slot}=$[5:00p.m.-9:00p.m]$ and data are aggregated over daily t_{window}. Similarly to the regression of a residential building presented in Figure 6.29, also the school building K_{tot} estimation (red line) from the linear regression has a low S value (1.39). The estimated K_{tot} value is 1.42, indicating a poor energy performance, at least when compared with residential buildings.

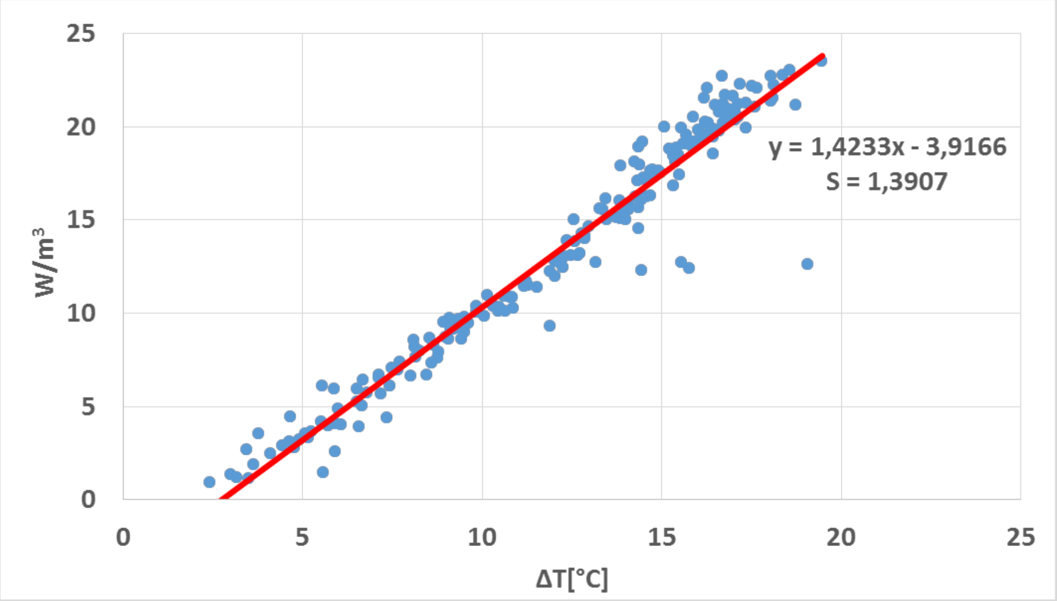
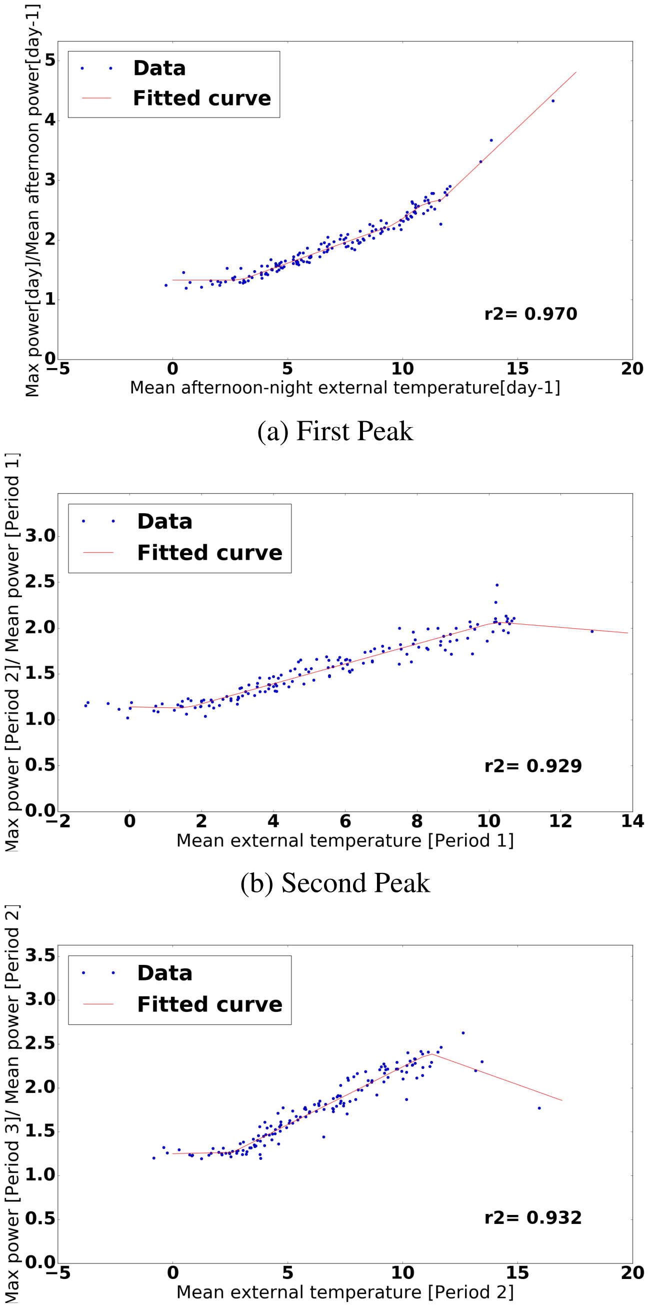


Fig. 6.31 School building, scatter plot of daily power consumption per unit of volume (W/m^3) with respect to T_{in}-T_{ex} difference (°C).

6.4.2 Characterization of the peak detection

The Peak Detection (PD) algorithm (Section 5.2.3.3.6) predicts the peak powers and the corresponding time instants for each heating cycle separately for each building. To evaluate the quality of the regression two indexes, all ready presented in Section 6.1 are exploited: (i) The R-squared and (ii) The Standard Error of Regression (denoted as S).

To evaluate the effectiveness of PD in correctly identifying the peak powers a given building (building ID no. 8) with a triple heating cycle is discussed as a representative example. The analysis has been performed considering the period from 1 November 2014 to 31 March 2015, which is almost a full Italian heating season. For each heating cycle Figure 6.32 reports the ratio of the peak power of the transient state and the mean power of previous (e.g., preceding day for the first peak) steady state (*y axis*) with respect to the mean external temperature in the previous steady state and OFF-line phase (x axis), together with the corresponding multivariate adaptive regression spline (continuous line in Figure 6.32 representing the peak power estimation). For each heating cycle, the (first/second/third) building peak power estimation has a very high r^2 value, as high as 0.92, or higher, representing a good approximation of the phenomenon under analysis.



(a) First Peak

(b) Second Peak

(c) Third Peak

Fig. 6.32 Triple heating cycle peaks

Table 6.10 shows the peak power estimation computed through PD and both r^2 and S values indicating how the regression is able to correctly model the studied phenomenon. Furthermore, the values of r^2 and S for all 12 buildings under analysis are computed separately for each heating cycle. Focusing on r^2, the first piece of evidence is that, for each peak, r^2 values are similar among different buildings. Furthermore r^2 values for the first peaks are slightly higher than the rest (second and third peaks), but always greater than 0.8 (except for one case with 0.62). Although the variability among S values is higher than r^2 among the considered buildings, the general trend is similar. Specifically, S values for the first peaks among different buildings are much higher than the rest, however all values are less than 1.7. In buildings with two or more heating cycles the prediction of the second and/or third peak become slightly weaker than the first one. The worst result is obtained on building number 7 where the first peak is modeled correctly, but the correlation of the second peak is the worst of the analyzed buildings. These results demonstrate the effectiveness of the proposed approach to predict the peak powers during the transient status with a limited error.

Table 6.10 Peak detection r^2 and S

Building	First Peak		Second Peak		Third Peak	
ID	r^2	S	r^2	S	r^2	S
1	0.92	1.05	-	-	-	-
2	0.90	1.44	-	-	-	-
3	0.95	1.71	-	-	-	-
4	0.96	1.26	-	-	-	-
5	0.96	0.92	-	-	-	-
6	0.96	1.06	0.85	0.52	-	-
7	0.89	1.01	0.62	0.75	-	-
8	0.97	0.51	0.92	0.32	0.89	0.38
9	0.96	0.47	0.93	0.52	0.89	0.42
10	0.94	1.15	0.94	0.54	0.89	0.76
11	0.92	1.01	0.86	0.63	0.80	0.79
12	0.92	0.66	0.89	0.48	0.81	0.58

6.4.3 Power prediction error

The values reported in Table 6.11 represent the average prediction errors for the 12 analyzed buildings. In particular, the MAPE and SMAPE values, all ready introduced in Section 6.1, refer to the power prediction performed using the Power Prediction (PP) algorithm described in Section 5.2.3.3.7. The average prediction errors are reported for each building and for each heating cycle of the day. Moreover, for each building, the overall MAPE and SMAPE values are reported, which include all predictions for both the transient and the steady state phases.

The reported values suggest an overall higher precision for predictions made on buildings with a single-cycle, since both overall MAPE and SMAPE increase with the number of heating cycles (even though some double-cycle buildings have lower error values than single-cycle buildings and some others have higher error values than triple-cycle buildings). This overall trend can be motivated by two mutually dependent reasons: (i) more heating cycles mean more (even if shorter) transient states, with higher prediction errors influencing the average values; (ii) more heating cycles mean also more separated steady states (rather than a continuous one) with different behaviors of the same heating system, also with similar weather conditions, depending on the period of the day.

