

Article

Deep and Machine Learning Models to Forecast Photovoltaic Power Generation

Sergio Cantillo-Luna ¹, Ricardo Moreno-Chuquen ^{2,*}, David Celeita ³ and George Anders ⁴¹ Faculty of Engineering, Universidad Autónoma de Occidente, Cali 760030, Colombia² Faculty of Engineering and Design, Universidad Icesi, Cali 760031, Colombia³ School of Engineering, Science and Technology, Universidad del Rosario, Bogotá 111221, Colombia; david.celeita@urosario.edu.co⁴ Faculty of Engineering, Technical University of Lodz, 90-924 Lodz, Poland

* Correspondence: rmoreno@icesi.edu.co

Abstract: The integration and management of distributed energy resources (DERs), including residential photovoltaic (PV) production, coupled with the widespread use of enabling technologies such as artificial intelligence, have led to the emergence of new tools, market models, and business opportunities. The accurate forecasting of these resources has become crucial to decision making, despite data availability and reliability issues in some parts of the world. To address these challenges, this paper proposes a deep and machine learning-based methodology for PV power forecasting, which includes XGBoost, random forest, support vector regressor, multi-layer perceptron, and LSTM-based tuned models, and introduces the ConvLSTM1D approach for this task. These models were evaluated on the univariate time-series prediction of low-volume residential PV production data across various forecast horizons. The proposed benchmarking and analysis approach considers technical and economic impacts, which can provide valuable insights for decision-making tools with these resources. The results indicate that the random forest and ConvLSTM1D model approaches yielded the most accurate forecasting performance, as demonstrated by the lowest RMSE, MAPE, and MAE across the different scenarios proposed.



Citation: Cantillo-Luna, S.; Moreno-Chuquen, R.; Celeita, D.; Anders, G. Deep and Machine Learning Models to Forecast Photovoltaic Power Generation. *Energies* **2023**, *16*, 4097. <https://doi.org/10.3390/en16104097>

Academic Editor: Carlo Renno

Received: 17 February 2023

Revised: 28 March 2023

Accepted: 26 April 2023

Published: 15 May 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/>).

Keywords: deep learning; machine learning; PV power forecasting; time-series analysis

1. Introduction

Recently, there has been a growing trend towards the widespread adoption of distributed generation (DG) powered by renewable energy. As a result, there is an increasing need to properly integrate and manage these resources into the existing infrastructure, which has become a major challenge for the energy industry, specially in regions highly dependent on energy resources strongly affected by climatic conditions [1]. These systems are subject to fluctuations in power generation, storage capacity, and demand, which can threaten the reliability and stability of the energy system as a whole [2].

The management and integration of distributed energy resources (DERs) have created new opportunities for the development of new business and market models that leverage the benefits offered by these technologies. Several enabling technologies and paradigms, such as blockchain, the Internet of Things, and especially artificial intelligence (AI), have emerged to facilitate these developments [3]. For example, the integration of PV systems with advanced grid management methods using AI can enable smart power flow optimization and reduce system losses. This can lead to the creation of new business models that allow consumers to trade excess energy and participate in demand–response programs.

In this context, AI has gained considerable interest as a means of managing and integrating DG assets, including local- and residential-scale photovoltaic (PV) systems.

The increasing popularity of residential PV systems is due to the availability of different incentives, such as the declining costs of PV panels, government incentives, new

revenue streams [4], and the consumer awareness of the environmental benefits of renewables. As a result, the residential sector has become a significant contributor to the growth of DERs, which are recognized as an essential component of the future power grids. However, the variability and uncertainty inherent in the inclusion of these PV systems into power grid operation has created new challenges.

The accurate forecasting of PV production is one of the most important and worthwhile tasks. The application of AI in this field has become increasingly common, as demonstrated by several developments, approaches, and studies [5–20]. It is important to note that among the most widely used approaches in these PV forecasting models are those based on the application of time series.

The accurate forecasting of PV production using time series is a difficult task, due to the highly nonlinear and complex nature of PV power output, as well as the low availability and quality of the data generated by these PV systems, especially in developing countries. Over the years, a variety of classic statistical methods based on time series have been applied for this purpose. Thus, methods such as auto-regressive moving average model (ARMA) and its other enhanced variations (AR, MA, SARIMA, ARIMA, and ARMAX), and exponential smoothing, among others, have been widely used for short-term PV production forecasting [21–26]. However, these methods have serious scalability, data volume, and uncertainty and variability robustness issues.

Therefore, the integration of AI techniques in PV production forecasting has become a research area of increasing importance in recent years. A wide range of AI paradigms, including machine learning (ML) and deep learning (DL), have been proposed as alternative methods for forecasting PV production.

The use of ML techniques for PV production forecasting has received considerable attention in recent years. In fact, several models applying machine learning techniques have been applied for this purpose. For example, in [27,28], different forecasting models were developed by applying support vector regressors (SVRs) based on pure or hybrid approaches. The models were used for 1 h ahead PV power production forecasting using historical solar data. The model applied performance metrics such as RMSE, MAPE, and R^2 . On the other hand, in [29–33], various short-term forecasting models were developed using pure and hybrid tree-based ML techniques, applying performance metrics such as R^2 and RMSE.

Likewise, DL-based architectures were recently proposed as PV power forecasting models due to their robustness and adaptability, since they can be easily and accurately fitted when nonlinear data patterns and relationships are provided. However, this feature tends to increase the complexity of these models. Consequently, the number of data and the amount of time required to train these forecasting models may be too large to apply in practice.

To overcome the size problem, modifications of the DL-based models were proposed in [34–38]. The authors developed deep RNNs and LSTM-based pure and hybrid models using real weather time-series and synthetic data to predict short-term (i.e., 1 h to 4 h) and very-short-term PV power production (i.e., 1 to 15 min). They used R^2 and normalized RMSE (nRMSE) as performance metrics. Meanwhile, in [39–41], the authors used different structures based on convolutional networks (CNN and CNN-LSTM, among others) to predict PV power production in the short term (hourly), medium term (daily), and long term (weekly), respectively. They used solar-related weather databases in 15-min intervals, and RMSE, MAE, and MBE performance metrics when developing these models.

The great variety of applied models and forecasting methods indicated above requires solid benchmarking approaches (i.e., techno-economic) to the selection of the method(s) best suited for this task. This topic is one of the the main contributions of the paper, as outlined in the next section.

Key Contributions

It is noteworthy to state that multiple research studies and advancements using DL and ML approaches have been conducted in the area of household-scale PV forecasting. However, these advancements have been applied with a reduced number of constraints in terms of PV capacity (i.e., regulatory and income constraints, among others), as well as insufficient availability of reliable PV production data and related exogenous variables. In fact, as these developments are mostly geographically targeted in areas where the solar resource is seasonally affected, meteorological data are a key input. In these circumstances, these advancements and thus their analyses are not really applicable in other areas, especially in the context of some developing countries.

This highlights the need for a comprehensive study that shows the development of different ML and DL forecasting models in areas with constant solar resource throughout the year with low availability of time-series data, in order to create robust and reliable tools for DER decision making based on these predictions considering different forecast windows. Table 1 summarizes important and recent developments related to the application of ML and DL approaches in residential PV forecasting tasks using time series. The levels of coverage and complexity related with scalability analysis (i.e., multiple forecast horizons performance, multi-criterion analysis, and elapsed time, among others) of each presented development are categorized as high (H), moderate (M), or low (L).

Table 1. Summary of some important developments on residential PV power forecasting.

Ref.	Input Data	PV Cap.	Scalability Analysis	Data Availability	Model Presented
[14]	PV power ^a	3 kW	M	36 months (30 min res)	LSTM ConvLSTM2D
[40]	PV power	5 kW	M	72 months (15 min res)	CNN CNN-LSTM
[42]	PV power Irradiance	15 kW	M	14 months (1 h res)	CNN-LSTM ConvLSTM2D
[43]	PV power Temperature	20 kW	H	37 months (7.5 min res)	ALSTM
[44]	Irradiance	NS ^b	L	21 months (15 min res)	CNN+GAN
[45]	PV power	10 kW	H	13 months (1 h res)	SVR
[46]	Sun angles Irradiance Cloud cover Temperature	4.5 kW	M	12 months (1 h res)	XGB RF

^a More than 1 PV installation; the PV cap value is the average. ^b NS: not specified.

To achieve the objectives outlined above, this paper presents the development, comparative benchmarking (MAE, RMSE, R^2 , and MAPE), and analysis of six different methods of DL and ML. We also introduce an additional method used in energy-related forecasting tasks, namely, a ConvLSTM1D-based model for different short-term forecasting of PV power production considering data availability and reliability issues. The aim of this study is to reduce potential barriers to the integration of residential-scale PV systems by empowering different stakeholders in the development of PV power forecast models to create decision-making tools with these results.

The key contributions of this paper are summarized as follows:

- Firstly, this study provides a comprehensive benchmark comparison of seven models (extreme gradient boosting algorithm (XGB), support vector regressor (SVR), random forest (RF), classic multi-layer perceptron (MLP), and LSTM-based models) that forecast 15 min, 30 min, and 1 h ahead of residential PV power production considering data availability

constraints (i.e., only a small amount of PV power production data and limited PV capacity (lower than 2 kW)) and a scalability analysis from a techno-economic perspective.

- Subsequently, we introduce a model based on stacked ConvLSTM1D layers that has been used in various energy-related prediction applications based on one-dimensional time-series forecasts. The performance of this model is benchmarked against other forecasting models that share operational similarities, such as LSTM-based models.
- Lastly, this study discusses some issues of the analyzed forecasting models and evaluates their usefulness for the development of new computational decision-making tools for the effective management and integration of this type of DER.

The paper is structured as follows: The methodology and theoretical method used for data preparation are presented in Section 2. This section contains a description of different ML and DL models, including their development and testing. Section 3 details the case study applying PV power output time-series data analysis. It discusses the different model hyperparameter tuning approaches, the forecasting outcomes, and the corresponding technical and economic analyses. Finally, Section 4 presents some concluding remarks.

2. Methodology

The methodology developed in this paper is divided into three parts: (1) Data preparation, which involves the conversion of the PV power output time series into a dataset compatible with supervised, non-time-dependent learning tasks. The data were split into training, validation, and test sets for each proposed model. (2) Tuning and development of the proposed ML and DL models based on the resulting datasets and different tests. (3) Model assessment applying selected benchmarks, and the testing of the proposed model's performance using different metrics and forecasting windows. Figure 1 summarizes the applied computational framework.

