D.2.3.1 EU printed publications

Units was involved in the preparation and publication of a number of publications regarding the optimization of the energy flows within a photovoltaic based charging station for nautical applications. The optimization was performed to minimize the cost of the charged electricity as well as the related CO2 emissions.
In particular, the studies included the forecasting of the power produced by photovoltaic generators, the prediction of the grid voltage, fault diagnosis methods for photovoltaic systems, and the development of a real time energy management systems for e-vehicles charging stations.

Renewable Energy 172 (2021) 276–288

Contents lists available at ScienceDirect

Renewable Energy

journal homepage: www.elsevier.com/locate/renene

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

Article history:
Received 2 October 2020
Received in revised form
3 February 2021
Accepted 27 February 2021
Available online 4 March 2021

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).

https://doi.org/10.1016/j.renene.2021.02.166
0960-1481/

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:

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

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, $\hat{x}_{t+1}$ 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:

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

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

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

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

where H is the forecast horizon, k is the number of samples, and $\{\hat{x}_{t+1}, \hat{x}_{t+2}, \ldots, \hat{x}_{t+H}\}$ 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:

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

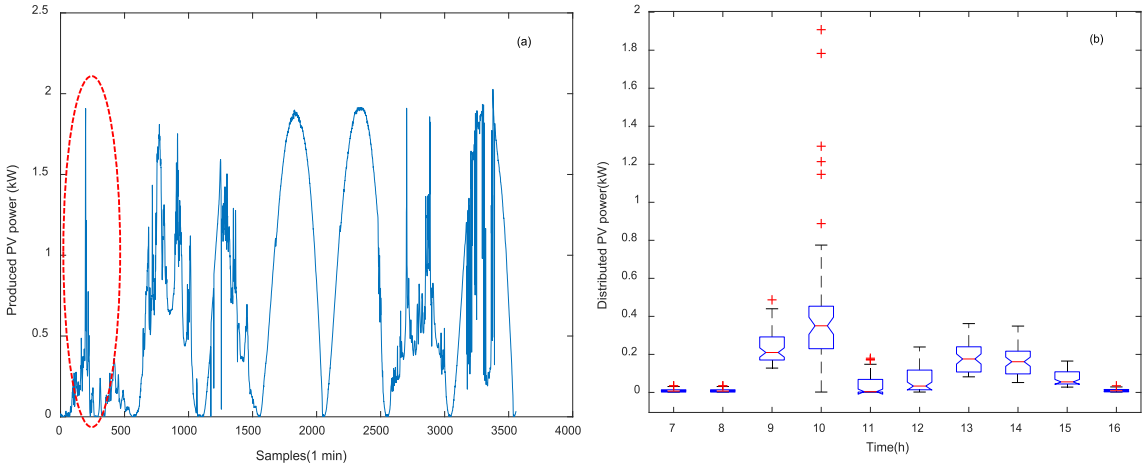


Fig. 2. a) Raw PV output power data (1–7 January 2020), b) Distribution of hourly produced PV power data (January 1, 2020).

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.

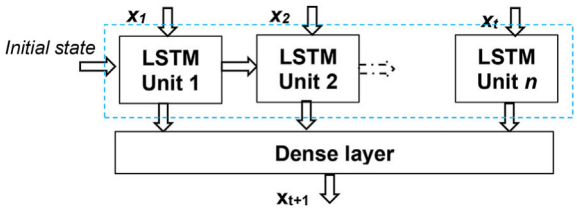


Fig. 3. The single LSTM configuration: one hidden layer with n units and one dense layer.

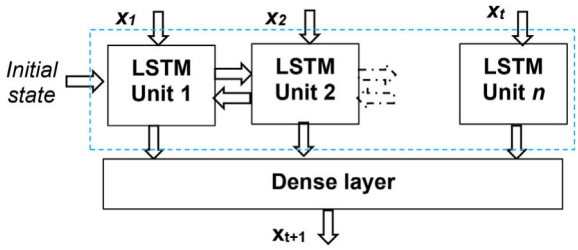


Fig. 4. The BiLSTM architecture of one hidden layer with n units and one dense layer.

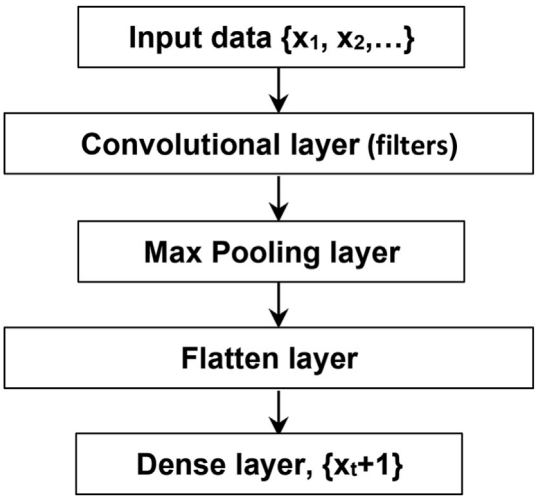


Fig. 5. CNN_{1D} architecture: one convolutional layer, max pooling layer, flatten layer, and dense layer.

2.5. Evaluation metrics and programming language

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

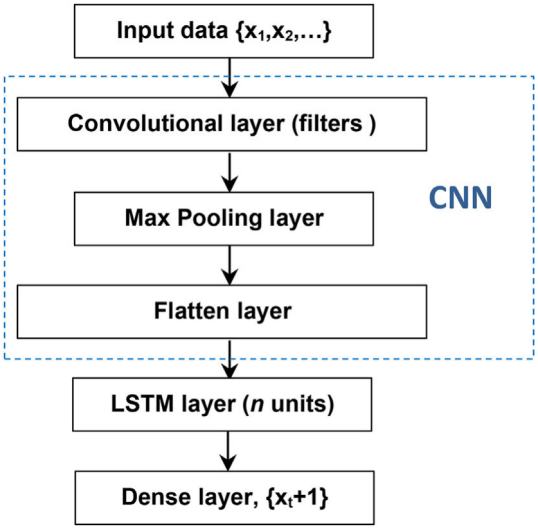


Fig. 6. Architecture of the CNN_{1D}-LSTM used in this work.

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).

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

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

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

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

where x_i and y_i are the measured and forecasted values, respectively, and $\overline{x}$ and $\overline{y}$ 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-

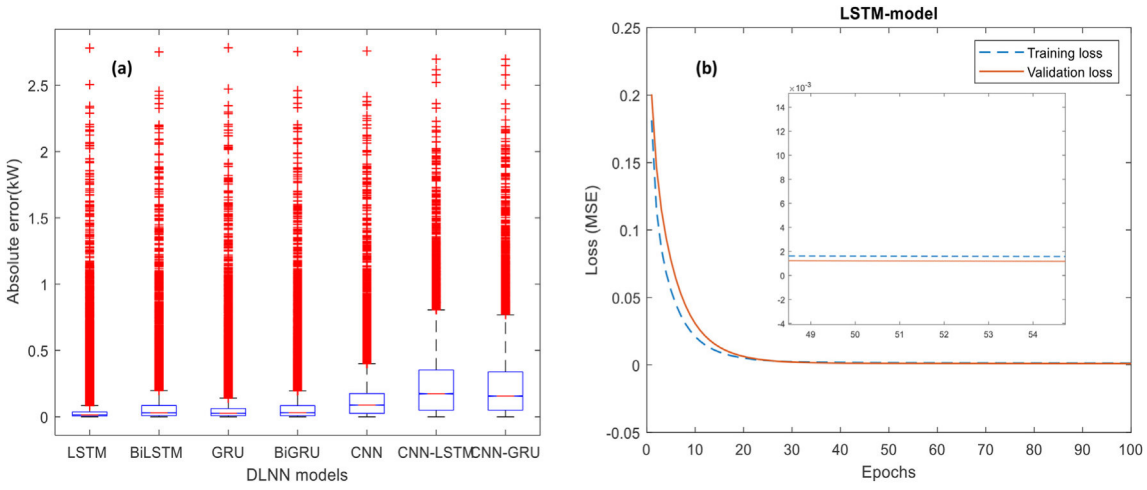


Fig. 7. a) Box plots of the absolute errors. b) Loss functions for the LSTM-based model.

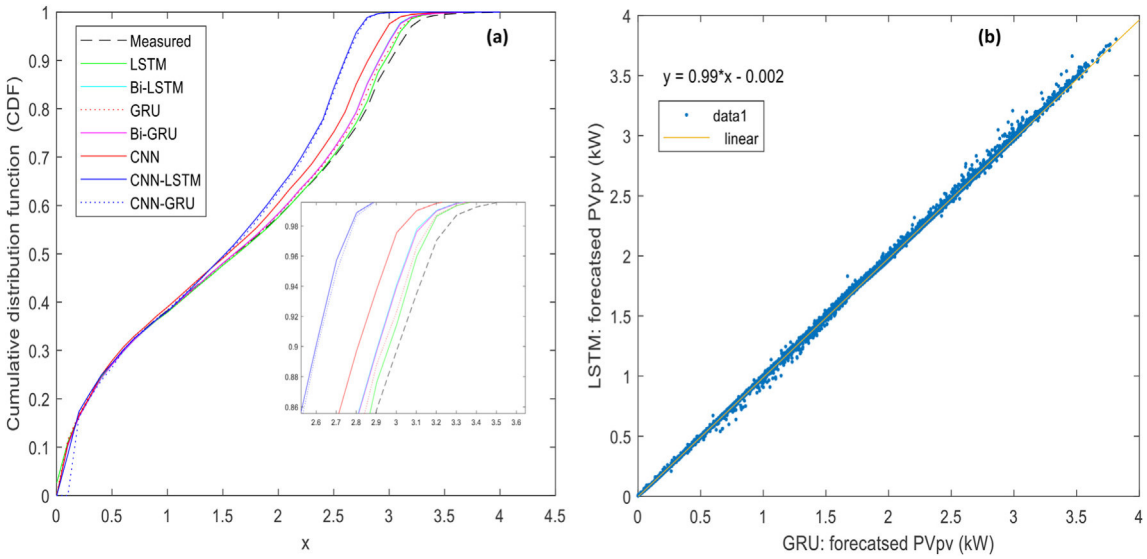


Fig. 8. a)Cumulative distribution functions. b) Correlation between the powers forecasted by the LSTM and the GRU-based models.

