Short-term solar irradiance forecasting in streaming with deep learning

Pedro Lara-Benítez [a,*], Manuel Carranza-García [b], José María Luna-Romera [b], José C. Riquelme [b]

[a] Process Engineering team in Global Risk Analytics, Bank of America, EC1A 1HQ, United Kingdom
[b] Division of Computer Science, University of Sevilla, ES-41012, Spain

ARTICLE INFO

Article history:
Received 14 May 2022
Revised 31 December 2022
Accepted 7 May 2023
Available online 11 May 2023

Keywords:
Solar irradiance
Deep learning
Data stream
Short-term forecasting
Time series
Online learning

ABSTRACT

Solar energy is one of the most common and promising sources of renewable energy. In photovoltaic (PV) systems, operators can benefit from future solar irradiance predictions for efficient load balancing and grid stability. Therefore, short-term solar irradiance forecasting plays a crucial role in the transition to renewable energy. Modern PV grids collect large volumes of data that provide valuable information for forecasting models. Although the nature of these data presents an ideal setting for online learning methodologies, research to date has mainly focused on offline approaches. Hence, this work proposes a novel data streaming method for real-time solar irradiance forecasting on days with variable weather conditions and cloud coverage. Our method operates under an asynchronous dual-pipeline framework using deep learning models. For the experimental study, two datasets from a Canadian PV solar plant have been simulated as streams at different data frequencies. The experiments involve an exhaustive parameter grid search to evaluate four state-of-the-art deep learning architectures: multilayer perceptron (MLP), long-short term memory network (LSTM), convolutional network (CNN), and Transformer network. The obtained results demonstrate the suitability of deep learning models for this problem. In particular, MLP and CNN achieved the best accuracy, with a high capacity to adapt to the evolving data stream.

1. Introduction

The field of renewable energies is attracting growing political and research attention due to its potential to address climate change and the energy crisis. Solar energy is one of the most sustainable energy sources used worldwide and is considered a powerful alternative to fossil fuel energy [1]. However, the volatility of photovoltaic (PV) affects energy production stability. A reliable and accurate short-term forecast of solar radiation is fundamental for the efficient and correct operational management of solar plants. Some studies have demonstrated that accurate predictions of solar irradiance could imply a substantial cost reduction [2].

The field of solar irradiance forecasting has received increasing interest during the past few decades. The different approaches found in the literature can be classified into three main subgroups: deterministic, statistical or machine learning, and hybrid methods [3]. Deterministic models are empirically based and use individual equations derived from physical phenomena specific to the electrical specification of PV devices. These models are presented as reliable and explainable solutions [4]. They became standard models in the industry during the late 1980s and 1990s. This approach is included in commonly used commercial software such as PVSyst, PV SunVisor, Energy-10, DOE-2, or the System Advisor Model (SAM). These tools provide support to diagnose problems with the performance of the array by monitoring the difference between the predicted and actual performance of PV systems [5]. However, deterministic methods face some drawbacks for short-term prediction performance. More complex methodologies, such as statistical models and machine learning techniques, are more suitable for this task.

Unlike deterministic methods, machine learning models ignore any technical specifications of PV systems. Instead, this approach attempts to model the relationship and patterns within the time series data. Many studies have demonstrated the benefits of machine learning applied to solar radiation forecasting. [6] carried out an experimental study to compare three techniques: k nearest neighbor (kNN), multiple regression, and decision tree regressors. Similarly, [7] compared kNN, random forest, linear regression, and multilayer perceptron for the hourly solar irradiance of a Taiwanese PV system. Both studies agreed on the remarkable effectiveness of machine learning models applied to solar irradiance forecasting, especially kNN, which outperformed the others in terms of mean absolute error. Other studies have explored the

* Corresponding author.
E-mail address: pedro.larabenitez@bofa.com (P. Lara-Benítez).

possibility of combining different methods in a hybrid approach. [8] combined four models - ARIMA, SVM, ANN, and ANFIS - using genetic algorithms for one-hour-ahead solar forecasting in a building in Malaysia. The performance of single and ensemble models for solar energy prediction is analyzed in a detailed experimental study [9], where voting-ensemble techniques proved to outperform single models.

Recent improvements in the solar radiation prediction task have benefited from novel deep learning-based methods [10–12]. Complex neural network architectures have been established as state-of-the-art approaches for time series forecasting problems. These models are capable of capturing and extracting rich features and patterns from time series to achieve higher predictive accuracy [13]. [14] provided a detailed review of the latest developments in deep learning methods for this problem. In [15], a comprehensive study on the application of artificial neural networks is presented. Similarly, a feed-forward network with wavelet-based analysis is proposed for short-term energy prediction in [16]. More complex architectures, such as recurrent neural networks (RNN), have been successfully applied for irradiance forecasting in a Canadian PV plant [17,18]. Likewise, a deep convolutional neural network (CNN) obtained the best accuracy results compared to other machine learning algorithms such as Random Forest, Decision Trees, or Support Vector Machines [19]. Furthermore, in [20], a combined CNN-LSTM network outperformed other traditional machine learning methods in terms of prediction accuracy. Other studies have proposed hybrid regression models to reduce the prediction error [21,22].

Most of the existing literature in this field focuses on offline prediction. However, considering the nature of this problem, an online approach would provide a more practical solution for real-world applications. [23] presented an online sequential extreme learning machine for real-time solar radiation prediction. This technique is compared to other batch-learning algorithms. The experimental results of online learning have demonstrated good adaptability to changes in data over time. [24] also proposed an online solution based on the least mean square algorithm. This study proved the advantages of the online scenario over batch approaches due to the better tracking ability. To the best of our knowledge, the applicability of deep learning models for real-time solar irradiance forecasting has not yet been studied.

The particular data-streaming scenario requirements present a challenging machine learning problem [25]. Unlike traditional batch learning methods, data stream models must be able to efficiently process high-speed data. Therefore, an online model must be able to provide a prediction each time a new instance arrives. At the same time, the model must be updated to changes in the data distribution, which is known as concept drift [26]. Furthermore, since the stream can be infinite, historical data cannot be stored in memory, and the training and prediction phases of the model must be interleaved. The literature presents decision trees and ensemble algorithms as state-of-the-art techniques for data stream mining [27]. For instance, Hoeffding adaptive trees introduced by [28] have proven to have an excellent capacity to adapt to evolving data. Ensemble methods, such as Adaptive Random Forest [29], ADWIN, and Leveraging bagging [30] or Kappa Updated Ensemble [31], also provide competitive performance for data stream problems [29]. Most of these models use the so-called *active approaches*, which monitor the distribution of the arriving data and update or discard the models when a concept drift is detected. Another solution is to use a *passive approach*, which aims to keep the model updated by continuously training the model over the most recent instances [32]. This is particularly suitable for incremental learning models such as k-nearest neighbors or neural networks. This concept was implemented in the Asynchronous dual-pipeline Deep Learning framework for data Streams (ADLStream) [33]. This framework alleviates the computational cost of deep learning models by splitting the training and inference phases into two parallel processes. A detailed description of the ADLStream framework can be found in the following section.

In this work, we study the applicability and performance of deep learning models for short-term solar irradiance forecasting in a real-time streaming setting. For the experiments, we simulate a data stream of two different weather conditions at different speeds for a Canadian PV grid. Our study aims to achieve high forecasting accuracy, using a real-time model that can adapt to the evolution of the stream. For that reason, we have designed an extensive experimental study to evaluate four deep learning architectures: MLP, LSTM, CNN, and Transformer.

The main contribution of this work can be summarized as follows:

- An application of the ADLStream deep learning framework to real-time solar irradiance prediction in an online learning scenario.
- A thorough experimental study comparing state-of-the-art deep learning architectures for solar irradiance forecasting in streaming.
- An analysis of the forecasting accuracy across the photovoltaic grid depending on the spatial position.

The rest of the paper has been divided into the following parts: Section 2 introduces the materials used and the methods explored in the experimentation, together with the design of the experimental setup; Section 3 reports and discusses the results obtained; Section 4 presents the conclusions and future work.