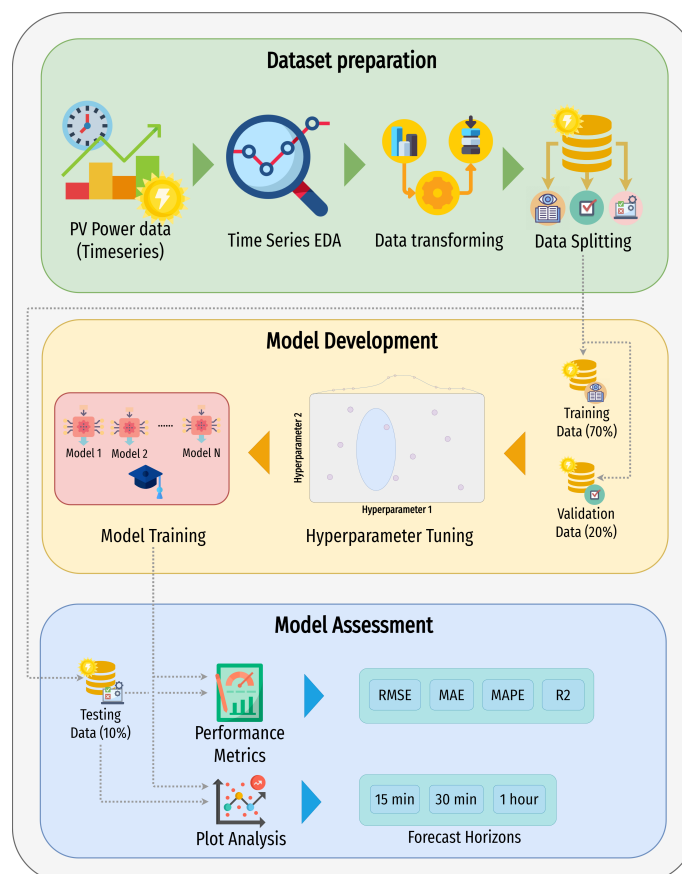


Figure 1. Framework of the proposed computational algorithm, including PV power output time-series conversion, data splitting, model development, comparison, and assessment.

2.1. Dataset Preparation

Time series are evenly spaced and ordered sequences of data collected at regular intervals (i.e., hourly, daily, weekly, etc.). As a consequence, there is a possibility of correlation among the response variables. It should be noted that many ML and DL algorithms used in regression tasks are ineffective when data are collected that way because of their way of fitting or learning (mainly due to multicollinearity and non-stationarity issues). This situation also applies to PV power output forecasting data.

Thus, to construct a dataset for the different ML and DL methods examined in this paper, it is necessary to restructure this time series by creating a transformed dataset with input features (X) and respective output variables (Y) (i.e., a supervised learning task dataset). Since PV production presents daily seasonality, the lag period method was chosen for dataset preparation using sliding windows, as shown in Figure 2.

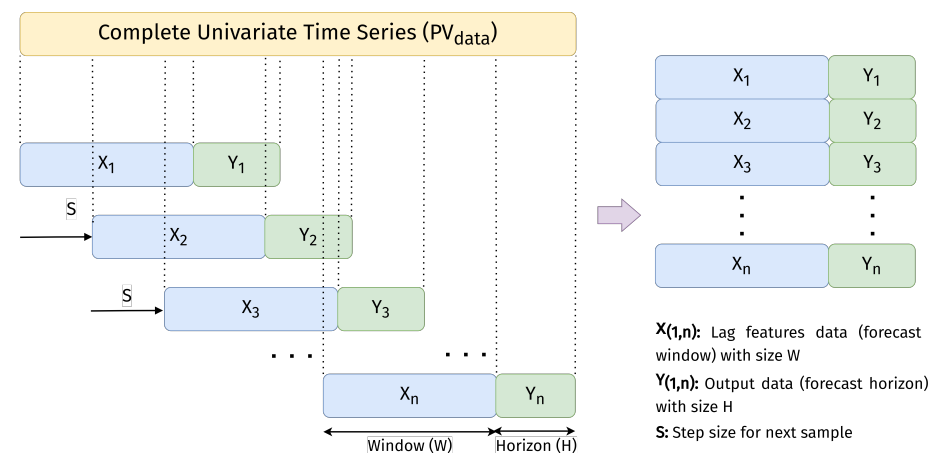


Figure 2. Dataset preparation with a lag period sliding window method.

To perform this dataset transformation, it was required to know how many lags were necessary according to the temporary nature of this time series. Therefore, a small exploratory data analysis (EDA) was performed, and an autocorrelation function (ACF) was implemented to determine this value.

The transformed dataset was divided into training, validation, and test sets, especially for neural network-based methods. It is important to mention that by transforming the time-series dataset into one focused on non-time-dependent supervised learning (due to the high seasonality of the data), this division can already be randomized. For the analysis presented in this paper, the dataset was split in the following proportions: 70%, 20%, and 10% for training, validation, and testing, respectively.

2.2. ML and DL Model Development

The forecasting models applied in this work are subjected to deep and machine learning algorithms, both commonly used in different disciplines, including energy-related regression tasks. It is important to mention that for the proposed models, the respective datasets were used for hyperparameter tuning with techniques such as random and genetic algorithm grid searches (i.e., RandomGridSearchCV and GASearchCV, respectively) and the geometric pyramid rule over neural net-based structures; subsequently, the respective fitting or training was performed.

2.2.1. Extreme Gradient Boosting Algorithm (XGB)

Extreme gradient boosting is a boosted tree-based algorithm (i.e., an ensemble method) that belongs to the supervised machine learning area for both regression and classification tasks. The XGB algorithm was introduced in 2016 by Chen and Guestrin in [47]. XGB is an improved and scalable version of the boosted gradient, consisting of weak base learners such as decision trees iteratively merging into a stronger one [31]. However, unlike other

algorithms based on this same approach, it is computationally faster and more efficient, as it uses parallel and distributed processing [38,48].

The learning process of these models consists in enhancing the fitting capacity by aggregating, one by one, the weak learners and adjusting model parameters to correct the prediction errors (expressed in residuals) made by previous learners. The algorithm is shown in Figure 3.

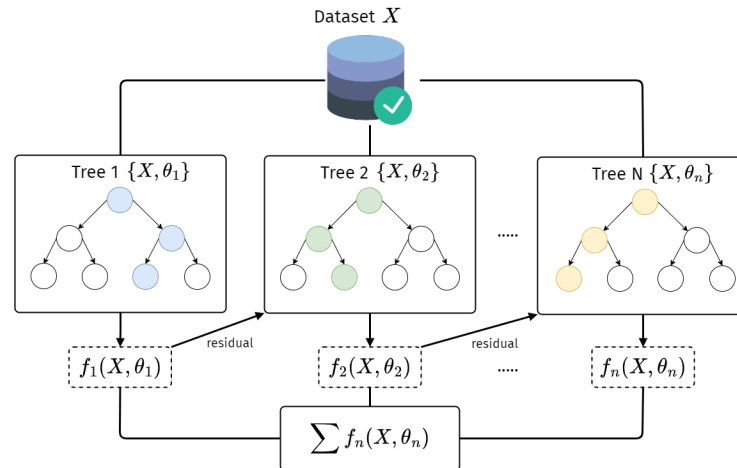


Figure 3. Simplified flowchart of the XGBoost algorithm.

Furthermore, since XGB is a tree-based algorithm, it incorporates nonlinearity, which increases its robustness in complex regression tasks. It also includes methods to avoid overfitting, such as the pruning of the different weak learners. These features made this algorithm one of the most frequently applied to the autocorrelation forecasting of renewable challenges such as energy outputs

XGBoost-based models focus on the minimization of an objective function with two different tasks: lowering the training error and regularization, as shown in Equation (1).

$$obj(x) = \sum_{i=1}^N |y - \hat{y}| + \sum_{k=1}^K \Omega(f_k) \quad (1)$$

where $|y - \hat{y}|$ represents the training error (in this case, MAE), K is the number of trees of the ensemble, T is the number of leaves, and f_k is the output value of the k -th tree. Moreover, Ω is the regularization function expressed in Equation (2).

$$\Omega(x) = \gamma \cdot T + \frac{1}{2} \cdot \lambda \cdot \sum_{i=1}^T w_i^2 \quad (2)$$

where λ and γ are coefficients directly related to the regularization term. To determine the complexity (Ω) of each tree, it is necessary to consider a new definition of the tree ($f_t(k)$), as expressed in Equation (3)

$$f_t(k) = w_{q(k)}, w \in R^T, q: R^d \rightarrow 1, 2, \dots, T \quad (3)$$

where w refers to the vector of leaf (or weight) scores of each tree and the function q assigns each data sample to its respective leaf. It is important to note that the fitting of the XGB algorithm is performed with the additive model and the forward stagewise approach. In other words, the value in $t - 1$ of the forecast variable added to the definition of the tree ($f_t(k)$) (i.e., as in the 1st Taylor approximation) is considered to be a \hat{y}_t value.

2.2.2. Support Vector Machine: Regression (SVR)

Support vector regression is widely used in renewable energy generation prediction tasks. Even with small data sets, this method can solve problems with nonlinear data. Since

SVR works with real values, unlike the classification variant (i.e., SVC), the support vectors define an epsilon— ε tolerance margin, which can contain as many data as possible [49,50].

The regression task with SVR is considered the search for the mapping between a high-dimensional input vector $x \in \mathbb{R}^d$, where d is the dimension of vector x , and an observable output $y \in \mathbb{R}$ from a specified set of independent and identically distributed samples (defined by N) all based on statistical learning theory [51,52], with a regression function $f(x)$. For this search, this technique solves the optimization problem by minimizing a risk function, as presented in Equations (4) and (5).

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) \quad (4)$$

$$\begin{aligned} \text{s.t.} \quad & y_i - (w \cdot \phi(x_i) + b) \leq \varepsilon + \xi_i \\ & (w \cdot \phi(x_i) + b) - y_i \leq \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned} \quad (5)$$

where vector and scalar coefficients (w and b , respectively) belong to the fitting parameters of the regression function $f(x)$. $\phi(x)$ denotes a nonlinear transfer function mapping model input into a higher-dimensional space, where ξ_i and ξ_i^* are the control or slack variables of the regression function, whereas C determines the balance between the regularity of $f(x)$ and the tolerance to deviations larger than ε . In other words, both epsilon and C are hyperparameters to be tuned in this method before its deployment, since their values do not depend on the optimization problem being solved.

On the other hand, the expressions presented in Equation (5) include the constraints of the optimization problem. In this case, they are all related to the support vectors and the error margin. Likewise, the slack variable values ξ and ξ_i^* are equal to or greater than zero, since they are expressed as distances between actual and predicted values.

Since SVR is an optimization problem, with its primal solution being presented in Equations (4) and (5), it also solves a dual problem linked with the values of its constraints (i.e., support vector and slack variable features), which is of great interest in this technique. Thus, it results in a different objective function. It does not consider the problem in d -dimensions of the input data x , since it only depends on the support vectors [53], as explained below.

The solution to the dual problem of SVR uses the Lagrangian method and Karush–Kuhn–Tucker (KKT) conditions, as expressed in Equations (6) and (7).

$$\max \quad -\frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(\alpha_i, \alpha_j) - \varepsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) + \sum_{i=1}^N y_i(\alpha_i - \alpha_i^*) \quad (6)$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \\ & 0 \leq \alpha_i, \alpha_i^* \leq C \end{aligned} \quad (7)$$

where α and α^* are the variables linked with the constraints (dual-problem solving as described above), which can exhibit values greater than zero and less than the adjustment hyperparameter C . $K(\alpha_i, \alpha_j)$ refers to the kernel function application (well known as *kernel trick*), which satisfies Mercer's conditions and transforms the data with nonlinearity into higher-dimensional feature space where linear separation is possible [54].

