energies

MDPI

Article

2-D Convolutional Deep Neural Network for the Multivariate Prediction of Photovoltaic Time Series

Antonello Rosato [1], Rodolfo Araneo [2], Amedeo Andreotti [3], Federico Succetti [1] and Massimo Panella [1,*]

[1] Department of Information Engineering, Electronics and Telecommunications, University of Rome "La Sapienza", Via Eudossiana 18, 00184 Rome, Italy; antonello.rosato@uniroma1.it (A.R.); federico.succetti@uniroma1.it (F.S.)
[2] Electrical Engineering Division of DIAEE, University of Rome "La Sapienza", Via Eudossiana 18, 00184 Rome, Italy; rodolfo.araneo@uniroma1.it
[3] Electrical Engineering Department, University Federico II of Napoli, 80125 Napoli, Italy; andreot@unina.it
* Correspondence: massimo.panella@uniroma1.it; Tel.: +39-0644585496

Abstract: Here, we propose a new deep learning scheme to solve the energy time series prediction problem. The model implementation is based on the use of Long Short-Term Memory networks and Convolutional Neural Networks. These techniques are combined in such a fashion that interdependencies among several different time series can be exploited and used for forecasting purposes by filtering and joining their samples. The resulting learning scheme can be summarized as a superposition of network layers, resulting in a stacked deep neural architecture. We proved the accuracy and robustness of the proposed approach by testing it on real-world energy problems.

Keywords: multivariate prediction; deep learning; energy time series; convolutional neural network; long short-term memory network

Citation: Rosato, A.; Araneo, R.; Andreotti, A.; Succetti, F.; Panella, M. 2-D Convolutional Deep Neural Network for the Multivariate Prediction of Photovoltaic Time Series. *Energies* **2021**, *14*, 2392. https://doi.org/10.3390/en14092392

Received: 19 February 2021
Accepted: 20 April 2021
Published: 23 April 2021

1. Introduction

The application of novel machine learning techniques to the prediction problem is a cornerstone for improving the penetration of Distributed Energy Resources (DERs) in modern flexible smart energy systems and grids [1,2]. Forecasting is becoming a fundamental building block of any energy management approach [3,4], including generation scheduling, planning and management [5], utility side management [6] and demand side management [7]. Goals such as resource conservation, cost minimization, widespread access to energy, market and production planning, and demand management can be achieved only by endowing expert systems with advanced and improved forecasting capabilities. The development of future cooperative Virtual Power Plants (VPPs) [8] and the integration of Local Energy Markets (LEMs) in energy communities [9] will rely on good and reliable forecasts, particularly concerning the output of Renewable Energy Sources (RESs). Due to their intermittent and fluctuating nature, RESs' power output forecasting can be considered as one of the most challenging and up-to-date prediction problems in the energy framework [10,11].

For practical reasons, forecasting in the energy context is usually carried out using univariate approaches, especially regarding time series retrieved from single independent physical quantities [12–15]. While this is a valuable approach, it does not consider how other data can be incorporated in the prediction scheme, with the drawback of specifically tailoring the method to the specific time series being predicted. The most used models for this purpose are statistical regression [14,16] or computational intelligence techniques [12,17,18], based on different mathematical backgrounds. Fuzzy predictors are used in [19] to predict RESs generation and load data in a microgrid framework, with the advantage of being able to incorporate the uncertainty of the future predictions, but, at the same time, they are stuck with those predicted errors for actual energy management.

Evolutionary-genetic algorithms are used in [20] for short-term wind speed forecasting, with the strict vertical application of refining the model parameters, and do not provide a comprehensive view of the role of those techniques in the prediction realm. Artificial neural networks are used in [21] for short-term load forecasting in a microgrids environment, employing a very simple multilayer perceptron structure and basic clustering algorithms. Support Vector Machines (SVM) are used in [2] for solar radiation forecasting, in which a good review of other prediction methods can be found.

In some contexts, several different observations can be drawn from two or more quantities of a time series. These multivariate data can pertain to a particular business sector [22], e.g., price, revenue, shares [23]. Such a multivariate approach calls for more complex models to describe the relationships inherent in the time series, and, consequently, more complex predictors for the forecasting problem. In the broad energy context, multivariate prediction approaches using machine learning have been applied to load time series [24], solar irradiation [25,26], wind speed [27], and RESs [28,29]. All of these techniques have the drawback of being specifically tailored to a single optimization due to the specific nature of the time series being forecasted; in the present work, we want to contribute to the provision of a general framework for multivariate prediction optimization.

In this article, the intuition lies in how the information contained in the time series is exploited. The time series samples are combined into a data structure that resembles a sequence of video frames. This sequence, due to its structure, can be sent as input to a Convolutional Neural Network (CNN) [30]. Thus, the idea is to explore time correlations among different time series related to solar power plant production, giving an analysis of how the sequences can be combined, using observations from different physical phenomena related to the same RES quantity (in our case, the Output Power of the plant). The main contribution of this article is the description of a deep learning scheme to solve the multivariate energy time series prediction problem. In this context, the main obstacles to a satisfactory prediction result are not to be searched for in the dimensionality or the structure of the convolutional operator, but rather in the overall construction of the layers' architecture in relation with the time series' properties. The actual prediction is carried out by the Long Short-Term Memory (LSTM) network, which is a special kind of Recurrent Neural Network (RNN) able to efficiently manage long-term dependencies through the 'gates' located in each cell.

Considering the broader context of data analysis in general, and without limiting the literature to multivariate prediction problems, there are indeed several approaches relying on the combination of CNN and LSTM. These methods do not follow the multivariate paradigm but are worth mentioning because of the similarity in their network structure compared to the present work. In [31], the authors combine CNN, LSTM and Deep Neural Networks (DNNs) into one unified architecture, validating the performance on a variety of vocabulary tasks. CNNs mainly stem from image and video analysis; in fact, authors in [32,33] combine them with LSTM networks to recognize human activity in video sequences. The CNN-LSTM approach is also used in biometric applications for both recognition [34] and anti-spoofing methods [35]. Instead, regarding univariate forecasting problems, CNN and LSTM are recently combined for stock-market analysis [36] and environmental time series prediction [37]. Relative to the relation between the number of time series and the CNN's ability to process low-dimension frames, there is evidence that the convolutional approach in time series analysis performs well regardless of the absolute value of the size of the filter [38,39]. Almost counter-intuitively, it has been suggested [40] that, in certain situations, the smaller the single-layer filter size, the better the overall CNN accuracy. In fact, the main obstacles to a satisfactory prediction result are not to be looked for in the dimensionality or the structure of the convolutional operator, but in the overall construction of the layers' architecture in relation with the time series, which is, indeed, the point of this work.

To the best of our knowledge, the pure multivariate prediction approach is scarcely studied in the literature. Only recently, some approaches using a combination of CNN and

LSTM layers have been proposed, such as the one in [41]. In fact, the authors firstly use a convolutional network to extract the interrelationship of different variables at the same time step (as well as by regularizing the dimension of the input structure). Then, a dual-attentioning mechanism combined with LSTM is used. While there are some similarities in the combination of the CNN and LSTM in the deep structure, the main difference compared to our approach is that those authors apply the CNN to the input frame, constructed as a combination of different samples taken from different sequences, at the same instant in time. Instead, we not only use the CNN to extract the spatial correlations among different sequences, but also to take into account the correlation in time among those sequences, similarly to a multivariate intrinsic embedding mechanism. Indeed, we construct the input frame in a different fashion, by feeding the network with different samples taken from different sequences at several times, as described more clearly in Section 2.

In a similar fashion, there are some works which use a multivariate scheme based on LSTM regarding vertical applications in specific sectors: cybersecurity [42], time series classification [43], industrial anomaly detection [44], point process analysis [45]. By following the approach proposed in this work, it is possible to view the novel adopted scheme as a superposition of neural layers, forming a stacked deep network. Each layer feeds the output to the input of the sequent layer. This increase in the dimension of the NN towards a deeper architecture has been linked to a positive correlation with prediction accuracy [46].