Table 6.11 MAPE and SMAPE values for each test building

Heating cycles	Building ID	Overall		First cycle		Second cycle		Third cycle	
		MAPE	SMAPE	MAPE	SMAPE	MAPE	SMAPE	MAPE	SMAPE
Single	1	15.56	6.78	15.56	6.78	-	-	-	-
	2	18.58	7.95	18.58	7.95	-	-	-	-
	3	20.48	8.35	20.48	8.35	-	-	-	-
	4	22.38	9.32	22.38	9.32	-	-	-	-
	5	20.42	8.46	20.42	8.46	-	-	-	-
Double	6	23.24	9.62	28.81	10.95	20.58	8.06	-	-
	7	22.02	9.56	36.98	13.35	15.52	7.10	-	-
Triple	8	23.11	9.72	35.35	13.90	17.38	7.67	18.33	7.63
	9	27.96	10.62	28.46	10.90	24.73	10.14	25.87	10.85
	10	33.75	11.64	39.70	14.40	38.44	14.49	26.53	10.21
	11	29.05	11.83	31.89	11.98	37.53	13.99	23.23	9.58
	12	27.26	11.56	32.62	13.26	28.39	11.42	23.01	9.27

The plots in Figures 6.33-6.34 show the comparison between the real and predicted power values of single buildings, during a single day, plotted as the average values over intervals of 15 minutes. The plot in Figure 6.33 refers to a single-cycle building and the power values are forecast with a prediction horizon of 1 hour. Even though the peak is predicted with a 15-minute delay, its value is very near to the real one, while the prediction of the overall trend of the transient phase is similar to the real one, even though some points are sensibly different. The error in the steady phase is constantly low and close to zero in some points. This high level of precision is favored by the regular trend of the single steady phase in single-cycle buildings, both in a single day and from one day to another. The plot in Figure 6.34 refers to a triple-cycle building and the power values are forecast with a prediction horizon of 1 hour. In this case, except for the first cycle, the trends of the predicted transient phases are very similar to the real ones and in the third cycle the predicted peak value is very near to the real one. The error in the steady phase is higher than in Figure 6.33, but still acceptable.

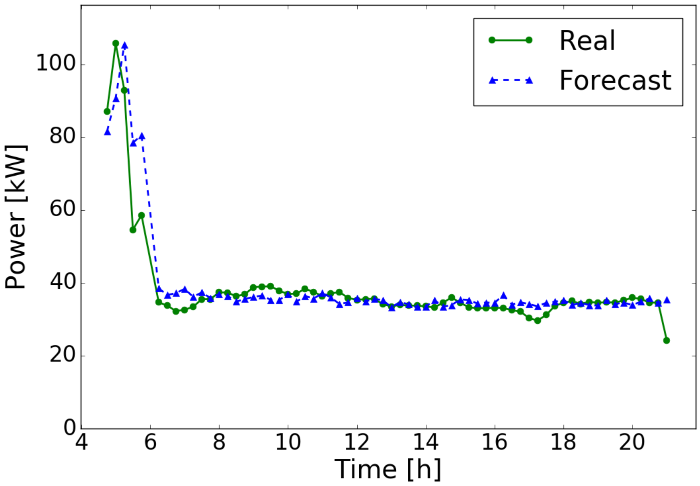


Fig. 6.33 Daily 15 minutes average power prediction for a single-cycle building with 1 hour advance (5% maximum error on weather forecast)

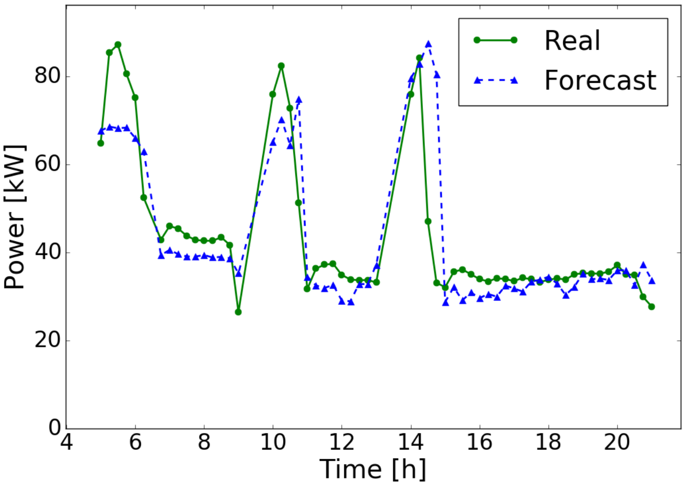
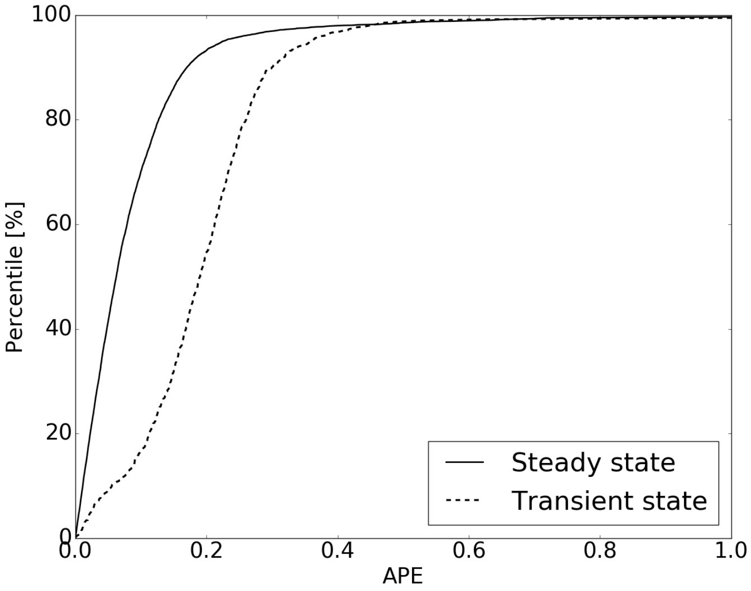
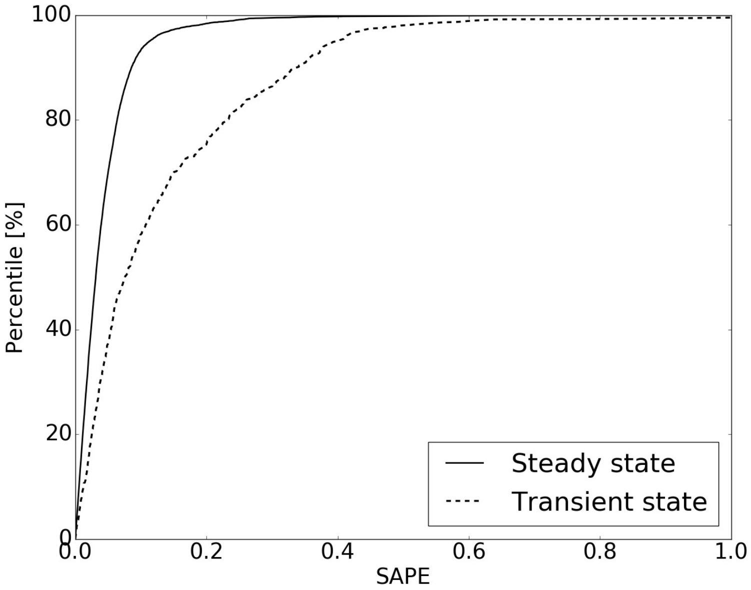


Fig. 6.34 Daily 15 minutes average power prediction for a triple-cycles building with 1 hour advance (5% maximum error on weather forecast)

The plots in Figure 6.35 represent the cumulative frequency of *Absolute Percentage Error* (APE) and of *Symmetric Absolute Percentage Error* (SAPE) of predictions for a single-cycle building during steady and transient states. These two metrics are the terms of the sums in the MAPE and SMAPE formulas respectively (see Section 6.1) and represent two measures of percentage error for single predictions. Over 90% of the predictions have a APE lower than 17% in the steady state and lower than 30% in the transient state. For the same percentile, SAPE is less than 8.6% in the steady state but about 33.7% in the transient state. However, in the same state a SAPE of just 15% is the 70th percentile. Therefore, roughly 90% of samples are predicted with a limited error, especially in the steady state. The steep initial growth of the two graphs in Figure 6.35 shows that only a very small number of predictions have high error values. Indeed, over 98% of the predictions have APE and SAPE lower than 50%, in both steady and transient states, while, among the remaining 2%, APE can have very high values (while SAPE $\leq$ 100% by definition). This suggests how few bad predictions can affect the overall MAPE and SMAPE values and explains why median error values are always lower than the corresponding means.



(a) Absolute Percentage Error



(b) Symmetric Absolute Percentage Error

Fig. 6.35 Percentile distribution of APE and SAPE over the whole season for a single-cycle building

6.4.4 Sensitivity analysis

6.4.4.1 Sensitivity and robustness of the energy signature method

The two main user-defined parameters of the proposed Energy Signature method are the aggregation period t_{window} and the t_{slot} day-time filter. Hence, an evaluation of the robustness of the proposed energy signature analysis to such parameter settings is presented. Table 6.12 shows the K_{tot} estimation computed through the linear regression and the S values indicating how the linear regression is able to correctly model the studied phenomenon. Furthermore, the comparison between the best and the average building allows to see the results from a different perspective.