From the SVR dual problem, it is possible to extract a forecasting function under support vector and kernel dependence conditions, as shown in Equation (8).

$$f(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*)K(\alpha_i, \alpha_j) + b \quad (8)$$

2.2.3. Random Forest (RF)

Random forests are non-parametric and randomized ensemble machine learning algorithms used for both regression and classification tasks proposed by Breiman in 2001 [55]. In other words, RFs are made up of less robust algorithms or weak learners. In this case, RFs include a set of different decision trees (DT) in parallel. In other words, a bagging method. It is important to emphasize that PV power production forecasting is modeled as a regression task [30].

The main idea of the RF operation is to create and fit a set of DTs from randomly selected data samples (i.e., bagging and random subspace methodology together) and then individually obtain predictions from each tree with different node activation, in order to select the best value, as shown in Figure 4. Consequently, the negative effects of bias and variance (e.g., overfitting, among others) on the model are avoided, thus improving its performance.

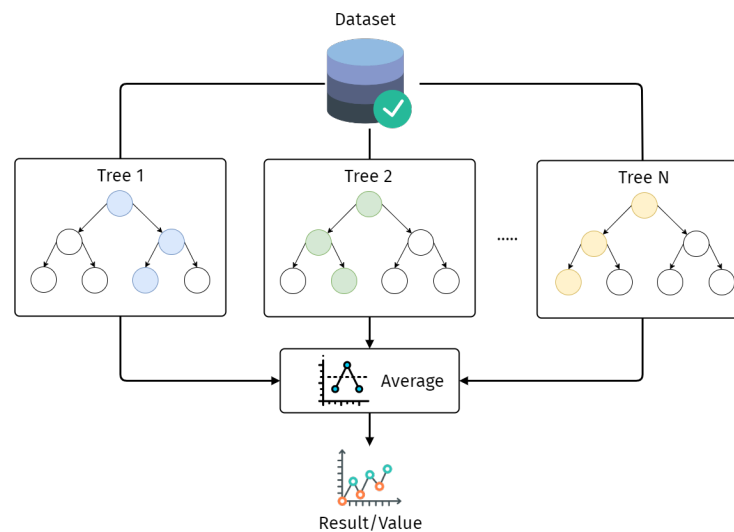


Figure 4. Simplified random forest regression flowchart.

In this way, this algorithm presents key features, such as random feature selection, bootstrap sampling, deep decision tree growth, and out-of-bag error estimation [29], which makes it suitable for the forecasting of PV system output as a supervised learning task. However, it is important to note that although it has features to avoid overfitting, as it is tree-based, this can still occur more easily than in other ML algorithms, so it is important to correctly choose parameters.

The random forest algorithm for regression tasks can be formulated as presented in Equation (9).

$$\hat{y}(t) = \frac{1}{B} \sum_{b=1}^B T_b(x; \theta_b) \quad (9)$$

where θ_b refers to the features of the b – th tree belonging to the random forest trees B , using the average and a bootstrap sample of the data with variations in its inputs (i.e., a set of random features and samples). On the other hand, the function T_b refers to the inference performed by the weak learner (in this case, a decision tree inference function, as explained in [56]), where with the absolute error, and left and right limit values, it expands its nodes prior to the application of this algorithm.

2.2.4. Multi-Layer Perceptron (MLP)

The multi-layer perceptron is a type of data-driven, forward-structured, fully connected architecture of artificial neural networks (ANNs) that assign a set of input vectors to a set of output vectors, as it occurs in a directed graph composed of several layers of nodes. It is important to note that all this mapping is performed using the complex relationships between input and output data [31]. In this case, each node is equivalent to

a unit processing neuron, which is connected with the following layers by other neurons. This model, similarly to those presented above in this paper, can be used in both regression and classification tasks.

The conventional architecture of the MLP widely used in time-series forecasting is presented in Figure 5. An MLP at least includes one input layer (presented in parallel form), one hidden layer (may be more than one considering the particular case) linked with complex pattern recognition, and one output layer. Except for the input layer (i.e., green and blue nodes), each node corresponds to a neuron with an activation function, which can be linear or nonlinear, which provides robustness and flexibility to the model.

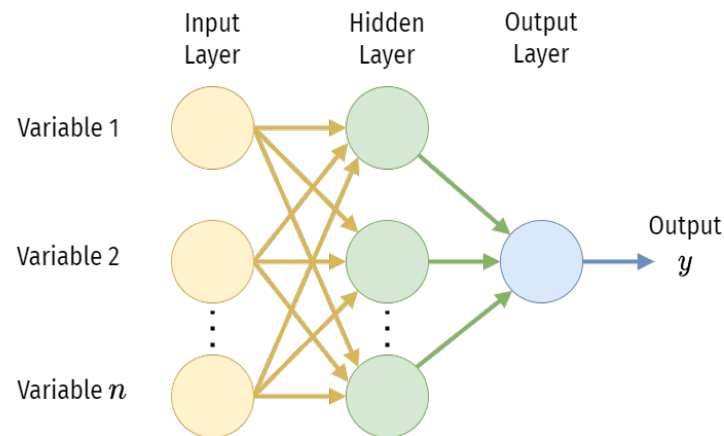


Figure 5. A schematic of an MLP neural net architecture.

In order to model the transformed time series with an MLP, a nonlinear power output function $y(t)$ is constructed where the MLP inputs correspond to the previous power values of a sequence $y(t-1)$ to $y(t-N)$ as in the universal approximation theorem applied to ANNs and explained in [57,58]. N corresponds to the number of time lags to be considered as MLP inputs (i.e. I_l , as shown in Equation (10)).

$$y(t) = \theta_o + \sum_{j=1}^{H_l} w_j \cdot f_1(\theta_j + \sum_{i=1}^{I_l} w_{ij} \cdot y(t-i)) \quad (10)$$

where at each considered time t , w_{ij} , w_j , θ_o , and θ_j correspond to the different hidden and output weights (slopes and bias, respectively) of the MLP-based model. On the other hand, H_l and I_l represent the number of hidden and input nodes, respectively. The nonlinear activation function of the hidden layers f_1 is generally sigmoid-based, such as tansig and logsig, among others. It is also possible to find other types of functions when applying these models.

The output layer (external summation with θ_o in Equation (10)) has no visible activation function. It is a modeling regression task, and the MLP, in its last layer, has identity activation functions (i.e., what arrives is equal to what leaves).

The strength of the MLP in the forecasting of variables such as PV output power comes from its flexibility in approximating any continuous function (as in the case of a regression) by directly modifying some hyperparameters (e.g., number of layers and neurons within them). However, the selection of these hyperparameters is crucial to model performance, because very large numbers of layers and neurons tend to require high training performance and high memorization of the data (i.e., not generalizing with new data), which may cause overfitting.

However, a very simple MLP (i.e., few layers and neurons) tends to have poor generalization, causing the opposite effect. Therefore, it is important to carry out strategies to properly define these elements. In this particular case, this process is defined in Section 3.2 of this paper.

2.2.5. LSTM-Based Models

An LSTM model (i.e., Vanilla LSTM and Stacked LSTM) is an RNN-based structure that includes cells with hidden (one layer in case of Vanilla) and single output layers, as shown in Figure 6. This structure is used to make predictions using sequential data (e.g., NLP, and univariate and multivariate time-series regressions, among other areas) [59]. Therefore, all LSTM theoretical background applies to these types of models. Thus, LSTM models are time-dependent recurrent nets (similar data flows with RNNs but with different operations) developed by Hochreiter et al. [60] for the purpose of learning from large data dependencies (well known as long-term dependencies) in one or more dimensions.

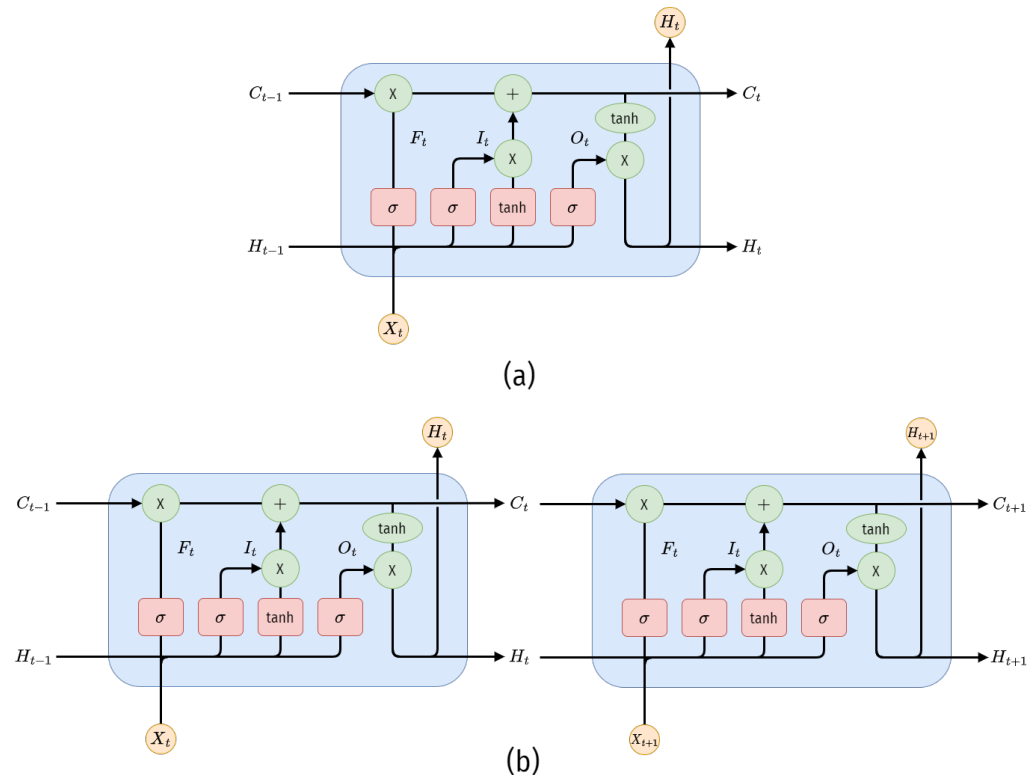


Figure 6. Schematics of (a) Vanilla and (b) Stacked LSTM architectures.

These structures, unlike other ANN-based models, involve mechanisms such as gates (i.e., input I_t , forget F_t , and output gates O_t) and internal memory units, which allow them to select, categorize, update, and decide to keep or forget the large number of data provided in a sequence. This overcomes common ANN and even RNN training-related issues such as vanishing and exploding gradients [9,61]. The LSTM cell operation can be formulated as presented in Equations (11) to (15).

$$F_t = \sigma(W_f \cdot X_t + W_{hf} \cdot H_{t-1} + b_f) \quad (11)$$

$$I_t = \sigma(W_i \cdot X_t + W_{hi} \cdot H_{t-1} + b_i) \quad (12)$$

$$O_t = \sigma(W_o \cdot X_t + W_{ho} \cdot H_{t-1} + b_o) \quad (13)$$

$$C_t = F_t \cdot C_{t-1} + I_t \cdot \tanh(W_c \cdot X_t + W_{hc} \cdot H_{t-1} + b_c) \quad (14)$$

$$H_t = O_t \cdot \tanh(C_t) \quad (15)$$

where F_t , I_t , and O_t are forget, input, and output gate values at time t , respectively, which are used to determine how many data (presented in serial form) are retained between each

gate (i.e., 1.0: no data retained; 0.0: all data retained). C_t and C_{t-1} are the current and previous states of the cell outputs (between -1.0 and 1.0) at time t and $t-1$, respectively, with sigmoid (σ) and hyperbolic tangent (\tanh) as the activation functions.

