Deep learning neural networks for short-term photovoltaic power forecasting

A. Mellit [a, b, *], A. Massi Pavan [c], V. Lughi [c]

[a] Renewable Energy Laboratory, University of Jijel, Jijel, Algeria
[b] The International Center of Theoretical Physics (ICTP), Trieste, Italy
[c] Department of Engineering and Architecture, University of Trieste, Trieste, Italy

ARTICLE INFO

Keywords:
Microgrid
Photovoltaic power
Forecasting
Short-term
One-step
Multi-step
Deep neural networks

ABSTRACT

Accurate short-term forecasting of photovoltaic (PV) power is indispensable for controlling and designing smart energy management systems for microgrids. In this paper, different kinds of deep learning neural networks (DLNN) for short-term output PV power forecasting have been developed and compared: Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), Gated Recurrent Unit (GRU), Bidirectional GRU (BiGRU), One-Dimension Convolutional Neural Network (CNN_{1D}), as well as other hybrid configurations such as CNN_{1D}-LSTM and CNN_{1D}-GRU. A database of the PV power produced by the microgrid installed at the University of Trieste (Italy) is used to train and comparatively test the neural networks. The performance has been evaluated over four different time horizons (1 min, 5 min, 30 min and 60 min), for one-Step and multi-step ahead. The results show that the investigated DLNNs provide very good accuracy, particularly in the case of 1 min time horizon with one-step ahead (correlation coefficient is close to 1), while for the case of multi-step ahead (up to 8 steps ahead) the results are found to be acceptable (correlation coefficient ranges between 96.9% and 98%).

1. Introduction

Over the past few years photovoltaic (PV) capacity worldwide has rapidly grown. At the end of 2019 the cumulative installed capacity exceeded 600 GW [1]. The integration of renewable energy, particularly PV in Micro-Grids (MGs) is becoming increasingly popular (e.g. for supplying electricity to households, electrical vehicle charging stations, etc.).

PV output forecasting has attracted, over the last two decades, the attention of many researchers and academics, including the authors [2], and is currently one of the hottest topics in the area of renewable energy integration. Due to the intermittent nature of solar energy, forecasting of the power produced by PV arrays is a crucial task and remains a challenging issue. Accurate PV power forecasting can be beneficial for grid planning and scheduling, energy management (for example for MGs), minimizing the operational costs, safe operation, quality and for balancing supply and demand. Generally, in MGs, very short-term forecasting (up to a few minutes) is mainly used for control purposes, while short-term forecasting (up to a few hours) is generally used for scheduling the energy flow between the loads, the sources and the battery storage.

In a large portion of the literature, PV power forecasting has been considered as a regression problem (time series prediction). In the early times of PV output forecasting, several approaches were based on shallow Artificial Neural Networks (ANNs), presenting a limited number of hidden layers. Recently, with the advent of supercomputers and the availably of a large amount of data collected worldwide, researchers and academics are interested in the application of Deep Learning (DL) to improve the forecasting accuracy. The topic of PV forecasting in general has been reviewed by the authors in a recent work [3].

Machine Learning (ML) algorithms do not scale well as complexity increases exponentially with the size of the dataset. DL algorithms have shown to be very powerful tools in time series forecasting [4], including Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), One-Dimensional Convolutional Neural Networks (CNN_{1D}) and other hybrid architectures. Deep Learning Neural Networks (DLNNs) are able to automatically learn arbitrary complex mappings from inputs to outputs and support multiple inputs and outputs.

* Corresponding author. Renewable Energy Laboratory, University of Jijel, Jijel, Algeria.
E-mail address: adel_mellit@uni-jijel.dz (A. Mellit).

To date, only a few forecasting methods based on DL have been proposed for PV applications. The first application of deep learning in solar power forecasting was introduced in Ref. [5], in which the authors used encoders and LSTM. It was shown that Auto-LSTM leads to good results compared to the well-known Multilayer Perceptron network (MLP). De and coworkers presented a Recurrent Neural Network (RNN) with a LSTM training algorithm to forecast PV power [6]. They concluded that the quality of the results improves with the size of the dataset, while the number of inputs has a negligible effect.

A GRU network was used to improve the accuracy of short-time PV power output forecasting [7], demonstrating a small but sensible positive effect with respect to other algorithms such as Back-Propagation (BP), Support Vector Machine (SVM), AutoRegressive Integrated Moving Average (ARIMA) and LSTM. In Ref. [8] the authors used LSTM for forecasting PV output power. Different configurations were evaluated by varying the number of epochs, batches and inputs. The results indicated that a LSTM model with three historical inputs (time Step) provides good results. In addition, the comparison with other models without memory, such as BP, Multiple Linear Regression (MLR) and Bagged Regression tree (BR), showed that the LSTM model performs better.

Weather classification was considered in Ref. [9] to forecast the output power from a PV plant using LSTM. A one-day-ahead forecasting based on LSTM was developed, showing good results for sunny days and performing better than other investigated methods (Wavelet-NN, LS-SVM and BP). A Grey Relational Analysis (GRA) and LSTM network were developed for hour-ahead PV power generation forecasting [10]. The experimental results demonstrated that the proposed model led to the smallest forecast error when compared to other methods such as RNN, Radial Basis Function (RBF) and BP. Finally, an LSTM-based model for four seasons was developed. Generally, the model can accurately predict the 1-h ahead PV power output [11].

In [12] the authors used CNN for short-term PV forecasting and showed that sky images and PV output are crucial for obtaining high accuracy. A hybrid model based on LSTM and attention mechanism was proposed in Ref. [13] for short-term forecasting of PV power (A-LSTM); the results demonstrated that this approach leads to a better performance on 15 min predictions, compared to other NNs. In Ref. [14], hybrid configurations combining CNN and LSTM (CNN-LSTM and LSTM-CNN) were proposed, demonstrating an improvement with respect to single-architecture LSTM and CNN. A LSTM-RNN was developed for day-ahead PV power forecasting [15], demonstrating the superiority of such a DL-based approach with respect to MLP, RBF, MLR, ARMA, ARIMA, SARIMA and SVM [16,17].

Despite the efforts displayed above, a complete analysis and application of DLNN algorithms for different time horizons and steps ahead is still missing. The main objective of this work is therefore to develop and evaluate the capability of a variety of DLNN algorithms for one-Step and multi-step ahead forecasting of PV output power at different time scales.

The present work differs from the previous literature in a number of aspects: a) one-Step and multi-step PV power forecasting are presented and deeply analyzed; b) a broad range of time-scale horizons are discussed (1 min, 5 min, 30 min and 60 min); c) different DLNNs have been developed, including LSTM, GRU, BiLSTM, BiGRU, CNN-LSTM$_{1D}$ and CNN$_{1D}$-GRU, some of which have not been investigated before (e.g., BiLSTM, BiGRU and CNN$_{1D}$-GRU); d) exogenous inputs (such as air temperature, wind speed, cloudy cover, and etc.) are not considered in this work – we focus only on the historical power measurements within a MG; e) most of the models presented in the literature are focused mainly on hourly or daily one-step ahead forecasting, and to the best of our knowledge very short-term (few minutes ahead) forecasting is not well addressed, despite the fact that it plays a very important role for controlling applications of PV installations, including grid-connected MGs. We expect that this work can help researchers to acquire a clearer and more systematic picture of the applications of different types of DLNNs for PV power forecasting, the challenging issues, and the suitable configurations for real-world applications.

The paper is organized as follows: methods and materials are presented in section 2, including the system description, the database, the one-Step and multi-step ahead strategies and the examined DLNNs. Results and discussion are reported in Section 3. Section 4 provides a comparative study between one DLNN (LSTM) and two other classical neural networks (Elman neural network and Nonlinear autoregressive neural networks). Concluding remarks are given in the last section.

Fig. 1. Key components of the MG investigated in this work, including the 4 kW PV array and the data logger.

2. Materials and methods

2.1. System description

Fig. 1 shows the elements of the MG investigated in this work. It consists of a 4 kWp PV array, a 4.6 kV A inverter with storage (Li-ion battery) capacity of 10 kWh, a charging electrical station and other communication links [18]. PV array specifications are reported in Table 1.

2.2. Database and normalization

The data were collected by a *Sonnen* data logger with a 1-min time Step for the period January 1st to August 18th, 2020, for a total of 337,545 samples. An example of the monitored PV output power (*Ppv*) for a few days is plotted in Fig. 2a. Fig. 2b shows the distribution of hourly *Ppv* for one day.

