UNIVERSIDAD DISTRITAL "FRANCISCO JOSÉ DE CALDAS"
FACULTY OF ENGINEERING
DOCTORATE IN ENGINEERING
EMPHASIS ON ELECTRICAL AND ELECTRONIC ENGINEERING

AUTONOMOUS CLOUD-OPERATED ENERGY MANAGEMENT SYSTEM TO OPTIMIZE THE ADMINISTRATION OF A CLUSTER OF MICROGRIDS

AUTHOR
MSC. ING. DAVID GUSTAVO ROSERO BERNAL

DIRECTOR OF THE DOCTORAL RESEARCH PROJECT
NELSON LEONARDO DIAZ ALDANA, Ph.D.
CO-DIRECTOR OF THE DOCTORAL RESEARCH PROJECT
CESAR LEONARDO TRUJILLO RODRIGUEZ, Ph. D.

FEBRUARY 2023

DOCTORATE COMMISION

Doctoral defense date: _____

	Approved
	Not approved

Name	Signature

ABSTRACT

Organizing the generation, storage, and management of electrical energy from the perspective of renewable energies, as well as the parameterization of the energy consumption characteristics of communities with limited access to the interconnected electricity supply, has taken more relevance in recent years due the demands that define the social welfare of this century. Complementary to the demand increase, other factors require the improvement and updating of the utility grid infrastructure and its opening to other technologies that meet the needs of end users. The interest in renewable energy sources, the evolution of energy storage technologies, the continuous research in microgrid management systems, and the massification of technologies and tools available in cloud computing, machine learning, big data, and the internet of things environment motivated the development of this doctoral research.

This doctoral research focuses on three tasks linked to the operation of a cluster of microgrids. The first task is the fluctuating integration of heterogeneous energy generation devices and objects whose mobility and distribution characteristics are particular over various geographical areas. The second is the need for real-time operation and extensive information processing and storage capabilities. Finally, the third task considers the conservation factors for critical applications linked to advanced optimization techniques, especially the operational cost and the battery's lifespan. An autonomous and scalable energy management model that follows the hierarchical control structure and bases its operation on cloud computing, the internet of things, machine learning, and big data solves the aforementioned tasks.

This research defines the elements considered by the real-time autonomous and scalable energy management system framework in a cluster of microgrids. For this, it is necessary to emulate the behavior of a group of interconnected microgrids and test the framework under real scenarios with the assistance of power-hardware-in-the-loop platforms connected to a cloud server. The server programming must include the implementation of the framework management protocol that exploits the optimization algorithm and state of charge equalization. Also, the framework takes advantage of machine learning and big data tools available in a cloud computing environment, ensuring the scalability of the framework's operation based on the fluctuation of the available resources in a microgrid or extending its operation to a cluster microgrids in a transparent way by the incorporation of IoT sensors or other tools. This doctoral thesis summarizes the framework research results and the published evidence released in one book, two journal papers, two international conferences, and one national conference.

KEYWORDS

Energy Management System, Internet of Things, Machine learning, Microgrid Cluster, Real-Life, Real-Time, Testbed.

ACKNOWLEDGEMENT

This document summarizes the Ph.D. project entitled AUTONOMOUS CLOUD-OPERATED ENERGY MANAGEMENT SYSTEM TO OPTIMIZE THE ADMINISTRATION OF A CLUSTER OF MICROGRIDS, developed at the Distrital University Francisco José de Caldas, under the supervision of Professor Nelson Díaz and Professor César Trujillo.

I like to express my sincere thanks to the Distrital University Francisco José de Caldas from Bogotá-Colombia, its Doctorate Program, and its staff. My gratitude to my director Nelson Díaz, Ph.D., and co-director, César Trujillo, Ph.D. Their supervision and guidance were essential in developing this project. Special gratitude to Fabio Andrade, Ph.D., Adriana Luna, Ph.D., M.Sc. Enrique Sanabria and the SEC team from the University of Puerto Rico. Great support and friendship when developing the internship process on a lovely island.

This work was developed under the research project "Cloud computing-based energy management for interoperability between a group of islanded microgrids" being part of the Program, Minciencias “Programa de Investigación en Tecnologías Emergentes para Microredes Eléctricas Inteligentes con Alta Penetración de Energías Renovables” Contract number 80740-542-2020.

Dynamic Defense Solutions SAS sponsors the development of this doctoral research.

DEDICATORY

To my beloved father, wherever you are, thank you.

To my mother, your unconditional support is essential in my life.

To my sons, most of the things that I do, I do for you.

To my wife, you are the base of most things I own. I love you.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Variables

$C_{i,t}$	Cost associated with the usage of energy	$p_{i,t}^{forecast}$	i-th IDG forecasted energy
C_i^{Batt}	Capacity in kWh	$p_{Grid,t}$	Energy exchanged with the utility grid
$E_{SoCi,t}$	Error between different SOCs in each i-th IDG at each time	$p_{RES_{nom_i}}$	Maximum power supplied by the RES
η_{ch_i}	Charge efficiency	$p_{d,t}^{smart}$	Load measured by the smart meter
η_{dis_i}	Discharge efficiency	$p_{RES_{i,t}}$	Energy generated by the i-th RES
$p_{Batt,ch}$	Charged ESS energy	$SoC_{i,t}^{min}$	Minimum SoC
$p_{Batt,disch_{i,t}}$	Discharged ESS energy	$SoC_{i,t}^{max}$	Maximum SoC
$p_{d,t}^{forecast}$	Forecasted load	$x_{bat,t}$	Binary variable

Abbreviations

BD	Big data	MGC	Microgrid cluster
BESS	Battery energy storage systems	MGCC	Microgrid Control Center
BRT	Bagged regression tree	MILP	Mixed Integer Linear Problem
CC	Cloud computing	ML	Machine learning
CREMS	Cloud-based real-time EMS	MPS	Matlab production server

DASP	Data acquisition, storage & processing
EC2	Elastic compute cloud
EPDP	Economic power dispatch problem
EMS	Energy management system
GPR	Gaussian process regression
HIL	Hardware-in-the-loop
PHIL	Power-Hardware-in-the-loop
IDG	Inverter-based distributed generator
IoT	Internet of things
JSON	JavaScript Object Notation
LP	Learning phase
LR	Linear regression
MAPE	Mean Absolute Percent Error
MG	Microgrid
MQTT	MQ telemetry transport
NN	Neural networks
NSRDB	National solar radiation database
PP	Preliminary phase
RES	Renewable energy sources
RT	Regression trees
S&O	Scheduling and optimization
S3	Simple storage service
SEC	Sustainable Energy Center
SL	Supervised learning
SoC	State of charge
SS	Supervisory system
SVM	Support vector machines
UDP	User datagram protocol

1. Introduction

The microgrid (MG) concept is a systematic approach for dealing with the distribution grid challenges, and this is considered an essential component of the new Smart Grids [1]. The successful implementation of policies to stimulate the integration of Renewable Energy Sources (RES) and the decreasing cost of Battery Energy Storage Systems (BESS) [2] have boosted the role of new energy system components that includes the concepts of Microgrids, Clusters, Hierarchical Control and Energy Management System. Integrating these concepts, a Microgrid Cluster (MGC) defines a collection of microgrids with capacities for energy generation, load control, and autonomous operation. The MGC will be able to configure new types of local energy markets and energy systems [2]. In this context, efficiency is the main objective for the planning, design, and operation of the MGC [3], where operation cost minimization is an essential control task [4].

Conventionally, the control approach to managing the power-sharing within the microgrids uses a droop control technique [5], [6] under safe operating conditions discarding economic operation criteria. In this research, the proposed energy management system (EMS) joins up maximum capacity generation, power consumption, and optimal economic operation under a machine learning (ML) perspective running in real-time, using cloud [7] resources, and supporting its functionality on the Internet of things (IoT). There is a need to improve the method to coordinate the generation, storage, and management of data coming from distributed RES. It is also relevant to determine the procedure of incorporating ML and IoT tools to predict energy generation capabilities and load consumption features of the related areas of interest. These methodologies have become more relevant in recent years to improve the requirement that defines the social welfare of this century [8]. In addition to the demand increase, other factors require improvement, continuous upgrade of the utility grid and openness to other technologies that meet the end-user needs and energy markets. The relevant interest in RES [9], [10], the evolution of BESS, the continuous research in EMS, the dissemination of cloud-computing (CC), the improvement of ML capabilities, the new availability of IoT sensors, the possibility of the real-time analysis and processing for large amounts of data and the need to improve the security and efficiency of the systems, motivates this doctoral research suggesting a complementary interaction with the technology available nowadays.

Several initiatives are referent of this doctoral research and support its development: Firstly, the initiatives of the Aalborg University in Denmark, which plays a consistent role in the development of autonomous microgrids allowing safe and efficient distribution of sustainable energy [11], the contribution of the Distrital University “Francisco José de Caldas” in Colombia, who continuously leads its application in the Colombian market [12],

the leadership of the Puerto Rico University Recinto Mayagüez, who constantly invest in its intelligent microgrid lab [13]. Secondly, the investigations of private organizations such as Navigant Research [14] focus on the coverage of microgrids, nanogrids, virtual power plants, and direct current distribution networks and associate them with a complex evaluation of deployment strategies, enabling technologies, regulatory factors, financial and business models and applications of these grids in vertical industries. Finally, efforts made by international financial organizations such as the IADB [15] and governmental organizations such as UPME [16], through technical cooperation, defined the most appropriate framework to carry out the implementation of microgrids in Colombia [17].

As stated in [18], from the scientific point of view, it is interesting to know the context and significance of CC, ML, and IoT to the value of the research, theories, approaches, and applicability in a cloud-based real-time EMS (CREMS) applied to an MGC; there may be gaps between the theoretical results, though limited, and the empirical verification or, if desired, between its degree of applicability and the actual result. Tasks related to the MGC operation, such as consumer consumption forecast, resource allocation prediction, advanced optimization techniques, power flow analysis, and core knowledge, among other activities, could be performed by a self-trained cloud-based real-time EMS model with a hierarchical and scalable structure [19]. Despite the above, and considering the consulted, not many studies verifies the advantages and benefits of using a cloud-based real-time EMS under an ML and IoT perspective.

1.1. Problem statement

Many questions arise about MGC management and the association of ML and IoT technologies with the proposed CREMS. Below, this document presents various components of this doctoral research that support its relevance: justification, problem formulation, major inquiry, objectives, methodological framework and hypothesis.

1.2. Justification

All the technological advances defined for ML, CC, IoT, dedicated systems, software, and network technologies have contributed positively to the emergence of many intelligent applications. These new technologies offer an unprecedented opportunity to develop a wide range of disruptive applications that integrate various physical objects and devices incorporated under the term Cloud of Things (CoT) [20].

In the CoT, applications span multiple domains. Sensors, actuators, vehicles, buildings, electronic commerce, logistics management, banking, health, social networks, and other

physical objects communicate through a processing systems network under a CC environment that operates intelligent software and self-managing applications. The processing service delivery performs as the public service, where service providers take responsibility for building the IT infrastructure, and end users use the service through an on-demand service [21].

CoT processing is demanding due to many reasons and has been analyzed from different perspectives because it favors:

- The use of a considerable number of devices and heterogeneous objects.
- The use of highly mobile devices.
- The distribution of components over large geographical areas.
- Operation in challenging areas such as forests, deserts, mountains, or underwater locations.
- Real-time support.
- The processing capacity and extensive storage of information, as well as its scalability.
- Consideration of critical applications that need to be protected.
- The management of advanced optimization techniques that improve the operation of the systems.

Due to this, and seeking to take advantage of these CoT potentialities in microgrid management systems, it is necessary to identify the essential elements of a scalable and autonomous architecture of the energy management system to manage a cluster of microgrids under a CoT environment. At the same time, after considering that the MGC could accommodate several resources with heterogeneous characteristics, it is necessary to coordinate the registration, storage, and processing of a considerable amount of information. This attribute facilitates the parameterization of the determining elements in the performance of MGC, such as energy demand, generation, availability, location and number of integrated resources, optimization adapted to the system, energy flow analysis, application of economies of scale, and consideration of maintenance cycles. Additionally, it is necessary to consider that the MGC will have to support exponential growth regarding energy demand from many devices and users depending on the application's influence area. This situation constantly forces adequate management of all the resources associated with the MGC to promote the minimization of administration costs.

1.3. Problem formulation

According to the justification presented and the revised information, it is necessary to propose an energy management system framework that contemplates the possibility of managing a cluster of interconnected microgrids in an autonomous and scalable manner and, in parallel, tends to optimize the administration through a real-time cloud-based operation.

1.4. Mayor inquiry

What elements should incorporate the automated and scalable real-time cloud-based energy management system framework to ensure the appropriate administration of a cluster of microgrids?

1.5. Objectives

After considering the precision of the EMS components and carrying out a detailed analysis of the possible strategies to be incorporated into its development, this doctoral research considers the following objectives:

1.5.1. General objective

Develop a scalable and autonomous framework of the EMS to administer an MGC under a CC environment.

1.5.2. Specific objectives

1. Generate an MG model for complex electrical systems supported on a real-time power-hardware-in-the-loop platform.

2. Implement a cloud-based autonomous management protocol in the EMS of an MG.

3. Verify the management framework scalability considering the fluctuation of the resources available in the MG and the MGC.

4. Validate the performance of the defined framework according to the delineated by the IEEE STD 2030.7 - 2017 [22] and other research results that consider the management of an MGC using a local server.

1.6. Methodological framework

The methodological framework of this doctoral research contemplates guiding the theoretical position toward specific procedures to achieve the objectives previously mentioned.

1.6.1. Methodological design

This research uses a quantitative methodological approach to achieve the objectives. This methodology makes it possible to explain the different technical-dynamic scenarios in which the management framework should run to operate in different MG topologies and how to incorporate it into a CC environment following the provisions of the IEEE 2030.7-2017 standard that specifies the most relevant properties in the management of MG.

For this purpose, it is necessary to incorporate the parameterized MG model into the simulation platform of complex electrical systems available in the laboratory of the Universidad Distrital - OPAL RT [23], improve the management protocol from the cloud, and scale it to an MGC in a controlled laboratory environment. In this way, the research process guarantees rigor, authenticity, and validity. In addition, it satisfies various criteria: integrity - since the management of an MGC includes the CC application, applicability - extending the use of the proposed model to different MG topologies, consistency - allowing the adjustment of available resources in the cloud according to the needs of the application, and neutrality - guaranteeing the objectivity of the results.

1.6.2. Design of experiments

This doctoral research considers three phases. Each phase presents the activities considered to achieve the assigned objective that allows empirical verification and validation of the proposed model.

1.6.2.1. First phase (permanent): documentary review

A bibliographic and documentary review of the relevant information related to the elements that constitute the different MG and their integration in the cloud is performed, classifying them according to their dynamic characteristics that determine considerations for their management and operation. The following highlights are used in this research: Connection topologies of the energy conversion equipment, mechanisms for operation in network and island mode, resource management mechanisms, strategies or techniques at the different

control levels, coordination between the control levels, coordination between MG, and implementation of routing and new technologies.

This phase allows us to propose the different technical-dynamic scenarios in which the management framework must be developed, operated in different topologies of MG, and incorporated into a CC environment.

1.6.2.2. Second phase: obtaining the architecture, simulations, and initial implementation

From the previous phase, the obtained scenarios define the technical-dynamic framework characteristics necessary for the different MG topologies implementation. In addition, this process allows the characterization of the required simulation and implementation tools where the operation of the proposed architecture can perform. The framework, simulation, and implementation tools definition, according to the established technical-dynamic scenarios, allow us to complete the operation, integration, and response analysis in both the MG and the MGC mode. Supported by the results obtained in this phase, it is necessary to adjust the framework along with its characterization, limitations, restrictions, and scope.

Next, the first physical implementation of the framework runs in the "MG Prototype at the District University Francisco José de Caldas PME-UD OPAL-RT " approved by the CIDC to the LIFAE and GECEM research group. This process considers the MG technical-dynamic conditions review and purchasing of the necessary elements and equipment to condition the implementation of the control, operation, coordination, and integrated system of the PME-UD. This action permits the framework operation validation at its isolated and integrated levels, under safe operating conditions, integrating the controllers, the energy conversion equipment, the test loads, and the coordination of the MG resources and their comparison to the established in the IEEE standard defined for the MG control. In this phase, the need to propose technical and electronic changes in the controllers and the conventional energy converter devices takes place so that they respond to the cloud environment framework.

1.6.2.3. Third phase: final implementation

This last phase focuses on testing the framework in a real-life power-hardware-in-the-loop environment. The doctoral internship contemplated this process at the Sustainable Energy Center at the University of Puerto Rico - Recinto Universitario Mayaguez (SEC) [24]. This phase develops through the following stages:

Control verification stage in a CC environment in grid-connected operation mode. This stage regards verifying the power quality and stability requirements under centralized cloud control in a grid-connected condition. In this stage, the controller reacts to disturbances and adjusts the active rejection to them. The cloud-deployment system using Amazon Web Services (AWS) [25] verifies the low-power, control, and operation signals. This system monitors the controllers' signal indicators. It is necessary to mention that the selection of this cloud service provider originates from the maturity exhibited by its platform and the compatibility with the components associated with MATLAB.

Control verification stage in a CC environment in island operation mode. This stage consists of verifying the same power quality and stability requirements under centralized control in the cloud at each control point, but now the verification performs when the microgrid operates in island mode. This stage tests the controllers' response to changes in load and induced disturbances.

Control verification stage for coordination in a scalable CC environment. This stage regards verifying the coordination between the previously adjusted control levels that operate in either *grid-connected* or island mode but, at this time, change dynamically compared to the setpoint values resulting in a scalable response. The cloud-based EMS interacts with a power-hardware-in-the-loop platform and integrates IoT tools by using JSON [26] and MQTT [27] protocols which permit the validation of the system operation in real-time concerning the behavior of the energy market and other reference variables. This stage contemplates the operational extension to multiple MGs.

Project documentation and disclosure stage. Finally, the collected results are analyzed to formulate recommendations and conclusions in this thesis. Likewise, this document summarizes the research results and the published evidence consigned in one book, two journal papers, two international conferences, and one national conference.

1.7. Hypothesis

The proposal of a management framework for the energy management system in a cloud computing environment for a cluster of microgrids will allow additional automation and scalability mechanisms for energy generation and storage resources improvement.

2. Research chapters

An innovative MG's EMS demands many features under a hierarchical structure perspective. This doctoral thesis chapter revises most of them before integrating and deploying the proposed CREMS architecture in a real-life scenario to verify its performance. The chapter includes theoretical sustentation for the ML and simulation results to validate its procedure. It also covers the mathematical model for solar PV, wind energy, and the BESS systems. The chapter introduces the implemented framework and describes its control. Next, section 2.9 summarizes the CREMS framework and depicts its appeals and differentiators. Finally, the last section of the chapter illustrates the performance verification of the CREMS framework under two optimization perspectives.

2.1. Machine learning

2.1.1. Section introduction

The results presented in this section are available in Applied Energy in the following paper [28], [29].

The essential concept in ML [30] is to estimate a set of parameters that describe the model using the available data and to make predictions based on low-level information and signals. It is possible to argue that there are not much intelligence built-in such approaches, but it is an acceptable approach to optimize the representation of the low-level input information to the computer. The representation term refers to how related information hidden in the input data is quantified/coded to allow a computer to process it. In more technical terminology, each piece of such information determines a feature, irradiance, wind speed, and load energy, in this case. This section's outcomes will facilitate the inclusion of autonomous capabilities in the EMS as defined in the second specific objective.

2.1.2. ML contextualization

It is necessary to emphasize that data lie on the heart of ML systems. Data are the beginning. It is the information hidden in the data, as underlying regularities, correlations, or structure, which a machine learning system tries to “learn.” Thus, regardless of how intelligent a software algorithm suggests, it cannot learn more than the data used for training. Depending on the information type needed considering a specific task, different types of ML are available [30], [31]:

2.1.2.1. Supervised learning

Supervised learning (SL) refers to the type of ML where all the available data have a label. All the data separates into a pair of observations, e.g., $(y_n, \boldsymbol{x}_n)$, $n = 1, 2, N$, where each $\boldsymbol{x}_n$ is a vector or, in general, a set of variables. The variables in $\boldsymbol{x}_n$ are called the input variables, also known as the independent variables or features, and the respective vector is known as the feature vector. The variables y_n represent the output or dependent or target or label variables. In some cases, y_n can also be a vector. The objective of learning is to obtain/estimate a functional mapping to, given the value of the input variables, predict the value of the respective output one. Two main tasks of SL are the classification and the regression tasks.

2.1.2.1.1. Classification

The goal in classification is to assign a pattern to one of a set of possible classes whose number could be known. The first step in designing any ML task is to decide how to represent each pattern in the computer. The preprocessing stage achieves this by encoding related information that resides in the raw data in an efficient and information-rich way.

It is usually done by transforming the raw data into a new space and representing each pattern by a vector, $\boldsymbol{x} \in R^l$, which comprises the feature vector and its l feature values. Each arrangement becomes a unique point in an l-dimensional space, the input space. The raw data transformation is the feature extraction stage. The stage starts with generating some extensive value, K, of possible features and eventually selects the l most informative ones via an optimizing procedure known as the feature selection stage.

After deciding the input space for representing the data, the system has to train a classifier, which is a predictor. This classifier evolves by selecting a class-known set with N data points/samples/examples that comprises the training set. It defines the set of observation pairs, $(y_n, \boldsymbol{x}_n)$, $n = 1, 2, \ldots, N$, where y_n is the (output) variable denoting the class in which $\boldsymbol{x}_n$ belongs, and it is known as the corresponding class label. Based on the training data, the system designs a function, f, which is used to predict the output label, given the input feature vector, $\boldsymbol{x}$.

Once the function, f, has been designed, the system is ready to make predictions. Given a pattern whose class is unknown, the raw data helps to obtain the corresponding feature vector, $\boldsymbol{x}$. Depending on the value of $f(\boldsymbol{x})$, the pattern classifies into one of the two classes. If the function f is linear (nonlinear), the respective classification task is linear (nonlinear). For a given set of points, each representing a pattern in the two-dimensional space (two

features used, x_1, x_2). For instance, stars belong to one class, and the crosses to the other, in a two-class classification task. These are the training points used to obtain a classifier. For this case, a linear function achieves this,

$$f(x) = \theta_1 x_1 + \theta_2 x_2 + \theta_0 \quad (1)$$

whose graph, for all the points such that $f(\boldsymbol{x}) = 0$, is the straight line. The values of the parameters θ_1, θ_2, and θ_0 came from an estimation method based on the training set. This phase, where a classifier is estimated, is also known as the training or learning phase.

Once a classifier has been "learned," the system is ready to perform predictions, that is, to predict the class label of a pattern **x**. The discussion about the classification task considers features that take numeric values. Classification tasks where they are of categorical type "do" exist and are of considerable importance, too.

2.1.2.1.2. Regression

Regression shares to a large extent the feature generation/selection stage, as described before; however, now the output variable, y, is not discrete, but it takes values in an interval in the real axis or a region in the complex numbers' plane. Generalizations to vector-valued outputs are possible, but this research concentrates on real variables. The regression task is a function (curve/surface) fitting problem.

For a set of training samples, $(y_n, \boldsymbol{x}_n)$, $y_n \in R$, $\boldsymbol{x}_n \in R^l$, $n = 1, 2, \ldots, N$, the task is to estimate a function f, whose graph fits the data. After having this function, when a new sample $\boldsymbol{x}$, outside the training set, arrives, the system can predict its output value.

2.1.2.2. Unsupervised learning

In the antipode of SL lies unsupervised learning (UL), where the known variables are the input ones. No output or label information is available. UL aims to unravel the structure that underlies the given data set. It is a relevant part of data learning methods but this doctoral research does not consider this part of the ML.

UL comes under several facets. One of the most critical types of UL is clustering. The goal of any clustering task is to unravel how the points in a data set are grouped, assuming that such a group structure exists. At the heart of any clustering algorithm lies the similarity concept since patterns that belong to the same group (cluster) are assumed to be more

similar than patterns that belong to different ones. One of the most classical clustering schemes is the so-called k-means clustering. However, clustering is not the main topic of this thesis, and the interested reader may look at more specialized references [32]. Another type of UL is dimensionality reduction. The goal is also to reveal a particular data structure different from the grouping structure. For instance, although a high-dimensional space represents the data, it may lie around a lower-dimensional subspace or a manifold. Such methods are relevant in ML for compressed representations or computational reduction reasons. [30] presents dimensionality reduction methods in detail. Probability distribution estimation can also be considered a particular case of UL.

2.1.2.3. Reinforcement learning

Finally, another type of learning, which is increasingly gaining importance, is the so-called reinforcement learning (RL). This is also an old field with origins in automatic control. At the heart of this learning type lies a set of rules, and the goal is to learn sequences of actions that will lead an agent to achieve its purpose or to maximize its objective function. RL attempts to determine the behavior by trial and error. In contrast to SL, optimal actions are not learned from labels but from what is known as a reward. This scalar value informs the system whether the outcome of whatever it did was right or wrong. Taking actions that maximize the award is the goal of RL.

2.1.3. Section conclusion

Based on the contextualization and the available data features, this doctoral research considers an SL model for this CREMS as it involves equivalent steps to the statistical modeling technique for developing, validating, and implementing the solution as stated in [29], [33] and following the established in the second specific objective.

2.2. SL model development and deployment

2.2.1. Section introduction

The results presented in this section are available in the IEEE in the following conference paper [29] and in the Editorial UD in the book [34].

The development and deployment of SL models involve a series of steps [33] that are almost similar to the statistical modeling process to develop, validate, and implement ML models:

1. Collection of data. Directly from structured source data, web scraping, API, and chat interaction, to mention a few.
2. Data preparation and missing/outlier treatment. Data formats as per the chosen ML algorithm. Additionally, missing value treatment performs by replacing missing and outlier values with the mean/median technique.
3. Data analysis. All data goes through scrutiny to find hidden patterns and relations between variables.
4. Train algorithm on training and validation data. Data divides into three chunks: train, validation, and test data. ML applies to train data, and the model's hyperparameters are tuned based on validation data to avoid overfitting.
5. Test the algorithm on test data. Once the model has shown good performance on train and validation data, its performance evaluates against unseen test data. The algorithm is ready for deployment if the performance is still good enough.
6. Deploy the algorithm. Deploy trained ML algorithms on live streaming data to classify the outcomes.