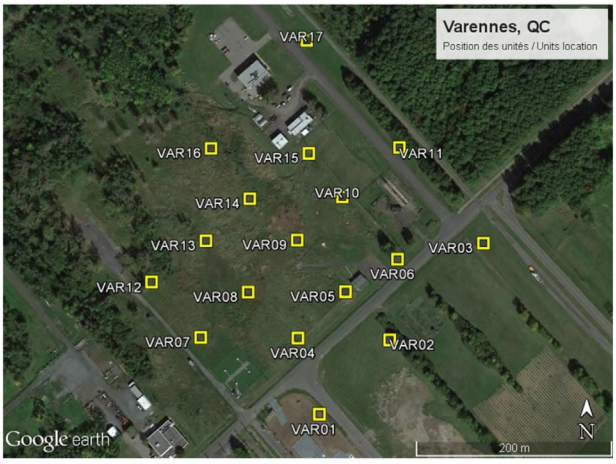
2. Materials and methods

This section introduces the materials and methods involved in the experimental study. SubSection 2.1 describes the details of the solar irradiance dataset and the preprocessing steps are discussed in SubSection 2.2.1. Furthermore, SubSections 2.2.2 and 2.2.3 provide a review of the asynchronous dual-pipeline deep learning framework used to simulate the streaming scenario and the different architectures considered. Finally, in SubSection 2.3 the design of the experimental setup is presented.

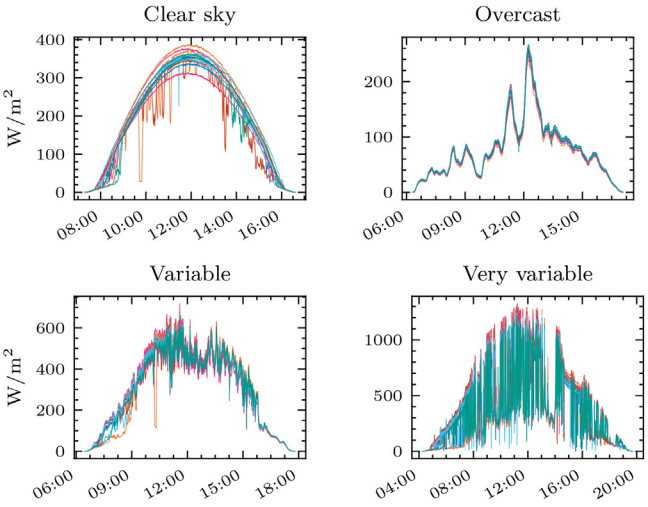
2.1. Datasets

The solar irradiance data used for the experimental study is obtained from the Canadian government [34]. The dataset provides short-term irradiance measurements for a PV system located in Varennes (Québec). As can be seen in Fig. 1a, the system consists of 17 irradiance sensors. Each PV unit takes a measurement every millisecond and averages them over a period of 10 ms. However, the data is not saved unless it changes by more than 5 W/m^2 or every 1 min. This setup allows recording cloud shades and high ramp-rate events, such as birds or insects obstructing the unit.

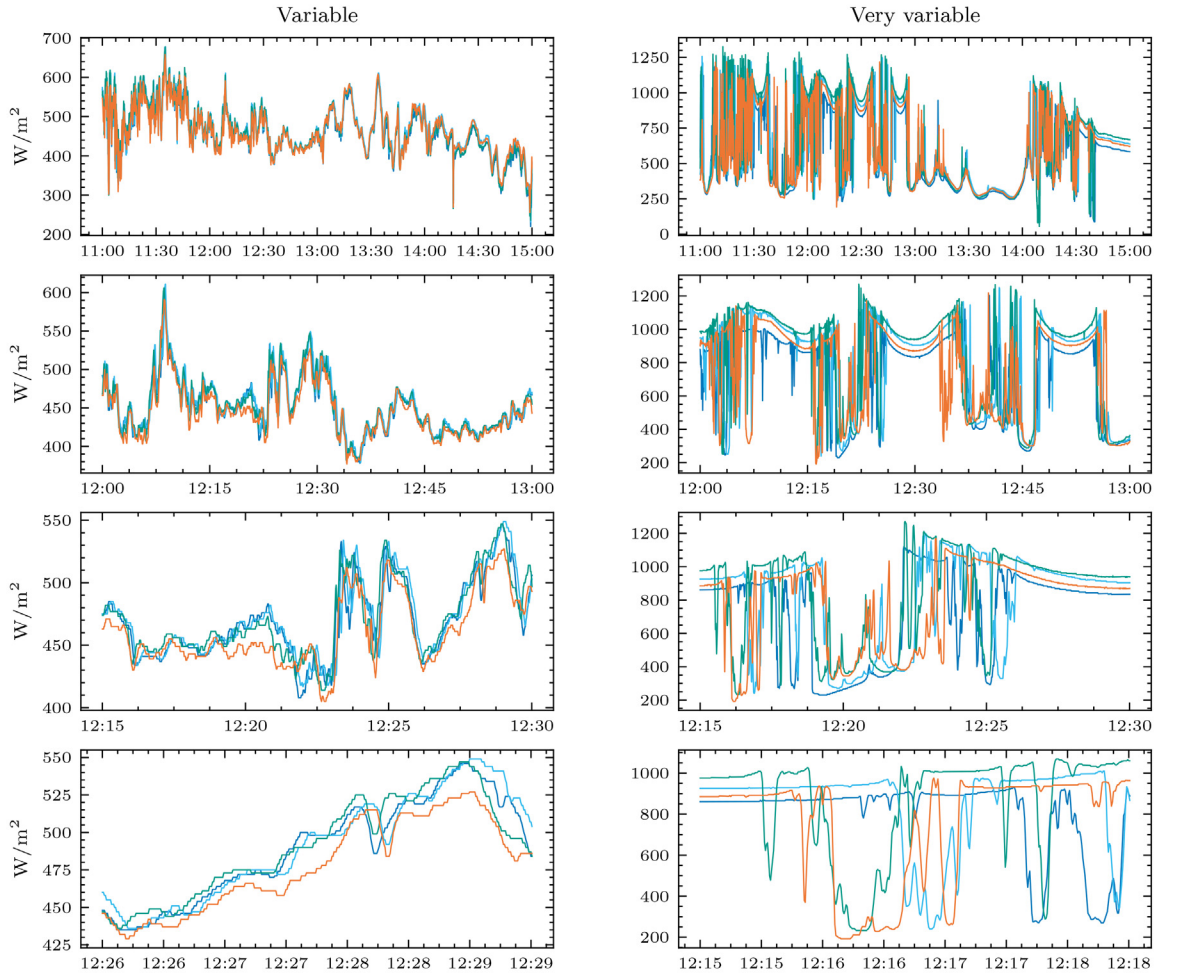
The dataset provides recordings for four complete days. Each day represents a different category based on cloud cover. As can be seen in Fig. 1b, the solar irradiance for the clear sky day presents a truly predictable distribution. With little to no short-term perturbations, this time series can be easily modeled using a traditional power modeling algorithm [35]. Similarly, the overcast day presents little variations in the short term. Therefore, we have excluded these two days from our study as they do not pose a significant challenge for short-term irradiance forecasting. Instead, we have selected the days with variable and very variable weather conditions. As shown in Fig. 1b, both datasets present a much more challenging problem with more disturbances.



(a) PV grid map of the Varennes solar farm [34].



(b) Solar irradiance time series at Varennes power plant. Each coloured line represents a different PV panel from the grid.



(c) Evolution of the *Variable* and *Very Variable* solar irradiance time series for four sensors (1, 3, 12 and 17) at different scales: 4 hours, 1 hour, 15 minutes and 3.5 minutes.

Fig. 1. Solar irradiance dataset.

Fig. 1c depicts several example instances at different time scales from the selected *Variable* and *Very Variable* datasets. This plot illustrates the difficulty of the short-term forecasting task addressed in this study. In the *Variable* case, the trend of the time series can be better observed, but there can be important changes, such as 100 units, in a matter of five minutes. The *Very variable* dataset poses an even more challenging problem. By observing the scale of the ordinate axis, it can be seen that the variations

can range more than 1000 units in just a couple of minutes. Furthermore, the figure shows that the spatial position of the panels plays an important role, given their different irradiance values at the same instant.

2.2. Methodology

In this subsection, we first describe the data preprocessing steps. Secondly, we describe the ADLStream framework, which is used to train and validate the models. Lastly, we present the four deep learning architectures used in this study.

2.2.1. Data preprocessing

As described in the previous section, the measuring process generates unevenly-spaced multivariate time-series data. In order to train the deep learning models, the datasets are sampled at specific rates: 1, 2, and 4 measurements per second for each sensor. This procedure is carried out by simply propagating the last valid observation forward. The resulting datasets are equally-spaced time series that can be used to simulate the streaming at three different speeds. Secondly, the values are scaled between 0 and 1 by applying min–max normalization. Finally, for the model selection experiments, the time series are shortened to the central six hours of the day, which represents the most challenging period given the high variability of the data.