The weight matrices W_i , W_f , W_o , W_c , W_{hi} , W_{hf} , W_{ho} , and W_{hc} correspond to the input, forget, and output gates in the LSTM model. The biases b_i , b_f , b_o , and b_c are associated with these weight matrices. The hidden states H_t and H_{t-1} represent the current (t) and previous ($t-1$) states of the model, respectively.

2.2.6. ConvLSTM

The ConvLSTM structure is a variant of LSTM. Therefore, as an RNN-based structure, it uses different elements (gates and memory units, among others) to extract time-dependent or sequence-based features as energy-related forecasting tasks. This structure combines temporal and spatial features offered by the processing of the convolution layers, so that more complex relationships among the data can be identified. Convolution appears within the operations required for this structure, which presents strong similarities with LSTM, as shown in Figure 7.

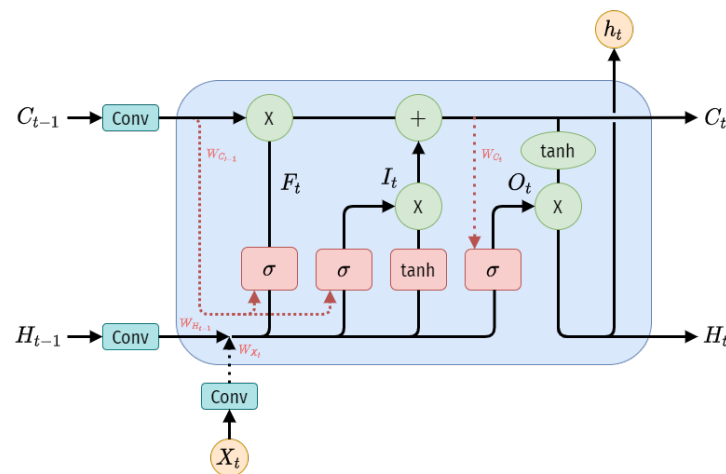


Figure 7. A schematic of ConvLSTM cell architecture.

The operation of this structure is largely based on that of the previously mentioned LSTM, with a slight modification of its internal structure. However, since the structures are 1D, the models are unable to capture complex data relationships. Therefore, many cell input operations are converted to convolutions (in this case, 1D convolutions; hence, these layers are referred to as ConvLSTM1D), where current and previous output cell values are included in the different gates. Thus, the ConvLSTM operation (whether in the 1D, 2D, or 3D version, as it only changes the model inputs in dimensions+1) can be formulated as shown in Equations (16) to (20), where “ $*$ ” denotes a convolution and “ \cdot ” refers to the element-wise product:

$$F_t = \text{logsig}(W_f * X_t + W_{hf} * H_{t-1} + W_{cf} \cdot C_{t-1} + b_f) \quad (16)$$

$$I_t = \text{logsig}(W_i * X_t + W_{hi} * H_{t-1} + W_{ci} \cdot C_{t-1} + b_i) \quad (17)$$

$$O_t = \text{logsig}(W_o * X_t + W_{ho} * H_{t-1} + W_{co} \cdot C_{t-1} + b_o) \quad (18)$$

$$C_t = F_t \cdot C_{t-1} + I_t \cdot \tanh(W_c * X_t + W_{hc} * H_{t-1} + b_c) \quad (19)$$

$$H_t = O_t \cdot \tanh(C_t) \quad (20)$$

where W_i , W_f , W_o , W_c , W_{hi} , W_{hf} , W_{ho} , W_{hc} , W_{ci} , W_{cf} , and W_{co} are the weight matrices (or parameters to be estimated) related to the different gates of the Conv-LSTM memory cell, and F_t , I_t , and O_t are the forget, input, and output gate function values at time t .

Constants b_i , b_f , b_o , and b_c correspond to the biases linked with the weight matrices described above. The variables C_t , C_{t-1} , H_t , and H_{t-1} represent the current and previous cell output and hidden state values of the memory cell at time t and $t - 1$, respectively. Finally, as in LSTM, sigmoid-based functions such as $\text{logsig}(\sigma)$ and tanh are the activation functions of each gate, considering its memory selection features. It is important to highlight that the LSTM and ConvLSTM equations show significant similarities. However, at an operational level, ConvLSTM has a higher processing rate.

2.3. Assessment of Forecasting Accuracy

The proposed ML and DL algorithms were evaluated using the following forecast throughput metrics: root mean square error (RMSE) [62,63], mean absolute error (MAE) [64], R-squared (R^2) [65,66], and mean absolute percentage error (MAPE) [29,52]. These performance metrics are expressed in Equations (21) to (24).

$$R^2 = 1 - \frac{\sum(\hat{y} - y)^2}{\sum(y - \bar{y})^2} \quad (21)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (22)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (23)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{y_i} * 100\% \quad (24)$$

where y stands for the actual value (or real), \hat{y} and \bar{y} correspond to the data forecast by the model and its mean, respectively. The R-squared metric is used to quantify the correlation between forecast (by the proposed model) and actual data. The ideal correlation corresponds to the unit value, so an R-squared value closer to 1.0 provides an accurate forecast, which is an insightful analysis performance metric.

Likewise, mean absolute percentage error (MAPE) complements the analysis that the R^2 coefficient can perform, considering its well-known overfitting trend, which can lead to bias in its analysis. Similar to R^2 , MAPE is an easy-to-explain and benchmark metric, and it does not depend on the scale of the variable being measured, making it desirable in this type of analysis. However, it is important to highlight that MAPE calculation tends to infinity (especially after 6 p.m., when actual PV power values tend to zero), so it requires some modifications to use it.

Moreover, the other performance metrics used in this study (i.e., RMSE and MAE) provide different model error dimensions (both absolute, with different risk approaches) related to PV power output prediction to provide a better and more comprehensive analysis.

3. Results and Discussion

To assess and benchmark the performance of the proposed methodology with different machine and deep learning forecasting approaches, we used data of power output from a small-scale and distributed PV resource. In this case, the data correspond to the measurements of a smart meter connected to residential PV modules located in Valle del Cauca, Colombia. These data were collected with fifteen-minute resolution between February and July 2016 (i.e., 16,512 data samples), as shown in Figure 8. It is important to mention that PV power production forecasting delivered by the proposed models is also given in fifteen-minute intervals.

All simulations and data processing were completed using a computer (PC) running Windows® with an Intel® Core I5+ 10300H processor @2.5 GHz with 16.00 GB RAM, using

Google[®] Colab in Scikit-learn 0.24 [67] and Keras Python API [68,69] for data processing, fitting (or training), and benchmarking.

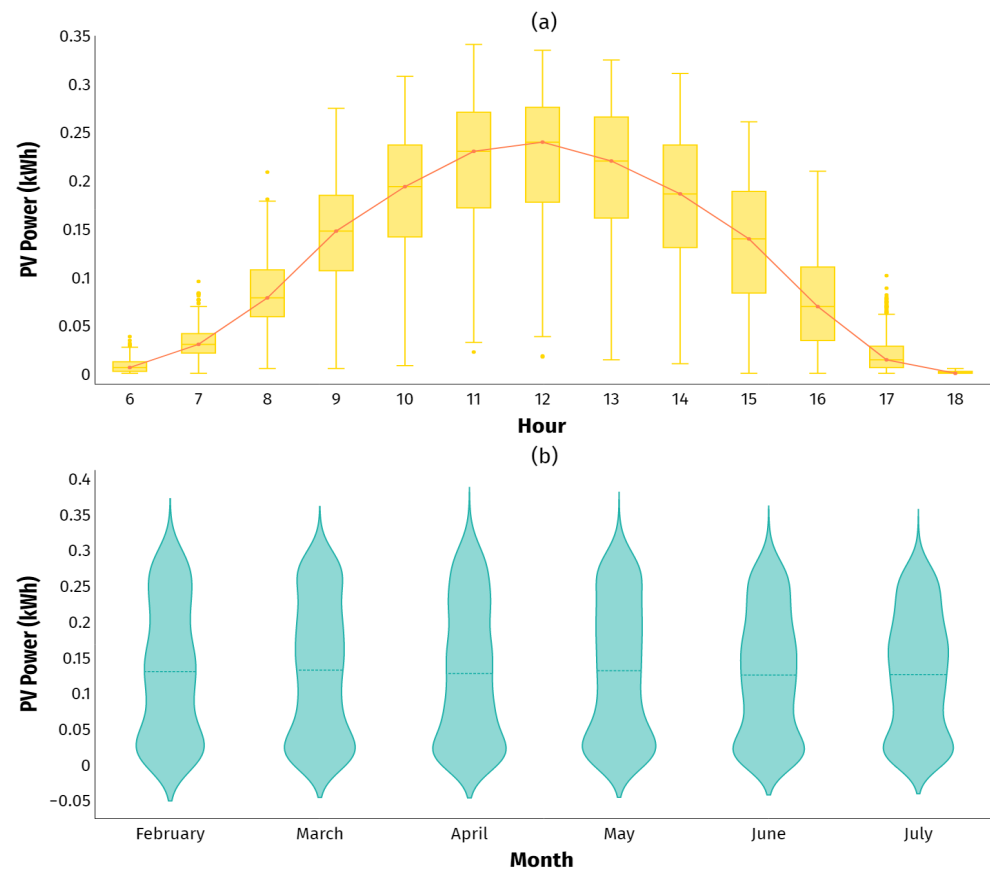


Figure 8. Power output data daily (a) and monthly (b) seasonality trends of proposed PV system production.

3.1. Exploratory Data Analysis

The overall statistics of the daily and monthly PV power output trends of the proposed system in the study period are illustrated in Figure 8. Daily seasonality is evident, with PV production increasing from 6 a.m. to 12–1 p.m. and decreasing from this point to 6 p.m., as expected in this type of system and shown in Figure 8a. Likewise, the PV system location and the capacity of the PV system (i.e., constant sunlight all year round due to the location of this zone close to the equatorial line, with rainy seasons in April and September) reflect that the gap between the minimum and maximum PV power production months (April and May, respectively, which are months linked to the rainy season in this area) was negligible. The average production value was stable in all months under review, as shown in the violin plot in Figure 8b.

On the other hand, it can also be seen that in the months of lower overall PV production (February and April), the peaks of PV production occurred. The same condition occurred in the hourly distribution of photovoltaic production, where in the hours of maximum PV production (i.e., hours 12, 13, and 14, as shown in the daily PV power boxplot), values close to zero were also present. This fact indicates high instantaneous variability of this time series and the challenge that the forecasting of this data type represents.