Focusing on S, the first evidence is that the longer the t_{window}, the better the linear regression. For each t_{slot}, S values are lower for weekly t_{window} than daily t_{window}. S values for hourly t_{window} are much higher than the rest. This general behavior is expected and stems from the data smoothing effect of considering averages over longer periods of time, which hides outliers or temporary exceptional behaviors. Even if the equation $\forall t_{slot}, t_1 \geq t_2 \Rightarrow S_{t_{window}=t_1} \geq S_{t_{window}=t_2}$ holds true in all reported cases but one, is possible to see that the S values for hourly t_{window} are more sensitive to the t_{slot} selection. In particular, both the 5:00-7:00pm and the 5:00-9:00pm t_{slot} ranges yields similar results in terms of S values (of course, periods also overlap), whereas the 6:00am-10:00pm t_{slot} range for hourly t_{window} has extremely high S values: 4.61 for the best building, 7.41 for a random building. From such values, is possible to note that (i) the 6:00am-10:00pm t_{slot} range is generally the best fit for the linear regression, thanks to the longer period facilitating steady state modeling of the heating system and limiting the dynamic and thermal inertia effects; (ii) the hourly t_{window} often leads to unsatisfactory K_{tot} estimations, due to the poor fit of the linear regression. The exception to these findings is the 5:00-7:00pm t_{slot} for hourly t_{window}, which has a low S value (0.88) with respect to the average hourly model behavior (S always above 1.11). Finally, is possible to note that all combinations of parameters that have a low S (from Table 6.12, lower than 1.2) lead to a coherent and stable K_{tot} estimation: for each fixed t_{slot}, the best building K_{tot} estimation delta is always lower than 0.02, and the random building is always lower than 0.01.

		Best Building		Building N.*		District Mean
t_{window}	t_{slot}	K_{tot}	S	K_{tot}	S	K_{tot}
Weekly	6:00am-10:00pm	0.46	0.35	0.53	0.47	0.55
Weekly	5:00-7:00pm	0.51	0.67	0.72	0.55	0.74
Weekly	5:00-9:00pm	0.54	0.57	0.68	0.68	0.68
Daily	6:00am-10:00pm	0.46	0.64	0.53	0.55	0.54
Daily	5:00-7:00pm	0.53	1.02	0.72	0.90	0.73
Daily	5:00-9:00pm	0.55	0.62	0.67	0.77	0.67
Hourly	6:00am-10:00pm	0.36	4.61	0.49	7.41	0.51
Hourly	5:00-7:00pm	0.52	1.16	0.71	0.88	0.73
Hourly	5:00-9:00pm	0.53	1.11	0.64	2.38	0.64

Table 6.12 t_{window} and t_{slot} sensitivity and robustness.

6.4.4.2 Sensitivity analysis of Power Prediction

Here the robustness of the Power Prediction (PP) algorithm to the variation of its parameters is analyzed. For each parameter (i.e., *training window size*, *slot duration*, *prediction horizon*, and *weather maximum error* described below), a set of experiments were run to find, when possible, a good input parameters setting. The *training window size* (*trWdw*) was set to 7 and 14 days; The *slot duration* (*slDur*) was set to 15, 30 and 60 minutes; For each value, the daily timeline is split in fixed time slots, hence with a granularity of 15 minutes the slots start at 00:00, 00:15, 00:30, and so on. A similar partitioning is done for granularities of 30 (00:00, 00:30, etc.) and 60 minutes (00:00, 01:00, etc.). Finally, even if (near-)real-time predictions are based on forecasts of weather data, validation was performed with real measures of past weather data. Therefore, to take into account the prediction error, a random percentage value was added to such measures. The percentage error was modeled as a uniform random variable W with a support defined by the *weather maximum error* (*weErr*) parameter, i.e., $W \sim U[-weErr, +weErr]$. The value of *weErr* was set to 0%, 5% and 10%. Finally, the *prediction horizon* (*prHor*) was been set to 1, 2, 4, 8 and 24 hours and analyzed in combination with the other parameters. These five values were chosen to consider not only short-term, but also medium-term predictions, which even with lower precision values can still be of interest for some end users.

Tables 6.13 to 6.15 show how percentage errors, i.e. mean (MAPE) and median values, vary with respect to the aforementioned parameters.

Table 6.13 highlights the variation between the two different values of *training window size* (which determines the amount of training data). A wider training window (14-days) corresponds to lower error values, in both transient and steady states. Indeed, the prediction algorithm learns from a larger training set and can fit overall a more accurate hyper-plane. Wider training window sizes (e.g. 30 days) have been tested too, but they are not reported in Table 6.13 because no significant improvement has been noticed. The difference with the 7-day window is reduced for shorter *prediction horizons* and becomes negligible for short term predictions (only 0.27% the overall MAPE for *prHor*=1), with a trend reversal in the steady state, where the lowest values of mean and median errors are registered with the 7-day window. This means that a stricter training window can be preferable for predictions over a shorter horizon (1 hour or less) to make the algorithm fit the most recent samples better. Hence, days as the default value for *trWdw* are selected.

Table 6.13 Sensitivity analysis on *training window size*

prHor	*trWdw*	Overall error (%)			Transient error (%)			Steady error (%)		
(hours)	(days)	mean	median	std dev	mean	median	std dev	mean	median	std dev
1	7	10.76	6.58	22.52	24.05	19.48	34.21	9.24	5.96	20.21
	14	10.49	6.96	20.37	19.80	19.05	22.38	9.42	6.30	19.84
2	7	11.38	6.81	27.15	23.52	18.44	37.43	9.99	6.17	25.34
	14	10.84	7.10	23.74	19.75	18.29	30.14	9.82	6.44	22.67
4	7	12.28	7.13	31.31	23.53	18.34	38.43	10.99	6.44	30.12
	14	11.31	7.29	26.81	19.64	18.17	31.09	10.36	6.63	26.10
8	7	13.43	7.57	35.70	23.53	18.34	38.43	12.27	6.85	35.19
	14	11.98	7.53	29.92	19.64	18.17	31.09	11.10	6.84	29.66
24	7	14.76	8.13	34.32	24.00	18.81	36.68	13.70	7.39	33.88
	14	12.90	7.85	36.90	20.25	18.69	32.13	12.06	7.14	37.31

prHor: prediction horizon in hours
trWdw: training window size in days

Table 6.14 reports the variation of prediction errors with respect to the *slots duration*. Overall, the prediction error for *slDur*=60 is always substantially higher than for the other two values (between 0.68% and 1.37%). The lowest values of prediction error for all the *prediction horizons* are obtained with *slDur*=30 instead. This is true both in the steady state and for the overall errors. The transient state exhibits a higher variability and no particular trend can be detected. Hence, 30 minutes as the default value for *slDur* are selected.

Table 6.14 Sensitivity analysis on *slots duration*

prHor	*slDur*	Overall error (%)			Transient error (%)			Steady error (%)		
(hours)	(min)	mean	median	std dev	mean	median	std dev	mean	median	std dev
	15	10.45	6.64	21.94	22.99	19.30	32.34	9.25	6.06	20.27
1	30	10.46	6.77	20.23	21.47	19.44	28.31	9.12	6.15	18.57
	60	11.54	7.33	21.98	20.31	18.92	21.41	10.03	6.33	21.72
	15	10.93	6.83	25.68	22.62	18.42	38.28	9.81	6.27	23.83
2	30	10.86	6.94	23.50	20.54	18.08	32.49	9.68	6.31	21.87
	60	12.23	7.47	28.26	21.08	18.78	25.58	10.71	6.47	28.42
	15	11.70	7.11	29.84	22.62	18.42	38.28	10.66	6.51	28.69
4	30	11.44	7.17	26.07	20.54	18.08	32.49	10.33	6.50	24.95
	60	12.79	7.74	31.93	20.88	18.21	30.87	11.40	6.76	31.90
	15	12.71	7.44	34.63	22.62	18.42	38.28	11.76	6.82	34.11
8	30	12.33	7.50	30.10	20.54	18.08	32.49	11.34	6.80	29.65
	60	13.39	8.03	31.88	20.88	18.21	30.87	12.10	7.06	31.88
	15	13.74	7.82	36.47	22.41	18.70	34.38	12.91	7.19	36.56
24	30	13.67	8.02	35.57	21.66	18.51	32.82	12.70	7.28	35.77
	60	14.44	8.56	32.75	22.17	19.02	37.05	13.11	7.51	31.76

prHor: prediction horizon in hours
slDur: slot duration in minutes

Table 6.15 reports the variation of prediction errors with respect to the *weather maximum error*. In this case, mean error (MAPE) and median error have opposite trends. While MAPE is lower for higher values of *weErr* (especially for longer *prediction horizons*), the median values exhibit more straightforward behavior, as they are lower for lower values of *weErr* with a monotonic trend, i.e. $error(weErr = 0\%) < error(weErr = 5\%) < error(weErr = 10\%)$. In this case, a wise setting is to use higher values of *weErr* for longer prediction horizons.

