Applied Energy 334 (2023) 120674

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Full-deployed energy management system tested in a microgrid cluster

D.G. Rosero [a,*], E. Sanabria [b], N.L. Díaz [a], C.L. Trujillo [a], A. Luna [c], F. Andrade [c]

[a] *Doctorado en Ingeniería, Universidad Distrital Francisco José de Caldas, Colombia*
[b] *Research and Development Department - Sonnen Inc, USA*
[c] *Universidad de Puerto Rico, Puerto Rico, USA*

HIGHLIGHTS

- Cloud-based technologies adoption in real-time microgrid management.
- Boosting the microgrid data forecasting by machine-learning techniques.
- Impact of the internet of things usage in data acquisition for a real-life energy management system.

ARTICLE INFO

Keywords:
Real-time
Real-life
Microgrid cluster
Testbed
Energy management system
Machine learning
Internet of things

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 power-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.

1. Introduction

The notable universal strategy that encourages renewable energy sources (RESs) integration [1] and the battery energy storage systems (BESS) evolution [2] has expanded the microgrid (MG), the microgrid cluster (MGC), and the energy management system (EMS) perspective. Certainly, state-of-the-art technologies application in an MGC could facilitate features like energy generation forecasting, load management estimation, real-time operation, and self-sustaining functionality, which leads to relevant opportunities for regional energy trading. Under this approach, methodical and cost-effective MGC management [3] needs to consider planning, designing, and operational demands [4].

This paper details the performance of the cloud-based real-time EMS (CREMS) architecture when the assessment employs a real-life power-hardware-in-the-loop (PHIL) testbed [5], improves the machine learning (ML) [6] algorithms and the optimization problem and includes the internet of things (IoT) resources [7]. Usually, the MG control defines proportional power-sharing as the principal goal. However, this approach does not include any economic operation criteria. In this research, the Microgrid Central Controller (MGCC) [8], represented by the CREMS, solves the Economic Dispatch Problem (EDP) to warrant the energy balance in the system and control the load in the MGC. The CREMS performs an online process on the tertiary control level using short time scales (order of minutes). As the EDP process executes near real-time, it results in optimal operation points that minimize the total

* Corresponding author.
E-mail addresses: dgroserob@udistrital.edu.co (D.G. Rosero), e.sanabria@sonnen-batterie.com (E. Sanabria), nldiaza@udistrital.edu.co (N.L. Díaz), cltrujillo@udistrital.edu.co (C.L. Trujillo), adriana.luna4@upr.edu (A. Luna), fabio.andrade@upr.edu (F. Andrade).

https://doi.org/10.1016/j.apenergy.2023.120674
Received 11 October 2022; Received in revised form 19 December 2022; Accepted 7 January 2023
Available online 23 January 2023
0306-2619/

Nomenclature

Variables

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

Abreviatures

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

energy cost avoiding changes in the load demand or the RES generation [9] while equalizing the state of charge (SoC) of the Energy Storage System (ESS) integrated to each i-th IDG. Previous EMS proposals require accurate total load demand and available RES generation information in each optimization period, in addition to load and generation forecasting incorporation, which is a complicated process in terms of software requirements [4]. In contrast with [10], the ultimate CREMS integrates data acquisition and generation and consumption forecasting under a cost-effective procedure beyond routinary cloud computing (CC) [11] using smart meters. It is essential to define a suitable technique to organize the IoT data arriving from devices placed in the distributed RESs as the results show that the load profile variations are detected, and the CREMS adjusts the dispatch of the microgrid. Also, it is required to determine the ML incorporation procedure to forecast generation and consumption attributes in a specific spot. Likewise, it is demanding to test the CREMS in a real-life PHIL testbed to validate the ML, IoT, and CC appliance's applicability in an MGC, in terms of minimizing the use of the utility grid and maximizing the use of renewable resources.

As characterized in [12], the CREMS divides into three sections (see Fig. 1): supervisory system (SS), data acquisition, storage & processing (DASP), and scheduling and optimization (S&O). In this paper, the SS section uses a Matlab Production Server (MPS) [13] 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) [14]. 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 architecture proposed in Fig. 1 sustains its implementation on cloud services, in particular, Amazon Elastic Compute Cloud (EC2) [15] but could extend to other platforms like Google Cloud and Microsoft Azure or consider a hybrid solution. This architecture evolves the MPS structure, considering the MGC management requirements, to effectively use advanced management procedures at the hierarchical structure's strategic level and maintain the MGC's secure process covering various functionality scenarios - the top decision layers specify the assignments and regulate lower layers without ignoring their decisions.

This research has considered various conditions to verify the CREMS capability to manage an MGC: First, the CC technology used to integrate the CREMS into the MGC control. Second, the learning algorithm capacities inclusion. Third, the CREMS competence to handle an MGC. Four, the IoT data collection techniques manage in a real-life production scenario. Five, the type of testbed used to validate the research results. Six, the optimization methodology used to warrant the energy balance and, finally, the measurement of the real-time latency. The proposals reported in Table 1 compare other authors' results with the ones developed in this paper. Nevertheless, those experiments did not settle a real-life CREMS framework extended to an MGC, they do indicate accelerated CC, IoT, and ML incorporation into MG's EMS. Also, the table shows that this issue has been less analyzed, and its experimentation within a real-life hardware-in-the-loop (HIL) or PHIL testbed is minimal - this argument motivated this experiment too.

The paper covers the following sections: Section II presents the materials and methods applied in this research. Section III outlines the CREMS architecture. Next, in section IV, the paper explains the implemented optimization model. Section V illustrates the ML procedure. Later, section VI details the real-life CREMS implementation. Section VII

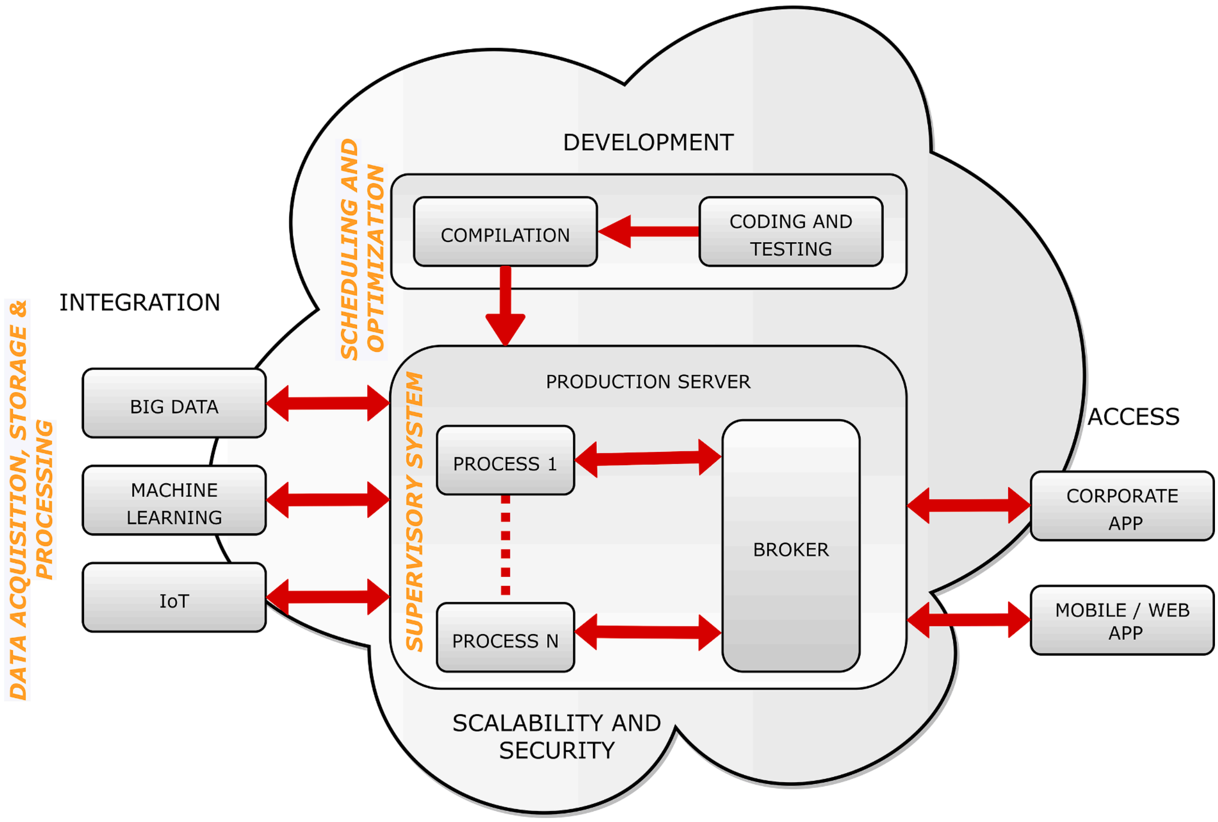


Fig. 1. Conceptual CREMS. Source: Authors.

Table 1
Related real-life CREMS usage.

