energies

MDPI

Article

Exploiting Digitalization of Solar PV Plants Using Machine Learning: Digital Twin Concept for Operation

Tolga Yalçin [1], Pol Paradell Solà [1,*], Paschalia Stefanidou-Voziki [2], Jose Luis Domínguez-García [1,*] and Tugce Demirdelen [3]

[1] Power Electronics Department, Catalonia Institut for Energy Research—IREC, Jardins de les Dones de Negre 1, 2ª pl., Sant Adrià del Besòs, 08930 Barcelona, Spain
[2] E.ON Digital Technology GmbH, Georg-Brauchle-Ring 52-54, 80992 Munich, Germany
[3] Departmentof Electrical and Electronics Engineering, Alparslan Turkes Science and Technology University—ATU, Balcalı Mah., South Campus 10 Street, No:1U, P.O. Box GP 561 Adana, Turkey
* Correspondence: pparadell@irec.cat (P.P.S.); jldominguez@irec.cat (J.L.D.-G.)

Abstract: The rapid development of digital technologies and solutions is disrupting the energy sector. In this regard, digitalization is a facilitator and enabler for integrating renewable energies, management and operation. Among these, advanced monitoring techniques and artificial intelligence may be applied in solar PV plants to improve their operation and efficiency and detect potential malfunctions at an early stage. This paper proposes a Digital Twin DT concept, mainly focused on O&M, to obtain more information about the system by using several artificial intelligence boxes. Furthermore, it includes the development of several machine learning (ML) algorithms capable of reproducing the expected behavior of the solar PV plant and detecting the malfunctioning of different components. In this regard, this allows for reducing downtime and optimizing asset management. In this paper, different ML techniques are used and compared to optimize the selected methods for enhanced response. The paper presents all stages of the developed Digital Twin, including ML model development with an accuracy of 98.3% of the whole DT, and finally, a communication and visualization platform. The different responses and comparisons have been made using a model based on MATLAB/Simulink using different cases and system conditions.

Keywords: Digital Twin; PV system; solar plant; machine learning; O&M systems

check for updates

Citation: Yalçin, T.; Paradell Solà, P.; Stefanidou-Voziki, P.; Domínguez-García, J.L.; Demirdelen, T. Exploiting Digitalization of Solar PV Plants Using Machine Learning: Digital Twin Concept for Operation. *Energies* **2023**, *16*, 5044. https://doi.org/10.3390/en16135044

Academic Editors: Manolis Souliotis and Sandro Nizetic

Received: 27 February 2023
Revised: 19 June 2023
Accepted: 22 June 2023
Published: 29 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Energy consumption is continually increasing globally, in parallel with the advancement of science and technology. To maintain a modern and appropriate technology level, nations must improve and sustain their energy resources. Today's principal challenge facing the energy sector is maintaining the balance between supply and demand. Furthermore, as the world population grows, the per capita consumption rate also increases, driven by technological advancements [1]. Thus, there exists a direct correlation between an individual's daily consumption rate in a country and the level of development of that country.

Energy resources are classified based on their consumption and convertibility. They are classified as renewable or non-renewable energy sources and primary or secondary energy sources. Non-renewable energy sources are finite, unchanging and discontinuous in nature and include fossil fuels such as oil, natural gas and coal. On the other hand, renewable energy resources can be replenished over time and are available for a prolonged period, including solar, wind, geothermal, biomass and hydro-power [2].

The economic feasibility and popularity of solar energy are increasing daily. However, regular solar energy monitoring is essential to ensure high efficiency and prevent problems. The importance of research in this field is directly related to the increase in the solar energy market share. The global solar energy market, which was valued at USD 86 billion in

2015, is projected to reach USD 422 billion by the end of 2022 [3]. It is estimated that approximately 2% of photovoltaic panels will fail after 11–12 years [4], and losses from dust collection (contamination) can be greater than losses from cell disruption. Therefore, the regular production monitoring and reporting of possible losses are essential to ensure early diagnosis and regular maintenance.

In the article by Rahman et al. [5], Artificial Neural Network (ANN) systems were explored for predicting renewable energy generation from solar, turbine and hydro-power sources. Similarly, Zheng et al. [6] utilized particle swarm optimization combined with long-short-term memory techniques to predict energy output from photovoltaic (PV) systems.

Various methodologies are reported in the literature for predicting the energy generated by photovoltaic systems. For example, some studies [7–9] have employed neural network techniques to make predictions of energy output. Additionally, a similar analysis has been applied to forecasting the temperature of photovoltaic modules. In addition, some of them focus [10,11] on the feature selection because it is believed that if it can be configured well, ML models can predict solar power better.

However, due to the non-linear and chaotic nature of solar power plants, the choice of prediction models must be made carefully. Therefore, this research decided to use the three most popular machine learning algorithms, and some of these models can work with few features while others prefer to have a larger set of variables. This is important because every stage type and feature amount will differ.

These articles describe the use of Digital Twin (DT) technology in various renewable energy systems. In [12], modules were designed to store, map and process data from a solar power plant to develop life-cycle management with DT. In [13], the authors designed an architecture, mathematical model and big data analytic engine to monitor the state of solar panels using DT. In [14], the authors proposed using DT for optimum control, virtual modeling and pre-diagnosis in production processes. In [15], it was suggested to use DT to monitor decentralized renewable energy sources in the electricity grid. In [16], the authors used DT to observe wind turbine fatigue failure and evaluate alternative processes for a floating wind turbine.

The articles reviewed in this study propose the use of Digital Twin (DT) technology to monitor and optimize various aspects of solar and wind power plants. The studies involve designing modules to collect and process data, creating virtual models of the physical systems and implementing AI algorithms and big data analytics to improve performance.

The implementation of Digital Twin (DT) technology faces a challenge in detecting errors or abnormalities, as it requires waiting for the entire cycle to complete, which slows down the system and reduces sensitivity. To overcome this, the authors suggest dividing the power plant into subsystems and using multiple models, each representing a specific component of the solar PV. This different idea provides detailed insights into the performance and health of individual components, enabling the identification of potential failures or degradation. Compared to the standard unique model version, this new approach, with three Digital twins (DT) inside one system, provides a comprehensive understanding of the overall power plant, facilitating proactive maintenance and optimizing performance.

Considering the aforementioned factors, this study aims to achieve the production and error detection of the system with machine learning models while creating the Digital Twins of the photovoltaic systems and transferring them to the virtual system. In summary, the paper's key contributions are focused on the development of an innovative DT for solar PV, which is based on the development of a DT of different components, allowing identification of the faulty component, which can be easily integrated into an online-based platform for real-time monitoring of a real SCADA system. The study is structured as follows: in Section 2, the methods used in this study and their working methods are explained. Section 3 is used to compare the results. Finally, Section 4 shows the conclusion. Although the paper provides results, the presented values come from simulations and lack real data due to the access restrictions of real environments.

2. Generalities on Machine Learning

Three different machine-learning algorithms were used in this research. Since our data are non-linear and based on time series, linear-based models would not perform well, so they were eliminated. These three regression models are Deep Neural Networks (DNN), Random Forest (RF) and CatBoost.

2.1. Random Forest Regression

Random Forest is a supervised machine learning technique that uses the group learning approach to perform classification and regression. Random Forest is built on the wisdom of crowds, which posits that a huge proportion of statistically independent models functioning together as a panel will outperform any constituent models individually [17,18]. This is because trees support each other by safeguarding one another from their faults. When creating branches, every tree pulls a representative selection from the raw data set, introducing an element of chance that avoids the fitting problem. A Random Forest regression model is effective and precise. It often outperforms many problems, including those with non-linear connections. However, there is no interpretability; over-fitting is readily possible, and the user must pick the number of trees to include in the model [19].

In a Random Forest regression, the final prediction for a new input x is typically made by averaging the predictions of the individual trees in the forest. Each tree i in the forest predicts x based on its own decision rules, represented by the function $f_i(x)$. The final prediction for x is the average of all the trees' predictions [20,21]:

$$y_{pred}(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x) \quad (1)$$

This equation represents the final prediction for a new input x. It is the average of the predictions made by each tree i in the forest, where N is the number of trees in the forest and $f_i(x)$ represents the decision rules (or function) of the tree i that predict x (Figure 1).