In this doctoral research, ML compiles the input-output variables interrelation, and the autonomous model uses empirical data to estimate the statistical dependencies of the mentioned variables. This probabilistic interpretation emerged because it is essential to gather usable patterns from actual logs, in special, the CREMS' non-existing connection with public irradiance and wind speed data, and measured energy utilization. This improves the framework's autonomy and adaptation to the resource fluctuations looking to satisfy specific objectives two and three.

2.2.2. The ML procedure

The computational techniques set and tools to support the modeling of a large number of data groups under the more general label of data science. This science deals with the idea of extracting knowledge from volumes of data and the autonomous design of models from them. ML [35] is postulated here as a problem of statistical estimation of the dependencies between variables based on empirical data. In this research, SL corresponds to modeling the relationship between a set of input variables and one or more output-dependent variables, as presented in figure 1. Correspondingly, the relevance of statistical analysis arises when there is a need to extract useful information from data records obtained by repeatedly measuring an observed phenomenon, in this case, the relationship between two quantitative variables, irradiance and wind speed, with a MG's energy generation. As there is not an adequate theoretical relationship between the variables, we have used a repeated measurement of them. There are two reasons for addressing this problem in this way: first, the more complex is the input/output relationship, the less effective will be the contribution

of a human expert in extracting a model of the relation, and second, data-driven modeling may be a valuable support for the designer. Likewise, the knowledge extracting process from observations lies at the root of modern scientific development, the notion of truth.

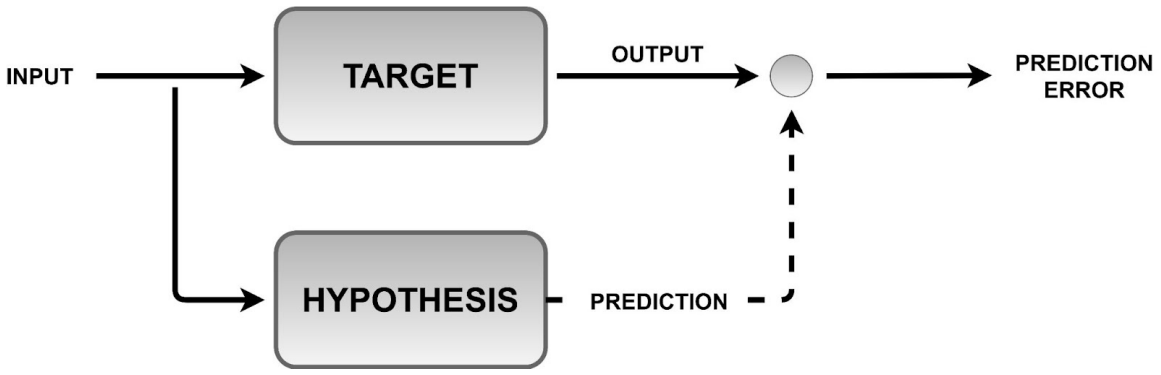


Figure 1. Supervised learning setting. Source: Authors

The modeling process consists of a preliminary phase (PP), and a learning phase (LP), as stated in figure 2. The first one, which brings the data from their original form to a structured configuration, called training set, can be decomposed in [35]:

- Problem formulation. Selection of a particular application domain, several descriptive variables, and hypothesization of the relation between the measurable variables.
- Experimental design. Delivery of a dataset made of samples that represent the phenomenon to maximize the performance of the model,
- Pre-processing [36]. Raw data cleaning actions like noise filtering, outlier removal, missing data treatment, and feature selection, to mention a few.

The LP begins with the training set, used by ML to formalize and optimize the procedure bringing data to a model and predictions. A LP procedure requires both a search space, which includes possible solutions, and an assessment criterion, which measures the quality of the solution to choose the best one [37]. The goal of learning is to attain good statistical generalization. This means that the selected model will return an accurate prediction of the dependent (output) variable when values of the independent (input) variable, which are not part of the training set, are presented. Any SL algorithm has two loops denoted as the structural loop and parametric loop. The first one seeks the model structure with the best accuracy by assessing each model structure in the validation phase and returning it to the selection phase. The second returns the best model for a fixed model structure. Parametric modeling is a theme that runs across the spine of this doctoral research. Various issues related to the parameter estimation task, such as regression, estimator efficiency,

bias-variance dilemma, ill-conditioning and overfitting, and the curse of dimensionality, are introduced and discussed.

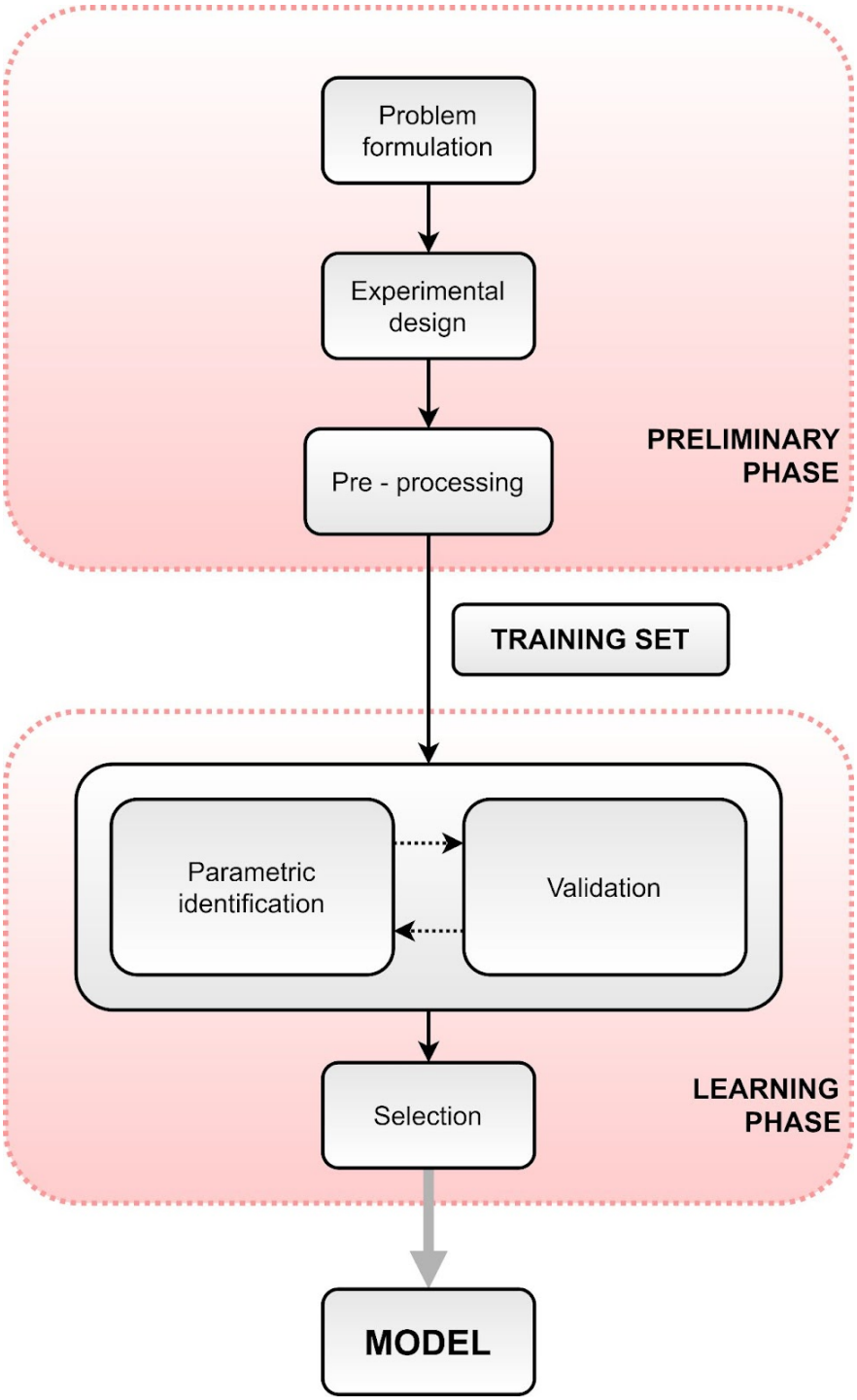


Figure 2. The modeling process. Source: Authors

2.2.2.1. Preliminary phase

In this research, the PP considers downloading, aggregating, and cleaning data for two cases: meteorological data, primarily irradiance and wind speed, and energy consumption. After having a clean version of the data, a dataset merging process completes steps 1 and 2 mentioned in section 2.2.1 - data preparation and missing/outlier treatment. The detailed algorithms used in each case are present hereafter [38] .

2.2.2.1.1. Model development - Downloading meteorological data

The scripts developed in Matlab [39] depend on the meteorological information available and the format obtainable. In this case, scripts will download the irradiance and wind data from the National Solar Radiation Database (NSRDB) [40]. The NSRDB is a serially complete collection of hourly and half-hourly values of meteorological data and the three most common measurements of solar radiation: global horizontal, direct normal, and diffuse horizontal irradiance. It covers the USA and a growing subset of international locations. These data have been collected at convenient locations with temporal and spatial scales to represent regional solar radiation. For a particular area covered by the dataset, it is possible to see the amount of solar energy in a given time and predict the potential future availability of solar energy based on past conditions. The algorithm (1) presents the detailed steps for meteorological data downloading.

Algorithm 1 DOWNLOAD METEOROLOGICAL DATA

1: **procedure** NSRDB download

2: **input** starting and ending points range to download per year

3: Create a range of months

4: If some files have already been downloaded, should they be redownloaded?

5: **if** redownloaded is not admitted **do**

6: get the months for existing zips

7: find which zips exist in the list to download the ones do not exist in zip directory

8: filter out dates that have already been downloaded

9: **end if**

10: Scrape NSRDB website and download meteorological data

11: **parfor** loop through dates to download in parallel **do**

12: name of the zip file

13: location of the zip file

14: destination on server

15: **end parfor**

16: Unzip files

17: Location to unzip contents to

18: Create directory if it does not exist

19: **parfor** loop through dates to unzip files in parallel **do**

20: unzip files on the server's selected location

21: **end parfor**

22: **end procedure**

2.2.2.1.2. Model development - Aggregating meteorological data

The script reads in the separate files for each day and combines them into a single table. As the amount of data could be massive, this script could take a few minutes (three minutes in this case). The algorithm (2) presents the detailed steps for the meteorological data aggregating.

Algorithm 2 AGGREGATE METEOROLOGICAL DATA

1: **procedure** NSRDB aggregate

2: **input** List the weather stations of interest and their ID

3: Select ID stations

4: Create table used for filtering out unwanted stations

5: Create datastore to access data

6: **for** stations available **do**

7: setup datastore options including file locations, variables to use, and how to read data

8: read all data

9: **end for**

10: Accommodate the table information creating a datetime variable

11: Create table “DataRaw” and bring irradiance and wind speed as main variables

12: Organize “DataRaw” with common timestamps on each row, columns containing irradiance, wind speed and other readings for each station in that time

13: Save aggregated data for later use

14: **end procedure**

2.2.2.1.3. Model development - Cleaning meteorological data

The development of this script focuses on getting the raw data aggregated in the previous step into a clean and usable form. The algorithm (3) presents the detailed steps for meteorological data cleaning.

Algorithm 3 CLEANING METEOROLOGICAL DATA

1: **procedure** NSRDB cleaning

2: **input** load in raw weather data

3: Use timetable function to associate every point with a timestamp

4: Use retime function to move the data into the time needed by using a linear interpolant

5: Use movmedian function to smooth the data behavior

6: Use ismissing function to find any missing values and plot the location of missing values by the weather station using a local function.

7: Use fillmissing function to interpolate the missing values

8: **if** station is within the convex hull of the data points **do**

9: Use scatteredInterpolant function to interpolate the data as Data = F(Latitude, Longitude, Time)

10: **end if**

11: Use machine learning to perform extrapolation

12: Save aggregated data for later use

13: **end procedure**

The figures presented next show a look at the data readings obtained after performing the cleaning process on the aggregated data. As a reference, figure 3 displays the location of the selected weather stations available at the NSRDB for Puerto Rico (PR). This selection considers the data availability and updates to proceed with this doctoral research. The figure shows five cities: San Juan (SJU), Aguadilla (AGU), Mayagüez (MAZ), Ponce (PON), and Fajardo (FAJ).

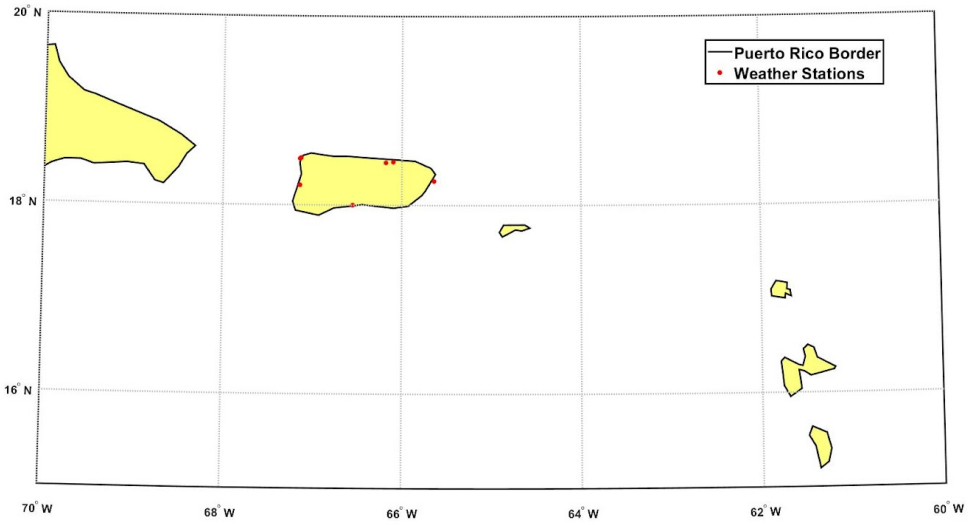


Figure 3. Data Readings - Location of the weather stations. Source: [29]

Figures 4 and 5 present the smoothing result for the irradiance and wind speed data in the PRMAZ station. The image exhibits a comparison between the measured data against the processed data, in this case, the smoothed data, and considering different time frames. The smoothing process is a powerful technique used across data analysis and responds to other names like *curve fitting* and *low pass filtering*. It helps to detect trends in the presence of noisy data in cases in which the shape of the tendency is unknown [41].

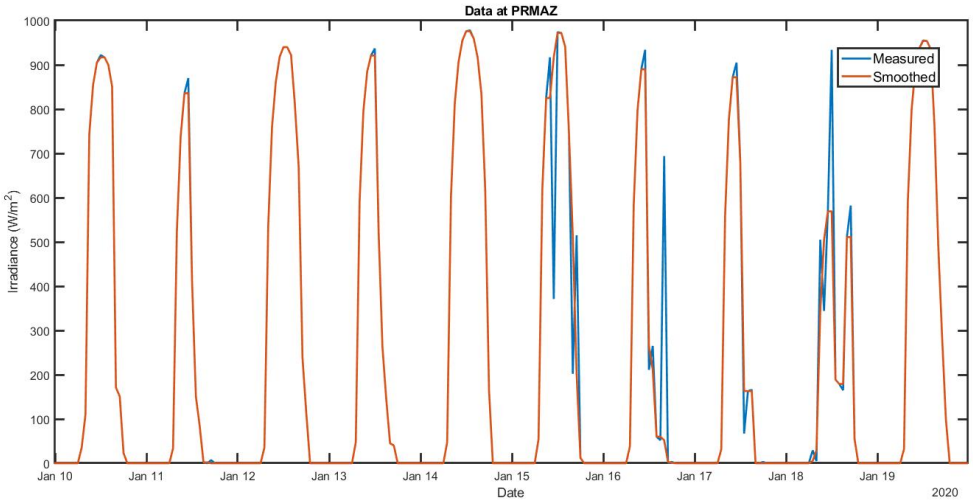


Figure 4. Data readings - Smoothed irradiance. Source: [29]

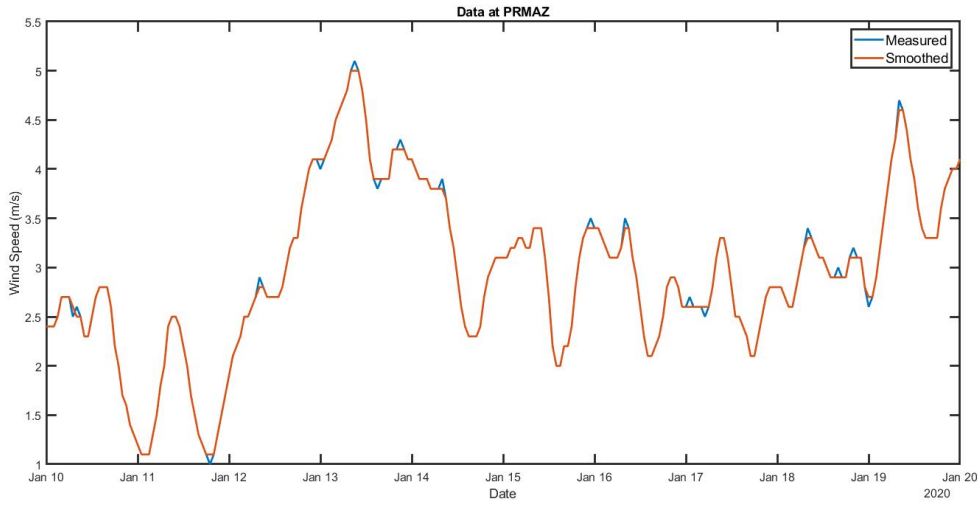


Figure 5. Data Readings - Smoothed wind speed. Source: Authors

Figure 6 shows the sparsity of the missing meteorological data. Some weather stations have a considerable number of small gaps, and it is necessary to handle them by utilizing a fill-missing algorithm to interpolate the missing values. In this case, the stations PRAGU and PRSJU exhibit a low percentage of missing data points (0.0%).

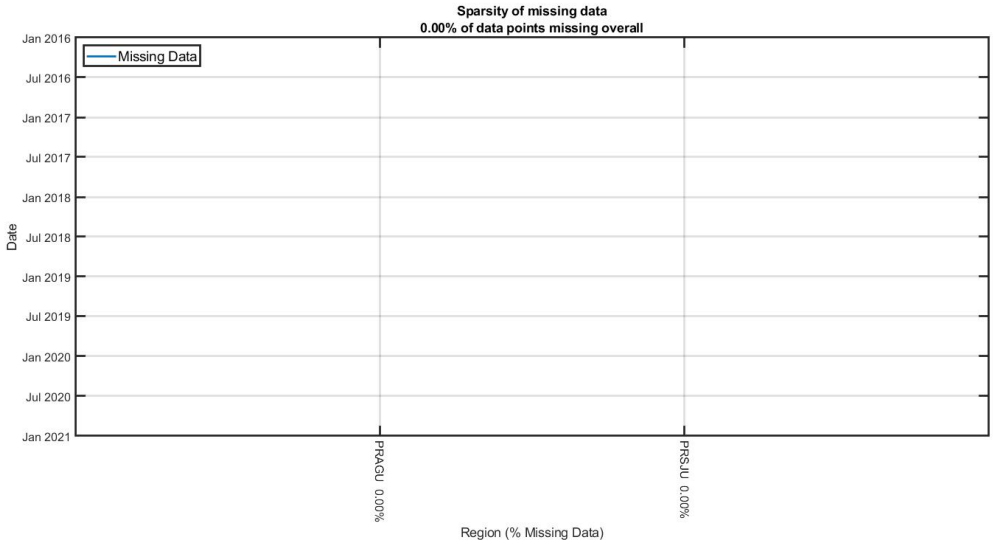


Figure 6. Data Readings - Missing data sparsity. Source: Authors

Figures 7, 8, and 9 show the results for interpolating missing values geographically. It is an alternative to fill-missing, which isolates some data to interpolate the missing values based on the weather stations around them. Using the station latitude, longitude, and timestamp at each point to interpolate based on the data available using the form Data = F(Latitude, Longitude, Time). The station must be within the convex hull of the data points obtainable to interpolate. If they are not within the convex hull, it is still possible to extrapolate, but the results may be poor. This is the case for the PRMAZ station (figure 7). It is necessary to extrapolate the results because the missing values aren't within the convex hull of the available data points, and the plot of the interpolated results (same timeframe) don't look that reasonable (figure 8) due to the peak presence. This means that extrapolation, its simple method, does not do well (same timeframe). To fill-missing more accurately, it is necessary to turn to more sophisticated techniques like ML or first principles modeling.

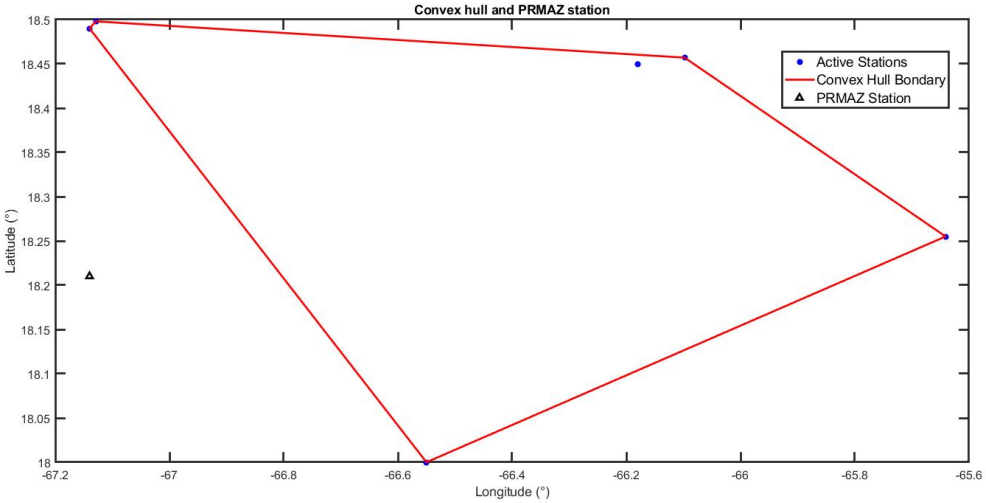


Figure 7. Data Readings - Convex hull of data points. Source: Authors

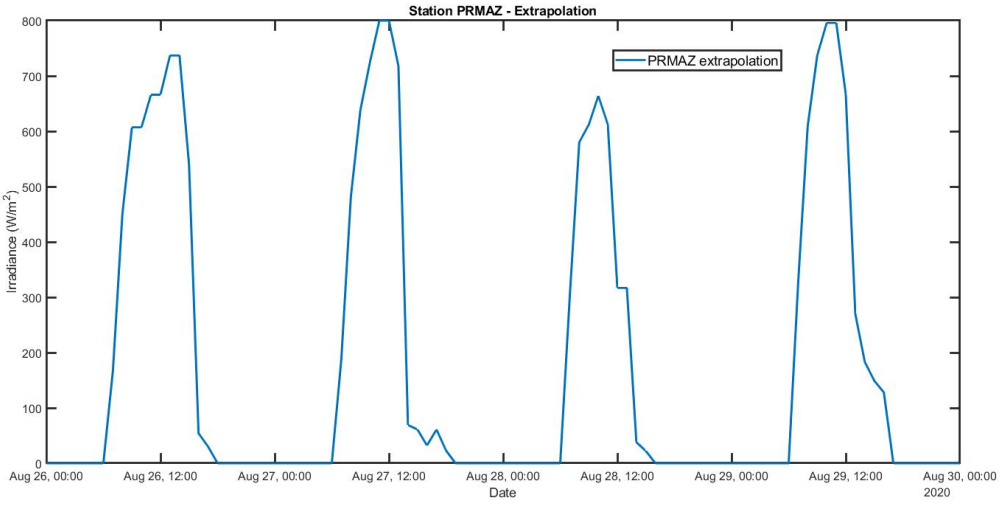


Figure 8. Data Readings - Irradiance data extrapolated for PRMAZ. Source: Authors

Figure 9 visualizes how the interpolation does on one of the missing areas for the PRAGU station. This station is within the convex hull, and considering the available data, the values for the PRAGU station after interpolation with the missing values perform reasonably.

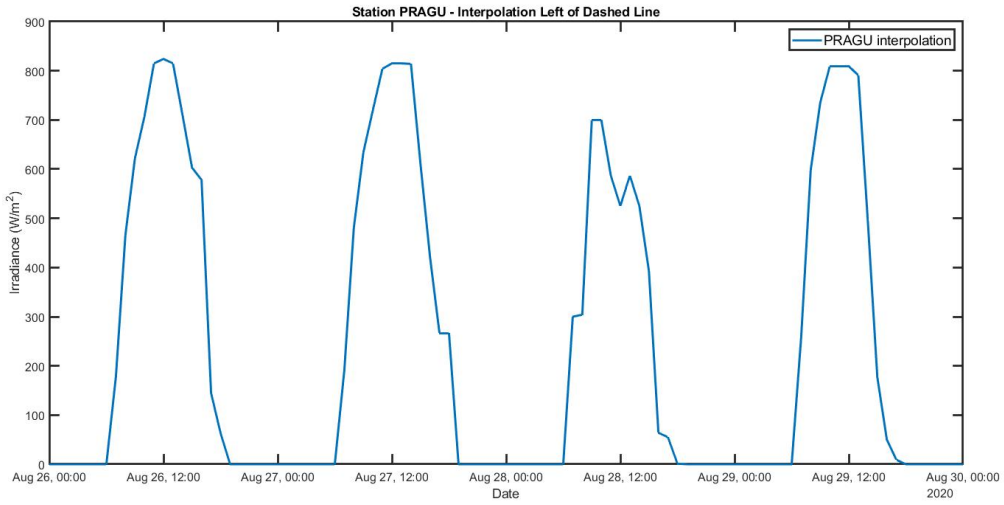


Figure 9. Data Readings - Irradiance data interpolated for PRAGU. Source: Authors

2.2.2.1.4. Model development - Downloading load energy data

The load energy data comes from the Emporia energy monitoring system [42] connected to residential houses in the Mayagüez area. The data collection started in March 2020 with 24 samples during the day. As the service provided by Emporia limits the access to the recollected data to one link sent to an email address, it is necessary to develop a script to centralize the downloaded information as a cloud object storage in an S3 bucket [43], facilitating real-time access to the data. The algorithms (4) and (5) present the detailed steps for writing and reading data from S3, respectively.