The paper is organized as follows: in Section 2, the novel 2-D convolutional embedding process is explained, in Section 3, the architecture of the forecasting system is described, experiments are reported in Section 4, and conclusions are drawn in Section 5.

2. 2-D Convolutional Embedding

It is widely known that LSTM networks are a staple technique for prediction purposes. In this work, our aim is to enhance the LSTM structure to achieve a deep NN model in which feature extraction (i.e., data representation) can be obtained in a robust and automatic fashion, considering a higher abstraction level [47].

In most use cases, known, past samples of the time series subject to forecasting are input into the LSTM model, achieving an univariate prediction method. In the present article, the aim is to extend this approach to a multivariate framework: the predictor's input will be built using several time series. We will do this by employing a bidimensional CNN layer which has the role of filtering all the known samples of the available time series to acquire the expanded set of generalized features. These features are then flattened and used as a multivariate (multivalued) time series to be processed by the actual LSTM.

Let $S_1[n]$, with $n > 0$, be the scalar time series to be predicted. Let $S_m[n]$, $m = 2 \dots M$, be further $M - 1$ scalar time series correlated with $S_1[n]$, i.e., the time series $S_m[n]$ contain additional information that can be used for forecasting $S_1[n]$. Our aim is to predict the sample $S_1[n + k]$, where k represents the prediction distance, considering that all samples prior to time step n (including the latter) are known. The input frames $\mathbf{x}[n]$ are sent into the CNN layer as if they were a sequence of images, where the single frame is given by

$$\mathbf{x}[n] = \begin{bmatrix} S_1[n] & S_1[n-1] & \dots & S_1[n-D+1] \\ S_2[n] & S_2[n-1] & \dots & S_2[n-D+1] \\ \dots & \dots & \dots & \dots \\ S_M[n] & S_M[n-1] & \dots & S_M[n-D+1] \end{bmatrix}, \quad (1)$$

which is a one-channel $M \times D$ frame, where M represents the number of correlated time series and D the number of past samples (for each time series), both used to forecast $S_1[n + k]$. This operation represents the main novelty of the proposed approach. In fact, the input frame structure allows the network to efficiently recognize the temporal correlation among the considered sequences. Furthermore, in the training phase, the back-propagation learning algorithm is able to infer different temporal correlation structures through the

use of several convolutional filters of size H. The result is stored in data structures called feature maps, as illustrated in Figure 1.

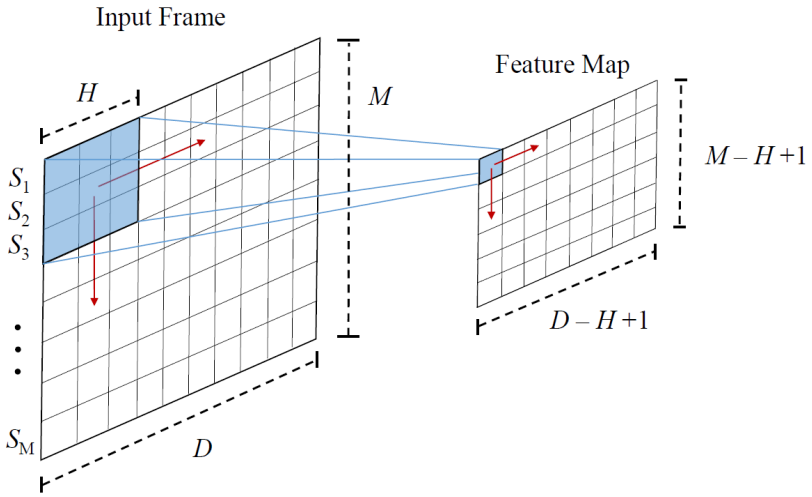

Figure 1. Application of 2-D convolutional filters to extract feature maps.

Usually, in the context of time series, forecasting is carried out using an embedding procedure as a preprocessing step in order to select the meaningful past samples of the time series through statistical or heuristic techniques [48]. In an univariate context, the estimated output is obtained as follows

$$\widetilde{S}_1[n+k] = f_u\left(S_1[n]\ S_1[n-T]\ \ldots\ S_1[n-(D-1)T]\right), \tag{2}$$

where T is called 'time lag' and D 'embedding dimension' [49].

The aforementioned inference process of the data structures in the input matrix $\mathbf{x}$ can be considered as an 'extended version' of the classic embedding. This is because, in the proposed approach, we use different time series and autonomously select the past samples. The sample $\widetilde{S}_1[n+k]$ is estimated by the deep neural network through a general non-linear, dynamical (recurrent) model $f_m(\cdot)$ as

$$\widetilde{S}_1[n+k] = f_m\left(\mathbf{x}[n]\right), \tag{3}$$

where the two embedding approaches are the same when $M = 1$ and $T = 1$.

3. Proposed Deep Neural Architecture

In this work, the backbone of an LSTM network [50] is used to predict time series data. The latter can have long term dependencies that make it difficult to locate relevant information in the data with respect to the current time step. LSTM networks solve the problem by managing the amount of information that passes in a single cell; for this reason, LSTM networks are used as an integral part of the predictive scheme. This scheme is represented in Figure 2 and its input is represented by the frame $\mathbf{x}[n]$. The layers in the architecture are:

- Sequence Input Layer: this inputs sequence data to the network by setting their dimension and building the related structures, as in (1) for $\mathbf{x}[n]$;
- 2-D Convolutional Layer: this applies F convolutional filters of dimension $H \times H$ to each input frame and outputs F feature maps with the dimension $1 \times (D - M + 1)$ (with no padding);
- Batch Normalization Layer: this normalizes the data (resulting from convolution) to reduce the sensitivity to network initialization;

- Flatten Layer: this collapses the spatial dimensions of the input to a vector time series (each sample is represented by an array of $F(D - M + 1)$ elements) to appropriately feed the LSTM layer;
- LSTM Layer: this represents the actual LSTM layer consisting of N hidden units;
- Fully Connected Layer: this is a standard feed-forward layer that connects the N LSTM hidden units to the scalar output which represents the sample to predict;
- Regression output layer: this computes the root–mean–squared-error loss for the regression problem under consideration.

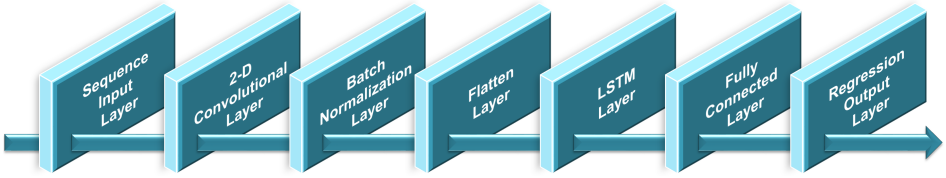

Figure 2. Architecture of the proposed deep neural network for multivariate prediction.

4. Experimental Results

We evaluate the performance of the proposed forecasting approach considering a real photovoltaic power production plant, identified geographically by the following coordinates: 39°54′38.2″ N, 105°14′04.9″ W, elevation 1855 m. The photovoltaic plant is named 'NWTC' and is located in Denver, CO, USA. Irradiance data, together with those relating to other meteorological factors, are retrieved through the Measurement and Instrumentation Data Center (MIDC) database. The output power to be predicted, which is denoted as S_1 in kW, is calculated from irradiance using a system balance of 0.9 and by applying an inverter curve with MPPT.

In addition to the sequence to be predicted, we also consider time series relative to temperatures, wind speed and wind direction, turbulence, humidity and pressure levels. Some of them are taken both at ground level (2 m above ground level) and at a higher altitude (80 m above ground level). The used sequences and their attributes are summarized in Table 1. The used sequences and their attributes are summarized in Table 1. All time series are collected in the same plant and sampled every hour (i.e., 24 samples a day). Each time series refers to the years 2017 and 2018 and is normalized between -1 and 1, for the sake of regularization, before applying the learning procedures. The extremes of normalization are given by the physical functioning of the plants (also reported in Table 1).

Table 1. Attributes of time series data.