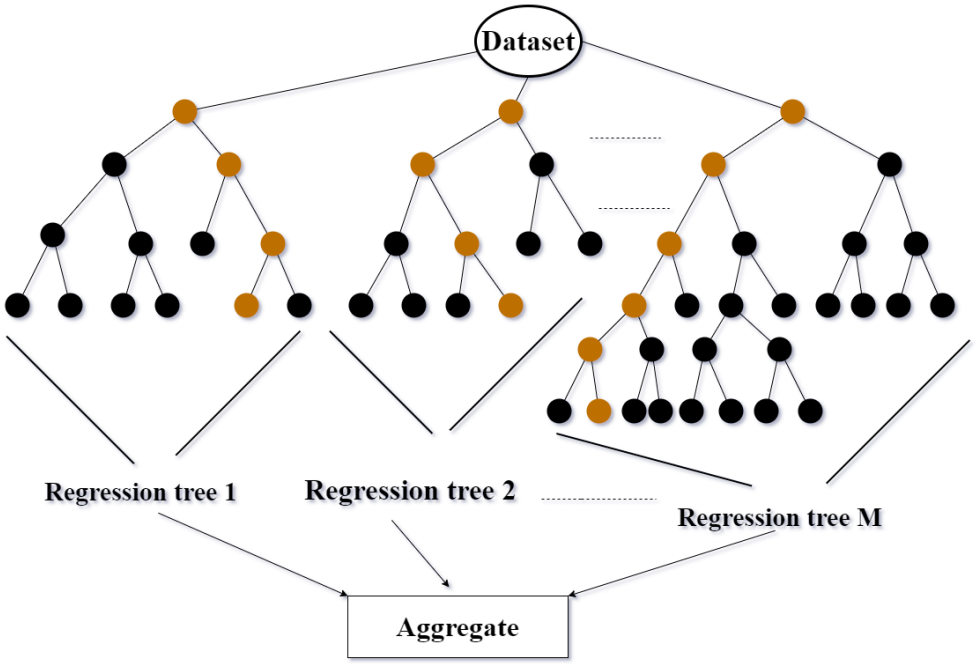

Figure 1. Random Forest architecture.

2.2. Deep Neural Network Regression

The algorithm consists of four steps: forward propagation, backpropagation to the output layer, backpropagation to the hidden layer and a weight updating process. In this

section, we present the main equations of the backpropagation algorithm with gradient descent for a three-layer neural network with a sigmoid activation function to relate it to the proposed hardware implementation [22,23]. In the forward propagation step, the dot product of input matrix X and weighted connections between the input layer and hidden layer w12 is calculated and passed through the sigmoid activation function: $Yh = \sigma(Xw12)$, where Yh is an output of the hidden layer. The forward propagation step is repeated in all the neural network layers. The output of the three-layer network Yo is calculated as $Yo = \sigma(Yhw23)$, where $w23$ is the matrix representing the weighted connections between the hidden and output layers [24]. The backpropagation algorithm uses the cost function defined in Equation (2) for the calculation of the derivative of the error concerning the weight change. In Equation (2), E is an error, N is the number of neurons in the layer, *ytarget* is an ideal output and *yreal* is the obtained output after the forward propagation [25].

Our DNN has five layers (one input layer, four hidden layers and one output layer) and the following parameters (Figure 2):

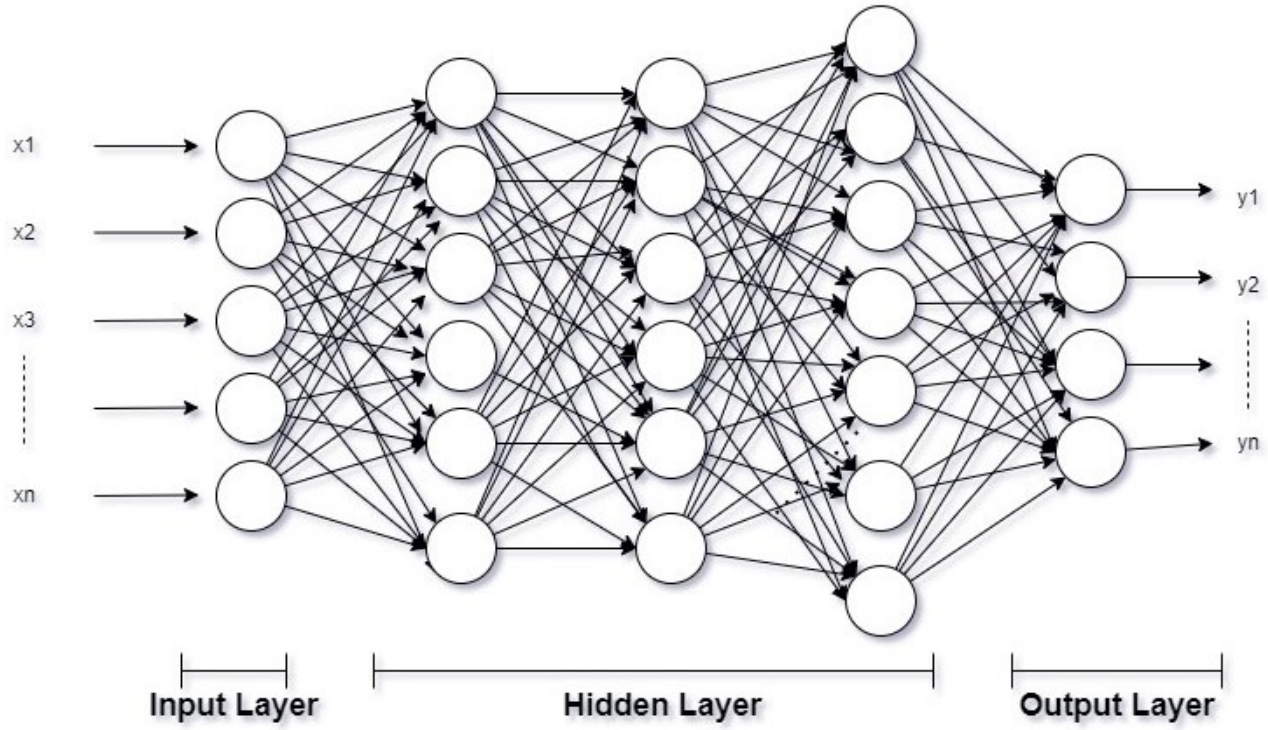

Figure 2. Deep Neural Network architecture.

Input layer: 6 neurons, represented by the column vector $\mathbf{x}$. First 3 hidden layers: 256 neurons each, represented by the column vectors $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}$. Fourth hidden layer: 512 neurons, represented by the column vector $\mathbf{h}^{(4)}$. Fifth hidden layer: 256 neurons, represented by the column vector $\mathbf{h}^{(5)}$. Output layer: 1 neuron, represented by the scalar y. The forward pass of the network would then be calculated as follows:

$$\mathbf{h}^{(1)} = relu\left(\mathbf{W}^{(1)} \cdot \mathbf{x} + \mathbf{b}^{(1)}\right)$$

$$\mathbf{h}^{(2)} = relu\left(\mathbf{W}^{(2)} \cdot \mathbf{h}^{(1)} + \mathbf{b}^{(2)}\right)$$

$$\mathbf{h}^{(3)} = relu\left(\mathbf{W}^{(3)} \cdot \mathbf{h}^{(2)} + \mathbf{b}^{(3)}\right)$$

$$\mathbf{h}^{(4)} = relu\left(\mathbf{W}^{(4)} \cdot \mathbf{h}^{(3)} + \mathbf{b}^{(4)}\right)$$

$$\mathbf{h}^{(5)} = relu\left(\mathbf{W}^{(5)} \cdot \mathbf{h}^{(4)} + \mathbf{b}^{(5)}\right)$$

$$y = \mathbf{W}^{(6)} \cdot \mathbf{h}^{(5)} + b^{(6)} \tag{2}$$

where *relu* is the rectified linear unit (ReLU) activation function.

The backpropagation algorithm would then be used to update the weights and biases in order to minimize the error between the predicted output y and the true output y_{true} using the mean squared error (MSE) as the loss function [26]:

$$E = \frac{1}{2}(y - y_{true})^2 \tag{3}$$

The gradients of the error function concerning each weight and bias in the network can be calculated using the chain rule.