Table 6.15 Sensitivity analysis on *weather maximum error*

prHor	*weErr*	Overall error (%)			Transient error (%)			Steady error (%)		
(hours)	(%)	mean	median	std dev	mean	median	std dev	mean	median	std dev
1	0	10.75	6.50	25.95	22.14	19.47	25.36	9.45	5.89	25.70
	5	10.49	6.74	19.55	22.40	19.32	34.31	9.12	6.10	16.52
	10	10.64	7.07	18.09	21.23	18.98	26.45	9.42	6.39	16.43
2	0	11.45	6.71	31.89	21.35	18.57	26.39	10.32	6.07	32.26
	5	10.92	6.91	22.76	22.52	18.49	42.53	9.58	6.27	18.79
	10	10.97	7.24	20.42	21.03	18.06	31.09	9.81	6.60	18.46
4	0	12.30	6.96	35.93	21.18	18.50	26.42	11.28	6.31	36.72
	5	11.61	7.17	27.52	22.36	18.37	43.10	10.38	6.48	24.83
	10	11.49	7.55	22.41	21.22	17.99	33.43	10.38	6.83	20.48
8	0	13.94	7.36	44.77	21.18	18.50	26.42	13.11	6.65	46.34
	5	12.21	7.48	27.09	22.36	18.37	43.10	11.05	6.79	24.33
	10	11.97	7.83	22.85	21.22	17.99	33.43	10.91	7.09	21.04
24	0	15.04	7.76	47.04	21.87	18.97	28.66	14.25	7.06	48.65
	5	13.49	7.91	31.32	23.13	18.84	43.26	12.39	7.18	29.44
	10	12.98	8.26	25.03	21.37	18.30	29.70	12.02	7.55	24.25

prHor: prediction horizon in hours
weErr: weather maximum percentage error

6.5 Simulation results of Indoor air Temperature

As case study for the *Building Temperature Simulation* methodology described in Section 5.2.4 a primary school of the EEB Project has been chosen. It is a public building of the *Settimo Torinese* Municipality with about $14,500$ m^2 spread on two floors (see Figure 6.36) with brick walls facades, double glazed windows and pitched roofs. The building is connected to the district heating distribution system. During working-days, the heating system cycle is from 4:00 a.m. to 7:30 p.m.. To ensure a comfortable environment for users the ignition of Monday is anticipated on previous Sunday at 11:00 p.m..

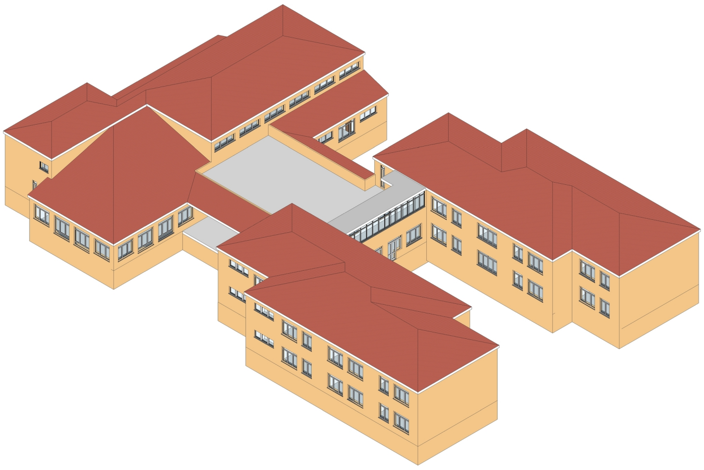


Fig. 6.36 BIM of Rodari Primary School

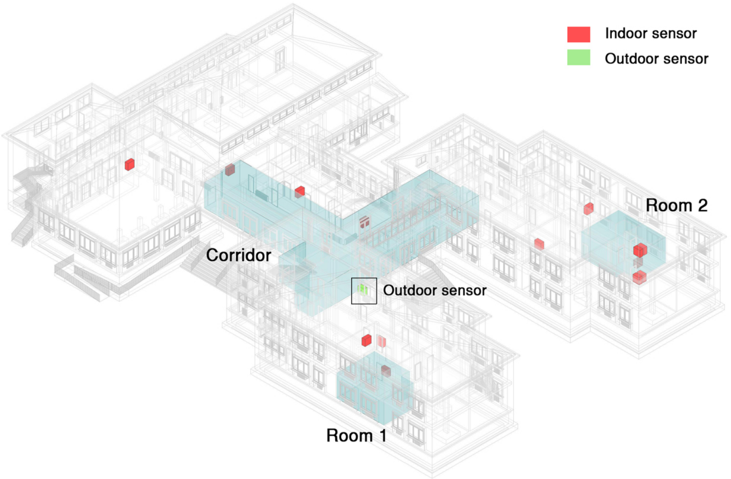


Fig. 6.37 Sensor positioning at Rodari Primary School

The building has been equipped with 16 IoT devices, 15 indoor and 1 outdoor (see Figure 6.37), to send air temperature and relative humidity. Sensors have been installed in the most meaningful building zones according to its intended use, construction type and floors number (i.e. main entrance, classrooms, gym and student canteen). The Wireless Sensor Network has been evaluated to optimize the employed IoT devices with respect to the good result of the energy simulation. In this study, indoor devices have been placed in comparable rooms in terms of use and dimension characterized by a different orientation. Instead the outdoor device has been placed at the worst solar exposure to detect the minor outdoor temperature. The symmetrical shapes and the regular internal distribution of the building have allowed us to select some reference rooms to collect enough data for energy analysis.

The energy simulation has been performed for the whole heating season. In this section, the results from January 9th to January 15th 2017 are presented. The validation model is achieved by comparing the Temperature trends, as described in Section 5.2.4. For this purpose, three selected rooms in the building (see Figure 6.37) are analyzed. These rooms have been chosen in relation to building shape and their occupancy during the week, as described in the following:

- **Room 1** is a classroom in the east part of the building occupied by 21 people. It is located in correspondence of thick trees that act as solar shield for the building.
- **Room 2** is a classroom in the west part of the building occupied by 22 people.
- **Corridor** is at the entrance of the school in a central position of the building. It is characterized by a very large environment with many openings and glazed windows. It does not have a constant occupancy during the day.

Both east- and west-oriented facades receive substantial contributions of thermal energy due to solar radiation. This is an advantage during winter season. Vice-versa, this translates into increased heat load during summer season, which would necessitate air conditioning. As the school is not equipped with such conditioning system, the simulations cover only the winter period.

Figure 6.38 reports three air temperature trends for the observation period: *i)* measured data coming from IoT devices (green line), *ii)* simulated with TMY Weather condition (red doted line) and *iii)* simulated with real-weather conditions (blue dashed line). The daily trends identifies the different phases of the heating cycle: *i)* ignition of the heating system (04:00 a.m.); *ii)* school entering (8:30 a.m.); *iii)* lunch break with opening windows for air circulation (12:30 a.m.); *iv)* school exiting (4:30 p.m.); *v)* shut-down of the heating system (07:30 p.m.). The air temperature chart highlights that measured data and simulation results with real-weather conditions have similar trends. On the contrary, the trend of TMY simulation results has the worst correlation with real samples. Especially during night hours, the temperature trend decreases to around 10 $°C$ with TMY simulations, while both measured and real-weather trends reaches about 16 $°C$. This because TMY refers to meteorological conditions, in

Table 6.16 Dispersion indicators of simulated indoor temperature against real measured values

Rooms	Indicator [%]	Real-weather Sim vs Measured	TMY Sim vs Measured
Room 1	MAD	8.02	16.82
	MBD	2.18	-16.64
	RMSD	9.78	19.01
Room 2	MAD	9.07	18.55
	MBD	0.10	-18.34
	RMSD	10.83	20.74
Corridor	MAD	9.35	16.94
	MBD	-0.17	-16.06
	RMSD	11.52	20.85

terms of temperature and solar radiation, significantly different to daily weather samples. Both simulations with real-weather data and TMY show a quicker slope of increase and decrease in the temperature trend when the heating system is switched *on* and *off*. This quicker response is related with the modelled heating capacity of the building. Indeed in the development of the BIM model the stratigraphy of the walls has been hypothesized fallowing the suggestions in [134]. Those hypothesis were necessary due to a lack of information on real wall stratigraphy data in the building documentation.

To evaluate the performance of the simulations three indicators of dispersion, all ready presented in section 6.1, have been used: *i) Mean Bias Difference* (*MBD*), *ii) Root Mean Square Difference* (*RMSD*) and *iii) Mean Absolute Difference* (*MAD*).

Table 6.16 details the error rates given comparing measured data with simulations performed with both real-weather and TMY conditions. As shown in Table 6.16, real-weather information improves the simulation results drastically with respect to TMY. Indeed, *MAD*, *MBD* and *RMSD* have lower values with real-weather conditions. In particular in *Room 1*, a *MAD* of 8.02% against 16.82% is obtained; a *MBD* of 2.18% against -16.64%; a *RMSD* of 9.78% against 19.01%. Similar results have been obtained for the other two rooms.

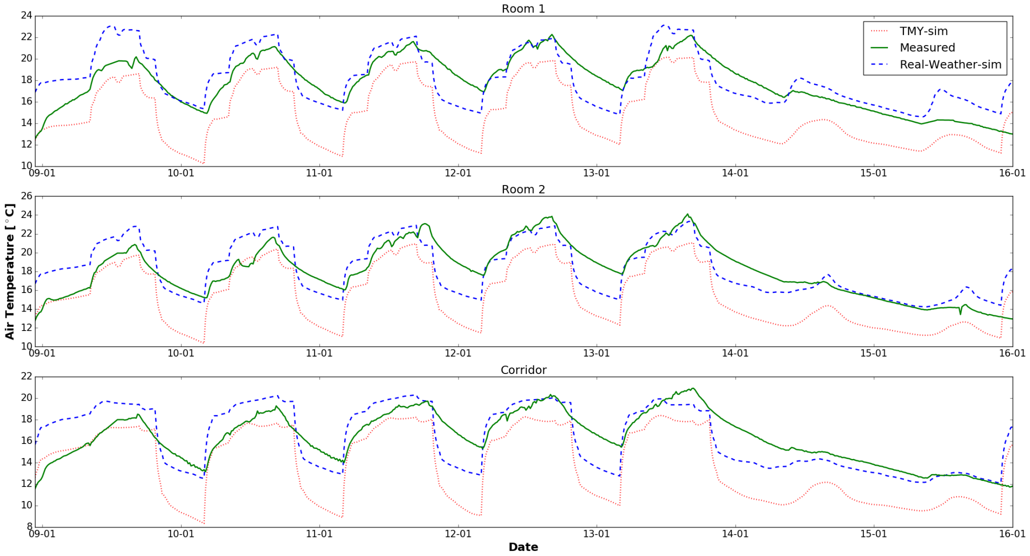


Fig. 6.38 Simulated and measured indoor air temperature trends between 9th-15th of January 2017.