Data	Time Series	Extremes	Unit
S_1	Output Power	{0, 100}	kW
S_2	Temperature @ 2 m	{−40, 40}	°C
S_3	Temperature @ 80 m	{−40, 40}	°C
S_4	Wind Speed @ 2 m	{0, 25}	m/s
S_5	Wind Speed @ 80 m	{0, 25}	m/s
S_6	Wind Direction @ 2 m	{0, 360}	deg
S_7	Wind Direction @ 80 m	{0, 360}	deg
S_8	Turbulence @ 2 m	{0, 1}	J/kg
S_9	Turbulence @ 80 m	{0, 1}	J/kg
S_{10}	Relative Humidity	{0, 100}	%
S_{11}	Pressure	{0, 1000}	mBar

To assess the validity of the proposed approach, we have carried out some experiments with classic and widely used benchmark architectures. In particular, we used the standard LSTM as the baseline model. Given that the goal of this work is to provide a reliable and sound solution to the multivariate prediction problem, and knowing the vast superiority of the LSTM approach in time series forecasting with respect to other algorithms (e.g.,

shallow predictors), it is sufficient to prove that our algorithm outperforms a multivariate implementation of the LSTM. Here, we will briefly indicate the methods considered for the tests:

- V-LSTM: this is the elementary multivariate implementation of the classic LSTM model, in which the multivariate sequences are fed directly to the LSTM layer, with a vector input of dimension equal to the number of considered sequences. When the number of adopted sequences is reduced to 1 (first row of the numerical results reported in the tables), this case coincides with the elementary univariate LSTM;
- 2D-CNN: this is the proposed method, where the convolutional layer is stacked before the LSTM predictor, as previously described. Similarly, for the 2D-CNN, the univariate case is considered as a benchmarking level to evaluate the multivariate performance.

A training set of 3 months (i.e., 90 consecutive days) is used for the experiments and it is associated with test sets whose lengths are 3 days (i.e., $k = 72$) and 7 days (i.e., $k = 168$) after the last available sample of the training set, respectively (the training set contains the known samples that are used to forecast the future ones). In all of the experiments, we set the embedding dimension to $D = 24$, the number of filters in the CNN layer to $F = 5$ and the number of hidden units in the LSTM layer to $N = 50$. These numbers were determined by using a grid search procedure on the training data in order to avoid overfitting. We used the ADAM algorithm [51] to train the deep neural network; the initial learning rate was set to 0.01 with 50% reduction every 30 iterations, the gradient decay factor was set to 0.9 and the maximum number of iterations to 300. All of the experiments were performed using Matlab® R2019a on a machine provided with an Intel® Core™ i7-3770K 64-bit CPU at 3.50 GHZ and with 32 GB of RAM.

For the sake of illustration, we considered several different situations to incorporate all seasonal effects in the analysis, relative to eight months of the year 2018: each test is composed of a single day, either the 15th day of the month (for March, June, September and December) or the 30th (for January, April, July and October), and by the successive 2 or 6 days after it, resulting in the 3-day and 7-day test sets introduced earlier. Ordinarily, in real-world applications, it is true that NN algorithms that are seasonally dependent must be retrained when the environmental changes affect the performance. For our solution, since training times are much smaller than the sampling intervals of the time series, the training of the network can be done at every iteration. For this reason, we care about the difference in seasonal performance and we test various times of the year, retraining the network each time, to ensure the stability and consistency of the results. In order to evaluate the possible improvements brought about by the proposed multivariate forecasting approach, we considered several options by testing different combinations of data to give an exhaustive study on how the relationship between data influences the forecasting procedure. Every network is trained on the same dataset considering 10 different runs. For each run, a different (random) initialization of the network parameters is performed. The performance is then evaluated considering the average RMSE in the 10 training sessions. The general optimization procedure is summarized in Algorithm 1.

The numerical results are reported in the Tables 2–9, which are one per considered month of 2018. This was done to explicitly evaluate the seasonality effect in different periods with different irradiation/meteorological conditions, along with the comparison between the 3-day and 7-day test sets. Additionally, as stated previously, by looking at the first rows of each table, it is possible to gain an insight into how well the univariate approach is performing; this makes the benchmarking more valuable and puts all the numerical results in the right perspective. As the reader may notice, we used quite a lot of combinations to analyze the composition of the input data, to better understand the information carried by each sequence in relation with the others. In particular, we tested the output power by itself and by coupling it with each and every other sequence. Then, we also tested the performance when constructing the input data frame using the sequences grouped in relation with their physical meaning (all the time series relative to temperature, all relative to wind, all relative to the same altitude). Specifically, we reported the RMSE

for these combinations of more than two sequences incorporated in the input data frame. A benchmarking test has also been carried out using all the considered sequences.

In the January results reported in Table 2, the RMSE is quite stable and there are no large variations in terms of performance. The univariate benchmark is satisfactory and accurate, while the multivariate approach and the proposed 2D-CNN model are better or on par. These considerations can be linked to the stability of the January set in terms of irradiation, given that the irradiation itself is quite low in winter.

Table 2. Average RMSE for January 2018 tests, carried out by using V-LSTM and 2D-CNN models and different combinations of time series.

Input Data	3-Day Test		7-Day Test	
	V-LSTM	2D-CNN	V-LSTM	2D-CNN
S_1 (univariate)	7.145	8.765	6.916	6.072
S_1, S_2	6.414	6.143	7.043	6.090
S_1, S_3	7.884	6.376	7.293	6.200
S_1, S_4	5.975	6.739	6.679	6.410
S_1, S_5	5.966	7.398	6.891	7.003
S_1, S_6	8.243	6.554	7.145	6.605
S_1, S_7	4.926	7.678	7.760	6.109
S_1, S_8	4.439	5.578	6.881	6.288
S_1, S_9	7.637	6.115	7.004	6.911
S_1, S_{10}	5.977	9.011	7.582	6.910
S_1, S_{11}	7.793	6.439	6.667	6.047
S_1, S_2, S_3	8.067	6.215	7.440	7.338
S_1, S_5, S_7, S_9	5.931	8.941	7.503	7.439
S_1, S_4, S_6, S_8	5.744	8.799	7.802	6.875
S_1, S_4, S_5, S_6, S_7	6.144	8.964	7.994	6.430
S_1, S_3, S_5, S_7, S_9	5.723	8.555	7.778	7.002
S_1, S_2, S_4, S_6, S_8	6.234	6.944	7.001	6.975
All data	6.193	7.873	8.004	5.949

For the results of mid-March, which can be seen in Table 3, similar considerations can be drawn relative to the differences between V-LSTM and 2D-CNN approaches. Additionally, the accuracy of the multivariate solution is considerably better than the univariate one.

Table 3. Average RMSE for March 2018 tests, carried out by using V-LSTM and 2D-CNN models and different combinations of time series.

Input Data	3-Day Test		7-Day Test	
	V-LSTM	2D-CNN	V-LSTM	2D-CNN
S_1 (univariate)	8.647	7.805	9.373	8.603
S_1, S_2	7.275	6.958	9.909	9.030
S_1, S_3	6.889	7.520	9.808	8.653
S_1, S_4	7.341	6.913	9.303	8.554
S_1, S_5	8.140	6.514	9.219	8.508
S_1, S_6	8.846	8.331	10.360	8.702
S_1, S_7	6.653	6.592	10.033	9.001
S_1, S_8	7.078	6.867	9.701	8.547
S_1, S_9	6.509	6.360	10.237	8.932
S_1, S_{10}	7.242	6.988	9.109	8.344
S_1, S_{11}	8.458	7.547	9.568	8.451
S_1, S_2, S_3	8.136	7.702	10.020	9.067
S_1, S_5, S_7, S_9	6.111	6.465	10.064	8.044
S_1, S_4, S_6, S_8	6.453	8.328	10.107	8.933
S_1, S_4, S_5, S_6, S_7	8.729	6.590	10.304	8.777
S_1, S_3, S_5, S_7, S_9	6.239	7.198	10.504	8.502
S_1, S_2, S_4, S_6, S_8	7.556	6.359	10.277	8.943
All data	7.935	6.527	10.963	8.033

Algorithm 1: Pseudocode of the validation scheme used for all the considered DNN architectures

Input: observed time series $\{S_1, S_2, \ldots, S_{11}\}$ of 2017 and 2018; a generic model N to be trained and optimized, liked with the architectures (i.e., either V-LSTM or 2D-CNN). For brevity, we refer solely to the multivariate case. The algorithm is the same for the univariate case, holding the same steps but only considering S_1.

