







A distributed IoT-based simulation framework for solar energy management and forecasting

Luis Almendras^{1,2, }, Javier Campoy^{3,4, }, Ignacio-Iker Prado-Rujas^{3,5, },
José L. Risco-Martín^{1,2, }, María S. Pérez^{3,5, }, Katalin Olcoz^{1,2, }

Abstract—Global electricity demand must account for the increasing use of renewable energy sources and their associated variability. Currently, solar is the source of renewable energy that is growing at the highest pace. However, solar power generation is intermittent and variable, which promotes the necessity of accurate forecasting models. There exists a plethora of accurate predictive models anticipating the amount of energy that can be produced in a given area, taking inputs from a solar irradiance sensor farm. However, the management of these plants and models is still rudimentary. New tools and technologies are needed to manage the deployment and control of these facilities. In this work, we consider a complex scenario composed of several solar irradiance sensor farms. Each farm is connected to a centralized management system. This can be straightforwardly conceptualized and deployed using a scalable complex model, through well structured Modeling and Simulation (M&S) methodologies. This paper introduces CAIDE, a novel framework for an integral, real-time monitoring, management, and forecasting of solar irradiance sensor farms. Its purpose is to support the deployment and analysis of solar plants following Model Based Systems Engineering (MBSE) and an Internet of Things (IoT) infrastructure for dynamic environments.

Keywords—Complex Systems, Discrete Event System Specification, Deep Learning, Solar Irradiance, Parallel and Distributed Simulation.

I. INTRODUCTION AND RELATED WORK

CURRENT demand on energy must account for the increasing use of renewable energy sources and their associated variability, and the pressure of reaching net zero emissions by 2050 [1]. Among all renewable energies, solar is the source of energy that is growing at the highest rate in the European Union (EU) [2]. As all the renewable energies, Photovoltaic (PV) power is intermittent and variable. This variability leads to the necessity of more accurate forecasting models, which has become increasingly important during the last decade. Accurate PV power modeling can help to determine the best power quality generation periods, reducing price volatility [3]. It can also play an essential role in ensuring that a smart grid deployment project is economically and ecologically feasible [4].

Most modern PV models are based on accurate predictions of what we call *base models*. They anticipate the amount of energy produced either

starting with the corresponding initial state (classic models) or with a training/inference data window (heuristic models). Current methods explored in the field can be grouped as Numerical Weather Prediction (NWP), image-based, statistical, and Machine Learning (ML). They can also be classified based on their characteristics, meaning whether a method takes into account spatio-temporal features or not, is deterministic or probabilistic, considers exogenous inputs (other inputs such as physical variables) or just its data features, etc. [5]. For instance, Arbizu-Barrena et al. [6] focus on the cloud index to forecast solar irradiance with the aid of an NWP. Ayet and Tandeo [7] aim to forecast solar irradiance based on geostationary satellite images. ML approaches are recently gaining attention: Alzahrani et al. [8], for example, employ Long Short-Term Memory (LSTM) networks for high resolution forecasting (100Hz) in a single location, obtaining high levels of precision. Prado-Rujas et al. [9] present several Artificial Neural Networks (ANNs) for solar irradiance forecasting, where the scenario is not bound to the specific number of available sensors or their distribution, and constitute a more resilient approach since algorithms can recover from sensor failures.

Our vision is, however, oriented to a systems of systems architecture, a more holistic and *integrative model*. In this work, we consider a more complex scenario composed of several solar irradiance sensor farms, deployed to analyze the PV solar production possibilities, for instance. Each farm is in turn connected to a centralized management system. Every sensor is supposed to be real or a virtual replica. Consequently, the data generated by the sensors can also be synthetic or authentic. This complex system can be conceptualized first by using a scalable complex model, where through a well structured Modeling and Simulation (M&S) methodology, all the aforementioned *base models* can be easily integrated. Figure 1 shows our conception of the integrative model. Following an Internet of Things (IoT)-based architecture, solar irradiance is monitored at the edge layer by a set of sensors that continuously send data to the server at the fog layer. There, domain experts can analyze data, run some tests, or schedule the execution of predictive models. There also exists a cloud layer, where authorities can compare different reports and make high-level decisions. The power of the cloud layer can also be used to train predictive base models, ensuring the scalability and durability of the system.

¹Architecture and Technology of Computing Systems, Universidad Complutense de Madrid

²E-mail: {luisalme, jlrisko, katalin}@ucm.es

³Ontology Engineering Group, ETSI Informáticos, Universidad Politécnica de Madrid, 28660 Madrid, Spain