In this study, we predict the next 30-s measurements of one sensor based on the previous values of all the sensors in the grid. The input of deep learning models must have a fixed length, known as the past history, which has been set to 3 min. By varying the frequency of the stream, the input and output length of the model changes accordingly. For instance, when the speed is set to 1 instance per second, the input length will be 180, corresponding to 3 min. However, when the speed is increased to 2 instances/s (0.5 Hz frequency), the 3 min of input will become 360 timesteps. Similarly, for a speed equal to 4 instances per second, the length of the input and output becomes 720 and 120 timesteps respectively. Table 1 shows the past-history and forecasting-horizon sequence length depending on the sample rate of the data. Furthermore, each timestep in the input sequence contains data from all 17 PV sensors at that particular moment in time. For example, for the frequency of 4 instances per second, the input sequence contains 720 timesteps $\times$ 17 sensors per timestep = 12240 sensor data points, compared to 3060 for the frequency of 1 instance per second.

The traditional moving-window scheme generates the input–output instances to feed the models [13]. The dashed box in Fig. 2 illustrates this process. The model's input is a 2D matrix with the past history information for each sensor. The aim of using this multivariate input is to capture not only temporal patterns within the time series, but also spatial dependencies between sensors in the grid [36].

2.2.2. ADLStream framework

In this study, we use the ADLStream framework introduced by [33]. The implementation is available as a Python library in [37].

Fig. 2 shows the architecture of this framework. As can be seen, the training and prediction phases are split into two asynchronous processes running in parallel. First, the data stream instances are preprocessed by the so-called stream generator. The preprocessing produces the input–output pairs for the models using a sliding window. As soon as an input window (x) is available, it passes to the prediction layer that generates a prediction (o). The input is then fed to the training process along with the expected output (y). The model is trained without blocking the prediction layer, which continues to generate forecasts as soon as a new input is available. Once a training epoch is over, the updated weights are forwarded to the prediction process. The training process uses the most recent instances, which allows adjusting the model to the evolution of the stream.

ADLStream is an end-to-end framework that integrates the whole data streaming process: from data preprocessing to model training and, ultimately, performance evaluation. The asynchronous nature of this framework facilitates the use of complex deep learning models, such as recurrent or convolutional, which would not be feasible in a data streaming scenario if they are trained sequentially. Furthermore, by constantly updating the model weights, it has shown to outperform current state-of-the-art alternatives for evolving data streams [25].

Table 1
Data preprocessing hyperparameters.

Data Stream Speed (instances/s)	**1**	**2**	**4**
Past History(3')	180	360	720
Forecasting Horizon(30")	30	60	120

2.2.3. Deep learning architectures

There exists a wide variety of architectures within the family of artificial neural networks. With the aim of improving solar irradiance forecasting, we explore the most relevant deep learning models from the literature: multi-layer perceptron, recurrent and convolutional networks, and transformer models.

Multi Layer Perceptron (MLP) It is the most basic type of artificial neural network. MLP is a feed-forward architecture that consists of an arbitrary number of hidden layers between the input and output layers. Each layer is composed of neurons called perceptrons. By combining these neurons, MLPs can approximate complex non-linear functions [38]. For regression tasks, the input layer has one neuron for each lagged feature, while the output layer has as many neurons as the prediction horizon.

Long-Short Term Memory Network (LSTM) The LSTM architecture is a recurrent neural network specifically designed for sequential data. Unlike the feed-forward approach, the LSTM neurons have a recurrent loop that connects the output of one layer with its input for the next timestep. The LSTM cell consists of three logic gates. The cell state acts as the long-term memory controlled by the input gate. In addition, the cell includes a forget gate that resets the cell state content when it becomes irrelevant. Finally, the output gate determines what information should propagate to the next step. This three-gate design makes this model capable of remembering patterns in larger horizons, hence suitable for time series data [39].

Convolutional Neural Network (CNN) Initially designed for image recognition, the CNN is a feed-forward architecture that assumes that inputs have a specific structure. These networks extract features from different regions of the input data. It is based on convolutional filters and the pooling operator. The model implements multiple convolutional filters that generate a feature map preserving the spatial information of the data. Then, the pooling operator reduces the dimensionality of the feature map. Unlike the MLP, each generated feature is connected to a small input region. This structure allows sharing the weights for each location of the input. By doing so, the network has significantly fewer weights. The result is a considerably faster and more efficient model. These characteristics also make CNNs suitable for dealing with sequential data such as time series [40]. In particular, one-dimensional convolutional filters extract meaningful features and capture repeated patterns within the input sequence.

Transformer Network Introduced by [41], the Transformer architecture is based on the Multi-Head Attention algorithm. The attention mechanism gives the model the ability to focus on the most relevant elements of long sequences. The attention layer has three main components, values, keys, and queries. The output is com-

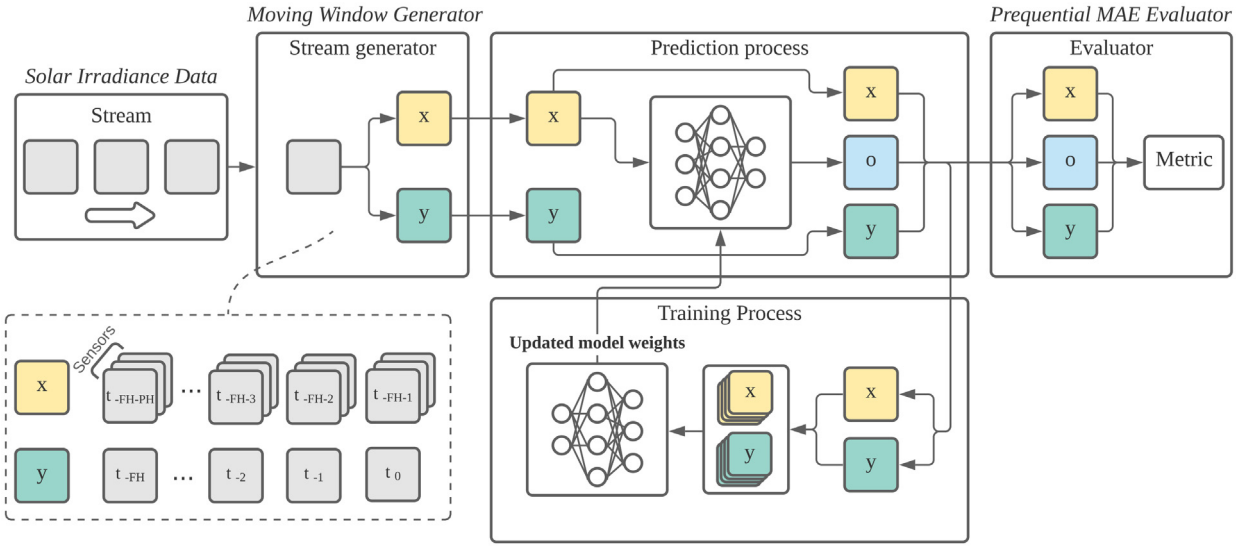


Fig. 2. Asynchronous dual-pipeline deep learning framework for solar irradiance forecasting. The dashed box illustrates how the input and output pairs are generated.

puted as a weighted sum of the values, where the weights are based on a compatibility function of the queries with the keys. For time series forecasting, a decode-only architecture is commonly used [42]. This transformer variant consists of several stacked decoder blocks that pass the features extracted from the previous decoder as input to the next one. The decoder block comprises a masked self-attention layer, followed by a multi-head attention layer and a fully-connected feed-forward layer. Unlike other deep learning architectures, the transformer predicts one step at a time for multi-step-ahead forecasting problems. Each prediction is used as an input to compute the next value. However, during the training process, the actual ground truth values are used at each step. This is known as the teacher forcing scheme, and it helps the model to converge faster [43].

Table 2
The parameter grid studied for the different deep learning architectures.

Model	Parameter	Values
MLP	Num H. Layers	{2, 4, 6}
	Neurons	{64, 128}
CNN	Num Layers	{2, 4}
	Filters	{64, 128}
	Pooling factor	{0, 2}
LSTM	Num Layers	{2, 4}
	Num Units	{64, 128}
	Return seq.	{True, False}
Transformers	Num Layers	{2, 4}
	d_{model}	{64, 128}
	h	{4, 8}

2.3. Experimental study

In this subsection, we present the design of the experimental study. First, in SubSection 2.3.1 the exhaustive grid-search parametrization conducted for each architecture is described. Then, in SubSection 2.3.2, the evaluation process is introduced. This involves the error metrics selected, the prequential methodology required for a correct evaluation in an online scenario, and the statistical analysis for a proper comparison of the models' performance.

2.3.1. Models

Finding the optimal architecture for deep learning models is a very challenging task. In order to find the best model for solar irradiance forecasting, we conducted an extensive grid search for each architecture. Table 2 presents the parameters and values explored for each deep learning architecture.

For the MLP, the experiments explore the main two properties of the model: the number of hidden layers and the number of neurons. The number of hidden layers defines the depth of the network, while the number of neurons defines the complexity of each layer. The resulting combinations derived from these two parameters provide a wide variety of architecture configurations.