Once the gradients have been calculated, the weights and biases can be updated using the Adam optimization algorithm [27], which is a variant of stochastic gradient descent that uses moving averages of the gradients and second moments to adjust the learning rate adaptively:

$$\mathbf{m}^{(t)} \leftarrow \beta_1 \mathbf{m}^{(t-1)} + (1 - \beta_1)\frac{\partial E}{\partial \mathbf{W}^{(5)}}$$

$$\mathbf{v}^{(t)} \leftarrow \beta_2 \mathbf{v}^{(t-1)} + (1 - \beta_2)\left(\frac{\partial E}{\partial \mathbf{W}^{(5)}}\right)^2$$

$$\mathbf{m}_{hat}^{(t)} \leftarrow \frac{\mathbf{m}^{(t)}}{1 - \beta_1^t}$$

$$\mathbf{v}_{hat}^{(t)} \leftarrow \frac{\mathbf{v}^{(t)}}{1 - \beta_2^t}$$

$$\mathbf{W}^{(5)} \leftarrow \mathbf{W}^{(5)} - \alpha \frac{\mathbf{m}hat^{(t)}}{\sqrt{\mathbf{v}hat^{(t)}} + \epsilon} \tag{4}$$

where $beta_1$ and $beta_2$ are the decay rates for the moving averages, $alpha$ is the learning rate, and *epsilon* is a small constant to prevent division by zero.

This process is repeated for the weights and biases in the other layers of the network.

2.3. Catboost Regression

CatBoost is a gradient-boosting library that uses decision trees as the base model. It is specifically designed to handle categorical variables and also includes some other features such as handling missing values and built-in cross-validation [28].

Like many other gradient-boosting libraries, CatBoost builds the model by training a sequence of decision trees. Each tree is trained on the residuals of the previous trees, where the residual is the difference between the true target value and the predicted value of the previous trees [29].

The mathematical expression for a single decision tree in CatBoost is determined by the specific algorithm used to construct the tree. However, in general, a decision tree is a series of simple decisions (or "splits") based on the values of the input features [30]. For example, a decision tree might split on the value of an input feature X and make different predictions depending on whether X is greater than or less than a certain threshold.

A CatBoost model is an ensemble of multiple decision trees, and the final prediction is made by summing the predictions of the individual trees (Figure 3).

The mathematical expression of the CatBoost regression model is given as follows:

$$y_{pred}(x) = \sum_{i=1}^{T} w_i h_i(x) \tag{5}$$

where T is the number of trees in the model, w_i is the weight assigned to the i-th tree, and $h_i(x)$ is the prediction made by the i-th tree. The weights are learned during the training process and are used to adjust the contribution of each tree to the final prediction.

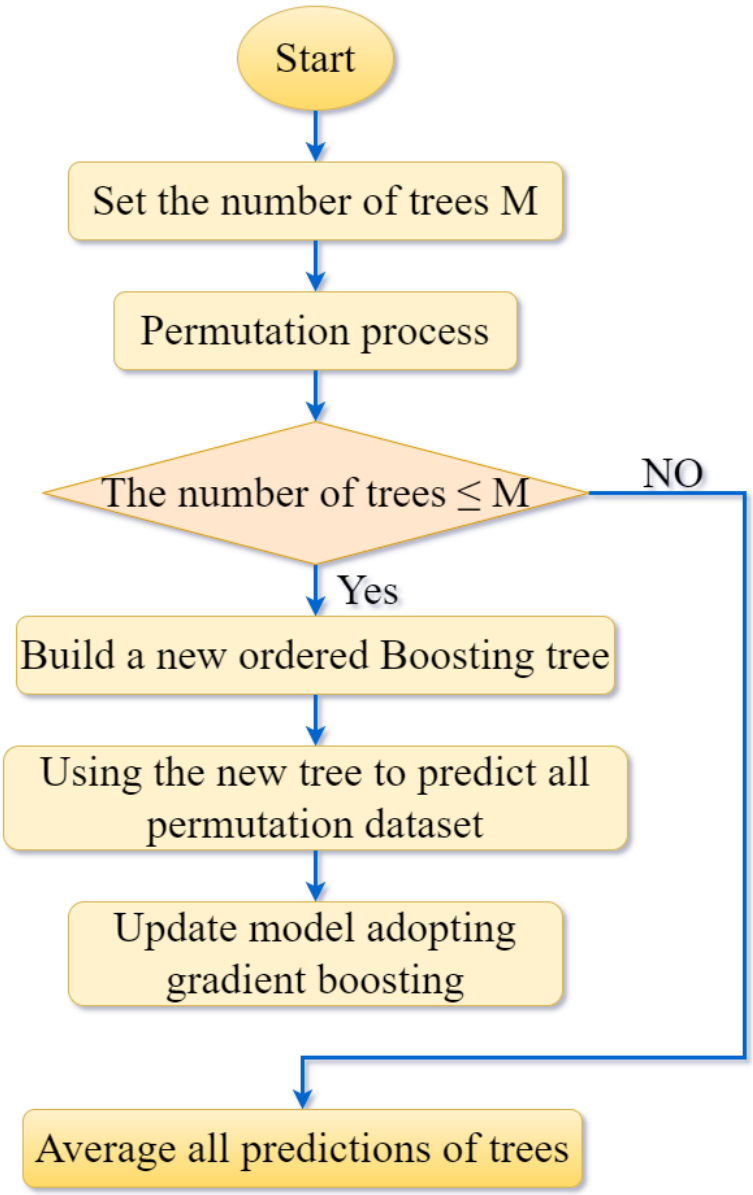

Figure 3. Flowchart of the Catboost regression.

CatBoost also uses some techniques that help to handle categorical features better than regular gradient-boosted decision trees (GBDT), such as handling the categorical features themselves and using permutation-based feature importance [31].

3. Methodology of Digital Twin

The fundamental objective of the DT concept is to enable real-time monitoring and detailed analysis of a solar panel system through a virtual model (Figure 4). With data from the real PV plant, the trained model can predict the system's behavior using one of the already explained machine learning methods. With all of this, the results from the ML model and the ones from the real plant can be compared to determine if there is any deviation and warn the responsible party to take countermeasures.

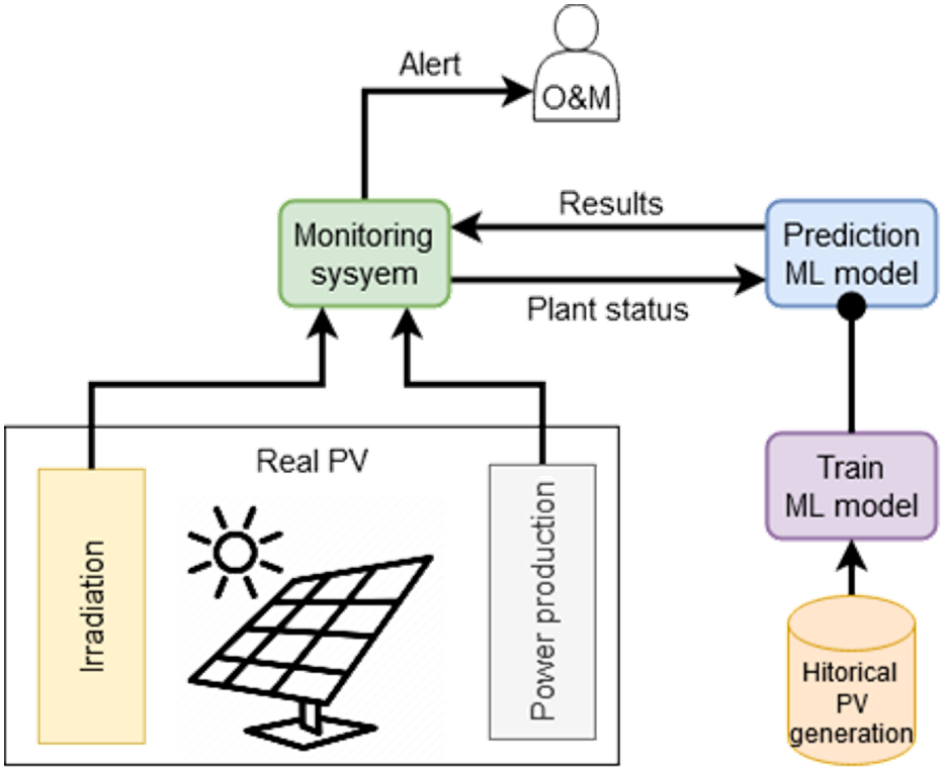

Figure 4. Digital Twin for solar plant architecture.