Considering only daytime values (*Ppv* > 0 W) and occasional interruptions of the data logging, the database is reduced to 164,171 samples, and is divided into three parts: a subset of 70% is used for training the neural networks, 15% for validation and the last 15% for testing. First, the database is normalized using the following equation:

$$(y_N) = (y - y_{\min}) / (y_{\max} - y_{\min}) \tag{1}$$

where y_N is the normalized value of *Ppv*, $y_{\min}$ and $y_{\max}$ are min and max value of *Ppv*, respectively.

Table 1
PV module specification.

PV Module Technology	m-Si
Minimum Efficiency	16%
Nominal Power	4 kWp
Area	25 m^2
Azimuth Angle	+30°
Tilt Angle	20°
Annual Yield	4400 kWh
Number of Strings	2
Power Factor	[-0,9 + 0,9]
Minimum Efficiency	95%

2.3. PV power forecasting strategies

2.3.1. Multi-input one-output (one-step ahead)

Let x_t be the measured output PV power (samples). At time t, the one-Step ahead forecasting consists in estimating the future value of the PV output power expected at time (t+1), based on the actual and previously observed data; this can be formulated as:

$$\widehat{x}_{t+1} = f(x_t, x_{t-1}, \ldots, x_{t-k+1}) \tag{2}$$

where $t \in \{k, .., N-1\}$, $\{x_t, x_{t-1}, \ldots, x_{t-k+1}\}$ are the actual and past values of the time series, $\widehat{x}_{t+1}$ is the forecasted value, f represents the forecasting model, k is the embedded dimension of the database (time series), and N is the size of the database. For example, in the case of one Step-ahead with three inputs, the relationship between the input and the output (training matrix) can be formulated as:

$$\begin{bmatrix} \widehat{x}_4 \\ \widehat{x}_5 \\ \widehat{x}_6 \\ \widehat{x}_7 \\ \vdots \end{bmatrix} = f \begin{bmatrix} x_1 & x_2 & x_3 \\ x_2 & x_3 & x_4 \\ x_3 & x_4 & x_5 \\ x_4 & x_5 & x_6 \\ \vdots & \vdots & \vdots \end{bmatrix} \tag{3}$$

2.3.2. Multi-input multi-output (multi-step ahead)

Using a similar notation, the multi-Step ahead forecasting problem can be formulated as:

$$\left\{\widehat{x}_{t+1}, \widehat{x}_{t+2}, \ldots, \widehat{x}_{t+H}\right\} = f(x_t, x_{t-1}, \ldots, x_{t-k+1}) \tag{4}$$

where H is the forecast horizon, k is the number of samples, and $\{\widehat{x}_{t+1}, \widehat{x}_{t+2}, \ldots, \widehat{x}_{t+H}\}$ is the forecasted time series. For example, in the case of two steps-ahead ($H = 2$) with five inputs ($k = 5$), the training matrix can be expressed as:

$$\begin{bmatrix} \widehat{x}_6 & \widehat{x}_7 \\ \widehat{x}_7 & \widehat{x}_8 \\ \widehat{x}_8 & \widehat{x}_9 \\ \vdots & \vdots \end{bmatrix} = f \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ x_2 & x_3 & x_4 & x_5 & x_6 \\ x_3 & x_4 & x_5 & x_6 & x_7 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{5}$$

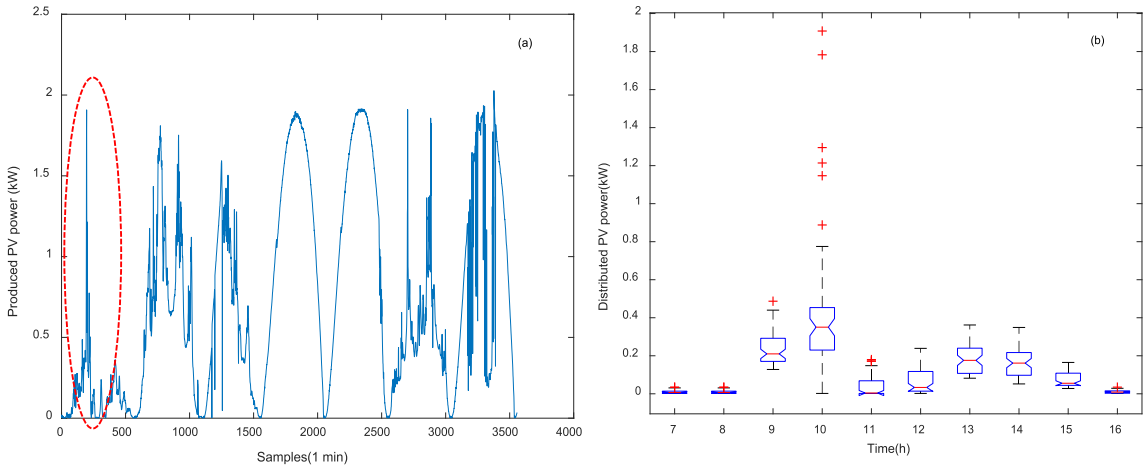

Fig. 2. a) Raw PV output power data (1–7 January 2020), b) Distribution of hourly produced PV power data (January 1, 2020).

2.4. Deep learning neural networks

DLNNs are an improvement over NNs, consisting in the addition of hidden layers – i.e. multiple processing layers to learn representations of data [19]. The main DL algorithms are [22]: DCNN, Deep Belief Networks (DBN), LSTM, Generative Adversarial Networks (GANs), Deep Convolutional GAN (DCGAN) and other hybrid combinations. A short description of the main DLNNs in PV power forecasting is given in the next subsections.

2.4.1. Long short-term memory (LSTM)

A LSTM network was first introduced in Ref. [20], and consists of a RNN modified to include a cell, an input gate, output gate and forget gate. A LSTM layer is able to learn long-term dependencies and is mainly used for time series prediction. A simple architecture consists of a set of LSTM cells and a dense output layer (See Fig. 3).

2.4.2. Bidirectional LSTM (BiLSTM)

First introduced in Ref. [21], it is a modified version of LSTM and consists of two separate hidden layers. First, the forward hidden sequence is computed, followed by the backward hidden sequence, and finally the two are combined to calculate the output (see Fig. 4).

2.4.3. Gated Recurrent Unit (GRU)

First introduced in Ref. [22], a GRU is similar to LSTM but requires a reduced number of parameters. These are learned through the gating mechanism embedded in this approach. GRU is computationally more efficient, needs less data to generalize, and can learn long-terms dependencies.

2.4.4. Bidirectional GRU (BiGRU)

BiGRU is an improved version of GRU [21]. The architecture is identical to that of a BiLSTM, in that it consists of two separate hidden layers, but fewer parameters are needed.

2.4.5. Convolutional neural network (CNN)

CNN is a regularized version of the well-known Feed-forward NNs. CNNs were firstly developed for 2D applications. It consists of a set of layers (See Fig. 5): Conv2D, Max Pooling, Flatten and Fully connected layer [19]. It can be also used for solving one dimensional problems (CNN_{1D}) such as time series classification and prediction.

2.4.6. CNN_{1D}-LSTM

CNN_{1D}-LSTM is the integration of CNN_{1D} with LSTM: the two configurations are arranged in cascade to obtain a hybrid architecture [23]. A simplified schematic of the one dimensional CNN_{1D}-LSTM used in this work is shown in Fig. 6. It comprises one convolutional, layer, one Max Pooling layer, a Flatten layer, a LSTM layer within *n* units (memory cells) and a fully dense layer (fully connected layer with one output).

2.4.7. CNN_{1D}-GRU

CNN_{1D}-GRU is a hybrid architecture combining CNN_{1D} with GRU. The architecture is similar to the one reported in Fig. 6, except for the fact that the LSTM layer is substituted by a GRU layer with *n* units.

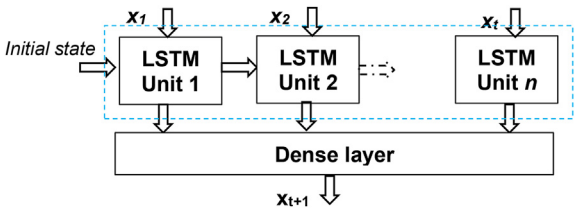

Fig. 3. The single LSTM configuration: one hidden layer with n units and one dense layer.