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

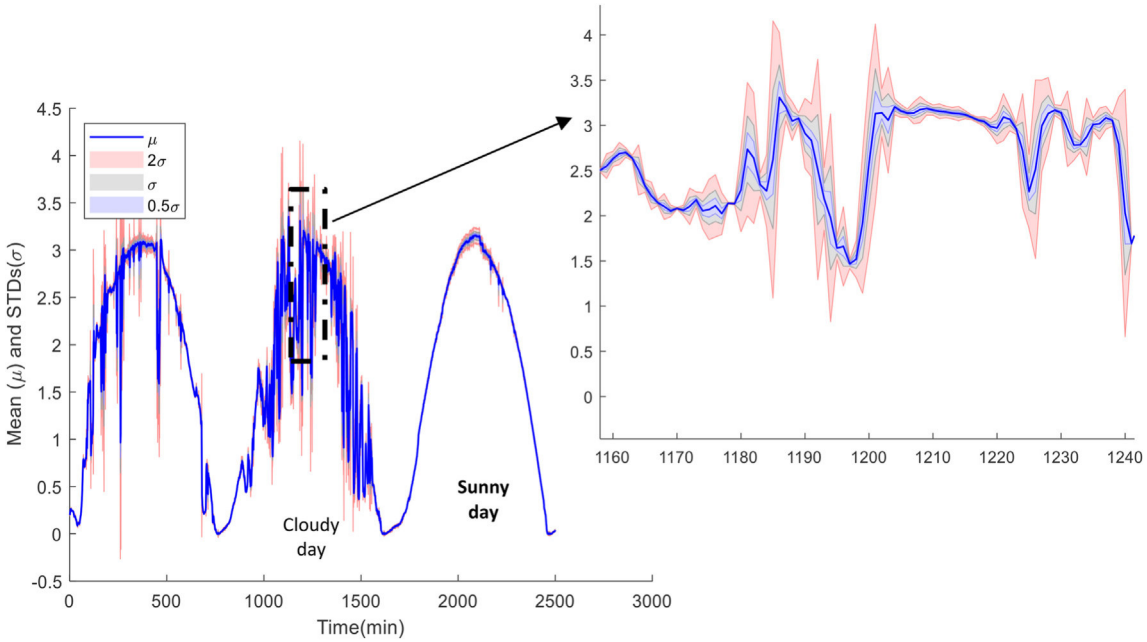


Fig. 9. Mean and standard deviations between the measured and the forecasted powers (developed LSTM model).

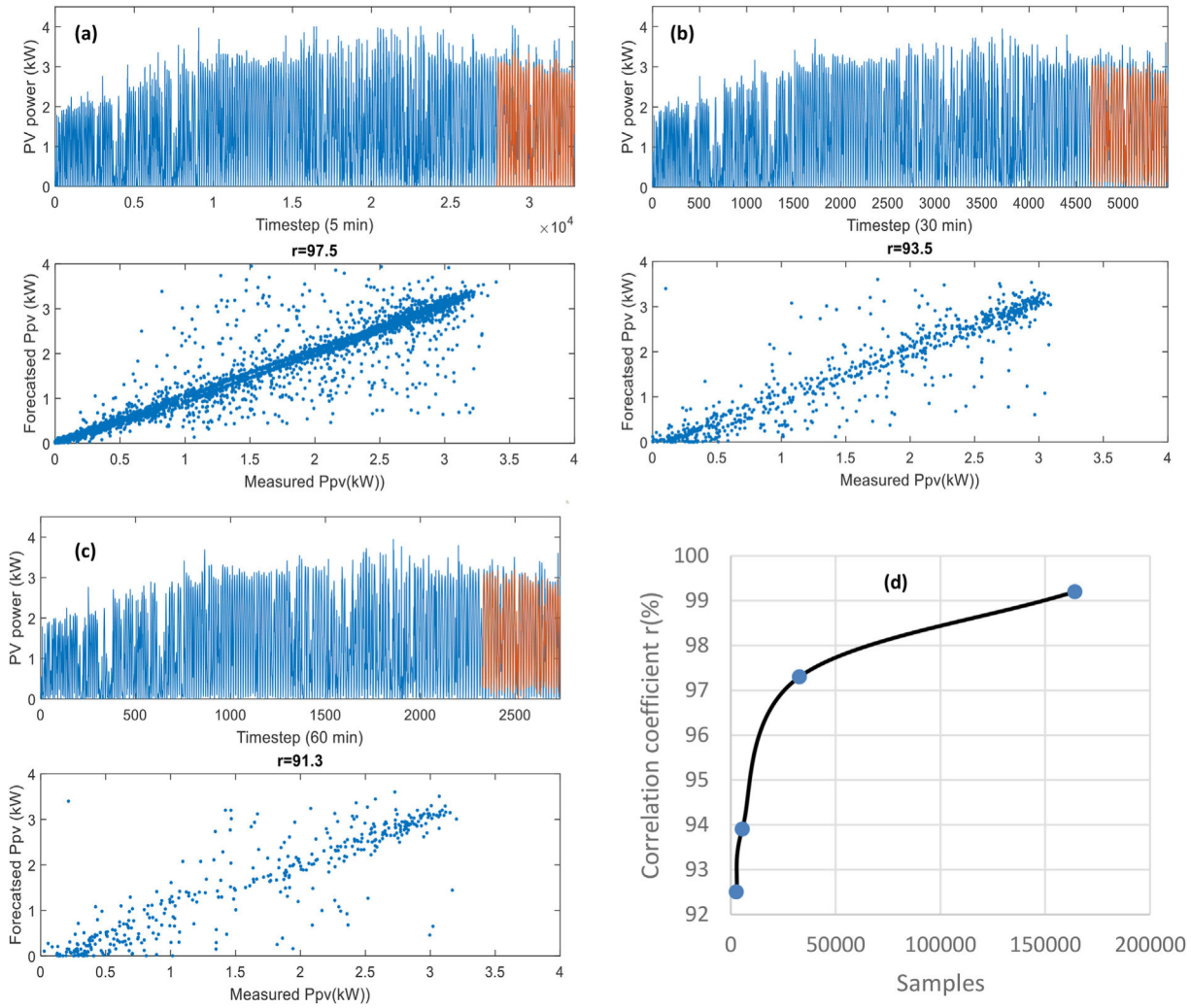


Fig. 10. Forecasted and measured power for the LSTM-based forecasters developed for different time horizons (the forecasted power is depicted in red).

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 × 100 × 50 × 1)	97.2	0.11	180	93.8	0.27	200	91.3	0.31	210
2nd architecture (5 × 100 × 50 × 25 × 1)	97.3	0.14	200	92.7	0.28	320	92.4	0.33	250
3rd architecture (9 × 50 × 50 × 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

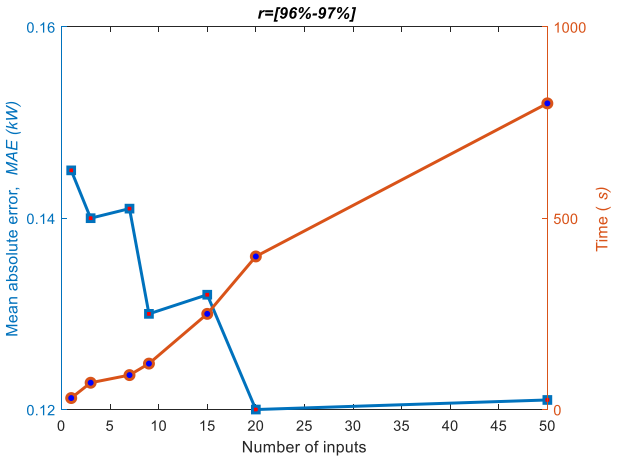


Fig. 11. Mean absolute error and training time for the LSTM-based forecasters.

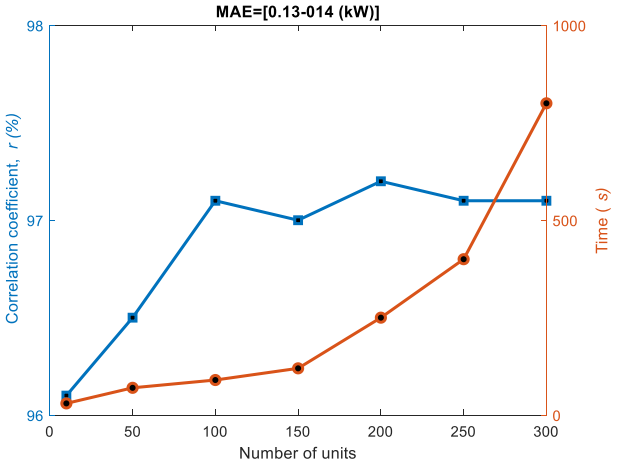


Fig. 12. Correlation coefficient and training time for the LSTM-based forecasters.

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

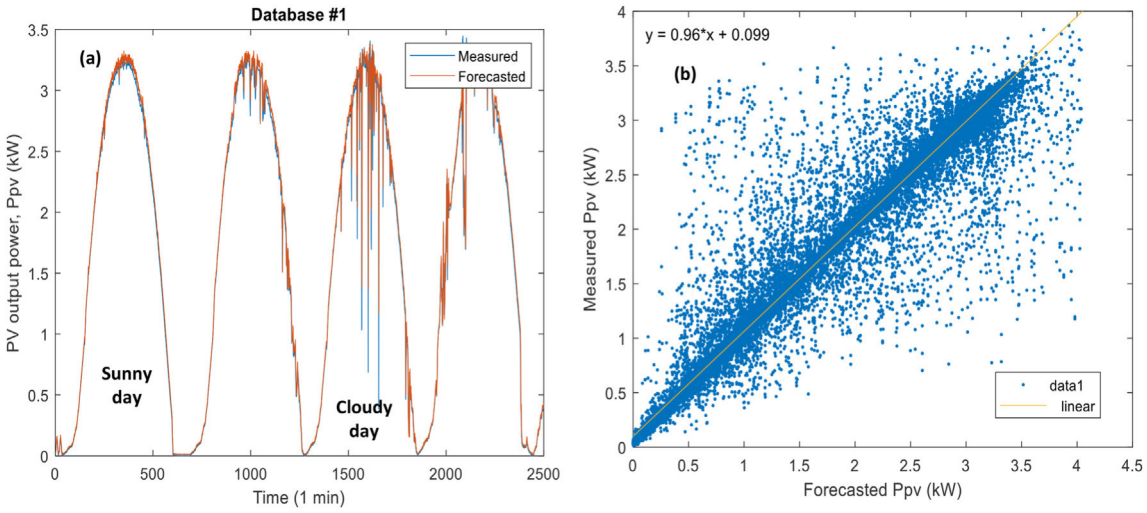


Fig. 13. LSTM-based multi-Step forecaster, configuration #3 (12 × 100 × 4). Forecasted and measured power.