1: **data preparation:** normalize time series, fill data, general cleaning, etc.
2: **experiment setup:** define a number of sets of hyperparameters of the training algorithms (i.e., regularization, learning rate, number of epochs, etc.) and the architecture (number of states, dimension of filters, etc.). Then, characterize a grid search space $\mathcal{G}$, where each point $p \in \mathcal{G}$ is a set combination taken from the parameters just described. Perform experiments with variation in input data and conditions of the tests. If we denote one of the multivariate approaches with the superscript 's', we can use superscript 'm' to denote the test month (January 2018, March 2018, etc.), and superscript 'd' to identify the chosen test set length (3 or 7 days).
3: **loop** {for each s, m, d}
4: **training/test setup:** for each sequence, organize the training and test sets, respectively identified with $\mathcal{R}^{(s,m,d)}$ and $\mathcal{T}^{(s,m,d)}$;
5: **training/validation setup:** extract a particular validation set $\mathcal{V}^{(s,m,d)}$ from $\mathcal{R}^{(s,m,d)}$ to be able to evaluate the performance during the training phase. The rest of the data $\mathcal{R}_{\text{red}}^{(s,m,d)}$ for actual model training. We obtain the validation set by extracting some samples from it. In detail, we use the sequence of the latest 3 or 7 days as the target and the preceding samples as inputs. In both test cases, the training will end one day before.
6: **end loop**
7: **loop** {for each point $p \in \mathcal{G}$}
8: **loop** {for each s, m, d}
9: **network training:** actual training of the chosen model, carried out by employing $\mathcal{R}_{\text{red}}^{(s,m,d)}$ with the set of hyperparameters chosen in p. The resulting trained network is denoted as $N_p^{(s,m,d)}$;
10: **network validation:** validate the network training by assessing the performance of $N_p^{(s,m,d)}$ using $\mathcal{V}^{(s,m,d)}$, let $e_p^{(s,m,d)}$ be the accuracy (RMSE) of the DNN.
11: **end loop**
12: **model performance:** evaluate the model performance, averaging the errors over all trained networks on the different dataset $\{s, m, d\}$, let be the mean performance indicator $\bar{e}_p$.
13: **end loop**
14: **network optimization:** let $\hat{p} = \arg\min_{p \in \mathcal{G}} \{\bar{e}_p\}$ be the resulting optimal set of hyperparameters to be chosen when doing the actual training in the next final stage.
15: **loop** {for each s, m, d}
16: **final training:** let $N_{\text{opt}}^{(s,m,d)}$ be the resulting optimal network model obtained via training with the setup $\hat{p}$ and the complete training set $\mathcal{R}^{(s,m,d)}$;
17: **final inference:** compute the final network error $\tilde{e}_{\hat{p}}^{(s,m,d)}$ for the specific test set $\mathcal{T}^{(s,m,d)}$ resulted using $N_{\text{opt}}^{(s,m,d)}$.
18: **end loop**

Output: model performance $\tilde{e}_{\hat{p}}$ for the complete set of tests (see Tables 2–9).

In the April tests shown in Table 4, since the irradiation starts to grow for the increasing number of sun hours in spring in the Boreal Hemisphere, the accuracy starts to degrade a bit, still retaining the balance in terms of RMSE that was highlighted in the previous comments. This consistency is expected and proves that our solution is robust.

Table 4. Average RMSE for April 2018 tests, carried out by using V-LSTM and 2D-CNN models and different combinations of time series.

Input Data	3-Day Test		7-Day Test	
	V-LSTM	2D-CNN	V-LSTM	2D-CNN
S_1 (univariate)	12.102	7.805	12.660	10.217
S_1, S_2	11.624	6.958	12.459	10.834
S_1, S_3	12.057	7.520	13.201	10.713
S_1, S_4	11.240	6.913	12.078	10.144
S_1, S_5	11.359	6.514	12.191	10.359
S_1, S_6	10.426	8.331	11.473	10.755
S_1, S_7	10.710	6.592	12.492	11.371
S_1, S_8	11.481	6.867	11.908	9.881
S_1, S_9	11.194	6.360	12.814	10.119
S_1, S_{10}	11.010	6.988	11.604	10.519
S_1, S_{11}	12.311	7.547	11.221	9.741
S_1, S_2, S_3	11.623	7.702	13.160	10.590
S_1, S_5, S_7, S_9	9.179	6.465	13.223	11.423
S_1, S_4, S_6, S_8	11.068	8.328	12.791	10.681
S_1, S_4, S_5, S_6, S_7	11.054	6.590	13.432	11.121
S_1, S_3, S_5, S_7, S_9	9.753	7.198	13.097	12.159
S_1, S_2, S_4, S_6, S_8	10.860	6.359	13.221	11.120
All data	13.505	6.527	13.044	9.980

The June results illustrated in Table 5 are very interesting, since they prove the vast superiority of the proposed approach with respect to both the univariate and multivariate LSTM adoption. In fact, when every time series is considered in a fully multivariate solution, the RMSE of 2D-CNN is far below almost all the results of the V-LSTM.

Table 5. Average RMSE for June 2018 tests, carried out by using V-LSTM and 2D-CNN models and different combinations of time series.

Input Data	3-Day Test		7-Day Test	
	V-LSTM	2D-CNN	V-LSTM	2D-CNN
S_1 (univariate)	18.390	10.942	16.707	16.911
S_1, S_2	11.310	10.231	19.412	9.202
S_1, S_3	10.811	10.765	19.530	10.930
S_1, S_4	10.784	9.772	17.243	12.410
S_1, S_5	11.503	9.170	17.500	16.353
S_1, S_6	18.082	10.277	16.161	14.162
S_1, S_7	21.279	10.564	15.775	20.470
S_1, S_8	15.264	9.570	18.128	19.118
S_1, S_9	15.744	10.983	17.092	17.399
S_1, S_{10}	10.030	10.978	18.110	16.710
S_1, S_{11}	21.222	11.213	20.244	14.311
S_1, S_2, S_3	12.947	10.946	19.487	16.321
S_1, S_5, S_7, S_9	17.858	11.038	15.668	19.782
S_1, S_4, S_6, S_8	11.890	10.941	16.668	13.160
S_1, S_4, S_5, S_6, S_7	16.806	10.404	16.777	20.174
S_1, S_3, S_5, S_7, S_9	12.157	9.321	16.059	19.882
S_1, S_2, S_4, S_6, S_8	10.369	9.678	18.140	14.106
All data	12.730	9.286	19.102	10.259

The representative behavior of 2D-CNN in June is confirmed by the figures shown for both the 3-day and 7-day tests: for 3-day tests, the univariate case is reported in Figure 3, the best case in Figure 4, and the multivariate case using all data in Figure 5; accordingly, for 7-day tests, the univariate case is shown in Figure 6, the best one in Figure 7, and the fully multivariate case in Figure 8. Once again, we remark that some predictions can be influenced by the meteorological variations in the training set, which can cause a worsening of the prediction for some hours continuously (e.g., the hours of the first day in Figure 5). Furthermore, it has to be noted that the 7-day test's accuracy is worse than the 3-day one, while remaining quite acceptable and stable, given the different meteorological implications of predicting longer sequences.