To establish the time-series lags needed for dataset conversion, an autocorrelation function plot (ACF) was used, as shown in Figure 9. The seasonal patterns could be confirmed, since the autocorrelations were larger for lags at multiples of the seasonal frequency than for other time lags (i.e., every 60 time lags, it reached its highest point of correlation, which coincided with the number of PV system output data obtained in one

day at 15-minute intervals in this case). This seasonality was also confirmed by applying an augmented Dickey–Fuller test (ADF), with which the null hypothesis H_0 of non-stationarity was rejected under confidence intervals of 95 and 99%. Therefore, this means that the data used showed high seasonality during the analysis period.

On the other hand, by using the confidence band in the ACF plot (gray band in Figure 9), it was possible to define the limit of lags to convert this time series into a supervised learning dataset. This was achieved by identifying this band in its upper zone just before cutting for the first time, which in this case required 60 time lags (whereas the nighttime periods between 8 p.m. and 4:45 a.m. were removed). It is important to mention that the choice of this number was based on a search for forecasting models with low computational effort.

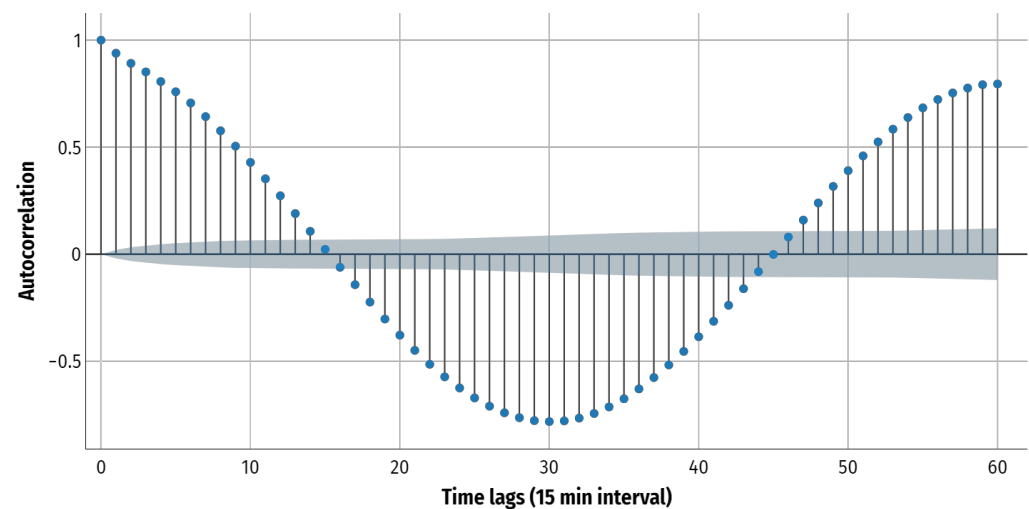


Figure 9. Autocorrelation function plot (ACF) of proposed power output data.

3.2. Hyperparameter Tuning

In order to identify and select a proper hyperparameter set, cross-validation randomized and genetic algorithm-based searches were used for tuning the proposed ML-based models. Let $P_1 \in \{P_{11}, P_{12}, P_{13}, \dots, P_{1m}\}$, $P_2 \in \{P_{21}, P_{22}, P_{23}, \dots, P_{2n}\}$, and $P_q \in \{P_{q1}, P_{q2}, P_{q3}, \dots, P_{qr}\}$, where m , n , and r refer to different hyperparameter set values to be considered and q belongs to the number of hyperparameters that each proposed model needs. For different random combinations $\{P_1, P_2, \dots, P_q\}$, the mean absolute error (since it is less outlier sensitive) was used to establish the best hyperparameter set for each model, as presented in Table 2.

Table 2. ML-related regressor model hyperparameters.

ML Technique	Hyperparameter	Value
SVR	C	0.12
	ϵ	0.01
	kernel	"rbf"
XGB	learning rate	0.15
	n_estimators	600
	max_depth	6
RF	n_estimators	600
	max_depth	5

On the other hand, there is no systematic and deterministic set of rules to decide the value of the hyperparameters of the proposed DL-related models, since they are based

on the ANN operation. Therefore, this study relied on criteria, such as the pyramidal geometric rule, widely used in different forecasting-related tasks [70] and presented in Equations (25) to (26), along with the development of an extensive set of MAE-oriented experiments used to tune them. The process outcomes are presented in Table 3.

$$R = \sqrt[L+1]{\frac{N_{input}}{N_{output}}} \quad (25)$$

$$N_L = \begin{cases} \sqrt{N_{output} * N_{input}}, & 1 \leq L < 2 \\ N_{output} \cdot R^L, & L \geq 2 \end{cases} \quad (26)$$

where L refers to the number of proposed hidden layers, N_{input} and N_{output} correspond to the number of inputs and outputs of the ANN model, and N_L is the number of nodes (or neurons) required for that layer.

Table 3 shows high coincidence between the autocorrelation function in the time series and the parameters used in the construction of the neuron-based architectures. In fact, with these parameters, the proposed structures delivered the best performance according to the available data. This fact also validates the pyramidal geometric rule as a good selection criterion for this case. Likewise, the difference in processing required by a conventional MLP to try to match more complex structures such as the proposed LSTM was also evident.

Table 3. DL-related regressor model hyperparameters.

DL Technique	Hyperparameter	Value
MLP	Hidden layers	3 (dense)
	Neurons	(60, 60, 30)
	Learning rate (α)	0.001
	Hidden layer activation function	(‘relu’, ‘relu’, ‘relu’)
Vanilla LSTM	Hidden layers	1 (LSTM cell)
	Units	60
	Learning rate (α)	0.001
	Activation function	‘tanh’
Stacked LSTM	Hidden layers	3 (LSTM cell)
	Units	(60, 60, 30)
	Learning rate (α)	0.001
	Activation function	(‘tanh’, ‘tanh’, ‘tanh’)
ConvLSTM	Hidden layers	3 (ConvLSTM1D cell)
	Units	(60, 60, 30)
	Learning rate (α)	0.001
	Activation function	(‘tanh’, ‘tanh’, ‘tanh’)

All DL-based approach models use Adam as optimizer.

It is important to highlight that, in the case of ML-based methods, MAE-based functions were used for model training tasks. In the case of DL-based models, the activation function of the output is represented by the ReLU layers based on the proposed time-series features (values greater than or equal to zero). Similarly, this table also evidences a sensitivity study of LSTM-based models for this task, since there were changes in both depth (Vanilla to Stacked LSTM) and processing (LSTM to ConvLSTM).

3.3. Model Performance Benchmarking and Analysis

To compare the forecasting throughput of PV power production, the models proposed for this study were separated according to the learning approach (i.e., machine or deep learning), to determine the techniques best suited to this task. Figures 10 and 11 present test data linked to PV power production against the forecasting results yielded by each one

of the classical (ML) and modern (DL) approaches, respectively, under low-data-availability conditions. The results correspond to a 2-day period with three different forecasting windows, i.e., 15 min, 30 min, and 1 h ahead, in order to perform a scalability analysis.

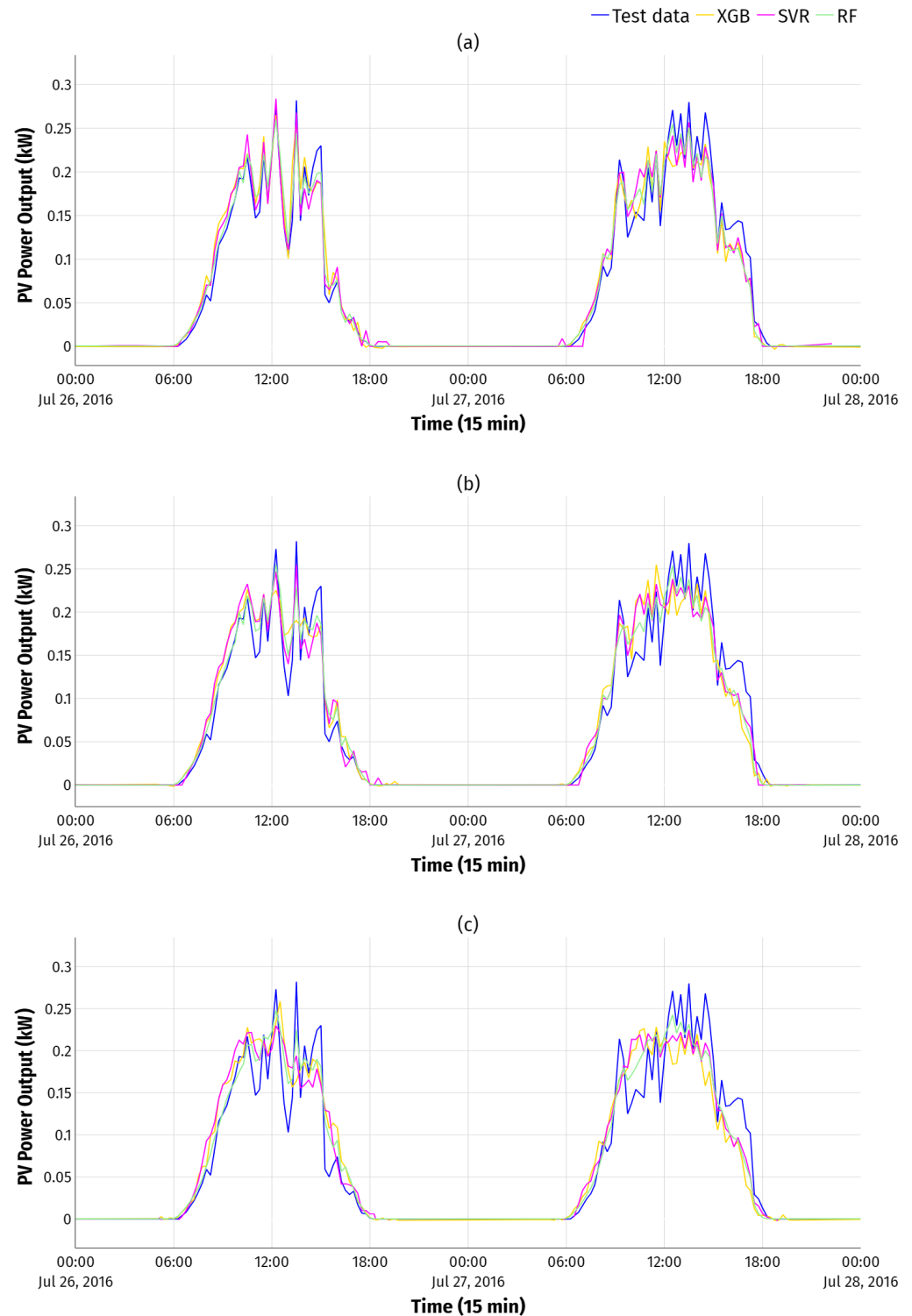


Figure 10. Comparison between PV power test data and proposed (a) 15 min ahead, (b) 30 min ahead, and (c) 1 h ahead ML-based forecasting models.