In the case of the CNN, the search considers three characteristics: the number of stacked convolutional layers, the number of filters, and whether adding a pooling step after each layer. The kernel sizes follow a decreasing pattern at each layer with values ranging from 9 to 3, as commonly found in the literature [39]. The two-layer models have kernel sizes of 5 and 3. The models with four layers have kernel sizes of 9, 7, 5, and 3.

Similarly to CNN, eight different LSTM architectures have been studied. The grid search considers three parameters: the number of recurrent layers, the number of units of each layer, and whether to return the complete sequences. If the return sequence is set to true, the output of the last layer will be the output of each unit for each timestep. Otherwise, only the last step is returned, with one value for each unit. This study considers a multi-step-ahead forecasting problem. Therefore, for both the CNN and LSTM, the output of the convolutional and recurrent block is connected to a fully-connected layer that consists of as many neurons as the prediction horizon.

Finally, the grid on the Transformer architecture involves three parameters: the number of decoder blocks, the dimension of the model (d_{model}), and the number of attention heads (h) are studied. The number of layers refers to the number of stacked decoder blocks, while the dimension of the model describes the number of features in the decoder input. Lastly, the number of heads represents the number of linear projections in the multi-head attention module.

Besides the architecture-specific parameters, the experimental study considers various data frequencies, hence different stream speeds. The set of training hyperparameters explored can be found in Table 3. The forecasting horizon has been fixed to 30 s, with the aim of evaluating the short-term prediction performance. The past

Table 3
Training hyperparameters grid.

Parameter	Values
Data Stream Frequency	1, 2, 4 (instances/s)
Past History	3 min
Forecasting Horizon	30 s
Batch Size	90
Num. Batches Fed	60

history window has been set to 3 min, since it provided better results after exploring with other values such as 1 or 5 min. This configuration results in different numbers of time steps as input and output depending on the data frequency (Table 1). For the batch size and the number of batches fed, the optimal values are selected as previously studied in [33].

2.3.2. Evaluation process

The deep learning models are evaluated using the prequential MAE and prequential MAPE as accuracy metrics. Furthermore, a statistical analysis of the results is carried out.*Prequential Evaluation* Unlike in a traditional batch-learning scenario, cross-validation techniques are not feasible for data stream problems. The high speed and unlimited data coming from the stream make cross-validation computationally expensive. Furthermore, in traditional batch learning, it is commonly assumed that the data is independently and identically distributed, the data has no particular order, and its distribution does not change. However, in a streaming scenario, this assumption cannot be sustained. The stream is generated in a particular order, and the data distribution evolves over time. Therefore, creating a picture of accuracy over time is essential to evaluate a data stream model [44]. This study implements the prequential evaluator with a decaying factor. This method uses an interleaved test-then-train approach with chunks of $k = 10$ instances. Moreover, a decaying factor of $\alpha = 0.99$ is used to give more importance to recent examples. The prequential accuracy (P_α) at the moment i can be computed as described in the following recursive equation.

$$P_\alpha(i) = \frac{\sum_{k=1}^{i} \alpha^{i-k} L(y_k, o_k)}{\sum_{k=1}^{i} \alpha^{i-k}} = \frac{S_\alpha(i)}{B_\alpha(i)} \tag{1}$$

$$S_\alpha(i) = L(y_i, o_i) + \alpha \times S_\alpha(i-1), B_\alpha(i) = n_i + \alpha \times B_\alpha(i-1)$$

where y and o refer to real and predicted values respectively, L_i is the loss function -MAE or MAPE in our study- and n_i is the number of processed instances at time i.*Validation Metric* Two error metrics are used to assess the accuracy of the models. First, the Mean Absolute Error (MAE) reports the absolute difference between the predicted and real values. The error is measured in the original units (W/m^2) as it is an absolute metric. In contrast, the other metric used provides the percentage error. The Mean Absolute Percentage Error (MAPE) metric provides a more intuitive interpretation in terms of relative error. These metrics are defined as follows.

$$MAE(y, o) = \frac{\sum_{i=1}^{n} |y_i - o_i|}{n}, MAPE(y, o) = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - o_i}{y_i} \right| \tag{2}$$

where y and o are two vectors with the real and predicted values, respectively. Both vectors have a length equal to the forecasting horizon (n). These metrics are among the most widely used for forecasting problems and will help to provide a complete analysis of the results.*Statistical analysis* For a correct comparison of the performance of the models, a statistical analysis is carried out using Friedman's procedure. This non-parametric test allows for the detection of global differences and provides a ranking of the algorithms. The Friedman test assumes the null hypothesis that all algorithms perform similarly. This hypothesis can be rejected if the p-value obtained is below the significance level ($\alpha = 0.05$). In that case, Hommel's post hoc analysis is conducted to determine whether there is statistical significance in the performance of the different techniques when compared pairwise.

3. Results and discussion

In this section, the experimental results are presented and discussed. For reproducibility purposes, the complete source code of the experimental study is available in [45]. The experiments were carried out with an Intel Core i7-770 K CPU and two NVIDIA GeForce GTX 1080 8 GB GPUs.

3.1. Hyperparameter sensitivity analysis

First, it is important to analyze the hyperparameter search carried out. For this set of experiments, we predict a single PV farm sensor (*VAR 01*) based on the solar irradiance history of the entire PV grid. The grid search with the different data frequencies results in 90 models for each dataset. The experiments are conducted in real-time, which implies a high computational cost for finding the best hyperparameter values. A single experiment with one model configuration lasts as many hours as the length of the dataset, which is six hours in this case. Therefore, a total of 1,080 h (45 days) have been required to complete the hyperparameter search for all 90 models. We explored two different datasets, and each experiment lasted 6 h (90 models x 6 h x 2 datasets = 1080 h).

Table 4 presents the MAE distribution of the experimental grid-search parametrization conducted for each architecture and stream speed. The results indicate that a speed of 4 instances per second is too fast for a prediction horizon of 30 s. At this speed, the input and output sequences reach a length of 720 and 120 timesteps, respectively, as explained in Table 1. The larger dimension of the output increases the complexity of the problem significantly. Furthermore, the input windows are larger, and the income rate of the data is much higher than the case of one instance per second. Considering the limitations of the available hardware, these aspects imply that the model is re-trained less frequently and it is harder to adapt to changes in the data stream.

In terms of mean performance among all configurations, the MLP is the best for both datasets, closely followed by the CNN architecture. By analyzing the standard deviation of the results reported in Table 4, it is possible to assess the complexity of the parameterization of each architecture. The transformer is the most difficult model to parametrize as it presents the widest variety of results. On the other hand, the easiest architectures to parametrize are, MLP, CNN, and LSTM, respectively.

After analyzing the average performance of all possible hyperparameter settings, Table 5 presents the best configuration found for each architecture. Furthermore, Table 6 reports the results obtained with these optimal configurations for both datasets. The experiments involve a multi-step forecasting problem, hence the models produce an output of length forecasting horizon. Therefore, we report the prequential MAE and MAPE averaged across the entire forecasting horizon (*Avg.* column). We also include the error for the last prediction - i.e. thirty seconds ahead (*30*" column). For the Variable dataset, the MLP model achieves the best results, closely followed by the CNN. With regard to the Very Variable dataset, the CNN provides the best performance. The LSTM ranked second in this dataset, and the MLP achieved significantly worse results.

Table 4
Mean and standard deviation of prequential MAE among all configurations for each model and each stream speed.

Weather	Model	Mean			Standard deviation		
Variable	MLP	**8.581**	**10.604**	342.540	**0.529**	0.927	**8.397**
	CNN	10.731	10.661	334.796	1.308	**0.848**	12.228
	LSTM	15.552	14.823	**334.605**	3.327	1.496	9.235
	Transformer	33.450	34.018	339.129	11.602	11.639	12.026
Very variable	MLP	**34.424**	26.794	477.299	**0.285**	**1.043**	20.993
	CNN	34.781	**17.027**	591.384	5.383	2.286	**0.013**
	LSTM	42.362	20.471	551.733	4.582	2.198	51.867
	Transformer	110.074	80.914	**423.333**	36.206	48.981	105.429
Stream speed (instances/s)		1	2	4	1	2	4

Table 5
Configuration of the best models for each architecture and dataset.