No.	Name	CC technology	ML type	MGC handling	IoT deployment	Testbed appeal	Optimization	Real-time latency measurement
1	This paper	AWS	Supervised Learning	Framework defined	Commercial broker	HIL, HPIL	Online	Available, four minutes for WAN
2	[16]	Local server	Predict Control	Compatible	Open source	Microgrid building	Online	N/A
3	[17]	Local server	Fuzzy adaptive method	N/A	N/A	Dispatcheable and non-dispatchable sources	Online	N/A
4	[18]	Azure	Supervised Learning	Framework defined	Raspberry Pi boards	N/A	N/A	N/A
5	[19]	Local server	Supervised Learning	Framework defined	N/A	N/A	Online	Available, five minutes for LAN
6	[20]	Local server	Supervised Learning	N/A	Open source	264 Kilowatt solar farm	Online	N/A
7	[21]	Local server	Supervised Learning	N/A	Matlab engine	N/A	Online	Available, five minutes for LAN
8	[3]	Local server	N/A	Framework defined	N/A	Microgrid building	Online	Available, 20 s for LAN
9	[22]	ThingSpeak and local server	N/A	Framework defined	Raspberry Pi boards	Scada controls	Online	Available, 0.5 s for LAN
10	[23]	Local server	Kernel Based Robust Random VectorFunctional Link Network	Framework defined	N/A	Ecosense PV emulator	Online	N/A
11	[24]	Local server	Explicit Model Predictive Control	Compatible	N/A	Sonnen Hybrid	Offline / online	N/A

analyzes the CREMS performance. And lastly, section VIII concludes this paper.

2. Material and methods

The authors consulted the following databases to consolidate this article: Science Direct, IEEE Xplore, and Scopus. Table 2 summarizes the keywords used in the search process in each database.

Fig. 2 exhibits the consolidation of the number of articles published using the search keywords: Microgrids, Cluster of microgrids, Cloud

Table 2
Keywords selected in the methodology.

Category	Keywords
Application	Microgrid, Cluster of Microgrids, Cloud Computing, IoT, Energy Management System, Testbed
Implementation	Real-time, Real-life, Testbed, EMS, Microgrids, Cluster of Microgrids
Integration	Machine Learning, Cloud Computing, IoT, Testbed, Real-Time
Autonomy	Machine Learning, IoT, Cloud Computing, EMS, Microgrids, Testbed

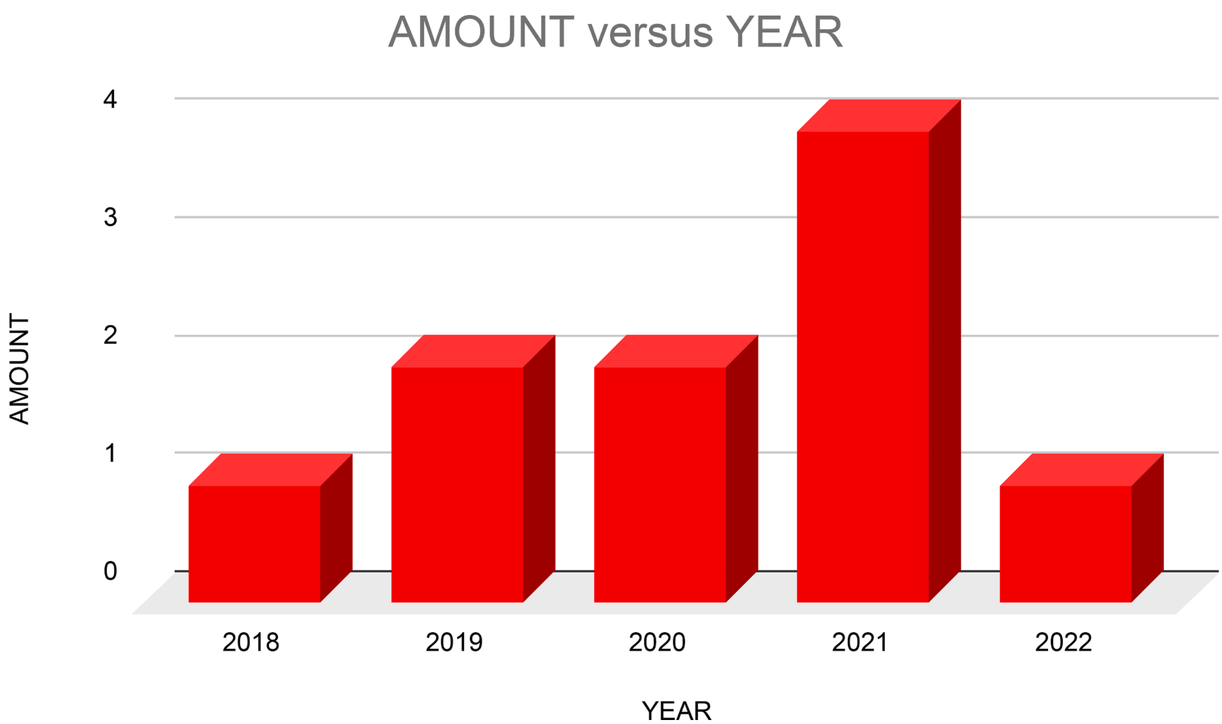


Fig. 2. Published articles. Source: Authors.

Computing Technology, Machine Learning Technology, IoT deployment, Testbed, Optimization, Real-Time, and Real-Life.

The selected papers are associated with the engineering environment and renewable and sustainable energy. It is clear that the number of publications on cloud computing and machine learning applied to microgrids has evolved recently, but just a few consider the conceptual CREMS as done in this paper. After an exhaustive review of the literature, this research identifies a real-life CREMS that incorporates state-of-the-art technologies in its implementation.

3. The CREMS architecture

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 MCG management execution. Lastly, the SS unit enhances the framework's capabilities to obtain low latency processing, expansion capability, and redundancy required by the CREMS. 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 [5,16], 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. Fig. 3 depicts the enhanced hierarchical

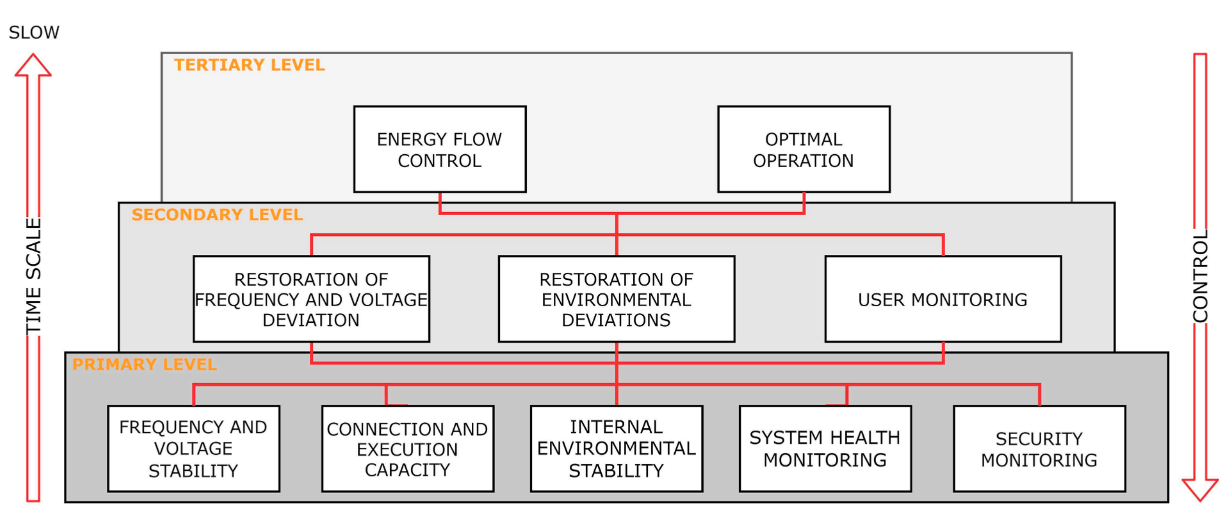


Fig. 3. Enhanced hierarchical structure. Source: Authors.

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.

Fig. 4 illustrates an overall scheme of the real-life PHIL testbed available in the Sustainable Energy Center (SEC) of the University of Puerto Rico at Mayagüez [25]. 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 [26]. 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.

4. Optimization problem

The optimization problem considered in this paper aims at minimizing operating costs by solving a typical economic power dispatch problem (EPDP) while equalizing the state of charge (SoC) of the ESSs integrated to each i-th IDG by means of a Mixed Integer Linear Problem (MILP). Additionally, in order to avoid uneven degradation of the BESS, it is recommended to perform an SoC equalization during the operation of the electrical system [27]. In this way, when a battery is replaced or the system is partially disconnected and reconnected, the management system will ensure SoC equalization. To include the equalization of the SoCs inside 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:

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

The equalization is achieved by minimizing this variable, but it must be positive to reach a feasible solution, which can be written as:

$$\left(SoC_{i,t} - SoC_{i+1,t}\right) \geq 0, \forall i \in N-1, t \in T \tag{2}$$

Then, data preprocessing is required to downsorting the SoCs of the ESSs prior to the optimization problem being conducted for all time t. To perform this task, a decision-making stage is included within the optimization process as shown in Fig. 5. In this way, the errors will be calculated between pairs (*k11 - k12*, *k21 - k22*, and *k31 - k32*), and their positivity is guaranteed. These auxiliary indexes are defined to be used in the constraint (2).

4.1. Objective function

To optimize the operating costs and the equalization, the objective function of the optimization problem is modeled as,

$$\sum\nolimits_{i \in N} \sum\nolimits_{t \in T} \left(C_{i,t} p_{i,t}^{forecast} \Delta t + C_{i,t} p_{Grid,t} \Delta t + C_{i,t} E_{SoCi,t}\right) \tag{3}$$

The first two terms in (3) minimizes the cost associated with the usage of energy from the IDGs and the main 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 for using any resource and the penalty coefficient related to the equalization. The coefficients were adjusted to give priority to the minimization of the cost.