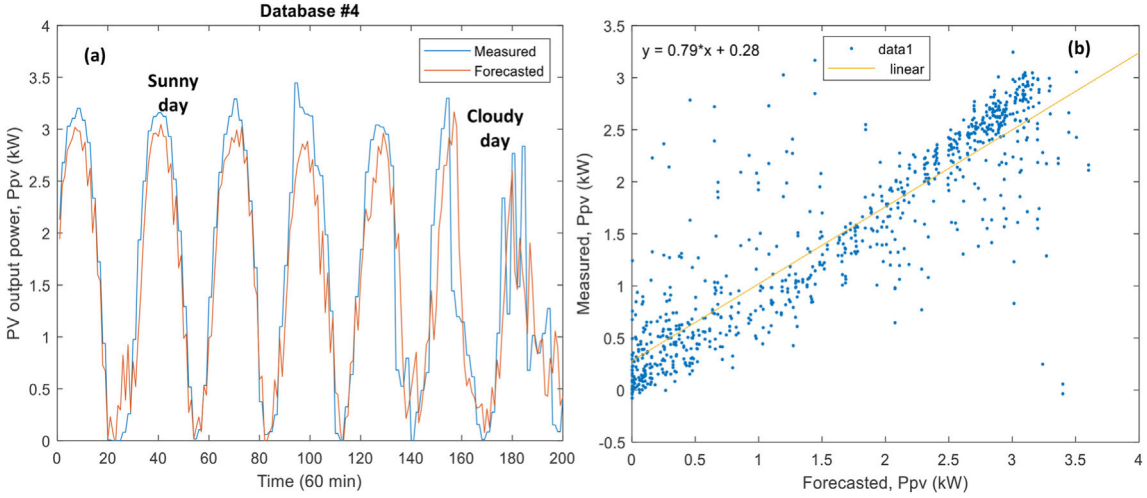


Fig. 14. LSTM-based multi-Step forecaster, configuration #1 (4 × 150 × 2). Forecasted and measured power.

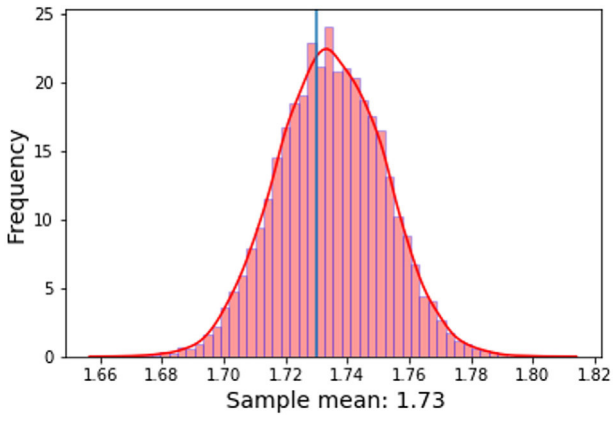


Fig. 15. The distribution function of the sample means.

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		

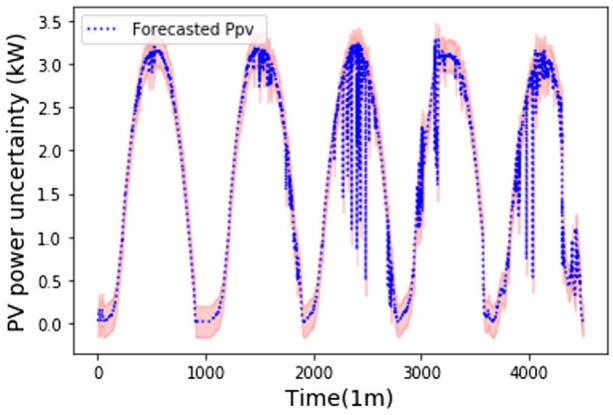


Fig. 16. Uncertainty quantification of the forecasted power.

- ✓ 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

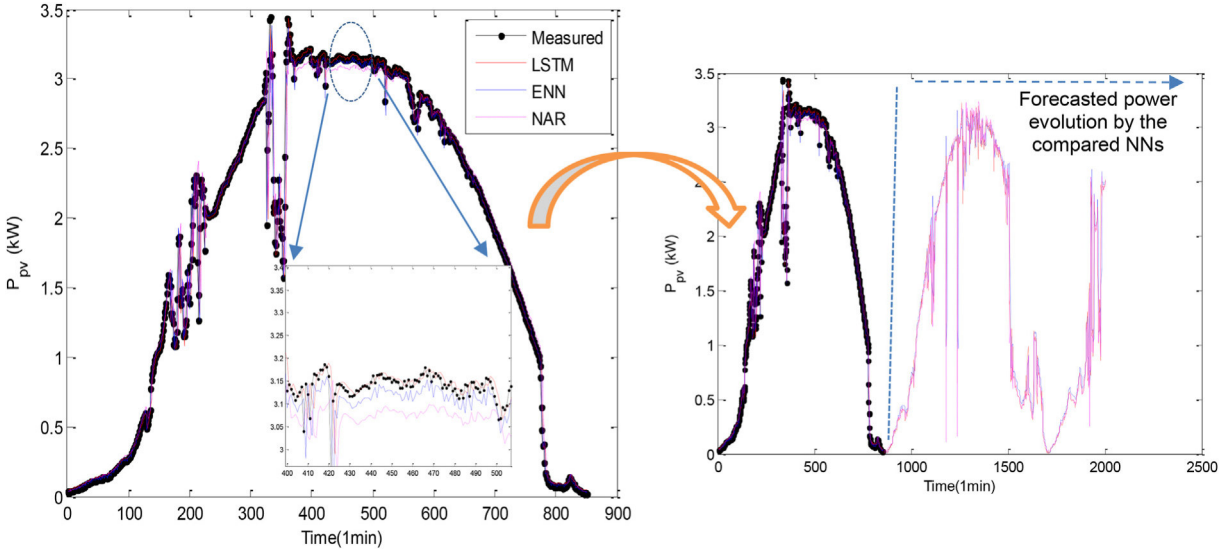


Fig. 17. Measured versus forecasted powers - one day (850 samples).

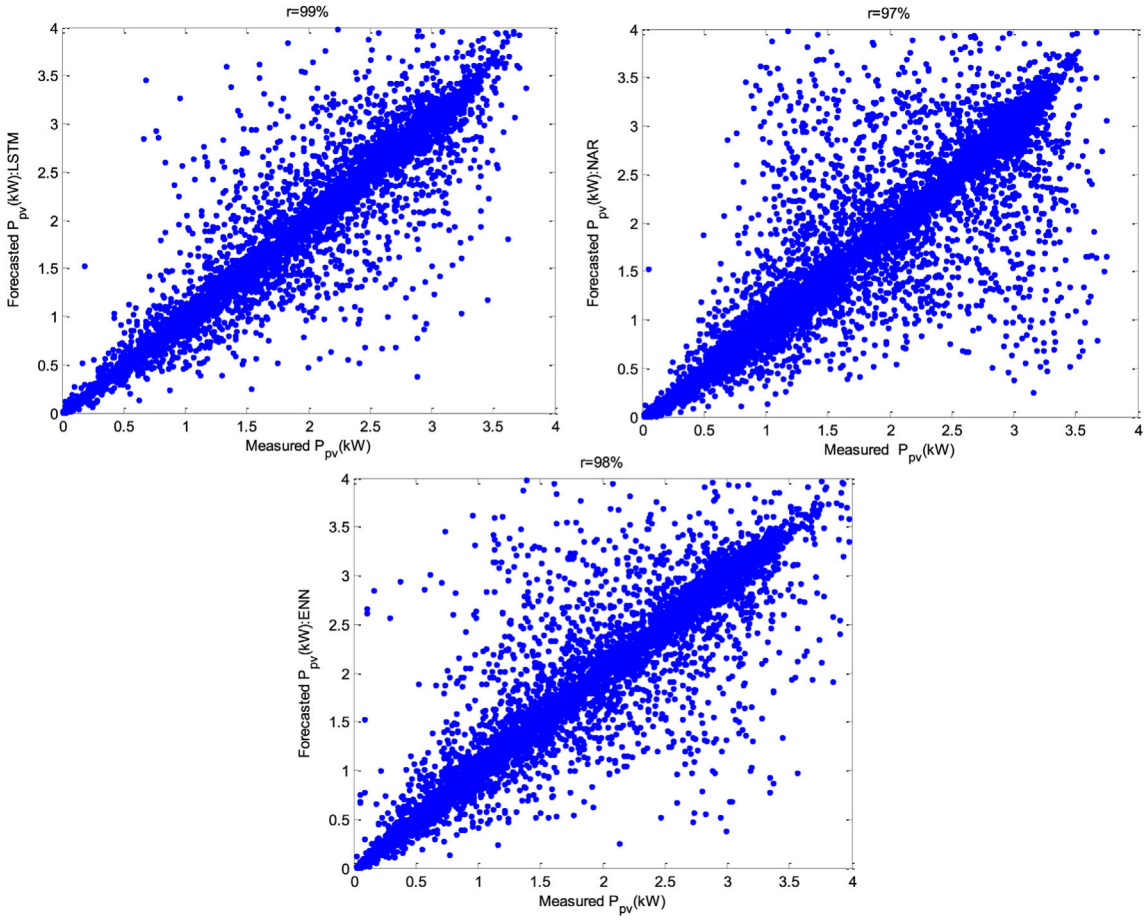


Fig. 18. Correlations between measured and forecasted powers.

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.