Chapter 7

Conclusions

This thesis presented SMIRSE a distributed infrastructure for modelling Renewable and smart energy integration in urban contexts. SMIRSE is a novel flexible and modular distributed infrastructure, based on real-time grid simulation, for multi-purpose Smart Grid studies. Hence, it is flexible in simulating different scenarios in power grids without affecting the whole infrastructure. SMIRSE combines different technologies and correlates heterogeneous information, also sent in (near-) real-time, to simulate multi-energy-flows and to evaluate the impact of novel policies in cities and distribution networks. It exploits *Communication Adapters* that implements both publish/subscribe (MQTT) and request/response (REST) paradigms for communication and data exchange among the real-time grid simulator and different external modules. These modules can be software components, IoT devices, or other HW/SW directly connected to the real-time simulation for HIL or SIL. Both publish/subscribe and request/response are two communication models peculiar of IoT platforms, such as future Smart Metering Infrastructures. Thus, the proposed solution is also flexible i) to retrieve information in (near-) real-time from real IoT devices (i.e. smart meters) installed along the distribution network and ii) to feed strategies and models with real data during their running time. First, the motivations and challenges that have been addressed to design such infrastructure are discussed. Then, the state of the art solution for MES modeling and simulation with a particular focus on PV systems are presented. Latter, the SMIRSE framework that is suitable for general purpose energy simulations with different spatio-temporal resolutions is presented. Finally, simulation results of RES production, integration in the distribution grid and building thermal loads are presented.

From the Photovoltaic modeling and simulation point of view it overcomes the limitations of literature solutions by providing fine grained real-sky simulations considering also meteorological data from weather stations. Thus, addressing the challenges highlighted in [24]. Furthermore, the proposed solution performs simulations on spatial and temporal domains, providing energy profiles of PV systems with a good accuracy, as reported by the experimental results. Finally, the SMIRSE infrastructure can be use as a tool to help *Energy managers* and *RES engineers* in monitoring the performances of existing PV systems as depicted in the analysed *GalFer* system, where a malfunction has been identified.

An exemplification is presented and applied to the study of PV and storage integration in the distribution network of an urban district. The proposed example is simple but it is useful to demonstrate how the different distributed modules of the infrastructure can interact and exchange data, cooperating for obtaining the final results which is expected from an integrated co-simulation. Different load and generation scenarios can be easily fed into the whole simulation independently and also during running simulation. Storage control strategies can be tested as well without being locally integrated inside the grid model.

Furthermore, the PPBEC module is presented. It provides a wide range of software services including data collection and integration as well as an advanced analytics task, i.e., predicting heating consumption in buildings. PPBEC has been developed in a distributed environment to efficiently handle big datasets by exploiting Apache Spark upon MongoDB data repository. The performed experiments on real data highlighted the ability of PPBEC to effectively predict heating consumption in buildings with a limited error, and its good scalability.

References

[1] United Nations, FCCC. Adoption of the Paris Agreement. Proposal by the President, 2015. Available: http://unfccc.int/resource/docs/2015/cop21/eng/l09r01.pdf.

[2] Luigi Dusonchet and Enrico Telaretti. Economic analysis of different supporting policies for the production of electrical energy by solar photovoltaics in western european union countries. *Energy Policy*, 38(7):3297–3308, 2010.

[3] International Energy Agency. Tracking clean energy progress 2016, 2016.

[4] Lorie Wigle. How the internet of things will enable vast new levels of efficiency alan rose, intel corporation dr. subramanian vadari, modern grid solutions. 2014.

[5] Kimmo Kauhaniemi and Lauri Kumpulainen. Impact of distributed generation on the protection of distribution networks. In *Proc. of Developments in Power System Protection. Eighth IEEE International Conference on*, volume 1, pages 315–318. IET, 2004.

[6] Fernando Bastiao, Paulo Cruz, and Rui Fiteiro. Impact of distributed generation on distribution networks. In *Proc. of 5th International Conference on the European Electricity Market*, pages 1–6. IEEE, 2008.

[7] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM CSUR*, June 2003.

[8] Filippo Spertino and Fabio Corona. Monitoring and checking of performance in photovoltaic plants: A tool for design, installation and maintenance of grid-connected systems. *Renewable Energy*, 60:722 – 732, 2013.

[9] Shahir Daya, Nguyen Van Duy, Kameswara Eati, Carlos M Ferreira, Dejan Glozic, Vasfi Gucer, Manav Gupta, Sunil Joshi, Valerie Lampkin, Marcelo Martins, et al. *Microservices from Theory to Practice: Creating Applications in IBM Bluemix Using the Microservices Approach*. IBM Redbooks, 2016.

[10] John A Duffie and William A Beckman. *Solar Engineering of Thermal Processes*. Wiley, 2013.

[11] S Karatasou, M Santamouris, and V Geros. Analysis of experimental data on diffuse solar radiation in athens, greece, for building applications. *International journal of sustainable energy*, 23(1-2):1–11, 2003.

[12] UN Habitat - Energy, 2017.

[13] United Nations, Department of Economic and Social Affairs, Population Division. *The World's Cities in 2016 – Data Booklet (ST/ESA/ SER.A/392).*

[14] Pierluigi Mancarella. Mes (multi-energy systems): An overview of concepts and evaluation models. *Energy*, 65:1–17, 2014.

[15] Christoph Molitor, Stephan Gross, Jakob Zeitz, and Antonello Monti. Mescos—a multienergy system cosimulator for city district energy systems. *IEEE Transactions on Industrial Informatics*, 10(4):2247–2256, 2014.

[16] Jonas Allegrini, Kristina Orehounig, Georgios Mavromatidis, Florian Ruesch, Viktor Dorer, and Ralph Evins. A review of modelling approaches and tools for the simulation of district-scale energy systems. *Renewable and Sustainable Energy Reviews*, 52:1391–1404, 2015.

[17] James Keirstead, Mark Jennings, and Aruna Sivakumar. A review of urban energy system models: Approaches, challenges and opportunities. *Renewable and Sustainable Energy Reviews*, 16(6):3847–3866, 2012.

[18] I Van Beuzekom, M Gibescu, and JG Slootweg. A review of multi-energy system planning and optimization tools for sustainable urban development. In *PowerTech, 2015 IEEE Eindhoven*, pages 1–7. IEEE, 2015.

[19] Ettore Bompard, Bei Han, Marcelo Masera, and Enrico Pons. Smart grid as multi-layer interacting system for complex decision makings. In *Networks of Networks: The Last Frontier of Complexity*, pages 187–201. Springer, 2014.

[20] Francesco Gavino Brundu, Edoardo Patti, Anna Osello, Matteo Del Giudice, Niccolò Rapetti, Alexandr Krylovskiy, Marco Jahn, Vittorio Verda, Elisa Guelpa, Laura Rietto, and Andrea Acquaviva. Iot software infrastructure for energy management and simulation in smart cities. *IEEE Transactions on Industrial Informatics*, 13(2):832–840, 2017.

[21] Vittorio Verda, Elisa Guelpa, Adriano Sciacovelli, Andrea Acquaviva, and Edoardo Patti. Thermal peak load shaving through users request variations. *International Journal of Thermodynamics*, 19(3):168–176, 2016.

[22] Lorenzo Bottaccioli, Alessandro Aliberti, Francesca Ugliotti, Edoardo Patti, Anna Osello, Enrico Macii, and Andrea Acquaviva. Building energy modelling and monitoring by integration of iot devices and building information models. In *Computer Software and Applications Conference (COMPSAC), 2017 IEEE 41st Annual*, volume 1, pages 914–922. IEEE, 2017.

[23] Lorenzo Bottaccioli, Edoardo Patti, Andrea Acquaviva, Enrico Macii, Matteo Jarre, and Michel Noussan. A tool-chain to foster a new business model for photovoltaic systems integration exploiting an energy community approach. In *Proc. of IEEE ETFA2015*. IEEE, 2015.

[24] Bernd Resch, Günther Sagl, Tobias Törnros, Andreas Bachmaier, Jan-Bleicke Eggers, Sebastian Herkel, Sattaya Narmsara, and Hartmut Gündra. GIS-based planning and modeling for renewable energy: Challenges and future research avenues. *ISPRS IJGI*, 3(2):662–692, 2014.

[25] S. Freitas, C. Catita, P. Redweik, and M.C. Brito. Modelling solar potential in the urban environment: State-of-the-art review. *Renew. Sustainable Energy Rev.*, 41:915–931, 2015.

[26] Ryan Firestone. Distributed energy resources customer adoption model technology data. *Berkeley Lab, Berkeley, CA, USA Case Study*, 2004.

[27] Tom Lambert, Paul Gilman, and Peter Lilienthal. Micropower system modeling with homer. *Integration of alternative sources of energy*, pages 379–418, 2006.

[28] Henrik Lund. Energyplan-advanced energy systems analysis computer model. *Documentation version*, 9, 2011.

[29] Kyle Anderson, Jimmy Du, Amit Narayan, and Abbas El Gamal. Gridspice: A distributed simulation platform for the smart grid. *IEEE Transactions on Industrial Informatics*, 10(4):2354–2363, 2014.

[30] Abdalkarim Awad, Peter Bazan, and Reinhard German. Sgsim: Co-simulation framework for ict-enabled power distribution grids. In *Proc. of MMB & DF*, pages 5–8. Springer, 2016.