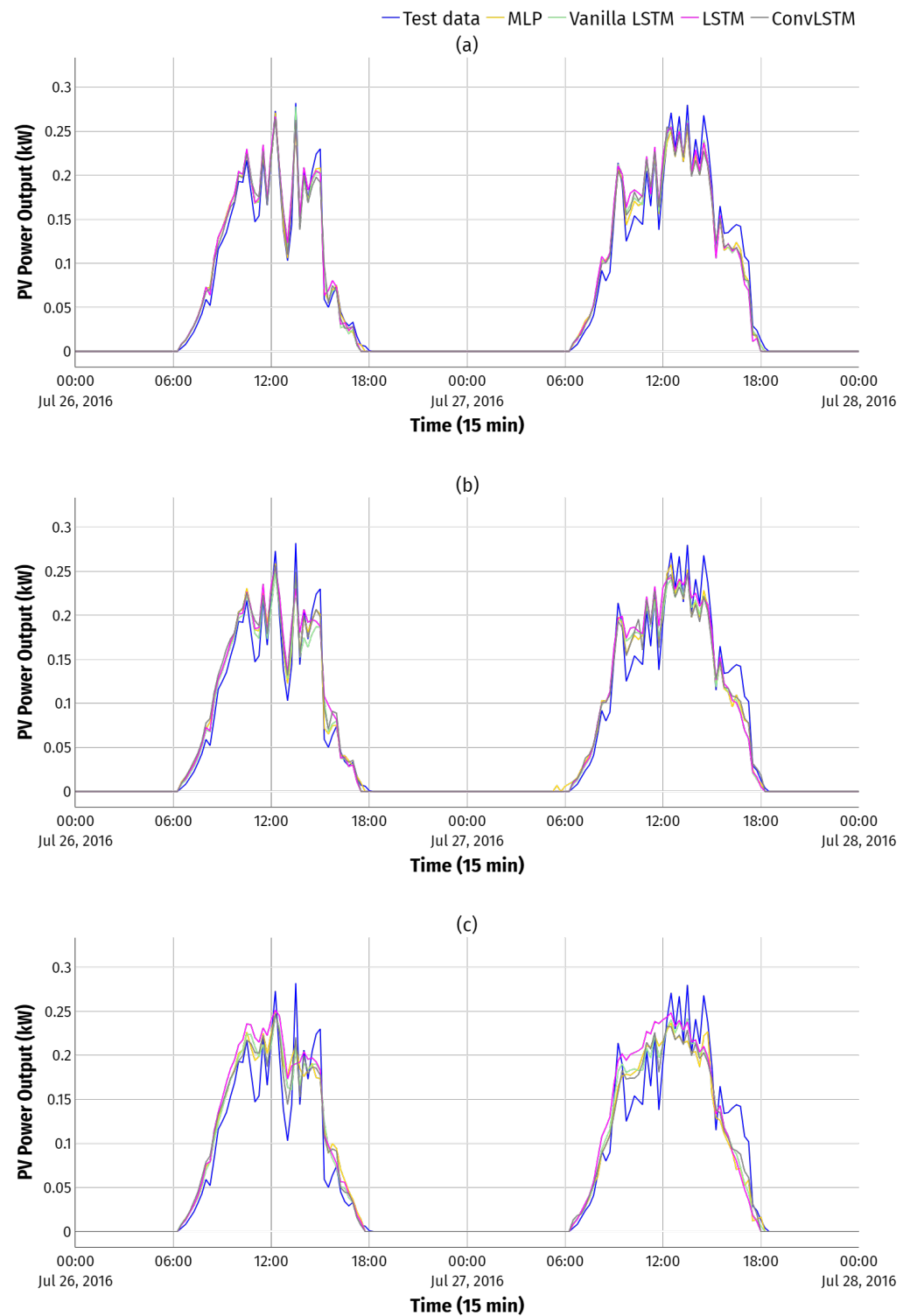


Figure 11. Comparison between PV power test data and proposed (a) 15 min ahead, (b) 30 min ahead, and (c) 1 h ahead DL-based forecasting models.

3.3.1. Technical Perspective

Firstly, with both learning approaches, a forecasting error increase (2.5% on average) was evident as the forecast horizon increased (horizon was doubled; in other words, from one to two steps and from two to four steps). However, the DL-based methods, by being able to modify their structure (output layer) for direct multi-step forecasting, significantly

reduced this effect on errors. In other words, these methods are more suitable for longer-term decision-making tasks under these circumstances and constraints.

Likewise, when comparing the different machine learning methods proposed for this study (see Figure 10), it is evident that the models based on tree ensembles (i.e., XGB and RF) were the techniques that presented the most accurate predictions. This is due to their robustness and suitability to this task. However, at maximum value points (i.e., PV power production data close to noon), the random forest model showed the best performance among all ML-based models presented in this study. This is because it better represented the daily seasonality patterns while avoiding overestimation (predicting more power than the real one) during high-variability periods for all forecast horizons. However, this model was not exempt from the decrease in performance due to the extension of the forecast horizon.

Moreover, it could be observed that the model based on the support vector regressor presented critical forecasting accuracy issues, especially at nightfall (i.e., when PV production tends to be zero), even with negative PV power values, regardless of the forecast horizon. Therefore, it was the model that presented the lowest forecasting performance in all proposed scenarios. However, when PV power instantaneous variability was present, the SVR-based model was one of the best at representing such behavior among the proposed models in the different prediction horizons. Thus, this feature could be exploited in an ensemble approach forecasting model as a sort of weak apprentice.

On the other hand, when comparing the modern learning approaches (or DL-based models; see Figure 11), smaller forecasting errors were evidenced in all proposed models by increasing the time horizon as discussed above, indicating better overall performance than the one presented by the ML approach models. In this sense, a remarkable closeness in the performance of the proposed models was observed. This was especially visible in the forecasting scenario closer to real time (15 min ahead) among those presented.

According to the results obtained with the models of this learning approach, several situations can be highlighted: First, the MLP-based model showed good performance in capturing both seasonality patterns and instantaneous variability in residential PV power production for any forecast horizon, considering that it was the least robust ANN-based model among those presented (only half as many parameters as Vanilla LSTM, which was the simplest of the proposed LSTM-based models). As in the case of the SVR-based model, MLP models can be used as inputs to more robust forecasting models, although with slightly better overall performance, as they are compelling for the development of very short-term decision-making tools.

Second, the sensitivity analysis of the LSTM-based models used in this study showed that there was no meaningful performance gap when increasing LSTM layers in their structure. Both Vanilla and Stacked LSTM had similar temporal behaviors regardless of the proposed prediction horizon. In other words, considering the temporal nature and availability of the data used in this study, a stacked model would have been less feasible given its learning requirements (computational resources and time) in this case. However, this assessment could change if other related variables (temperature, irradiance, and time-related variables, among others) or more PV production data were available, since more robustness is required when handling these data.

Finally, the ConvLSTM1D model showed consistency and accurate results, since it performed well in capturing both seasonality trends and high variability during sudden power production changes (i.e., atypical situations where PV power rapidly increases or decreases, for example, due to cloudiness). This model, unlike its 2D version (ConvLSTM2D) used in some methods for PV power prediction (see Table 1), performed adequate feature extraction according to data conditions. This implied a smaller number of parameters, thus requiring less training time and data availability, and a smaller investment in computing resources. These are expected features in modules that supply decision-making tools in the short and medium terms.

Table 4 summarizes these performance indicators for each forecasting window with the test data for all forecasting models (i.e., SVR, RF, XGB, MLP, LSTM, and ConvLSTM1D).

applied in this study. This table shows different regression metrics, such as MAE, RMSE, MAPE, and R^2 .

Table 4. Benchmark of different performance metrics applied to the proposed ML and DL models, and forecast horizons.

Forecasting Horizon	Model	RMSE (kWh)	MAE (kWh)	R^2	MAPE (%)
15 min ahead	SVR	0.0263	0.0143	0.9014	18.87
	XGB	0.0263	0.0131	0.9021	14.39
	RF	0.0251	0.0121	0.9104	14.49
	MLP	0.0262	0.0127	0.9029	14.06
	Vanilla LSTM	0.0259	0.0122	0.9050	14.57
	LSTM	0.0261	0.0123	0.9039	14.02
	ConvLSTM1D	0.0264	0.0125	0.9015	14.38
30 min ahead	SVR	0.0288	0.0159	0.8822	21.99
	XGB	0.0287	0.0145	0.8838	17.41
	RF	0.0274	0.0134	0.8937	16.91
	MLP	0.0279	0.0138	0.8901	16.30
	Vanilla LSTM	0.0285	0.0137	0.8849	16.55
	LSTM	0.0287	0.0139	0.8838	16.45
	ConvLSTM1D	0.0273	0.0135	0.8855	16.28
1 h ahead	SVR	0.0318	0.0179	0.8567	25.70
	XGB	0.0307	0.0157	0.8663	20.20
	RF	0.0296	0.0149	0.8760	19.67
	MLP	0.0304	0.0153	0.8691	17.85
	Vanilla LSTM	0.0304	0.0152	0.8693	18.54
	LSTM	0.0314	0.0155	0.8601	17.14
	ConvLSTM1D	0.0275	0.0134	0.8898	17.65

This benchmark confirmed the previous plot analysis, where the SVR had the lowest performance out of all the presented models for any forecast horizon (about 4% higher MAPE on average). Furthermore, the models based on tree ensembles (i.e., random forest and XGBoost) were the best-performing ML techniques among those presented, especially in 15 min ahead forecasting, with mean absolute error smaller than 15% (i.e., 14.39% and 14.49% for XGB and RF, respectively).

However, by increasing the forecast horizon, the prediction error in ML-based models increased by an average of 2.5% (about 0.002 kWh), as discussed above. As the prediction horizon increased, the XGB model presented more forecasting errors than RF. Thus, RF had better overall performance.

On the other hand, all DL-based models showed similar performance in any forecast horizon, where MAPEs were around 14% (0.5% gap between models with the worst and the best performance) in the 15 min ahead forecast and an average of 2.3% (0.6% gap between models with the worst and the best performance) by doubling the prediction horizon. This indicates that all proposed models represented the PV power behavior features without overfitting or suffering any noticeable negative effect on their performance. This fact is supported by the performance metrics, where there was no dominant model in the proposed prediction horizons.

Finally, the inclusion of the ConvLSTM1D model in the regression tasks of forecasting PV production showed better performance with the increase in the prediction horizon. In other words, in the longest forecasting window, it presented the best MAE and RMSE performance metrics among the proposed models. Therefore, together with RF, these were the models best suited to modeling the PV production behavior in the cases examined in this study.

3.3.2. Economical Perspective

The objective of enhancing residential-scale PV production forecasts is to reduce the unpredictability associated with this source, leading to more secure and effortless integration and management of power grids. As a result, in some energy markets, residential-scale energy producers and owners may incur in penalties when the discrepancy between predicted and actual PV power production exceeds a predetermined threshold. In fact, when PV systems produce more than scheduled (above the threshold), the mismatch is paid at lower value, whereas in the opposite case, producers must obtain the remaining energy from the market or pay the difference. Therefore, decision-making tools are critical for dealing with these risk situations.

In this sense, by considering the forecast mismatches and performance metrics presented above, it is possible to perform this analysis of indirect approaches using the performance metrics used above: MAE (risk neutral) and RMSE (risk averse) metrics. Thus, if these predictions were used in real-time applications (storage and surplus sale, among others) or where variable energy tariffs are applied, the risk-neutral and risk-averse approaches showed that the best model in this case would be the RF-based model, because it presented the lowest MAE and RMSE metrics in the closest prediction horizon (15 min ahead).