Architecture	Parameter	Dataset	
		Variable	Very Variable
MLP	Num. H. Layers	6	2
	Neurons	64	64
CNN	Num. Layers	2	4
	Filters	128	64
	Pooling factor	0	0
LSTM	Num. Layers	2	2
	Num. Units	64	128
	Return seq.	True	False
Transformer	Num. Layers	4	2
	d_{model}	64	128
	h	4	8

We hypothesize that the poor performance of the Transformer architecture is due to their longer training time, which prevents the model from adapting to the fast changes in the stream. In terms of stream frequency, the results show that all models agree on the best stream speed for each dataset. For the Variable weather, the slowest stream achieved the best result. However, for the Very Variable dataset, a faster speed (2 instances per second) helps to improve the forecasting accuracy. Overall, the 2% and 6% MAPE results in the Variable and Very variable datasets, respectively, demonstrate the effectiveness of our proposal given the complexity of this short-term forecasting task. The worse results obtained with Transformers also demonstrate that the errors can be doubled if a wrong architecture configuration is selected. This fact illustrates the importance of the extensive grid search carried out in the experimental study.

Fig. 3 shows the evolution of the prequential MAE over time for each model for both data sets. For variable weather, all models show similar behavior. They achieve fast convergence and good adaptation to the concepts drifts in the data. However, the very variable weather dataset seems to be a significantly more complex dataset, as all architectures encounter more difficulties in modeling the distribution of the series. The prequential MAE plot exhibits some up-and-down patterns corresponding to drifts in the data distribution. It is worth noting that CNN and LSTM are the fastest to converge and adjust to concept drifts. The MLP performed well for the Variable dataset, but struggled to give accurate predictions for more complex time series such as the Very Variable dataset. On the other hand, the Transformer fails to converge and adapt to the evolving data stream.

3.2. Qualitative analysis

The findings discussed in the previous section are further confirmed when analyzing visually particular examples of predictions. In this part of the study, we select the best model configuration for each type of network and analyze the MAE error over the complete datasets. In this case, the datasets are not limited to the central six hours of the day.

Figs. 4 and 5 illustrate four examples of predictions for each dataset. The first two line plots of each figure show the actual values of the time series, and the evolution of the prequential error obtained for each model architecture over time. The position of the selected examples within the series is indicated and numbered with purple rectangles.

For the *Variable* dataset (Fig. 4), the irradiance values in the central hours of the day range from 400 to 600 W/m^2. The prequential MAE during this period remains below 25 W/m^2 for CNN and MLP models. In contrast, the MAE for LSTM is always higher, reaching peaks of 40 W/m^2. Four time intervals have been selected as diverse examples for further analysis. Example 1, which covers the period from 10:15 to 10:25, shows a relatively gradual drop in irradiance lasting about 3 min (dashed grey line). It can be seen that both CNN and MLP are able to predict the time series behavior nearly in real-time. Example 2 (from 10:55 to 11:05) displays sharp rises and falls between 300 and 600 W/m^2 within a minute. In this case, the models experience a significant rise in MAE. As can be observed, the CNN and MLP react fairly quickly and efficiently to the drift. However, the Transformer could not adapt to the evolving stream, with its output being rather stable and unaware of the data

Table 6
Best results for each model and dataset. The prequential MAE averaged over the forecasting horizon predictions and the prequential MAE of the single last prediction (at second 30) are reported.

Dataset	Model	Stream speed (instances/s)	MAE		MAPE	
			Avg.	30"	Avg.	30"
Variable	MLP	1	**8.034**	**11.925**	**2.272**	**3.317**
	CNN	1	8.749	14.141	2.458	3.934
	LSTM	1	12.415	17.765	3.526	5.133
	Transformer	1	19.383	21.964	5.494	6.053
Very Variable	MLP	2	25.621	38.103	7.244	11.86
	CNN	2	**14.591**	**20.399**	**6.106**	**8.136**
	LSTM	2	17.562	22.344	7.304	8.606
	Transformer	2	42.386	42.671	12.556	13.152

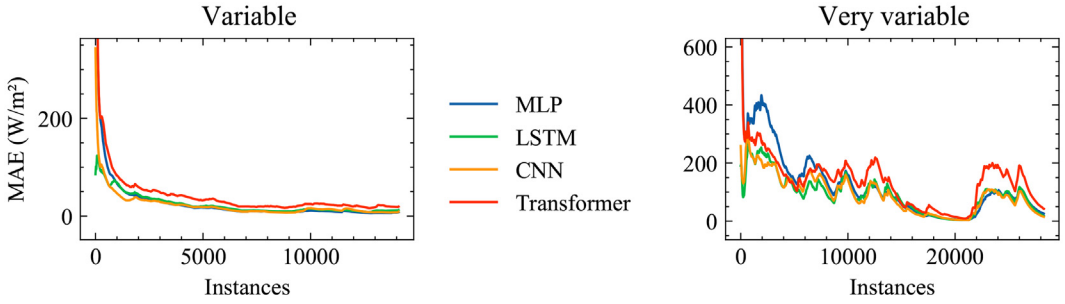


Fig. 3. Prequential MAE over time for the best models for each dataset.

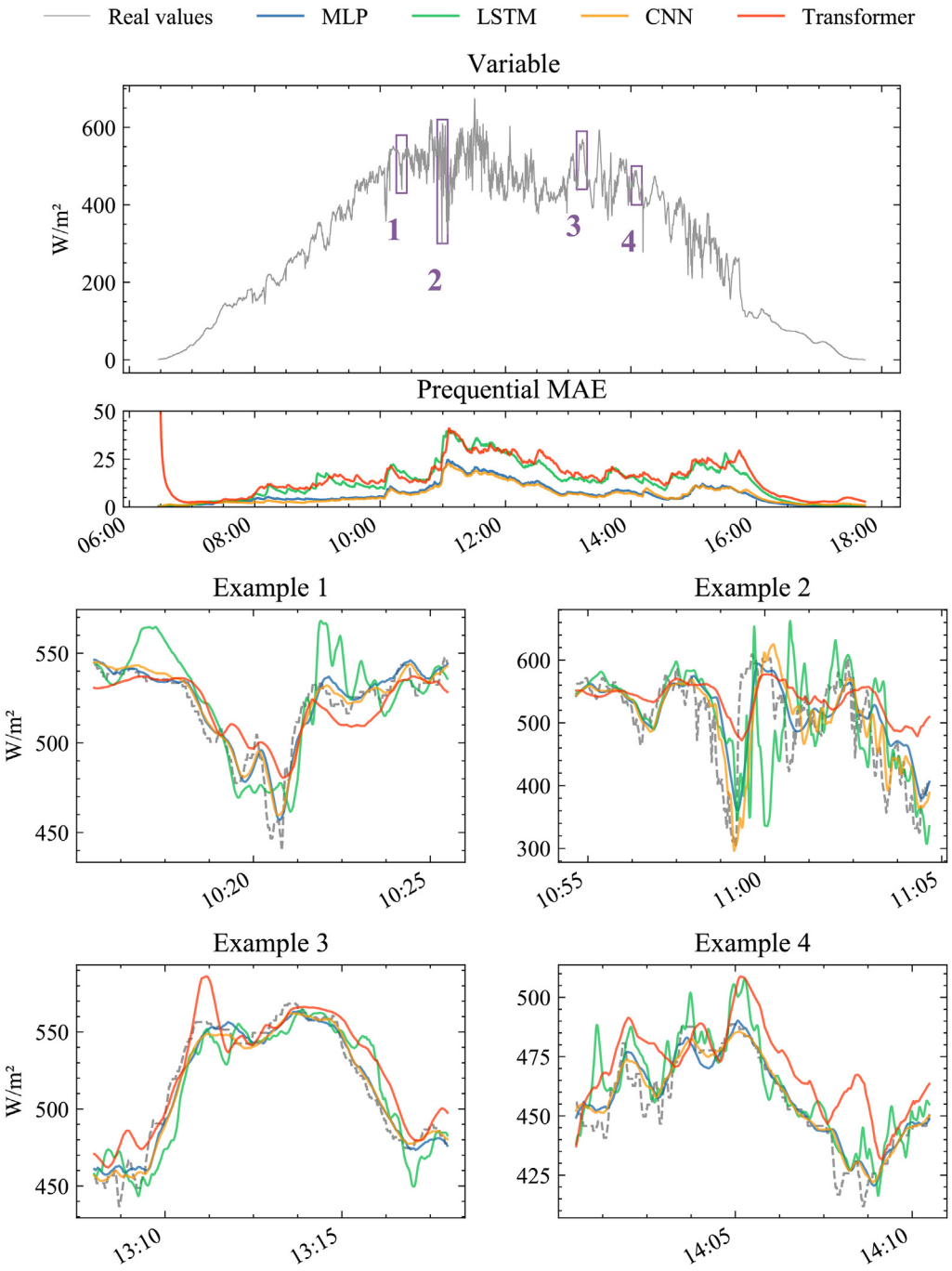


Fig. 4. Examples of some of the worst and best predictions for *Variable* weather. The first two plots show the actual value of the time series (grey line) and the evolution of the prequential error for each model over time (colored lines). The purple numbered squares correspond to each selected and zoomed example in the subsequent figures.