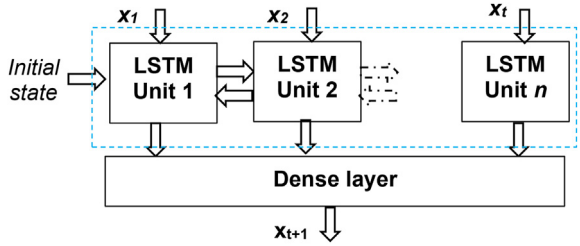

Fig. 4. The BiLSTM architecture of one hidden layer with n units and one dense layer.

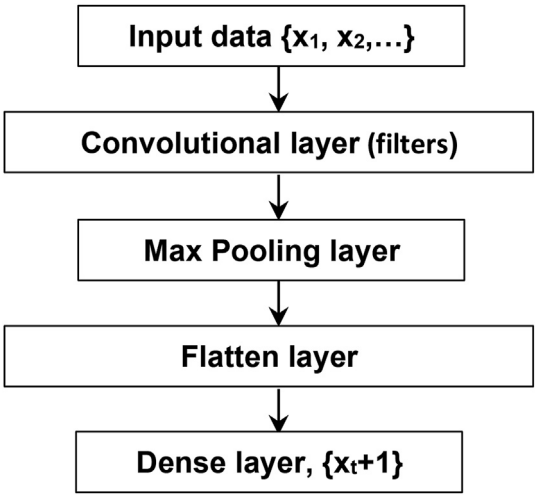

Fig. 5. CNN_{1D} architecture: one convolutional layer, max pooling layer, flatten layer, and dense layer.

2.5. Evaluation metrics and programming language

To evaluate the performance of the developed DLNN-based models, the common error metrics are used: correlation

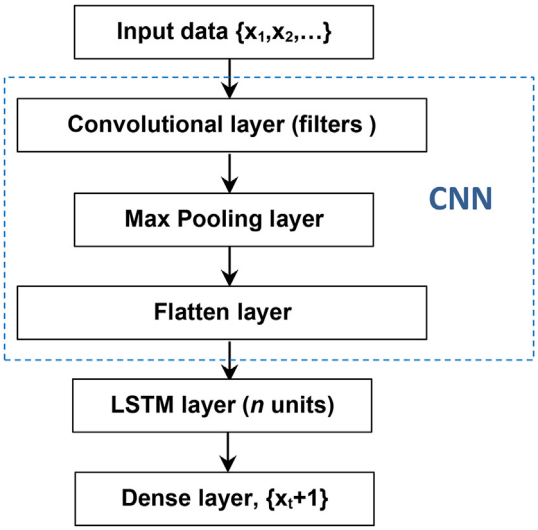

Fig. 6. Architecture of the CNN_{1D}-LSTM used in this work.

Table 2
Statistical errors for the forecasted PV powers.

Model and configuration	RMSE (kW)	MAPE (%)	r (%)	MAE (kW)
LSTM (3 × 100 × 1)	0.16	12.45	99.2	0.05
GRU (3 × 100 × 1)	0.17	8.00	99.0	0.07
BiLSTM (5 × 50 × 1)	0.17	13.96	99.0	0.07
BiGRU (5 × 50 × 1)	0.17	12.61	99.1	0.07
CNN_{1D} (7 × 100 × 1)	0.21	45.55	99.0	0.12
CNN_{1D}-LSTM (7 × 64 × 100 × 1)	0.31	127.09	98.5	0.22
CNN_{1D}-GRU (7 × 64 × 100 × 1)	0.31	139.67	98.6	0.21

coefficient (r), root mean squared error (RMSE), mean absolute error (MAE) and mean relative percent error (MAPE).

$$r = \frac{\sum_{i=1}^{n}\left((x_i - \bar{x})(y_i - \bar{y})\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i - \bar{x}_i\right)^2 \sum_{i=1}^{n}(y_i - y_i)^2}} \tag{6}$$

$$RMSE = \sqrt{\frac{1}{n}\left(\sum_{i=1}^{n}(x_i - y_i)^2\right)} \tag{7}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i| \tag{8}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{x_i - y_i}{x_i}\right| \tag{9}$$

where x_i and y_i are the measured and forecasted values, respectively, and $\bar{x}$ and $\bar{y}$ are the average values of the measured and the forecasted data, respectively.

Python language and the Keras library have been used to develop and compare the DLNNs listed above, for PV output power forecasting at different time horizons. The experiments have been conducted on a laptop i5-2540M CPU CPU @2.60 GHz, 8 GB of DDR3 RAM.

3. Results and discussion

The performance of a number of DLNNs-based models developed for the prediction of the power produced by a PV array have been evaluated for different time horizons. The models have been developed using the Adam optimizer [24] together with the ReLU (Rectified Linear Unit) [25] activation function.

With reference to the one Step-ahead prediction, the best accuracy was obtained with a number of epochs equal to 100, a batch size (BS) in the range (32–64), and an input time step in the range (3–7).

With reference to Table 2, the correlation coefficient r is in the range (98.5–99.2%) revealing a good correlation between the measured and the forecasted powers for all the tested models. LSTM, GRU, BiGRU, and BiLSTM-based models gave the best results in terms of RMSE, MAPE and MAE, while CNN together with the hybrid configurations performed worse.

Fig. 7a shows the box plots of the absolute errors for the investigated DLNN-based models. The developed LSTM and GRU-based models gave the lowest errors together with the smallest variations around the mean value. Larger variations came from the hybrid models. As an example, Fig. 7b shows the evolution of the training and validation loss functions for the developed LSTM-based model. The MSE converges after about 50 epochs to a value close to zero.

With reference to Fig. 8a, the developed LSTM and the GRU-based models gave the best results as the corresponding cumulative distribution functions (CDF) are the closest to the measured data. Moreover, Fig. 8b reveals that the correlation between the power forecasted using these two methods is very strong.

Finally, Fig. 9 shows the mean and the standard deviations between the measured and the forecasted power for the LSTM model during a period of about three days (2500 samples). The results show a very good correspondence in the case of sunny periods, and some spikes in the case of cloudy periods.

The next subsections describe five different tests that have been carried out in order to show the influence of some parameters on the forecasting accuracy and convergence time. The considered parameters are: number of units, of inputs, of epochs, of layers, of output steps, and database size.

For this purpose, only the developed LSTM-based model has been investigated because of the good accuracy and simplicity of implementation.

3.1. Test #1: different time horizons

In order to check the performance of the LSTM network developed for different time horizons, the original dataset of 164,171 1-

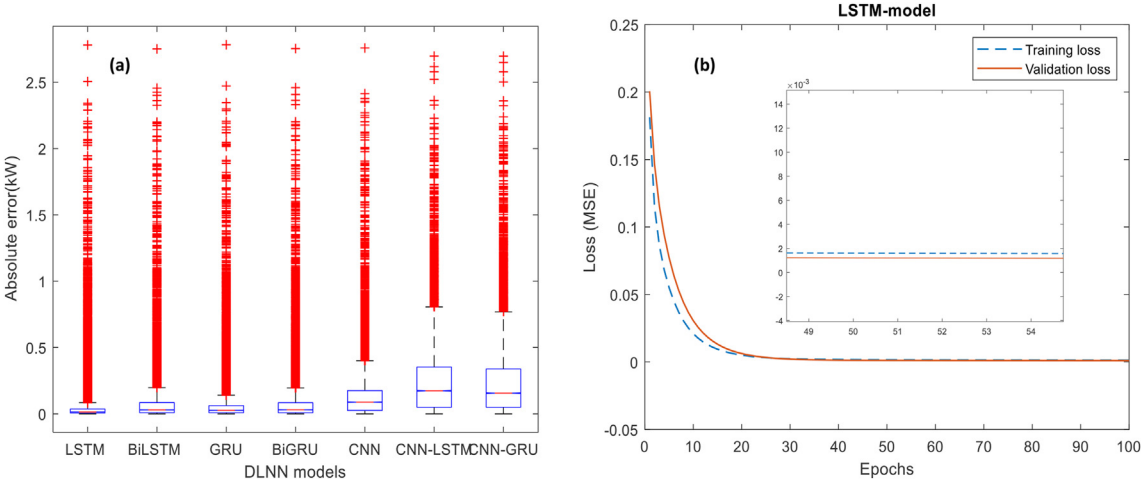

Fig. 7. a) Box plots of the absolute errors. b) Loss functions for the LSTM-based model.