All the data can come from different sources, such as different IoT devices. The unique condition is to have a time stamp and the minimum data to make a correlation between all the inputs. This approach simplifies the process by consolidating all necessary information on a single platform and obviates the necessity for intricate and burdensome systems that entail an abundance of data. The details of the internal architecture are depicted in Figure 5. The platform uses docker to split the components into small modules that are easy to manage and maintain. To interact with the external elements and to receive the data, we created a REST API supported by the FasAPI framework. The machine learning element component uses the framework sckitlearn and keras. It uses the Redis database to receive the orders to perform prediction or re-training. To store the data, it uses Influx DB—a tool capable of managing time series data efficiently and sufficiently fast to hold all the needed data. Finally, the tool called Grafana can be used to visualize the data from the different sources and also the Digital Twin.

The methodology used to obtain the results follows four different steps. First, the weather data are collected from the PVGIS system for several years. The second step is to use a part of the obtained data to run the power plant model using Matlab/Simulink and obtain the experimental data as if the plant was a real plant. Then, the model uses the generated data to train the ML models. Finally, after all these steps, the DT is run using the same plant model again, but in this case, the weather data that have not been used from the previous steps are used. In this study, there were no data from a real plant, so the two initial steps were needed. All these steps used docker to build containers, which are Influx DB, grafana, Redis, ML models and FastAPI.

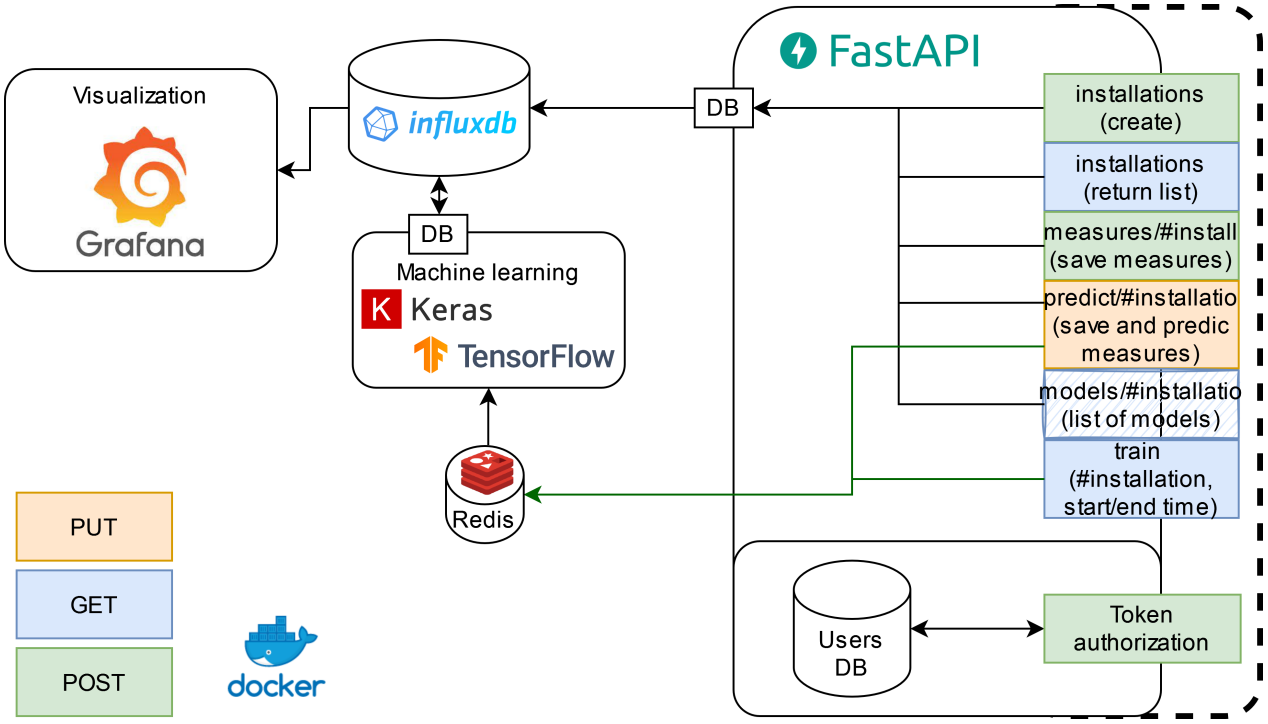

Figure 5. Digital Twin internal architecture.

4. Results

In Figure 6, there is a comparison between the original concept of DT, where there is a unique model of the system to predict the whole system, and the proposed concept of a "box of boxes", where it has the same inputs and outputs but also has other intermediate variables that provide more information about the twin system.

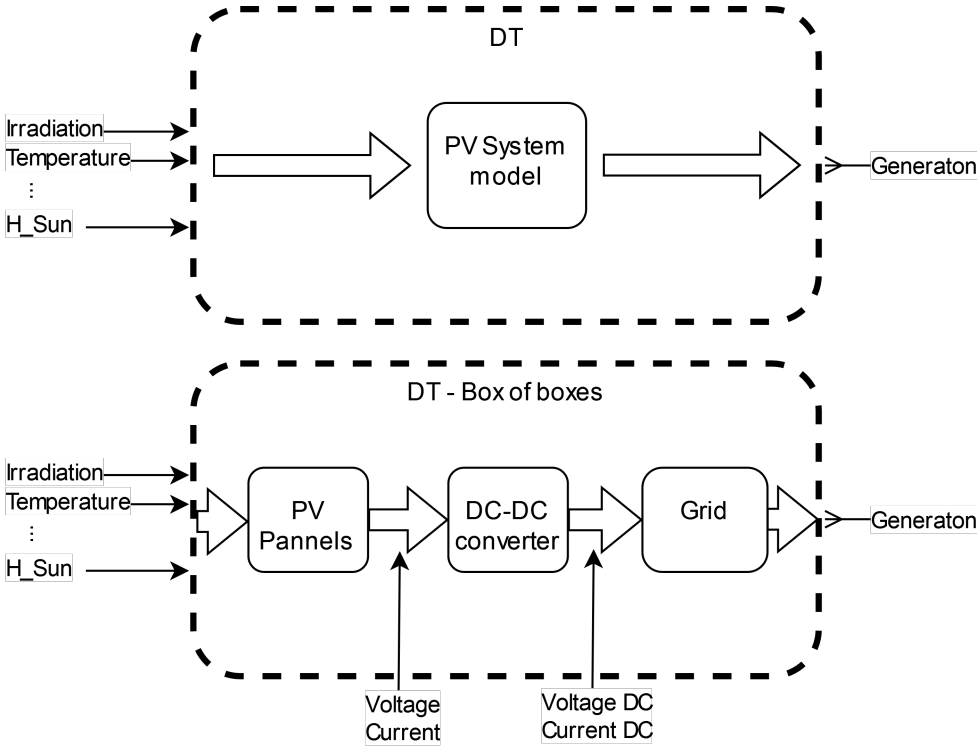

Figure 6. General diagram comparing the regular DT with the new proposed concept of a box of boxes.

The data used for these results are generated using a Matlab/Simulink model of a solar system with 150 kW of power as a real installation (Figure 7) [32]. In addition, the weather input data from PVGIS for two years are used to evaluate the first part as training and the second for evaluation [33]. An installation with fixed panels at 30 degrees has been used to evaluate this research. The idea of this concept is to have a DT designed for a specific installation. If it changes—for example, the slope of the panels or the system tracks the sun—then the models have to be retrained with the data of this new configuration.

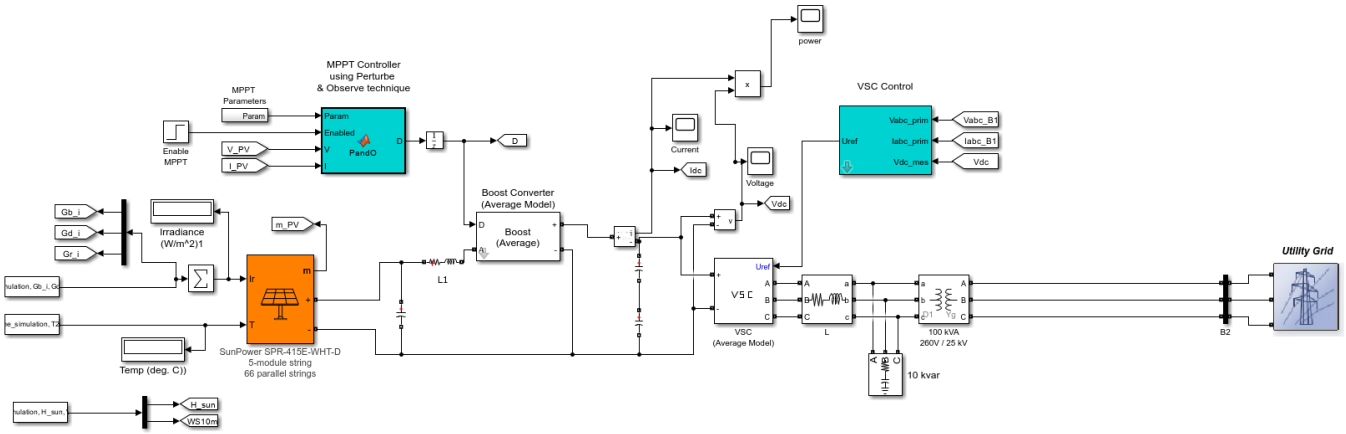

Figure 7. The Matlab/Simulink model used in this work to generate the data.