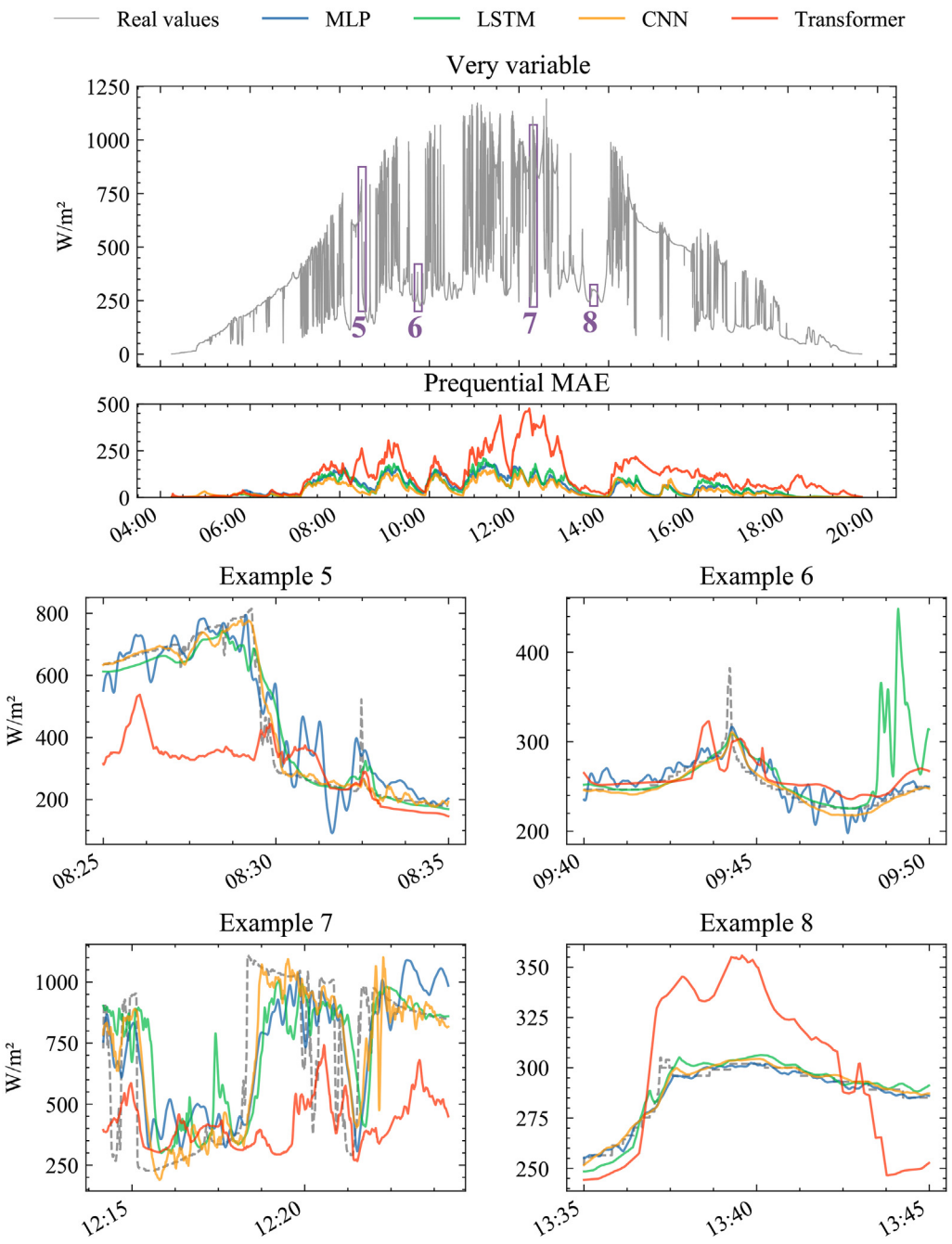


Fig. 5. Examples of some of the worst and best predictions for *Very Variable* weather. The first two plots show the actual value of the time series (grey line) and the evolution of the prequential error for each model over time (colored lines). The purple numbered squares correspond to each selected and zoomed example in the subsequent figures.

changes. Similarly, the LSTM did not adapt effectively to the concept drift. Although it showed a positive reaction to the drift, the predictions are quite variable, leading to high errors. Example 3 (between 13:10 and 13:20) presents a gradual rise and fall of solar irradiance of almost 100 W/m^2. In this example, all models perform well in capturing the rise and fall. Nevertheless, as in the previous examples, MLP and CNN achieve the best results. Finally, Example 4 (between 14:00 and 14:10) illustrates an interval with minor but short upward and downward movements. In this scenario, CNN and MLP were able to predict the trend. However, they were not able to capture all the changes in the stream and suffered delays in some predictions. Again, LSTM and Transformer experience difficulties in this scenario. They exhibit what could be an overfitting pattern and perform significantly worse than the CNN and MLP.

In the *Very Variable* dataset (Fig. 5), the irradiance varies more than 800 W/m^2 within a few minutes. The prequential MAE plot shows that the CNN model offers the best performance, remaining almost always below the other three models. Over the entire day, the maximum MAE value of the CNN was below 100 W/m^2, except before 4 p.m., when it occasionally exceeded 400 W/m^2. In Example 5 (between 8:25 and 8:35), the CNN model behaves in a fairly stable way even with drops up to 600 W/m^2. CNN outperforms MLP, which has more erratic behavior, achieving worse results than LSTM. In Example 6, a very good performance of the CNN can be observed again. While MLP is able to model the trend, the predictions present too many oscillations. In this more stable example, even Transformer performs better than LSTM. Example 7 is almost an impossible prediction case, as it presents consecu-

tive and almost instantaneous rises and falls of more than 700 W/m^2. Understandably, no model is capable of accurately predicting this series. However, CNN can quickly capture the changes in both upward and downward trends, albeit with some delay. Finally, Example 8 is a stable window in which all models, except the Transformers, are able to adjust smoothly to the data. In summary, the Transformer struggles to model the time series, resulting in inaccurate predictions. The LSTM performed better than in the *Variable* dataset, achieving results similar to those of the MLP. In general, CNN shows the best performance among all architectures. It achieves nearly ideal predictions for the most stable examples (Ex. 6 and 8) and shows a fast adaptation to the changes in the data for the more variable examples (Ex. 5 and 7).

3.3. Statistical analysis

The MAE and MAPE results have been analyzed through a statistical test to correctly evaluate the performance of each DL architecture. The global ranking obtained from the application of the Friedman test is presented in Table 7. CNN leads, the ranking followed by MLP and LSTM. The null hypothesis can be rejected since the p-value (< 0.00001) is below the significance level ($\alpha = 0.05$). This means that the experimental data provide enough evidence to confirm that the different architectures do not behave similarly. Therefore, we have to carry on with the post hoc analysis where a pair-wise comparison between the CNN and the rest of the models. The result of this analysis is reported in Table 8. Using a significance level of $\alpha = 0.05$, Hommel's procedure determines that CNN outperforms the Transformer significantly. However, there are no significant differences compared to MLP and LSTM.

3.4. Forecasting accuracy across the PV grid

As the final part of our contribution, we have studied the spatial dependencies in the dataset and how the position of the sensors may affect the forecasting performance. We have evaluated the results of the best CNN model across different PV sensors of the grid. As shown in Fig. 6, we selected a sample of 6 sensors covering the edges of the PV plant map.

Table 9 reports the MAE and MAPE results for the six sensors for both datasets: Variable and Very Variable. There is a considerable variation between the results obtained for the different PV sensors. For the Variable dataset, sensors 03 and 01 obtain the best results, while sensors 12 and 17 perform the worst. By observing the location of these sensors, it is evident that the best results are obtained for those on the southeast edge, while the worst results are found in the opposite location (northwest). Something similar happens for the Very Variable dataset, but in this case, the best results are obtained for the sensors positioned on the right side of the map (sensor 03), while the worst ones are located on the left side (sensor 12). In both datasets, a clear spatial dependence is observed. This behavior allows us to deduce the wind direction for the measurement days: from northeast to southwest in the case of Variable, and from east to west for Very Variable. These results confirm the effectiveness of the deep learning model in accurately capturing the spatial information, as well as external factors such as wind speed and direction, without the need to explicitly provide them. Furthermore, this study demonstrates the suitability of using an online-learning scenario for this problem. The proposed model is able to learn and adapt to meteorological changes instantly. Instead, a model trained offline would need a sufficiently large dataset to cover all possible meteorological scenarios, which still does not ensure encountering different conditions.

Table 7
Friedman Test Ranking.

Model	Rank
CNN	1.79
MLP	2.25
LSTM	2.46
Transformer	3.50

Table 8
Hommel's post hoc analysis.