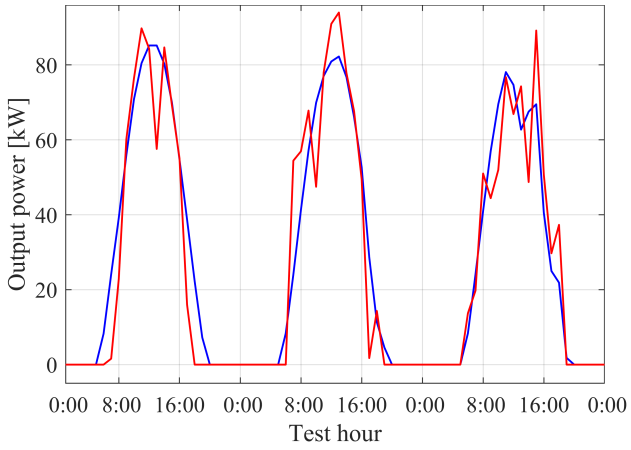

Figure 3. Predicted (red) and real (blue) value of output power in the mid of June 2018 by using the 2D-CNN approach with the univariate input on 3-day test set (see Table 5, row 1, column 2).

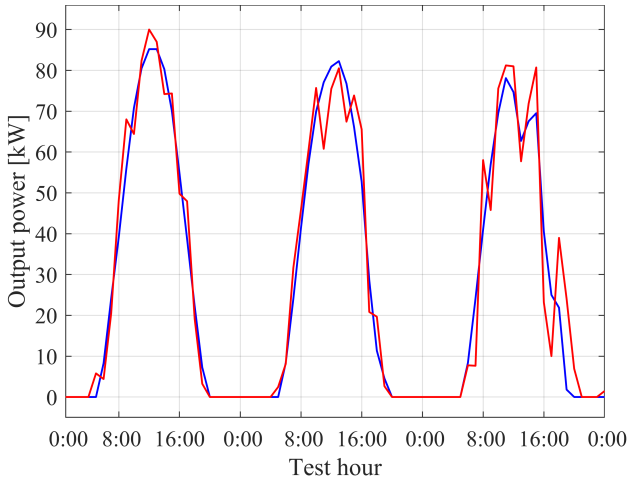

Figure 4. Predicted (red) and real (blue) value of output power in the mid of June 2018 by using the 2D-CNN approach with $\{S_1, S_5\}$ as the best combination on 3-day test set (see Table 5, row 5, column 2).

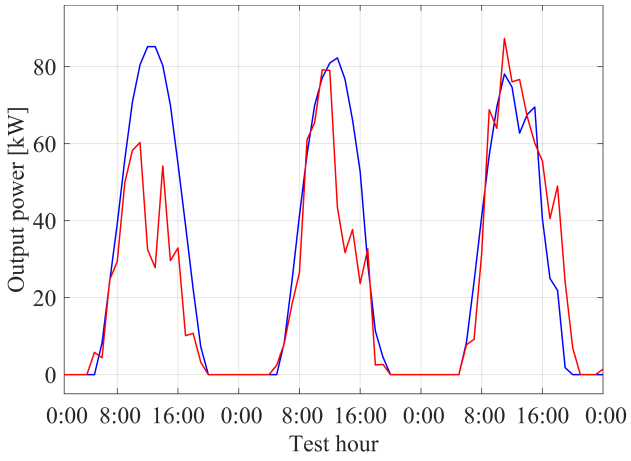

Figure 5. Predicted (red) and real (blue) value of output power in the mid of June 2018 by using the 2D-CNN approach with all the time series on 3-day test set (see Table 5, row 18, column 2).

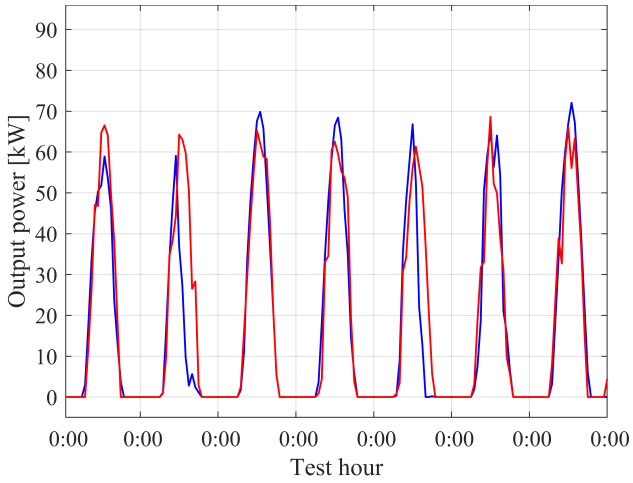

Figure 6. Predicted (red) and real (blue) value of output power in the mid of June 2018 by using the 2D-CNN approach with the univariate input on 7-day test set (see Table 5, row 1, column 4).

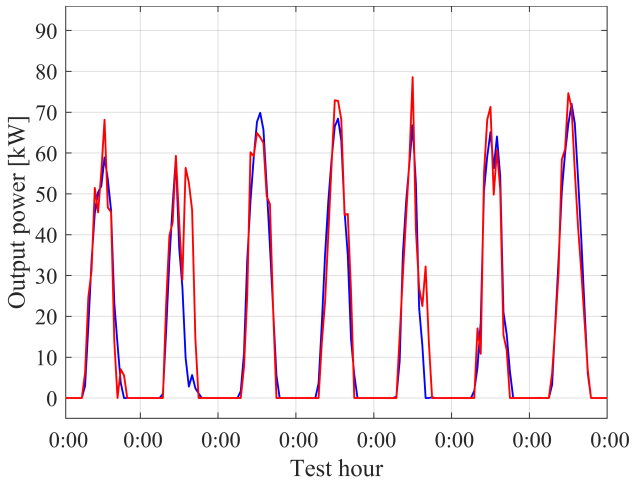

Figure 7. Predicted (red) and real (blue) value of output power in the mid of June 2018 by using the 2D-CNN approach with $\{S_1, S_2\}$ as the best combination on 7-day test set (see Table 5, row 2, column 4).

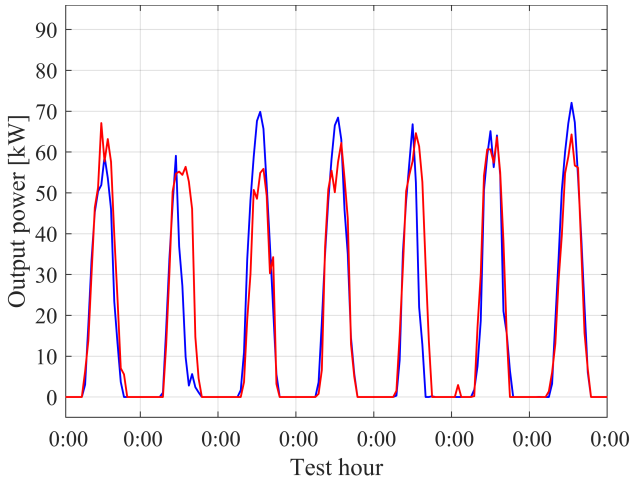

Figure 8. Predicted (red) and real (blue) value of output power in the mid of June 2018 by using the 2D-CNN approach with the combination of all the time series on 7-day test set (see Table 5, row 18, column 4).

The results for the July tests, which are reported in Table 6, show a similar behavior given the small difference in terms of irradiation and physical phenomena variations with respect to June. Additionally, the increasing stability in weather conditions gives a prominent effect that can be seen in the similarities of the RMSE values in all cases. The results for the September tests in Table 7 are definitely on par with the aforementioned cases. Here, the difference between the V-LSTM and 2D-CNN is evident, especially for the 3-day test set. The same analysis can be drawn for the October results shown in Table 8. The only noticeable difference is the general higher error due to the worsening of weather conditions, resulting in a higher variability in the associated physical quantities.

Table 6. Average RMSE for July 2018 tests, carried out by using V-LSTM and 2D-CNN models and different combinations of time series.