⁴E-mail: javier.campoy@alumnos.upm.es

⁵E-mail: {iker.prado.rujas, maria.s.perez}@upm.es

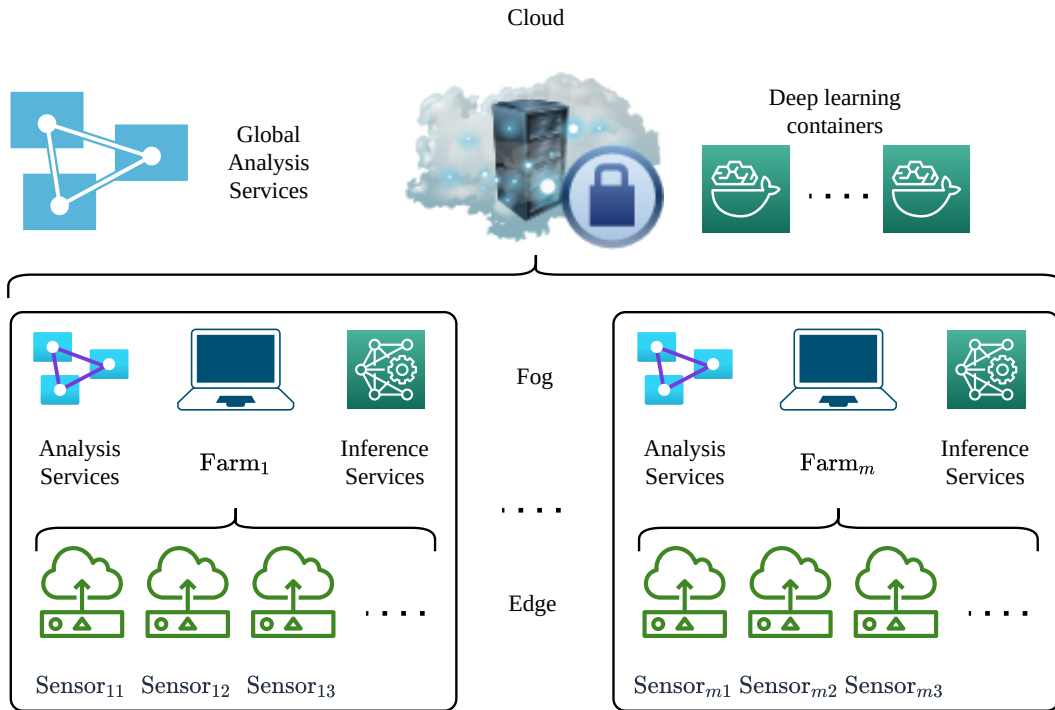


Fig. 1: Big picture of the proposed integrative model.

To simulate and operate this complex model, in this paper we propose CAIDE, a novel framework that enables real-time monitoring and prediction of solar energy for PV sensor plants. The purpose of our approach, based on the principles of Model Based Systems Engineering (MBSE), is to ensure the reliability and scalability of the design and deployment of infrastructures. At the same time, our aim is to provide high-performance real-time services, such as the detection of outliers or the execution of complex forecasting methods. All this is achieved through the implementation of model-driven technologies and an infrastructure based on IoT. As a result, we address three main topics in the sustainable management of solar irradiance monitoring and prediction under the umbrella of model-driven technologies: (i) provide a robust interface to simultaneously manage different solar irradiance sensor farms, (ii) provide vertical scalability, modeling the whole structural pyramid from the sensors to the authorities, and (iii) provide horizontal scalability, being able of adding more sensors and farms with the support of parallel and distributed simulation.

At this stage of research, the framework is built upon a conceptual layer, using formal models and synthetic data to prove the feasibility of the entire architecture. To this end, we define the virtual structure and modular behavior of CAIDE, which is capable of carrying out initial synthetic experiments through Discrete Event System Specification (DEVS) [10], a well-known M&S formalism. In the future, (i) real data will feed our framework, and (ii) virtual elements described in this work will be replaced by physical ones (e.g., sensors or servers) following an incremental MBSE procedure.

This paper is organized as follows. Section II

presents the foundational technologies used to design our integrative model. Section III presents the architecture of our framework, based on a well-known M&S formalism and able to perform parallel and distributed simulations. Section IV illustrates the simulations performed to test our hypotheses and shows the results obtained in a synthetic scenario, fed with real monitoring data. Finally, Section V draws some conclusions and introduces future research lines.

II. BACKGROUND

Our integrative framework must be able to run scalable simulation scenarios based on the template provided in Figure 1. To build the integrative model, we have selected a M&S formalism named Discrete Event System Specification (DEVS). To address this complex predictive scenario, we have included a base model from the family of Deep Learning (DL) models published in [9]. In the following, we introduce these two technologies. Other services included in CAIDE, such as the detection of outliers and data augmentation, are based on well-known foundations and do not require a deep explanation; appropriate references will be included in Section IV.

A. The DEVS formalism

The simulation framework presented in this article is based on the ground foundations of parallel DEVS.

Parallel DEVS is a modular and hierarchical formalism for modeling discrete event systems based on set theory [10]. It includes two types of models, atomic and coupled, that have an interface consisting of input (X) and output (Y) ports to communicate with other models. Additionally, in atomic models, every model state (S) is associated with the time advance function ta , which determines the duration in