Comparison	z	p-value
CNN – Transformer	4.58	<0.0001
CNN – LSTM	1.79	0.15
CNN – MLP	1.23	0.22

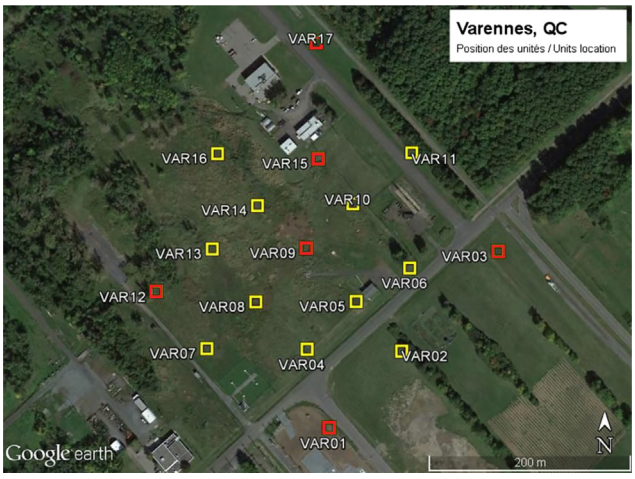


Fig. 6. PV grid map with the selected sensors colored in red.

Table 9
Accuracy results for CNN model across different sensors of the PV grid.

Weather	Sensors	MAE		MAPE	
		Avg	30"	Avg	30"
Variable	VAR 01	8.72	14.10	2.45	3.93
	VAR 03	7.82	14.78	2.18	4.02
	VAR 09	10.39	17.58	2.73	4.57
	VAR 12	12.97	21.37	3.62	6.00
	VAR 15	11.70	20.70	3.31	5.81
	VAR 17	14.60	26.25	3.86	6.93
Very variable	VAR 01	14.44	20.19	6.25	8.29
	VAR 03	10.93	15.46	1.97	2.84
	VAR 09	11.82	17.84	2.43	3.85
	VAR 12	22.28	31.42	6.13	9.17
	VAR 15	13.30	20.01	2.27	3.44
	VAR 17	16.24	20.87	2.49	3.17

4. Conclusions

In this study, we propose a deep learning model to perform short-term solar irradiance forecasting in an online-learning scenario. For the experiments, we simulated a data stream at three different speeds based on a Canadian photovoltaic (PV) system for two separate days with high solar radiation volatility. The state-of-the-art deep learning architectures for time series forecasting were considered: multilayer perceptron (MLP), long-short term memory network (LSTM), convolutional neural network (CNN), and Transformer network. Consequently, an extensive grid search parametrization was conducted to obtain the best model architecture, testing a total of 90 models for each dataset. Furthermore, the mean absolute error (MAE) and the mean absolute per-

centage error (MAPE) were reported and analyzed in order to compare the forecasting performance of the different models.

Overall, these results confirmed the great potential of deep learning models for short-term solar irradiance forecasting. In particular, the MLP achieved the best performance for the dataset with less irradiance variability. The simplicity of this architecture allows rapid convergence, and efficient adaptation to the concept drifts when the problem is not overly complex. However, MLP performed worse in the most volatile and challenging dataset. In this case, the CNN architecture outperformed the forecasting accuracy of the other architectures, closely followed by LSTM. The statistical analysis found no significant differences in the performance of these three architectures. In contrast, the Transformer architecture could not successfully capture the temporal dependencies of the series, achieving the worst results for both datasets. This issue confirmed that the complexity of the Transformer poses a disadvantage in a streaming scenario. In addition, the results of the grid search showed that LSTM and Transformer are significantly more difficult to parametrize than MLP and CNN. Regarding the stream frequency, the results showed that, for the variable dataset, a slow stream of 1 instance per second is preferred. In contrast, for the very variable dataset, a faster stream of 2 instances per second helps to capture the higher-frequency disturbance of the stream. Finally, streams at 4 instances per second proved to be too fast and too complex to successfully address the problem. Furthermore, the forecasting accuracy analysis across the different PV sensors proved the advantages of using an online-learning scenario for this problem. The model effectively captured the spatial and external-factors information, such as wind speed and direction, from the input data.

Future research should consider the potential benefit of combining various deep learning architectures for this problem, such as CNN-LSTM hybrid models. Additional studies should explore other sampling methods that consider the distribution of the stream, and more sophisticated hyper-parameter search techniques. Another interesting research direction is the use of external data, such as environmental conditions variables, to improve the accuracy of irradiance forecasting. Finally, a more in-depth analysis of the spatial component of the solar irradiance problem could be carried out. Future work should study the importance of feature selection for forecasting tasks based on the spatial distribution of the photovoltaic grid.

Funding

This research has been funded by Ministerio de Ciencia e Innovación under the projects: Aprendizaje Profundo y Transferencia de Aprendizaje Eficientes para Salud y Movilidad (PID2020-117954RB-C22), and Soluciones Digitales para Mantenimiento Predictivo de Plantas Eólicas (TED2021-131311B); and by the Andalusian Regional Government under the project Modelos de Deep Learning para Sistemas de Energía Renovable: Predicción de Generación y Mantenimiento Preventivo y Predictivo (PYC20 RE 078 USE).

Data availability

Data is available online. See reference in the manuscript.

Acknowledgments

We are grateful to NVIDIA for their GPU Grant Program that has provided us the high-quality GPU devices to carry out the study.

References

[1] Y. Chu, M. Li, C.F. Coimbra, D. Feng, H. Wang, Intra-hour irradiance forecasting techniques for solar power integration: A review, iScience 24 (10) (2021), https://doi.org/10.1016/j.isci.2021.103136, ISSN 2589-0042.
[2] Y. Wang, D. Millstein, A.D. Mills, S. Jeong, A. Ancell, The cost of day-ahead solar forecasting errors in the United States, Solar Energy 231 (2022) 846–856, ISSN 0038–092X, DOI: 10.1016/j.solener.2021.12.012.
[3] G. Graditi, S. Ferlito, G. Adinolfi, Comparison of Photovoltaic plant power production prediction methods using a large measured dataset, Renewable Energy 90 (2016), https://doi.org/10.1016/j.renene.2016.01.027, ISSN 0960-1481.
[4] H. Morf, A validation frame for deterministic solar irradiance forecasts, Renewable Energy 180 (2021) 1210–1221, https://doi.org/10.1016/j.renene.2021.08.032.
[5] D.L. King, J.A. Kratochvil, W.E. Boyson, Photovoltaic array performance model (2004) 1–43.
[6] A. Abubakar Mas'ud, Comparison of three machine learning models for the prediction of hourly PV output power in Saudi Arabia, Ain Shams Eng. J. 13 (4) (2022), https://doi.org/10.1016/j.asej.2021.11.017, ISSN 2090-4479.
[7] C.-C. Wei, Predictions of Surface Solar Radiation on Tilted Solar Panels using Machine Learning Models: A Case Study of Tainan City, Taiwan, Energies 10 (10), ISSN 1996–1073, DOI: 10.3390/en10101660.
[8] Y.-K. Wu, C.-R. Chen, H. Abdul Rahman, A Novel Hybrid Model for Short-Term Forecasting in PV Power Generation, International Journal of Photoenergy ISSN 1110-662X, 10.1155/2014/569249.
[9] M. Ayoub, Contrasting accuracies of single and ensemble models for predicting solar and thermal performances of traditional vaulted roofs, Solar Energy 236 (2022) 335–355, ISSN 0038–092X, DOI: 10.1016/j.solener.2022.02.053.
[10] B.K. Fontes Rodrigues, M. Gomes, Â.M. Oliveira Santanna, D. Barbosa, L. Martinez, Modelling and forecasting for solar irradiance from solarimetric station, IEEE Latin Am. Trans. 20 (2) (2022) 250–258, https://doi.org/10.1109/TLA.2022.9661464.
[11] A. Bhatt, W. Ongsakul, N.M.M., J.G. Singh, Sliding window approach with first-order differencing for very short-term solar irradiance forecasting using deep learning models, Sustain. Energy Technol. Assessments 50 (2022) 101864, ISSN 2213-1388, 10.1016/j.seta.2021.101864.
[12] S. Wang, Y. Wang, Y. Liu, N. Zhang, et al., Hourly solar radiation forecasting based on EMD and ELM neural network, Electric Power Autom. Equipment 34 (8) (2014) 7–12.
[13] P. Lara-Benítez, M. Carranza-García, J.C. Riquelme, An Experimental Review on Deep Learning Architectures for Time Series Forecasting, Int. J. Neural Syst. 31 (03) (2021) 2130001, https://doi.org/10.1142/S0129065721300011.
[14] S. Aslam, H. Herodotou, N. Ayub, S.M. Mohsin, Deep Learning Based Techniques to Enhance the Performance of Microgrids: A Review, in: 2019 International Conference on Frontiers of Information Technology (FIT), 116–1165, 2019, DOI: 10.1109/FIT47737.2019.00031.
[15] A. Geetha, J. Santhakumar, K.M. Sundaram, S. Usha, T.T. Thentral, C. Boopathi, R. Ramya, R. Sathyamurthy, Prediction of hourly solar radiation in Tamil Nadu using ANN model with different learning algorithms, Energy Reports 8 (2022) 664–671, ISSN 2352–4847, DOI: 10.1016/j.egyr.2021.11.190, 2021 The 8th International Conference on Power and Energy Systems Engineering.
[16] F. Rodríguez, I. Azcárate, J. Vadillo, A. Galarza, Forecasting intra-hour solar photovoltaic energy by assembling wavelet based time-frequency analysis with deep learning neural networks, Int. J. Electr. Power Energy Syst. 137 (2022) 107777, ISSN 0142-0615, 10.1016/j.ijepes.2021.107777.
[17] A. Alzahrani, P. Shamsi, M. Ferdowsi, C. Dagli, Solar irradiance forecasting using deep recurrent neural networks, in: 2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA), 988–994, 2017a, DOI: 10.1109/ICRERA.2017.8191206.
[18] A. Alzahrani, P. Shamsi, C. Dagli, M. Ferdowsi, Solar Irradiance Forecasting Using Deep Neural Networks, Procedia Computer Science 114 (2017b) 304–313, ISSN 1877-0509, 10.1016/j.procs.2017.09.045, complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017, Chicago, Illinois, USA, 2017.
[19] P.-H. Kuo, C.-J. Huang, A Green Energy Application in Energy Management Systems by an Artificial Intelligence-Based Solar Radiation Forecasting Model, Energies 11 (4), ISSN 1996–1073, DOI: 10.3390/en11040819.
[20] N. Elizabeth Michael, M. Mishra, S. Hasan, A. Al-Durra, Short-Term Solar Power Predicting Model Based on Multi-Step CNN Stacked LSTM Technique, Energies 15 (6), ISSN 1996–1073, DOI: 10.3390/en15062150.
[21] J.-L. Casteleiro-Roca, P. Chamoso, E. Jove, A. González-Briones, H. Quintián, M.-I. Fernández-Ibáñez, R.A. Vega Vega, A.-J. Piñón Pazos, J.A. López Vázquez, S. Torres-Álvarez, T. Pinto, J.L. Calvo-Rolle, Solar Thermal Collector Output Temperature Prediction by Hybrid Intelligent Model for Smartgrid and Smartbuildings Applications and Optimization, Appl. Sci. 10 (13), ISSN 2076–3417, DOI: 10.3390/app10134644.
[22] M.T. García-Ordás, H. Alaiz-Moretón, J.-L. Casteleiro-Roca, E. Jove, J.A. Benítez-Andrades, I. García-Rodríguez, H. Quintián, J.L. Calvo-Rolle, Clustering