[31] Peter Riederer, Vincent Partenay, Nicolas Perez, Christophe Nocito, Romain Trigance, and Thierry Guiot. Development of a simulation platform for the evaluation of district energy system performances. In *Fourteenth International IBPSA Conference*, pages 2499–2506, 2015.

[32] Steffen Schütte, Stefan Scherfke, and Martin Tröschel. Mosaik: A framework for modular simulation of active components in smart grids. In *Proc. of SGMS11*, pages 55–60. IEEE, 2011.

[33] S. Rohjans, S. Lehnhoff, S. Schütte, S. Scherfke, and S. Hussain. mosaik - a modular platform for the evaluation of agent-based smart grid control. In *Proc. of PES ISGT*, pages 1–5, Oct 2013.

[34] T. Hess, J. Dickert, and P. Schegner. Multivariate power flow analyses for smart grid applications utilizing mosaik. In *Proc. of PES ISGT*, pages 1–6, Oct 2016.

[35] Ruben Baetens, Roel De Coninck, Filip Jorissen, Damien Picard, Lieve Helsen, and Dirk Saelens. Openideas-an open framework for integrated district energy simulations. In *Proceedings of Building Simulation 2015*, 2015.

[36] L Andrew Bollinger and Ralph Evins. Facilitating model reuse and integration in an urban energy simulation platform. *Procedia Computer Science*, 51:2127–2136, 2015.

[37] Hanno Georg, Sven Christian Müller, Christian Rehtanz, and Christian Wietfeld. Analyzing cyber-physical energy systems: The inspire cosimulation of power and ict systems using hla. *IEEE Trans. on Industrial Informatics*, 10(4):2364–2373, 2014.

[38] Omid Abrishambaf, Pedro Faria, Luis Gomes, João Spínola, Zita Vale, and Juan M Corchado. Implementation of a real-time microgrid simulation platform based on centralized and distributed management. *Energies*, 10(6):806, 2017.

[39] Chia-han Yang, Gulnara Zhabelova, Chen-Wei Yang, and Valeriy Vyatkin. Cosimulation environment for event-driven distributed controls of smart grid. *IEEE Trans. on Industrial Informatics*, 9(3):1423–1435, 2013.

[40] M. Manbachi, A. Sadu, H. Farhangi, A. Monti, A. Palizban, F. Ponci, and S. Arzanpour. Real-time co-simulation platform for smart grid volt-var optimization using iec 61850. *IEEE Trans. on Industrial Informatics*, 12(4):1392–1402, Aug 2016.

[41] Lorenzo Bottaccioli, Abouzar Estebsari, Edoardo Patti, Enrico Pons, and Andrea Acquaviva. A novel integrated real-time simulation platform for assessing photovoltaic penetration impacts in smart grids. *Energy Procedia*, 111:780–789, 2017.

[42] Lorenzo Bottaccioli, Enrico Macii, Edoardo Patti, Abouzar Estebsari, Enrico Pons, and Andrea Acquaviva. Pvingrid: A distributed infrastructure for evaluating the integration of photovoltaic systems in smart grid. In *Proc. of 8th Advanced Doctoral Conference on Computing, Electrical and Industrial Systems*. Springer, 2017.

[43] V. Venkataramanan, A. Srivastava, and A. Hahn. Real-time co-simulation testbed for microgrid cyber-physical analysis. In *Proc. of MSCPES*, pages 1–6, April 2016.

[44] Morris Brenna, Ettore De Berardinis, Luca Delli Carpini, Federica Foiadelli, Pietro Paulon, Paola Petroni, Gianluca Sapienza, Giorgio Scrosati, and Dario Zaninelli. Automatic distributed voltage control algorithm in smart grids applications. *IEEE Trans. on Smart Grid*, 4(2):877–885, 2013.

[45] A. Estebsari, E. Pons, E. Patti, M. Mengistu, E. Bompard, A. Bahmanyar, and S. Jamali. An iot realization in an interdepartmental real time simulation lab for distribution system control and management studies. 2016.

[46] Luca Bergamasco and Pietro Asinari. Scalable methodology for the photovoltaic solar energy potential assessment based on available roof surface area: further improvements by ortho-image analysis and application to turin (italy). *Solar Energy*, 85(11):2741–2756, 2011.

[47] Md. Humayun Kabir, Wilfried Endlicher, and Jonas Jägermeyr. Calculation of bright roof-tops for solar pv applications in dhaka megacity, bangladesh. *Renewable Energy*, 35(8):1760 – 1764, 2010.

[48] Jaroslav Hofierka and Ján Kaňuk. Assessment of photovoltaic potential in urban areas using open-source solar radiation tools. *Renewable Energy*, 34(10):2206–2214, 2009.

[49] Marcel Šúri and Jaroslav Hofierka. A new gis-based solar radiation model and its application to photovoltaic assessments. *Trans. on GIS*, 8(2):175–190, 2004.

[50] J. Alstan Jakubiec and Christoph F. Reinhart. A method for predicting city-wide electricity gains from photovoltaic panels based on lidar and gis data combined with hourly daysim simulations. *Solar Energy*, 93:127–143, 2013.

[51] Niko Lukac, Sebastijan Seme, Danijel Zlaus, Gorazd Stumberger, and Borut Zalik. Buildings roofs photovoltaic potential assessment based on lidar (light detection and ranging) data. *Energy*, 66:598–609, 2014.

[52] Luis Ramirez Camargo, Roland Zink, Wolfgang Dorner, and Gernot Stoeglehner. Spatio-temporal modeling of roof-top photovoltaic panels for improved technical potential assessment and electricity peak load offsetting at the municipal scale. *Comput. Environ. Urban Syst.*, 52:58–69, 2015.

[53] Miguel C Brito, Nuno Gomes, Teresa Santos, and José A Tenedório. Photovoltaic potential in a lisbon suburb using lidar data. *Solar Energy*, 86(1):283–288, 2012.

[54] Xiaoyan Li, Liping Di, Weiguo Han, Peisheng Zhao, and Upendra Dadi. Sharing geoscience algorithms in a web service-oriented environment (grass gis example). *Computers & Geosciences*, 36(8):1060–1068, 2010.

[55] GRASS Development Team. *Geographic Resources Analysis Support System (GRASS GIS) Software*. Open Source Geospatial Foundation, USA, 2015.

[56] Fang Qiu, Feng Ni, Bryan Chastain, Haiting Huang, Peisheng Zhao, Weiguo Han, and Liping Di. Gwass: Grass web application software system based on the geobrain web service. *Computers & Geosciences*, 47:143–150, 2012.

[57] Manan Suri, Thomas Huld, ED Dunlop, and T Cebecauer. Geographic aspects of photovoltaics in europe: contribution of the pvgis website. *IEEE J-STARS*, 1(1):34–41, 2008.

[58] PVGIS. http://re.jrc.ec.europa.eu/pvgis/apps4/pvest.php.

[59] Bill Marion and Mary Anderberg. PVWATTS - an online performance calculator for grid-connected PV systems. In *Proc. of ASES solar conference*, pages 119–124, 2000.

[60] Mapdwell Solar System. http://www.mapdwell.com.

[61] Luıs de Sousa, Christopher Eykamp, Ulrich Leopold, Olivier Baume, and Christian Braun. iguess-a web based system integrating urban energy planning and assessment modelling for multi-scale spatial decision making. In *Porc. of iEMSs 2012*, 2012.

[62] Raffaele De Amicis, Giuseppe Conti, Daniela Patti, Martin Ford, and Pietro Elisei. I-scope-interoperable smart city services through an open platform for urban ecosystems, 2012.

[63] Matej Brumenm, Niko Lukac, and Borut Zalik. GIS application for solar potential estimation on buildings roofs. In *Proc. of IARIA WEB*, 2015.

[64] Lionel Ménard, Isabelle Blanc, Didier Beloin-Saint-Pierre, Benoît Gschwind, Lucien Wald, Philippe Blanc, Thierry Ranchin, Roland Hischier, Simone Gianfranceschi, Steven Smolders, et al. Benefit of geoss interoperability in assessment of environmental impacts illustrated by the case of photovoltaic systems. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 5(6):1722–1728, 2012.

[65] Edoardo Patti, Angeliki Lydia Antonia Syrri, Martin Jahn, Pierluigi Mancarella, Andrea Acquaviva, and Enrico Macii. Distributed software infrastructure for general purpose services in smart grid. *IEEE Transactions on Smart Grid*, 7(2):1156–1163, March 2016.

[66] E. Patti, E. Pons, D. Martellacci, F. B. Castagnetti, A. Acquaviva, and E. Macii. multiflex: Flexible multi-utility, multi-service smart metering architecture for energy vectors with active prosumers. In *Proc. of SMARTGREENS*, pages 1–6, May 2015.

[67] Kyle Anderson, Jimmy Du, Amit Narayan, and Abbas El Gamal. Gridspice: A distributed simulation platform for the smart grid. *IEEE Trans. on Industrial Informatics*, 10(4):2354–2363, 2014.

[68] M. Fowler and J. Lewis. Microservices, 2014. Available: http://martinfowler.com/articles/microservices.html.

[69] S. Newman. *Building Microservices*. O'Reilly Media, Inc., 2015.

[70] A. Krylovskiy, M. Jahn, and E. Patti. Designing a smart city internet of things platform with microservice architecture. In *2015 3rd International Conference on Future Internet of Things and Cloud*, pages 25–30, Aug 2015.

[71] R. T. Fielding. *Architectural styles and the design of network-based software architectures*. Open Source Geospatial Foundation, 2000.

[72] Roy Thomas Fielding. Rest: architectural styles and the design of network-based software architectures. *Doctoral dissertation, University of California*, 2000.