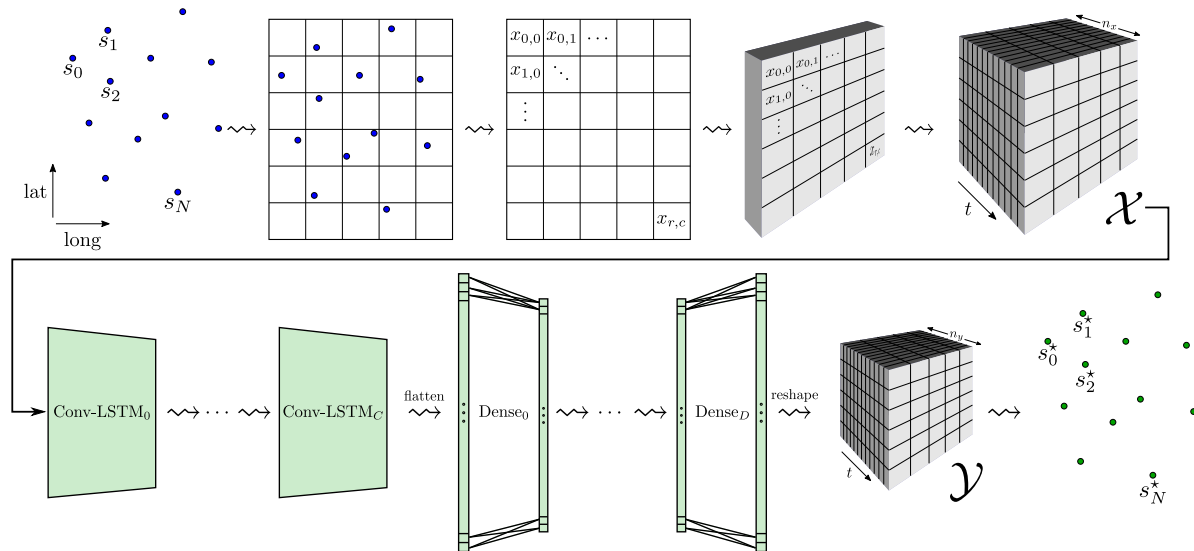


Fig. 2: Diagram that shows the DL-based model for solar irradiance forecasting from end to end.

which the state remains unchanged.

Once the time assigned to the state has passed, an internal transition function ($\delta_{\text{int}} : S \rightarrow S$) is fired and an internal transition is triggered, producing a local state change ($\delta_{\text{int}}(s) = s'$). At that time, the results of the model execution are spread through the output ports of the model by activating an output function (λ).

Furthermore, external input events (received from other models) are collected in the input ports. An external transition function ($\delta_{\text{ext}} : S \times e \times X \rightarrow S$) specifies how to react to those inputs, using the current state (s), the elapsed time since the last event (e) and the input value (x) ($\delta_{\text{ext}}((s, e), x) = s'$). Parallel DEVS introduces a confluent function ($\delta_{\text{con}}((s, ta(s)), x) = s'$), which decides the next state in cases of collision between external and internal events.

Coupled models are the aggregation/composition of two or more models (atomic and/or coupled), connected by explicit couplings. This makes DEVS closed under coupling and allows us to use networks of systems as components in larger coupled models, leading to hierarchical and modular constructions.

Overall, DEVS provides a framework for information modeling that has several advantages in the analysis and design of complex systems: completeness, verifiability, extensibility, and maintainability.

Once a system is described according to DEVS theory, it can be easily implemented using one of the many DEVS M&S engines that have come into existence in the last decades.

Among them, xDEVS [11] offers a good alternative to parallelize or distribute simulations in the Cloud, following a microservices architecture and containerization. As a result, any DEVS model can be parallelized or distributed by assigning resources (threads or processes) to different transition and output functions, as parallel or distributed functional programming.

B. Deep Learning-Based Method for Solar Irradiance Forecasting

As discussed in Section I, anticipating the amount of solar energy that can be produced in a given area is fundamental. In the described scenario, we are presented with several solar farms, each with a variable distribution and number of sensors. Therefore, beyond precision, characteristics such as flexibility are required. This set of characteristics is commonly known as *non-functional* [12].

In this article, we adapt the DL-based model from our previous work [9] and incorporate it into the CAIDE framework. Our model receives as input several temporal series produced by real-world sensors, which are arranged in an arbitrary spatial configuration. The data in each instant can then be transformed into a two-dimensional tensor by means of a nearest-neighbor interpolation. This process gives us a stream of the so-called mesh-grids over time. The mesh-grids can be stacked into a three-dimensional object that encapsulates the temporal and spatial (longitude and latitude) properties. This three-dimensional tensor can then be fed into a Deep Neural Network (DNN), which consists of Convolutional LSTM and fully connected layers. The output also has a three-dimensional structure, covering the whole region where the farm is located and several forecast horizons h (1, 11, 31 and 61 minutes in this study). The interpolation process can then be reversed to obtain the predicted values for the specified horizons and for each sensor location. The entire forecasting process is depicted in Figure 2.

Once trained, the model can be deployed on the corresponding fog server, allowing us to run asynchronous predictions at any moment. The neural network can also be trained in the cloud layer with new data accumulated by the system, achieving better accuracy when needed.

Thanks to the forecasting process described previously, our model gains non-functional characteristics that are beneficial to our framework:

- **Flexibility:** Adaptability to a variable number of sensors on each farm or over time.
- **Robustness:** Ability to work at suitable levels of performance even if some sensors fail, as demonstrated in [9].
- **Portability:** Ability to retrain and deploy the model on other farms with different sensors arrangements without modifying the model architecture.

These characteristics are valuable for the described scenario since they allow us to develop an independent model for each farm, with variable spatial configurations and number of sensors.