Solar energy plants consist of many complex parts, and they work intertwined with each other; with the proposed design in this study (Figure 8), we aim to reduce the complexity by examining the whole system in three parts separately from each other. However, on the other hand, the whole system is like a chain reaction, and it is desired to emphasize that there is a natural bond between them.

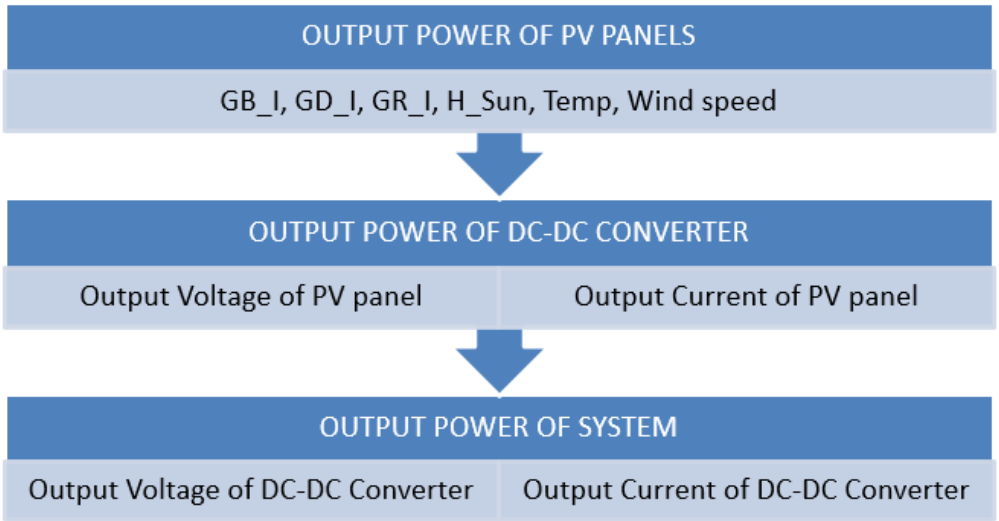

Figure 8. This graph describes the input variables used to train machine learning algorithms and the corresponding outputs they aim to predict.

This study utilized various variables for training and estimating algorithms based on examining solar panel systems and identifying key system characteristics. The relationships and connections between these variables were thoroughly analyzed to optimize the data structure for optimal results (Figure 8).

4.1. Machine Learning

This section explains, for each part of the system of a PV plant, the results obtained comparing the three chosen machine-learning methods, evaluating the performance of each in the three different situations. In this study, we collected data for a period of two years, with data points recorded every hour. This resulted in a dataset containing 17,544 rows for each parameter measured. The large amount of data collected allowed for a thorough analysis of trends and patterns in the measured parameters over time. In this research study, a data partitioning approach was employed, whereby 30% of the available data were reserved for testing purposes, while the remaining 70% were partitioned into training and validation sets. Specifically, 80% of the data were allocated to the training set, and the remaining 20% were designated as the validation set.

4.1.1. Pv Panel Part

The solar panel component is the central focus of this study. It is heavily dependent on various input variables and requires preprocessing for estimation, except for electricity generation. Furthermore, given that its performance is affected by weather conditions and environmental factors, it requires continuous monitoring and protection.

The first stage of this study focuses on the direct effect of various types of radiation and temperature on solar panel performance. If Figure 9 is examined, there is a linear relationship between power generation and irradiation types and temperature. In contrast, this link is not as clear as the others in $Hsun$, which specifies the height of the sun (degrees). While it is stated in the literature that wind speed $(WS10m)$ does not has a significant impact, this study found that these variables still have an effect. This effect comes indirectly because, with the wind, the panels can be covered with sand or vice versa.

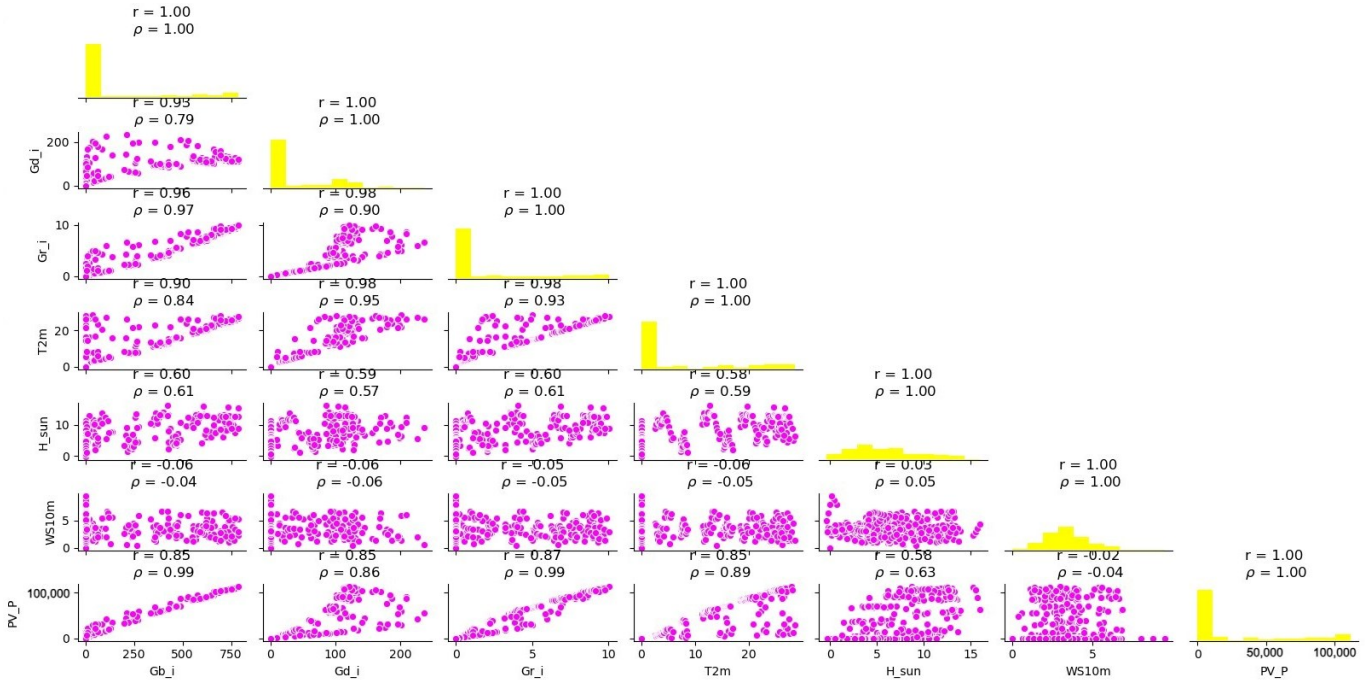

Figure 9. The correlation matrix of the variables used in this research.

Figure 10 compares real electrical energy output with projections given by several machine learning algorithms. To evaluate the performance of our model, we selected four random days for testing in each season. Specifically, the selected days were 1 January, 7 March, 3 June and 3 September. The data collection for these testing days began at 3 am and continued until 4 am the next day, providing a sufficient amount of data to test the model's accuracy and generalizability. It is evident from the comparison that the current generation graph (represented in blue) and the prediction graph (represented in

red) exhibit a high degree of similarity for all days analyzed using the different methods. This implies that the machine learning techniques' predictions capture the trends in the data. Among the different methods used, it is noteworthy that the only method whose estimates are below the peak values of the actual data for each observed day is the Catboost method.