[73] Open Geospatial Consortium. Available: http://www.opengeospatial.org/.

[74] Roy T. Fielding and Richard N. Taylor. Principled design of the modern web architecture. *ACM Trans. Internet Technol.*, 2(2):115–150, May 2002.

[75] MQTT, Accessed Jan. 2016.

[76] Weather Underground. http://www.wunderground.com.

[77] Edoardo Patti, Angeliki Lydia Antonia Syrri, Marco Jahn, Pierluigi Mancarella, Andrea Acquaviva, and Enrico Macii. Distributed software infrastructure for general purpose services in smart grid. *IEEE Transactions on Smart Grid*, 7(2):1156–1163, 2016.

[78] Marco Pau, Edoardo Patti, Luca Barbierato, Abouzar Estebsari, Enrico Pons, Ferdinanda Ponci, and Antonello Monti. A cloud-based smart metering infrastructure for distribution grid services and automation. *Sustainable Energy, Grids and Networks*, 2017.

[79] E. Patti, A. Acquaviva, and E. Macii. Enable sensor networks interoperability in smart public spaces through a service oriented approach. In *5th IEEE International Workshop on Advances in Sensors and Interfaces IWASI*, pages 2–7, June 2013.

[80] Marco Pau, Edoardo Patti, Luca Barbierato, Abouzar Estebsari, Enrico Pons, Ferdinanda Ponci, and Antonello Monti. A cloud-based smart metering infrastructure for distribution grid services and automation. *SUSTAINABLE ENERGY, GRIDS AND NETWORKS*.

[81] JA Ruiz-Arias, H Alsamamra, J Tovar-Pescador, and D Pozo-Vázquez. Proposal of a regressive model for the hourly diffuse solar radiation under all sky conditions. *Energy Conversion and Management*, 51(5):881–893, 2010.

[82] Lorenzo Bottaccioli, Edoardo Patti, Enrico Macii, and Andrea Acquaviva. Gis-based software infrastructure to model pv generation in fine-grained spatio-temporal domain. *IEEE Systems Journal*, 2017.

[83] Vinco Sara, Bottaccioli Lorenzo, Patti Edoardo, Acquaviva Andrea, Macii Enrico, and Poncino Massimo. Gis-based optimal photovoltaic panel floorplanning for residential installations.

[84] Sara Vinco, Lorenzo Bottaccioli, Edoardo Patti, Andrea Acquaviva, and Massimo Poncino. A compact pv panel model for cyber-physical systems in smart cities. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2018.

[85] Lorenzo Bottaccioli, Abouzar Estebsari, Enrico Pons, Ettore Bompard, Enrico Macii, Edoardo Patti, and Andrea Acquaviva. A flexible distributed infrastructure for real-time co-simulations in smart grids. *IEEE Transactions on Industrial Informatics*, 2017.

[86] Andrea Acquaviva, Daniele Apiletti, Antonio Attanasio, Elena Baralis, Lorenzo Bottaccioli, Federico Boni Castagnetti, Tania Cerquitelli, Silvia Chiusano, Enrico Macii, Dario Martellacci, et al. Energy signature analysis: Knowledge at your fingertips. In *Big Data (BigData Congress), 2015 IEEE International Congress on*, pages 543–550. IEEE, 2015.

[87] Cory RA Hallam and Carolina Contreras. Evaluation of the levelized cost of energy method for analyzing renewable energy systems: A case study of system equivalency crossover points under varying analysis assumptions. *Systems Journal, IEEE*, 9(1):199–208, 2015.

[88] Jeremy Lin. Potential impact of solar energy penetration on pjm electricity market. *Systems Journal, IEEE*, 6(2):205–212, 2012.

[89] Chandrashekhar Lavania, Shrisha Rao, and Eswaran Subrahmanian. Reducing variation in solar energy supply through frequency domain analysis. *Systems Journal, IEEE*, 6(2):196–204, 2012.

[90] T. R. Ender, J. Murphy, and C. L. Haynes. A framework for portfolio management of renewable hybrid energy sources. *IEEE Systems Journal*, 4(3):295–302, Sept 2010.

[91] Jáchym Čepickỳa and Luís Moreira de Sousa. New implementation of ogc web processing service in python programming language. pywps-4 and issues we are facing with processing of large raster data using ogc wps. *ISPRS–International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41:927–930, 2016.

[92] Fr Linke. Transmissions-koeffizient und trübungsfaktor. *Beitr. Phys. Fr. Atmos*, 10:91–103, 1922.

[93] SoDa. http://www.soda-is.com/.

[94] D.G. Erbs, S.A. Klein, and J.A. Duffie. Estimation of the diffuse radiation fraction for hourly, daily and monthly-average global radiation. *Solar Energy*, 28(4):293–302, 1982.

[95] NA Engerer. Minute resolution estimates of the diffuse fraction of global irradiance for southeastern australia. *Solar Energy*, 116:215–237, 2015.

[96] J.A. Ruiz-Arias, H. Alsamamra, J. Tovar-Pescador, and D. Pozo-Vázquez. Proposal of a regressive model for the hourly diffuse solar radiation under all sky conditions. *Energy Conversion and Management*, 51(5):881 – 893, 2010.

[97] Douglas T Reindl, William A Beckman, and John A Duffie. Diffuse fraction correlations. *Solar energy*, 45(1):1–7, 1990.

[98] Arvid Skartveit and Jan Asle Olseth. A model for the diffuse fraction of hourly global radiation. *Solar Energy*, 38(4):271–274, 1987.

[99] Christian A Gueymard and Jose A Ruiz-Arias. Extensive worldwide validation and climate sensitivity analysis of direct irradiance predictions from 1-min global irradiance. *Solar Energy*, 2015.

[100] CP Jacovides, FS Tymvios, VD Assimakopoulos, and NA Kaltsounides. Comparative study of various correlations in estimating hourly diffuse fraction of global solar radiation. *Renewable Energy*, 31(15):2492–2504, 2006.

[101] Pierre Ineichen. Comparison and validation of three global-to-beam irradiance models against ground measurements. *Solar Energy*, 82(6):501–512, 2008.

[102] Amos Ronzino, Anna Osello, Edoardo Patti, Lorenzo Bottaccioli, Chiara Danna, Andrea Maria Lingua, Andrea Acquaviva, Enrico Macii, Michelangelo Grosso, Gianluca Messina, and Gaetano Rasconà. The energy efficiency management at urban scale by means of integrated modelling. In *Proc.of SEB-15*. Elsevier.

[103] Fouzia Brihmat and Said Mekhtoub. Pv cell temperature/pv power output relationships homer methodology calculation. In *International Journal of Scientific Research & Engineering Technology*, volume 1. International Publisher &C. O, 2014.

[104] Michel Mattei, Gilles Notton, Christian Cristofari, Marc Muselli, and Philippe Poggi. Calculation of the polycrystalline pv module temperature using a simple method of energy balance. *Renewable energy*, 31(4):553–567, 2006.

[105] Clemens Schwingshackl, Marcello Petitta, Jochen Ernst Wagner, Giorgio Belluardo, David Moser, Mariapina Castelli, Marc Zebisch, and Anke Tetzlaff. Wind effect on pv module temperature: Analysis of different techniques for an accurate estimation. *Energy Procedia*, 40:77–86, 2013.

[106] Arno HM Smets, Klaus Jäger, Olindo Isabella, René ACMM Van Swaaij, and Miro Zeman. *Solar Energy: The physics and engineering of photovoltaic conversion, technologies and systems*. UIT Cambridge Limited, 2016.

[107] A. M. Reis, N. T. Coleman, M. W. Marshall, P. A. Lehman, and C. E. Chamberlin. Comparison of PV module performance before and after 11-years of field exposure. In *Proc. of IEEE PVSC*, pages 1432–1435, 2002.

[108] Solar Power World. What is a combiner box? https://www.solarpowerworldonline.com.

[109] Hermann Kopetz. Internet of things. In *Real-Time Systems*, chapter 13, pages 307–323. Springer US, 2011.

[110] LinkSmart Middleware, 2016.

[111] Weather Underground. Weather underground web service, 2016.

[112] Città di Torino. Geoportale del comune di torino, 2017.

[113] E. Patti, A. L. A. Syrri, M. Jahn, P. Mancarella, A. Acquaviva, and E. Macii. Distributed software infrastructure for general purpose services in smart grid. *IEEE Transactions on Smart Grid*, 7(2):1156–1163, March 2016.

[114] Kristina Chodorow and Michael Dirolf. *MongoDB: The Definitive Guide*. O'Reilly Media, Inc., 1st edition, 2010.

[115] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. In *USENIX Hot Topics in Cloud Computing*, pages 10–10, 2010.

[116] SJ Qin and Weihua Li. Detection and identification of faulty sensors with maximized sensitivity. In *American Control Conference, 1999. Proceedings of the 1999*, volume 1, pages 613–617. IEEE, 1999.

[117] Jerome H Friedman. Multivariate adaptive regression splines. *The annals of statistics*, pages 1–67, 1991.

[118] Min-Yuan Cheng and Minh-Tu Cao. Accurately predicting building energy performance using evolutionary multivariate adaptive regression splines. *Applied Soft Computing*, pages 178–188, 2014.

[119] Andrew Ng. Cs229 lecture notes: Supervised learning. Stanford University, 2012.