Algorithm 4 WRITING DATA TO S3

1: **procedure** Writing to S3

2: **input** S3 credentials and bucket configuration

3: Specify full path to the files or folders using a uniform resource locator

4: Write to S3

5: **end procedure**

Algorithm 5 READING DATA FROM S3

1: **procedure** Reading from S3

2: **input** S3 credentials and bucket configuration

3: Specify full path to the files or folders using a uniform resource locator

4: Read from S3

5: **end procedure**

2.2.2.1.5. Model development - Aggregating load energy data

This script will read in separate files for each day and combine them into a single table. Depending on the amount of data aggregated, the process could take a considerable time (10 minutes in this case). The algorithm (6) presents the detailed steps for load data aggregating.

Algorithm 6 AGGREGATE LOAD DATA

1: **procedure** LOAD aggregate

2: **input** Data collection

3: Create datastore to access data

4: Accommodate the table information creating a datetime variable

5: Create table “LoadRaw”

6: Organize “LoadRaw” with common timestamps on each row for each region

13: Save aggregated data for later use

14: **end procedure**

2.2.2.1.6. Model development - Cleaning load energy data

This script focuses on getting the raw data aggregated explained in the previous section in a clean and usable form. The algorithm (7) presents the detailed steps for the load data cleaning.

Algorithm 7 CLEANING LOAD DATA

1: **procedure** LOAD cleaning

2: **input** load in raw load data

3: Use timetable function to associate every point with a timestamp

4: Use retime function to move the data into the time needed by using a linear interpolant

5: Use movmedian function to smooth the data behavior

6: Use ismissing function to find any missing values

7: Use fillmissing function to interpolate the missing values

8: Save aggregated data for later use

9: **end procedure**

The figures presented below exhibit a look at the data readings obtained after performing the cleaning process on the aggregated load energy data. The energy monitoring systems identify according to the installation city and a sequence number. For instance, MAZ1 represents the first device installed in the first house in Mayagüez. Figure 10 displays an energy load data sample for the MAZ1 device.

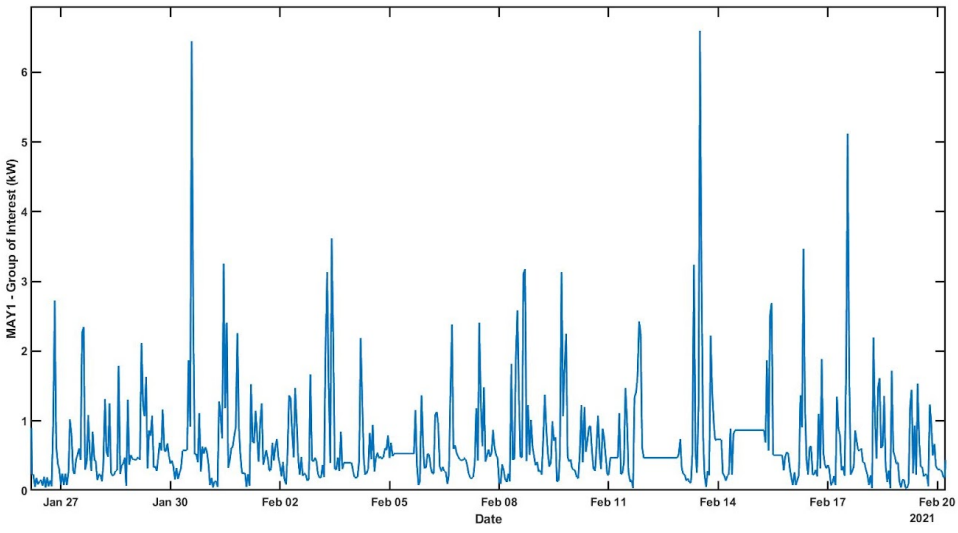


Figure 10. Data Readings - Load data raw. Source: Authors

Figure 11 shows the sparsity of the missing energy load data. Similarly to the weather case, some energy monitoring devices could have a considerable number of small gaps, and it is

necessary to handle them by utilizing a fill-missing algorithm to interpolate the missing values. In this case, the devices MAZ2 and MAZ3 display a low percentage of missing data points (0.0%).

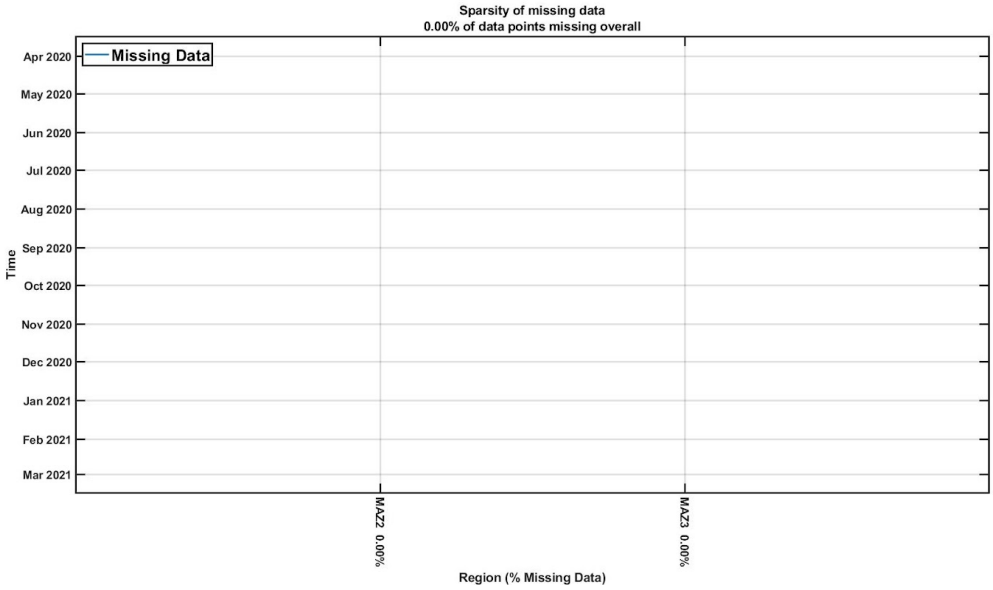


Figure 11. Data Readings - Missing load data sparsity. Source: Authors

Figure 12 shows a zoomed-in portion of the signal, with some of the spikes identified. This process uses a local find-spikes function that uses a distribution-based approach to the slopes to identify multipoint outliers. There are several positions where multipoint spikes in the signal appear.

It is necessary to consider some way to identify these automatically. There are many possibilities, including:

- Simple thresholding based on signal value or rate of change.
- Construct distributions of the values or slopes to recognize outliers in the distribution.
- Smoothing data with a spline.
- Find peaks, the Hampel filter, and other filters from Signal Processing.
- Training an ML model to recognize outliers

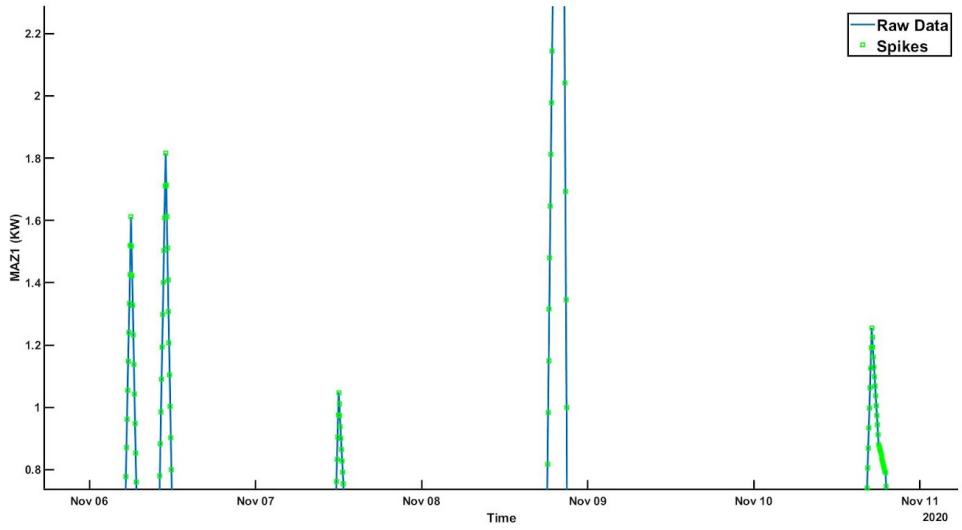


Figure 12. Data Readings - Spikes on load data. Source: Authors

Figure 13 presents the smoothing result for the energy load data in the MAZ1 device. The picture exhibits a comparison between the measured data against the processed data, in this case, the smoothed data, considering different time frames.

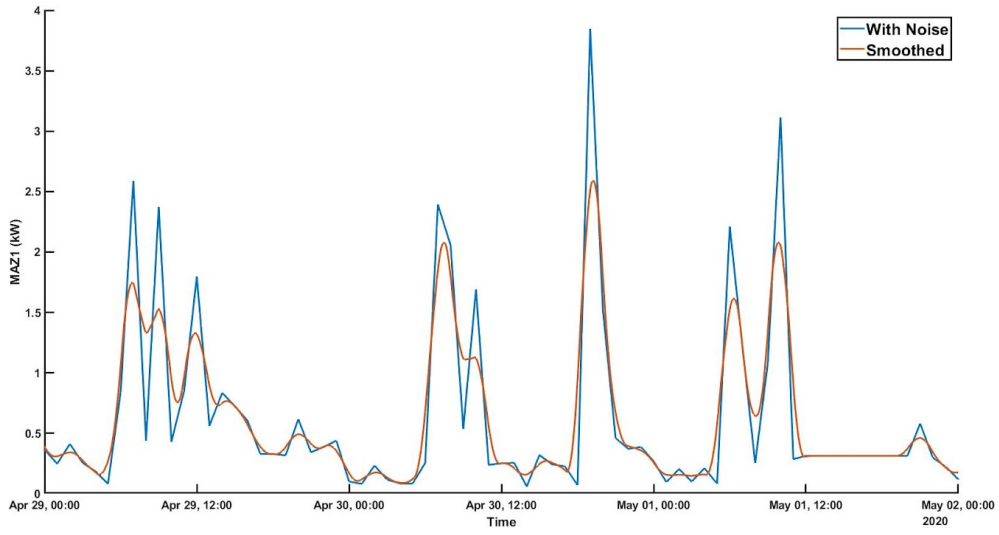


Figure 13. Data Readings - Smoothed load data. Source: [29]

2.2.2.1.7. Model development - Merging meteorological and load data

This script merges the weather and loads data sets together. The algorithm (8) presents the detailed steps for the meteorological and load-data merging.

Algorithm 8 MERGING METEOROLOGICAL AND LOAD DATA

```
1:  procedure DATA MERGING
2:  input load in cleaned load and meteorological data
3:  Revise whether the data set are sampled and sampling rates
4:  Synchronize timetables
5:  Double check
6:      if data is not sorted do
7:          sort data
8:      end if
9:      if exists duplicate do
10:         locate and eliminate duplicates
11:     end if
12:     if dates are missing do
13:         locate gaps
14:     end if
15: Save merged data
16: end procedure
```

2.2.2.2. Learning phase

The learning phase begins with the merged data set used by ML to formalize and optimize the procedure bringing data to a model and data to predictions.

2.2.2.2.1. Linear regression

Generally, the term regression defines the task of modeling the relationship of a dependent random variable y when a set of random variables x_1, x_2, … x_l, manage the relationship and considers an unobserved additive disturbance η. The goal of the regression task is to estimate the training data set or parameter vector $\boldsymbol{\theta}$, given a set of available measurements/observation (y_n, x_n), n = 1, 2, .., N. The dependent variable is known as the output variable and the vector **x** as the input vector and linearly combined is written as [35]:

$$y = \boldsymbol{\theta}_0 + \boldsymbol{\theta}_1 x_1 + \ldots + \boldsymbol{\theta}_1 x_1 + \boldsymbol{\eta} = \boldsymbol{\theta}_0 + \boldsymbol{\theta}^T \mathbf{x} + \boldsymbol{\eta} \quad (2)$$

The parameter θ_0 is known as the bias. The parameter vector θ usually absorbs it. This absorption implies a simultaneous increase in the dimension of **x** by adding the constant 1 as its last element. With this consideration, the regression model writes as:

$$y = \boldsymbol{\theta}^T \mathbf{x} + \boldsymbol{\eta} \quad (3)$$

Because the noise variable is unobserved, it is necessary to be able to predict the output value of y, given an observed value (x) of the random vector x. In linear regression, given an estimate $\widehat{\theta}$ of θ, it is possible to adopt the following prediction model

$$\widehat{y} = \widehat{\theta}_o + \widehat{\theta}_1 x_1 + \ldots + \widehat{\theta}_l x_l = \widehat{\theta}^T x \quad (4)$$

By using the squared error loss function, the estimate $\widehat{\theta}$ is set equal to θ, which minimizes the square difference $\widehat{y}_n$ and y_n over the set of variable observations, namely:

$$\left(\sum_{n=1}^{N} x_n x_n^T\right)\widehat{\theta} = \sum_{n=1}^{N} y_n x_n \quad (5)$$

Considering the input matrix X, defined as N x (l + 1) matrix, which has as rows the extended regressor vectors x_n^T, $n = 1, 2, \ldots, N$, equation (5) can be written as

$$\left(X^T X\right)\widehat{\theta} = X^T y \quad (6)$$

and, assuming that $\left(X^T X\right)^{-1}$ exists, the LS estimate is given by

$$\widehat{\theta} = \left(X^T X\right)^{-1} X^T y \quad (7)$$

In other words, a linear set of equations represents the obtained estimate of the parameter vector. This is a major advantage of the squared error loss function when applied to a linear model. Moreover, this solution is unique, provided that the (l + 1) x (l + 1) matrix $X^T X$ is invertible.

2.2.2.2.1.1. Model development - initial considerations

Using this script allows us to train different models and check its response. The algorithm (9) presents the detailed steps for the model development. The algorithm presented is general and applies to all the data sets available.

Algorithm 9 MODEL DEVELOPMENT

1: **procedure** MODEL development
2: **input** Merged data
3: Revise how similarly the different regions behave.
4: **if** regions don't behave similarly **do**
5: model each zone separately
6: **else** regions behave similarly **do**
7: create one model that understands all the zones
8: **end if**
9: Create temporal predictors
10: Transform 24 hours spot into a cyclical behavior
11: Define lagged predictors
12: Split into training and testing datasets
13: Train different models using Regression Learner App
14: Improve regression performance

15: Compare performance of the models on the test set

16: Compare the measure data with the 24 hour ahead prediction using Mean Absolute Percent Error (MAPE) metric

It is necessary to determine how to model the interest zone. There are two possibilities: create one model that understands all the zones or model each region separately. The selection depends on how similarly the different zones behave. If they exhibit similar behavior, it is possible to use one model. Figure 14 displays that the magnitudes and ranges presented by the energy demand are comparable in the stipulated zones, so one model considers all the zones.

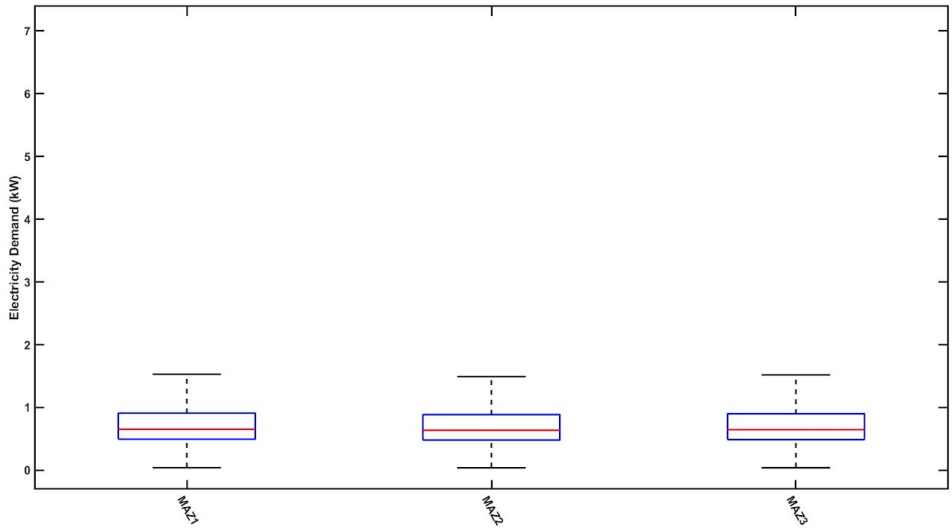


Figure 14. Data Readings - Similar behavior in zones. Source: Authors

Sometimes, the impact of a predictor included in a model is not simple and immediate. In some situations, it is necessary to allow lagged effects of the predictor. This allows the incorporation of changed amounts of recent history into the forecast [44] . Also, the lagging of independent variables is often necessary to activate the model's ability to predict the future - to predict what will happen in period *t* based on knowledge of what happened up to period *t-1*. In this case, the energy load data can work as a predictor. It is possible to use a traditional time series analysis, but considering the autocorrelation will show an interesting pattern. The peaks in the autocorrelation at the 24 and 168-hour lags suggest the utilization of lagged predictors of 1 and 7 days. Figure 15 exhibits this pattern.

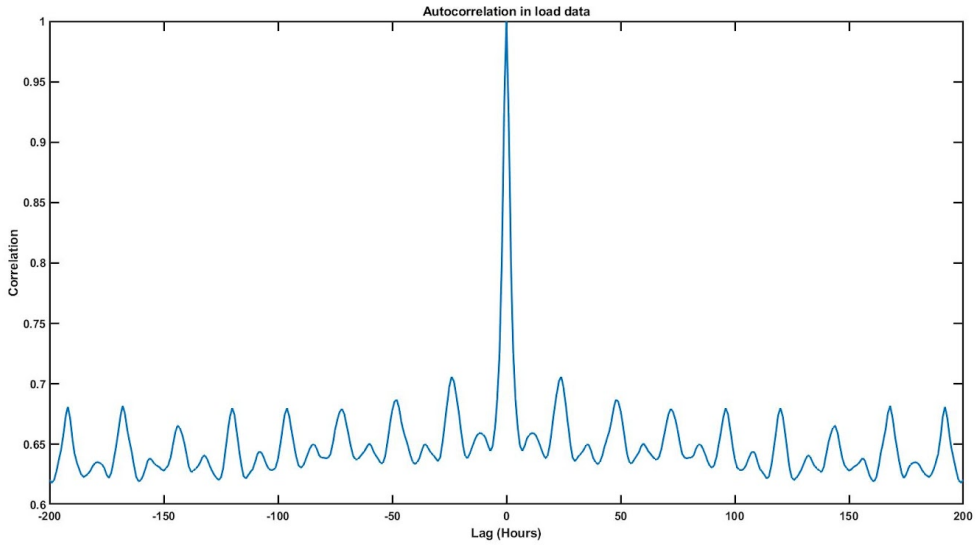


Figure 15. Data Readings - Lagged predictors of 1 and 7 days. Source: Authors

2.2.2.2.2. Biased and unbiased estimation

In supervised learning with a set of training points, $\left(y_n, x_n\right)$, $n = 1, 2, \ldots, N$, is possible to return an estimate of an unknown parameter vector $\widehat{\theta}$. However, as the training points are random variables, having another set of N different observations of the same random variables, the resulting estimate will also be different - changing the training data will change the result.

An estimate, such as $\widehat{\theta}$, has a specific value. This comes from a function acting on a set of observations on which the estimation depends [30]. A general form of equation (7) is

$$\widehat{\theta} = g(y, X) \tag{8}$$

By allowing the set of observations to change randomly, the estimate becomes itself a random variable, and (8) becomes the estimator of the unknown vector θ

$$\widehat{\theta} = g(\mathbf{y}, \mathrm{X}) \tag{9}$$

Let $\widehat{\theta}$ denote the random variable of the associated estimator. By adopting the squared error loss function to quantify deviations, a reasonable criterion to measure the performance of an estimator is the mean-square error (MSE),

$$\text{MSE} = \mathbb{E}\left[\left(\widehat{\theta} - \theta_o\right)^2\right], \tag{10}$$

where the mean $\mathbb{E}$ is taken over all possible training data sets of size N. If the MSE is small, the resulting estimates are close to the true-value. Although θ_o is not known, studying how the MSE depends on various terms will still help to learn how to proceed in practice and unravel possible paths that one can follow to obtain good estimators. Let's insert the mean value $\mathbb{E}\left[\widehat{\theta}\right]$ of $\widehat{\theta}$ in equation (10),

$$\begin{aligned}\text{MSE} &= \mathbb{E}\left[\left\{\left(\widehat{\theta} - E\left[\widehat{\theta}\right]\right) + \left(E\left[\widehat{\theta}\right] - \theta_o\right)\right\}^2\right] \\ &= \mathbb{E}\left[\left(\widehat{\theta} - E\left[\widehat{\theta}\right]\right)^2\right] + \left(E\left[\widehat{\theta} - \theta_o\right]\right)^2,\end{aligned} \tag{11}$$

Equation (11) suggests that the MSE consists of two terms, the first one, the variance around the mean value, and the second one is due to the bias, that is, the deviation of the mean value of the estimator from the true one. Considering that an unbiased parameter vector estimator satisfies:

$$\mathbb{E}\left[\widehat{\theta}\right] = \boldsymbol{\theta}_o \tag{12}$$

the MSE around the true value of $\boldsymbol{\theta}_o$ is defined as

$$\text{MSE} = \mathbb{E}\left[\left(\widehat{\theta} - \theta_o\right)^T\left(\widehat{\theta} - \theta_o\right)\right] = \sum_{i=1}^{l} E\left[\left(\widehat{\theta}_i - \theta_{oi}\right)^2\right] \tag{13}$$

Looking carefully at (13) reveals that the MSE for a parameter vector is the sum of the MSEs of the components $\widehat{\theta}_i$, i = 1,2,..., l, around the corresponding true-values θ_{oi}. Hence, by averaging a large number of the unbiased estimator, it is expected to get an estimate close to the accurate value. However, in practice, data are a commodity that is not always abundant. In such cases, where one cannot afford to obtain and average a considerable number of estimators, an unbiased estimator may not necessarily be the best choice.

2.2.2.2.2.1. Model development - regression learner

One of the most demanding issues of a ML workflow is that there is no way to know, with confidence, what modeling technique will work best on a particular dataset. That is why various authors suggest trying different regression techniques as an iterative process.

Results	
RMSE	0.26478
R-Squared	0.94
MSE	0.070107
MAE	0.16985

Table 1. Trained model results for wind speed. Source: Authors

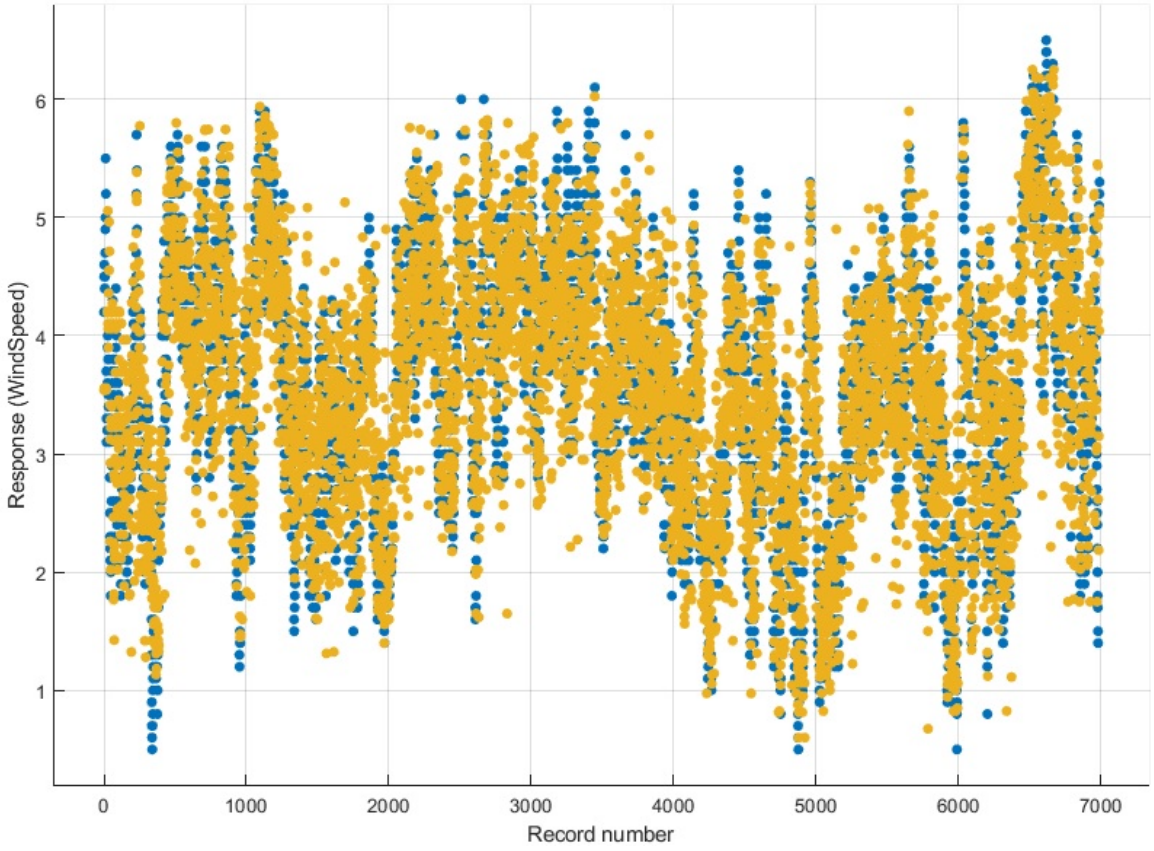


Figure 16. Data Readings - Response plot for GPR model. Source: Authors

This research tested 19 models covering linear regression (LR), bagged regression trees (BRT), support vector machines (SVM), Gaussian process regression (GPR), ensembles of tree models, and Neural Networks (NN), among others. Figure 16 shows the response plot that displays the predicted response versus the record number considering the results for a GPR - Matern 5/2 GPR for the wind speed. Likewise, table 1 exhibits the performance of the model considering different error measures.