4.2. Problem constraints

The first constraint that must be considered is 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

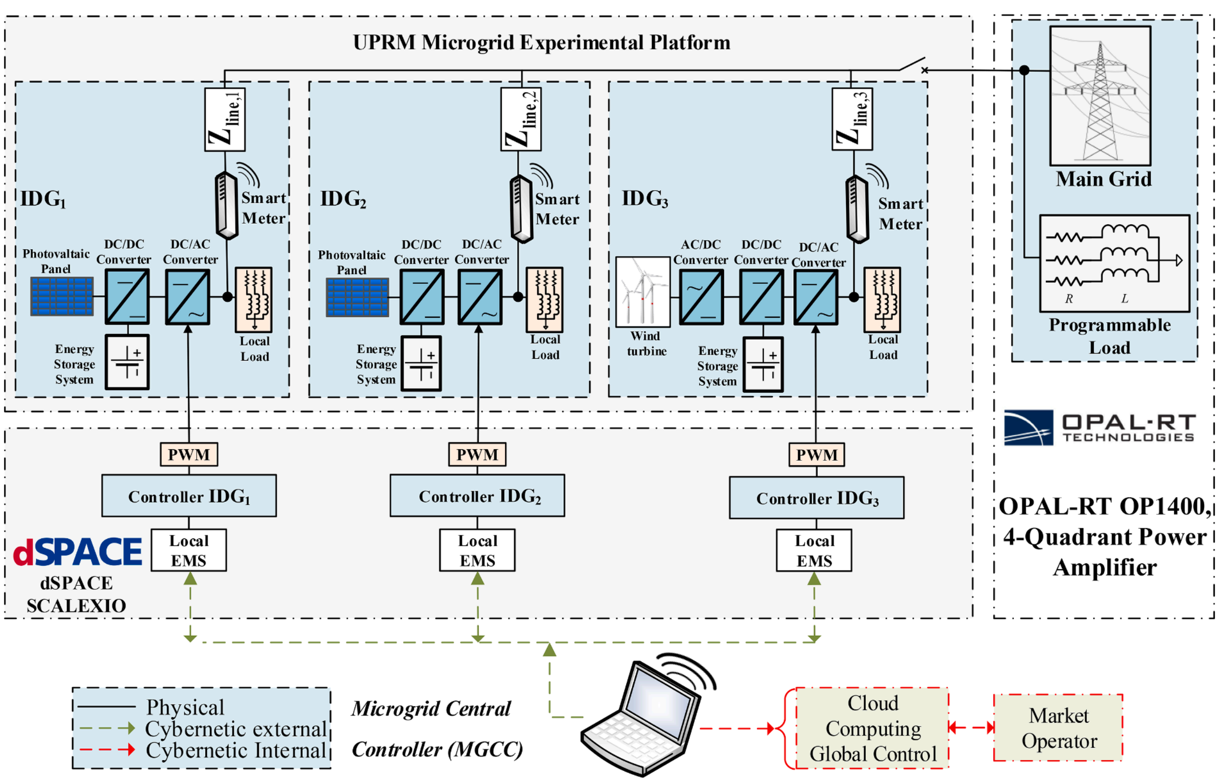


Fig. 4. Tested MGC. Source: Authors.

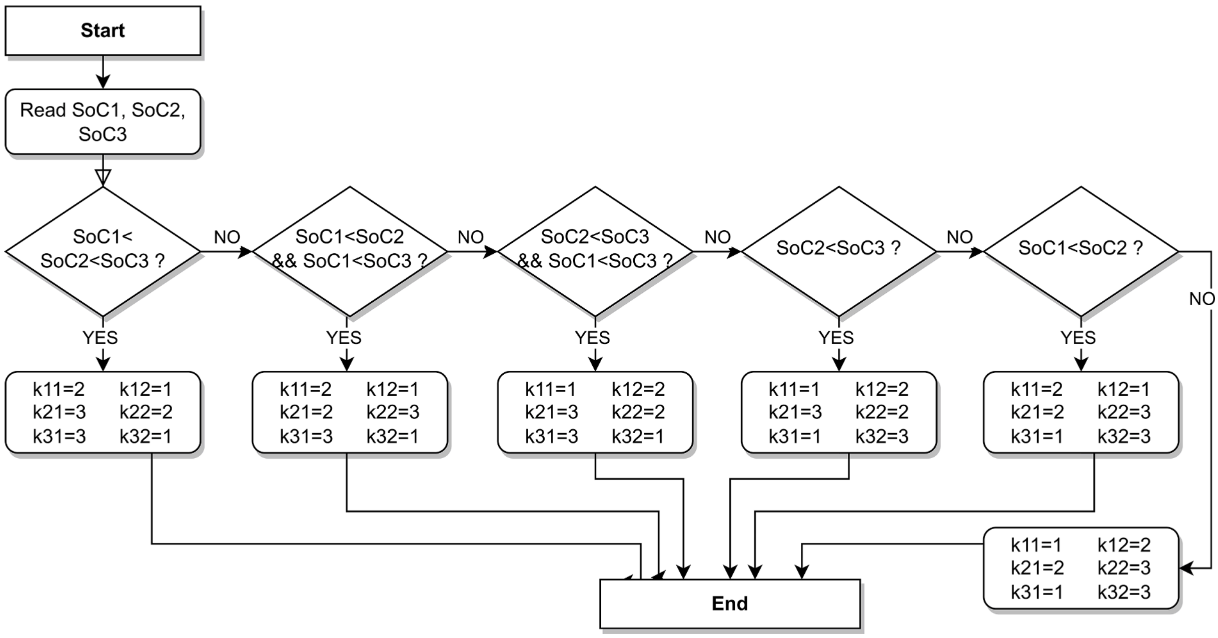


Fig. 5. Flow diagram for the SoC's downstring associated with the SOC error constraint. Source: Authors.

the load. In this scenario, the problem considers two kinds of load, the measured by the smart meters, $p_{d,t}^{smart}$, and the forecasted load $p_{d,t}^{forecast}$. This fact enables the real-time EMS to react online to mismatches between forecasted and real load, and consequently, recalculate the dispatched power for each IDG. Therefore, the energy balance can be written as,

$$\sum\nolimits_{i,N} p_{i,t}^{forecast}\Delta t + p_{Grid,t}\Delta t - p_{d,t}^{forecast}\Delta t - p_{d,t}^{smart}\Delta t = 0, \forall t \tag{4}$$

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, which are modeled as,

$$p_{i,t}^{forecast}\Delta t = p_{RES_{i,t}}\Delta t + \left(p_{Batt,disch_{i,t}}\Delta t - p_{Batt,ch\ i,t}\Delta t\right), \forall t \tag{5}$$

In turn, the power provided by the IDGs and the main grid should be bounded. On one part, the maximum power supplied by the RES is limited by the manufacturer, which defined the $p_{RES_{nom_i}}$. In the case of the main grid, it depends on the infrastructure and contracts with the local operator to establishp_{Grid}. In this way, the boundaries of the power of IDGs and main grid are defined as,

$$0 \le p_{RES_{i,t}} \le p_{RES_{nom_i}}, \forall i, t \tag{6}$$

$$0 \le P_{Grid,t} \le p_{Grid}, \forall t \tag{7}$$

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

$$SoC_{i,t} = \begin{cases} Soc_{i,0} - \dfrac{\left\{p_{Batt,disch_{i,t}}\Delta t\right\}.100\%}{\eta_{dis_i}\cdot C_i^{Batt}} + \dfrac{\eta_{ch_i}\cdot\left\{p_{Batt,ch_{i,t}}\Delta t\right\}\cdot 100\%}{C_i^{Batt}} & \forall i, t = 1 \\ Soc_{i,t-1} - \dfrac{\left\{p_{Batt,disch_{i,t}}\Delta t\right\}.100\%}{\eta_{dis_i}\cdot C_i^{Batt}} + \dfrac{\eta_{ch_i}\cdot\left\{p_{Batt,ch_{i,t}}\Delta t\right\}\cdot 100\%}{C_i^{Batt}} & \forall i, 1 < t \le T \end{cases} \tag{8}$$

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,

$$SoC_{i,t}^{min} \le SoC_{i,t} \le SoC_{i,t}^{max}, \forall i, t \tag{9}$$

$$0 \le p_{Batt,ch\ i,t} \le p_{Batt,chm}.x_{batt,t}, \forall i, t \tag{10}$$

$$0 \le p_{Batt,disch_{i,t}} \le p_{Batt,dischm}\cdot(1 - x_{bat,t}), \forall i, t \tag{11}$$

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.

5. The Machine learning procedure

The self-trained CREMS model with a scalable structure based on the cloud performs load consumption, generation prediction, advanced optimization, power flow analysis, and core knowledge. The ML procedure aims to forecast photovoltaic power generation, wind power generation, and energy load curve through an SL model trained in advance with historical data. Next, the optimization model solves the EDP using the forecasted data to obtain optimal DER usage while equalizing the ESSs' SoC and assigning the output variables as the IDG setpoints in the MGT. The CREMS optimization model reduces the cost charged by the utility grid by considering real-time forecast generation and demand power. Moreover, the control strategy uses a 4-minute temporal resolution data feedback to adjust the EDP power reference levels leading to a CREMS fully deployed. The Supervised Learning (SL) model included in this CREMS involves equivalent steps to the statistical modeling technique for developing, validating, and implementing the solution as stated and executed in [6]:

1. Data collection.
2. Data preparation and missing/outlier treatment.

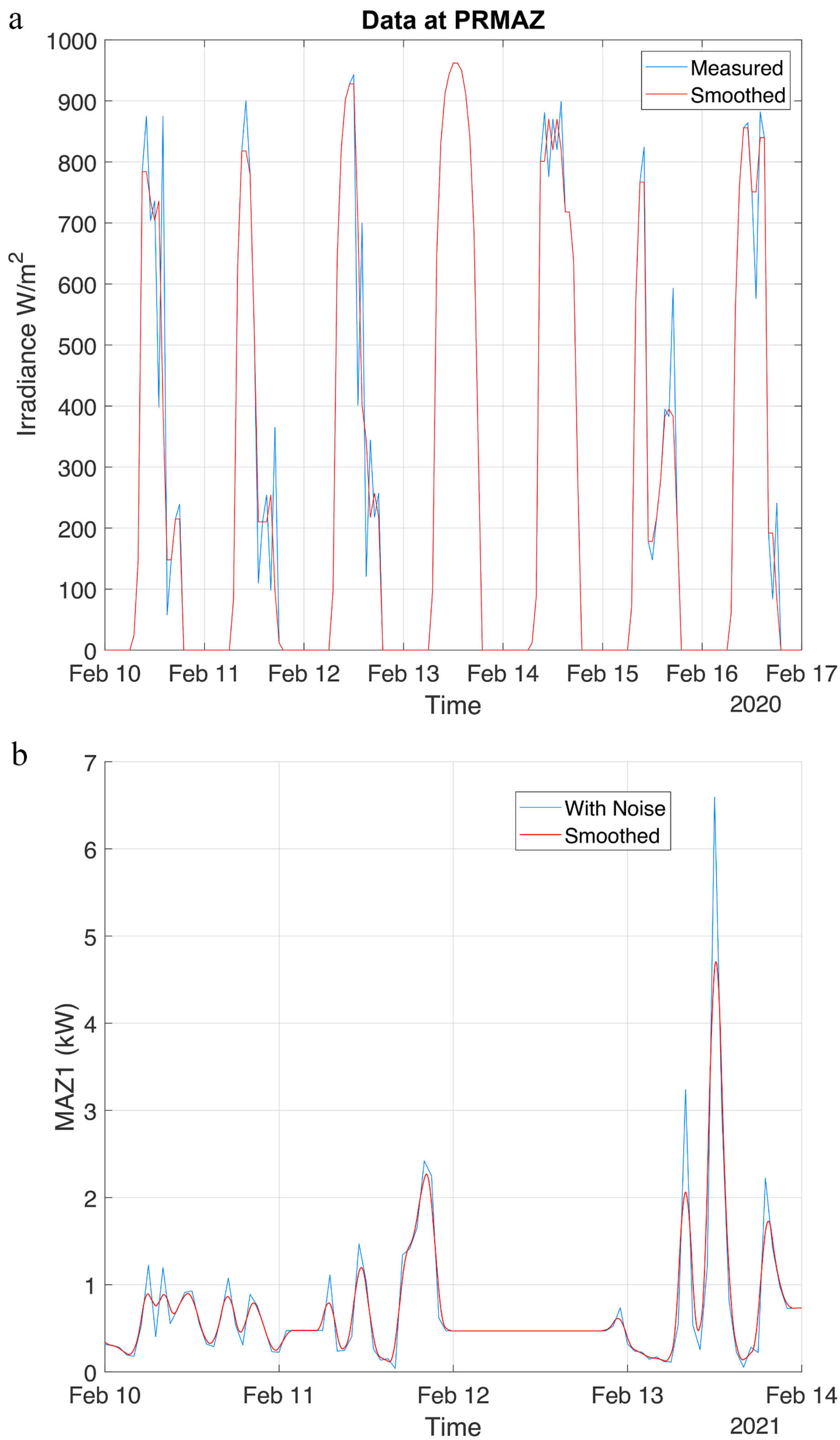


Fig. 6. Preliminary phase results. Source: Authors.

3. Data analysis.
4. Train algorithm on training and validation data.
5. Test the algorithm on test data.
6. Deploy the algorithm.

SL constructs the input–output variables interrelation then ML uses empirical data to estimate the cited variables' statistical dependencies. The purpose of this probabilistic interpretation emerges because it is essential to gather usable patterns from actual logs, especially the non-existing connection amid the CREMS with public irradiance and wind speed data and measured energy utilization. The justification for this follows the relationship complexity. As the input/output variables' relationship complexity is high, the human capacity for extracting a relationship model is low. Therefore, an expert contribution is not feasible. The ML modeling process considers both the preliminary phase (PP) and the learning phase (LP). All the scripts developed in Matlab and stored in the S&O section rely on the data and format available [28]. PP transforms the original data into an acceptable scheme through a cleaning process that includes noise cleaning, deviance subtraction, omitted data treatment, and attribute selection in the weather and consumption data to get the training set [29].

It is a standard technique to alter the original data to moderate the learning procedure and enhance the ML execution. The data pre-processing [30] scheme conceived for the CREMS considers the weather information available in the National Solar Radiation Database (NSRDB) and the energy consumption measured with state-of-the-art IoT meters deployed in a residential area in Mayagüez, Puerto Rico (PRMAZ). The CREMS looks up into information divulged constantly, cleans the data, and merges them into a specific format. It also incorporates one single timestamp on the meteorological and energy consumption material. The gathered information started in March 2020 with a minimum of 24 samples per day, and the CREMS framework enables real-time data availability, centralizing the archives in an S3 bucket, primarily as a cloud object. Fig. 6 exhibits some evidence for the mentioned location.

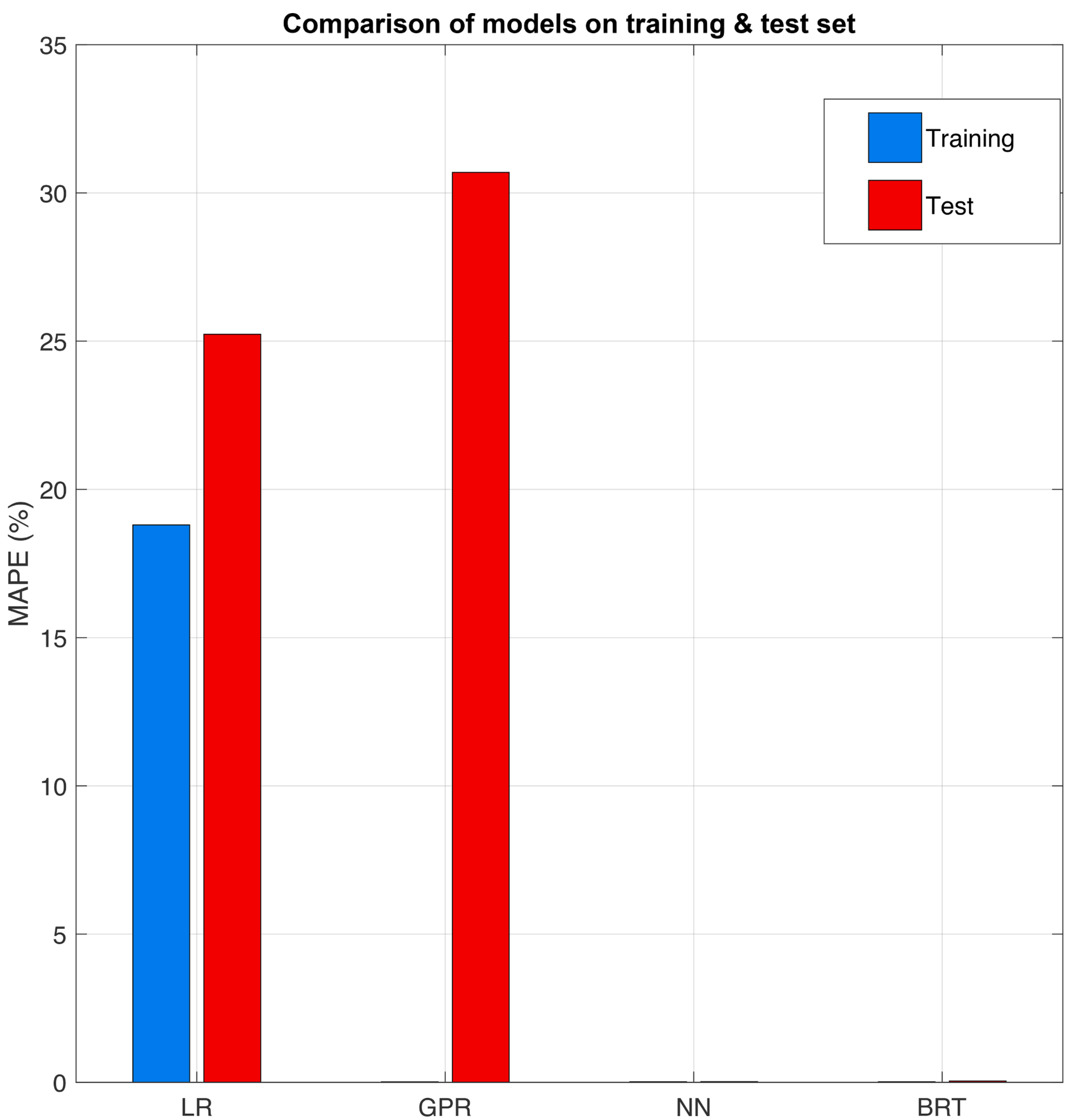


Fig. 7. Learning phase results.

Fig. 6a illustrates the impact on the irradiance information after the PP in PRMAZ, and Fig. 6b displays the PP result for the consumption data in the first residence (MAZ1).

The CREMS uses the LP to convert the optimized data into a prediction model. Like other solutions [31], the LP presented in this research requests a search space and an evaluation criterion to define a parametric model. The parameter estimation scheme used by the CREMS considers regressions, over-fitting, estimators' efficiency, ill-conditioning, the curse of dimensionality, and bias-variance dilemmas [32] but could not grant a modeling approach for one information set. So, the CREMS performs an iterative strategy using independent regression approaches such as linear regression (LR), bagged regression tree (BRT), regression trees (RT), Gaussian process regression (GPR), neural networks (NN), and support vector machines (SVM). Likewise, the ML procedure uses coefficient values optimization techniques to restrain definitive attributes under a square error standpoint, providing function convexity.