[120] Andrea Acquaviva, Daniele Apiletti, Antonio Attanasio, Elena Baralis, Lorenzo Bottaccioli, Federico Boni Castagnetti, Tania Cerquitelli, Silvia Chiusano, Enrico Macii, Dario Martellacci, and Edoardo Patti. Energy signature analysis: Knowledge at your fingertips. In *2015 IEEE International Congress on Big Data, New York City, NY, USA, June 27 - July 2, 2015*, pages 543–550, 2015.

[121] Lorenzo Belussi and Ludovico Danza. Method for the prediction of malfunctions of buildings through real energy consumption analysis: Holistic and multidisciplinary approach of energy signature. *Energy and Buildings*, 55:715–720, 2012.

[122] Jimmy Vesterberg, Staffan Andersson, and Thomas Olofsson. Robustness of a regression approach, aimed for calibration of whole building energy simulation tools. *Energy and Buildings*, 81(0):430 – 434, 2014.

[123] Cristian Ghiaus. Experimental estimation of building energy performance by robust regression. *Energy and buildings*, 38(6):582–587, 2006.

[124] S Danov, J Carbonell, J Cipriano, and J Martí-Herrero. Approaches to evaluate building energy performance from daily consumption data considering dynamic and solar gain effects. *Energy and Buildings*, 57:110–118, 2013.

[125] European Parliament. Directive 2010/31/EU of the European Parliament and of the Council of 19 May 2010 on the energy performance of buildings, 2010.

[126] Autodesk. Revit, 2017. Available at: http://www.autodesk.com/products/revit-family/overview/.

[127] U.S. Department of Energy's (DOE) Building Technologies Office (BTO). EnergyPlus, 2017. Available at: https://energyplus.net/.

[128] Christian A Gueymard. A review of validation methodologies and statistical performance indicators for modeled solar radiation data: Towards a better bankability of solar projects. *Renewable and Sustainable Energy Reviews*, 39:1024–1034, 2014.

[129] S. Younes, R. Claywell, and T. Muneer. Quality control of solar radiation data: Present status and proposed new approaches. *Energy*, 30(9):1533 – 1549, 2005. Measurement and Modelling of Solar Radiation and Daylight-Challenges for the 21st Century.

[130] Christian A. Gueymard. A review of validation methodologies and statistical performance indicators for modeled solar radiation data: Towards a better bankability of solar projects. *Renewable and Sustainable Energy Reviews*, 39:1024 – 1034, 2014.

[131] Marco Pau, Edoardo Patti, Lorenzo Barbierato, Abouzar Estebsari, Enrico Pons, Ferdinanda Ponci, and Antonello Monti. Low voltage system state estimation based on smart metering infrastructure. In *2016 IEEE International Workshop on Applied Measurements for Power Systems (AMPS)*, pages 1–6, 2016.

[132] Simone Maggiore. Analisi di impatto dell'introduzione della tariffa bioraria obbligatoria. *Prot: 13000580 RSE SpA*, 2012.

[133] Gianfranco Chicco, Valeria Cocina, Paolo Di Leo, Filippo Spertino, and Alessandro Massi Pavan. Error assessment of solar irradiance forecasts and ac power from energy conversion model in grid-connected photovoltaic systems. *Energies*, 9(1):8, 2015.

[134] Project TABULA, 2017. Available at: http://episcope.eu/index.php?id=97/.

Appendix A

List of Publications

Lorenzo Bottaccioli has been author/coauthor in:

- 4 accepted papers in peer review international journals
- 11 accepted in peer review international national conferences
- 1 accepted papers in peer review Italian national conference

Peer reviewed International Journals:

J.1 Lorenzo Bottaccioli, Abouzar Estebsari, Enrico Pons, Ettore Bompard, Enrico Macii, Edoardo Patti, Andrea Acquaviva, A Flexible Distributed Infrastructure for Real-Time Co-Simulations in Smart Grids. In: IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS vol. 13 n. 6, pp. 3265-3274. - ISSN 1551-3203 .

J.2 Lorenzo Bottaccioli, Edoardo Patti, Enrico Macii, Andrea Acquaviva, GIS-based Software Infrastructure to Model PV Generation in Fine-grained Spatio-temporal Domain. In: IEEE SYSTEMS JOURNAL. - ISSN 1932-8184 (In Press)

J.3 R. Borchiellini, S.P. Corgnati, C. Becchio, C. Delmastro, M.C. Bottero, F. Dell'Anna, A. Acquaviva, L. Bottaccioli, E. Patti, E. Bompard, E. Pons, A. Estebsari, V. Verda, M. Santarelli, P. Leone, A. Lanzini, The Energy Center

Initiative at Politecnico di Torino: Practical experiences on energy efficiency measures in the municipality of Torino. In: INTERNATIONAL JOURNAL OF HEAT AND TECHNOLOGY, vol. 35 n. Specia, pp. 1-9. - ISSN 0392-8764

J.4 Alberto Bocca, Lorenzo Bottaccioli, Eliodoro Chiavazzo, Matteo Fasano, Alberto Macii, Pietro Asinari, Estimating photovoltaic energy potential from a minimal set of randomly sampled data, Renewable Energy, Volume 97, November 2016, Pages 457-467, ISSN 0960-1481

Peer reviewed International Conference:

IC.1 Amos Ronzino, Anna Osello, Edoardo Patti, Lorenzo Bottaccioli, Chiara Danna, Andrea Lingua, Andrea Acquaviva, Enrico Macii, Michelangelo Grosso, Gianluca Messina, Gaetano Rasconà; The energy efficiency management at urban scale by means of integrated modelling. In: Proceeding of 7th International Conference on Sustainability in Energy and Buildings (SEB-15), Elsevier

IC.2 Lorenzo Bottaccioli, Edoardo Patti, Andrea Acquaviva, Enrico Macii, Jarre Matteo, Noussan Michel; A tool-chain to foster a new business model for photovoltaic systems integration exploiting an Energy Community approach. In: Procideings of 20th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2015), Luxembourg; 09/2015, IEEE.

IC.3 Andrea Acquaviva, Daniele Apiletti, Antonio Attanasio, Elena Baralis, Federico Boni Castagnetti, Lorenzo Bottaccioli, Tania Cerquitelli, Silvia Chiusano, Enrico Macii, Dario Martellacci, Edoardo Patti; Energy Signature Analysis: Knowledge at Your Fingertips. In: Big Data (BigData Congress), 2015 IEEE International Congress on. IEEE, 2015. p. 543-550.

IC.4 Lorenzo Bottaccioli, Edoardo Patti, Michelangelo Grosso, Gaetano Rasconà, Angelo Marotta, Salvatore Rinaudo, Andrea Acquaviva, Enrico Macii; Distributed software infrastructure for evaluating the integration of photovoltaic systems in urban districts. In: 5th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS 2016), Rome, Italy, 23-25 April, 2016. pp. 357-362

IC.5 Lorenzo Bottaccioli, Abouzar Estebsari, Edoardo Patti, Enrico Pons, Andrea Acquaviva; A novel integrated real-time simulation platform for assessing PV penetration impacts in smart grids. In: Proceeding of 8th International Conference on Sustainability in Energy and Buildings (SEB-16), Elsevier Turin, Italy, 11-13 September 2016. pp. 780-789

IC.6 Lorenzo Bottaccioli, Enrico Macii, Edoardo Patti, Abouzar Estebsari, Enrico Pons, Andrea Acquaviva; PVInGrid: A Distributed Infrastructure for evaluating the integration of Photovoltaic systems in Smart Grid. In: 8th Advanced Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS 2017), Caprica (Lisbon), Portugal, 03-05 May 2017

IC.7 Lorenzo Bottaccioli, Alessandro Aliberti, Francesca Maria Ugliotti, Anna Osello, Enrico Macii, Edoardo Patti, Andrea Acquaviva; Building energy modelling and monitoring by integration of IoT devices and Building Information Models. In: Building Digital Autonomy for a Sustainable World (COMPSAC), Torino, Italy, 4-8 July 2017.

IC.8 Sara Vinco, Lorenzo Bottaccioli, Edoardo Patti, Andrea Acquaviva, Enrico Macii and Massimo Poncino; GIS-Based Optimal Photovoltaic Panel Floorplanning for Residential Installations, DATE-2018, Dresden .

IC.9 Sara Vinco, Lorenzo Bottaccioli, Edoardo Patti, Andrea Acquaviva, Massimo Poncino; A Compact PV Panel Model for Cyber-Physical Systems in Smart Cities. In: IEEE International Symposium on Circuits and Systems (ISCAS), Firenze, 27-30/05/2018. pp. 1-5 (2018)

IC.10 Lorenzo Bottaccioli, Edoardo Patti, Enrico Macii, Andrea Acquaviva; Distributed Infrastructure for Multi-Energy-Systems Modelling and Co-simulation in Urban Districts. In: 7th Conference on Smart Cities and Green ICT Systems (SMARTGREENS 2018), Funchal, Madeira, Portugal, 16 - 18 March 2018.

IC.11 Alessandro Aliberti, Lorenzo Bottaccioli, Giansalvo Cirrincione, Enrico Macii, Andrea Acquaviva, Edoardo Patti; Forecasting short-term solar radiation for photovoltaic energy predictions. In: 7th Conference on Smart Cities and Green ICT Systems (SMARTGREENS 2018), Funchal, Madeira, Portugal, 16 - 18 March 2018.

Peer reviewed National Conference:

NC.1 Edoardo Patti, Lorenzo Bottaccioli, Enrico Pons, Dario Martellacci, Federico Boni Castagnetti,Fabio L. Bellifemine, Andrea Acquaviva, Enrico Macii; Flexible smart metering for multiple energy vectors with active prosumers. In: CINI Annual Workshop on ICT for Smart Cities and Communities (I-Cities 2015), Palermo; 10/2015