Input Data	**3-Day Test**		**7-Day Test**	
	V-LSTM	**2D-CNN**	**V-LSTM**	**2D-CNN**
S_1 (univariate)	12.010	12.020	15.421	14.650
S_1, S_2	12.200	12.003	15.250	14.783
S_1, S_3	12.310	11.552	15.304	14.568
S_1, S_4	12.840	11.268	15.512	15.013
S_1, S_5	12.502	11.171	15.472	15.005
S_1, S_6	12.561	12.005	17.309	14.960
S_1, S_7	17.710	12.210	18.385	14.610
S_1, S_8	11.860	11.409	15.548	14.778
S_1, S_9	12.498	11.016	15.821	14.023
S_1, S_{10}	13.610	11.890	16.140	14.577
S_1, S_{11}	11.910	9.307	17.607	14.990
S_1, S_2, S_3	12.320	11.024	15.325	14.329
S_1, S_5, S_7, S_9	15.754	11.002	18.740	14.380
S_1, S_4, S_6, S_8	11.682	11.599	16.547	15.062
S_1, S_4, S_5, S_6, S_7	13.656	12.067	18.638	14.830
S_1, S_3, S_5, S_7, S_9	13.935	11.944	18.240	14.245
S_1, S_2, S_4, S_6, S_8	11.864	10.048	16.397	14.768
All data	13.601	10.916	19.128	13.960

Table 7. Average RMSE for September 2018 tests, carried out by using V-LSTM and 2D-CNN models and different combinations of time series.

Input Data	3-Day Test		7-Day Test	
	V-LSTM	2D-CNN	V-LSTM	2D-CNN
S_1 (univariate)	10.679	9.949	10.210	10.122
S_1, S_2	11.524	6.220	10.215	9.305
S_1, S_3	11.311	6.431	10.630	8.402
S_1, S_4	10.743	5.480	10.437	8.410
S_1, S_5	10.561	6.805	10.576	8.251
S_1, S_6	11.624	8.266	10.606	9.600
S_1, S_7	12.272	9.372	10.780	10.017
S_1, S_8	10.829	8.681	10.386	9.768
S_1, S_9	10.972	7.791	10.697	9.190
S_1, S_{10}	11.602	7.310	10.130	9.351
S_1, S_{11}	12.101	7.511	10.115	8.511
S_1, S_2, S_3	12.034	7.324	11.204	10.003
S_1, S_5, S_7, S_9	11.952	7.507	11.709	8.975
S_1, S_4, S_6, S_8	11.604	7.814	10.685	8.942
S_1, S_4, S_5, S_6, S_7	12.766	7.505	11.506	9.147
S_1, S_3, S_5, S_7, S_9	12.297	7.311	11.397	10.007
S_1, S_2, S_4, S_6, S_8	12.068	6.682	10.980	9.668
All data	12.568	7.457	11.560	8.004

Table 8. Average RMSE for October 2018 tests, carried out by using V-LSTM and 2D-CNN models and different combinations of time series.

Input Data	3-Day Test		7-Day Test	
	V-LSTM	2D-CNN	V-LSTM	2D-CNN
S_1 (univariate)	14.220	13.102	12.340	12.550
S_1, S_2	13.801	12.053	12.590	12.090
S_1, S_3	13.190	12.740	12.580	11.931
S_1, S_4	13.047	12.940	12.724	11.401
S_1, S_5	13.404	12.068	12.240	11.150
S_1, S_6	13.507	11.997	14.201	11.670
S_1, S_7	13.777	13.001	14.701	11.670
S_1, S_8	14.180	12.916	12.840	11.810
S_1, S_9	13.442	12.274	12.395	11.190
S_1, S_{10}	16.500	12.422	14.205	11.310
S_1, S_{11}	14.201	11.980	13.011	11.210
S_1, S_2, S_3	12.933	12.005	12.930	11.320
S_1, S_5, S_7, S_9	13.570	12.836	13.079	11.790
S_1, S_4, S_6, S_8	13.940	11.664	12.884	11.680
S_1, S_4, S_5, S_6, S_7	13.645	11.741	13.460	11.451
S_1, S_3, S_5, S_7, S_9	13.920	10.960	14.290	11.879
S_1, S_2, S_4, S_6, S_8	13.588	10.760	14.116	11.003
All data	13.092	11.002	14.889	10.360

Lastly, some interesting remarks can be drawn from the analysis of the December results reported in Table 9. In fact, along a similar line with the analysis of the June test set, the December case proves the high stability and accuracy of our proposed method. The V-LSTM model and both the univariate methods are always worse than the proposed 2D-CNN method applied on a multivariate dataset. Furthermore, we show, for December tests, the graphical results of 2D-CNN for both the 3-day and 7-day tests. For 3-day tests, the univariate case is reported in Figure 9, while the best case coincides, in this case, with the multivariate one using all data in Figure 10; for 7-day tests, the univariate case is shown in Figure 11, and the best one is associated with the fully multivariate case in Figure 12. These figures are in accordance with the numerical results and they highlight that the

variability is scarce in all cases. While the error is still low, due to the lowest irradiance for the winter season, the accuracy of the multivariate 2D-CNN is consistently better.

Table 9. Average RMSE for December 2018 tests, carried out by using V-LSTM and 2D-CNN models and different combinations of time series.

Input Data	3-Day Test		7-Day Test	
	V-LSTM	2D-CNN	V-LSTM	2D-CNN
S_1 (univariate)	5.904	3.840	5.863	4.920
S_1, S_2	5.328	3.404	5.497	4.810
S_1, S_3	5.993	3.764	4.305	4.704
S_1, S_4	5.041	4.492	5.840	4.050
S_1, S_5	5.640	4.603	5.964	5.004
S_1, S_6	5.031	4.201	4.280	4.890
S_1, S_7	5.194	3.790	5.170	4.395
S_1, S_8	6.240	3.343	5.810	4.568
S_1, S_9	5.840	3.992	5.795	4.750
S_1, S_{10}	5.902	4.688	6.110	4.851
S_1, S_{11}	6.002	4.487	6.011	4.970
S_1, S_2, S_3	5.618	4.440	6.230	4.120
S_1, S_5, S_7, S_9	5.904	3.091	6.790	4.390
S_1, S_4, S_6, S_8	5.767	3.295	6.264	4.887
S_1, S_4, S_5, S_6, S_7	5.671	3.099	6.470	4.461
S_1, S_3, S_5, S_7, S_9	5.821	2.541	6.853	4.608
S_1, S_2, S_4, S_6, S_8	6.005	3.540	6.267	4.040
All data	5.687	2.480	7.340	4.010

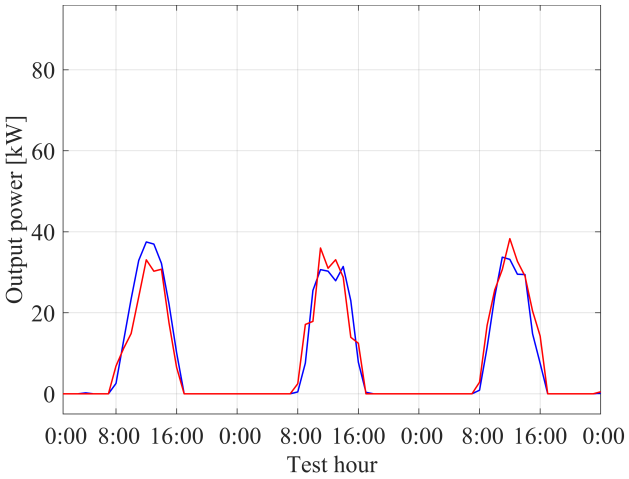

Figure 9. Predicted (red) and real (blue) value of output power in the mid of December 2018, by using the 2D-CNN approach with the univariate input on the 3-day test set (see Table 9, row 1, column 2).

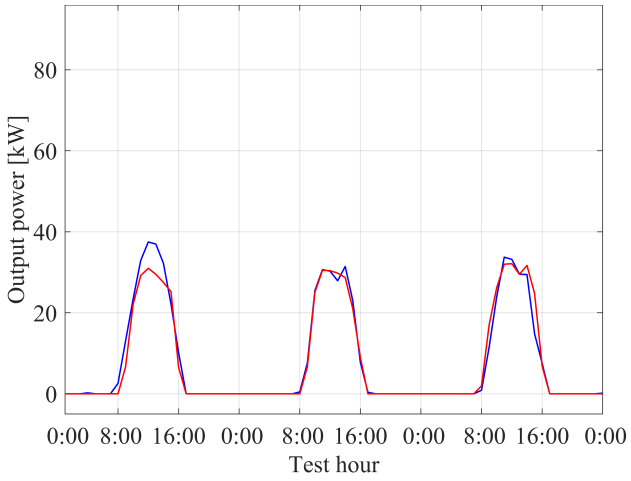

Figure 10. Predicted (red) and real (blue) value of output power in the mid of December 2018, by using the 2D-CNN approach with the combination of all the time series, which is also associated with the best performance on the 3-day test set (see Table 9, row 18, column 2).