CatBoost does not perform as well as RF or DNN, but it is the fastest algorithm to train and predict, but the error rate cannot be considered acceptable. DNN and RF gave perfect results, but still it is hard to say which one performed better from these images. As a result, the prediction error must be calculated to measure the forecasts' accuracy. The RMSE and MAE values for the comparative approaches are shown in Table 1.

Table 1. Values of error measures for validating the compared techniques.

	DNN	RF	CatBoost
RMSE	0.8	6.10	12.20
MAE	0.2	3.06	8.80

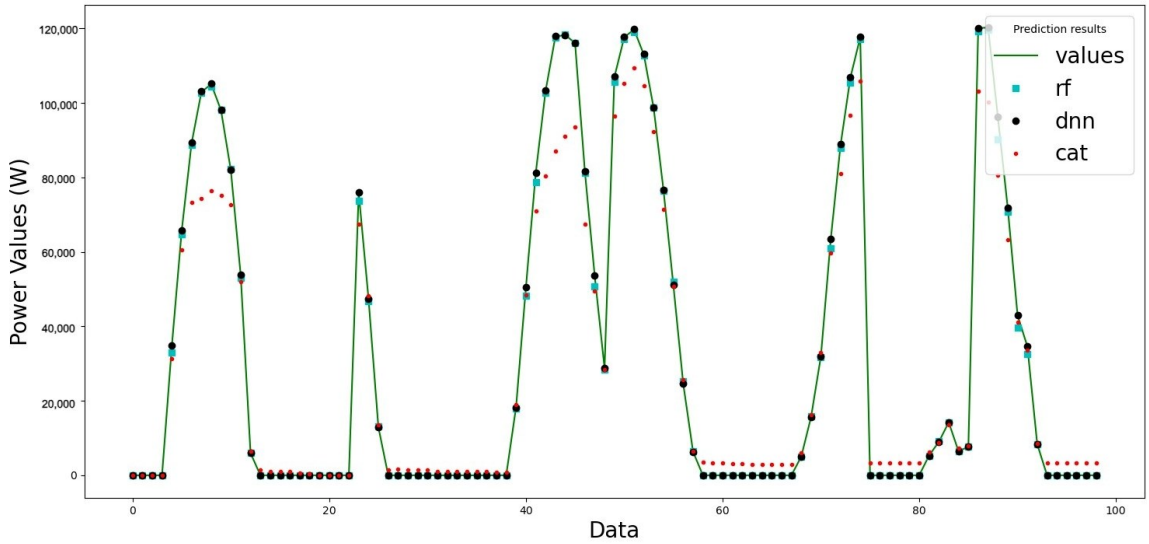

Figure 10. Comparison between the actual production results with the predictions made by ML models.

As depicted in the aforementioned Table 1, the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) values for each of the proposed machine learning techniques are minimal, indicating that the estimation for the test dataset is satisfactory. Both metrics indicate that the DNN model yields the least prediction error. Conversely, the Catboost method yields the highest RMSE and MAE values. Despite the slight variations in the errors obtained for all methods, they are of a similar magnitude. This can be attributed to the lack of significant variations in the meteorological conditions of the location under test.

4.1.2. DC–DC Converter Part

Unlike PV panels, DC–DC converters exhibit nonlinear behavior and are dynamic systems that may adapt quickly to changes in the system. The semiconductor devices utilized in the converter, as well as the nonlinear phenomena generated by parasitic capacitances and inductances in the system, are principally responsible for this non-linearity [34]. As a result, DC–DC converters are intrinsically nonlinear, making accurate modeling by machine learning techniques difficult. Furthermore, because of the semiconductor architectures utilized, the quick reaction time of DC–DC converters offers extra challenges for machine learning algorithms with large gaps between sample periods [35]. To address this challenge, the machine learning algorithm employs a MIMO (Multiple Input Multiple

Output) architecture, which allows for a more detailed analysis of the system by estimating both current and voltage as output.

Figure 11 reveals that the predictive performance of all DC current models is excellent, and this is not an easy task, considering that the current values fluctuate sporadically from 0 to 250 amps. However, these good results make it difficult to decide on one of them, so it is necessary to check the voltage estimation results in Figure 12.

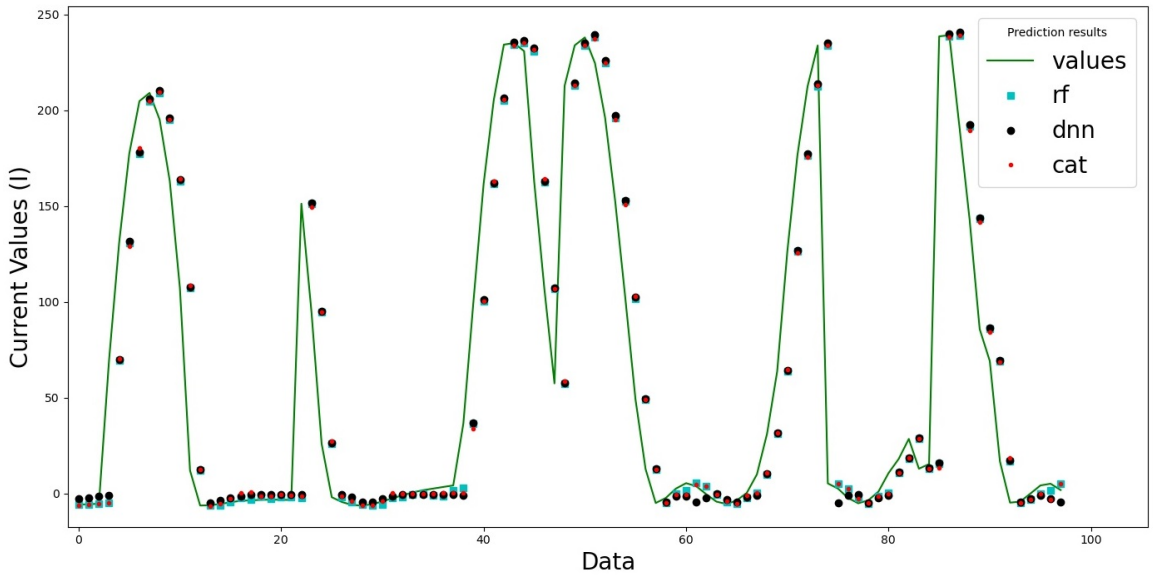

Figure 11. Comparison between the actual current values of the DC–DC part with the predictions made by ML models.

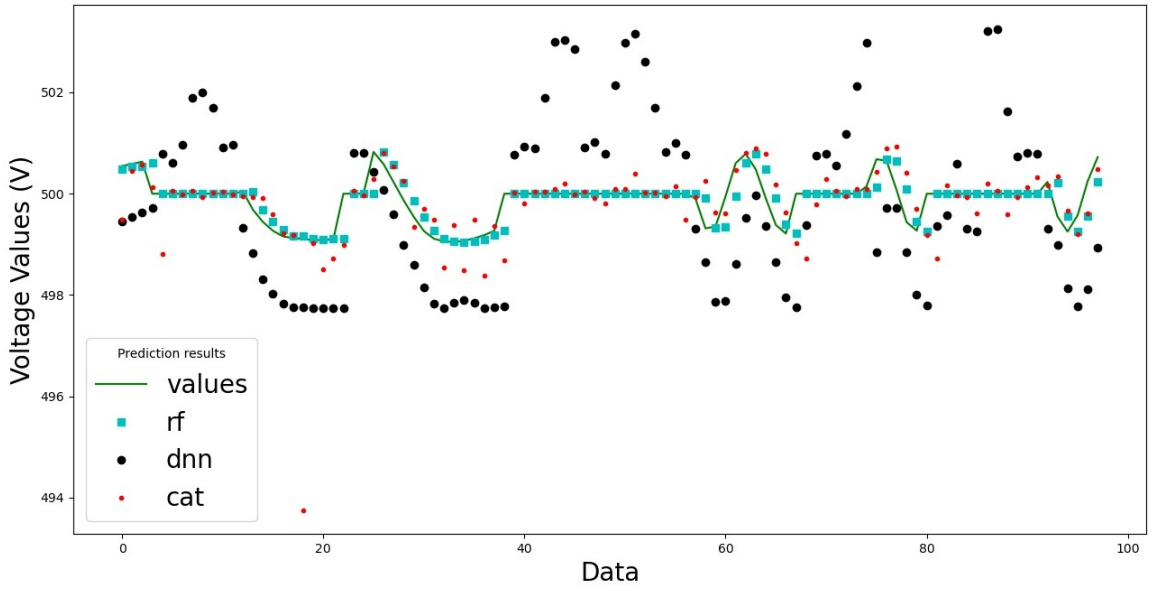

Figure 12. Comparison between the actual voltage value of the DC–DC part with the predictions made by ML models.