Table 5 presents the average economic losses incurred during the test data period (i.e., 18 days) by the different forecasting models in the 15 min forecast horizon. To estimate this value, we used the electricity spot price (2.82 USD/kWh) and the average COP-to-USD conversion rate (2966.92 COP = 1 USD) at the time when the PV production data were collected.

Table 5. Economic losses in 15 min forecast horizon.

Approach	XGB (USD)	RF (USD)	SVR (USD)	MLP (USD)	Vanilla (USD)	LSTM (USD)	ConvLSTM (USD)
Risk neutral	0.65	0.60	0.71	0.63	0.61	0.61	0.62
Risk averse	1.31	1.22	1.31	1.30	1.29	1.30	1.31

However, when considering a longer prediction horizon and thus focusing on other applications (e.g., scheduling, demand response, and supply–demand balance, among others), the ConvLSTM1D-based model performed better in both risk scenarios (lower MAE and RMSE metrics in 1 h ahead forecasting) under these low-data-availability conditions. In other words, there is little likelihood of PV production overestimation or underestimation. This leads to lower perceived economic income risks for owners, which is essential to building accurate decision-making tools that work with these results.

As in the previous case, Table 6 presents the average economic losses incurred during the test data period (i.e., 18 days) by the different forecasting models in the 1 h forecast horizon.

Table 6. Economic losses in 1-h forecast horizon.

Approach	XGB (USD)	RF (USD)	SVR (USD)	MLP (USD)	Vanilla (USD)	LSTM (USD)	ConvLSTM (USD)
Risk neutral	0.78	0.74	0.89	0.76	0.76	0.77	0.67
Risk averse	1.53	1.47	1.58	1.51	1.51	1.56	1.37

Importantly, the development of these losses in both forecast horizons has equally penalized over- and underestimation for practical purposes. However, according to each

energy market and how these events are penalized, this could change. This is a direction of research towards future development in this field.

Likewise, for decision-making tools, the deterministic inputs from these forecasting models do not provide complete information on the unexpected prediction scenarios. This uncertainty can lead to errors in the actions to be taken and thus lead to economic losses. Therefore, to reduce this uncertainty, these forecasting models require complementary elements that address these cases, such as electricity price data correlation or a PV power probabilistic forecast approach, which would give a broader view of these cases and guide better decision making. This also implies future development and specific analysis of this topic.

4. Conclusions

It is of great importance to improve power system operation with the rapidly growing inclusion and widespread use of distributed energy resources (DERs) to increase the reliability and resilience of these systems. To this end, the accurate forecasting of household-scale PV power in intra-day forecast horizons is essential. To address this issue, a comprehensive assessment (technical and economic perspectives) of different machine and deep learning techniques focused on time-series regression tasks is presented in this paper.

Firstly, the datasets used were limited 15 min resolution time-series data from a smart meter in a residential PV system in Valle del Cauca (Colombia). The benchmark results show that from a technical perspective and considering this data availability issue, the best overall performance was obtained by RF and LSTM-based models. However, there was no dominant model in the different forecast horizons according to the performance metrics used. In other words, considering the decision-making tool to be developed, and above all, the objective to be pursued with it and the respective time frame, the required forecasting model and the chosen performance metric can be different, as in this case (i.e., one step ahead: RF; four steps ahead: ConvLSTM1D).

On the other hand, considering the data conditions used in this study (univariate time series and limited historical data of residential PV production), including more layers of LSTM processing had little impact on model performance, as evidenced by the sensitivity analysis using both Vanilla and Stacked LSTM. Therefore, other processing elements are needed to achieve this objective. In this context, the inclusion of models such as ConvLSTM1D offers the best overall performance among the presented forecast horizons. It is an effective learning method for this task, since it achieves the benefits of ANNs, and the complex relationships and feature extraction based on 1D convolution (requiring less computational effort than 2D convolution), in addition to the temporal memory factor. In this case, this feature had a greater impact by increasing the prediction horizon. This is very advantageous in longer-term DER decision-making tool development, since these models can provide better PV power prediction under these conditions.

Furthermore, an economic analysis was conducted using the performance metrics as proxy measures of risk neutrality and risk aversion, i.e., MAE and RMSE, respectively. This was performed in order to evaluate which DL and ML models are less susceptible to lower-profit scenarios for the owners and producers of these generation assets. In very-short-term forecasting scenarios, the RF-based ML model performed better. As the prediction window increased, despite the expected overall decrease in the performance of the proposed models, the ConvLSTM1D model performed better under these conditions. Thus, these models can feed decision-making tools that focus on economic interests.

Based on the conducted study, in the future, we will work on performing new techno-economic assessments in the development of different residential-scale PV forecasting model approaches considering accuracy and uncertainty issues. In this context, more robust prediction approaches will be developed, moving from a deterministic approach to a probabilistic one for the development of scenarios that guide better decision making and identifying market opportunities for different market players and new related business

models. These approaches will be tested under different data availability conditions, with a particular focus on developing countries.

Additionally, the use of ensemble methods and hybrid models that combine statistic and deep learning techniques will be explored to improve the accuracy of the predictions. The proposed methodology will allow the development of tools that can be used to support the design of regulatory policies and incentives to promote the integration of renewable energy sources into power grids to be performed.

Author Contributions: Conceptualization, R.M.-C. and S.C.-L.; methodology, S.C.-L. and R.M.-C.; software, S.C.-L.; validation, S.C.-L. and R.M.-C.; formal analysis, S.C.-L. and D.C.; investigation, S.C.-L., R.M.-C., D.C. and G.A.; data curation, S.C.-L.; writing—original draft preparation, S.C.-L., R.M.-C. and D.C.; writing—review and editing, S.C.-L., R.M.-C., D.C. and G.A.; visualization, S.C.-L.; supervision, R.M.-C.; funding acquisition, D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the starting grant IV-TFA056 entitled “Machine learning for Smart Energy Systems” by the Research Direction at Universidad del Rosario.

Acknowledgments: The authors would like to thank the support of the Universidad Icesi.

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

References

- Restrepo-Trujillo, J.; Moreno-Chuquen, R.; Jiménez-García, F.; Chamorro, H.R. Scenario Analysis of an Electric Power System in Colombia Considering the El Niño Phenomenon and the Inclusion of Renewable Energies. *Energies* **2022**, *15*, 6690. [\[CrossRef\]](#)
- Ufa, R.; Malkova, Y.; Rudnik, V.; Andreev, M.; Borisov, V. A review on distributed generation impacts on electric power system. *Int. J. Hydrogen Energy* **2022**, *47*, 20347–20361. [\[CrossRef\]](#)
- Cantillo-Luna, S.; Moreno-Chuquen, R.; Chamorro, H.R.; Sood, V.K.; Badsha, S.; Konstantinou, C. Blockchain for Distributed Energy Resources Management and Integration. *IEEE Access* **2022**, *10*, 68598–68617. [\[CrossRef\]](#)
- Burger, S.P.; Luke, M. Business models for distributed energy resources: A review and empirical analysis. *Energy Policy* **2017**, *109*, 230–248. [\[CrossRef\]](#)
- Zaouali, K.; Rekik, R.; Bouallegue, R. Deep learning forecasting based on auto-lstm model for home solar power systems. In Proceedings of the 2018 IEEE 20th International Conference on High Performance Computing and Communications, IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data science and Systems (HPCC/SmartCity/DSS), Exeter, UK, 28–30 June 2018; pp. 235–242.
- Wang, K.; Qi, X.; Liu, H. A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network. *Appl. Energy* **2019**, *251*, 113315. [\[CrossRef\]](#)
- Akhter, M.N.; Mekhilef, S.; Mokhlis, H.; Mohamed Shah, N. Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques. *IET Renew. Power Gener.* **2019**, *13*, 1009–1023. [\[CrossRef\]](#)
- Hafiz, F.; Awal, M.; de Queiroz, A.R.; Husain, I. Real-time stochastic optimization of energy storage management using deep learning-based forecasts for residential PV applications. *IEEE Trans. Ind. Appl.* **2020**, *56*, 2216–2226. [\[CrossRef\]](#)
- Rajagukguk, R.A.; Ramadhan, R.A.; Lee, H.J. A review on deep learning models for forecasting time series data of solar irradiance and photovoltaic power. *Energies* **2020**, *13*, 6623. [\[CrossRef\]](#)
- Santhosh, M.; Venkaiah, C.; Vinod Kumar, D. Current advances and approaches in wind speed and wind power forecasting for improved renewable energy integration: A review. *Eng. Rep.* **2020**, *2*, e12178. [\[CrossRef\]](#)
- Gupta, P.; Singh, R. PV power forecasting based on data-driven models: A review. *Int. J. Sustain. Eng.* **2021**, *14*, 1733–1755. [\[CrossRef\]](#)
- Massaoudi, M.; Chihi, I.; Abu-Rub, H.; Refaat, S.S.; Oueslati, F.S. Convergence of photovoltaic power forecasting and deep learning: State-of-art review. *IEEE Access* **2021**, *9*, 136593–136615. [\[CrossRef\]](#)
- Tina, G.M.; Ventura, C.; Ferlito, S.; De Vito, S. A state-of-art-review on machine-learning based methods for PV. *Appl. Sci.* **2021**, *11*, 7550. [\[CrossRef\]](#)
- Costa, R.L.D.C. Convolutional-LSTM networks and generalization in forecasting of household photovoltaic generation. *Eng. Appl. Artif. Intell.* **2022**, *116*, 105458. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Gaviria, J.F.; Narváez, G.; Guillen, C.; Giraldo, L.F.; Bressan, M. Machine learning in photovoltaic systems: A review. *Renew. Energy* **2022**, *196*, 298–318. [\[CrossRef\]](#)
- Zhao, E.; Sun, S.; Wang, S. New developments in wind energy forecasting with artificial intelligence and big data: A scientometric insight. *Data Sci. Manag.* **2022**, *5*, 84–95. [\[CrossRef\]](#)