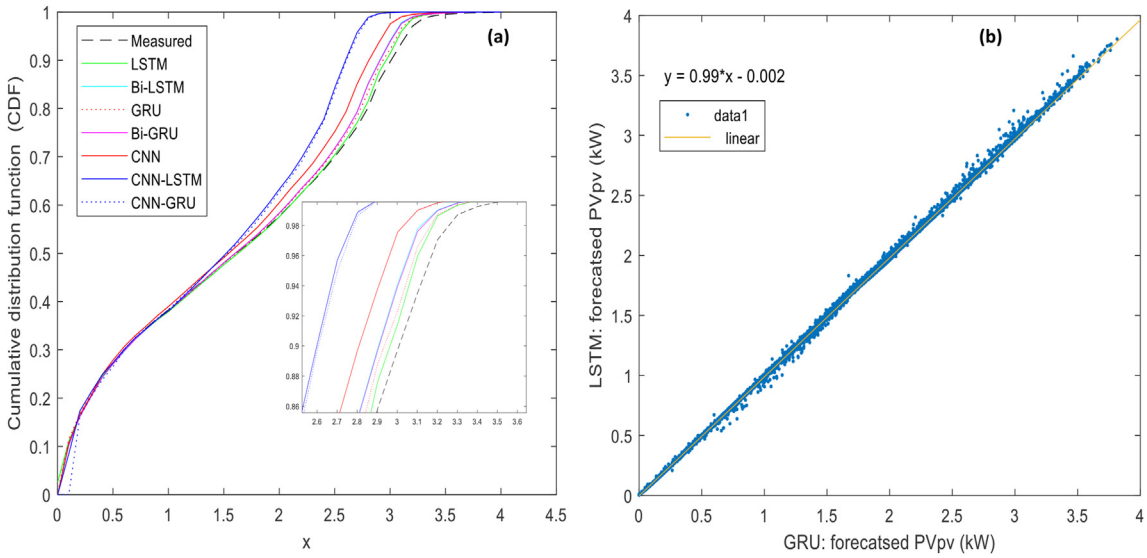

Fig. 8. a)Cumulative distribution functions. b) Correlation between the powers forecasted by the LSTM and the GRU-based models.

min samples has been divided into three databases. The first database contains 32,834 5-min samples, the second 5472 30-min samples, and the last 2736 sixty -minute samples. The results are shown in Fig. 10. The correlation coefficient, that was 99.2% in the case of 1-min samples, decreases from 97.5 to 91.3% as the duration of the sample increases.

In this case, the LSTM architecture consists of 17 inputs, 100 units, and one output. The number of epochs was 100, and the batch size was 64.

3.2. Test #2: stacked configuration

In this second test the number of hidden layers – which was one in the previous experiments – and the configuration of the LSTM architecture have been varied according to Table 3. While considering different time horizons, the correlation coefficient *r* did not change much, showing that the accuracy of the model only mildly affected by the complexity of the model. On the other hand, the new configurations require longer times for the training, making them more suitable for small databases (i.e. for longer time horizons).

3.3. Test #3: time step

In order to evaluate the effect of the input size (i.e. the number of historical power values) on the convergence time and accuracy of the LSTM forecasters, the time Step has been varied in the range (1–50) using the 5-min samples database. With reference to Fig. 11, the lowest mean absolute error corresponds to a number of inputs equal to 20, while the convergence time increases with the number

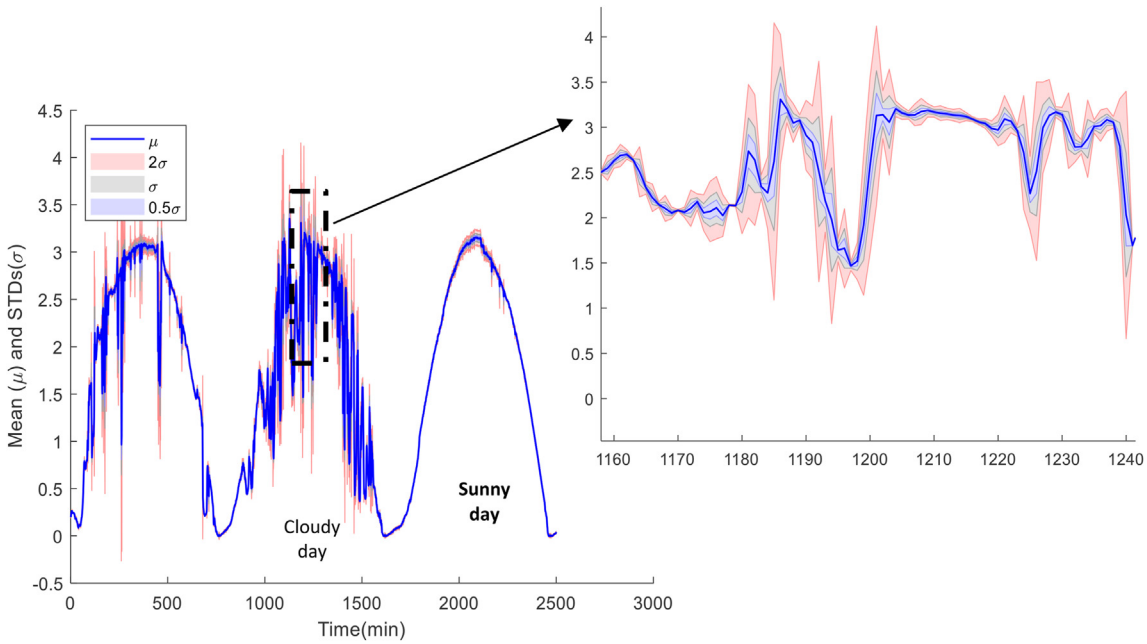

Fig. 9. Mean and standard deviations between the measured and the forecasted powers (developed LSTM model).

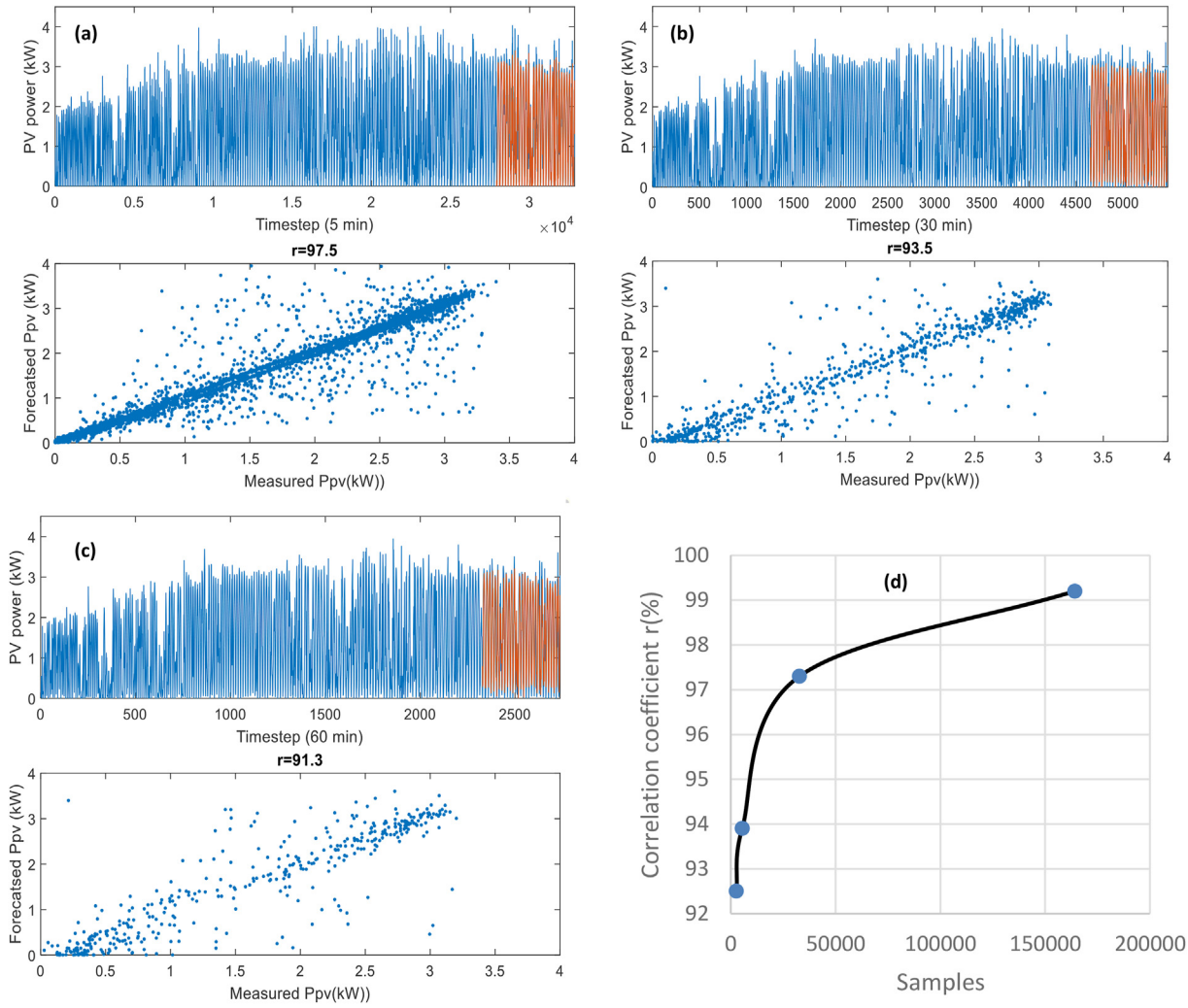

Fig. 10. Forecasted and measured power for the LSTM-based forecasters developed for different time horizons (the forecasted power is depicted in red).