Sustainable Energy, Grids and Networks 27 (2021) 100502

Contents lists available at ScienceDirect

Sustainable Energy, Grids and Networks

journal homepage: www.elsevier.com/locate/segan

Real-time prediction of grid voltage and frequency using artificial neural networks: An experimental validation

N. Chettibi [a], A. Massi Pavan [b,*], A. Mellit [a,c], A.J. Forsyth [d], R. Todd [d]

[a] Renewable Energy Laboratory, University of Jijel, Jijel, Algeria
[b] Department of Engineering and Architecture, and Center for Energy, Environment and Transport Giacomo Ciamician - University of Trieste, Italy
[c] ICTP, Trieste, Italy
[d] Department of Electrical and Electronic Engineering, University of Manchester, Manchester, UK

ARTICLE INFO

Article history:
Received 10 February 2021
Received in revised form 16 May 2021
Accepted 6 June 2021
Available online 9 June 2021

Keywords:
Voltage and frequency forecasting
Real-time
Artificial neural networks
Multi-step prediction

ABSTRACT

In grid-connected Distributed Generation (DG) systems, with high-penetrations of renewable and energy storage assets, the prediction of grid voltage and frequency plays an important role in enabling the power quality support, the stabilization and monitoring of distribution networks. In this paper, a method based on Artificial Neural Networks (ANNs) and Deep Recurrent Neural Networks (DRNN) has been developed for very short-term prediction of grid voltage and frequency. For different time scales (183ms, 1s, 10s, 60s), one-step and multistep ahead forecasters are developed to predict the future behavior of grid parameters. This type of predictors can be used in distributed generation systems to enhance the control performance, to prevent the occurrence of grid faults and to improve the power systems stability. The data used to establish and validate the ANNs forecasters are provided from grid connected battery storage system installed at the University of Manchester. The developed prediction models have been validated experimentally via a dSPACE real-time controller. The obtained results show that the ANNs forecasters are able to predict in real time the grid voltage and frequency with satisfactory accuracy as the largest mean absolute percent error is 0.32%.

1. Introduction

Renewable energy source-based Distributed Generation (DG) systems are growing exponentially worldwide. In 2017 the share of electricity generation capacity from renewables exceeded 60% of the total; a 40% increase from 2002 [1]. Several factors contribute to this significant increase: emissions targets and public concerns related to climate change, depletion of conventional energy resources, air pollution in cities, the fact that distributed generators can be installed very quickly, and the decreasing cost of electricity from photovoltaic (PV) and wind power [2].

Non-dispatchable renewable energy resources provide 10% of the global electricity demand [1], and will play a more important role in the near future. However, the intermittent nature of renewable DG, which depend mainly on climatic conditions, has negative impact on the power quality of distribution grids especially in terms of frequency and voltage stability [3]. On the other hand, with the emergence of Electric Vehicles (EVs) interconnected to low voltage grid feeders, extra load is introduced to distribution networks that yields to voltage deviations at the point of coupling when charging EVs [4]. This new scenario of smart grid brings technical challenges in matching generation and demand, improving the power quality and maintaining the grid voltage and frequency at acceptable levels [5,6]. For these reasons, regulators are today issuing new grid codes where prosumers are called to participate in the regulation of both frequency and voltage at the point of connection [7,8].

In this context, the capability to forecast the grid voltage and frequency is of paramount importance for the monitoring, control and protection of power systems. Predictions can be used as a benchmark to determine whether there is a fault in the system, and if there is any problem in the measurement or transmission of signals [9]. Other applications are the optimization of the state-of-charge (SOC) of energy storage systems [10], and the optimal control of microgrid (MG) [11] and virtual power plants [12]. In these cases, the order of magnitude for the forecasted horizon varies from a few seconds to several hours. Another application, where the desired forecasted horizon is shorter (a few milliseconds, i.e. very short term forecasting), regards the control of power systems when a delay in the communication of the variables between different equipment can compromise the stability of the system [13]. In this case, statistical modeling techniques can overcome communication delays [14].

* Corresponding author.
E-mail address: apavan@units.it (A. Massi Pavan).

https://doi.org/10.1016/j.segan.2021.100502
2352-4677/

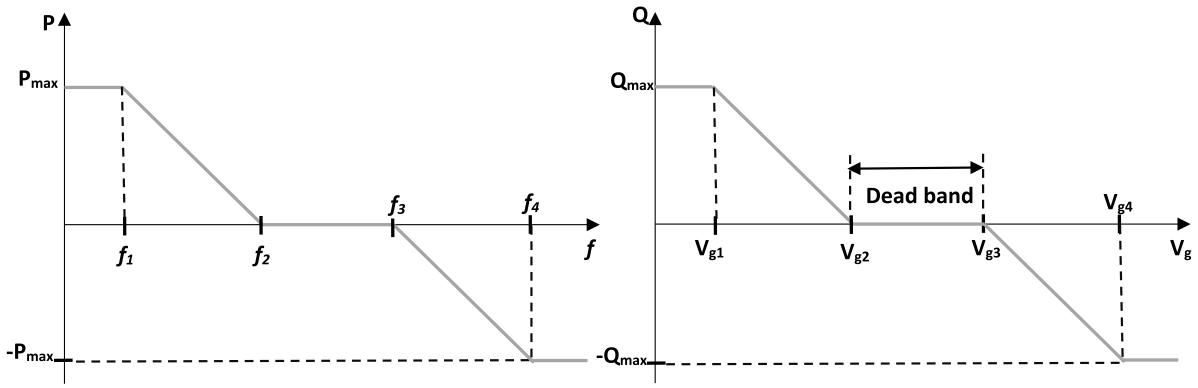


Fig. 1. Watt-frequency and Volt-Var curves used to mitigate frequency and voltage oscillations.

Different approaches have been proposed in the literature for the power-line frequency prediction. In [6] a weighted-nearest-neighbor predictor is developed to forecast the frequency profile for time horizon of one hour. A state–space model with Kalman filter and the basic functions method are used in [9] for dynamic forecast of grid frequency. The performance of proposed methods were assessed in terms of root mean square error (RMSE) that was about 0.002 Hz and 0.01 Hz for the sampling interval of 0.1 s and 1 s respectively. The authors in [15] have applied a cellular computational extreme learning machine network for prediction of bus frequencies in a power system. In [16], a feed-forward neural network is developed for hourly prediction of power system frequency. The suggested model takes into account the dependency of frequency on various parameters such as power demand, available generation and wind power. Another approach based on e-Support Vector Regression is proposed in [17] for dynamic prediction of the samples of powerline frequency in a wide area measurement system. The frequency prediction of synchronous generators in a power system is performed in [18] using cellular generalized neuron networks, where the performance are checked based on real time tests.

On the other hand, a limited number of papers have addressed the forecast problem of grid voltage. In [19] an approach for short-term voltage prediction is proposed based on convolutional neural network and empirical mode decomposition. The suggested prediction method was validated for three time scales (1 h, 6 h and 12 h) and was compared with different forecast models. A method to enhance real-time forecast of voltage by leveraging a limited set of real-time measurement is proposed in [20]. A feed-forward neural network approach is proposed in [21] for real time voltage estimation in low voltage distribution grid. The authors have demonstrated that the performance of the ANN model are not sensitive to the number of hidden nodes as well as to the level of PV generation. A local estimator based on feed-forward ANN is also presented in [22] for real-time estimation of voltage profile in distribution system. Several tests have been performed for low /medium voltage feeders to check the accuracy of proposed estimator, which increases with the number of measurements available at its input.

The novelty of the present study regards the capability to simultaneously predict in real-time both the voltage and the frequency of a low-voltage feeder. Starting from the preliminary work described in [23], eight (four for the frequency and four for the voltage) one-step ahead forecasters have been developed with four different time horizons: 183 ms, 1 s, 10 s, and 60 s. Another major contribution of this work is the introduction of multi-step forecasters that can predict the grid voltage and frequency up to three steps ahead, and with three different time horizons: 183 ms, 1 s, and 10 s. Moreover, the simulation and comparisons between different types of ANNs (FFNN, RBF, ENN and LSTM) based forecasters is carried out in this paper.

The paper is organized as follows: the problem related to the forecast of the grid quantities is defined in the next section, while the main contributions are summarized in Section 3. Section 4 describes the ANN-based forecasters developed in this study. Section 5 presents the results, while Section 6 deals with the conclusions.

2. Problem statement

The grid voltage and frequency are important parameters commonly used in the control schemes and monitoring systems of DG. Thus, Perform an accurate forecast of grid quantities can significantly enhance the power quality and grid stability [6]. The one-step ahead forecast of voltage and frequency allows the compensation of the measurement delay and consequently the improvement of dynamical performance of control systems like VF-droop control of microgrid [24], Frequency-Watt Control (FWC) [25] and Volt-VAr Control (VVC) of smart inverters [3]. The Fig. 1 illustrates the FWC and VVC functions that are usually incorporated in grid connected smart inverters. The active and reactive powers can be controlled according to the grid frequency and voltage at the point of coupling, respectively [3,25]. Hence, the multi-step ahead prediction will give information on future trajectory of grid frequency and voltage that helps to make decision on the suitable control actions to regulate the active and reactive powers. On the other hand, the multi-step ahead prediction of grid voltage and frequency can be used in monitoring systems to prevent and mitigate undesirables effects of grid faults (like three phase voltage unbalance, voltage flicker, frequency fluctuation, etc.).

3. Main contributions

In general, forecasting methods based on machine learning use databases (i.e. time series, in this work voltages and frequencies) in order to produce predictions for a given horizon. In the recent years, classical and deep neural networks like radial basis function neural networks [26] and long short-term memory (LSTM) [27] have been widely used for the forecasting of time series data. This is due to the ability of ANNs to capture sharp changes in the input–output relationship [28]. Without need for mathematical development, ANNs have shown higher accuracy than classical stochastic approaches such as autoregressive moving average and autoregressive integrated moving average, Kalman filter and other statistical methods [2,28,29].