2.2.2.2.3. Regularization

Regularization [30] is a mathematical tool to impose a priori information on the structure of the solution that comes as the outcome of the optimization task. A regularized version of the LS solution for the linear regression task given by equation (5) is,

$$\left(\sum_{n=1}^{N} x_n x_n^T + \lambda I\right)\widehat{\theta} = \sum_{n=1}^{N} y_n x_n \tag{14}$$

where $\boldsymbol{I}$ is the identity matrix of appropriate dimensions. The presence of λ biases the new solution away from that which would have been obtained from the unregularized LS formulation and attempts to reduce the norm of the estimated vector and simultaneously try to keep the sum of squared errors low. In order to achieve this combined goal, the vector components, $\boldsymbol{\theta}_i$, are modified in such a way so that the contribution in the misfit measuring term from the less informative directions in the input space is minimal. In other words, those of the components that are associated with less informative orientation will be pushed to smaller values to keep the norm small and concurrently have minimal influence on the misfit measuring term.

2.2.2.2.4. Ill-conditioning and overfitting

Inverse problems are typically ill-posed, as opposed to the well-posed ones [30]. Well-posed problems involve the existence of a solution, its uniqueness, and its stability. The latter condition is usually violated in ML problems, meaning that the obtained solution may be very sensitive to changes in the training set. That means that the obtained solution may be very sensitive to changes in the training set, a sensitivity known as ill-conditioning. This behavior results when the model used to describe the data can be complex concerning the large number of unknown free parameters concerning the number of data points. The profile with which this problem manifests itself in machine learning is known as overfitting, denoting that during training, the estimated parameters of the unknown model learn too much about the idiosyncrasies of the specific training data set, and the model

poorly performs when it deals with data sets other than the used for training purposes. The MSE criterion discussed in equation (11) attempts to quantify this data dependence of the task through the mean deviation of the obtained estimates from the accurate value by changing the training sets. When the number of training samples is small for the number of unknown parameters, the available information is not enough to reveal a sufficiently good model that fits the data, and it can be misleading to the presence of the noise and possible outliers.

Regularization is an elegant and efficient tool to cope with the complexity of the model, that is, to make it less intricate and more smooth. There are different ways to achieve this. One way is by constraining the norm of the unknown vector, where regularization helps to replace the original ill-conditioned problem with a nearby one, which is well-conditioned and whose solution approximates the target one. Another example where regularization can help to obtain a solution, or even a unique solution to an otherwise unsolvable problem, is when the model's order is large compared to the number of data, only a minor percentage of the model's parameters are nonzero, and a standard LS linear regression approach has no solution. Regularizing the sum of a squared errors cost function using the norm of the parameter vector can lead to a unique solution.

2.2.2.2.5. Curse of dimensionality

A low MSE depends on two things. First, the model's complexity (number of parameters) should be small enough concerning the number of training points. Second, to eliminate the overfitting, the number of training points should be more than the number of parameters. How big a data set should be to be more relaxed concerning the performance of the designed predictor?. The answer to this question depends on the dimensionality of the input space; it turns out that the larger the dimension of the input space is, the more data points are needed. This is related to the curse of dimensionality [45].

There are various ways to cope with the curse of dimensionality and try to exploit the available data set correspondingly. A popular direction is to resort to suboptimal solutions by projecting the input/feature vectors in a lower-dimensional subspace. This approach leads to small performance losses because the original training data lives in a lower-dimensional subspace due to physical dependencies restricting the number of free parameters. The intrinsic dimensionality of the problem is the number of free parameters leading to learning about the subspace onto which to project. Remarkably, the dimensionality of the input space is not always crucial. On the contrary, the critical factor is the so-called VC dimension [46] of a classifier. In several classifiers, such as linear

classifiers or neural networks, the VC dimension relates directly to the dimensionality of the input space.

2.2.2.2.6. Validation

A leading phase in any ML task is to quantify the performance that the designed model exhibits in practice - measuring the performance against the training data set would lead to an optimal value of the performance index because it uses the same set on which the estimate was optimized [47]. For instance, if the model is complex enough, with a broad number of free parameters, the training error may even become zero since a perfect fit to the data can be achieved. However, it is more meaningful to look for the generalization performance of the estimator, specifically its average operation computed over different data sets which did not participate in the training and the error associated with this average performance, known as generalization error or test error.

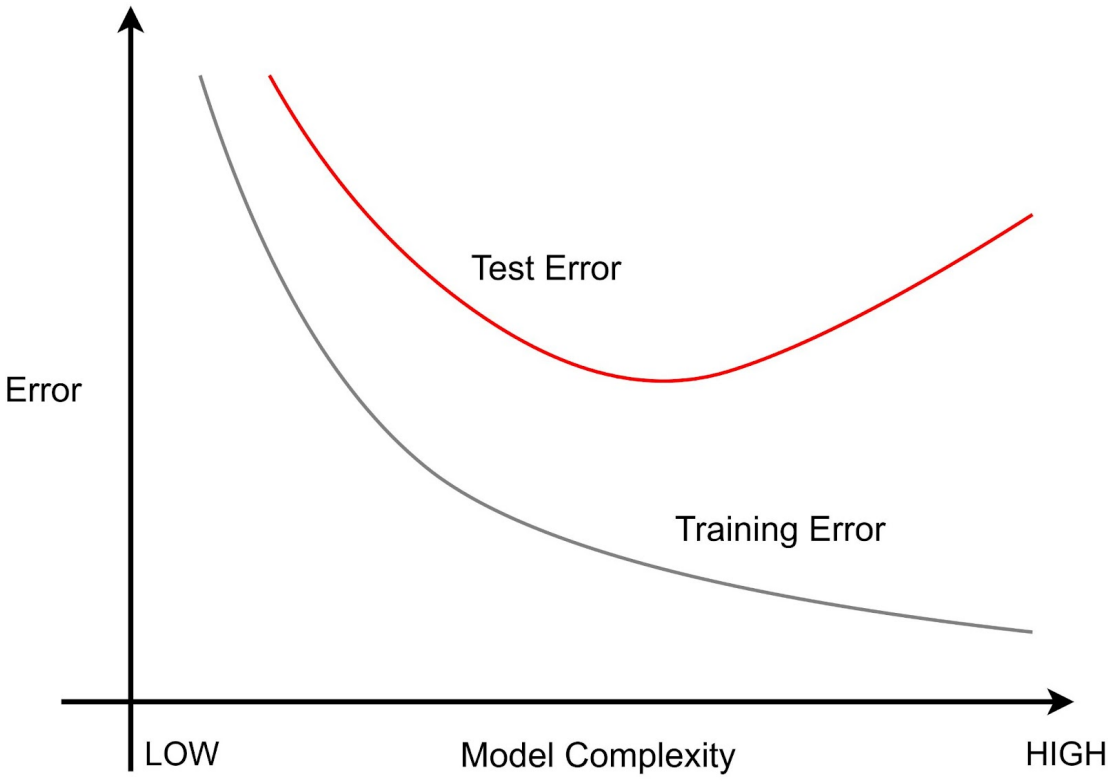


Figure 17. Model performance. Source: Authors.

Figure 17 shows a typical performance expected as a result in practice. It exhibits the error measured on the single training data set with the average test error as the model complexity varies. If the model is complex concerning the size of the available training set, then the

error measured on the training set will be overoptimistic, and the generalization error takes large values based on the variance term of the MSE given by equation (10).

Contrastingly if the model is too simple, the generalization error will also attain large values in the MSE case. In this case, the contribution is mainly due to the bias term. The idea is to have a model complexity corresponding to the minimum of the respective curve. In practice, it is necessary to test the performance of a predictor using different data sets, a process known as validation. Assuming that enough data are at the designer's disposal, it is recommendable to split the data into three parts: one for training, another for testing, and the other to validate the performance. Following this path implies the confidence that the training set, the validation set, and the test set are big enough for the model complexity. A large-enough test data set is required to provide a statistically good result on the generalization error.

Likewise, frequently the size of the available data is not sufficient, but it is necessary to validate the outcome. In this case, cross-validation is a common technique usually employed. According to this method, the data set is split into K roughly equal-sized parts. The training is repeated K times, each time selecting one different part of the data for testing and validation and the remaining K-2 parts for training. This is advantageous as it permits testing with a part of the data not involved in the training phase, so it can be considered independent. Eventually, it is possible to combine the errors from the test sets to get a better estimate of the generalization error that the estimator exhibits in real-life applications. In practice, the K value depends very much on the application, and typical values are of the order of 5 to 10 [30]. The price paid for K-fold cross-validation is the complexity of training K times, high estimator's variance, and the unknowable dependencies between runs, revealing that the validation task is far from innocent. Ideally, there should be available large data sets divided into several non-overlapping training sets along with separate validation sets and test sets that are large enough.

2.2.2.2.6.1. Model development - models comparison on training and test set

A convenient measure for the model generalization capability considers its performance on a holdout dataset. It could be possible that even though a BRT model has a lower MAPE on the training set than an NN model, it has a higher MAPE on the test set. This situation indicates that it is likely to have an overfit on the training set. Figure 18 compares the LM, GPR, NN, and BRT models using training and test data sets for available wind speed data. This comparison allows the selection of two models to advance with the ML procedure, NN and BRT. A similar selection process applies to the available energy load and irradiance data sets.

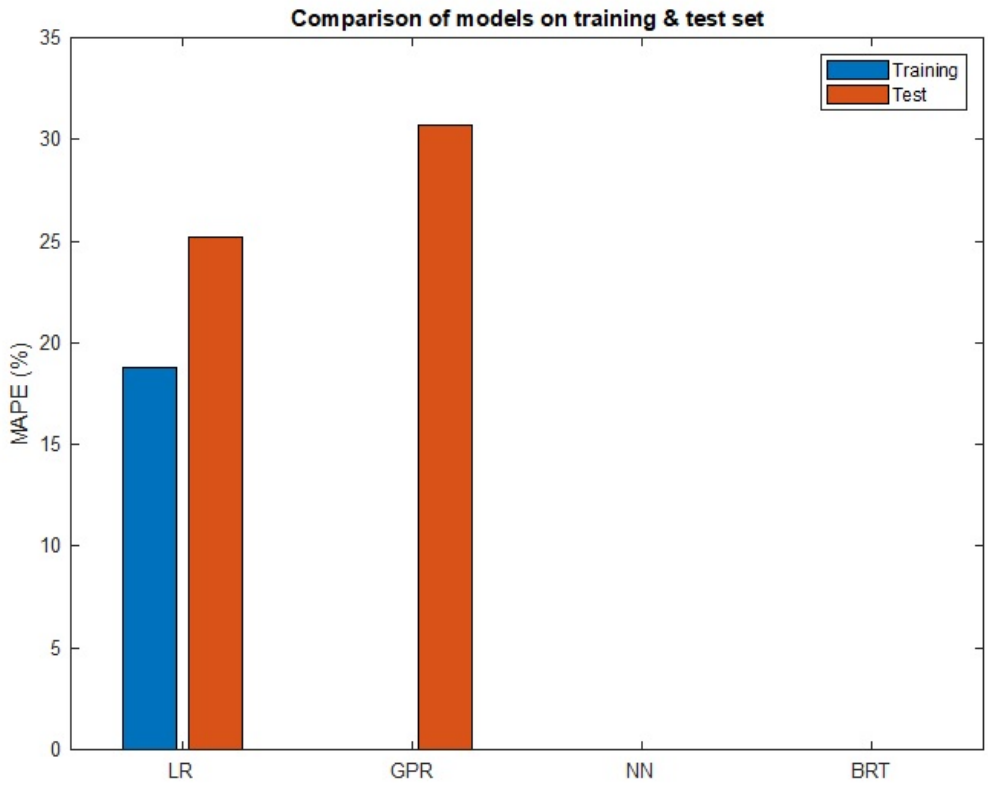


Figure 18. Data Readings - Comparison response for the wind model. Source: Authors

2.2.2.2.6.2. Model development - further validation

While the model may do well on average, it could have problems in isolated cases. Therefore, it is necessary to look at how it does on the validation data. Figures 19, 20, and 21 present the measured response compared with the ahead prediction for the GPR, BRT, and NN models considering the available wind speed data. This exercise confirms the selection of the NN and BRT models to move forward with the ML procedure for the wind speed case.

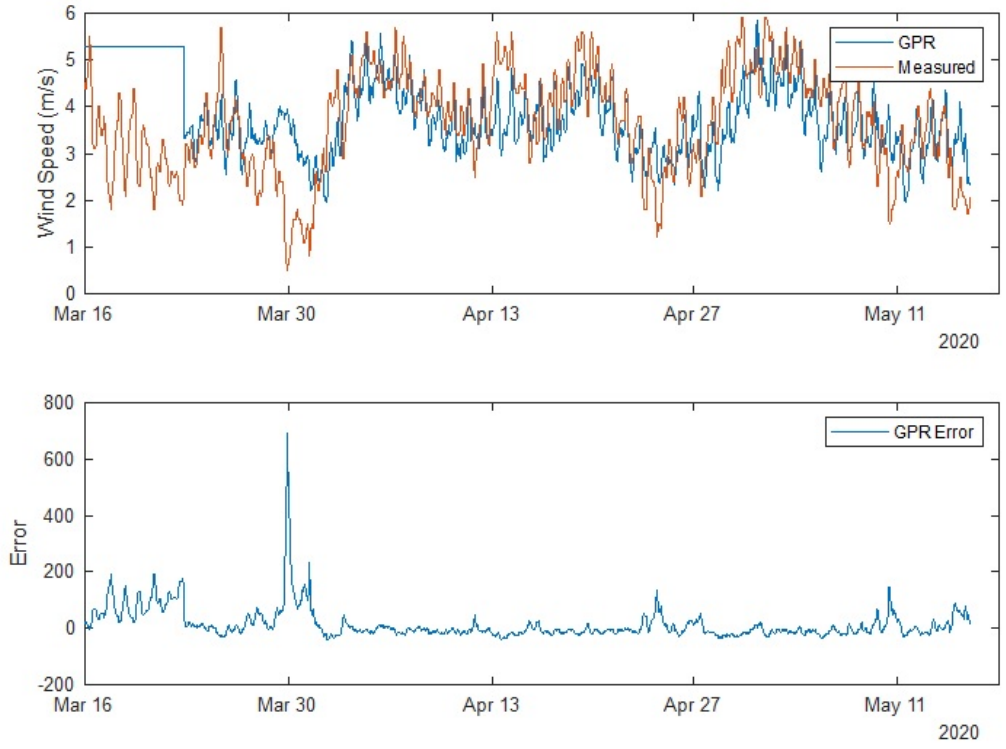


Figure 19. Data Readings - Overfit response plot for GPR model. Source: [29]

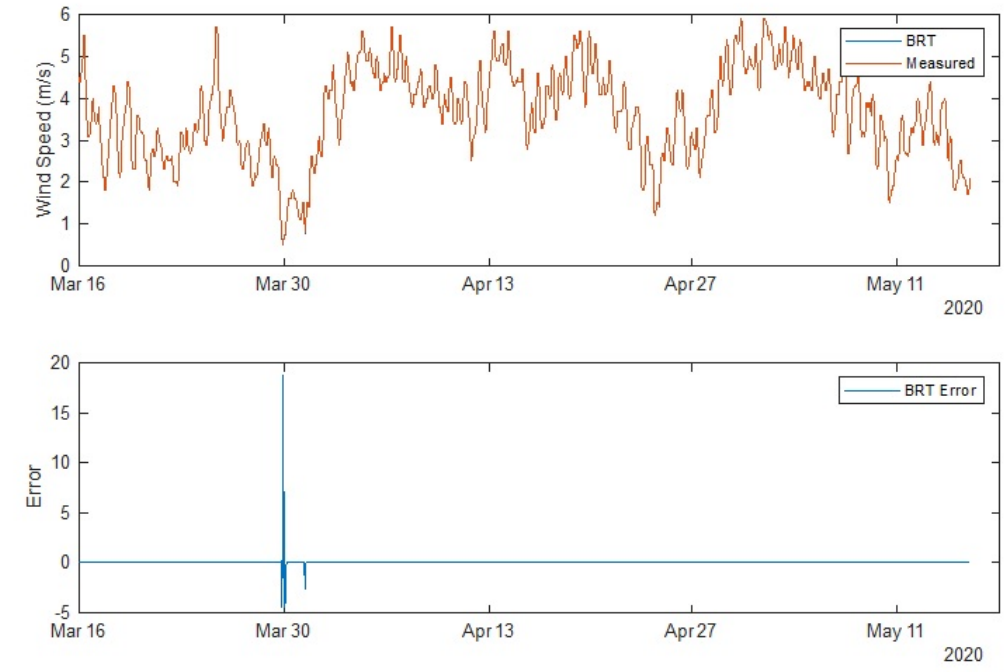


Figure 20. Data Readings - Response plot for BRT model. Source: [29]

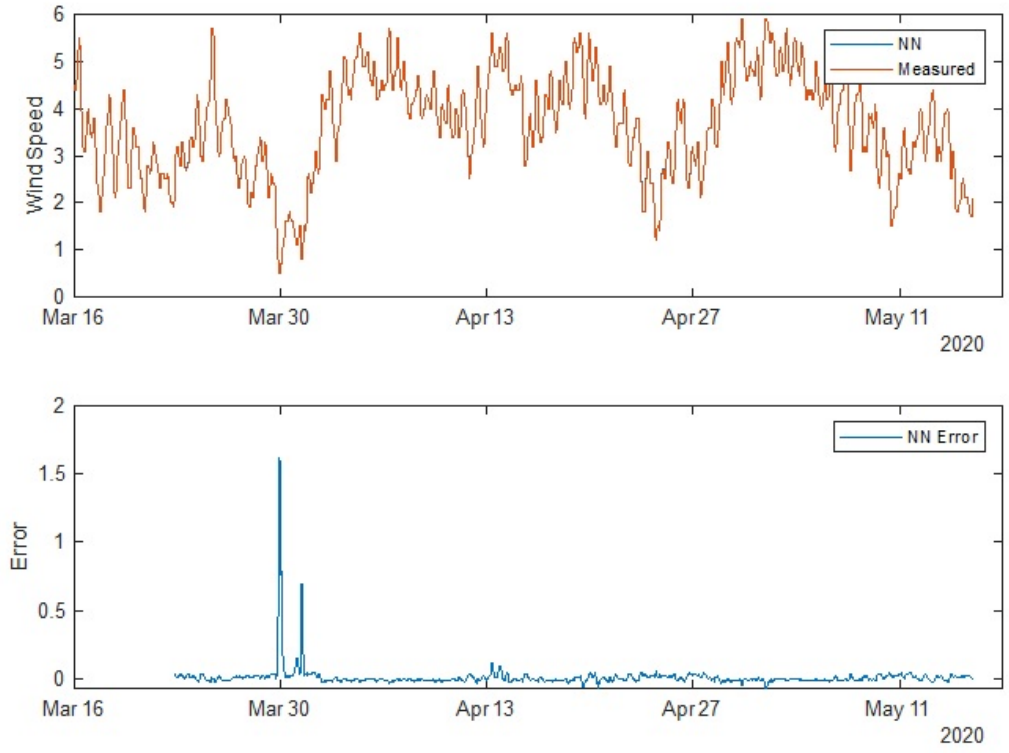


Figure 21. Data Readings - Response plot for NN model. Source: Authors

2.2.3. Section conclusion

The CREMS uses the LP to convert the optimized data into a prediction model. As with other solutions [37], the phases presented in this research request a search space and an evaluation criterion to define a parametric model. The parameter estimation scheme used by the CREMS considers regressions, estimators' efficiency, bias-variance dilemmas, ill-conditioning, over-fitting, and the curse of dimensionality [31]. Nevertheless, no method guarantees a modeling approach for one information set. So, the CREMS performs an iterative strategy using independent regression approaches such as LR, RT, SVM, GPR, BRT, and NN [29]. Likewise, the ML procedure uses coefficient values optimization techniques to restrain definitive attributes under a square error standpoint, providing function convexity. The CREMS uses the MAPE to estimate the model's abstraction capacity and divides the information into three parts [31] training, testing, and validation, considering 50%, 25%, and 25% as percentage partitions and regarding the model's complexity, respectively. The CREMS framework evaluates the MAPE in both the testing and training sets, discards problems associated with parameter estimation as estimator efficiency or over-fitting, and corroborates the predictor performance using the validation

set. This enhances the framework's autonomy and adaptation to resource fluctuations and assures specific objectives two and three.

2.3. Implemented framework

2.3.1. Section introduction

The results presented in this section are available in the IEEE in the following conferences papers [29], [48].

This doctoral research involves developing an experimental MG testbed (MG) that allows real-time task emulation. These tasks include predicting energy consumption and generation in an emulated MGC managed by an on-cloud EMS that runs ML methods to solve an EDP. The application of supervised learning techniques to estimate future consumption and generation in an MGC helps the management system verify and modify itself according to resource fluctuation. The MGT considers PV, wind energy, and BESS system models and includes hardware-in-the-loop (HIL), power-hardware-in-the-loop (PHIL), and real-life inverters to obtain a close emulation of the MGC. It also regards a communication protocol to connect the cloud-based EMS with the PHIL and an IoT strategy to track consumption patterns and execute experiments related to solving the EDP or other experiments related to demand response. These features permit compliance with specific objective number one and validate the delineated by the IEEE STD 2030.7 - 2017, as stipulated in specific objective four.

2.3.2. Model for a solar PV system and the MPPT principle

This research uses a standard single exponential model with a series and shunt resistance [49]–[51] shown in figure 22. The equivalent circuit model equations result by applying Kirchhoff's voltage and current laws to the circuit model. Equations (15) and (16) exhibit these V-I relationships:

$$i = I_{ph} - I_0\left(e^{\frac{v + iRs}{n_s V_t}} - 1\right) - \frac{v+iR_s}{R_{sh}} \tag{15}$$

$$v = -R_{sh}\left[i - I_{ph} + I_o\left(e^{\frac{v+iR_s}{n_s V_t}} - 1\right)\right] - iR_s \tag{16}$$

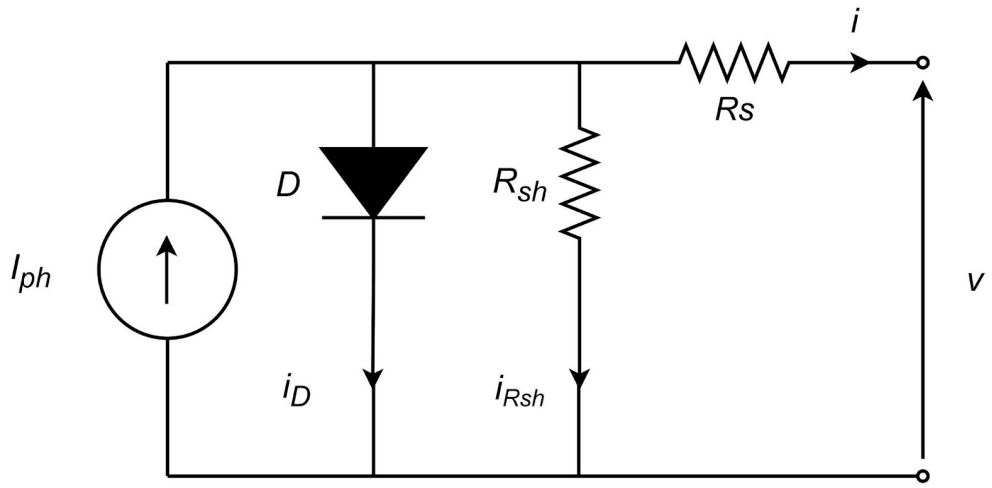


Figure 22. PV's equivalent circuit model. Source: Authors.

At open circuit, where no current flows, (17) applies

$$I_{oc} = 0 = I_{ph} - I_0 e^{\frac{V_{oc}}{n_s V_t}} - \frac{V_{oc}}{R_{sh}} \tag{17}$$

At short circuit where there is no voltage across the panel, (18) is applicable.

$$I_{sc} = I_{ph} - I_o e^{\frac{R_s I_{sc}}{n_s V_t}} - \frac{I_{sc} R_s}{R_{sh}} \tag{18}$$

Finally, at maximum power point

$$I_{mpp} = I_{ph} - I_o e^{\frac{V_{mpp} + R_s I_{mpp}}{n_s V_t}} - \frac{I_{mpp} R_s}{R_{sh}} \tag{19}$$

Equation (20) shows the relationship between thermal voltage at standard conditions V_{tstc} and the diode ideality factor (A).

$$V_{t_{stc}} = \frac{AkT_{stc}}{q} \tag{20}$$

Equations (17) - (20), in conjunction with the measurements typically provided by solar PV panel datasheets (Table 2), characterize the PV model. Further equation manipulation remains with three unknown variables: R_s, R_{sh}, and A.

Datasheet Params	Make and Model of Solar Panel				
	SLP020	***MSX120 (24V)***	***MSX60***	***MSX64***	***KC200GT***
Isc (A)	1.31	3.87	3.8	4.0	8.21
Voc (V)	21.6	42.1	21.1	21.3	32.9
Impp (A)	1.2	3.52	3.5	3.66	7.66
Vmpp (V)	18.6	33.7	17.1	17.5	26.3
Ns	36	72	36	36	54
ki (%/°C)	0.065	0.065	0.065	0.065	3.18 mA /°C
kv (mV/°C)	-65	-160	-80	-80	-123

Table 2. Datasheet values for a selection of commercially available PV panels at STC. Source: Authors.