III. SYSTEM ARCHITECTURE AND DESIGN

CAIDE’s model divides the proposed framework for sensor farms management into the three classical IoT layers: edge, fog, and cloud. While Figure 1 has already illustrated the general picture of the architecture of the framework, Figure 3 depicts its DEVS structure, which is described below. For the sake of space, we describe the high-level architecture of CAIDE instead of formally describing the DEVS structure and behavior of all its atomic and coupled models. The reader is referred to Section II to better follow this section through the DEVS formalism.

The root coupled model is depicted in Figure 3. It does not contain input or output ports. The components included in this coupled model are: a simulation file, F farms, and the cloud.

The *simulation file* atomic model is just a source that reads from a text file all the events that are being injected into the simulation process through the *out* output port. Each entry in this file is composed of: a time mark indicating the virtual instant in which this event will be triggered, the command type associated to the event, and the arguments that each command could need. As a result, this file replicates the set of external events that could happen in a real world scenario. As the excerpt of Figure 3 illustrates, it always begins and ends with the triggering of the initialization and finalization of the simulation experiment (see **START** and **STOP** commands). In the middle, some services can be triggered, like outliers detection or solar irradiance forecasting. This is a pure virtual element, which does not have an equivalent in the real world.

The *farm* coupled model represents a set of solar irradiance sensors located in a geographical area of the Earth and with a control station to monitor and control the facility. This coupled model has an input port, which receives the events sent by the *simulation file* atomic model, and two output ports that send raw data collected by the sensors to the cloud, as well as augmented or fixed sensor data through outliers detection or data analysis services. The *farm* coupled model contains several atomic models:

- A set of S solar irradiance sensor atomic models. They are all located in the edge layer of our IoT architecture. Currently, our sensors are

purely virtual. They read real Global Horizontal Irradiance (GHI) data from a database and send them through the output port, following the timing registered in the database. In the future, we plan to include the possibility of adding synthetic sensors with the objective of analyzing scalability, and real sensors with the aim of testing the possibilities of CAIDE regarding simulations with Hardware In the Loop (HIL).

- A set of atomic models in charge of executing services. We have currently deployed three services: one to detect and fix outliers, another one to perform an inference of the selected predictive algorithm, and the last one to perform data analysis and report. They all follow the same atomic model template. At this moment, the services are executed as part of the fog layer, i.e. they are local processes (to the control station or fog server) that can be parallelized but not distributed. However, this can be modified at any moment, just by adding more computing resources at the fog layer or by externalizing some services to the cloud.
- An atomic model that represents the server located at the control station, named *fog server*. Firstly, it receives simulation commands from the *simulation file* atomic model, which tell the server when to start reading data, execute an outliers detection service, an inference, etc. When the simulation starts, sensor data are received through the e_i input ports. When a set of data is received, it is stored in the local database and sent to the cloud atomic model through the f output port. On the other hand, when a service request is received from the *simulation file*, it is propagated to the corresponding atomic model. Fixed or predicted data are also stored in the local database and sent to the cloud atomic model through the \hat{f} output port.

Finally, the *cloud* atomic model is located in the cloud layer. It receives all the data coming from the different farms (both raw and estimated, i.e., fixed or predicted) and stores them into the cloud database. As in the *fog server*, the cloud atomic model can run heavier services, such as performing big data analyses including data stored on all farms or performing a training service to update current inference models. In any case, these actions are always commanded by the atomic model *simulation file*. In this particular case, we have not included dedicated atomic models that run services because they are always processes installed in docker containers, i.e., they have a distributed architecture in nature and do not need to be encapsulated as DEVS models.

CAIDE has been implemented using xDEVS [11], a cross-platform DEVS simulator. As in xDEVS, CAIDE can use virtual or real time simulation. It can run sequential, parallel and distributed simulations or a combination of all of them, without modifying a single line of code in the underlying simulation model presented in Figure 3.