In this work, a number of ANNs based forecast models such as Feed-Forward Neural Networks (FFNNs), Radial Basis Function

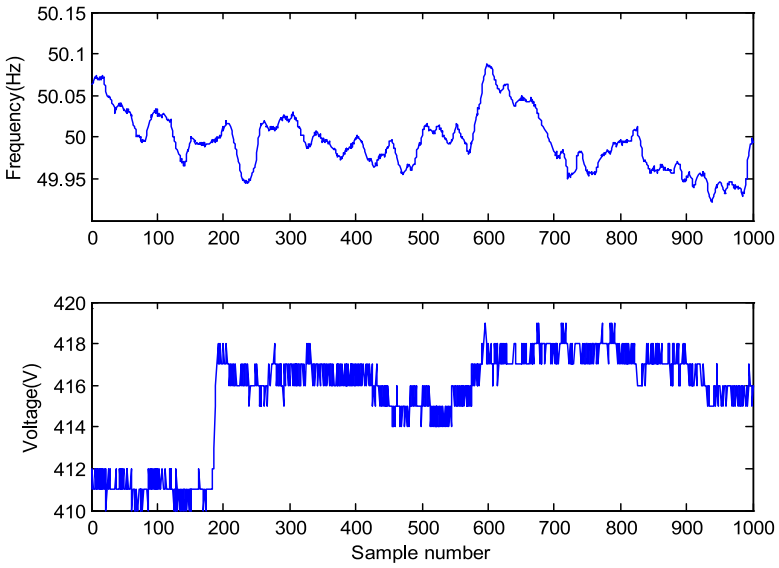


Fig. 2. One thousand seconds of measured frequency and line-to-line voltage data from the grid, sampled every one second.

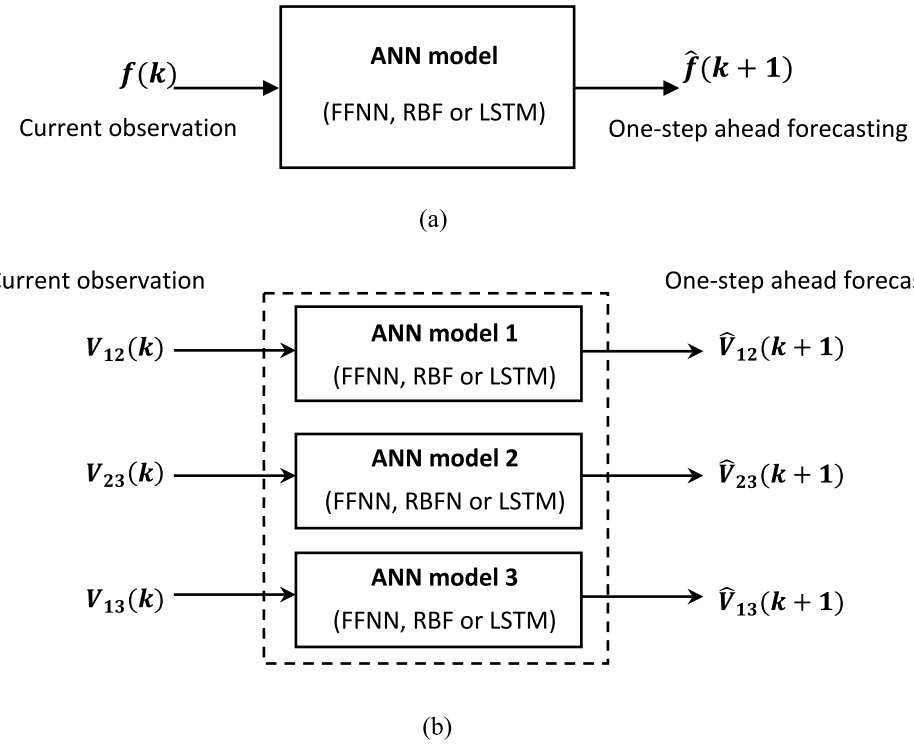


Fig. 3. One-step ahead ANN-based forecast models adopted for: (a) Frequency, (b) Voltage.

Neural Network (RBFN), LSTM, and online-trained Elman Neural Networks (ENNs) are developed to predict the grid voltage and frequency for four timescales (183 ms, 1 s, 10 s, 60 s). A comparative study between the FFNNs, RBFNs and LSTMs based forecasters, trained using an historical dataset, is conducted for both one-step and multistep ahead predictions. On the other hand, the ENNs-models established in Simulink environment are simulated in real-time, in addition to the FFNNs forecasters, using a dSPACE controller to enable online forecasting of grid quantities. In this case, the adaptation of ENNs parameters is performed in real-time without prior knowledge on voltage and frequency profiles. The chosen ANNs have simple structures and can be easily implemented in real time power systems.

4. Forecasters implementation

4.1. Database

The data used for the training and validation of voltage and frequency forecasters are recorded at a low voltage (of 400 V) grid connection. The data come from the point of common coupling with the grid of a battery storage system installed at the University of Manchester [23,30]. As an example, Fig. 2 shows part (one thousand seconds) of the measured frequency and line-to-line voltage (voltage between phases one and two) measured on the 29th of June 2018 from 2.43 pm. The data used for the ANNs training was logged every 183 ms, 1 s, 10 s and 60 s in the

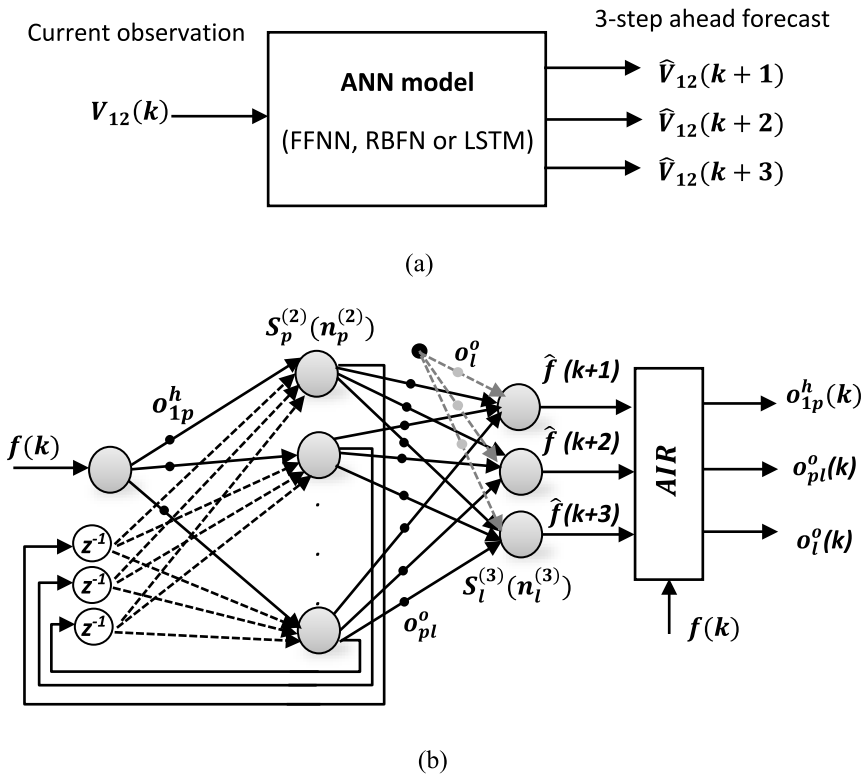


Fig. 4. Structure of the proposed three-step ahead forecaster for: (a) voltage, (b) frequency.

dSPACE controller. The number of samples for the 183 ms, 1 s, 10 s and 60 s time scales are 21,858, 15,000, 10,001 and 10,001 respectively.

4.2. One-step ahead forecasting

At the time instant, t, the one-step ahead forecaster estimates the future value of y expected at $(t+1)$ based on the actual and previously observed data [31]:

$$\hat{y}_{t+1} = f(y_t, y_{t-1}, \ldots, y_{t-d+1}) \quad (1)$$

where $t \in \{d, \ldots, N-1\}$, $\{y_t, y_{t-1}, \ldots, y_{t-d+1}\}$ are the actual and past values of the time series, $\hat{y}_{t+1}$ is the forecasted value, f represents the forecasting model, d is the embedded dimension of the database (time series), and N is the size of the database.

4.2.1. Offline trained ANNs

The developed ANNs-based forecasters (FFNNs, RBFNs and LSTM) are trained offline using a historical dataset logged for different time scales. During the offline training phase, the networks learn from the input-target examples in the database and adapt theirs parameters accordingly. The general form of ANN models proposed for frequency and voltage magnitude prediction is given Fig. 3. The adopted topology enables the prediction of the future value for the given timescale based on only the current observation of grid frequency or voltage.

To enhance the forecast accuracy and to avoid the overfitting problem, we have established a separate ANN forecaster for each phase-to-phase voltage (see Fig. 3.b). The topology of the developed FFNNs is based on a single hidden layer with minimal number of nodes (between 4 and 6). The algorithms of Levenberg–Marquardt (*trainlm*) and Bayesian regularization (*trainbr*) are used for the offline training of the ANNs (FFNNs and RBFNs). For the development of each network, the dataset has been randomly divided into three subsets using the function

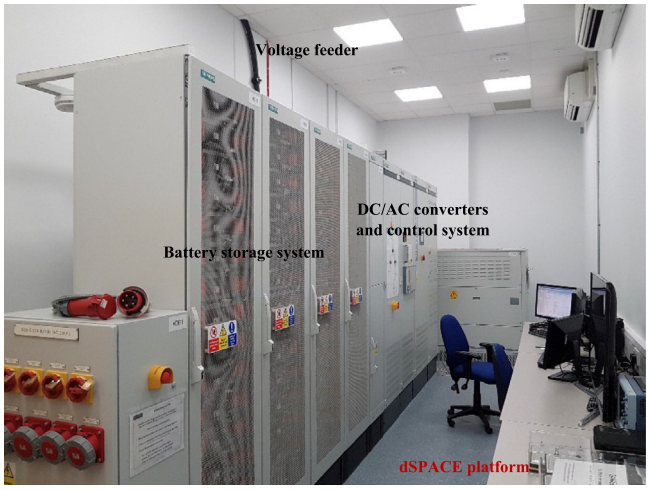


Fig. 5. Test facility at Manchester University [30].

"*dividerand*" available in MATLAB (Ver 2013a). The first subset has been used for the training, the second for the validation, and the third for the preliminary testing of the forecasters. LSTMs forecasters are developed with Python language using Keras library functions [32].

4.2.2. Online trained ENN

As suggested in [23], recurrent ENNs can be used for the one-step ahead prediction of the grid voltage and frequency for different timescales. These neural predictors are based on an online adjustment of the weights so that the forecast capability is adapted to any change in the system dynamic. The update of the ENN weights is performed using the Adaptive Interaction Rule (AIR) described in [33,34].