This work derives circuit parameters by solving the non-linear equations with a Newton Raphson method and using a Jacobian matrix [52]. In this case, the value of the three unknowns should satisfy

$$\begin{bmatrix} f_1(R_{sh},\, R_s,\, A) \\ f_2(R_{sh},\, R_s,\, A) \\ f_3(R_{sh},\, R_s,\, A) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{21}$$

The iterative solution reaches by using a Newton-Raphson progression with a Jacobian matrix as in (22)

$$x_{n+1} = x_n - J(x_n)^{-1} f(x_n) \tag{22}$$

The Jacobian matrix follows a matrix of partial derivatives of the three equations f_1, f_2 and f_3:

$$J = \begin{bmatrix} \frac{\delta f_1}{\delta R_{sh}} & \frac{\delta f_1}{\delta R_s} & \frac{\delta f_1}{\delta A} \\ \frac{\delta f_2}{\delta R_{sh}} & \frac{\delta f_2}{\delta R_s} & \frac{\delta f_2}{\delta A} \\ \frac{\delta f_3}{\delta R_{sh}} & \frac{\delta f_3}{\delta R_s} & \frac{\delta f_3}{\delta A} \end{bmatrix} \tag{23}$$

The progression in (23) must be initialized with estimates of the three parameters to be solved for the Newton-Raphson iterative process to converge.

$$x_0 = \begin{bmatrix} R_{sh_o} & R_{s_o} & A_0 \end{bmatrix}^T \tag{24}$$

Once ascertained, the parameters R_s, R_{sh}, and A are assumed locked over a range of panel temperatures and insolation. It still provides a good approximation of PV panel behavior. Most of the effects of temperature and insolation on photocurrent I_{ph}, dark saturation current I_o, and short circuit current I_{sc} can be considered linear, as can the effect of temperature on open-circuit voltage, V_{oc}. However, V_{oc} varies logarithmically with insolation, and as a non-linear effect, its solution is dealt with iteratively. The PV panel output voltage and current will depend upon the panel's operating temperature and insolation. A given panel output voltage will give rise to a given output current, and vice-versa. The V-I curve point where the panel is operating will determine these factors. From the perspective of modeling the PV panel, the choice is to model the output voltage as a function of output current or the output current as a function of output voltage.

2.3.3. Model for wind energy system

Wind power varies throughout the day as wind speed fluctuates. The power produced by a wind turbine is given by [53]

$$P = 0.5\pi\rho Cp(\lambda, \beta)R^2V_w^3 \tag{25}$$

where R is the turbine radius, V_w is the wind speed, ρ is the air density, Cp is the power coefficient, λ is the tip speed ratio, and β is the pitch angle. In this work, β takes a zero value. The tip speed ratio is given by:

$$\lambda = \frac{\Omega R}{V_w} \tag{26}$$

where **Ω** is the turbine angular speed. The dynamic equation for the wind turbine is

$$\frac{d\Omega}{dt} = \frac{1}{J\left[T_m - T_L - Fw_r\right]} \tag{27}$$

where J is the system inertia, F is the viscous friction coefficient, T_m is the torque developed by the turbine, T_L is the torque due to load, in this case, the generator torque, and w_r is the turbine rotor speed. The turbine power coefficient is a non-linear function given by:

$$Cp(\lambda, \beta) = 0.5176\left(\frac{116}{\delta} - 0.4\beta - 5\right)e^{-\frac{21}{\delta}} \tag{28}$$

where,

$$\frac{1}{\delta} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{1 + \beta^3} \tag{29}$$

The optimum power obtained from a wind turbine is

$$P_{opt} = K_{opt}\Omega_{opt}^3 \tag{30}$$

where,

$$K_{opt} = \frac{0.5\pi\rho Cp_{max}(\lambda,\beta)R^5}{\lambda_{opt}^3} \text{ and } \Omega_{opt} = \frac{\lambda_{opt} V_w}{R} \tag{31}$$

The power for a certain wind speed is maximum at a defined value of rotor speed called optimum rotor speed Ω_{opt}. It is the speed that corresponds to the optimum tip speed ratio $\Omega_{opt}\lambda_{opt}$. The turbine should operate consistently at λ_{opt} to achieve maximum possible power by controlling the turbine's rotational speed to get the optimum rotation speed. Table 3 presents the specifications of the wind turbine.

Parameter	Value
Mechanical output power	210 kW
Base wind speed	11 m/s
Maximum power at base wind speed	0.9 pu

Pitch angle	0°

Table 3. Parameters of 210 kW wind turbine. Source: [54]

2.3.4. Model of the battery energy storage system

The battery model proposed in [55] and presented in figure 23 summarizes a comprehensive battery performance by coupling the dynamic electrical circuit characteristics with non-linear capacity effects on the battery.

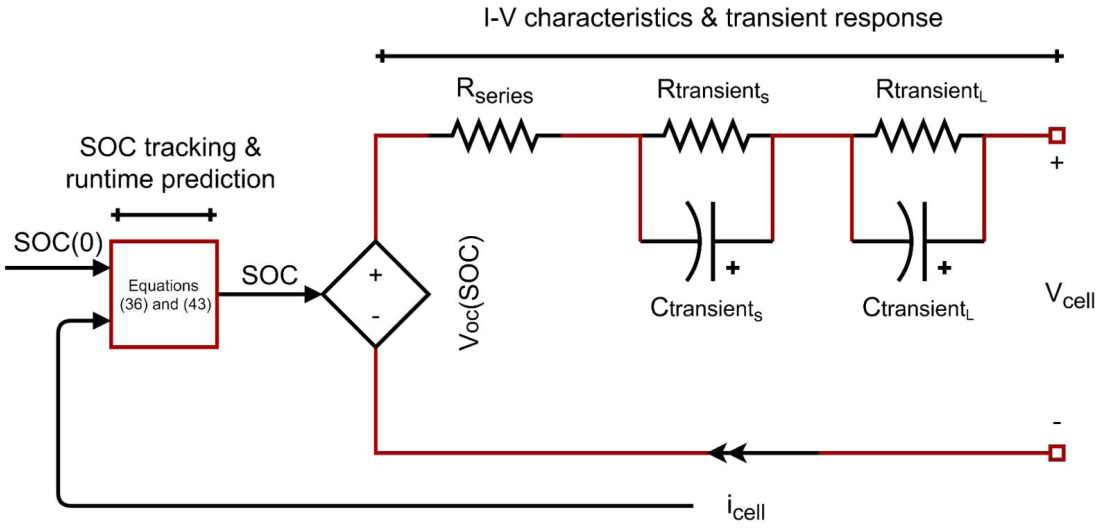


Figure 23. Proposed battery model. Source: Authors

Consider a period of $t_0 < t < t_r$ in which the battery cell is first discharged with a constant current and then rests for the remainder of the period. The model expresses the following:

$$SoC(t) = \frac{C_{available}(t)}{C_{max}} = SoC_{initial} - \frac{1}{C_{max}}\left[\int i_{cell}(t)dt + C_{unavailable}(t)\right] \quad (32)$$

$$V_{oc}[SoC(t)] = a_0 e^{-a_1 SoC(t)} + a_2 + a_3 SoC(t) - a_4 SoC^2(t) + a_5 SoC^3(t) \quad (33)$$

$$V_{cell}(t) = V_{OC}[SoC(t)] - i_{cell}(t)R_{series} - V_{transient}(t) \quad (34)$$

$$V_{transient}(t) = V_{transient_S}(t) + V_{transient_L}(t) \quad (35)$$

C_{max}, $C_{available}$, and $C_{unavailable}$ are the maximum, available and unavailable battery capacities. The state of charge (SoC) of the battery reduces when it delivers charge to the load as expressed in (32) by the current integration term. $C_{unavailable}$ represents the nonlinear SoC variation due to the non-linear capacity effects of the battery. $SoC_{initial}$ is the estimated SoC at the end of the last operating period before t_0. In practice, $SoC_{initial}$ can correct by using (33) with the open-circuit voltage measured during some resting intervals. The terminal voltage V_{cell} is estimated by V_{oc}, the voltage across R_{series}, and the transient voltage term $V_{transient}$, which represents the short-term response of the RC network. The RC network parameters are the functions of SoC:

$$R_{series}(SoC) = b_0 e^{-b_1 SoC} + b_2 + b_3 SoC - b_4 SoC^2 + b_5 SoC^3 \quad (36)$$

$$R_{transient_S}(SoC) = c_0 e^{-c_1 SoC} + c_2 \quad (37)$$

$$C_{transient_S}(SoC) = d_0 e^{-d_1 SoC} + d_2 \quad (38)$$

$$R_{transient_L}(SoC) = e_0 e^{-e_1 SoC} + e_2 \quad (39)$$

$$C_{transient_L}(SoC) = f_0 e^{-f_1 SoC} + f_2 \quad (40)$$

These parameters are constant when the SoC is 20-100% and change exponentially when the SoC varies below 20%. The available capacity is determined by

$$C_{available}(t) = C_{initial} - \int i_{cell}(t)dt - C_{unavailable}(t) \quad (41)$$

where the unavailable charge of the battery can be expressed as [56]

$$C_{unavailable}(t) = u(t) \quad (42)$$

Likewise, all of the electrical circuit parameters of the model came from the least-squares curve fitting of the experimental data obtained at room temperature using pulse discharge currents with an interval of 5% SoC.

2.3.5. MG testbed

The physical system under test is a 300 kW MGC consisting of three MG considering photovoltaic and wind generation in combination with storage under the active generation context, as illustrated in figure 24 and scaled to a laboratory prototype at the SEC:

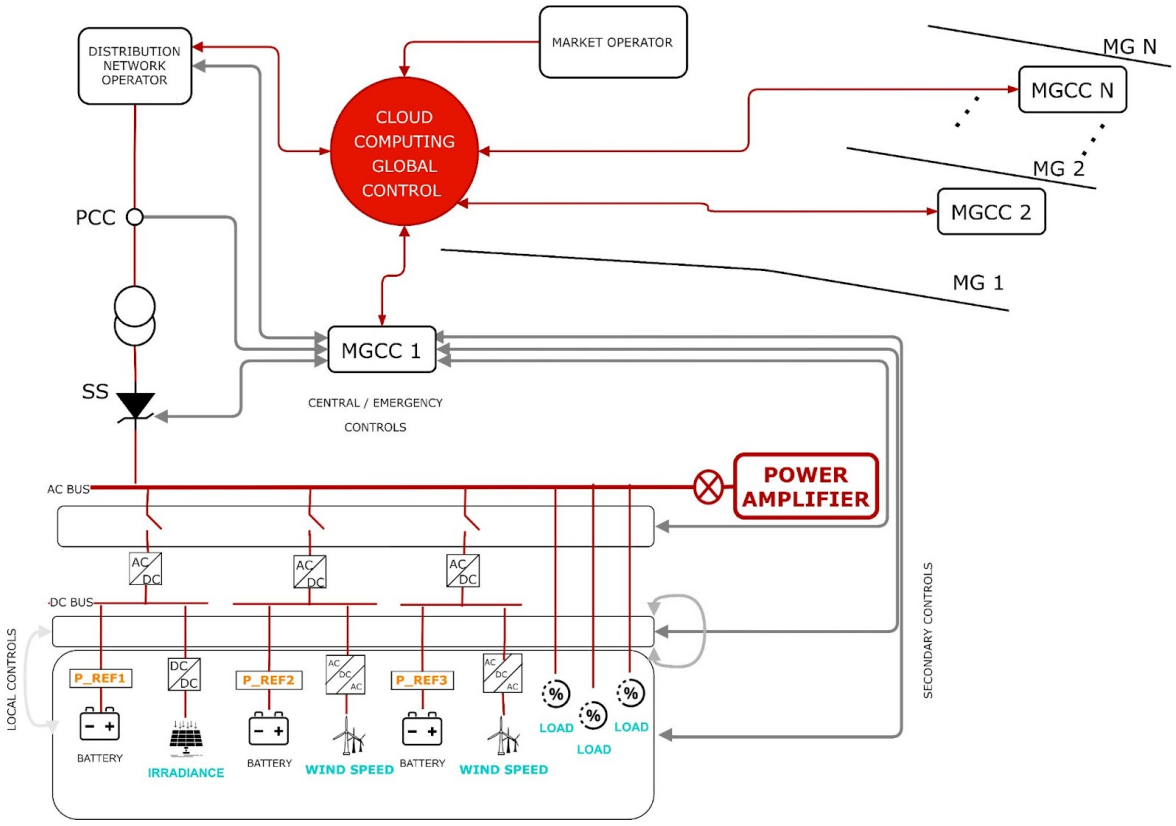


FIGURE 24. General scheme of the MGC implemented in SEC. Source: [28]

Figure 25 illustrates an overall scheme of the real-life PHIL testbed available in the SEC. For this experiment, the MGC merged three (03) Inverter-based Distributed Generators (IDGs) coupled to individual BESS to resemble three dispatchable units: two photovoltaic systems and one wind turbine. This procedure assumes that each MG chose to contribute to augmenting the assemblage's total profit. Also, to secure the prototype's accuracy and power production, the assembly incorporates the utility grid through a four-quadrant power amplifier [57]. This component improves the cluster's adequate reference when considering features such as frequency and voltage and facilitates balancing the energy demand with the energy generated when the load or the generation varies.

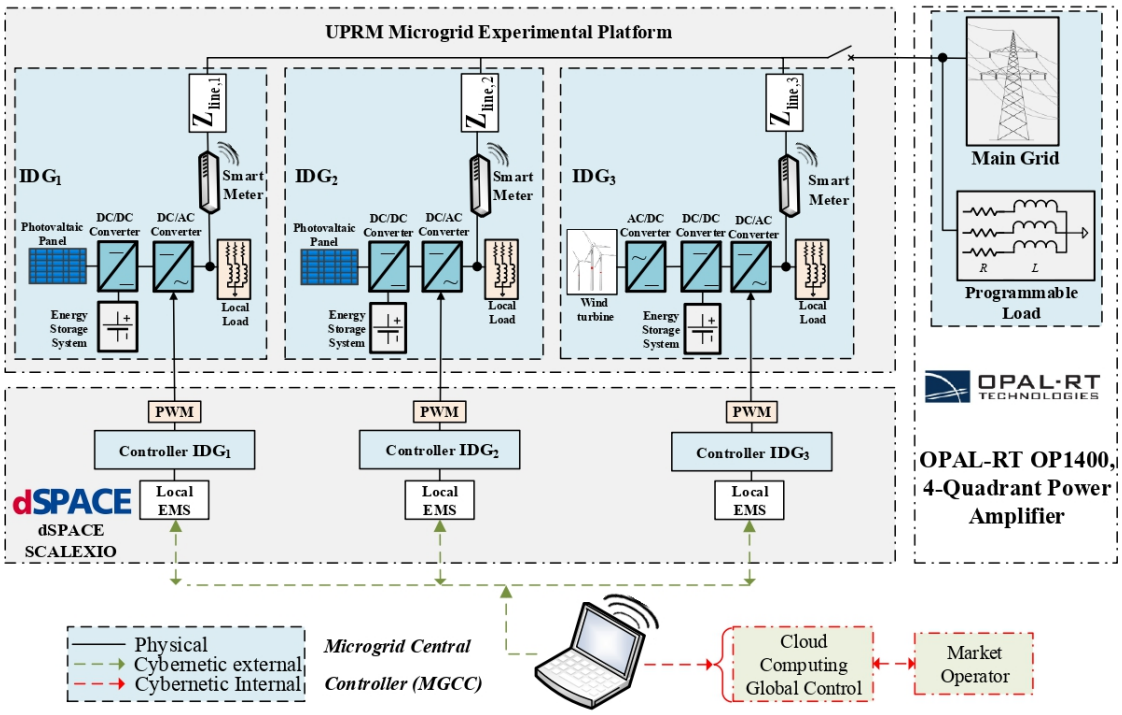


FIGURE 25. Implemented framework. Source: [29]

The MG testbed (MGT) includes a dSPACE SCALEXIO [58] system working as PHIL interface that controls the test bench inverter-based generators (Figure 26). It also incorporates an OPAL-RT OP1400, a four-quadrant power amplifier [59] used as PHIL that can emulate the utility grid. By default, the MGT includes four 2.2 kW three-phase DANFOSS VLT-302s [60] that symbolize each IDG. In this application, three inverters emulate the IDG units, while the fourth acts like the load profile. Three-Phase LCL modular filters connect to each inverter output to mitigate harmonic components. Also, each inverter can be any time connected or disconnected from the common AC bus using a three-phase solid-state relay. A PLC is used for controlling the three-phase solid-state relay to connect the inverters as desired. In this research, the four-quadrant power amplifier works as the utility grid to set the operative frequency, provide voltage coupling, and balance the consumption demand. Likewise, the MGT modular structure allows connecting or disconnecting additional loads to the AC bus to test any load variation. In addition, the MGT uses LEM sensor boxes to measure the three-Phase LCL modular filters for analog measurement.

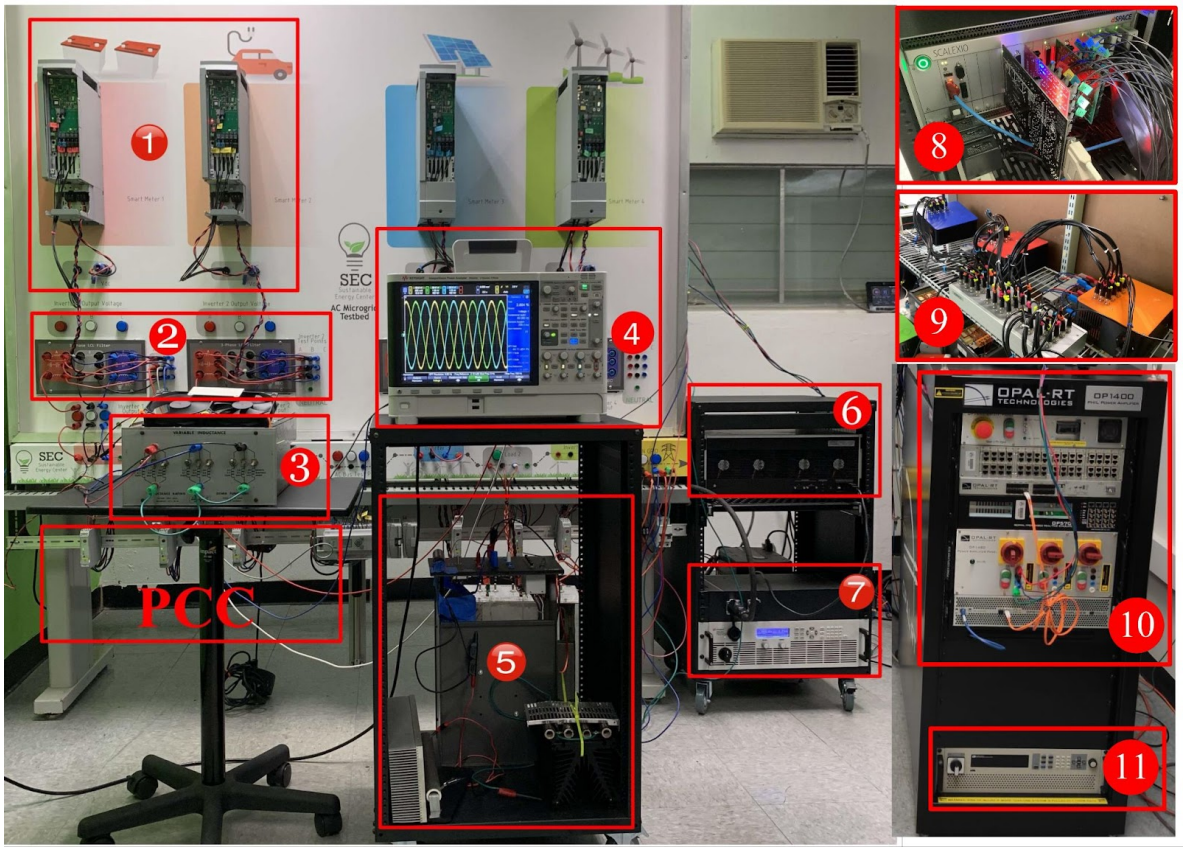


FIGURE 26. Microgrid Testbed, 1-IDGs, 2-LCL output filters, 3-Linear Load, 4-Power analyzer, 5-Non-Linear load, 6-dSPACE-Scalexio (back view), 7-DC Supply (IDGs), 8-dSPACE-Scalexio (front view), 9-Sensor Boxes, 10-OP1400 Power Amplifier, 11-DC Supply (OP1400). Source: [61]

The MGT also includes IoT energy efficiency features as it includes smart-meters to send measured data to the CREMS via JavaScript Object Notation (JSON) advanced format setting in MQ Telemetry Transport (MQTT) protocol in one-minute intervals (see figure 27). In this way, it is possible to achieve two advantages: first, the direct use of object data format that facilitates direct integration with diverse applications without forcing to change the existing JSON format into a numeric data type, and second, the graphical UI for setting JSON data structure is intuitive and easy to understand. In addition, the protocol used to communicate the CREMS with the real-life PHIL platforms is the user datagram protocol (UDP) [62].

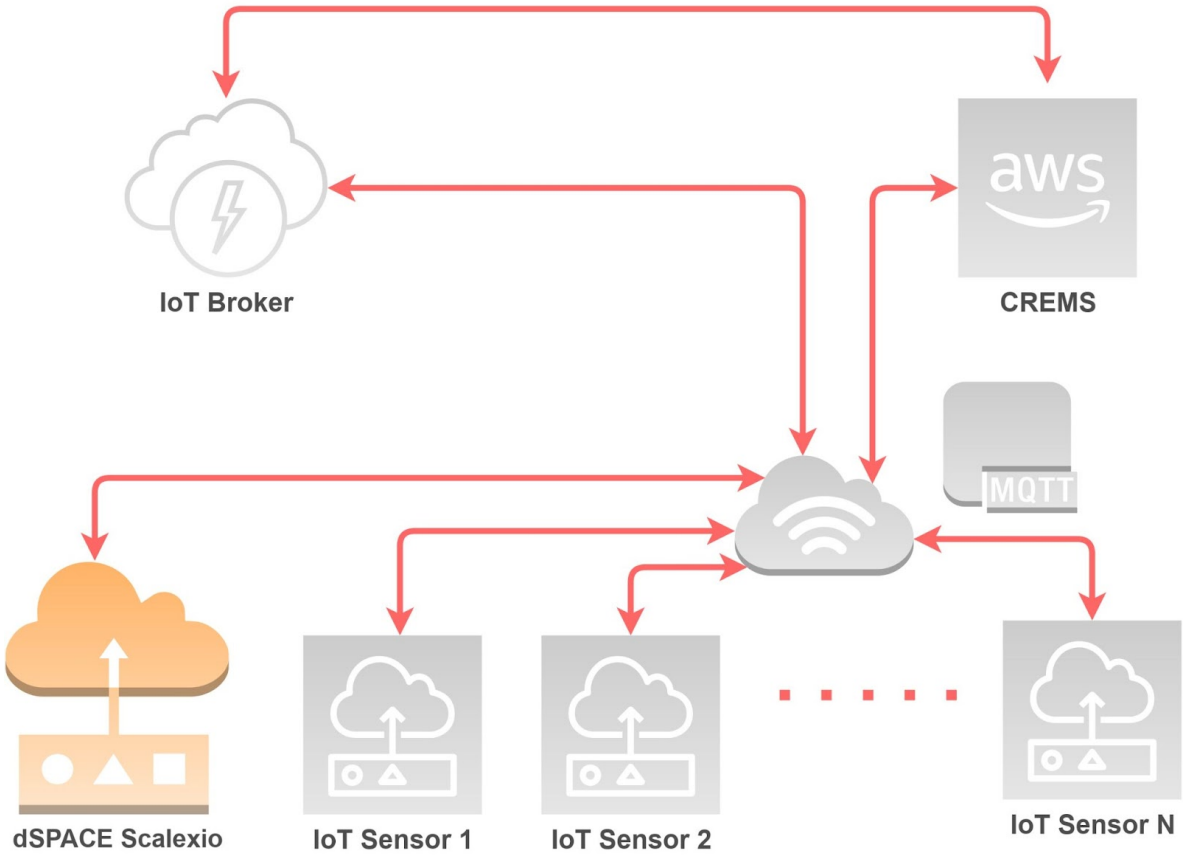


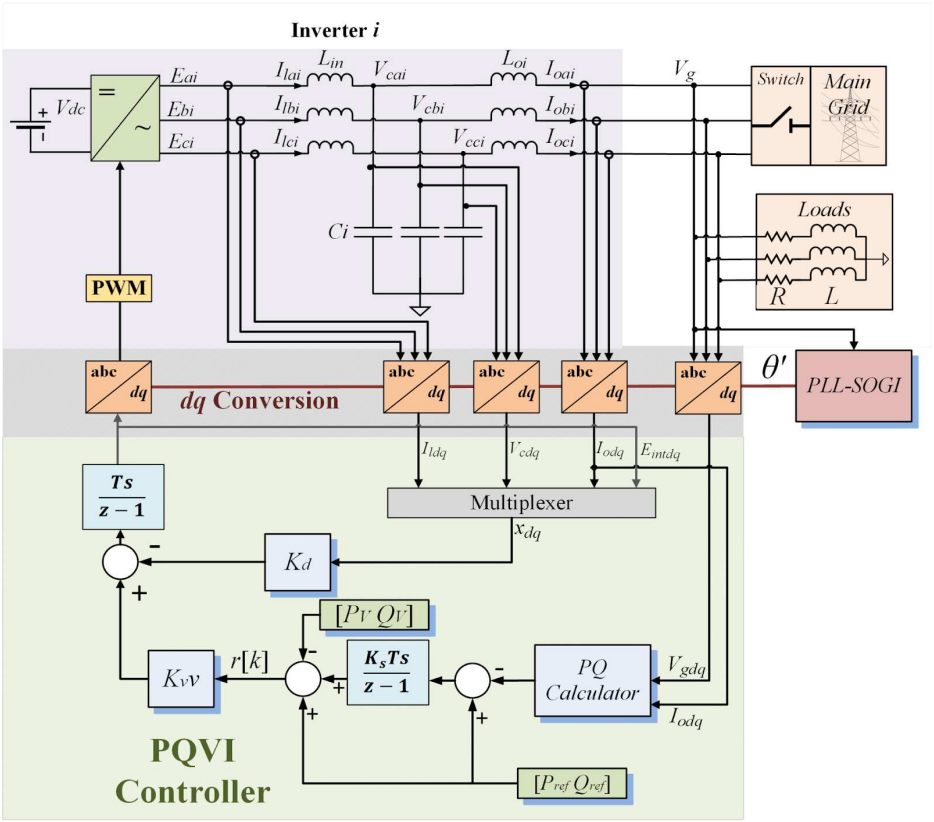
FIGURE 27. IoT strategy. Source: [61]

As mentioned above, for this test, the main feature of the MGT is having a fully-deployed CREMS using AWS. The CREMS test operation considers controlling the battery's SoC, exploiting usable renewable resources data, and analyzing usage-energy historical data to reduce the price paid to the grid provider. Via the ML model usage, the economic power dispatch problem (EPDP) should reduce the utility grid price by considering the forecast generation and demand power [63]. Other conditions [64] comprehends the production level limits and the balance between the predicted energy production, the battery features, and the utility assistance. The batteries' status breaks into charging and negative discharging phases submitted to power limits. Likewise, the battery's SoC stays within its bounds by permanent observation. In addition, it is necessary to define the battery charging power coming from the utility or the RES. Finally, discharging into the utility is restrained to the residual demand, and the proposed SoC equalization model is verified. As discussed, the cloud infrastructure allows the operating of massive data with reliability and validity, adding secure management and simplifying the implementation of massive data processing. According to this research performance, solving problems related to the demand response and utility grid stability via the IoT sensors inclusion facilitated the definition of the

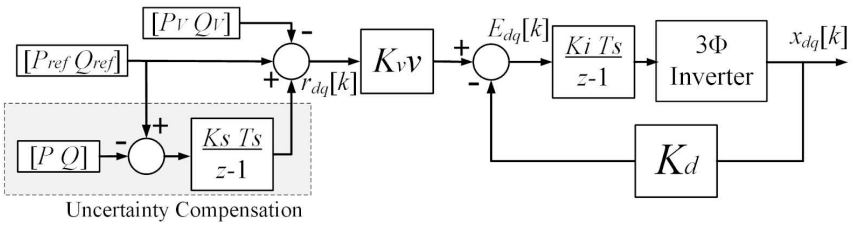
infrastructure in the cloud scalability in terms of storage and data processing. Also, to improve the MGT security, the operation considered a network topology where the dSPACE Scalexio had a public IP configured with firewall rules that accepts sending and receiving data coming only from the AWS server static IP [65]. On the other hand, importing data from the IoT broker's server considers another methodology as the CREMS connects directly to the IoT broker's server via the MQTT protocol. However, the MGT facilitates the incorporation of other commercial IoT meters using a local wireless access point to send the data directly to the CREMS.