The CREMS uses Mean Absolute Percent Error (MAPE) to estimate the model's abstraction capacity and divides the information into three parts: training, testing, and validation, considering 50%, 25%, and 25% as percentage partitions and regarding the model's complexity. The 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. Fig. 7 outlines the LP's results for the irradiance estimator and compares the MAPE for the LR, GPR, NN, and BRT estimators using the testing and training sets. Then, Fig. 8 validates the NN model's performance. This iterative process permits the selection of the NN estimator as a predictor model for irradiance. The CREMS framework uses a parallel process for the wind-speed and energy consumption estimator.

6. The CREMS implementation

This section explains the SEC's MGT adaption to emulate an MGC running on three IDGs individually associated with a BESS and functioning as a dispatchable unit (Fig. 9). As discussed above, the self-trained CREMS performs duties linked to the MGC hyper-converged analytics like customer utilization, resource assignment, energy estimation, and centralized learning within a user-defined time resolution. This experiment operates within a half-hour time resolution and real-time information to accommodate the IDG's setpoints leading to a fully deployed CREMS. The MGT includes a dSPACE SCALEXIO [33] system working as PHIL interface that controls the test bench inverter-based generators. It also incorporates an OPAL-RT OP1400, a four-quadrant power amplifier used as PHIL that can emulate the utility grid. By default, the MGT includes four 2.2 kW three-phase DANFOSS VLT-302s 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.

For this test, the MGT includes IoT energy efficiency features as it includes eGEO 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 Fig. 10a). 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) [20]. 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 (see Fig. 10b) 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. On the

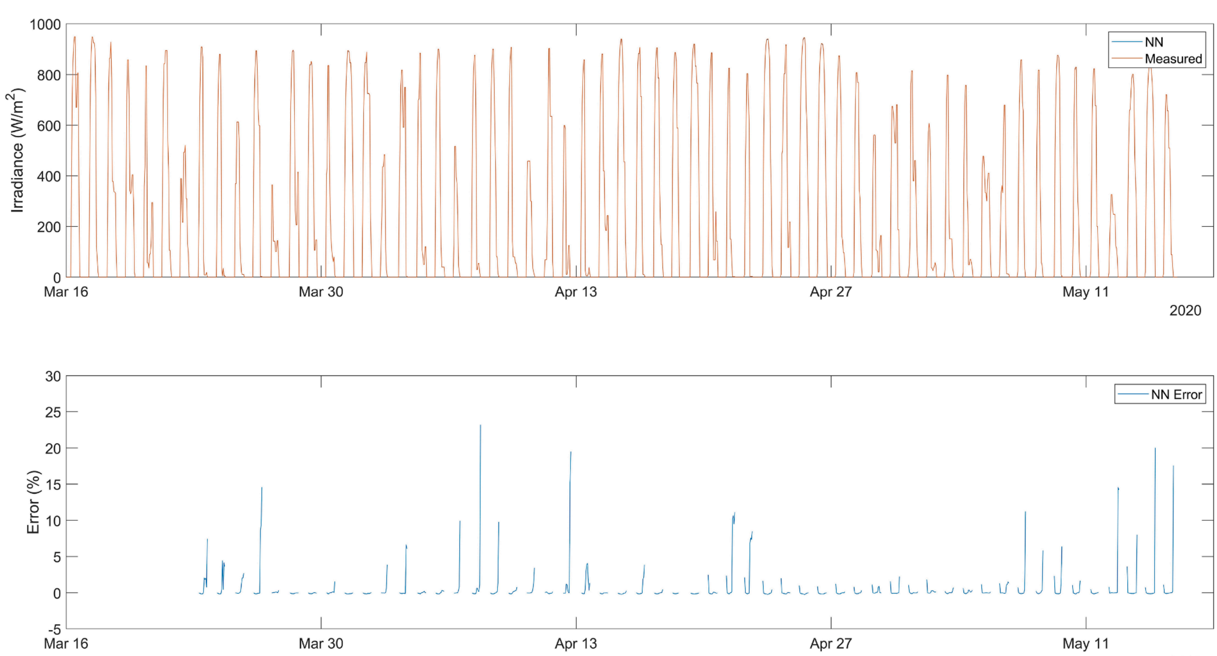


Fig. 8. Model's performance.

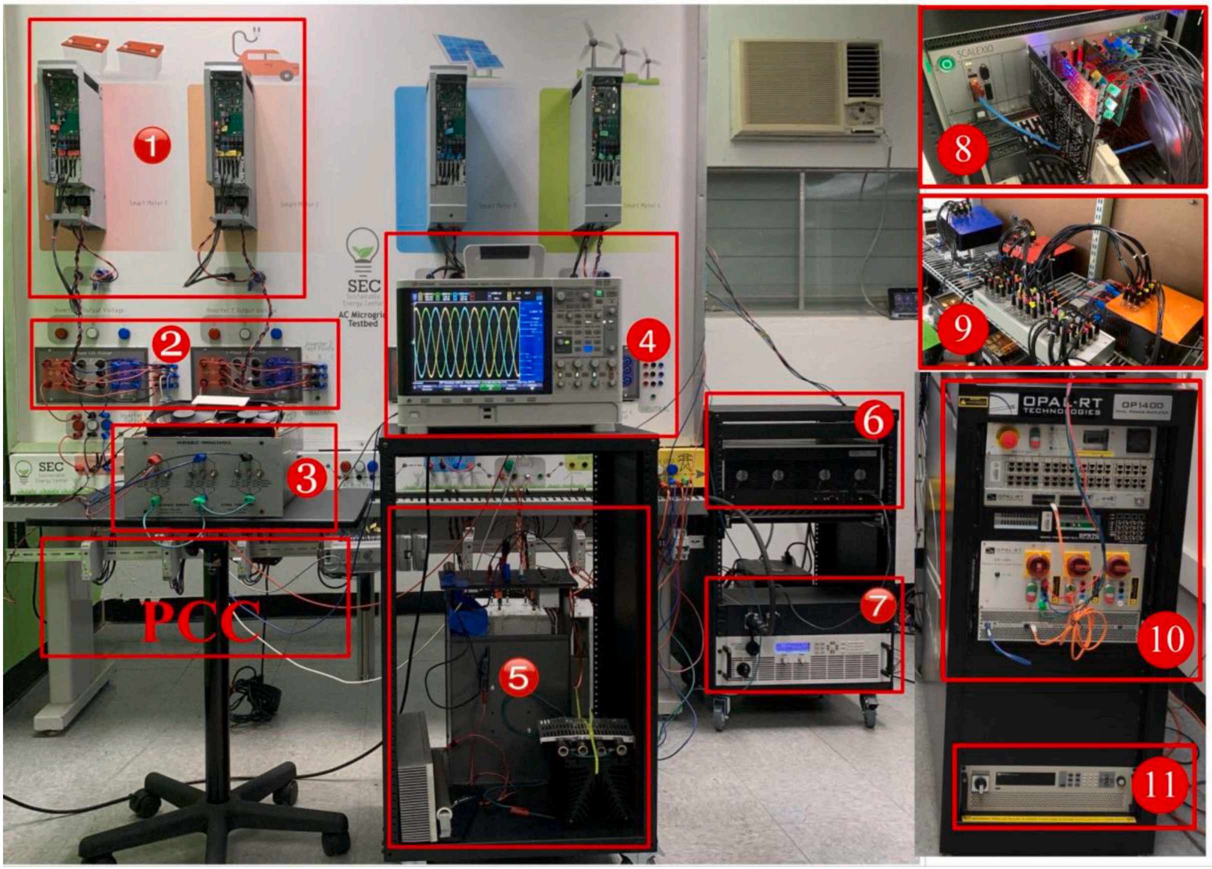
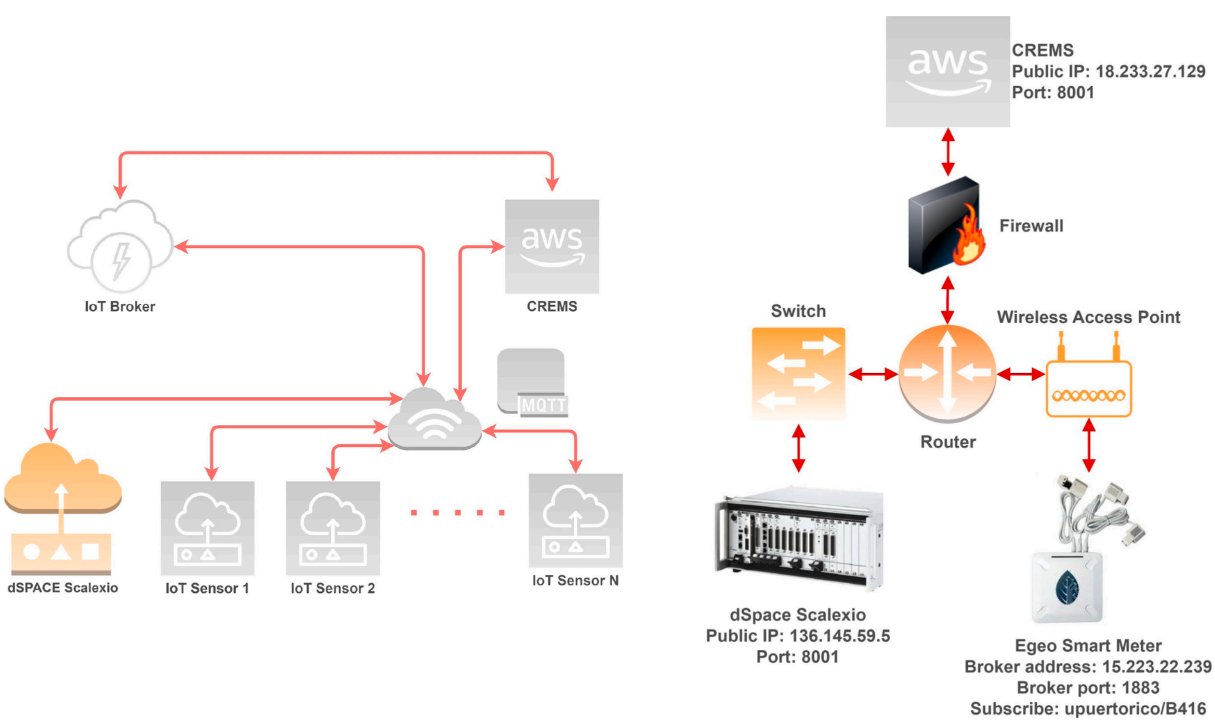


Fig. 9. Microgrid Testbed, 1-IDGs, 2-LCL output filters, 3-Linear Load, 4-Power analyzer, 5-Non-Linear load, 6-dSPACE-Scalexio (back view), 7-DC Supply (IDGs), 8-dSPACE-Scalexio (front view), 9-Sensor Boxes, 10-OP1400 Power Amplifier, 11-DC Supply (OP1400).



a. Communication protocols b. Network Topology

Fig. 10. IoT strategy. Source: Authors.