Table 3
Statistical errors and training times for different LSTM configurations.

	Time horizon 5 min samples Epoch = 100, BS = 64			Time horizon 30 min samples Epoch = 150, BS = 32			Time horizon 60 min samples Epoch = 200, BS = 32		
Stacked-LSTM architecture	r (%)	MAE (kW)	Training time (s)	r (%)	MAE (kW)	Training time (s)	r (%)	MAE (kW)	Training time (s)
1st architecture ($7 \times 100 \times 50 \times 1$)	97.2	0.11	180	93.8	0.27	200	91.3	0.31	210
2nd architecture ($5 \times 100 \times 50 \times 25 \times 1$)	97.3	0.14	200	92.7	0.28	320	92.4	0.33	250
3rd architecture ($9 \times 50 \times 50 \times 1$)	97.0	0.13	190	93.9	0.29	200	92.1	0.30	230

of inputs. The correlation coefficient r is not much affected by the number of inputs and varies in the range (96%–97%).

3.4. Test #4: number of units

In this experiment, the batch size, the epochs and the number of inputs have been set to 64, 100 and 20 respectively, while the number of units has been varied in the range [0–250]. As for the previous case study, the models have been trained using the 5-min database. The results shown in Fig. 12 reveal that the best correlation coefficient corresponds to 100 units, while the convergence time increases with the number of units. Also, the mean average error is in the range [0.13–0.14 kW].

3.5. Test #5: multi-step forecasting

In this case, the LSTM architecture has been chosen to develop a number of multi-Step forecasters. Seven forecasters were trained using the 1-min database and can predict 2, 3, 4, 5, 6, 7 and 8 steps ahead. A second set of four multi-step forecasters was developed using the 60-min database; in this case, the predictions were performed on 2, 3, 4 and 5 steps ahead. The configurations of the forecasters together with the statistical errors are listed in Table 4. For both the 1-min and 60-min databases, the best MAE, RMSE and MAPE correspond to the first configuration (2-steps ahead), and the convergence time increases with the number of inputs and outputs.

With reference to the 1-min database, Fig. 13 shows as an

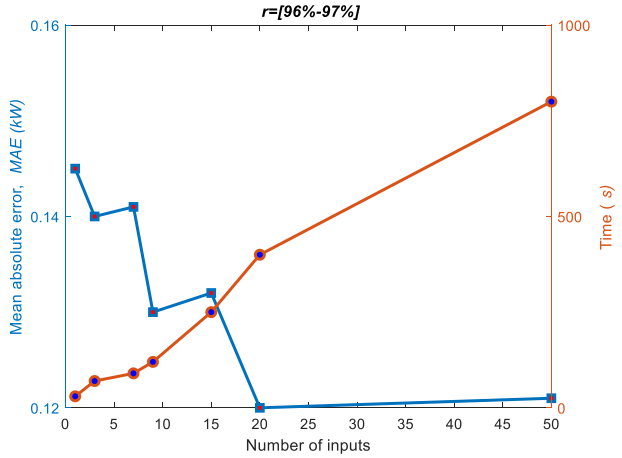

Fig. 11. Mean absolute error and training time for the LSTM-based forecasters.

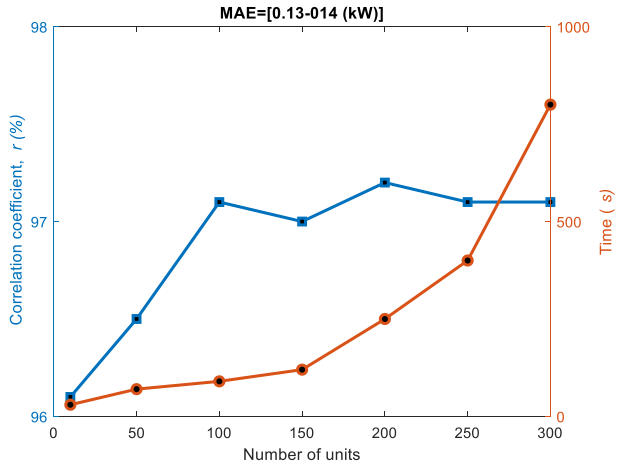

Fig. 12. Correlation coefficient and training time for the LSTM-based forecasters.

example the comparison between the measured and the forecasted power values in the case of configuration #3 (12 × 100 × 4) corresponding to a 4-steps ahead prediction. The correlation is quite good in the case of sunny days, while the forecaster performs slightly worse in the case of cloudy days.

With reference to the 60 min-database, Fig. 14 shows as an example the comparison between the measured and the forecasted power values in the case of configuration #1 (4 × 150 × 2). In this case, the forecaster performance is not satisfactory.

3.6. Test #6: uncertainty quantification

Quantification of uncertainties associated with PV power forecasts is essential for optimal management and control of PV plants. Here we use the Bootstrap Confidence Intervals (CI) [26] in order to evaluate the uncertainty of the power forecasted by the LSTM that has been developed using a 2% confidence power measurement.

The Boostrap CI procedure is summarized as follows [27]:

Step #1: Draw N samples from the original sample with replacement (N = 10,000)
Step #2: Find the median for each samples
Step #3: Arrange these sample medians in order of magnitude
Step #4: Calculate middle 95% of the medians in order to get a 95% confidence percentile

The procedure was implemented in Python using the *percentile*, *resample* and *accuracy_score* functions considering 4500 samples of the forecasted PV power. Fig. 15 shows the distribution function for 10,000 bootstrap resampling; the mean value is 1.73 kW.

Table 5 presents the calculated bootstrap CI (lower and upper intervals) at different confidence percentiles (80%, 85%, 90% and 95%)

As shown in Table 5, for all confidence percentiles the mean forecasted power (1.73 kW) is included within the confidence interval.

As an example, Fig. 16 shows the calculated uncertainty quantification interval (95% confidence percentile) for the forecasted power. The variation of the forecasted PV power never exceeds the confidence interval (shaded area).

With reference to the tests described in the previous sections, we can conclude that:

- ✓ All the investigated DLNN-based forecasters are very promising; LSTM, GRU, BiLSTM and BiGRU-based architectures work really good especially for very short-term forecasting.
- ✓ A number of input time steps in the range (3–20) is enough to give satisfactory results. However, when the number of time steps is too high the forecasters become too complicated and the training process takes too much time.
- ✓ In general, with 100 epochs the forecasters converge well. However, in the case of small databases (e.g. the 60-min

Table 4
Statistical errors and training times for different LSTM-based multi-Step forecasters.