In Figure 12, the situation is completely different, because the voltage values only range from 499 V to 501 V, and most are clustered above 500 V. ML models have difficulty predicting these stable movements. As shown in the graph, DNN usually makes predictions above or below the true value, while CatBoost was unable to accurately predict the upper and lower values, and the average remained around 500 V, so the graph shows that RF predicted with the best accuracy.

It can be confirmed with the help of Table 2 when comparing the model in terms of MAE and RMSE that RF had the best numbers, while DNN showed the worst. This may lead to the idea that tree-based models give better results because they cannot go beyond the maximum and minimum values shown in the training data.

Table 2. Values of error measures for validating the compared techniques.

	DNN	RF	CatBoost
RMSE	2.58	0.59	1.25
MAE	1.68	0.17	0.67

4.1.3. Grid Part

In this section, variables from the DC part have been used as input (Figure 8), and we have tried to determine whether there is any loss in the system due to extra resistors, some cables or malfunctions of some electronic devices, but it should be noted here that ML models cannot say anything about the type or cause of the loss but only show the damage these losses caused to the system.

When looking at Figure 13, it is clear that all the models give very good results. To see which one performs better, a graph can usually give us an idea, but here again, all the points are on top of each other, which means that a perfect fit has been reached. This raises the question of whether there is an over-fitting or not, but to avoid this, the used data were shuffled to different months and different hours of the day and night. Moreover, as explained from the PV part, the environmental weather conditions of the test place are pretty stable for the whole year, so predicting the grid part is also quite a straightforward process.

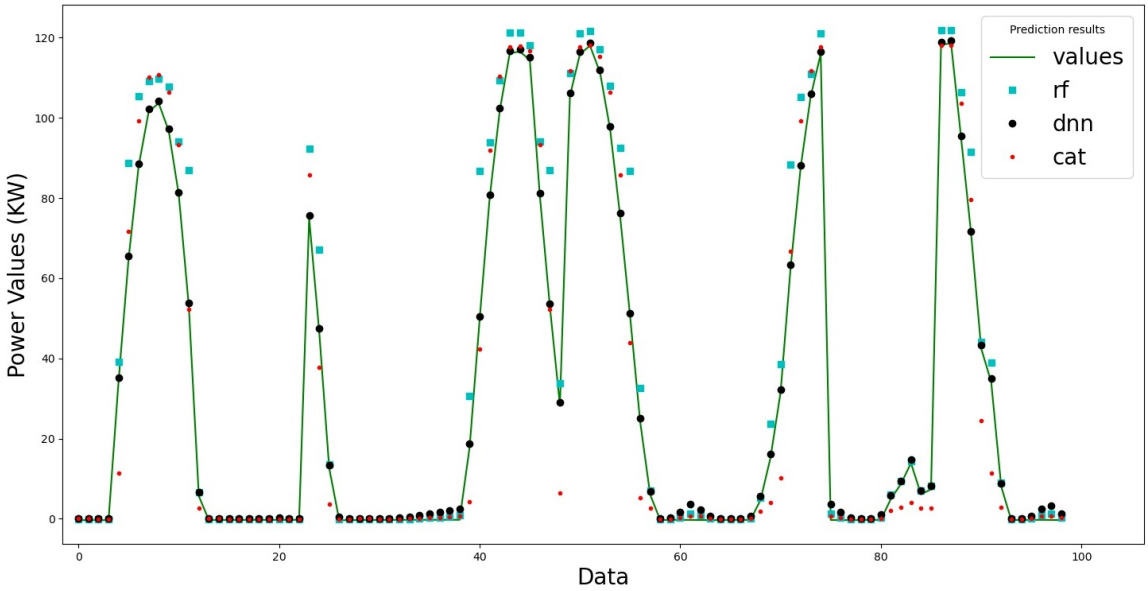

Figure 13. Hourly comparison of actual grid Power data against the estimation made using various techniques.

Figure 13 shows that all the models predict the same values, so it is necessary to check Table 3 to see the details and decide which one is the best algorithm. Thus, based on the RMSE and MSE, we can choose one model to use for the grid part.

It is evident that all models exhibit excellent performance, with Mean Squared Error (MSE) or Root Mean Squared Error (RMSE) values below one. Therefore, it is justifiable to select a model based on its categorization. Here, three criteria were used to make the decision: the size, speed and complexity of the model. Since Catboost was the winner of

these criteria, it was preferred as the main model, while the RF model showed the best performance in terms of error rates.

Table 3. Values of error measures for validating the compared techniques.

	DNN	RF	CatBoost
RMSE	0.19	0.02	0.3
MAE	0.10	0.01	0.2

4.2. Digital Twin

Integrating Digital Twin technology and Internet of Things (IoT) devices is a promising approach for the real-time monitoring and analysis of power grid systems. Utilizing a Digital Twin concept, a virtual replica of the physical system allows for a seamless connection between IoT devices and data analytics, enabling rapid assessment and real-time decision-making based on reliable data.

One example of such an application is the use of Digital Twin technology in predicting the electricity production of a solar power plant to the grid. These predictions from the system can be used to compare with the plant, the real twin, to increase any deviation from the expected results, increasing the system's reliability.

Figure 14 depicts the Grafana charts that allow for real-time system examination and alerting. The top graph shows the prediction of machine learning applied to the electricity passing through the grid. The central dashboard displays the error rate between the estimated and actual power. This study uses a tolerance rate of 20% as an example, and if the error rate exceeds this threshold, the system sends an error to the user. The bottom portion of the figures also displays the types of irradiation, with data obtained and sent to the system via IoT devices. This aspect of the system can be further supported with additional sensors to facilitate monitoring.

The idea to split the model into small models starts with the thesis that if there is a deviation from the Digital Twin, something is not correct, but there is no more information. Dividing the system into small pieces allows us to see the location of the issue and facilitates decision making. However, this method needs more data not only for the training but also for monitoring. It needs the status of intermediate points of the system and forces digitalization, adding more sensors to the parts of the system that require more information.

Figure 14. Dashboard in Grafana with model input data with grid part.

5. Conclusions

The paper has presented an innovative concept of an AI-based Digital Twin as a "box of black boxes". This innovative concept is different from the previous research in this field. Instead of focusing on one big model, the authors designed a concept like a puzzle, creating small parts and connecting them. This research culminated in the development of three unique AI models, each representing a different component of the overall solar PV power plant system with a global accuracy of 98.3%. These models allow for gathering complex and granular insights into the detection and evaluation of possible faults or performance deterioration inside specific components that are part of the overall system. The investigation included an in-depth assessment of each model using three machine-learning algorithms to discover the most appropriate approaches. Notably, the findings revealed that the performance of various strategies differed depending on the individual system components with the investigation's distinct traits and qualities.

Further steps of this research are to test this development in a real field and apply this concept to another energy generation system where there is the need to create a DT, such as wind turbines and other renewable resource systems. Furthermore, in future work, there is the need to not only use the error and its variations to consider a good fit, but also, the amount of data and time for the training could be important depending on the application.