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 [5].

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 EPDP should reduce the utility grid price by considering the forecast generation and demand power [34]. Other conditions [35] 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.

7. Discussion about the CREMS performance

The IoT devices continuously track the load consumption patterns and update the information to the CREMS to solve the economic dispatch problem 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. Fig. 11 shows one PV generation profile that the IDG units used for testing the operation in a one-week timeframe. The forecast profile has been intentionally overfitted to show an example of poor performance of the ML model, in this case the GPR model.

Similarly, Fig. 12 shows two profiles related to the load: the real-load profile and the forecast-load profile with and without smart meters. Fig. 12 a considers that the CREMS does not include the smart meter and generation dispatched from the CREMS will always try to match the consumption. Fig. 12b exhibits the performance of the CREMS with the smart meter. In this case, the real-load profile comes from the IoT devices previously presented, and the forecast-load results from the ML process performed by the CREMS. The load profiles represent the three-house energy consumption in a one-week timeframe. Also, Fig. 12 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 min for the rolling horizon.

As observed, Fig. 12a shows a case where the system does not include the smart meter in close-loop with the CREMS. For this reason, the system does not provide an optimal solution for the load profile variation. On the other hand, in the scenario considered in Fig. 12b, 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 min. 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 Fig. 12b by the blue line. Fig. 13 develops the discrepancy compensation in utility cost terms. As intended, the experimental results try to minimize the utility grid usage and maximize the renewable resources use. Fig. 13a exhibits time sections where the utility grid power remains as flat as possible and other spots where the utility grid behavior shows variations throughout the day [36]. Fig. 13b displays the time sections where the CREMS adjusts the load profile variations forcing the RES to compensate for the additional utility grid power.

This adjustment process requires one minute under real-time consideration - one hour in experimental results, and it is evident at the beginning and finalization of the load variation where the utility grid provides the microgrid's energy balance with positive or negative power during the transient involving a more expensive solution regarding the utility grid. Nevertheless, the load profile variation inclusion leads to an optimal solution within the considered experiment.

Finally, Fig. 14 shows the results of the SoC equalization process between batteries with and without smart meters. This experiment contemplates intentionally different initialization values for the SoC in all the IDGs (90%, 50%, and 35%, as observed in Fig. 14). As illustrated, in both cases, 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. It is notorious that in both cases, SoC equalization takes place. However, Fig. 14b exhibits a more optimal solution by including the load profile variation and its corresponding effects on a real-life scenario. 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

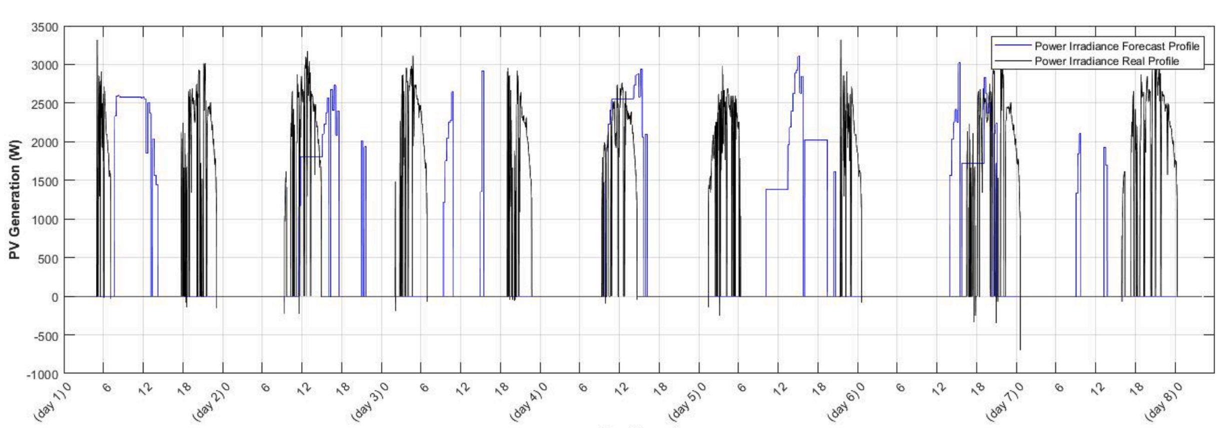
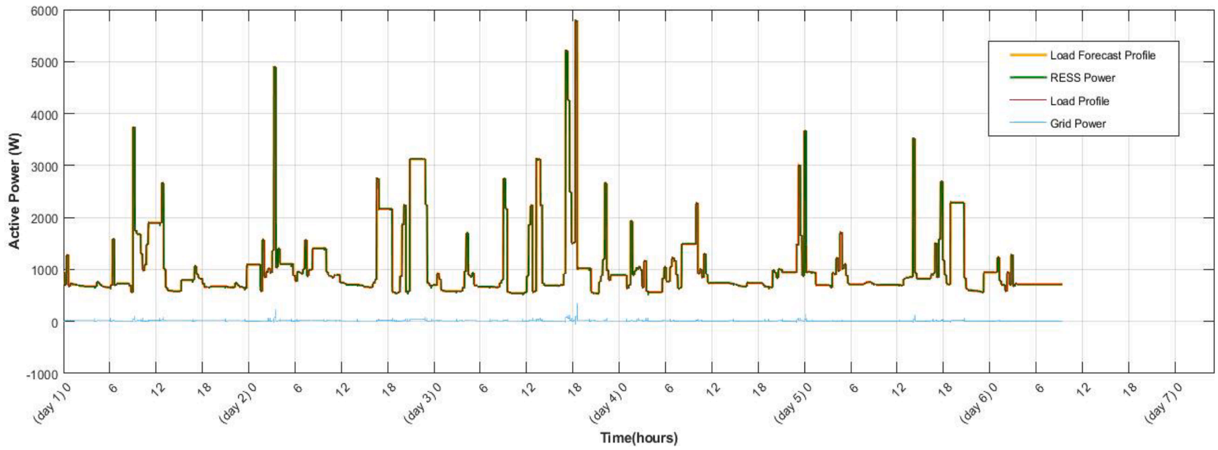
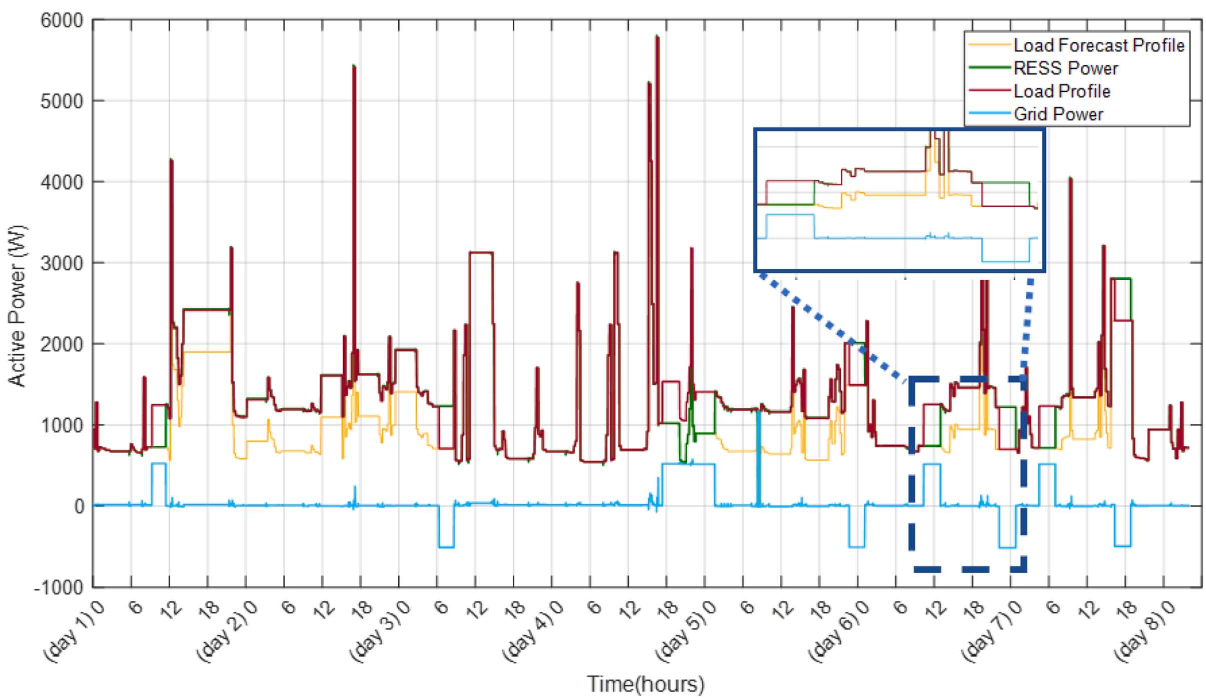