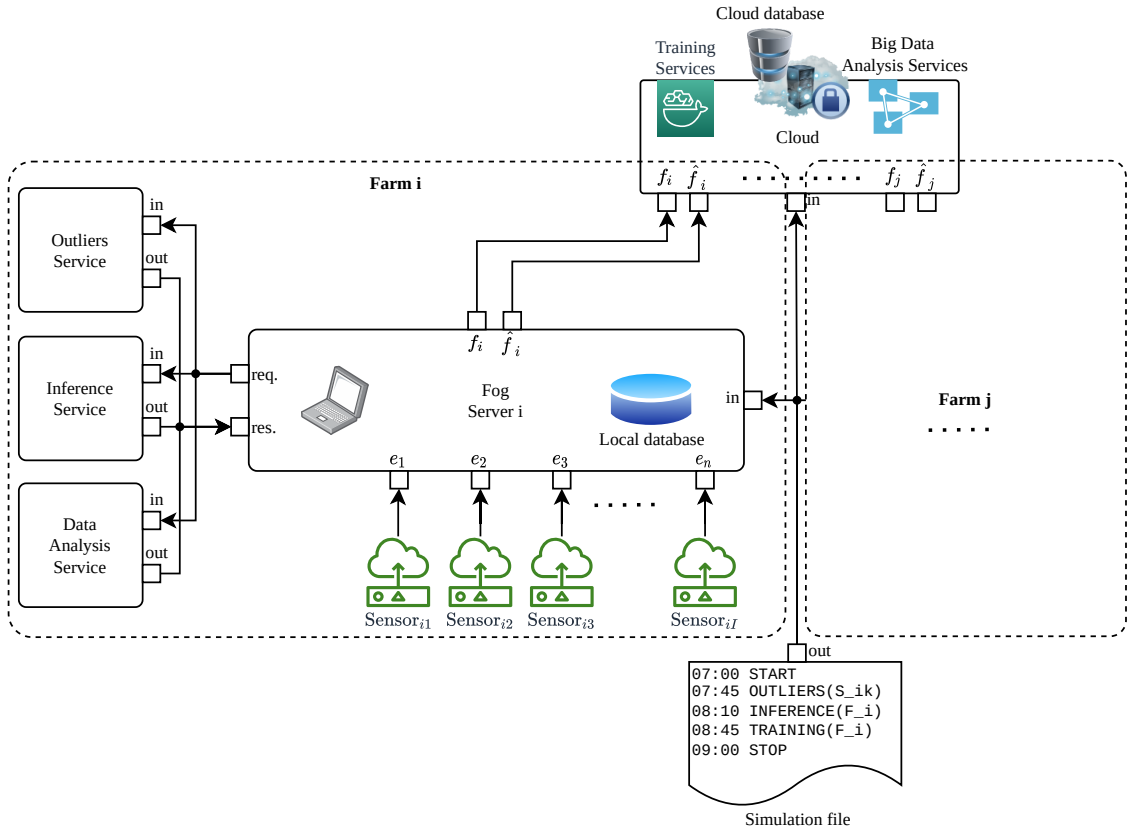


Fig. 3: DEVS system architecture.

IV. USE CASE

In this section, a simulation is performed using real monitoring data obtained from the Measurement and Instrumentation Data Center (MIDC), which come from the Oahu Solar Measurement Grid [13]. These data correspond to 17 sensors, located near the Honolulu airport and geographically distributed as shown in Figure 4. Thus, the system to be simulated includes 17 sensors in a single sensor farm.

Both outlier and inference services will be used during the simulation, as explained hereafter.

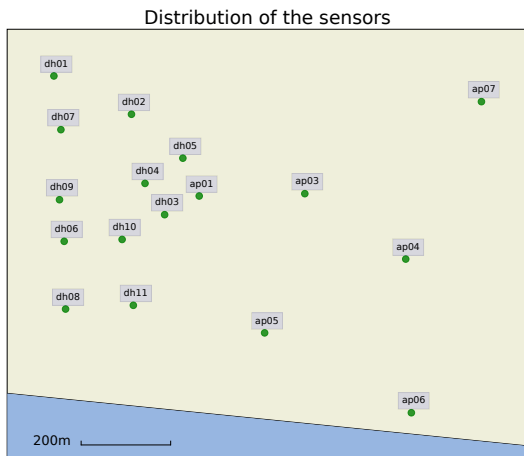


Fig. 4: Sensors in the Oahu Solar Measurement Grid.

A. Outlier service

The outlier service allows for both detection and correction of irradiance measurements that are not

in the same range as the remaining values from the same sensor in a time window.

In this particular use case, outlier detection was performed using the interquartile range algorithm. This algorithm calculates the first (Q_1) and third quartile (Q_3) and then performs the operation: $IQR = Q_3 - Q_1$. Subsequently, a threshold value is established according to the severity of detection to be obtained, which in the case analyzed corresponds to 50% of the interquartile range. The lower threshold is $T_1 = Q_1 - 1.5 \cdot IQR$ and the upper threshold is $T_2 = Q_3 + 1.5 \cdot IQR$. Therefore, all values in the analyzed data set that do not fall between T_1 and T_2 are considered outliers.

For the correction of outliers, we have used the Cubic Spline method, which calculates low-degree polynomials in each interval in order to obtain polynomial parts that fit in a linearity. Cubic Spline introduces a minor error in a linear regression and is recommended for continuous data sets. In our case, the irradiance sensor values are collected every second. Thus, the previous and next values usually have little variability.

Both detection and correction can be triggered as a result of a command in the simulation file. As an example, the following group of commands in Figure 5 would generate the results shown in Figure 6.

CAIDE reads and executes each command in the virtual time indicated by the `request_datetime` column. The simulator can not only execute outlier detection and correction commands, such as the one specified in the column `command_request`, but is also

request_datetime	command_request	initial_datetime	end_datetime	argument
21/03/2010 8:29	cubicspline	20/03/2010 7:30	20/03/2010 8:29	DataCenter FogServer01 dh1 120
22/03/2010 8:29	cubicspline	20/03/2010 7:30	20/03/2010 8:29	DataCenter FogServer01 ap1 120
23/03/2010 8:29	cubicspline	20/03/2010 7:30	20/03/2010 8:29	DataCenter FogServer01 dh2 120
24/03/2010 8:29	cubicspline	20/03/2010 7:30	20/03/2010 8:29	DataCenter FogServer01 ap3 120

Fig. 5: Commands to detect and correct outliers.