Parameters: HL = 1, NU = 100 BS = 64, Epoch = 50 *One minute-database*	RMSE (kWh)	MAPE (%)	*r* (%)	MAE (kW)	Training time (s)
Configuration #1 (6 × 100 × 2)	0.20	21.47	98.0	0.08	390
Configuration #2 (9 × 100 × 3)	0.22	50.80	98.0	0.10	400
Configuration #3 (12 × 100 × 4)	0.23	41.94	98.0	0.09	**450**
Configuration #4 (3 × 100 × 3)	0.22	30.65	98.0	0.10	380
Configuration #5 (10 × 100 × 5)	0.22	30.65	98.0	0.10	520
Configuration #6 (12 × 100 × 6)	0.27	136.68	97.3	0.14	730
Configuration #7 (14 × 100 × 7)	0.28	128.79	97.1	0.15	850
Configuration #8 (16 × 100 × 8)	0.30	140.90	96.9	0.17	1050
Parameters: HL = 1, NU = 150 BS = 64, Epoch = 200 ***Sixty minutes database***					
Configuration #1 (4 × 150 × 2)	0.55	311.90	89.0	0.37	**295**
Configuration #2 (6 × 150 × 3)	0.58	460.32	88.6	0.41	350
Configuration #3 (8 × 150 × 4)	0.62	475.66	85.5	0.43	520
Configuration #4 (10 × 150 × 5)	0.63	537.60	84.3	0.43	700

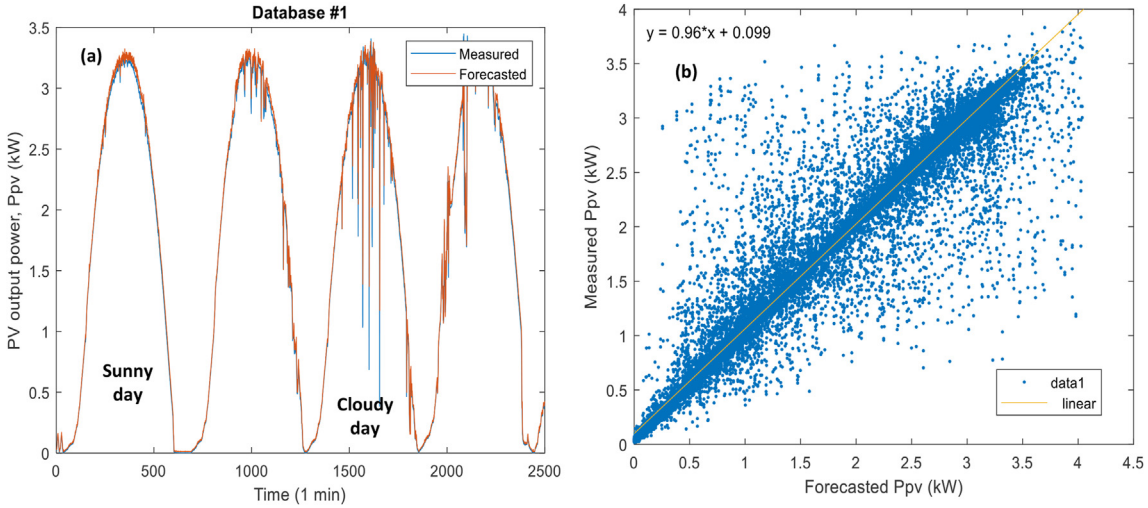

Fig. 13. LSTM-based multi-Step forecaster, configuration #3 (12 × 100 × 4). Forecasted and measured power.

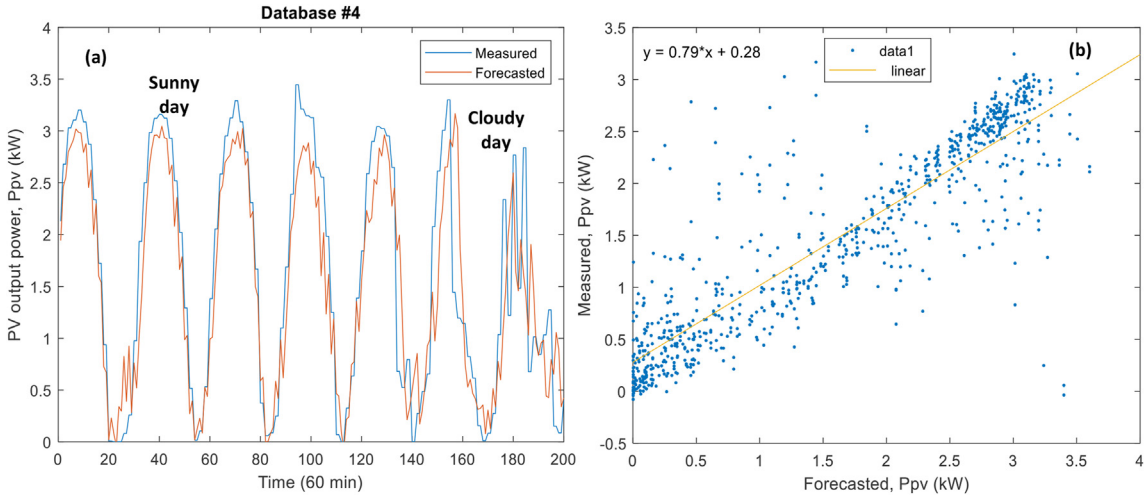

Fig. 14. LSTM-based multi-Step forecaster, configuration #1 (4 × 150 × 2). Forecasted and measured power.

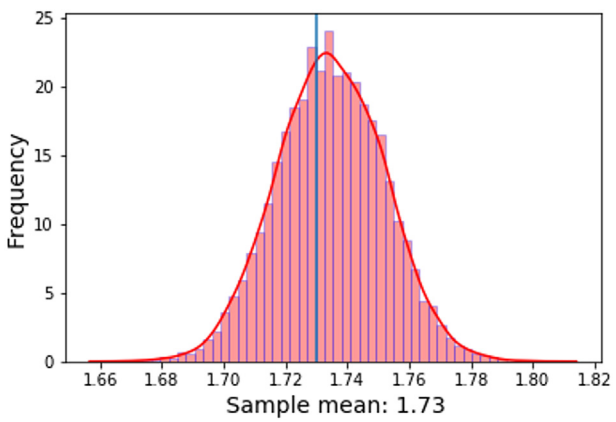

Fig. 15. The distribution function of the sample means.

database), a higher number of epochs can improve the accuracy of the forecasters.

- ✓ Increasing the batch size can significantly reduce the time needed for the training process. Conversely, the accuracy is negligibly affected by the batch size.
- ✓ The use of complicated architectures with many hidden layers can increase the accuracy, although only moderately. The use of many hidden layers is not recommended in the case of large databases.
- ✓ The forecasters perform better in the case of small time horizons (1 min). Moreover, the use of larger databases can significantly increase the accuracy.
- ✓ In the case of one-Step ahead forecasting, the use of large databases gives satisfactory results even with the simplest DLNN architectures. In the case of multi-step ahead forecasting, simple LSTM architectures provide acceptable results for up to 8 steps ahead only.
- ✓ In the case of particular configurations such as multi-input/output, complicated architectures, large databases or high number of layers, the running time can take more than 1 h on an average desktop workstation.

Table 5
CI for different confidence percentiles.

Confidence percentiles	Bootstrap CI		Mean (kW)	Standard error $\frac{\sigma}{\sqrt{n}}$
	Lower (kW)	Upper (kW)	1.73	0.0172
80%	1.724	1.756		
85%	1.710	1.760		
90%	1.706	1.762		
95%	1.701	1.768		

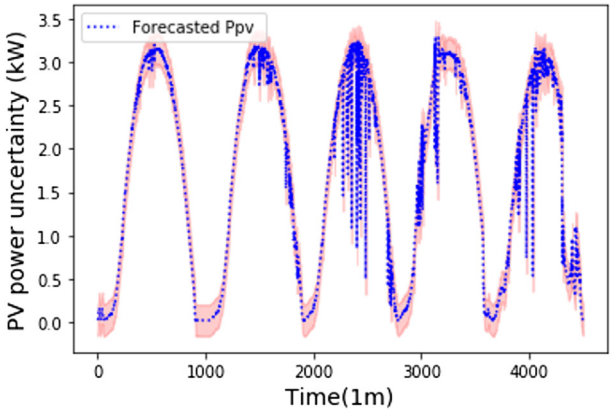

Fig. 16. Uncertainty quantification of the forecasted power.

- ✓ The performance of the considered DLNN-based forecasters is highly dependent on the weather conditions: the errors are higher in the case of cloudy days.
- ✓ The LSTM model performs well for different confidence percentiles (80%, 85%, 90% and 95%), and the uncertainty quantification determines the accuracy of thepredicted values.

4. Comparative study

This section aims at comparing one LSTM-based forecaster with two classical time series prediction algorithms: the first is a nonlinear autoregressive neural network (NAR) and the second an Elman recurrent neural network (ENN). These types of neural networks have been implemented in many programming languages including Matlab and Python, so they can be easily used to forecast the output PV power.

In order to compare the performance of the LSTM, ENN and NAR networks, a number of experiments have been carried out considering different architectures (i.e. time steps, number of units, hidden layers, activation functions and training algorithms) and using the first dataset of 337,545 samples.