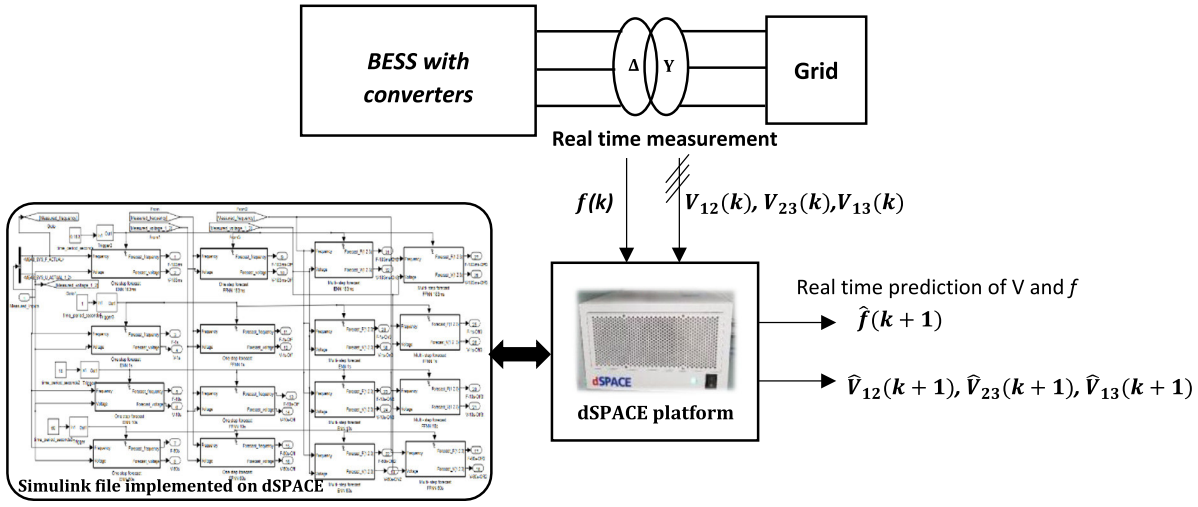


Fig. 6. Real time test of voltage and frequency forecasters.

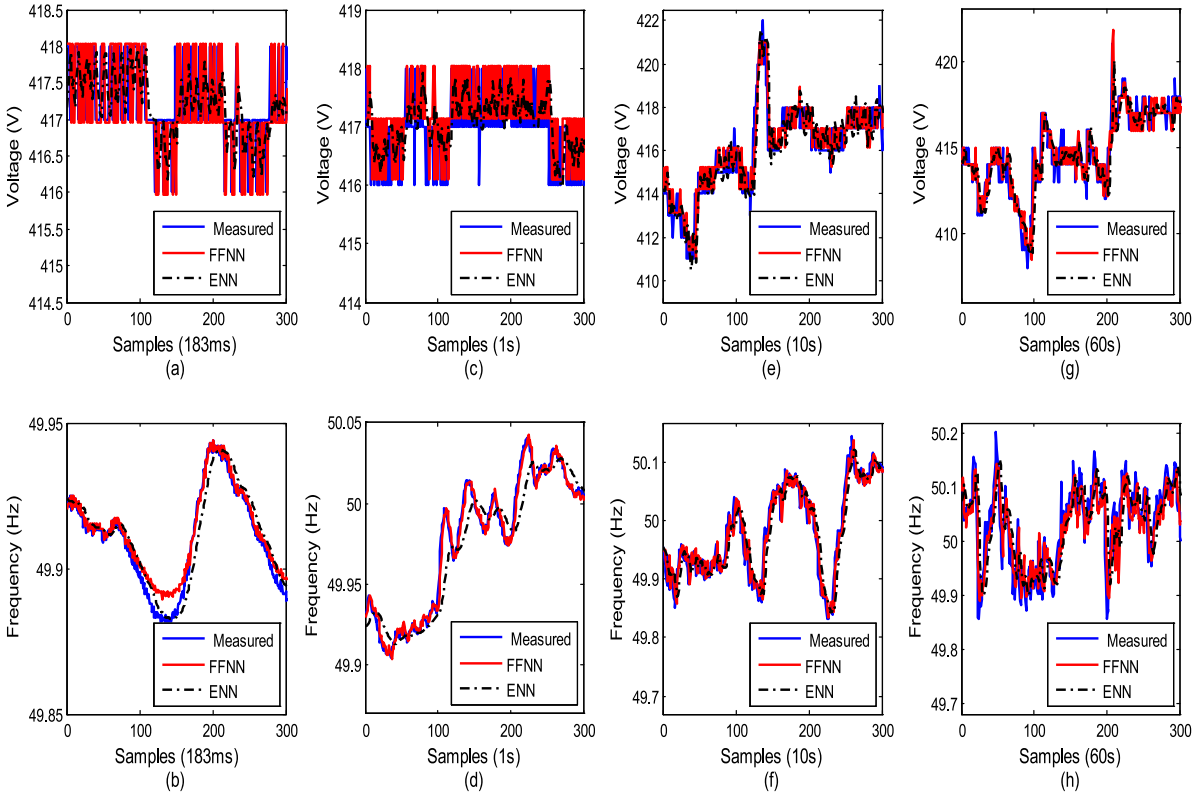


Fig. 7. Real-time simulation for the one-step ahead forecasters with four time horizons (183 ms, 1 s, 10 s, and 60 s, period: March).

4.3. Multi-step ahead forecasting

Multi-step ahead forecasting is more complicated than the one-step forecasting because of the possibility of accumulation of errors, a reduced accuracy, and an increased uncertainty [31,35]. For this reason, in the last few decades one-step ahead forecasting has been the most investigated area, but recently, due to advances achieved in computing and data analysis, multi-step forecasting are become more and more viable.

Different strategies for multi-step forecasting have been presented in the literature including the multi-output strategy, recursive method, and direct strategy [35]. A multi-output forecasting strategy has been used in this work in order to improve the precision of the network. According to the multi-output strategy, the multi-step ahead prediction problem can be formulated as [35]:

$$\{\hat{y}_{t+1}, \hat{y}_{t+2}, \ldots, \hat{y}_{t+H}\} = f\,(y_t, y_{t-1}, \ldots, y_{t-d+1}) \quad (2)$$

where H is the forecast horizon, d is the number of samples, and $\{\hat{y}_{t+1}, \hat{y}_{t+2}, \ldots, \hat{y}_{t+H}\}$ is the forecasted time series. Three-step ahead forecasters have been developed for the 183 ms, 1 s, and 10 s time horizons. A two-step ahead predictors has been used for the 60 s horizon.

4.3.1. Offline trained ANN

The three-step ahead ANN forecasters (FFNN, RBFN, and LSTM) are able to determine the next three values of voltage and frequency that correspond to the instants *t+1, t+2, t+3* based on

the actual observation at the time instant t. The 60 s forecaster estimates the future two steps of the grid quantities. The general structure of the multistep ahead prediction model of line-to-line voltage (V_{1-2}) is presented in Fig. 4.a.

4.3.2. Online trained ENN

As an example, with reference to the frequency, the adaptive ENN-based multi-output forecaster used in this work is shown in Fig. 4.b. This can be described by [34]:

$$\begin{cases} h^{(1)}(n^{(1)}) = n^{(1)}(t) \\ h_p^{(2)}(t) = S_p^{(2)}\left(o_{1p}^h h^{(1)}(t) + \sum_r h_r^{(2)}(t-1)\right) \\ h_l^{(3)}(t) = S_l^{(3)}\left(\sum_p o_{pl}^o h_p^{(2)}(t) - o_l^o\right) \end{cases} \tag{3}$$

where $l \in \{1, 2, 3\}$, p and $r \in \{1, 2, \ldots, N_h\}$, N_h is the number of hidden nodes, $h^{(1)}$, $h_p^{(2)}$, $h_l^{(3)}$ are the input, the pth hidden node output and the lth network output respectively. $S_p^{(2)}$ and $S_l^{(3)}$ are the activation functions of the hidden and output neurons respectively. o_{1p}^h and o_{pl}^o are the weights of the network.

Hence, the following AIRs have been used for the online adaptation of the ENN weights [33,34]:

$$\begin{cases} \dot{o}_{1p}^h = S_p^{(2)'}\left(n_p^{(2)}\right)\frac{h^{(1)}}{h_p^{(2)}}\sum_{l=1}^3 o_{pl}^o \dot{o}_{pl}^o \\ \dot{o}_{pl}^o = -\alpha_l S_l^{(3)'}\left(n_l^{(3)}\right) h_p^{(2)} e_l \end{cases} \tag{4}$$

where α_l is the learning rate, and e_l is the forecast error for the lth output neuron. The prediction errors that has to be minimized at each iteration are defined as:

$$\begin{cases} e_1(k) = \hat{f}(k+1) - f(k+1) \\ e_2(k) = \hat{f}(k+2) - f(k+2) \\ e_3(k) = \hat{f}(k+3) - f(k+3) \end{cases} \tag{5}$$

Once the learning sample is presented at the ENN input, the network weights vectors are updated at the same instant in such a manner that the output errors e_l can be reduced.

5. Results and discussion

In this work, two comparative studies are performed. The first one is done for the offline trained FFNNs, RBFNs and LSTMs based forecasters. The goal is to understand the best option to be implemented on the dSPACE controller. The second study is realized experimentally using a dSPACE platform to perform real-time prediction of grid quantities using ENNs and offline-trained ANNs-models.

5.1. Simulation results

The offline-trained forecasters presented in the previous section have been simulated and compared for different timescales (183 ms, 1 s, 10 s, 60 s). To check the performance and the generalization ability of developed FFNNs, RBFNs and LSTMs, a dataset that has not be seen during the offline training phase is used to test them. Table 1 summarizes the results of one-step and multistep forecasting using the test data for different timescales. It can be seen that the predicted values of voltage and frequency have a good agreement with the measured values for all cases (see Table 1). Nevertheless, it can be noted that the one-step and multistep forecast accuracy decreases for a long time scale of 60 s. Besides, the frequency prediction performance are better than those of the voltage forecasting. The voltage errors (i.e. RMSE) obtained for different time horizons arrange between 0.6 V-1.9 V, whereas the frequency errors (RMSE) are less than 0.056 Hz. It can be seen from Table 1 that the results of the FFNNs models are very similar to those obtained by RBFNs with slight differences. The performance of the LSTMs-based forecasters get worse when the timescale increases. This is because this type of DRNN requires a large dataset in order to give accurate results.