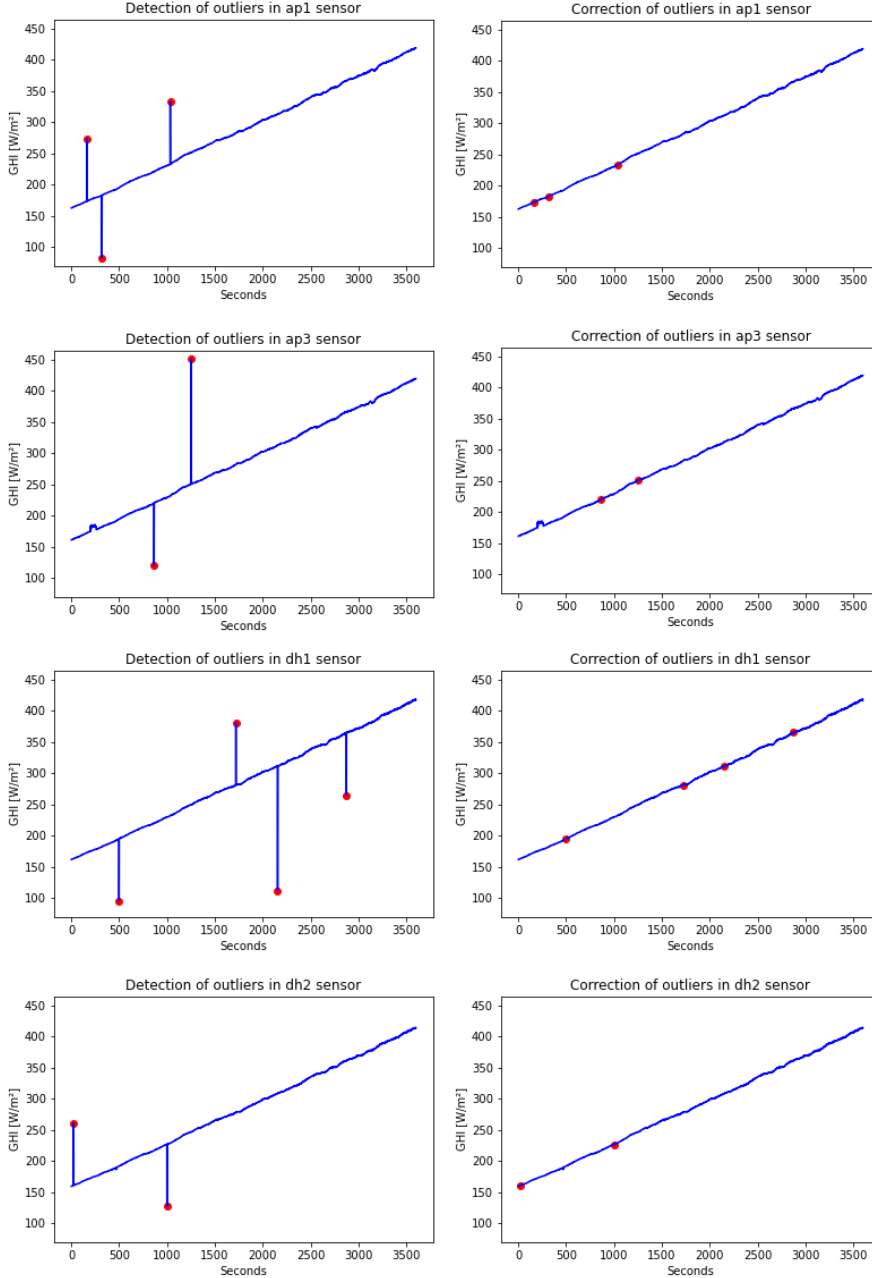


Fig. 6: Detection (left) and correction (right) of outliers.

capable of executing report generation, additional calculations that need to be performed on the sensor readings, and calls to other libraries or services, such as the execution of the inference discussed in this article. The time range over which the outlier detection and correction are performed is defined in columns `initial_datetime` and `end_datetime`. Likewise, the parameters that will be needed in the execution of this process are specified in the column `argument`, which contains the name of the sensor on which the command is requested to be executed.

The plots on the left of Figure 6 represent the irradiance measurements classified as outliers, whereas the blue lines are normal measurements. The plots on the right show the outliers that have been corrected as red points.

The large amount of data generated by each of the sensors (one for every second of operation) is evident, which means that post-storage data operations are resource intensive. To mitigate this problem and obtain timely computation times and data transformations, it was decided to use the non-relational Mon-

goDB database as a tool to support the fog server. With MongoDB, counting, aggregation, arithmetic mean and standard deviation and transposition operations were performed at a rate of 55 million data in less than 5 seconds (on a computer with 16 Gb RAM and 6 processor cores). These operations were necessary to provide the base data for the inference system. On the other hand, the search and update operations, which were used for outlier detection and correction, were executed in a matter of milliseconds. Despite the good response times, there is room for further improvement by adding nodes to the database server, thus contemplating any future growth needs of the CAIDE system in terms of data processing caused by some of the services offered.

B. Inference service

Inferences are also triggered as a result of a command in the simulation file. In this case, a DNN was trained with data collected by the pyranometers mentioned above (see Figure 4). For simplicity, the time granularity is converted into minutes, obtained as the average every 60 seconds.