Fig. 17 shows the measured power together with the one predicted by the different techniques, and Fig. 18 depicts the correlation between measured and forecasted power values. The trends shown in the right part of the plot are similar, while a small difference with the measured data can be observed in the left part of the plot. The correlation is good, being in the range (97%–99%) as shown in Fig. 18.

From a quantitative point of view, the error metrics are listed in Table 6.

Table 5 shows that the LSTM-based model performs better than ENN and NAR neural networks in terms of both accuracy and convergence time. This is mainly due to the architecture of the LSTM network (forget gate) [20] and to the functions used in this case, i.e the optimizer '*Adam*' and the activation function '*ReLU*'. In fact, the Adam optimizer [24] outperforms both the Levenberg-Marquardt algorithm (used in the NAR neural network), and the Gradient descent with momentum and adaptive learning (used in the ELN network). Moreover, the activation function '*ReLU*' is faster [28] than other activation functions (*Tansig* and *Logsig*).

Finally, the Dropout layer [29] used in the LSTM network helps preventing overfitting.

Furthermore, from the point of view of implementation and computation complexity the compared neural networks are all simple, but LSTM can support a larger database than other classical

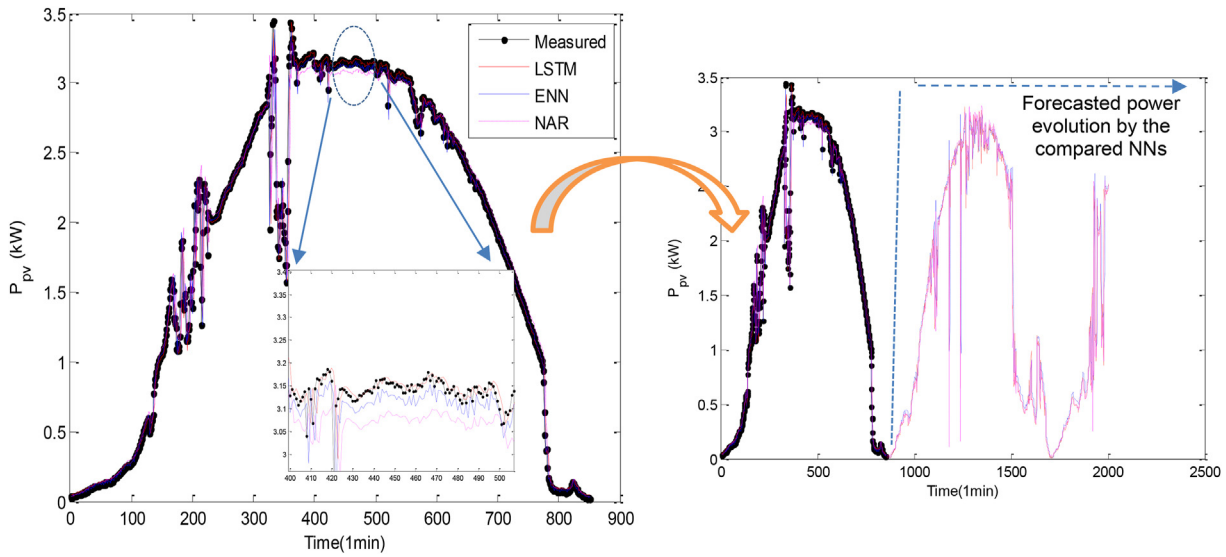

Fig. 17. Measured versus forecasted powers - one day (850 samples).

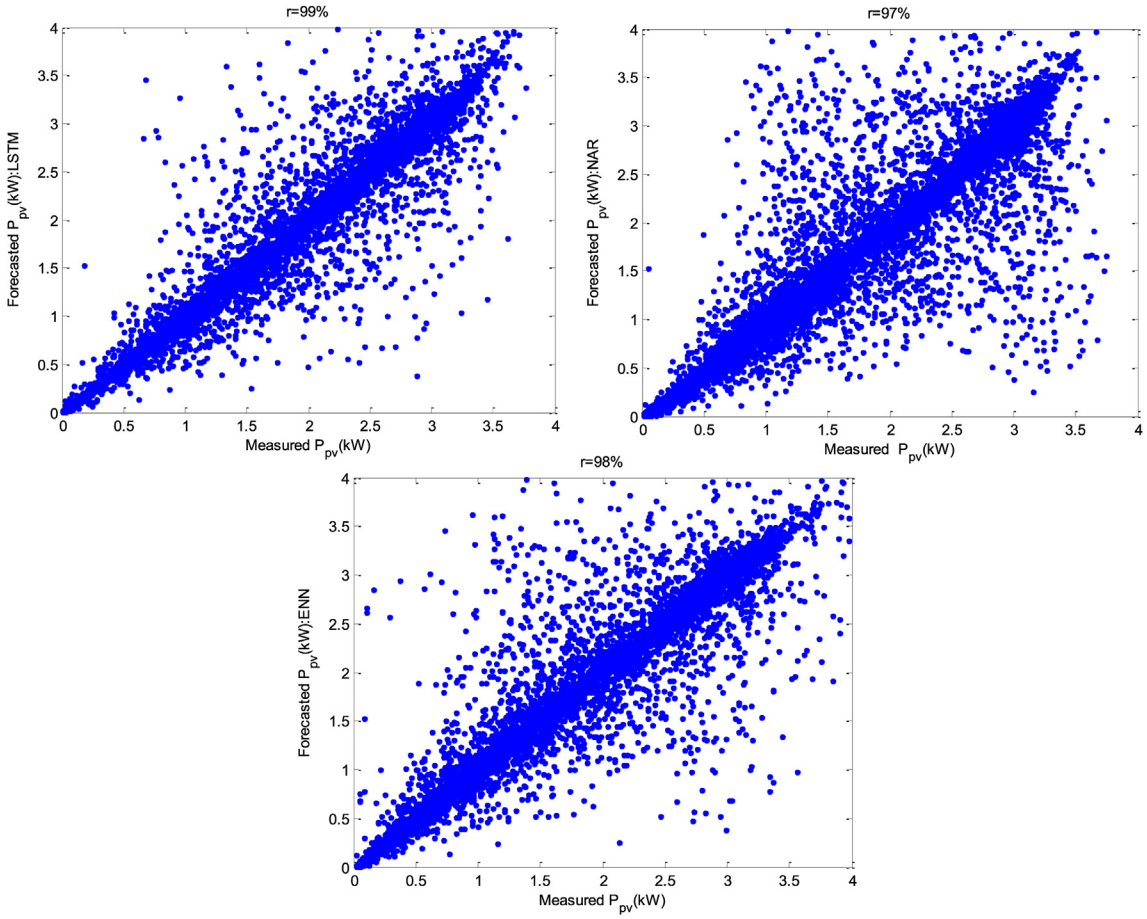

Fig. 18. Correlations between measured and forecasted powers.

Table 6
Error metrics for the considered models.

Model	*r* (%)	MAE (kW)	RMSE (kW)	Training Time (s)	Computation complexity
LSTM time steps = 5 NU = 100 BS = 64 Training function = *Adam* Loss = *MSE* Activation function = *ReLu*	99	0.054	0.16	160 s	low
ENN time steps = 5 NU = 27 Activation function: *Tansig* Training function: *Traingdx* Loss: *MSE*	98	0.067	0.21	225 s	low
NAR time steps = 5 NU = 45 Activation function: *Logsig* Feedback delays = *5* Training function: *Trainlm* Loss: *MSE* Feedback delays = 3	97	0.10	0.25	328 s	low

ML algorithms [30]. These results are not surprising if we consider the fact that LSTM neural networks were designed in order to overcome some of the drawbacks of classical RNNs such as long-term dependencies [31] and the exploding gradient (vanishing problem) [20].

5. Conclusions

In this paper, a variety of DLNNs has been developed for one-Step and multi-step ahead forecasting of PV output power, over different time horizons (1 min, 5 min, 30 min and 60 min). It has

been demonstrated that a simple DLNN architecture (such as LSTM or GRU) can provide a very good accuracy ($r = 99\%$) for one step-ahead forecasting. Good results are also obtained for multi-step ahead forecasting ($r = 96.9\%$, e.g., for 8 steps ahead).

It should be pointed out that parameters such as batch size, number of units, number of hidden layers, filters size, dropout, and kernel size, should be also carefully selected; the value of these parameters differs from one model to another. It has been verified that a large database is necessary to achieve good results.