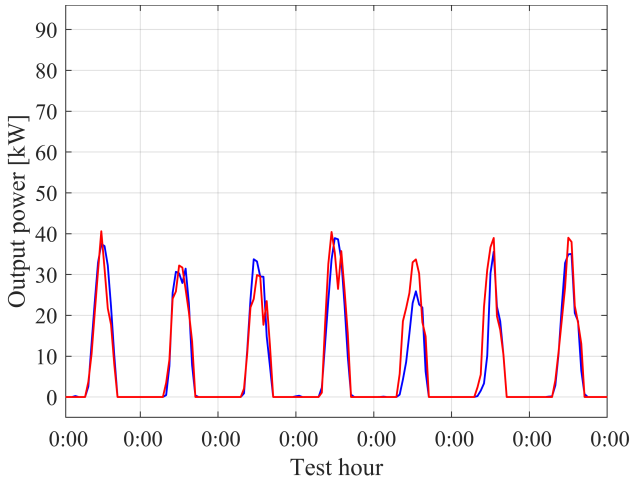

Figure 11. Predicted (red) and real (blue) value of output power in the mid of December 2018, by using the 2D-CNN approach with the univariate input on the 7-day test set (see Table 9, row 1, column 4).

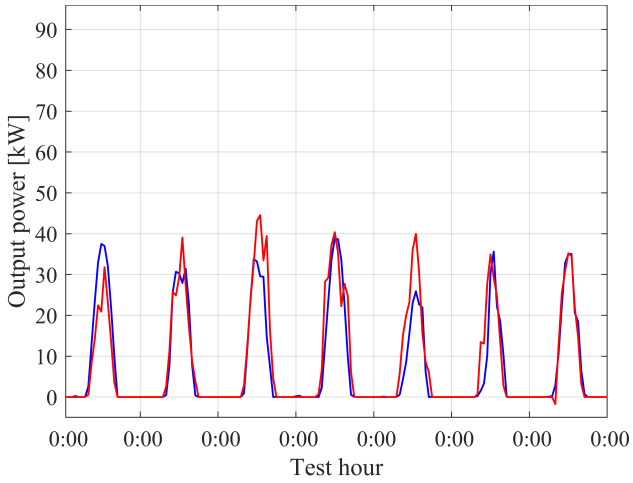

Figure 12. Predicted (red) and real (blue) value of output power in the mid of December 2018, by using the 2D-CNN approach with the combination of all the time series, which is also associated with the best performance on the 7-day test set (see Table 9, row 18, column 4).

Summarizing the results by a general outlook, the most significant improvement with respect to the univariate approach is met when considering the time series relative to wind and the ones relative to a higher altitude. Overall, this is in accordance with [27], because irradiance is strictly linked with cloud movement, which is predictable using wind information at high altitudes. In fact, as in December, the variation in weather is more frequent, the prediction performance is evidently improved by the multivariate approach. This improvement is less evident in more stable months, as in June.

As expected, the results of the 7-day case are slightly worse than the ones of the 3-day case. While this is obviously due to the longer forecasting horizon, the difference is not so large; the proposed method can be valued as stable and consistent for the tested methodologies. It is also evident that the proposed convolutive structure gives advantages over the pure LSTM, despite the low dimensionality of the frame processed by the convolutional filter. In fact, all the numerical results sport the same trend, despite the heavy changes during different seasons.

An additional remark can be drawn from the visual analysis of the results; there is quite a difference in the absolute value of the analyzed time series. This is the result of a different general dynamic in the physical system, which is quite complicated. For this matter, the complex multivariate analysis, along with the higher variability, impose that changes should be carried out in the training phase, and thus the different dynamics are reflected in the network tuning. In fact, we suggest that the latter is retrained when those drastic changes occur in actual operational applications.

5. Conclusions

In this work, we studied a new technique for multivariate forecasting of energy time series. Our contribution relies on a bidimensional convolutional layer to obtain an enriched embedding representation, feeding an LSTM model used to take advantage of long-term dependencies among the different physical time series. We carried out several tests on real world data to assess the efficient performance of the proposed model, comparing it with state-of-the-art LSTM techniques. It is possible to conclude that this novel system is a viable and robust solution for prediction applications, with the main advantage being its efficient and smart ways of exploiting access to different physical data. Future works might focus on considering architectures employing multi-stacked CNNs and tentatively applying the same concepts to distributed forecasting models.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

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

References

1. Nosratabadi, S.M.; Hooshmand, R.A.; Gholipour, E. A comprehensive review on microgrid and virtual power plant concepts employed for distributed energy resources scheduling in power systems. *Renew. Sustain. Energy Rev.* **2017**, *67*, 341–363. [CrossRef]
2. Voyant, C.; Notton, G.; Kalogirou, S.; Nivet, M.L.; Paoli, C.; Motte, F.; Fouilloy, A. Machine learning methods for solar radiation forecasting: A review. *Renew. Energy* **2017**, *105*, 569–582. [CrossRef]
3. Rosato, A.; Panella, M.; Araneo, R. A Distributed Algorithm for the Cooperative Prediction of Power Production in PV Plants. *IEEE Trans. Energy Convers.* **2019**, *34*, 497–508. [CrossRef]
4. Rosato, A.; Panella, M.; Araneo, R.; Andreotti, A. A Neural Network Based Prediction System of Distributed Generation for the Management of Microgrids. *IEEE Trans. Ind. Appl.* **2019**, *55*, 7092–7102. [CrossRef]
5. Zhang, N.; Kang, C.; Xia, Q.; Liang, J. Modeling Conditional Forecast Error for Wind Power in Generation Scheduling. *IEEE Trans. Power Syst.* **2014**, *29*, 1316–1324. [CrossRef]