Once the command is executed, the model reads the ten previous instances of the data from each sensor. From there, it produces a new file with the desired inferences for the specified forecast horizons. It is important to note that predictions can only be made within a time window of each day, based on the temporal size of the input tensor ($n_x = 10\text{min}$ in this case) and the farthest prediction horizon ($h = 61\text{min}$). For example, if data are collected daily from 7:30 to 17:30, the first inference will take as input data from the interval [7:30, 7:39] ($n_x = 10\text{min}$) to predict for instants {7:40, 7:50, 8:10, 8:40}. Similarly, the last inference of the day will be based on the data between [16:20, 16:29] to forecast GHI at the instants {16:30, 16:40, 17:00, 17:30}. In reality, predictions can be obtained up to the input interval [17:21, 17:30], but we cannot know the accuracy of the model after 17:30 if the sensors do not record irradiance values then. Furthermore, multiple consecutive predictions can be requested to the simulator at once, as long as the input window is available, as explained before. This feature can be useful for consecutive sample analyses as well as for evaluating the performance of the model itself by studying larger samples of predicted data.

To test the inference capabilities of our framework, we requested an inference service for a full day. The data were generated for 20/03/2010 ranging from 07:30 to 17:30, which leaves us with a total of 601 instants and 531 valid inferences. Once the data was generated, it took 9.4 seconds to simulate the 531 predictions for the four horizons. This time includes loading the relevant Python libraries, the data, and the model itself. Taking into account only the actual computation time, that figure is reduced to 7 seconds for a whole day of predictions and to less than a tenth of a second for a single prediction. Figure 7 shows the forecast for each of the four considered horizons,

	1min	11min	31min	61min
Skill MAE	15.8%	31.8%	37.2%	35.8%
Skill RMSE	24.4%	29.3%	27.8%	29.9%

Table I: Skills calculated for sensor ap01 on the 20/03/2010.

compared to the simulated values for a single sensor.

To assess the quality of our predictions, we can use the *forecast skill* metric, defined in Eq. 1:

$$S = \left(1 - \frac{\text{error}_{\text{proposed}}}{\text{error}_{\text{reference}}} \right) \cdot 100\% \quad (1)$$

This metric gives a performance improvement of a model with respect to that of a base model expressed as a percentage, and it is commonly employed in the field of solar irradiance forecasting. In our case, we used Mean Absolute Error (MAE) and Root-Mean-Squared Error (RMSE) as error metrics and a persistent model for comparison.

Table I shows the skill of the model for each horizon. As can be seen, our model shows improvements of up to 36% for 1 hour horizons when using MAE as error and up to 30% at the same horizon when using RMSE. In both cases, the proposed model shows a consistent improvement over the base model for all horizons.

V. CONCLUSION AND FUTURE WORK

The global demand for electricity is growing faster than the production of renewable energy. As a result, renewable energy generation is expected to grow strongly around the world over the next few years. To this end, it is necessary to envision well-structured and robust methods to analyze the appropriate geographical areas for the installation of power plants. Modeling and Simulation (M&S) can help with this task, as a secure and non-expensive way to have an initial picture of the final deployment project. However, current M&S approaches do not provide integrative solutions to this problem.

In this paper, we have introduced CAIDE, a novel integrative framework to enable real-time monitoring of solar irradiance sensor farms and to make decisions about the evolution of state-of-the-art predictive models, allowing us to detect outlier values and perform missing data estimation. Our framework is also designed to automatically manage the deployment of the infrastructure. Our approach is based on the solid principles of Model Based Systems Engineering (MBSE) and the Discrete Event System Specification (DEVS) M&S formalism. Furthermore, the entire infrastructure is built upon the IoT paradigm. All these elements allow for an incremental design, assuring reliability and scalability to the management of multiple farms. Our framework also allows different resolution views for the interpretation of a domain expert at the Fog layer and for the interpretation of authorities at the Cloud layer, following the IoT nomenclature.

Future work includes, on the one hand, the inclusion of new algorithms (e.g. related to the training of