Fig. 11. PV Generation Profile. Source: Authors.



a. Scheduling no smart meter

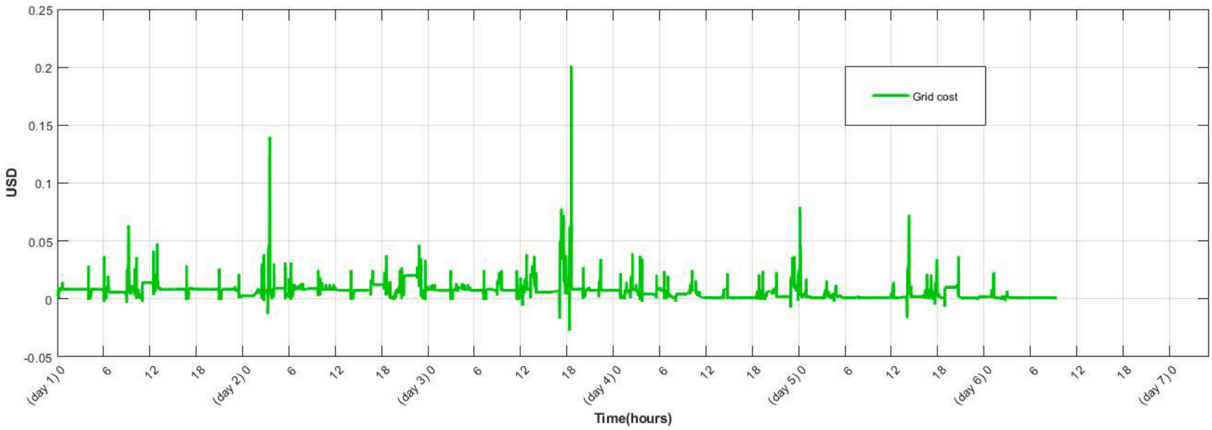


b. Scheduling with smart meters

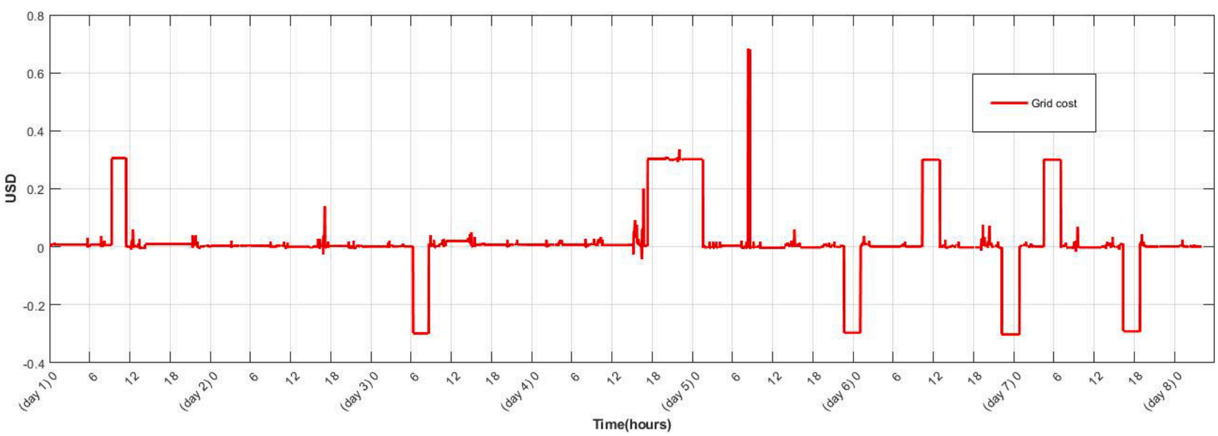
Fig. 12. Experimental generation and consumption schedule results. Source: Authors.

hours and days.

Comparing the results presented in this section with other authors' results is not feasible because, to our knowledge, no similar experiments test the CREMS in a real-life HIL and PHIL testbed considering all the conditions highlighted in Table 1. However, considering specific requirements such as real-time latency performance, CC technology deployment cost, and SoC equalization approach, the following comparison is worthy. In this experiment, the latency caused by the IoT services is close to 34 s. This means a 10% improvement (34 versus 38 s for each RESS), according to the details given in [37]. Based on the AWS reserved instances pricing [38] and the CREMS configuration, 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 value presented in [37]. The SoC equalization technique among distributed ESS with different capacities looks for proportional power sharing and equal discharge/charge profiles [39]. Most of the reviewed literature tries to maintain the SOC of the battery within an appropriate range (for instance, 40% to 80%) during the consecutive energy dispatch process, as indicated in [19]. The results exhibited in [40] and [41], investigations not involving ML or IoT within their framework, reach the SOC equalization by the fourth time-spot considered in their validation timing. It means a one-time spot later than the results presented in Fig. 13. This indicates that the CREMS gets a 25% improvement considering the SoC equalization reachability time for both cases, with and without IoT performance.



a. Grid cost no smart meter



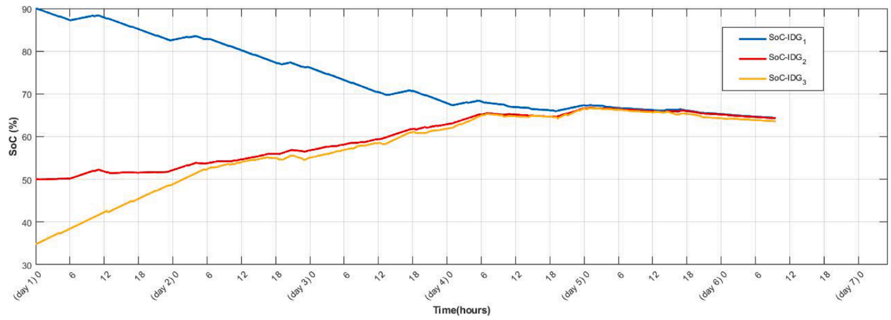
b. Grid cost with smart meter

Fig. 13. Grid cost comparison. Source: Authors.

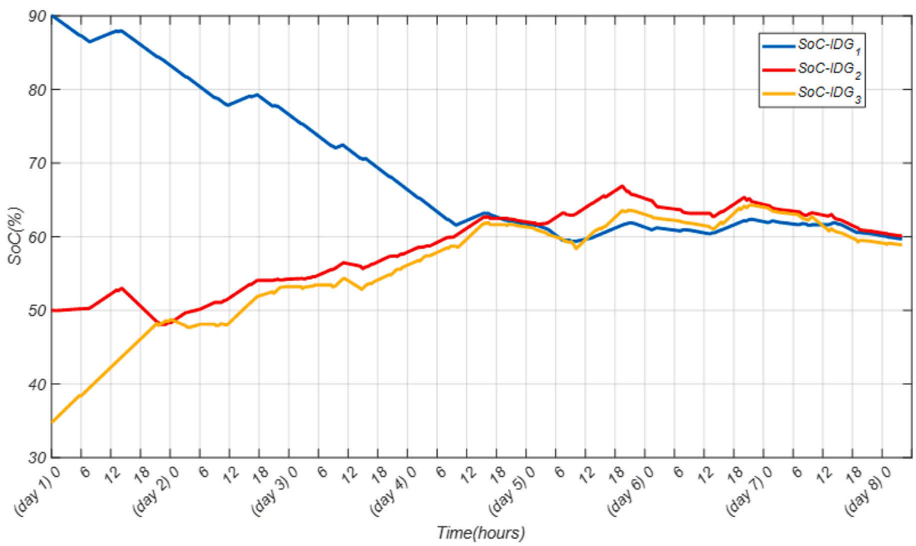
8. Conclusion

The main contribution of this paper is to test real-life CREMS. Unlike previous approaches in this experiment, the system optimization algorithm executes in an MPS instance using CC, ML, and IoT. This approach improves CREMS performance by eliminating offline optimization. The CC usage improves the findings presented in [10] as the CREMS relieves the utility costs and optimizes the battery usage. As the system can manipulate energy decisions near to real-time, the EMS structure adapts conveniently to the amount of MGs available at a specific time. Also, the energy production prediction provides 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 in this experiment by the power amplifier, balances the energy demand with the generation in specific simulation periods. Fig. 12 displays the load-generation balance impact over the cost assumed by the MGC final users in particular time-spots linked to the utility grid usage. As expected, the utility grid and RES usage consider operating fees. The market operator defines the utility cost, and maintenance, replacement, and lifespan determine the long-term RES expense. The ML model specifies the dispatched power according to the forecasted power availability within the specific timeframe. Likewise, 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.

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 a supervised learning model definition beyond offline computing limitations. Conveniently, this close to real-time approach improves the standardized strategic level days-to-hours time frame and decreases that to minutes-to-seconds. In fact, and according to these experimental results, the CREMS architecture could expand to the MG tactical level - power management. The hardware-in-the-loop and power-hardware-in-the-loop testbed inclusion allows us to examine some framework features like autonomy and scalability. Additionally, it permitted defining other attributes like cost-effectiveness compared to [19] - close to 50% savings in the overall CC deployment, and 70% in the utility grid usage (USD 6.55 per day) [36] -, solving the dispatch under a 4-minute rolling-horizon approach, and settling the Ethernet bandwidth requirements – 2.56 Mbit/s. Neither [3,10,19] nor [28] defined it. For future work, two research directions are worth the analysis effort. The proposed CREMS needs to run in customer-owned cloud-premises and cross-platform commercial scenarios. In addition, it would be interesting to improve the optimization problem and include state-of-the-art technologies for the internet of things.



a. SoC with no smart meters



b. SoC with smart meters

Fig. 14. SoC equalization. Source: Authors.