6. Newsham, G.R.; Bowker, B.G. The effect of utility time-varying pricing and load control strategies on residential summer peak electricity use: A review. *Energy Policy* **2010**, *38*, 3289–3296. [CrossRef]
7. Shariatzadeh, F.; Mandal, P.; Srivastava, A.K. Demand response for sustainable energy systems: A review, application and implementation strategy. *Renew. Sustain. Energy Rev.* **2015**, *45*, 343–350. [CrossRef]
8. Kardakos, E.G.; Simoglou, C.K.; Bakirtzis, A.G. Optimal Offering Strategy of a Virtual Power Plant: A Stochastic Bi-Level Approach. *IEEE Trans. Smart Grid* **2016**, *7*, 794–806. [CrossRef]
9. Mengelkamp, E.; Gärttner, J.; Rock, K.; Kessler, S.; Orsini, L.; Weinhardt, C. Designing microgrid energy markets: A case study: The Brooklyn Microgrid. *Appl. Energy* **2018**, *210*, 870–880. [CrossRef]
10. Rosato, A.; Altilio, R.; Araneo, R.; Panella, M. Takagi-Sugeno fuzzy systems applied to voltage prediction of photovoltaic plants. In Proceedings of the 2017 IEEE International Conference on Environment and Electrical Engineering (EEEIC/I&CPS Europe), Milan, Italy, 6–9 June 2017; pp. 1–6.
11. Rosato, A.; Altilio, R.; Araneo, R.; Panella, M. A Smart Grid in Ponza Island: Battery Energy Storage Management by Echo State Neural Network. In Proceedings of the 2018 IEEE International Conference on Environment and Electrical Engineering (EEEIC/I&CPS Europe), Palermo, Italy, 12–15 June 2018; pp. 1–4.
12. Das, U.K.; Tey, K.S.; Seyedmahmoudian, M.; Mekhilef, S.; Idris, M.Y.I.; Deventer, W.V.; Horan, B.; Stojcevski, A. Forecasting of photovoltaic power generation and model optimization: A review. *Renew. Sustain. Energy Rev.* **2018**, *81*, 912–928. [CrossRef]
13. Rosato, A.; Rosa, A.; Araneo, R.; Panella, M. Prediction in Photovoltaic Power by Neural Networks. *Energies* **2017**, *10*, 1003. [CrossRef]
14. Antonanzas, J.; Osorio, N.; Escobar, R.; Urraca, R.; de Pison, F.J.M.; Antonanzas-Torres, F. Review of photovoltaic power forecasting. *Sol. Energy* **2016**, *136*, 78–111. [CrossRef]
15. Baños, R.; Manzano-Agugliaro, F.; Montoya, F.G.; Gil, C.; Alcayde, A.; Gómez, J. Optimization methods applied to renewable and sustainable energy: A review. *Renew. Sustain. Energy Rev.* **2011**, *15*, 1753–1766. [CrossRef]
16. Wang, X.; Guo, P.; Huang, X. A Review of Wind Power Forecasting Models. *Energy Procedia* **2011**, *12*, 770–778. [CrossRef]
17. Yadav, A.K.; Chandel, S.S. Solar radiation prediction using Artificial Neural Network techniques: A review. *Renew. Sustain. Energy Rev.* **2014**, *33*, 772–781. [CrossRef]
18. Agüera-Pérez, A.; Palomares-Salas, J.C.; de la Rosa, J.J.G.; Florencias-Oliveros, O. Weather forecasts for microgrid energy management: Review, discussion and recommendations. *Appl. Energy* **2018**, *228*, 265–278. [CrossRef]
19. Sáez, D.; Ávila, F.; Olivares, D.; Cañizares, C.; Marín, L. Fuzzy Prediction Interval Models for Forecasting Renewable Resources and Loads in Microgrids. *IEEE Trans. Smart Grid* **2015**, *6*, 548–556. [CrossRef]
20. Liu, D.; Niu, D.; Wang, H.; Fan, L. Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm. *Renew. Energy* **2014**, *62*, 592–597. [CrossRef]
21. Hernández, L.; Baladrón, C.; Aguiar, J.M.; Carro, B.; Sánchez-Esguevillas, A.; Lloret, J. Artificial neural networks for short-term load forecasting in microgrids environment. *Energy* **2014**, *75*, 252–264. [CrossRef]
22. Rosato, A.; Altilio, R.; Araneo, R.; Panella, M. Neural Network Approaches to Electricity Price Forecasting in Day-Ahead Markets. In Proceedings of the 2018 IEEE International Conference on Environment and Electrical Engineering, Palermo, Italy, 12–15 June 2018; pp. 1–5.
23. Ziel, F.; Weron, R. Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks. *Energy Econ.* **2018**, *70*, 396–420. [CrossRef]
24. Khan, M.; Javaid, N.; Iqbal, M.N.; Bilal, M.; Zaidi, S.F.A.; Raza, R.A. Load Prediction Based on Multivariate Time Series Forecasting for Energy Consumption and Behavioral Analytics. In *Complex, Intelligent, and Software Intensive Systems*; Barolli, L., Javaid, N., Ikeda, M., Takizawa, M., Eds.; Springer International Publishing: Cham, Switherland, 2019; pp. 305–316.
25. Sfetsos, T.; Coonick, A. Univariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques. *Sol. Energy* **2000**, *68*, 169–178. [CrossRef]
26. Voyant, C.; Muselli, M.; Paoli, C.; Nivet, M.L. Optimization of an artificial neural network dedicated to the multivariate forecasting of daily global radiation. *Energy* **2010**, *36*, 348–359. [CrossRef]
27. Cadenas, E.; Rivera, W.; Campos-Amezcua, R.; Heard, C. Wind Speed Prediction Using a Univariate ARIMA Model and a Multivariate NARX Model. *Energies* **2016**, *9*, 109. [CrossRef]
28. Rana, M.; Koprinska, I.; Agelidis, V. Univariate and multivariate methods for very short-term solar photovoltaic power forecasting. *Energy Convers. Manag.* **2016**, *121*, 380–390. [CrossRef]
29. Bai, W.; Lee, D.; Lee, K.Y. A Multivariate Time Series Forecast Model for Wind and Storage integrated system operation. In Proceedings of the 2017 IEEE Power Energy Society General Meeting, Chicago, IL, USA, 16–20 July 2017; pp. 1–5.
30. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*; Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates, Inc.: New York, NY, USA, 2012; pp. 1097–1105.
31. Sainath, T.N.; Vinyals, O.; Senior, A.; Sak, H. Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Queensland, Australia, 19–24 April 2015; pp. 4580–4584.
32. Wang, X.; Gao, L.; Song, J.; Shen, H. Beyond Frame-level CNN: Saliency-Aware 3-D CNN With LSTM for Video Action Recognition. *IEEE Signal Process. Lett.* **2017**, *24*, 510–514. [CrossRef]

33. Ullah, A.; Ahmad, J.; Muhammad, K.; Sajjad, M.; Baik, S.W. Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features. *IEEE Access* **2018**, *6*, 1155–1166. [CrossRef]
34. Zhao, J.; Mao, X.; Chen, L. Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomed. Signal Process. Control.* **2019**, *47*, 312–323.
35. Xu, Z.; Li, S.; Deng, W. Learning temporal features using LSTM-CNN architecture for face anti-spoofing. In Proceedings of the 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, 3–6 November 2015; pp. 141–145.
36. Kim, T.; Kim, H.Y. Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. *PLoS ONE* **2019**, *14*, e0212320. [CrossRef]
37. Huang, C.J.; Kuo, P.H. A Deep CNN-LSTM Model for Particulate Matter (PM2.5) Forecasting in Smart Cities. *Sensors* **2018**, *18*, 2220. [CrossRef]
38. Borjesson, L.; Singull, M. Forecasting Financial Time Series through Causal and Dilated Convolutional Neural Networks. *Entropy* **2020**, *22*, 1094. [CrossRef]
39. Succetti, F.; Rosato, A.; Araneo, R.; Panella, M. Deep Neural Networks for Multivariate Prediction of Photovoltaic Power Time Series. *IEEE Access* **2020**, *8*, 211490–211505. [CrossRef]
40. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
41. Xiao, Y.; Yin, H.; Zhang, Y.; Qi, H.; Zhang, Y.; Liu, Z. A dual-stage attention-based Conv-LSTM network for spatio-temporal correlation and multivariate time series prediction. *Int. J. Intell. Syst.* **2021**, *36*, 2036–2057. [CrossRef]
42. Filonov, P.; Lavrentyev, A.; Vorontsov, A. Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-based Predictive Data Model. *arXiv* **2016**, arXiv:1612.06676.
43. Karim, F.; Majumdar, S.; Darabi, H.; Harford, S. Multivariate LSTM-FCNs for time series classification. *Neural Netw.* **2019**, *116*, 237–245. [CrossRef]
44. Li, D.; Chen, D.; Goh, J.; kiong Ng, S. Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series. *arXiv* **2018**, arXiv:1809.04758.
45. Mei, H.; Eisner, J.M. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 6754–6764.
46. Hermans, M.; Schrauwen, B. Training and Analysing Deep Recurrent Neural Networks. In *Advances in Neural Information Processing Systems 26*; Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q., Eds.; Curran Associates, Inc.: New York, NY, USA, 2013; pp. 190–198.
47. Graves, A.; Mohamed, A.R.; Hinton, G. Speech Recognition with Deep Recurrent Neural Networks. In Proceedings of the 2013 IEEE international conference on acoustics, speech and signal processing, Vancouver, BC, Canada, 26–30 May 2013; Volume 38, pp. 6645–6649.
48. Rosato, A.; Altilio, R.; Araneo, R.; Panella, M. Embedding of time series for the prediction in photovoltaic power plants. In Proceedings of the 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), Florence, Italy, 7–10 June 2016; pp. 1–4.
49. Abarbanel, H.D.I. *Analysis of Observed Chaotic Data*; Springer: New York, NY, USA, 1996.
50. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. In Proceedings of the 1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470), Edinburgh, UK, 7–10 September 1999; Volume 2, pp. 850–855.
51. Kingma, D.P.; Ba, J.L. ADAM: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.