A comparative study confirmed the effectiveness of DLNNs (e.g. LSTM) with respect to the traditional neural networks (such as ELN and NAR). The bootstrap CI is used to quantify the uncertainty of the forecasted PV power by the LSTM model. The model exhibits good accuracy for different confidence percentiles.

The new advanced DLNN algorithms lead to acceptable accuracy in the case of cloudy days, however further improvements are needed for a fully satisfactory planning and management of energy systems that include PV sources. Strategies for further enhancing the forecasting accuracy in the case of cloudy days should account for a combination of weather forecast data, sky images, clearness index, etc.

Multi-Step forecasting of PV power also remains an open challenge. Strategies for further improving forecasting accuracy in this case include testing other DLNNs (e.g., Seq2Seq learning) or developing more advanced algorithms.

The performance of DLNNs tested in this work, however, are satisfactory for what concerns the design of a smart energy management system for a microgrid that includes a PV generator, an electrical storage, and an electrical vehicle charging station – such as the one that has been used in this work for generating the database for training and testing the neural networks. In general, we expect rather simple DLNN architectures to be sufficient for most real-world applications. Furthermore, other prediction interval methods will be considered in the future for an in-depth analysis of the uncertainty associated with the produced solar PV power. The techniques illustrated here for short-term forecasting have the potential for being further adapted for medium- and long-term forecasting as well.

CRediT authorship contribution statement

A. Mellit: Methodology, Writing – original draft, code development, preparation. **A. Massi Pavan:** Making measurement, preparing databases, Formal analysis, Investigation. **V. Lughi:** Conceptualization, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Dr. Vanni Lughi acknowledges the financial support provided by "MUSE – Cross-border collaboration for a sustainable and energetically efficient university mobility", a project co-financed by the European Regional Development Fund (ERDF) via the cross-border cooperation program Interreg Italy-Slovenia.Dr. A. Massi Pavan acknowledges the financial support provided by "DEEP-SEA – Development of energy efficiency planning and services for the mobility of Adriatic Marinas", a project co-financed by the European Regional Development Fund (ERDF) via the cross-border cooperation programme Interreg Italy-Croatia.

References

[1] IEA, Snapshot of Global Photovoltaic Markets, (Accessed April 2020).
[2] A. Mellit, A. Massi Pavan, V. Lughi, Short-term forecasting of power production in a large-scale photovoltaic plant, Sol. Energy 105 (2014) 401–413, https://doi.org/10.1016/j.solener.2014.03.018.
[3] A. Mellit, A. Massi Pavan, E. Ogliari, S. Leva, V. Lughi, Advanced methods for photovoltaic output power forecasting: a Review, Appl. Sci. 10 (2020) 487, https://doi.org/10.3390/app10020487.
[4] B. Jason, Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python, Machine Learning Mastery, 2020, pp. 42–52.
[5] A. Gensler, J. Henze B. Sick , N. Raabe . Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks. IEEE international conference on systems, man, and cybernetics (SMC), October 2016, https://doi.org/10.1109/SMC.2016.7844673.
[6] V. De, T.T. Teo, W.L. Woo, T. Logenthiran, Photovoltaic power forecasting using LSTM on limited dataset. IEEE innovative smart grid technologies-Asia (ISGT Asia), May 2018, https://doi.org/10.1109/ISGT-Asia.2018.8467934, 2018.
[7] W. Yusen, W. Liao, Y. Chang, Gated recurrent unit network-based short-term photovoltaic forecasting, Energies 11 (2018) 2163, https://doi.org/10.3390/en11082163.
[8] M. Abdel-Nasser, M. Karar, Accurate photovoltaic power forecasting models using deep LSTM-RNN, Neural Comput. Appl. 31 (2019) 2727–2740, https://doi.org/10.1007/s00521-017-3225-z.
[9] M. Gaoa, J. Lia, F. Honga, D. Longb, Day-ahead power forecasting in a large-scale photovoltaic plant based on weather classification using LSTM, Energy 187 (2019), https://doi.org/10.1016/j.energy.2019.07.168, 115838.
[10] B. Chen, P. Lin, Y. Lin3, Y. Lai, S. Cheng, Z. Chen, L. Wu, Hour-ahead photovoltaic power forecast using a hybrid GRA-LSTM model based on multivariate meteorological factors and historical power datasets, IOP Conf. Ser. Earth Environ. Sci. 431 (2020), https://doi.org/10.1088/1755-1315/431/1/012059, 1012059.
[11] M. Gao, J. Li, F. Hong, D. Long, Short-term forecasting of power production in a large-scale photovoltaic plant based on, LSTM. Appli. Sciences 9 (2019) 3192, https://doi.org/10.3390/app9153192.
[12] Y. Yuchi, V. Venugopal, A. Brandt, Short-term solar power forecast with deep learning: exploring optimal input and output configuration, Sol. Energy 188 (2019) 730–741, https://doi.org/10.1016/j.solener.2019.06.041.
[13] H. Zhou, Y. Zhang, L. Yang, Q. Liu, K. Yan, Y. Du, Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism, IEEE Access 7 (2019) 78063–78074, https://doi.org/10.1109/ACCESS.2019.2923006.
[14] W. Kejun, X. Qi, H. Liu, Photovoltaic power forecasting based LSTM-Convolutional Network, Energy 189 (2019), https://doi.org/10.1016/j.energy.2019.116225, 116225.
[15] F. Wang, Z. Xuan, Z. Zhen, K. Li, T. Wang, M. Shi, A day-ahead PV power forecasting method based on LSTM-RNN model and time correlation modification under partial daily pattern prediction framework, Energy Convers. Manag. 212 (2020), https://doi.org/10.1016/j.enconman.2020.112766, 112766.
[16] M.A.F.B. Limaa, P.C.M. Carvalho, L.M. Fernández-Ramírez, A.P.S. Braga, Improving solar forecasting using deep learning and portfolio theory integration, Energy 195 (2020), https://doi.org/10.1016/j.energy.2020.117016, 117016.
[17] H. Sharadga, Hussein, S. Hajimirza, R.S. Balog, Time series forecasting of solar power generation for large-scale photovoltaic plants, Renew. Energy 150 (2020) 797–807, https://doi.org/10.1016/j.renene.2019.12.131.
[18] A. Massi Pavan, V. Lughi, M. Scorrano, Total Cost of Ownership of Electric Vehicles Using Energy from a Renewable-Based Microgrid, IEEE Power Tech, Milan, June 2019, https://doi.org/10.1109/PTC.2019.8810736.
[19] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444, https://doi.org/10.1038/nature14539.
[20] A. Hochreiter, Sepp, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.
[21] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (1997) 2673–2681, https://doi.org/10.1109/78.650093.
[22] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv:1406.1078 (2014), https://arxiv.org/abs/1406.1078, 2014.
[23] N. Xue, I. Triguero, G.P. Figueredo, D. Landa-Silva, Evolving deep CNN-LSTMs for inventory time series prediction, IEEE congress on evolutionary computation (CEC). https://doi.org/10.1109/CEC.2019.8789957, June 2019.
[24] D.P. Kingma, J. Ba Adam, A method for stochastic optimization, arXiv: 1412.6980, https://arxiv.org/abs/1412.6980, 2014.
[25] R. Hahnloser, H.S. Seung, Permitted and forbidden sets in symmetric

threshold-linear networks, Adv. Neural Inf. Process. Syst. 15 (2001) 217–223, https://doi.org/10.1162/089976603321192103.
[26] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman and Hall, New York, NY, 1993, https://doi.org/10.1007/978-1-4899-4541-9.
[27] D. Wackerly, W. Mendenhall, R.L. Scheaffer, Mathematical Statistics with Applications, Cengage Learning, 2014.
[28] A.F. Agarap, Deep Learning Using Rectified Linear Units (Relu), 2018 arXiv:1803.08375.
[29] B. Lengerich, E.P. Xing, R. Caruana, On Dropout, Overfitting, and Interaction Effects in Deep Neural Networks, 2020 arXiv:2007.00823.
[30] A. Mellit, An Overview on the Application of Machine Learning and Deep Learning for Photovoltaic Output Power Forecasting. In International Conference on Electronic Engineering and Renewable Energy, Springer, Singapore, 2020, pp. 55–68. https://link.springer.com/chapter/10.1007/978-981-15-6259-4_4.
[31] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Network. 5 (2) (1994) 157–166, https://doi.org/10.1109/72.279181.