CRediT authorship contribution statement

D.G. Rosero: Supervision, Investigation, Validation, Writing – original draft, Formal analysis. **E. Sanabria:** Investigation, Validation. **N.L. Díaz:** Supervision, Writing – review & editing. **C.L. Trujillo:** Supervision, Writing – review & editing. **A. Luna:** Investigation, Validation, Writing – review & editing. **F. Andrade:** Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This paper represents part of the results obtained under the support of the Doctorate in Engineering from the Universidad Distrital Francisco José de Caldas.

Funding

This endeavor was financed by Minciencias with the project "Programa de Investigación en Tecnologías Emergentes para Microredes Eléctricas Inteligentes con Alta Penetración de Energías Renovables" [grant number 80740-542-2020] and Dynamic Defense Solutions SAS.

Data availability

Data will be made available on request.

References

[1] Manz D, Walling R, Miller N, LaRose B, D'Aquila R, Daryanian B. The grid of the future: ten trends that will shape the grid over the next decade. IEEE Comput Appl Power 2014;12(3):26–36.
[2] Parag Y, Sovacool BK. Electricity market design for the prosumer era. Nat Energy 2016;1(4). doi: 10.1038/nenergy.2016.32.
[3] Macana CA, Pota HR, Zhu Q, Guerrero JM, Vasquez JC. Experiments on a real-time energy management system for islanded prosumer microgrids. Electronics 2019;8(9):925. https://doi.org/10.3390/electronics8090925.
[4] Luna AC, Meng L, Diaz NL, Graells M, Vasquez JC, Guerrero JM. Online energy management systems for microgrids: experimental validation and assessment framework. IEEE Trans Power Electron 2018;33(3):2201–15. https://doi.org/10.1109/tpel.2017.2700083.
[5] Sanabria-Torres EA, Rosero-Bernal D, Diaz NL, Trujillo CL, Andrade-Rengifo F. Microgrid Structure for testing a Real-Time Energy Management System Model. In: 2022 IEEE Applied Power Electronics Conference and Exposition (APEC); 2022. https://doi.org/10.1109/apec43599.2022.9773556.

[6] Paluszek M, Thomas S. Machine learning examples in MATLAB. MATLAB Machine Learning. pp. 85–88; 2017. doi: 10.1007/978-1-4842-2250-8_6.
[7] eGEO. eGEO. https://www.egeo.co/ (accessed Sep. 06, 2022).
[8] Kaur A, Kaushal J, Basak P. A review on microgrid central controller. Renew Sustain Energy Rev 2016;55:338–45. https://doi.org/10.1016/j.rser.2015.10.141.
[9] Macana CA, Mohiuddin SM, Pota HR, Mahmud MA. Online energy management strategy for islanded microgrids with feedback linearizing inner controllers. 2017 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia) 2017. https://doi.org/10.1109/isgt-asia.2017.8378319.
[10] Rosero-Bernal D, Sanabria-Torres EA, Andrade-Rengifo F, Diaz NL, Trujillo CL. Machine Learning Experiments for a Real-Time Energy Management in a Microgrid Cluster. In: in IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society; 2021. p. 1–6.
[11] Aws Cloud Computing. Servicios de informática en la nube. Amazon Web Services, Inc; 2019. https://aws.amazon.com/es/ (accessed Apr. 06, 2019).
[12] Rosero DG, Díaz NL, Trujillo CL. Cloud and machine learning experiments applied to the energy management in a microgrid cluster. Appl Energy 2021;304:117770. https://doi.org/10.1016/j.apenergy.2021.117770.
[13] MATLAB Production Server. https://la.mathworks.com/products/matlab-production-server.html (accessed Aug. 24, 2022).
[14] AWS. Amazon Web Services, Inc. https://aws.amazon.com/es/s3/ (accessed Aug. 24, 2022).
[15] Secure and resizable cloud compute – Amazon EC2 – Amazon Web Services. Amazon Web Services, Inc. https://aws.amazon.com/ec2/ (accessed Sep. 09, 2022).
[16] Wu Y, Wu Y, Guerrero J, Vasquez J, Palacios-García E, Guan Y. IoT-enabled microgrid for intelligent energy-aware buildings: a novel hierarchical self-consumption scheme with renewables. Electronics 2020;9(4):550. https://doi.org/10.3390/electronics9040550.
[17] Tajalli SZ, Tajalli SAM, Kavousi-Fard A, Niknam T, Dabbaghjamanesh M, Mehraeen S. A Secure Distributed Cloud-Fog Based Framework for Economic Operation of Microgrids. In: 2019 IEEE Texas Power and Energy Conference (TPEC); 2019. https://doi.org/10.1109/tpec.2019.8662201.
[18] Albataineh H, Nijim M, Bollampall D. The design of a novel smart home control system using smart grid based on edge and cloud computing. In: 2020 IEEE 8th International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada; 2020. doi: 10.1109/sege49949.2020.9181961.
[19] Dong W, Yang Q, Li W, Zomaya AY. Machine-learning-based real-time economic dispatch in islanding microgrids in a cloud-edge computing environment. IEEE Internet Things J 2021;8(17):13703–11.
[20] Lei L, Tan Y, Dahlenburg G, Xiang W, Zheng K. Dynamic energy dispatch based on deep reinforcement learning in IoT-driven smart isolated microgrids. IEEE Internet Things J 2021;8(10):7938–53.
[21] Ali W, et al. Hierarchical control of microgrid using IoT and machine learning based islanding detection. IEEE Access 2021;9:103019–31.
[22] Moghimi M, Jamborsalamati P, Hossain J, Stegen S, Lu J. A Hybrid Communication Platform for Multi-Microgrid Energy Management System Optimization. In: 2018 IEEE 27th International Symposium on Industrial Electronics (ISIE); 2018. https://doi.org/10.1109/isie.2018.8433861.
[23] Real-time Energy Management for PV–battery–wind based microgrid using on-line sequential Kernel Based Robust Random Vector Functional Link Network. Appl Soft Comput 2021;101:107059.
[24] Real time Energy Management System of a photovoltaic based e-vehicle charging station using Explicit Model Predictive Control accounting for uncertainties. Sustain Energy Grids Networks 2022;31:100769.
[25] Patarroyo-Montenegro JF, Salazar-Duque JE, Alzate-Drada SI, Vasquez-Plaza JD, Andrade F. An AC microgrid testbed for power electronics courses in the university of Puerto Rico at mayagüez. In: 2018 IEEE ANDESCON, Santiago de Cali, Aug. 2018. doi: 10.1109/andescon.2018.8564636.
[26] "OP1400," *OPAL-RT*, Aug. 30, 2019. https://www.opal-rt.com/op1400-4-quadrant-power-amplifier/ (accessed Aug. 24, 2022).
[27] Diaz NL, Vasquez JC, Guerrero JM. A Communication-less distributed control architecture for islanded microgrids with renewable generation and storage. IEEE Trans Power Electron 2018;33(3):1922–39. https://doi.org/10.1109/tpel.2017.2698023.
[28] "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 Aug. 29, 2022).
[29] Little RJA, Rubin DB. Statistical Analysis with Missing Data; 2002. doi: 10.1002/9781119013563.
[30] Website. https://www.researchgate.net/publication/242692234_Statistical_foundations_of_machine_learning_the_handbook.
[31] Läuter H, Hand DJ. Discrimination and classification. John Wiley & Sons, Chichester — Brigbane — New York — Toronto 1981. XII, 218 pp., $ 15.50. Biom J 1985;27(2):148–148.
[32] Dangeti P. Statistics for Machine Learning; 2017.
[33] "SCALEXIO." https://www.dspace.com/en/pub/home/products/hw/simulator_hardware/scalexio.cfm (accessed Aug. 31, 2022).
[34] Imtiaz B, Cui Y, Zafar I. Economic Dispatch of Microgrid Incorporating Demand Response Using Dragonfly Algorithm. In: 2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA); 2021. https://doi.org/10.1109/aeeca52519.2021.9574430.
[35] Mojica-Nava E, Macana CA, Quijano N. Dynamic population games for optimal dispatch on hierarchical microgrid control. IEEE Trans Syst Man Cybernet: Syst 2014;44(3):306–17. https://doi.org/10.1109/tsmcc.2013.2266117.
[36] Torres S, Alberto E. Cloud-based energy management system for grid-connect prosumer microgrid; 2022. Accessed: Dec. 17, 2022. [Online]. Available: https://hdl.handle.net/20.500.11801/2921.
[37] Wang S, Wang X, Wu W. Cloud computing and local chip-based dynamic economic dispatch for microgrids. IEEE Trans Smart Grid 2020;11(5):3774–84.
[38] AWS Pricing Calculator. https://calculator.aws/#/ (accessed Oct. 11, 2022).
[39] Diaz NL, Luna AC, Vasquez JC, Guerrero JM. Centralized control architecture for coordination of distributed renewable generation and energy storage in islanded AC microgrids. IEEE Trans Power Electron 2017;32(7):5202–13. https://doi.org/10.1109/tpel.2016.2606653.
[40] Fagundes TA, Fuzato GHF, Ferreira PGB, Biczkowski M, Machado RQ. "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 2022;3(1):90–100.
[41] Zhang L, Li T. SOC Dynamic Balance Strategy Based on Adaptive Droop Control with Variable Regulating Factor. In: 2021 3rd Asia Energy and Electrical Engineering Symposium (AEEES); 2021. https://doi.org/10.1109/aeees51875.2021.9403165.