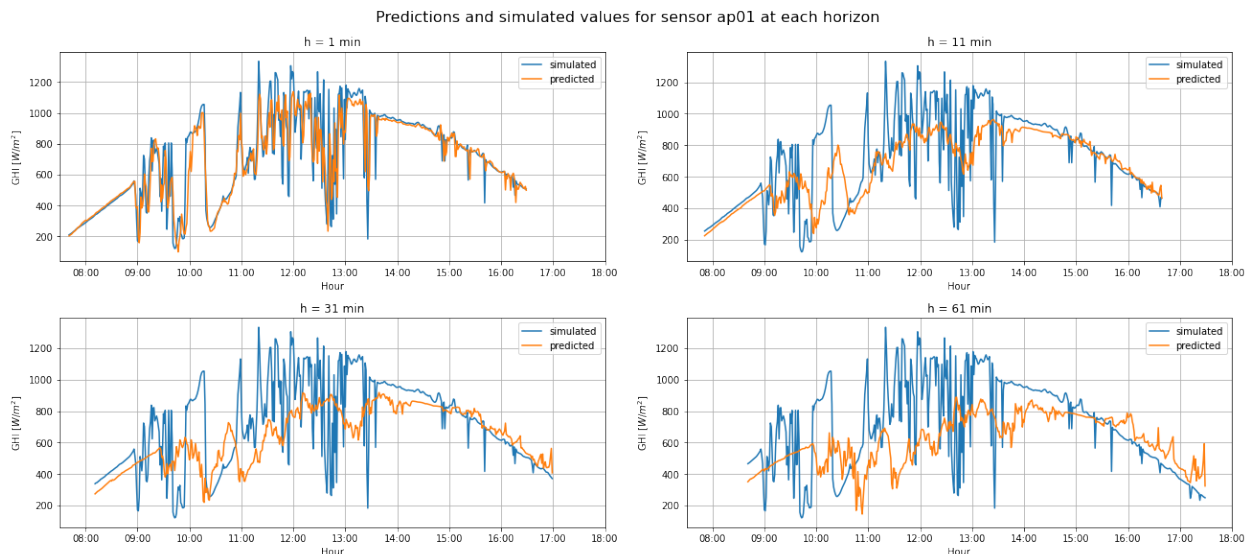


Fig. 7: Predicted and simulated values at each horizon for sensor ap01 on the 20/03/2010.

the Deep Learning (DL) models) in CAIDE, the improvement of visualization tools, or the integration of virtual sensors capable of generating synthetic data. On the other hand, the parallel and distributed nature of CAIDE will allow us to study the scalability of the system by specifying virtual farms within a large synthetic scenario.

Finally, we emphasize that having a scientific framework for predicting solar irradiation and taking management actions also provides an organizing principle for fundamental research. This will serve and benefit the engagement of theory with M&S foundations. Complementary research on mathematical models or systems engineering can be easily integrated into our CAIDE framework. This will improve the scientific exploitation of new discoveries and support the development of new bases for forecasting future effects, not only in the generation of solar energy, but also in other renewable energy challenges such as cost analysis or modeling of storage systems.

ACKNOWLEDGEMENTS

This work has been supported by the Autonomous Region of Madrid through the program CABAHLA-CM (GA No. P2018/TCS-4423).

REFERENCES

- [1] Stéphanie Bouckaert, Araceli Fernandez Pales, Christophe McGlade, Uwe Remme, Brent Wanner, Laszlo Varro, Davide D'Ambrosio, and Thomas Spencer, "Net zero by 2050: A roadmap for the global energy sector," <https://trid.trb.org/view/1856381>, June 2021.
- [2] "EU Solarw Energy Strategy," Tech. Rep., European Commission, 2022.
- [3] Hong-Tzer Yang, Chao-Ming Huang, Yann-Chang Huang, and Yi-Shiang Pai, "A weather-based hybrid method for 1-day ahead hourly forecasting of pv power

- output," *IEEE transactions on sustainable energy*, vol. 5, no. 3, pp. 917–926, 2014.
- [4] Wenqi Zhang, William Kleiber, Anthony R Florita, Bri-Mathias Hodge, and Barry Mather, "Modeling and simulation of high-frequency solar irradiance," *IEEE Journal of Photovoltaics*, vol. 9, no. 1, pp. 124–131, 2018.
- [5] Dazhi Yang, "A guideline to solar forecasting research practice: Reproducible, operational, probabilistic or physically-based, ensemble, and skill (ropes)," *Journal of Renewable and Sustainable Energy*, vol. 11, no. 2, pp. 022701, 2019.
- [6] Clara Arbizu-Barrena, José A Ruiz-Arias, Francisco J Rodríguez-Benítez, David Pozo-Vázquez, and Joaquín Tovar-Pescador, "Short-term solar radiation forecasting by advecting and diffusing msg cloud index," *Solar Energy*, vol. 155, pp. 1092–1103, 2017.
- [7] Alex Ayet and Pierre Tandeo, "Nowcasting solar irradiance using an analog method and geostationary satellite images," *Solar Energy*, vol. 164, pp. 301–315, 2018.
- [8] Ahmad Alzahrani, Pourya Shamsi, Cihan Dagli, and Mehdi Ferdowsi, "Solar irradiance forecasting using deep neural networks," *Procedia Computer Science*, vol. 114, pp. 304–313, 2017.
- [9] Ignacio-Iker Prado-Rujas, Antonio García-Dopico, Emilio Serrano, and María S Pérez, "A flexible and robust deep learning-based system for solar irradiance forecasting," *IEEE Access*, vol. 9, pp. 12348–12361, 2021, doi:10.1109/ACCESS.2021.3051839.
- [10] Bernard P. Zeigler, Alexandre Muzy, and Ernesto Kofman, *Theory of modeling and simulation: discrete event & iterative system computational foundations*, Academic press, 2018.
- [11] José L. Risco-Martín, Kevin Henares, Saurabh Mittal, Luis F. Almendras, and Katalin Olcoz, "A unified cloud-enabled discrete event parallel and distributed simulation architecture," *Simulation Modelling Practice and Theory*, 2022.
- [12] Ignacio-Iker Prado-Rujas, Emilio Serrano, Antonio García-Dopico, M. Luisa Córdoba, and María S. Pérez, "Predicción espacio-temporal: más allá del error/precisión," in *Congreso Español de Informática 2021 (CEDI'21)*, Málaga, Spain, Sept. 2021, doi:10.5281/zenodo.5530048.
- [13] M. Sengupta and A. Andreas, "Oahu Solar Measurement Grid (1-Year Archive): 1-Second Solar Irradiance; Oahu, Hawaii (Data)," doi:10.5439/1052451, March 2010, Type: dataset (last accessed 22/6/2022).