2.3.5.1. Power Controller Implementation

Figure 28a exhibits the control block diagram of an IDG. The diagram shows that the generator has two main components: a physical power system and a control system.



a)



b)

FIGURE 28. Optimal Power Control Strategy for Grid Connect Mode. Source: [66]

The physical power system consists of a three-phase DC–AC converter, an LCL filter, and a Pulse Width Modulation (PWM) block. On the other hand, the control system utilizes an integrated PQVI controller based on a linear quadratic regulator (LQR) with optimal reference tracking (ORT). Figure 28b shows the complete scheme of the LQR-ORT controller. For getting the gains (Kd, Kvv), the system utilizes a modified cost function that weights the tracking error and the control input using the equation and process defined in [66].

2.3.6. Section conclusion

Unlike previous approaches, the implemented framework uses IoT data and deploys an ML model to forecast the MGC's generation and load energy curve, as the system optimization algorithm executes in an MPS instance using AWS. This approach defines the CREMS performance and eliminates the offline optimization process [66]. The changes implemented in the SEC's testbed configuration support the findings exposed in [28]:

- The CREMS intends to reduce utility grid costs and optimize battery usage.
- The implemented framework efficiently combines PHIL interfaces, ML, IoT, and decision-making.
- The energy production prediction provides an initial assessment to the energy administrator and the final user.
- The system can manipulate energy decisions close to real-time.
- The EMS structure adapts conveniently to the amount of MG available at a specific time.

The above-mentioned standpoints permit corroborating the compliance of specific objectives numbered one and four.

2.4. Scalable and autonomous framework of the EMS to administer an MGC under a CC environment

2.4.1. Section introduction

The results presented in this section are under revision by Applied Energy in the paper titled: Full-deployed energy management system tested in a microgrid cluster.

The fulfillment of the general objective of this doctoral research demands definite requirements. The ultimate CREMS, described in this section, integrates data acquisition, generation, and consumption forecasting under a cost-effective procedure beyond routinary CC [67] to satisfy them. For this, it was essential to define a suitable technique to organize the IoT data arriving from devices placed in distributed RESs. Also, it was required to determine the ML incorporation procedure to forecast generation and consumption attributes in a specific spot. Likewise, it was demanding to test the CREMS in a real-life PHIL testbed to validate the ML, IoT, and CC appliance's applicability in an MGC, recognizing the variances between the simulation results and the practical verification [61].

2.4.2. Cloud-based real-time EMS framework

As characterized in [28], the CREMS divides into three sections (see Figure 29): supervisory system (SS), data acquisition, storage & processing (DASP), and scheduling and optimization (S&O). In this research, the SS section uses a Matlab Production Server (MPS) appliance where the broker links each MG with a specific procedure allowing a CREMS managerial standpoint with each MG, and permits corporate, mobile, or web applications to call Matlab code. The incorporation of novel IoT devices lets dispatch the data gathered by either the SS or the IoT instruments into the DASP using Big Data (BD) mechanisms available in the Amazon Simple Storage Service (S3) [68]. The DASP section is a data file collection whose analysis provides the pattern and predictive framework needed to execute unique duties. Lastly, the S&O section runs improved optimization patterns and frameworks to get optimal references timely returned to the RES.

The resultant architecture facilitates select multi-cloud supported framework services that allow scalable CREMS characterization for an MGC control. Likewise, the ML active algorithm gives sufficient endurance and excellent energy reference update under unforeseen changes in generation and consumption. The architecture improvement allows the S&O section transparently to create, compile, and deploy the CREMS algorithm to the server unaltering its functioning. Similarly, it facilitates the DASP section accessing the mechanisms that the CREMS demands to incorporate BD, ML, and IoT routines to enhance

the MGC management execution. Lastly, the SS unit enhances the framework's capabilities to obtain low latency processing, expansion capability, and redundancy required by the CREMS.

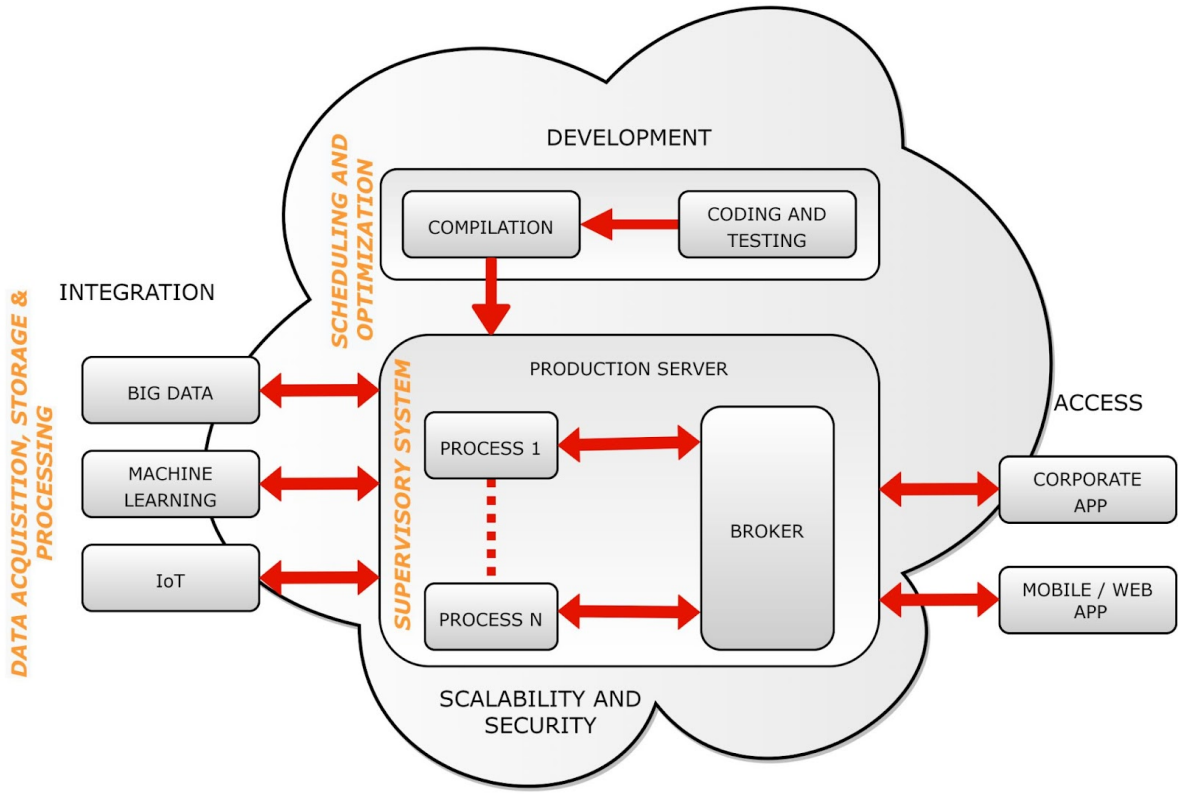


FIGURA 29. Conceptual CREMS. Source: [61]

The IoT adaptation involves measurement units' on-field deployment in specific points to warranty transmission links oriented to the IoT broker or accessing points to retrieve the measured data in the S3 bucket. The streamed data is first gathered in the physical layer and then transmitted to superior layers to provide appropriate management. Consequently, as declared in [48], [69], a practical method to incorporate IoT devices with an MG is to reorient the initial operations of the Microgrid Control Center (MGCC) to match the required IoT capacities. Figure 30 depicts the enhanced hierarchical structure and its incorporation into the framework.

The primary level permits the device's communication over diverse transmission protocols. Numerous standardization institutions have made notable endeavors to deliver advanced solutions for industrial, residential, and building segments that have permitted developing the data networks and protocols. The secondary level connects the IoT for a better user

experience. The tertiary level considers scenarios where the convenience and financial standpoints alter to accomplish an optimal general service.

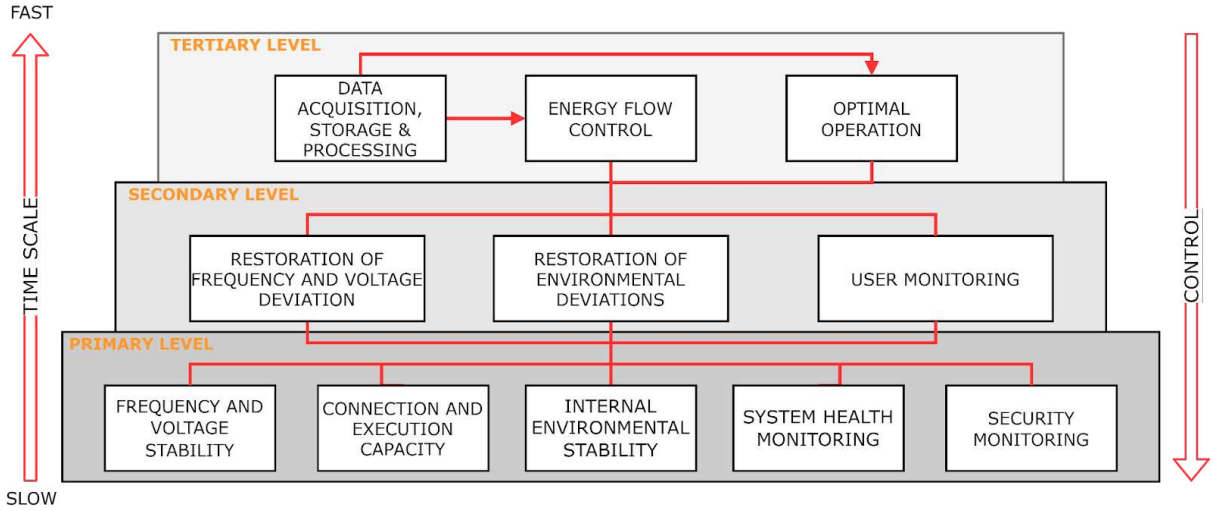


FIGURE 30. Enhanced hierarchical structure. Source: [61]

2.4.3. Section conclusion

The CC, IoT, and ML capabilities experimented into the EMS of a cluster of integrated MG facilitate data analysis under a real-time experience and permit an SL model definition beyond offline computing limitations, giving autonomy features to the ultimate framework. Also, these capabilities offer scalability to the CREMS framework beyond the inherent CC features. A particular issue calls attention to the framework's experimental performance. The IoT strategy changed transparently from a data acquisition & storage to a DASP with the IoT broker inclusion. The mentioned permits comply with the requirements of this doctoral research's general objective.

2.5. CREMS performance verification

2.5.1. Section introduction

The experimental verification of the CREMS performance follows an EPDP, ensuring the IoT applicability and the suitable ML functionality in a cloud-connected MG testbed assemblage. This section presents two optimization models, a general optimization model and an SoC equalization one. Every model is described conveniently with its particular considerations, objective functions, problem constraints, and performance under the

CREMS execution. In this way, this doctoral research validates the restrained by the specific objective number four.

2.5.2. General optimization model

The results presented in this section are available in the IEEE in the following conference paper [48] and will be socialized in [70].

The suggested CREMS considered a mixed-integer linear programming (MILP) problem, as defined in [28], and employs it to decrease the expenditure applicable to the utility grid [3], [71]. The scheduling executes T hours with timeframes of 30 minutes. Where t represents the T_{th} hour (for $t = 1, 2, 3, \ldots, T$), and i describes the I_{th} IDG (for $i = 1, 2, 3$). Table 4 summarizes the MGT parameters setup used in this trial, and the CREMS output variables define the setpoints for each IDG, $P_{pv}(1, t)$, $P_{pv}(2, t)$, and $P_{wt}(1, t)$, for this case.

Parameter	**Grid**	**P_{V1}**	**B_{PV1}**	**P_{V2}**	**B_{PV2}**	**WT**	**B_{WT}**
Selling cost, *$/kWh*	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Operational cost, *$/kWh*	0.21	0.00	0.05	0.00	0.05	0.00	0.05
Production (min), *kW*	-0.30	0.00	0.00	0.00	0.00	0.00	0.00
Production (max), *kW*	0.30	0.30	0.30	0.30	0.30	0.30	0.30
Initial SoC, %	-	-	100	-	100	-	100
Minimum SoC, %	-	-	20	-	20	-	20
Maximum SoC, %	-	-	100	-	100	-	100
Storage capacity, *kW*	-	-	0.5	-	0.5	-	0.5
Battery efficiency, %	-	-	0.93	-	0.93	-	0.93

TABLE 4. MGT configuration parameters. Source: [48]

2.5.2.1. Objective function

Conveniently, the CREMS expands progressively according to the number of MGs available in the cluster, namely, ∀i ∈ *I* and ∀t ∈ *T* [48], [72]:

$$min_{\ p(1,t),\ldots,p(n,t)} \quad \sum_{i \in N} \sum_{t \in T} C_{i,t} p_{i,t}^{forecast} + C_t p_{Grid,t} \tag{45}$$

$$subject\ to \quad \sum_{i \in N} \sum_{t \in T} p_{i,t}^{forecast} + p_{grid,\,t} \leq p_{d,t}^{forecast}$$

Where, $p_{d,t}$ describes the MGC power requirement at time *t*, p_{it} regards the power produced by the I_{th} DG at time *t*, the IDG set restricts to *i = {1, 2, 3}*, the scheduling time considers *t = {1, … , T}*, $C_{i,t}(p_{i,t}^{forecast})$ defines the cost function for the I_{th} DG at time *t*, and $C_t(p_{grid,t})$ describes the cost function for the utility at the same time.

2.5.2.2. Problem constraints

Equation 46 sets limits for production by considering [18]. Equation 47 settles the cluster output balance corresponding to the estimated production, the battery availability, and the utility load. The battery condition separates into charging and discharging states, as denoted by equation 48, and restrains to specific boundaries with equation 49.

$$p_{i,t}^{min} \leq p_{i,t} \leq p_{i,t}^{max} \tag{46}$$

$$p_{i,t}^{forecast} = p_{i,t}^{pv,\,forecast} - p_{i,t}^{battery,\,pv} + p_{i,t}^{wt,\,forecast} - p_{i,t}^{battery,\,wt} + p_{grid,t} \tag{47}$$

$$p_{i,t}^{battery} = p_{i,t}^{battery,\,charge} + p_{i,t}^{battery,\,discharge} \tag{48}$$

$$p_{i,t}^{battery,min} \leq p_{i,t}^{battery} \leq p_{i,t}^{battery,\,max} \tag{49}$$

Equation 50 follows the batteries' SoC, which in correspondence, must remain within its limits as indicated in equation 51. Equations 52 and 53 establish the battery charging power being achievable from both the utility (equation 54) and the generation (equations 55 and 56). Battery discharging into the utility is controlled by equation 57, mainly limited to the not satisfied demand.

$$SoC_{i,t} = SoC_{i,t-1} + \frac{p_{i,t-1}^{battery,\,charge} \cdot \eta_i^{battery} \cdot \Delta t}{C_i^{battery}} + \frac{p_{i,t-1}^{battery,\,discharge} \cdot \Delta t}{\eta_i^{battery} C_i^{battery}} \quad (50)$$

$$SoC_i^{min} \leq SoC_{i,t} \leq SoC_i^{max} \quad (51)$$

$$p_{i,t}^{battery,\,charge} = p_{i,t}^{battery,\,charge\,PV} + p_{i,t}^{battery,\,charge\,Grid} \quad (52)$$

$$p_{i,t}^{battery,\,charge} = p_{i,t}^{battery,\,charge\,WT} + p_{i,t}^{battery,\,charge\,Grid} \quad (53)$$

$$p_{i,t}^{battery,\,charge\,Grid} \geq 0 \quad (54)$$

$$0 \leq p_{i,t}^{battery,\,charge\,PV} \leq p_{i,t}^{pv,\,forecast} \quad (55)$$

$$0 \leq p_{i,t}^{battery,\,charge\,WT} \leq p_{i,t}^{WT,\,forecast} \quad (56)$$

$$0 \leq p_{i,t}^{battery,\,\text{discharge}} \begin{cases} -p_{d,t}^{forecast} + p_{i,t}^{forecast} & if\ p_{d,t} \geq p_{i,t} \\ = 0 & \text{else} \end{cases} \quad (57)$$

2.5.2.3. General model performance

In this case, the CREMS execution considered a summer week with a moderate PV and WT generation and the batteries' SoC initial state set to 100%. Figure 31 exhibits the power scheduling for the MGC. The three lines present the demanded, dispatch, and utility grid energy in red, black, and purple, respectively. As proposed, the CREMS optimizes both the energy delivered by each RES and the one supplied by the utility grid to cover the load. Likewise, the utility grid, emulated here by the power amplifier, balances the energy demand with the generation in specific periods - days 2, 3, and 7. Figure 32 regards these specific time spots and exhibits the load generation balance impact over the cost assumed by the MGC final users. As stated in table 4, the utility grid and RES usage consider operating fees. The market operator defines the utility cost and the maintenance, the replacement, and the lifespan determine the long-term RES expense. Then, figure 33 shows the number-one PV-system performance during the week. It includes power available, delivered, and charging state. The ML model specifies the dispatched power according to the forecasted power availability. Likewise, within the timeframe considered, the batteries' SoC exhibits favorable results as its percentages stay regulated to the restrained ones reducing high discharge repercussions, expanding batteries' lifespan, and reducing utility grid usage in the charging process - most of the charging power comes from RES. Figure 34 exhibits this, and figure 31 corroborates this statement. A similar analysis applies to the second PV and the wind turbine systems.

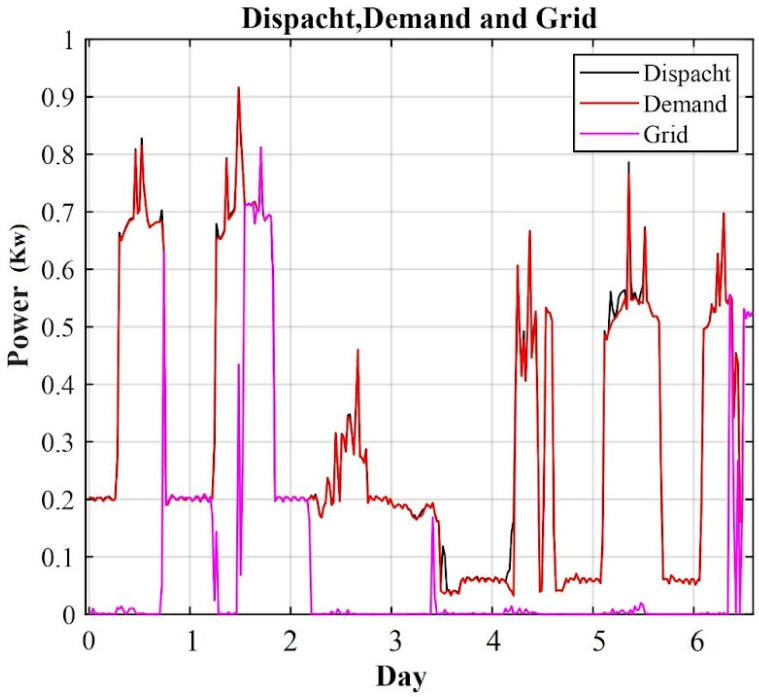


FIGURE 31. Scheduling results. Source: [72].

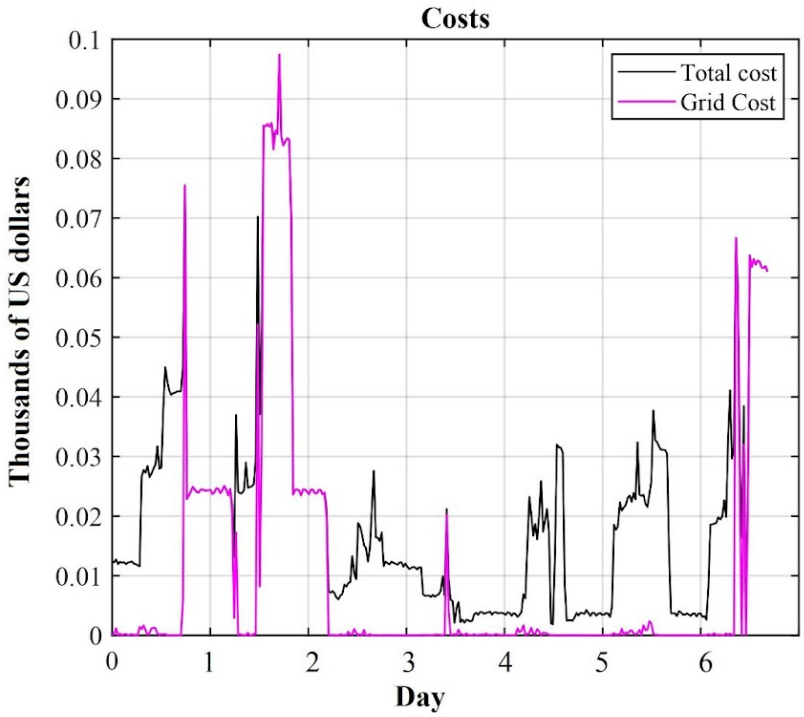


FIGURE 32. Cost results. Source: [72].

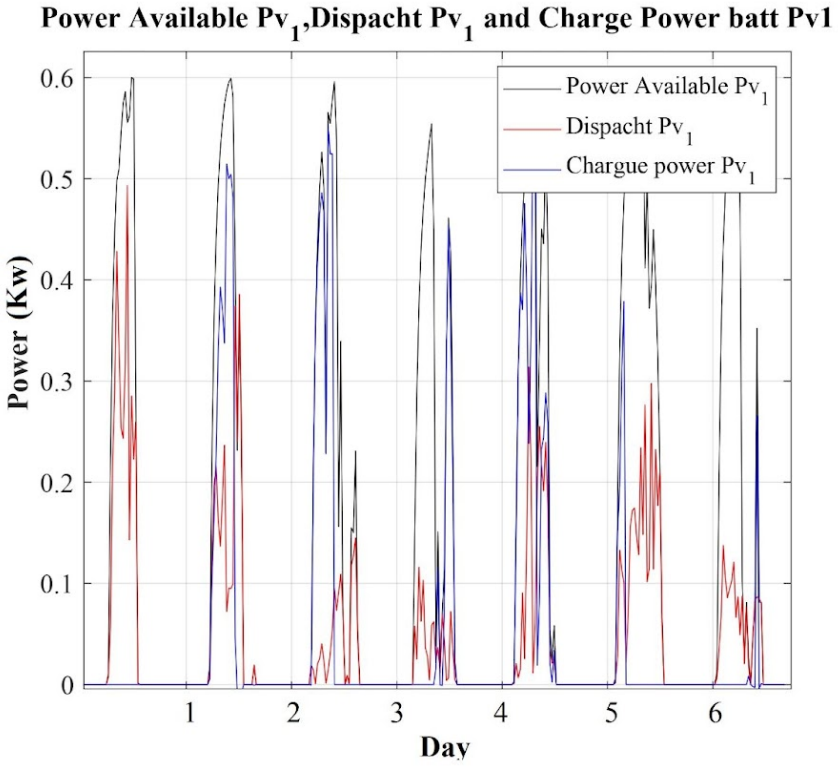


FIGURE 33. PV1 results. Power units expressed in kW. Source: [72] .

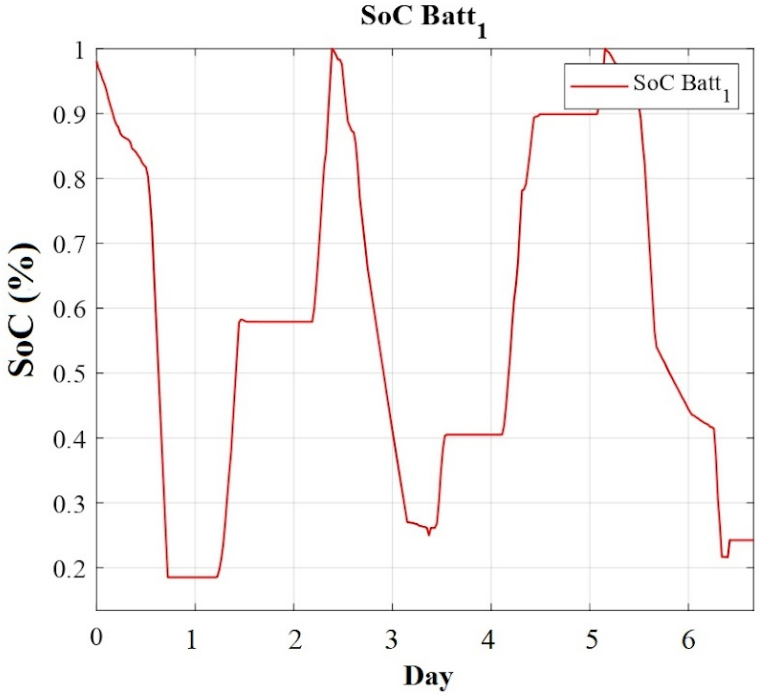


FIGURE 34. Battery´s Soc results. SoC level expressed in percentage. Source: [72].