Author Contributions: Conceptualization, P.P.S., J.L.D.-G. and P.S.-V.; methodology, P.P.S., P.S.-V. and J.L.D.-G.; software, P.P.S. and T.Y.; validation, T.Y. and P.S.-V.; formal analysis, P.S.-V. and J.L.D.-G.; investigation, T.Y.; resources, P.P.S.; data curation, T.Y.; writing—original draft preparation, T.Y.; writing—review and editing, T.Y., P.P.S., J.L.D.-G. and T.D.; visualization,T.Y. and P.P.S.; supervision, P.P.S., J.L.D.-G. and T.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PV	Photovoltaic
DC	Direct current
AC	Alternating current
MAE	Mean absolute error
MSE	Mean square error
RMAE	Root mean absolute error
RF	Random forrest
DNN	Deep neural network
KW	Kilowatt
GW	Gigawatt
DT	Digital Twin
IoT	Internet of Things
MIMO	Multiple Input Multiple Output
AI	Artificial intelligence
MLP	Multi layer perceptron
ML	Machine learning
O&M	Operation and maintenance

References

1. BP. bp Energy Outlook 2022; Technical Report. 2022. Available online: https://www.bp.com/content/dam/bp/businesssites/en/global/corporate/pdfs/energy-economics/energy-outlook/bp-energy-outlook-2022.pdf (accessed on 25 January 2023).
2. International Renewable Energy Agency. *Renewable Capacity Highlights*; Irena: Abu Dabi, United Arab Emirates, 2021; pp. 1–3.
3. Giving Intelligence Teams an AI-Powered Advantage; Technical Report. 2019. Available online: https://www.reportlinker.com/ (accessed on 20 January 2023)

4. Hernández-Callejo, L.; Gallardo-Saavedra, S.; Alonso-Gómez, V. A review of photovoltaic systems: Design, operation and maintenance. *Sol. Energy* **2019**, *188*, 426–440. [CrossRef]
5. Rahman, M.M.; Shakeri, M.; Tiong, S.K.; Khatun, F.; Amin, N.; Pasupuleti, J.; Hasan, M.K. Prospective methodologies in hybrid renewable energy systems for energy prediction using artificial neural networks. *Sustainability* **2021**, *13*, 2393. [CrossRef]
6. Zheng, J.; Zhang, H.; Dai, Y.; Wang, B.; Zheng, T.; Liao, Q.; Liang, Y.; Zhang, F.; Song, X. Time series prediction for output of multi-region solar power plants. *Appl. Energy* **2020**, *257*, 114001. [CrossRef]
7. Martín, L.; Zarzalejo, L.F.; Polo, J.; Navarro, A.; Marchante, R.; Cony, M. Prediction of global solar irradiance based on time series analysis: Application to solar thermal power plants energy production planning. *Sol. Energy* **2010**, *84*, 1772–1781. [CrossRef]
8. Testad, I.; Corbett, A.; Aarsland, D. ORE Open Research Exeter. *J. Clean. Prod.* **2013**.
9. Essam, Y.; Ahmed, A.N.; Ramli, R.; Chau, K.W.; Idris Ibrahim, M.S.; Sherif, M.; Sefelnasr, A.; El-Shafie, A. Investigating photovoltaic solar power output forecasting using machine learning algorithms. *Eng. Appl. Comput. Fluid Mech.* **2022**, *16*, 2002–2034. [CrossRef]
10. Gutiérrez, L.; Patiño, J.; Duque-Grisales, E. A comparison of the performance of supervised learning algorithms for solar power prediction. *Energies* **2021**, *14*, 4424. [CrossRef]
11. O'Leary, D.; Kubby, J. Feature selection and ANN solar power prediction. *J. Renew. Energy* **2017**, *2017*, 2437387. [CrossRef]
12. Zheng, Y.; Yang, S.; Cheng, H. An application framework of digital twin and its case study. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 1141–1153. [CrossRef]
13. Asimov, R.M. Digital twin in the analysis of a big data. In Proceedings of the 4th International Conference on Scientific Practice "Big data and Advanced Analysis", "Big data and high-level analysis", Minsk, Republic of Belarus, 3–4 May 2018.
14. He, R.; Chen, G.; Dong, C.; Sun, S.; Shen, X. Data-driven digital twin technology for optimized control in process systems. *ISA Trans.* **2019**, *95*, 221–234. [CrossRef] [PubMed]
15. Nguyen, V.H.; Tran, Q.T.; Besanger, Y.; Jung, M.; Nguyen, T.L. Digital twin integrated power-hardware-in-the-loop for the assessment of distributed renewable energy resources. *Electr. Eng.* **2022**, *104*, 377–388. [CrossRef]
16. Pimenta, F.; Pacheco, J.; Branco, C.M.; Teixeira, C.M.; Magalhaes, F. Development of a digital twin of an onshore wind turbine using monitoring data. *J. Phys. Conf. Ser.* **2020**, *1618*. [CrossRef]
17. Yuchi, W.; Gombojav, E.; Boldbaatar, B.; Galsuren, J.; Enkhmaa, S.; Beejin, B.; Naidan, G.; Ochir, C.; Legtseg, B.; Byambaa, T.; et al. Evaluation of random forest regression and multiple linear regression for predicting indoor fine particulate matter concentrations in a highly polluted city. *Environ. Pollut.* **2019**, *245*, 746–753. [CrossRef] [PubMed]
18. Ishwaran, H.; Lu, M. Standard errors and confidence intervals for variable importance in random forest regression, classification, and survival. *Stat. Med.* **2019**, *38*, 558–582. [CrossRef]
19. Zhang, J.; Ma, G.; Huang, Y.; Sun, J.; Aslani, F.; Nener, B. Modelling uniaxial compressive strength of lightweight self-compacting concrete using random forest regression. *Constr. Build. Mater.* **2019**, *210*, 713–719. [CrossRef]
20. Mercadier, M.; Lardy, J.P. Credit spread approximation and improvement using random forest regression. *Eur. J. Oper. Res.* **2019**, *277*, 351–365. [CrossRef]
21. Singh, B.; Sihag, P.; Singh, K. Modelling of impact of water quality on infiltration rate of soil by random forest regression. *Model. Earth Syst. Environ.* **2017**, *3*, 999–1004. [CrossRef]
22. He, T.; Kong, R.; Holmes, A.J.; Nguyen, M.; Sabuncu, M.R.; Eickhoff, S.B.; Bzdok, D.; Feng, J.; Yeo, B.T. Deep neural networks and kernel regression achieve comparable accuracies for functional connectivity prediction of behavior and demographics. *NeuroImage* **2020**, *206*, 116276. [CrossRef] [PubMed]
23. Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Netw.* **1991**, *4*, 251–257. [CrossRef]
24. Xu, Y.; Du, J.; Dai, L.R.; Lee, C.H. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 7–19. [CrossRef]
25. Bosse, S.; Maniry, D.; Müller, K.R.; Wiegand, T.; Samek, W. Deep Neural Networks for No-Reference and Full-Reference Image Quality Assessment. *IEEE Trans. Image Process.* **2018**, *27*, 206–219. [CrossRef] [PubMed]
26. Achieng, K.O. Modelling of soil moisture retention curve using machine learning techniques: Artificial and deep neural networks vs support vector regression models. *Comput. Geosci.* **2019**, *133*, 104320. [CrossRef]
27. Du, J.; Tu, Y.; Dai, L.R.; Lee, C.H. A regression approach to single-channel speech separation via high-resolution deep neural networks. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 1424–1437. [CrossRef]
28. Massaoudi, M.; Refaat, S.S.; Abu-Rub, H.; Chihi, I.; Wesleti, F.S. A hybrid Bayesian ridge regression-CWT-catboost model for PV power forecasting. In Proceedings of the 2020 IEEE Kansas Power and Energy Conference (KPEC), Manhattan, KS, USA, 13–14 July 2020; pp. 1–5.
29. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased boosting with categorical features. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 6639–6649.
30. Hancock, J.T.; Khoshgoftaar, T.M. CatBoost for big data: An interdisciplinary review. *J. Big Data* **2020**, *7*, 94. [CrossRef]
31. Zhang, Y.; Ma, J.; Liang, S.; Li, X.; Li, M. An evaluation of eight machine learning regression algorithms for forest aboveground biomass estimation from multiple satellite data products. *Remote Sens.* **2020**, *12*, 4015. [CrossRef]
32. Detailed Model of a 100-kW Grid-Connected PV Array. 2022. Available online: https://es.mathworks.com/help/sps/ug/detailed-model-of-a-100-kw-grid-connected-pv-array.html (accessed on 25 January 2023).

33. European Commission, Joint Research Centre Energy Efficiency and Renewables Unit. Photovoltaic Geographical Information System. Available online: https://re.jrc.ec.europa.eu/pvg_tools/en/ (accessed on 25 January 2023) .
34. Martínez, R.; Bolea, Y.; Grau, A.; Martínez, H. Fractional DC/DC converter in solar-powered electrical generation systems. In Proceedings of the ETFA 2009—2009 IEEE Conference on Emerging Technologies and Factory Automation, Palma de Mallorca, Spain, 22–25 September 2009; pp. 1–6. [CrossRef]
35. Del Moral, D.L.; Barrado, A.; Sanz, M.; Lazaro, A.; Fernandez, C.; Zumel, P. High efficiency DC-DC autotransformer forward-flyback converter for DMPPT architectures in solar plants. In Proceedings of the 2015 9th International Conference on Compatibility and Power Electronics, CPE 2015, Costa da Caparica, Portugal, 24–26 June 2015; pp. 431–436. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.