18. Markovics, D.; Mayer, M.J. Comparison of machine learning methods for photovoltaic power forecasting based on numerical weather prediction. *Renew. Sustain. Energy Rev.* **2022**, *161*, 112364. [\[CrossRef\]](#)
19. Shabbir, N.; Kütt, L.; Raja, H.A.; Jawad, M.; Allik, A.; Husev, O. Techno-economic analysis and energy forecasting study of domestic and commercial photovoltaic system installations in Estonia. *Energy* **2022**, *253*, 124156. [\[CrossRef\]](#)
20. Luo, Z.; Peng, J.; Tan, Y.; Yin, R.; Zou, B.; Hu, M.; Yan, J. A novel forecast-based operation strategy for residential PV-battery-flexible loads systems considering the flexibility of battery and loads. *Energy Convers. Manag.* **2023**, *278*, 116705. [\[CrossRef\]](#)
21. Phinikarides, A.; Makrides, G.; Kindyni, N.; Kyprianou, A.; Georghiou, G.E. ARIMA modeling of the performance of different photovoltaic technologies. In Proceedings of the 2013 IEEE 39th Photovoltaic Specialists Conference (PVSC), Tampa, FL, USA, 16–21 June 2013; pp. 797–801.
22. Dong, Z.; Yang, D.; Reindl, T.; Walsh, W.M. Short-term solar irradiance forecasting using exponential smoothing state space model. *Energy* **2013**, *55*, 1104–1113. [\[CrossRef\]](#)
23. Li, Y.; Su, Y.; Shu, L. An ARMAX model for forecasting the power output of a grid connected photovoltaic system. *Renew. Energy* **2014**, *66*, 78–89. [\[CrossRef\]](#)
24. Raza, M.Q.; Nadarajah, M.; Ekanayake, C. On recent advances in PV output power forecast. *Sol. Energy* **2016**, *136*, 125–144. [\[CrossRef\]](#)
25. Sobri, S.; Koohi-Kamali, S.; Rahim, N.A. Solar photovoltaic generation forecasting methods: A review. *Energy Convers. Manag.* **2018**, *156*, 459–497. [\[CrossRef\]](#)
26. Atique, S.; Noureen, S.; Roy, V.; Subburaj, V.; Bayne, S.; Macfie, J. Forecasting of total daily solar energy generation using ARIMA: A case study. In Proceedings of the 2019 IEEE 9th annual computing and communication workshop and conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; pp. 114–119.
27. Assouline, D.; Mohajeri, N.; Scartezini, J.L. Quantifying rooftop photovoltaic solar energy potential: A machine learning approach. *Sol. Energy* **2017**, *141*, 278–296. [\[CrossRef\]](#)
28. Preda, S.; Oprea, S.V.; Bâra, A.; Belciu, A. PV forecasting using support vector machine learning in a big data analytics context. *Symmetry* **2018**, *10*, 748. [\[CrossRef\]](#)
29. Ahmad, M.W.; Mourshed, M.; Rezgui, Y. Tree-based ensemble methods for predicting PV power generation and their comparison with support vector regression. *Energy* **2018**, *164*, 465–474. [\[CrossRef\]](#)
30. Huertas Tato, J.; Centeno Brito, M. Using smart persistence and random forests to predict photovoltaic energy production. *Energies* **2018**, *12*, 100. [\[CrossRef\]](#)
31. Zhu, R.; Guo, W.; Gong, X. Short-term photovoltaic power output prediction based on k-fold cross-validation and an ensemble model. *Energies* **2019**, *12*, 1220. [\[CrossRef\]](#)
32. Munawar, U.; Wang, Z. A framework of using machine learning approaches for short-term solar power forecasting. *J. Electr. Eng. Technol.* **2020**, *15*, 561–569. [\[CrossRef\]](#)
33. Phan, Q.T.; Wu, Y.K.; Phan, Q.D. Short-term Solar Power Forecasting Using XGBoost with Numerical Weather Prediction. In Proceedings of the 2021 IEEE International Future Energy Electronics Conference (IFEEEC), Taipei, Taiwan, 16–19 November 2021. [\[CrossRef\]](#)
34. Wang, Y.; Liao, W.; Chang, Y. Gated recurrent unit network-based short-term photovoltaic forecasting. *Energies* **2018**, *11*, 2163. [\[CrossRef\]](#)
35. Lee, D.; Jeong, J.; Yoon, S.H.; Chae, Y.T. Improvement of short-term BIPV power predictions using feature engineering and a recurrent neural network. *Energies* **2019**, *12*, 3247. [\[CrossRef\]](#)
36. Hossain, M.S.; Mahmood, H. Short-term photovoltaic power forecasting using an LSTM neural network and synthetic weather forecast. *IEEE Access* **2020**, *8*, 172524–172533. [\[CrossRef\]](#)
37. Ahn, H.K.; Park, N. Deep RNN-based photovoltaic power short-term forecast using power IoT sensors. *Energies* **2021**, *14*, 436. [\[CrossRef\]](#)
38. Khan, W.; Walker, S.; Zeiler, W. Improved solar photovoltaic energy generation forecast using deep learning-based ensemble stacking approach. *Energy* **2022**, *240*, 122812. [\[CrossRef\]](#)
39. Ghimire, S.; Deo, R.C.; Raj, N.; Mi, J. Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms. *Appl. Energy* **2019**, *253*, 113541. [\[CrossRef\]](#)
40. Suresh, V.; Janik, P.; Rezmer, J.; Leonowicz, Z. Forecasting solar PV output using convolutional neural networks with a sliding window algorithm. *Energies* **2020**, *13*, 723. [\[CrossRef\]](#)
41. Tovar, M.; Robles, M.; Rashid, F. PV power prediction, using CNN-LSTM hybrid neural network model. Case of study: Temixco-Morelos, México. *Energies* **2020**, *13*, 6512. [\[CrossRef\]](#)
42. Agga, A.; Abbou, A.; Labbadi, M.; El Houm, Y.; Ali, I.H.O. CNN-LSTM: An efficient hybrid deep learning architecture for predicting short-term photovoltaic power production. *Electr. Power Syst. Res.* **2022**, *208*, 107908. [\[CrossRef\]](#)
43. Zhou, H.; Zhang, Y.; Yang, L.; Liu, Q.; Yan, K.; Du, Y. Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism. *IEEE Access* **2019**, *7*, 78063–78074. [\[CrossRef\]](#)
44. Wang, F.; Zhang, Z.; Liu, C.; Yu, Y.; Pang, S.; Duić, N.; Shafie-Khah, M.; Catalao, J.P. Generative adversarial networks and convolutional neural networks based weather classification model for day ahead short-term photovoltaic power forecasting. *Energy Convers. Manag.* **2019**, *181*, 443–462. [\[CrossRef\]](#)

45. Perera, M.; De Hoog, J.; Bandara, K.; Halgamuge, S. Multi-resolution, multi-horizon distributed solar PV power forecasting with forecast combinations. *Expert Syst. Appl.* **2022**, *205*, 117690. [\[CrossRef\]](#)
46. Grzebyk, D.; Alcañiz, A.; Donker, J.C.; Zeman, M.; Ziar, H.; Isabella, O. Individual yield nowcasting for residential PV systems. *Sol. Energy* **2023**, *251*, 325–336. [\[CrossRef\]](#)
47. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
48. Nielsen, D. Tree Boosting with Xgboost-Why Does Xgboost Win “Every” Machine Learning Competition? Master’s Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2016.
49. Xu, S.; An, X.; Qiao, X.; Zhu, L.; Li, L. Multi-output least-squares support vector regression machines. *Pattern Recognit. Lett.* **2013**, *34*, 1078–1084. [\[CrossRef\]](#)
50. Ahmad, W.; Ayub, N.; Ali, T.; Irfan, M.; Awais, M.; Shiraz, M.; Glowacz, A. Towards Short Term Electricity Load Forecasting Using Improved Support Vector Machine and Extreme Learning Machine. *Energies* **2020**, *13*, 2907. [\[CrossRef\]](#)
51. Das, U.K.; Tey, K.S.; Seyedmahmoudian, M.; Idna Idris, M.Y.; Mekhilef, S.; Horan, B.; Stojcevski, A. SVR-based model to forecast PV power generation under different weather conditions. *Energies* **2017**, *10*, 876. [\[CrossRef\]](#)
52. Cantillo-Luna, S.; Moreno-Chuquen, R.; Chamorro, H.R.; Riquelme-Dominguez, J.M.; Gonzalez-Longatt, F. Locational Marginal Price Forecasting Using SVR-Based Multi-Output Regression in Electricity Markets. *Energies* **2022**, *15*, 293. [\[CrossRef\]](#)
53. Hong, W.C.; Fan, G.F. Hybrid Empirical Mode Decomposition with Support Vector Regression Model for Short Term Load Forecasting. *Energies* **2019**, *12*, 1093. [\[CrossRef\]](#)
54. Fu, T.; Zhang, S.; Wang, C. Application and research for electricity price forecasting system based on multi-objective optimization and sub-models selection strategy. *Soft Comput.* **2020**, *24*, 15611–15637. [\[CrossRef\]](#)
55. Majidpour, M.; Nazarpouya, H.; Chu, P.; Pota, H.R.; Gadh, R. Fast univariate time series prediction of solar power for real-time control of energy storage system. *Forecasting* **2018**, *1*, 107–120. [\[CrossRef\]](#)
56. Wang, J.; Li, P.; Ran, R.; Che, Y.; Zhou, Y. A short-term photovoltaic power prediction model based on the gradient boost decision tree. *Appl. Sci.* **2018**, *8*, 689. [\[CrossRef\]](#)
57. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [\[CrossRef\]](#)
58. Haykin, S. *Neural Networks and Learning Machines*, 3/E; Pearson Education: Noida, India, 2009.
59. Wang, F.; Xuan, Z.; Zhen, Z.; Li, K.; Wang, T.; Shi, M. A day-ahead PV power forecasting method based on LSTM-RNN model and time correlation modification under partial daily pattern prediction framework. *Energy Convers. Manag.* **2020**, *212*, 112766. [\[CrossRef\]](#)
60. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#)
61. Liu, Y.; Guan, L.; Hou, C.; Han, H.; Liu, Z.; Sun, Y.; Zheng, M. Wind power short-term prediction based on LSTM and discrete wavelet transform. *Appl. Sci.* **2019**, *9*, 1108. [\[CrossRef\]](#)
62. Pourdayaei, A.; Mokhlis, H.; Illias, H.A.; Kaboli, S.H.A.; Ahmad, S.; Ang, S.P. Hybrid ANN and Artificial Cooperative Search Algorithm to Forecast Short-Term Electricity Price in De-Regulated Electricity Market. *IEEE Access* **2019**, *7*, 125369–125386. [\[CrossRef\]](#)
63. Azam, M.F.; Younis, S. Multi-Horizon Electricity Load and Price Forecasting using an Interpretable Multi-Head Self-Attention and EEMD-Based Framework. *IEEE Access* **2021**, *9*, 85918–85932. [\[CrossRef\]](#)
64. Ağbulut, Ü.; Gürel, A.E.; Ergün, A.; Ceylan, İ. Performance assessment of a V-Trough photovoltaic system and prediction of power output with different machine learning algorithms. *J. Clean. Prod.* **2020**, *268*, 122269. [\[CrossRef\]](#)
65. Jiang, L.; Hu, G. A Review on Short-Term Electricity Price Forecasting Techniques for Energy Markets. In Proceedings of the 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 18–21 November 2018. [\[CrossRef\]](#)
66. Hong, Y.Y.; Taylar, J.V.; Fajardo, A.C. Locational marginal price forecasting in a day-ahead power market using spatiotemporal deep learning network. *Sustain. Energy Grids Netw.* **2020**, *24*, 100406. [\[CrossRef\]](#)
67. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
68. Chollet, F. Keras. 2015. Available online: <https://github.com/fchollet/keras> (accessed on 25 January 2023).
69. Gulli, A.; Pal, S. *Deep Learning with Keras*; Packt Publishing Ltd.: Birmingham, UK, 2017.
70. Rachmatullah, M.I.C.; Santoso, J.; Surendro, K. A novel approach in determining neural networks architecture to classify data with large number of attributes. *IEEE Access* **2020**, *8*, 204728–204743. [\[CrossRef\]](#)

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.