2.5.3. SoC equalization optimization model

The results presented in this section are under revision by Applied Energy in the paper titled: Full-deployed energy management system tested in a microgrid cluster.

The CREMS also considers minimizing operating costs by solving a typical EPDP while equalizing the SoC of the ESSs integrated into each i_{th} IDG through a MILP to avoid uneven degradation of the BESS [61]. In this way, when the system disconnects and reconnects or, a battery replacement occurs, the CREMS will ensure SoC equalization. This follows the recommendation to perform an SoC equalization during the electrical system operation defined in [73]. To include the SoC equalization in the optimization model, the error between the different SoCs in the i_{th} IDG and at each time t, $E_{SoCi,t}$, is defined as:

$$E_{SoCi,t} = \left(SoC_{i,t} - SoC_{i+1,t}\right), \forall i \in N\ -\ 1, t \in T \tag{58}$$

The equalization reaches by minimizing this variable. The variable must be positive to indicate a feasible solution and is given by:

$$\left(SoC_{i,t} - SoC_{i+1,t}\right) \geq 0, \forall i \in N\ -\ 1, t \in T \tag{59}$$

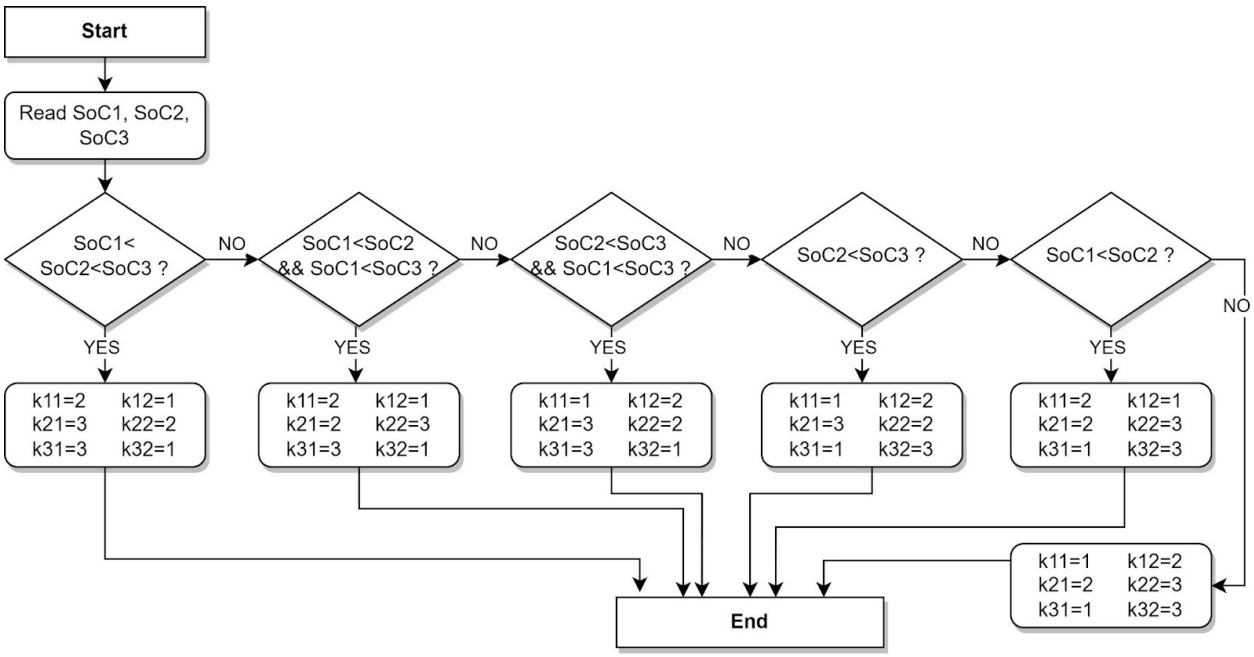


FIGURE 35. Flow diagram for the SoC's downstring associated with the SoC error constraint. Source: [61]

Then, data preprocessing is required to down sorting the SoC of the ESSs before conducting the optimization problem for all time *t*. A decision-making stage is included within the optimization process to perform this task, as shown in figure 35. Thus, the algorithm calculates the error between pairs (k11 - k12, k21 - k22, and k31 - k32) and guarantees their positiveness. The definitions of the ancillary indexes guide constraint (59).

2.5.3.1. Objective function

The objective function of the optimization problem presented in (60) optimizes the operating costs and favors the SoC equalization,

$$\sum_{i\in N}\sum_{t\in T}\left(C_{i,t}p_{i,t}^{forecast}\Delta t + C_{i,t}p_{Grid,t}\Delta t + C_{i,t}E_{SoCi,t}\right) \quad (60)$$

The first two terms in (60) minimize the cost associated with the usage of energy from the IDGs and the utility grid, where ($p_{i,t}^{forecast}\Delta t$) is the i_{th} IDG forecasted energy and ($p_{Grid,t}\Delta t$) is the energy exchanged with the utility grid. The third term is a penalty for having an equalization error between SoCs. $C_{i,t}$ includes the cost of using any resource and the penalty coefficient related to the equalization. The coefficients were adjusted to give priority to the minimization of the cost.

2.5.3.2. Problem Constraints

The first constraint details the energy balance. At each time spam, t, which lasts Δt, the energy provided by all the IDGs, $\sum_{i,N} p_{i,t}^{forecast}$, together with the energy absorbed from the grid, $p_{Grid,t}$, fed the load. In this scenario, the problem considers two kinds of load, the one measured by the smart meters, $p_{d,t}^{smart}$, and the forecasted load $p_{d,t}^{forecast}$. This fact enables the CREMS to react online to mismatches between the forecasted load and real one and, consequently, recalculate the dispatched power for each IDG. Therefore, the equation that represents the energy balance is,

$$\sum_{i,N} p_{i,t}^{forecast}\Delta t + p_{Grid,t}\Delta t - p_{d,t}^{forecast}\Delta t - p_{d,t}^{smart}\Delta t = 0, \forall\, t \quad (61)$$

The energy of each IDG comes from RES, $p_{RES_{i,t}}\Delta t$, and ESSs, modeled as the subtraction between the discharged and charged ESS energy,

$$p_{i,t}^{forecast}\Delta t = p_{RES_{i,t}}\Delta t + \left(p_{Batt,disch_{i,t}}\Delta t - p_{Batt,ch_{i,t}}\Delta t\right), \forall\, t \tag{62}$$

In turn, the power provided by the IDGs and the utility grid should be bounded. On one part, the maximum power supplied by the RES considers the manufacturer's recommendations, which defines the $p_{RES_{nom_i}}$. In the case of the utility grid, it depends on the infrastructure and contracts with the local operator to establish p_{Grid}. In this way, the boundaries of the power of IDGs and the utility grid are,

$$0 \leq p_{RES_{i,t}} \leq p_{RES_{nom_i}}, \forall i, t \tag{63}$$

$$0 \leq P_{Grid,t} \leq p_{Grid}, \forall t \tag{64}$$

To model the behavior of SoC of the ESSs, its relationship with the exchanged energy is,

$$SoC_{i,t} = \begin{cases} SoC_{i,0} - \frac{\left\{p_{Batt,disch_{i,t}}\Delta t\right\}.100\%}{\eta_{dis_i}.C_i^{Batt}} + \frac{\eta_{ch_i}.\left\{p_{Batt,ch_{i,t}}\Delta t\right\}.100\%}{C_i^{Batt}} & \forall i, t = 1 \\ SoC_{i,t-1} - \frac{\left\{p_{Batt,disch_{i,t}}\Delta t\right\}.100\%}{\eta_{dis_i}.C_i^{Batt}} + \frac{\eta_{ch_i}.\left\{p_{Batt,ch_{i,t}}\Delta t\right\}.100\%}{C_i^{Batt}} & \forall i, 1 < t \leq T \end{cases} \tag{65}$$

In this way, the SoC available in the ESS in the previous time step, $SoC_{i,t-1}$, is updated by algebraically adding the charged or discharged energy to the ESS, $p_{Batt,ch_{i,t}}$ or $p_{Batt,disch_{i,t}}\Delta t$ respectively, previously weighted with the ESS physical parameters: capacity in kWh, C_i^{Batt}, and charge or discharge efficiency η_{ch_i} and η_{dis_i}, respectively.

Finally, the variables of the boundaries also should be defined. In this formulation these constraints are modeled as,

$$SoC_{i,t}^{min} \leq SoC_{i,t} \leq SoC_{i,t}^{max}, \forall i, t \tag{66}$$

$$0 \leq p_{Batt,ch_{i,t}} \leq p_{Batt,chm}.x_{batt,t}, \forall i, t \tag{67}$$

$$0 \leq p_{Batt,disch_{i,t}} \leq p_{Batt,dischm}.(1 - x_{bat,t}), \forall i, t \tag{68}$$

The SoC boundaries, $SoC_{i,t}^{min}$ and $SoC_{i,t}^{max}$, should be in accordance with the recommendations of the provider to preserve the ESS life spam. Regarding the ESS power limits, $p_{Batt,chm}\ and\ p_{Batt,dischm}$, the charging and discharging processes also should be kept below the recommended values to avoid damaging the ESS. The model includes a binary variable, $x_{bat,t}$, that establishes the status of the ESS, i.e. the variable is 1 when the ESS is being charged and is 0 when it is discharged. In this way, the ESS is modeled in a way that is not charged and discharged simultaneously.

2.5.3.3. SoC equalization performance

The IoT devices continuously track the load consumption patterns and update the information to the CREMS to solve the EPDP in almost real-time to verify the performance of the proposed optimization problem within the MGT. With the updated information, the CREMS recalculates the dispatched power for the IDG in a rolling-horizon scheme. Figure 36 shows one PV generation profile that the IDG units used for testing the operation in a one-week timeframe.

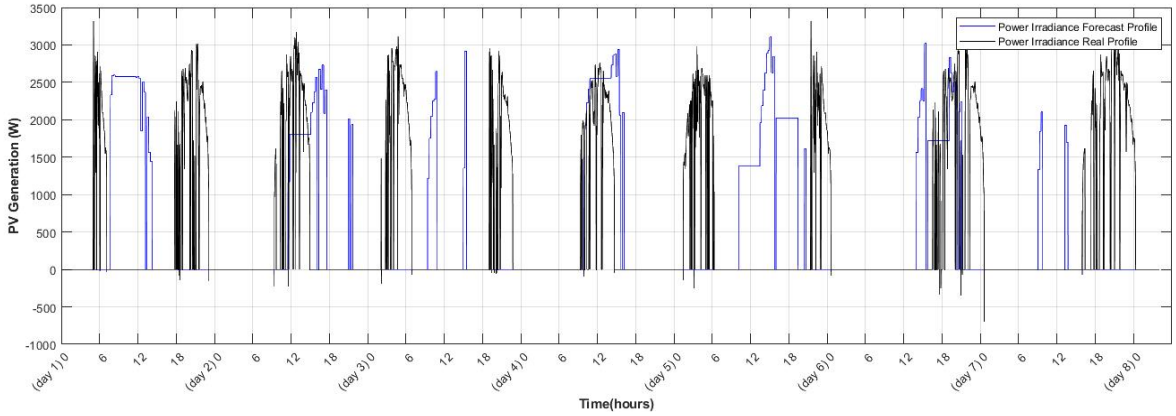


FIGURE 36. PV Generation Profile. Source: [61]

Similarly, figure 37 shows two profiles related to the load: the real-load profile and the forecast-load profile. The first comes from the IoT devices previously presented, and the second results from the ML process performed by the CREMS. The load profiles represent the three-house energy consumption in a one-week timeframe. Also, figure 37 shows the added generation from all the IDGs (RESS power) and the power exchanged with the utility grid. Regarding temporization, this test considered a 30-minute time step for the profiles and a one-week window repeated every 15 minutes for the rolling horizon.

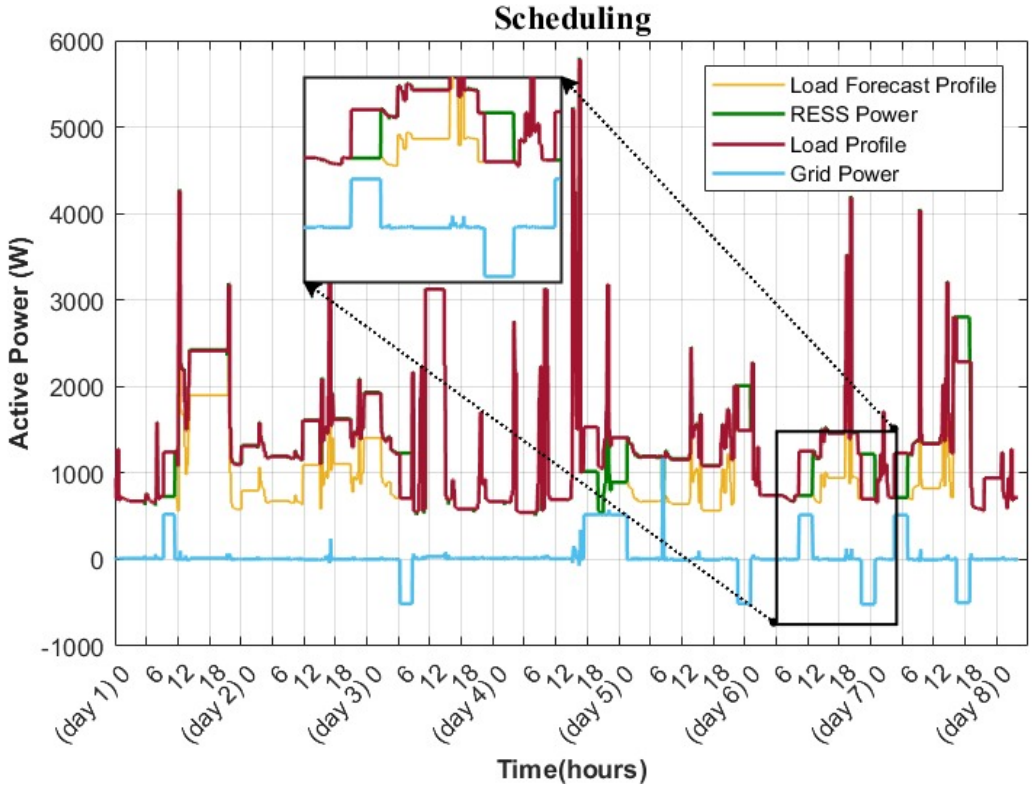


FIGURE 37. Experimental generation and consumption schedule results. Source: [61]

During the experiment, the real-load profile modifies according to the predicted-load profile. It happens by intentionally employing a variable load whose behavior is tracked constantly by the CREMS. The experiment design considers that the generation dispatched from the CREMS will always try to match the consumption. However, under a sudden load variation, there would be a latency caused by the time required by the IoT devices to update the measured data, which is close to one minute, the communication delay, less than one minute, and the time needed by the CREMS to solve the dispatch under a rolling-horizon approach, about 3 minutes. So, there is an inherent time mismatch between generation and consumption due to various factors, basically four minutes, where the utility grid needs to compensate for the discrepancy, as presented in figure 37 by the blue line. This also means having a 10% improvement (34 versus 38 seconds for each RESS), according to the details given in [74]. Based on the AWS reserved instances pricing [75], the CC total operation cost per year in the AWS EC2 platform comes to USD 220. This includes a t4.medium instance in a standard one-year term. It considers that the EC2 system can serve up to 15 processes simultaneously. So the cost of one microgrid per year is USD 14.66. Something similar to the one presented in [74].

Finally, figure 38 shows the results of the SoC equalization process between batteries. This experiment contemplates intentionally different initialization values for the SoC in all the IDGs (90%, 50%, and 35%, as observed in figure 38). As illustrated, the SoC equalization gets complete within days four and five - third time spot, remaining equalized during the rest of the experiment to the value defined by the CREMS following the manufacturer's recommendations to extend the battery's life span [76].

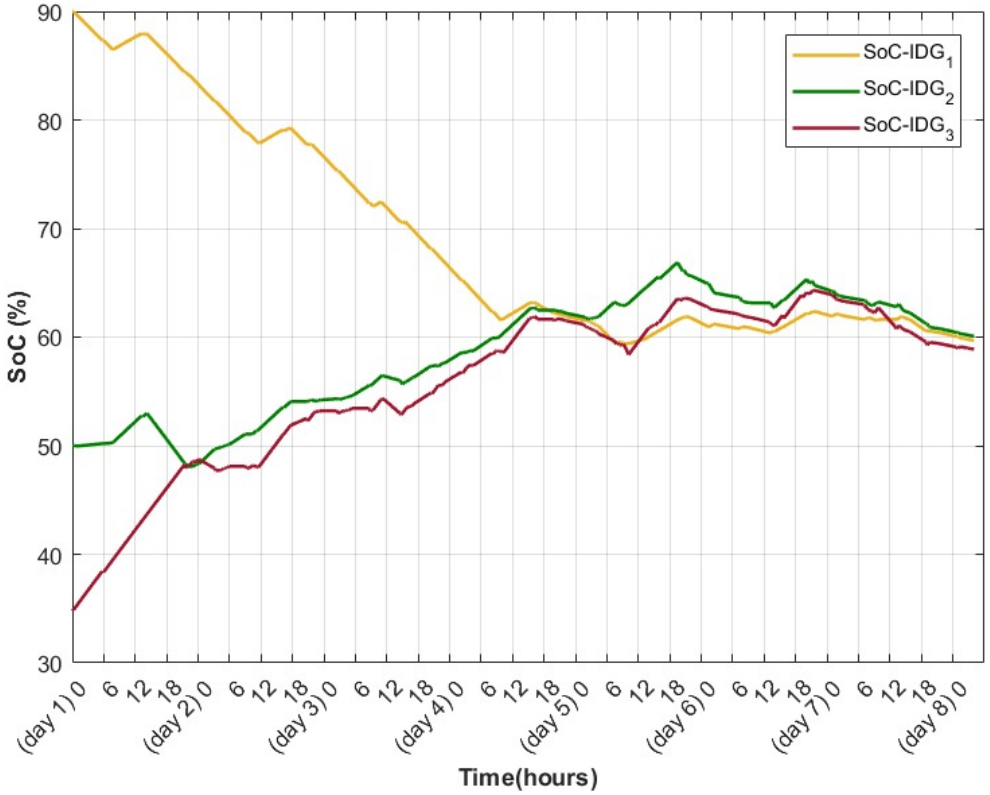


FIGURE 38. SoC equalization. Source: [61]

The SOC equalization time depends on various elements but rests on the available power in the RESS and the load profile associated with the time frame analyzed, and it is notorious that the equalization process moves as soon as the CREMS assumes control and follows the expected at the tertiary level within hours and days. Comparing this outcome with other authors' results is not feasible because, to our knowledge, no similar experiments are testing a real-life PHIL within the CREMS. However, most of the reviewed literature tries to maintain the SOC of the battery within an appropriate range (for instance, 20% to 60%) during the consecutive energy dispatch process, as indicated in [77] or others like [78] and [79], not using ML or IoT, reach the SOC equalization to the four time-spot, one time-spot later that the behavior presented in figure 38. This indicates a 25% improvement, considering the equalization time for the CREMS.

2.5.4. Section conclusion

Unlike previous approaches, the system optimization algorithm executes in an MPS instance using CC, ML, and IoT. This approach improves CREMS performance by eliminating offline optimization. Also, the CC usage improves the findings presented in [29] as the CREMS relieves the utility costs and optimizes the battery usage. As the system can manipulate energy decisions near real-time, the EMS structure adapts conveniently to the amount of MGs available at a specific time, providing an initial assessment to the energy administrator and the final user. As proposed, the CREMS optimizes both the energy delivered by each RES and the one supplied by the utility grid to cover the load. Likewise, the utility grid, emulated by the power amplifier, balances the energy demand with the generation in specific simulation periods. It is interesting to see how the ML model specifies the dispatched power according to the forecasted power availability within the specific timeframe. Besides, the IoT tracking helps the batteries' SOC exhibit favorable results as its percentages stay regulated to the restrained values reducing high discharge repercussions, expanding batteries' lifespan, and reducing utility grid usage in the charging process - most of the charging power comes from RES.

As expected, the results presented in this section and the comparison with other authors' outcomes give sufficient evidence to support the specific objective number four defined for the doctoral research.

3. Conclusions

Renewables were the only energy source for which demand increased in 2020 despite the pandemic, while consumption of all other fuels declined [80]. This means that integrating higher shares of renewable energy systems technologies, such as wind and solar photovoltaic, in grid systems is essential for decarbonizing the power sector while continuing to meet the growing energy demand. As a result of the sharply falling costs and supportive policies, RES deployment has expanded dramatically in recent years [81]. However, the inherent variability of wind and solar photovoltaic energy generation raises challenges for power system operators and regulators.

For this reason, this doctoral research defines a scalable and autonomous CREMS framework that joins up maximum capacity generation, power consumption, and EPDP operation under an ML and IoT perspective running in real-time and using cloud resources to administer an MGC. Conveniently, ML capabilities deployment inside the CREMS allows data sampling at much higher intervals improving the generalized one-day ahead

range and decreasing it to minutes-seconds, overcoming the limitations given by data network congestion, storage, and offline computing. The inclusion of ML, IoT, and CC capabilities in the EMS of a cluster of integrated MG facilitates data analysis under a real-time experience and permits the SL model definition beyond offline limitations.

Additionally, the MGT inclusion permitted testing of some framework features like autonomy and scalability. The availability of PHIL interfaces facilitates defining other attributes like cost-effectiveness compared to [77] - close to 50% savings in the overall CC deployment and 70% in the utility grid usage, solving the dispatch under a 4-minute rolling-horizon approach and settling the Ethernet bandwidth requirements - 2.56 Mbit/s. Neither [3], [29], [77] nor [82] defined it. Moreover, the IoT broker inclusion supports the CC deployment as it reduces the required bandwidth, a key factor for accessing CC infrastructure.

Also, the cloud-based autonomous algorithms tested to solve the EPDP secure adequate stability performance and optimal power reference tracking under unexpected load and generation changes in the MGC - a crucial business issue for the renewable energy market growth. In fact, and according to the experimental results, the CREMS framework could expand to the MG secondary level, power management.

Finally, the inclusion of new information and communications technologies such as CC, ML, and IoT and the decentralization of information and management systems bring additional challenges, such as the reliability and vulnerability of the information shared by the distributed units or the centralized management unit. The framework´s security warrants data protection. This key feature becomes a takeaway element for the CC provider and IoT broker selection according to the IEEE STD 2030.7 - 2017 and compared to local server research.

4. Thesis contribution

The main contribution of this doctoral research is defining a CREMS framework that includes the concept of CC, ML, and IoT technologies to solve an EPDP in an MGC. Unlike most previous approaches, the cloud-deployed real-time framework deployed in the MPS [83] solves the EPDP using recurrent MGC data for generation and load energy in real time. Likewise, the experimental verification utilizes a real-life PHIL testbed to validate the functionality of the proposed autonomous and scalable CREMS deployed in the MPS under an AWS environment. The AWS implementation guarantees fast convergence and optimality of the dynamic algorithm as it demands low implementation complexity. This approach also leads to a real-time EMS by eliminating the requirement of the offline

optimization process proposed previously [4], [84]. Additionally, the experimental validation strategy results presented permits to differentiate this research from theoretical perspectives reported in the literature [64], [85],[86]. Finally, this doctoral research corroborates that:

1. The CREMS aims to reduce power costs and optimize battery usage.
2. The CREMS combines prediction, decision-making, CC, IoT, and ML tools.
3. Prediction of the renewable energy generation curve gave preliminary feedback to the market operator.
4. Prediction of the energy consumption curve gave preliminary feedback to the final user.
5. Considering the optimization model used by the EPDP and the tools utilized for real-life implementation, the CREMS can manage the energy decision close to real-time.
6. The CREMS framework expands transparently and autonomously according to the number of MG connected to the system. The final user can create algorithms, package them, and deploy them to the cloud server without recoding or creating custom infrastructure.
7. Users can access the latest version of the analytics automatically because the cloud server runs on a multiprocessor and multicore machine, allowing simultaneous job request processing with low latency.

5. Suggestions and future work

Two research directions are worth the analysis effort for future work and suggestions. It is necessary to assess the behavior of the proposed CREMS in customer-owned cloud-premises and cross-platform commercial scenarios such as Google Cloud [87] or Microsoft Azure [88]. In addition, it would be interesting to include other state-of-the-art technologies in the framework, like blockchain and deep learning capabilities. Blockchain inclusion would support increasing the framework's security considering energy negotiation within the MGC. Likewise, deep learning capabilities could give a different perspective regarding framework autonomy features.

6. Publications

6.1. Book

Gestión y ciberseguridad para microredes eléctricas residenciales

Elvis Eduardo Gaona García, David Gustavo Rosero Bernal, Eduardo Alirio Mojica Nava, César Leonardo Trujillo Rodríguez, Nelson Leonardo Díaz Aldana

Abstract

Electrical microgrids have enabled the integration of distributed energy resources with different generation characteristics, but located locally into a well-defined area. The microgrids look for ensuring the energy supply for a specific load based on local energy resources. This fact has allowed not only the possibility of supplying energy to remote communities, not interconnected to the utility grid, by means of the integration of renewable energy sources. Also, the microgrid concept has become a change in the paradigm of how electricity users interact with the conventional energy grid. All this is possible by taking advantage of the potentiality of heterogeneous distributed generators and their integration within microgrids, always ensuring local consumption based on local resources while reducing the dependence of the utility grid.