For example, in the case of one-step and three-step frequency prediction, with the time scales (183 ms and 1 s), the values of correlation coefficients (R) are close to one and statistical errors (RMSE and MAE) are less than 10^{-3}, obtained by the three types of ANNs. The results of FFNNs and RBFNs are identical in these two cases. Besides, in the case of one-step forecast of voltage, the FFNN models have given the best values of error metrics for the four time scales. On the other hand, the 183 ms, 10 s and 60 s voltage forecasters based on RBFNs offer better performance than the FFNNs and LSTMs for the case of multi-step ahead prediction. In addition, it can be seen that the LSTMs models have given the worst results for 60 s one-step and 2-step ahead voltage forecast. The accuracy of the two-step ahead 60 s-forecasters is satisfactory, but it can be improved by taking into account more measurement variables (like voltage and frequency errors, …) at the ANNs input.

In term of complexity, the adopted FFNNs topology (hidden nodes: less than 6) is much simpler than those of the RBFNs (hidden nodes: more than 10) and LSTMs (150 cells, batch size = 64, and 1 dense layer) forecasters. Furthermore, it has be seen that LSTMs have not given precise results, that means that this type of neural networks, based on deep learning, is not very suitable for this application. For these reason and as satisfactory results are obtained with simple FFNN models, it have been chosen for simulation in real time using dSPACE controller. Hence, the Simulink blocks of the developed FFNN models are generated using the Matlab command "*gensim*" to perform their real validation. The ENNs forecasters described in the previous section are also constructed in the Simulink environment based on Eq. (3) and using the AIRs. The main features of the developed FFNNs and ENNs prediction models are given in Tables 2 and 3.

5.2. Experimental results

The developed FFNNs and ENNs based forecast models have been implemented in the dSPACE real-time control system that records the measured frequency and three-phase line-to-line voltages of grid feeder (415 V–50 Hz). The photograph of Fig. 5 gives a general description of grid connected storage system installed at the University of Manchester, where the real time test is performed.

In order to check the prediction accuracy of the developed forecasters, we performed three experimental tests in three different periods:

- The first test was performed during a weekend, only for the one-step ahead forecasters. The record of measured and forecasted data started at 7:04pm of 1th March 2019;
- For the second experiment, we recorded for more than 2 h and we started the data logging at 4:04pm of the 26th April 2019. This test was to validate both the one-step and the multi-step ahead ANN predictors;
- The last experiment lasted 540 min, starting at 9:15 am of the 13th of May 2019.

All data is considered in this section to evaluate the performance of the forecasters. Fig. 6 gives a general description of the experimental validation method. Tables 4–6 summarize the forecasters performance presenting different metrics such as R, RMSE, and MAPE. Bold numbers indicate the best results for each time scale.

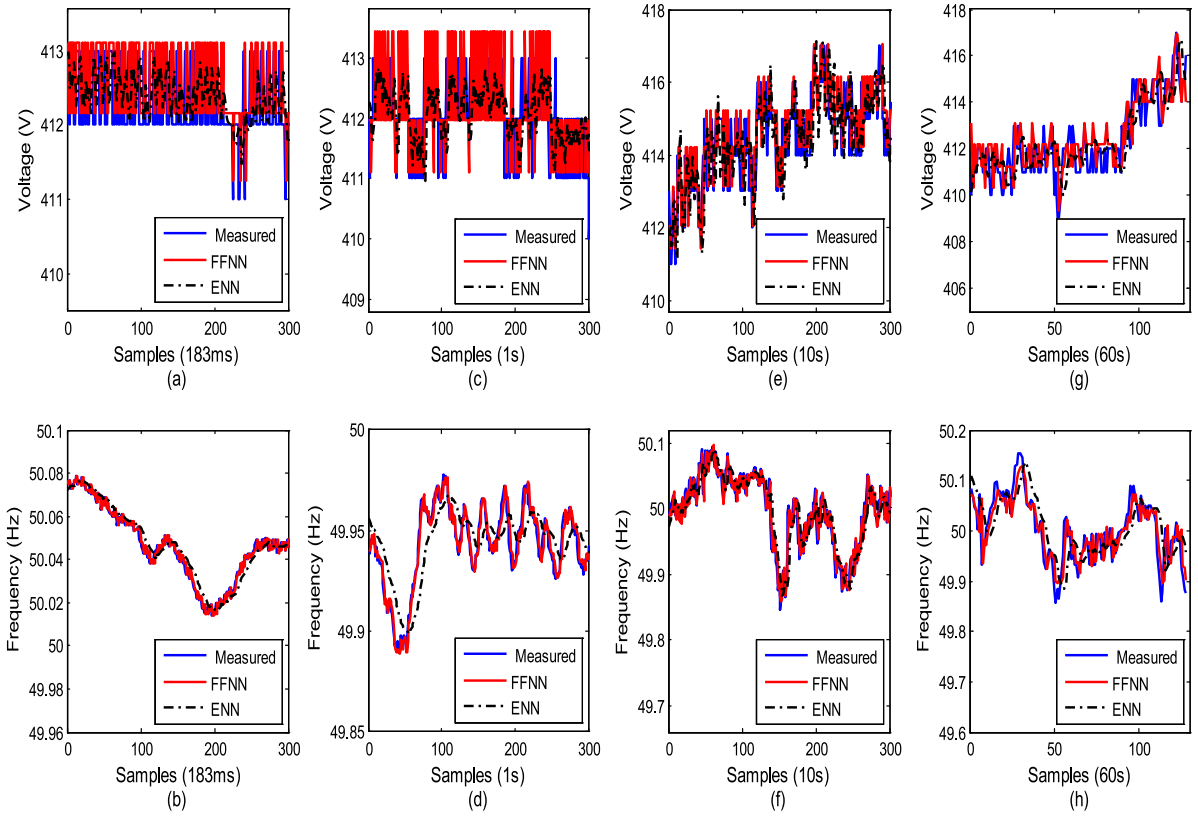


Fig. 8. Real-time simulation for the one-step ahead forecasters with four time horizons (183 ms, 1 s, 10 s, and 60 s, period: April).

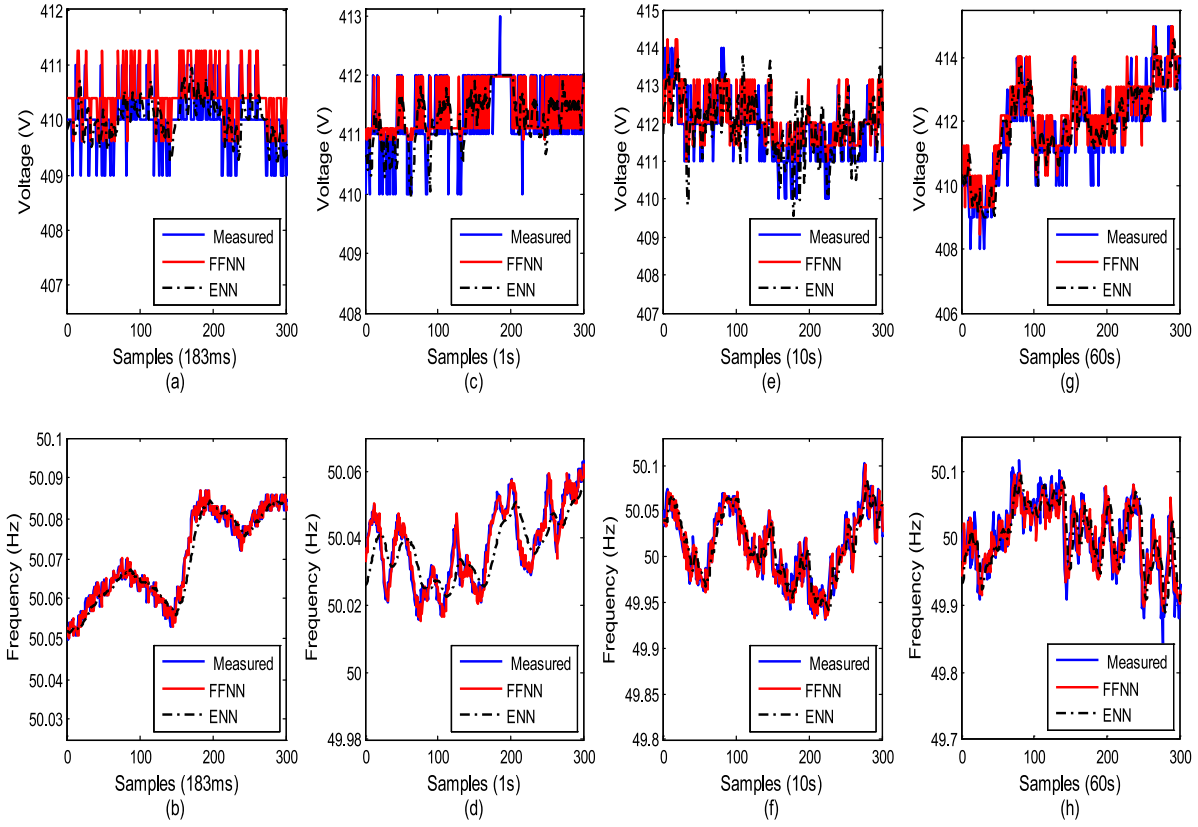


Fig. 9. Real-time simulation for the one-step ahead forecasters with four time horizons (183 ms, 1 s, 10 s, and 60 s, period: May).

Table 1
Errors metrics calculated for the test dataset with offline-trained ANNs predictors.