Techniques Selection for a Hybrid Regression Model: A Case Study Based on a Solar Thermal System, Cybern. Syst. (2022) 1–20, https://doi.org/10.1080/01969722.2022.2030006.
[23] J. Zhang, Y. Xu, J. Xue, W. Xiao, Real-time prediction of solar radiation based on online sequential extreme learning machine, in: 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), 53–57, 2018, DOI: 10.1109/ICIEA.2018.8397688.
[24] S.A. Fatemi, A. Kuh, M. Fripp, Online and batch methods for solar radiation forecast under asymmetric cost functions, Renewable Energy 91 (2016), https://doi.org/10.1016/j.renene.2016.01.058, 397–408, ISSN 0960-1481.
[25] P. Lara-Benítez, M. Carranza-García, D. Gutiérrez-Avilés, J.C. Riquelme, Data streams classification using deep learning under different speeds and drifts, Logic J. IGPL ISSN 1367–0751, DOI: 10.1093/jigpal/jzac033, jzac033.
[26] T. Lesort, M. Caccia, I. Rish, Understanding continual learning settings with data distribution drift analysis, arXiv preprint arXiv:2104.01678.
[27] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under Concept Drift: A Review, IEEE Trans. Knowl. Data Eng. 31 (12) (2019) 2346–2363, https://doi.org/10.1109/TKDE.2018.2876857.
[28] P. Domingos, G. Hulten, Mining high-speed data streams, in: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, 71–80, 2000, DOI: 10.1145/347090.347107.
[29] H.M. Gomes, A. Bifet, J. Read, J.P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, T. Abdessalem, Adaptive random forests for evolving data stream classification, Mach. Learn. 106 (9) (2017) 1469–1495, ISSN 1573–0565, DOI: 10.1007/s10994-017-5642-8.
[30] A. Bifet, R. Gavaldà, Adaptive Learning from Evolving Data Streams, in: Advances in Intelligent Data Analysis VIII, Springer, Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 249–260, ISBN 978-3-642-03915-7,.
[31] A. Cano, B. Krawczyk, Kappa Updated Ensemble for drifting data stream mining, Mach. Learn. 109 (2019), https://doi.org/10.1007/s10994-019-05840-z, 109–218, ISSN 1573-0565.
[32] A. Bifet, B. Hammer, F. Schleif, Recent trends in streaming data analysis, concept drift and analysis of dynamic data sets, in: 27th European Symposium on Artificial Neural Networks, ESANN 2019, Bruges, Belgium, April 24–26, 2019, 2019.
[33] P. Lara-Benítez, M. Carranza-García, J. García-Gutiérrez, J.C. Riquelme, Asynchronous dual-pipeline deep learning framework for online data stream classification, Integr. Comput.-Aided Eng. 27 (2) (2020) 101–119, https://doi.org/10.3233/ICA-200617.
[34] Government of Canada, High-Resolution Solar Radiation Datasets.,https://www.nrcan.gc.ca/energy/renewable-electricity/solar.
[35] C.T.M. Clack, Modeling Solar Irradiance and Solar PV Power Output to Create a Resource Assessment Using Linear Multiple Multivariate Regression, J. Appl. Meteorol. Climatol. 56 (1) (2017) 109–125, https://doi.org/10.1175/JAMC-D-16-0175.1.
[36] M. Carranza-García, P. Lara-Benítez, J.M. Luna-Romera, J.C. Riquelme, Feature Selection on Spatio-Temporal Data for Solar Irradiance Forecasting, in: 16th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2021), Springer International Publishing, Cham, 654–664, ISBN 978-3-030-87869-6, 2022, DOI: 10.1007/978-3-030-87869-6_62.
[37] P. Lara-Benítez, M. Carranza-García, ADLStream: Asynchronous dual-pipeline deep learning framework for online data stream mining,https://github.com/pedrolarben/ADLStream, [Online; accessed 15-February-2022], 2020.
[38] S. Panigrahi, H. Behera, A study on leading machine learning techniques for high order fuzzy time series forecasting, Eng. Appl. Artif. Intell. 87 (2020), https://doi.org/10.1016/j.engappai.2019.103245, 103245, ISSN 0952-1976.
[39] J. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, A. Troncoso, Deep Learning for Time Series Forecasting: A Survey, Big Data 9 (1) (2021) 3–21, https://doi.org/10.1089/big.2020.0159.
[40] P. Lara-Benítez, M. Carranza-García, J.M. Luna-Romera, J.C. Riquelme, Temporal Convolutional Networks Applied to Energy-Related Time Series Forecasting, Appl. Sci. 10 (7), ISSN 2076–3417, DOI: 10.3390/app10072322.
[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inform. Process. Syst. 30.
[42] P. Lara-Benítez, L. Gallego-Ledesma, M. Carranza-García, J.M. Luna-Romera, Evaluation of the Transformer Architecture for Univariate Time Series Forecasting, in: Advances in Artificial Intelligence, Springer International Publishing, Cham, 106–115, ISBN 978-3-030-85713-4, 2021, DOI: 10.1007/978-3-030-85713-4_11.
[43] R.J. Williams, D. Zipser, A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, Neural Comput. 1 (2) (1989) 270–280, ISSN 0899–7667, DOI: 10.1162/neco.1989.1.2.270.
[44] J. Gama, R. Sebastião, P.P. Rodrigues, On evaluating stream learning algorithms, Mach. Learn. 90 (3) (2013) 317–346, ISSN 1573–0565, DOI: 10.1007/s10994-012-5320-9.
[45] P. Lara-Benítez, P. Reina-Jiménez, Solar Irradiance Forcasting in Streaming, https://github.com/pedrolarben/solar-radiation-forecasting-streaming, [Online; accessed 15-February-2022], 2021.