Given the above points, and in addition to an increasingly growing concern for reducing the impact of conventional energy sources based on fossil fuels in the environment. Those facts have driven the use of renewable energy sources with the minimum environmental impact such as photovoltaic panels and wind turbine generators within the household domain. In this context, the residential microgrid concept emerges as an alternative for reducing the dependence from the utility grid, looking for an effective reduction on the energy bill, with a minimum environmental impact.

Therefore, in a residential microgrid, it is necessary to ensure proper interaction between heterogeneous and highly variable energy resources with loads of several different characteristics, mainly electronic loads, which should be available to satisfy the requirements of the household owners. To perform this important task, energy management systems are required for coordinating and managing all the distributed energy resources and loads within the residential microgrid, in such a way that reliability of the local energy system is ensured. To do this, the energy management system is supported on dedicated communication channels enabling the communication between the distributed energy resources and loads with the management units. The communication channel enables the definition of set points and control commands for the different loads and distributed generators, and also allows the energy management system to receive information about loads and consumption profiles in order to define the proper control and coordination action. This layer in the management structure is commonly known as the cybernetic layer.
Due to the large amount of information and complexity in the management of this amount of information, energy generation systems in residential microgrids have been supported in

emerging technologies, but already well positioned, such as the Internet of things, advanced measurement infrastructure, or even cloud computing looking for a faster an enhanced information processing and more efficient energy management that responds to user needs. However, the use of new technologies supported in communications systems means a great risk in the management of information and the reliability of the microgrid, due to its interaction with the energy management system. In this case, not only the stability of the electrical system can be compromised. Also, sensitive user information such as consumption preferences, residence time at home, etc. may be compromised, manipulated or obtained from undesired entities. Since, the information can be acquired externally without the consent of the users.

Based on the above, the need to address the cybersecurity of electric micro networks as the main theme of this book is raised. The document addresses the general characteristics for an energy management system in residential microgrids and explores the vulnerability of its communications systems. At the end, an exploration of strategies and architectures that guarantee the cybersecurity of residential microgrids are explored.

Print ISBN: 978-958-787-228-6
Digital ISBN: 978-958-787-229-3

6.2. Journal paper

Cloud and machine learning experiments applied to the energy management in a microgrid cluster

David Gustavo Rosero Bernal, Nelson Leonardo Díaz Aldana,César Leonardo Trujillo Rodríguez

Abstract

The way to organize the generation, storage, and management of renewable energy and energy consumption features has taken relevance in recent years due to demands that define the social welfare of this century. Like demand increases, other factors require grid infrastructure improvement, updates, and opening to other technologies that assuage the final customer needs. Precisely, the interest in renewable energy sources, the constant evolution of energy storage technologies, the continuous research involving microgrid management systems, and the evolution of cloud computing technologies and machine learning strategies motivate the development of this article. Tasks associated with a microgrid cluster like the integration of a considerable number of heterogeneous devices,

real-time support, information processing, massive storage capabilities, security considerations, and advanced optimization techniques usage could take place in an autonomous and scalable energy management system architecture under a machine learning perspective running in real-time and using Cloud resources. This paper focuses on identifying the elements considered by different authors to define a cloud-based architecture and ensure the appropriately supervised learning functionality under a microgrids cluster environment. Namely, it was necessary to revise and run microgrid simulations, real-time simulation platforms usage, connection to a virtual server for microgrid control and set the energy management system using cloud computing and machine learning. Based on the review and considering the scenarios mentioned, this article presents a scalable and autonomous cloud-based architecture that allows power generation forecast, energy consumption prediction, a real-time energy management system using machine learning techniques.

This paper has been published in *Applied Energy*, volume 304, 15 December 2021, 11770

6.3. Journal paper

Full-deployed energy management system tested in a microgrid cluster

David Gustavo Rosero Bernal, Enrique Sanabria Torres, Nelson Leonardo Díaz Aldana, César Leonardo Trujillo Rodríguez, Adriana Carolina Luna Hernández, Fabio Andrade Rengifo

Abstract

Under a hierarchical structure perspective, the integration of tools like the internet of things, cloud computing, and machine learning into a microgrid real-time energy management system involves superior capabilities like an autonomous and scalable design, massive storage capabilities, real-time information analysis and processing, and security issues control, to mention a few. This paper evaluates most of them using a cloud-based real-time energy management system integrated into a real-life hardware-in-the-loop testbed. The proposed cloud-based system is tested by solving an economic power dispatch problem including the equalization of the multiple battery-based energy storage systems interacting within a multi-microgrid environment. The test assessment combined reviewing and running microgrid models, incorporating these models into a real-life hardware-in-the-loop unit, linking the testbed to a cloud server, and merging the energy management system with on-demand computing tools, primarily machine learning and the internet of things. As established by the experimental evidence, this paper cites the benefits

of combining machine learning techniques and internet of things tools with a scalable and autonomous real-life cloud-based energy management system architecture to improve the framework's functionality, enhance the energy forecasting for generation and usage, and cut down the price paid to the service provider.

This paper has been published in *Applied Energy*, volume 334, 15 March 2023, 120674

6.4. International conference paper

Machine Learning Experiments for a Real-Time Energy Management in a Microgrid Cluster

David Gustavo Rosero Bernal, Enrique Sanabria Torres, Fabio Andrade Rengifo, Nelson Leonardo Díaz Aldana, César Leonardo Trujillo Rodríguez

Abstract

In a Microgrid, the integration of many tasks makes possible an adequate energy management system. Jobs involving real-time support and information procession for having an autonomous and scalable management system, and others such as massive storage capabilities and security considerations to guarantee reliability and validity, are a few. This paper considers them to propose a real-time energy management system based on the economic dispatch problem under a cloud-based architecture, ensuring the appropriately supervised learning functionality in a Microgrid cluster. Namely, it was necessary to revise and run Microgrid implementations, integrate real-time simulation platforms, connect to a virtual server from a Microgrid control, and set the energy management system using cloud computing and machine learning. Based on the results, this article presents a scalable and autonomous cloud-computing architecture for a real-time energy management system using machine learning techniques that allows power generation and energy consumption prediction.

This paper has been published in *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*

6.5. International conference paper

Microgrid Structure for testing a Real-Time Energy Management System Model

Enrique Sanabria Torres, David Gustavo Rosero Bernal, Fabio Andrade Rengifo, Nelson Leonardo Díaz Aldana, César Leonardo Trujillo Rodríguez

Abstract

Proper management of the tasks in a microgrid makes the energy management system successful. These tasks are based on analysis, control, and predictions in real-time, which makes the system capable of autonomous and guarantees its reliability and validity. In this paper, an experimental Microgrid testbed is proposed to allow emulating tasks in real-time that involve predicting energy consumption and generation in an emulated Microgrid cluster with an on-cloud energy management system. It involves machine learning methods in order to solve economic dispatch problems. Since the use of the supervised machine learning technique to estimate future consumption and generation in a system, helps the control system verify and modify itself. The Microgrid testbed includes system hardware in the loop interface, hardware power in the loop, and real inverters to have close emulation of the Microgrid cluster. A communication protocol is also implemented to connect to the on-cloud energy management system and the system hardware in the loop.

This paper has been published in *2022 IEEE Applied Power Electronics Conference and Exposition (APEC)*

6.6. National conference paper

Real-Time Energy Management System Supported on Cloud Services for Cluster of Microgrids

David Gustavo Rosero Bernal, Enrique Sanabria Torres, Fabio Andrade Rengifo, Nelson Leonardo Díaz Aldana, César Leonardo Trujillo Rodríguez

Abstract

An innovative microgrid's energy management system demands many features under a hierarchical structure perspective: an autonomous and scalable design, massive storage capabilities, real-time information analysis, and fast-paced processing are a few, and others such as cybersecurity issues to maintain trustworthiness and viability, are a must. This research revised most of them before integrating and deploying the proposed cloud-based real-time energy management system architecture in a real-life scenario. The implementation solved an economic dispatch problem, incorporated internet of things materials, and used suitable machine learning functionalities in an interconnected microgrid assemblage. For this, the authors studied and ran microgrid models, deployed the models into hardware-in-the-loop units, linked the consolidated hardware to a production cloud server, and merged the energy management system with machine learning, and the internet

of things tools. As established by real-life evidence, this research defines relevant aspects for a fully-deployed scalable and autonomous real-time cloud-computing energy management system architecture to optimize energy generation, usage forecasting, and energy trade.

This paper has been accepted in *2022 Workshop on Engineering Applications (WEA)*

REFERENCES

[1] D. Manz, R. Walling, N. Miller, B. LaRose, R. D'Aquila, and B. Daryanian, "The Grid of the Future: Ten Trends That Will Shape the Grid Over the Next Decade," *IEEE Power and Energy Magazine*, vol. 12, no. 3. pp. 26–36, 2014. doi: 10.1109/mpe.2014.2301516.

[2] Y. Parag and B. K. Sovacool, "Electricity market design for the prosumer era," *Nature Energy*, vol. 1, no. 4. 2016. doi: 10.1038/nenergy.2016.32.

[3] C. A. Macana, H. R. Pota, Q. Zhu, J. M. Guerrero, and J. C. Vasquez, "Experiments on a Real-Time Energy Management System for Islanded Prosumer Microgrids," *Electronics*, vol. 8, no. 9. p. 925, 2019. doi: 10.3390/electronics8090925.

[4] A. C. Luna, L. Meng, N. L. Diaz, M. Graells, J. C. Vasquez, and J. M. Guerrero, "Online Energy Management Systems for Microgrids: Experimental Validation and Assessment Framework," *IEEE Transactions on Power Electronics*, vol. 33, no. 3. pp. 2201–2215, 2018. doi: 10.1109/tpel.2017.2700083.

[5] C. Macana, A. Abdou, H. Pota, J. Guerrero, and J. Vasquez, "Cyber Physical Energy Systems Modules for Power Sharing Controllers in Inverter Based Microgrids," *Inventions*, vol. 3, no. 3. p. 66, 2018. doi: 10.3390/inventions3030066.

[6] Y. Han, H. Li, P. Shen, E. A. A. Coelho, and J. M. Guerrero, "Review of Active and Reactive Power Sharing Strategies in Hierarchical Controlled Microgrids," *IEEE Transactions on Power Electronics*, vol. 32, no. 3. pp. 2427–2451, 2017. doi: 10.1109/tpel.2016.2569597.

[7] "Cloud Computing Services." https://azure.microsoft.com/en-us/ (accessed Jun. 22, 2021).

[8] E. Rincón, "EN NARIÑO LE APUESTAN A LA ENERGÍA ALTERNATIVA," 2018. https://electrica.uniandes.edu.co/es/home/9-espanol/departamento/noticias/589-en-narino-le-apuestan-a-la-energia-alternativa (accessed Mar. 04, 2019).

[9] Celsia, "Granjas Solares," 2016. https://www.celsia.com/es/Granjas-Solares (accessed Mar. 04, 2019).

[10] reve, "Energía Eólica de La Guajira para Colombia | REVE – Revista Eólica y del Vehículo Eléctrico," 2019. https://www.evwind.com/2019/03/28/energia-eolica-de-la-guajira-para-colombia/ (accessed Apr. 21, 2019).

[11] J. Brodersen and B. Kommunikation, "Unleashing the power of the microgrids," Apr. 02, 2019. https://www.news.aau.dk/news//unleashing-the-power-of-the-microgrids.cid402952 (accessed Apr. 08, 2019).

[12] "Grupo de Investigación LIFAE (Laboratorio de Investigación de Fuentes Alternativas de Energía)," 2001. https://comunidad.udistrital.edu.co/lifae/ (accessed Apr. 21, 2019).

[13] "Homepage," Apr. 08, 2019. https://inec.uprm.edu/sec/ (accessed Jun. 22, 2021).

[14] Navigant Research, "Microgrids," 2019. https://www.navigantresearch.com/research-solutions/microgrids (accessed Apr. 08, 2019).

[15] "Inter-American Development Bank." https://www.iadb.org/ (accessed Jun. 22, 2021).

[16] "UPME." https://www1.upme.gov.co/Paginas/default.aspx (accessed Jun. 22, 2021).

[17] UPME, "Estudio: Smart Grids Colombia Visión 2030 - Mapa de ruta para la implementación de redes inteligentes en Colombia," 2016. http://www1.upme.gov.co/Paginas/Smart-Grids-Colombia-Visi%C3%B3n-2030.aspx (accessed Apr. 08, 2019).

[18] J. H. Torres Acosta, "Inteligencia integral de negocios," *Doctorado en Ingeniería*. p. 15, 2017.

[19] "MATLAB Production Server." https://la.mathworks.com/products/matlab-production-server.html (accessed Jun. 22, 2021).

[20] N. Mohamed, J. Al-Jaroodi, I. Jawhar, H. Noura, and S. Mahmoud, "UAVFog: A UAV-based fog computing for Internet of Things," in *2017 IEEE SmartWorld, Ubiquitous Intelligence &*

Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 2017.
[21] L. Wang and E. Gelenbe, "Adaptive Dispatching of Tasks in the Cloud," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 33–45, 2018.
[22] IEEE Standards Association, "IEEE Standard for the Specification of Microgrid Controllers." 2017. doi: 10.1109/ieeestd.2018.8340204.
[23] "Real-Time simulation - Real-Time Solutions," *OPAL-RT*, Mar. 02, 2016. https://www.opal-rt.com/home/ (accessed Oct. 25, 2022).
[24] "Home." https://inec.uprm.edu/sec/ (accessed Oct. 25, 2022).
[25] "AWS," *Amazon Web Services, Inc.* https://aws.amazon.com/es/ (accessed Oct. 25, 2022).
[26] "JSON." https://www.json.org/json-en.html (accessed Oct. 25, 2022).
[27] "MQTT - The Standard for IoT Messaging." https://mqtt.org/ (accessed Oct. 25, 2022).
[28] D. G. Rosero, N. L. Díaz, and C. L. Trujillo, "Cloud and machine learning experiments applied to the energy management in a microgrid cluster," *Applied Energy*, vol. 304. p. 117770, 2021. doi: 10.1016/j.apenergy.2021.117770.
[29] D. Rosero-Bernal, E. A. Sanabria-Torres, F. Andrade-Rengifo, N. L. Diaz, and C. L. Trujillo, "Machine Learning Experiments for a Real-Time Energy Management in a Microgrid Cluster," in *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2021, pp. 1–6.
[30] S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*. Academic Press, 2015.
[31] P. Dangeti, *Statistics for Machine Learning*. Packt Publishing Ltd, 2017.
[32] S. Theodoridis, A. Pikrakis, K. Koutroumbas, and D. Cavouras, *Introduction to Pattern Recognition: A Matlab Approach*. Academic Press, 2010.
[33] M. Paluszek and S. Thomas, "Machine Learning Examples in MATLAB," *MATLAB Machine Learning*. pp. 85–88, 2017. doi: 10.1007/978-1-4842-2250-8_6.
[34] "Detalle." https://editorial.udistrital.edu.co/detalle.php?id=1288&f=6 (accessed Nov. 03, 2022).
[35] G. Bontempi, "'Statistical foundations of machine learning' (2nd edition) handbook," Feb. 2021, Accessed: May 18, 2021. [Online]. Available: http://dx.doi.org/
[36] R. J. A. Little and D. B. Rubin, "Statistical Analysis with Missing Data." 2002. doi: 10.1002/9781119013563.
[37] H. Läuter, "Hand, D. J.: Discrimination and Classification. John Wiley & Sons, Chichester — Brigbane — New York — Toronto 1981. XII, 218 pp., $ 15.50," *Biometrical Journal*, vol. 27, no. 2. pp. 148–148, 1985. doi: 10.1002/bimj.4710270204.
[38] "Data-Driven Insights with MATLAB Analytics: An Energy Load Forecasting Case Study." https://la.mathworks.com/company/newsletters/articles/data-driven-insights-with-matlab-analytics-an-energy-load-forecasting-case-study.html (accessed Oct. 26, 2022).
[39] "MATLAB." https://www.mathworks.com/products/matlab.html (accessed Jul. 07, 2021).
[40] "Home - NSRDB." https://nsrdb.nrel.gov/ (accessed Jul. 07, 2021).
[41] R. A. Irizarry, "Chapter 28 Smoothing." http://rafalab.dfci.harvard.edu/dsbook/smoothing.html (accessed Oct. 25, 2022).
[42] "How the Gen 2 Vue Energy Monitor Works." https://www.emporiaenergy.com/how-the-vue-energy-monitor-works (accessed Jul. 08, 2021).
[43] "AWS." https://aws.amazon.com/es/s3/ (accessed Jul. 08, 2021).
[44] R. J. Hyndman and G. Athanasopoulos, "9.6 Lagged predictors." http://Otexts.org/fpp2/ (accessed Oct. 26, 2022).
[45] S. Vajda and R. Bellman, "Dynamic Programming," *Econometrica*, vol. 27, no. 3. p. 537,

1959. doi: 10.2307/1909506.
[46] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. Springer Science & Business Media, 1997.
[47] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, no. none. 2010. doi: 10.1214/09-ss054.
[48] E. A. Sanabria-Torres, D. Rosero-Bernal, N. L. Diaz, C. L. Trujillo, and F. Andrade-Rengifo, "Microgrid Structure for testing a Real-Time Energy Management System Model," *2022 IEEE Applied Power Electronics Conference and Exposition (APEC)*. 2022. doi: 10.1109/apec43599.2022.9773556.
[49] B. Bendib, F. Krim, H. Belmili, M. F. Almi, and S. Boulouma, "Advanced Fuzzy MPPT Controller for a Stand-alone PV System," *Energy Procedia*, vol. 50. pp. 383–392, 2014. doi: 10.1016/j.egypro.2014.06.046.
[50] T. Bogaraj, J. Kanakaraj, and J. Chelladurai, "Modeling and simulation of stand-alone hybrid power system with fuzzy MPPT for remote load application," *Archives of Electrical Engineering*, vol. 64, no. 3. 2015. doi: 10.2478/aee-2015-0037.
[51] J. Kitson *et al.*, "A photovoltaic panel modelling method for flexible implementation in Matlab/Simulink using datasheet quantities," *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*. 2017. doi: 10.1109/isie.2017.8001373.
[52] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*. SIAM, 2003.
[53] E. Dursun and O. Kilic, "Comparative evaluation of different power management strategies of a stand-alone PV/Wind/PEMFC hybrid power system," *International Journal of Electrical Power & Energy Systems*, vol. 34, no. 1. pp. 81–89, 2012. doi: 10.1016/j.ijepes.2011.08.025.
[54] Gaona, E. et Al. "Gestión y ciberseguridad para microrredes eléctricas residenciales." https://doctoradoingenieria.udistrital.edu.co/libroGestCibersegParaMircorredElectResiden (accessed Jun. 09, 2021).
[55] T. Kim and W. Qiao, "A hybrid battery model capable of capturing dynamic circuit characteristics and nonlinear capacity effects," *2012 IEEE Power and Energy Society General Meeting*. 2012. doi: 10.1109/pesgm.2012.6345454.
[56] M. R. Jongerden and B. R. Haverkort, "Which battery model to use?," *IET Software*, vol. 3, no. 6. p. 445, 2009. doi: 10.1049/iet-sen.2009.0001.
[57] "OP1400," *OPAL-RT*, Aug. 30, 2019. https://www.opal-rt.com/op1400-4-quadrant-power-amplifier/ (accessed Aug. 24, 2022).
[58] "SCALEXIO." https://www.dspace.com/en/pub/home/products/hw/simulator_hardware/scalexio.cfm (accessed Aug. 31, 2022).
[59] "OP1400," Aug. 30, 2019. https://www.opal-rt.com/op1400-4-quadrant-power-amplifier/ (accessed Jul. 06, 2021).
[60] "VLT® AutomationDrive FC 301 / FC 302." https://www.danfoss.com/en/products/dds/low-voltage-drives/vlt-drives/vlt-automationdrive-fc-301-fc-302/ (accessed Jul. 06, 2021).
[61] D. G. Rosero, E. Sanabria, N. L. Díaz, C. L. Trujillo, A. Luna, and F. Andrade, "Full-deployed energy management system tested in a microgrid cluster," *Applied Energy*, vol. 334. p. 120674, 2023. doi: 10.1016/j.apenergy.2023.120674.
[62] L. Lei, Y. Tan, G. Dahlenburg, W. Xiang, and K. Zheng, "Dynamic energy dispatch based on deep reinforcement learning in IoT-driven smart isolated microgrids," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 7938–7953, May 2021.
[63] B. Imtiaz, Y. Cui, and I. Zafar, "Economic Dispatch of Microgrid Incorporating Demand Response Using Dragonfly Algorithm," *2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*. 2021. doi:

10.1109/aeeca52519.2021.9574430.
[64] E. Mojica-Nava, C. A. Macana, and N. Quijano, “Dynamic Population Games for Optimal Dispatch on Hierarchical Microgrid Control,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 3. pp. 306–317, 2014. doi: 10.1109/tsmcc.2013.2266117.
[65] “Seguridad en la Nube - Amazon Web Services (AWS),” *Amazon Web Services, Inc.* https://aws.amazon.com/es/security/ (accessed Mar. 02, 2023).
[66] J. F. Patarroyo-Montenegro, J. D. Vasquez-Plaza, F. Andrade, and L. Fan, “An Optimal Power Control Strategy for Grid-Following Inverters in a Synchronous Frame,” *Applied Sciences*, vol. 10, no. 19. p. 6730, 2020. doi: 10.3390/app10196730.
[67] Aws Cloud Computing, “Servicios de informática en la nube,” *Amazon Web Services, Inc.*, 2019. https://aws.amazon.com/es/ (accessed Apr. 06, 2019).
[68] “AWS,” *Amazon Web Services, Inc.* https://aws.amazon.com/es/s3/ (accessed Aug. 24, 2022).
[69] “The Design of a Novel Smart Home Control System using Smart Grid Based on Edge and Cloud Computing.” https://ieeexplore.ieee.org/document/9181961 (accessed Oct. 28, 2022).
[70] “Workshop on Engineering Applications (WEA 2022) – Artificial intelligence, Optimization and Simulation.” https://ieee.udistrital.edu.co/wea2022/ (accessed Nov. 03, 2022).
[71] A. C. Luna, N. L. Diaz, M. Savaghebi, J. C. Vasquez, and J. M. Guerrero, “Optimal power scheduling for an islanded hybrid microgrid,” *2016 IEEE 8th International Power Electronics and Motion Control Conference (IPEMC-ECCE Asia)*. 2016. doi: 10.1109/ipemc.2016.7512565.
[72] Rosero, D.G., Sanabria, E., Díaz, N.L., Trujillo, C. L., Andrade, F., “Real-Time Energy Management System Supported on Cloud Services for Cluster of Microgrids,” in *2022 Workshop on Engineering Applications (WEA)*, Bogota, 2022, p. 5.
[73] N. L. Diaz, J. C. Vasquez, and J. M. Guerrero, “A Communication-Less Distributed Control Architecture for Islanded Microgrids With Renewable Generation and Storage,” *IEEE Transactions on Power Electronics*, vol. 33, no. 3. pp. 1922–1939, 2018. doi: 10.1109/tpel.2017.2698023.
[74] S. Wang, X. Wang, and W. Wu, “Cloud computing and local chip-based dynamic economic dispatch for microgrids,” *IEEE Trans. Smart Grid*, vol. 11, no. 5, pp. 3774–3784, Sep. 2020.
[75] “AWS Pricing Calculator.” https://calculator.aws/#/ (accessed Oct. 11, 2022).
[76] “Battery Lifespan.” https://www.nrel.gov/transportation/battery-lifespan.html (accessed Feb. 21, 2023).
[77] W. Dong, Q. Yang, W. Li, and A. Y. Zomaya, “Machine-learning-based real-time economic dispatch in islanding microgrids in a cloud-edge computing environment,” *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13703–13711, Sep. 2021.
[78] T. A. Fagundes, G. H. F. Fuzato, P. G. B. Ferreira, M. Biczkowski, and R. Q. Machado, “Fuzzy controller for energy management and SoC equalization in DC microgrids powered by fuel cell and energy storage units,” *IEEE J. Emerg. Sel. Top. Ind. Electron.*, vol. 3, no. 1, pp. 90–100, Jan. 2022.
[79] L. Zhang and T. Li, “SOC Dynamic Balance Strategy Based on Adaptive Droop Control with Variable Regulating Factor,” *2021 3rd Asia Energy and Electrical Engineering Symposium (AEEES)*. 2021. doi: 10.1109/aeees51875.2021.9403165.
[80] “Renewable Energy Market Update 2021.” https://www.iea.org/reports/renewable-energy-market-update-2021 (accessed Jun. 23, 2021).
[81] “2021 Renewable Energy Industry Outlook,” Dec. 10, 2018. https://www2.deloitte.com/us/en/pages/energy-and-resources/articles/renewable-energy-outlook.html (accessed Jun. 23, 2021).
[82] “Data-Driven Insights with MATLAB Analytics: An Energy Load Forecasting Case Study.” https://la.mathworks.com/company/newsletters/articles/data-driven-insights-with-matlab-analy

tics-an-energy-load-forecasting-case-study.html (accessed Aug. 29, 2022).
[83] “MATLAB Production Server.” https://la.mathworks.com/products/matlab-production-server.html (accessed Jun. 16, 2021).
[84] N. I. Nwulu and X. Xia, “Optimal dispatch for a microgrid incorporating renewables and demand response,” *Renewable Energy*, vol. 101. pp. 16–28, 2017. doi: 10.1016/j.renene.2016.08.026.
[85] L. Meng, E. R. Sanseverino, A. Luna, T. Dragicevic, J. C. Vasquez, and J. M. Guerrero, “Microgrid supervisory controllers and energy management systems: A literature review,” *Renewable and Sustainable Energy Reviews*, vol. 60. pp. 1263–1273, 2016. doi: 10.1016/j.rser.2016.03.003.
[86] Y. Gao, J. Li, and M. Hong, “Machine Learning Based Optimization Model for Energy Management of Energy Storage System for Large Industrial Park,” *Processes*, vol. 9, no. 5. p. 825, 2021. doi: 10.3390/pr9050825.
[87] “Servicios de cloud computing,” *Google Cloud*. https://cloud.google.com/?hl=es (accessed Mar. 02, 2023).
[88] “Cloud Computing Services.” https://azure.microsoft.com/en-us (accessed Mar. 02, 2023).