Parameter Forecaster		One-step ahead prediction				Multi-step ahead prediction				
		R	RMSE	MAE	MAPE (%)	R	RMSE	MAE	MAPE (%)	
Voltage V1-2 (timescale 183 ms)	FFNN	**0.9500**	**0.6002 V**	**0.3945 V**	**0.0950**	0.9546	0.6977 V	0.5477 V	0.1307	
	RBFN	0.9500	0.6689 V	0.5144 V	0.1241	**0.9607**	**0.6754 V**	**0.5350 V**	**0.1286**	
	LSTM	0.9481	0.7513 V	0.6533 V	0.1849	0.9333	0.7564 V	0.5917 V	0.1425	
Frequency (timescale 183 ms)	FFNN	**0.999**	**0.0020 Hz**	**0.0016 Hz**	**0.0031**	**0.9969**	**0.0023 Hz**	**0.0018 Hz**	**0. 0036**	
	RBFN	**0.999**	**0.0020 Hz**	**0.0016 Hz**	**0.0031**	**0.9969**	**0.0023 Hz**	**0.0018 Hz**	**0.0036**	
	LSTM	0.999	0.0021 Hz	0.0019 Hz	0.0024	0.9960	0.0036 Hz	0.0023 Hz	0.0038	
Voltage V1-2 (timescale 1 s)	FFNN	**0.9428**	**0.7484 V**	**0.5988 V**	**0.1443**	**0.9196**	**0.6980 V**	**0.5289 V**	**0.1260**	
	RBFN	0.9424	0.7689 V	0.6306 V	0.1520	0.9192	0.7062 V	0.5339 V	0.1283	
	LSTM	0.9325	0.8876 V	0.7753 V	0.277	0.9187	0.8024 V	0.6239 V	0.1345	**3-step ahead**
Frequency (timescale 1 s)	FFNN	**0.9975**	**0.0035 Hz**	**0.0028 Hz**	**0.0055**	**0.9938**	**0.0055 Hz**	**0.0043 Hz**	**0.0085**	
	RBFN	**0.9975**	**0.0035 Hz**	**0.0028 Hz**	**0.0055**	**0.9938**	**0.0055 Hz**	**0.0043 Hz**	**0.0085**	
	LSTM	0.9974	0.0039 Hz	0.0031 Hz	0.0033	0.9937	0.0060 Hz	0.0046 Hz	0.0092	
Voltage V1-2 (timescale 10 s)	FFNN	**0.9436**	**0.7270 V**	**0.5287 V**	**0.1274**	0.9407	0.8414 V	0.6333 V	0.1528	
	RBFN	0.9437	0.7495 V	0.5880 V	0.1418	**0.9438**	**0.8229 V**	**0.6214 V**	**0.1499**	
	LSTM	0.9336	0.8046 V	0.6987 V	0.2185	0.8584	1.0292 V	0.7375 V	0.1791	
Frequency (timescale 10 s)	FFNN	0.9806	0.0155 Hz	0.0120 Hz	0.0236	0.9525	0.0248 Hz	0.0194 Hz	0.0385	
	RBFN	**0.9808**	**0.0154 Hz**	**0.0117 Hz**	**0.0234**	**0.9531**	**0.0246 Hz**	**0.0190 Hz**	**0.0381**	
	LSTM	0.9577	0.0387 Hz	0.0147 Hz	0.0352	0.9045	0.0254 Hz	0.0196 Hz	0.0393	
Voltage V1-2 (timescale 60 s)	FFNN	0.9245	**0.7362 V**	**0.5660 V**	**0.1331**	0.8734	1.1462 V	0.8234 V	0.1984	
	RBFN	**0.9277**	0.7493 V	0.5728 V	0.1378	**0.8857**	**1.0882 V**	**0.8234 V**	**0.1984**	
	LSTM	0.8994	0.9498 V	0.7954 V	0.2345	0.8394	1.956 V	2.0435 V	0.5358	**2-step ahead**
Frequency (timescale 60 s)	FFNN	**0.9000**	**0.0307 Hz**	**0.0233 Hz**	**0.0465**	**0.8495**	0.0378 Hz	0.0295 Hz	0.0580	
	RBFN	0.9000	0.031 Hz	0.0238 Hz	0.0476	0.8483	**0.0377 Hz**	**0.0289 Hz**	**0.0579**	
	LSTM	0.8672	0.0566 Hz	0.0940 Hz	0.0875	0.8203	0.0425 Hz	0.0320 Hz	0.0665	

Table 2
Features of the offline-trained FFNNs based one-step and multistep ahead forecasters for different time scales.

Parameter Forecaster		*One-step ahead*						*Multi-step ahead*					
		TrF	N_h	N_e	Data division	TF_h	TF_o	TrF	N_h	N_e	Data division	TF_h	TF_o
183 ms	FFNN (V)	trainlm	4	1500	(80,10,10)%	logsig	purelin	trainbr	5	1200	(80,10,10)%	logsig	purelin
	FFNN (F)	trainlm	4	1500	(70,15,15)%	logsig	purelin	trainlm	4	1500	(70,15,15)%	logsig	purelin
1 s	FFNN (V)	trainlm	4	1000	(80,10,10)%	tansig	tansig	trainlm	4	300	(70,15,15)%	logsig	purelin
	FFNN (F)	trainrp	5	1600	(70,15,15)%	logsig	purelin	trainlm	4	300	(70,15,15)%	logsig	purelin
10 s	FFNN (V)	trainlm	4	1100	(80,10,10)%	tansig	tansig	trainbr	4	300	(80,10,10)%	tansig	purelin
	FFNN (F)	trainlm	6	1000	(70,15,15)%	tansig	purelin	trainlm	4	300	(70,15,15)%	logsig	purelin
60 s	FFNN (V)	trainbr	4	500	(80,10,10)%	logsig	purelin	trainbr	4	1500	(80,10,10)%	logsig	purelin
	FFNN (F)	trainbr	4	800	(80,10,10)%	tansig	purelin	trainlm	5	400	(80,10,10)%	logsig	purelin

$\boldsymbol{N_h}$ is the number of hidden nodes. ***TrF*** is the Training function, $\boldsymbol{N_e}$ is the number of epochs, $\boldsymbol{TF_h}$ is the hidden layer transfer function, $\boldsymbol{TF_o}$ is the output layer transfer function.

Table 3
Features of the online-trained ENNs based one-step and multistep ahead forecasters for different time scales.

Parameter Forecaster		*One-step ahead*				*Multi-step ahead*			
		Learning rate	N_h	TF_h	TF_o	Learning rates (α_l)	N_h	TF_h	TF_o
183 ms	ENN (V)	0.42	5	logsig	purelin	(0.52,0.455,0.455)	5	logsig	purelin
	ENN (F)	0.9	5	logsig	purelin	(0.72,0.72,0.8)	5	logsig	purelin
1 s	ENN (V)	0.42	5	logsig	purelin	(0.35,0.33,0.35)	5	logsig	logsig
	ENN (F)	0.98	5	logsig	purelin	(1.93,1.92,2.2)	5	logsig	purelin
10 s	ENN (V)	0.6	3	logsig	logsig	(0.35,0.35,0.35)	3	logsig	logsig
	ENN (F)	0.35	3	logsig	purelin	(0.62,0.6,0.6)	3	logsig	purelin
60 s	ENN (V)	0.008	4	logsig	logsig	(0.0082,0.008)	4	logsig	logsig
	ENN (F)	0.2	4	logsig	purelin	(0.05,0.1)	4	logsig	purelin

5.2.1. One step ahead forecasting

Table 4 and the left side of Tables 5 and 6 show the error metrics regarding the one-step ahead forecasters based on FFNNs and ENNs for different time scales (183 ms, 1 s, 10 s, 60 s).

With reference to the frequency prediction, the FFNNs and ENNs-based models perform well for all the considered time scales and periods. Moreover, we observe that the frequency forecast errors increase with the time horizon and the RMSEs are smaller than 0.055 Hz. The 1 s, 10 s and 60 s frequency FFNNs-based forecasters have shown the best performance with a correlation factor greater than 84% and a MAPE smaller than 0.052%. With reference to the 1 s-frequency prediction, the FFNN-based forecaster performs much better than the ENN-based one.

On the other hand, the voltage forecasters perform well as the maximum MAPE is 0.215%; this refers to the 60 s ENN-voltage predictor. The ENNs models used for the 183 ms, 1 s and 10 s

Table 4
Errors metrics calculated for the 1-step forecasters with four different time horizons (period: March).

Parameter / Forecaster		One-step ahead prediction			
		R	RMSE	MAE	MAPE (%)
Voltage V_{1-2} (timescale 183 ms)	FFNN	0.9361	0.6451 V	**0.4420** V	**0.1063**
	ENN	**0.9554**	**0.5417** V	0.4446 V	0.1069
Frequency (timescale 183 ms)	FFNN	0.9977	0.0062 Hz	0.0032 Hz	0.0064
	ENN	**0.9991**	**0.0033 Hz**	**0.0026 Hz**	**0.0052**
Voltage V_{1-2} (timescale 1 s)	FFNN	0.8801	0.7295 V	0.5562 V	0.1334
	ENN	**0.9281**	**0.5599** V	**0.4641** V	**0.1113**
Frequency (timescale 1 s)	FFNN	**0.9964**	**0.0055** Hz	**0.0036** Hz	**0.0072**
	ENN	0.9785	0.0130 Hz	0.0101 Hz	0.0201
Voltage V_{1-2} (timescale 10 s)	FFNN	0.9356	0.7758 V	**0.5791** V	**0.1396**
	ENN	**0.9363**	**0.7714 V**	0.6017 V	0.1449
Frequency (timescale 10 s)	FFNN	**0.9704**	**0.0172 Hz**	**0.0130 Hz**	**0.0260**
	ENN	0.9447	0.0235 Hz	0.0175 Hz	0.0349
Voltage V_{1-2} (timescale 60 s)	FFNN	**0.9355**	**0.8784** V	**0.6286** V	**0.1519**
	ENN	0.9216	0.9707 V	0.7084 V	0.1711
Frequency (timescale 60 s)	FFNN	**0.8759**	**0.0342 Hz**	**0.0257 Hz**	**0.0515**
	ENN	0.7000	0.0531 Hz	0.0404 Hz	0.0807

Table 5
Statistical errors between measured and forecasted grid quantities for different time horizons (period: April).

Parameter / Forecaster		One-step ahead prediction				Multi-step ahead prediction				
		R	RMSE	MAE	MAPE (%)	R	RMSE	MAE	MAPE(%)	
Voltage V_{1-2} (183 ms)	FFNN	0.7700	0.6571 V	0.5197 V	0.1263	0.6208	0.6720 V	0.5397 V	0.1312	
	ENN	**0.8000**	**0.5388 V**	**0.4445 V**	**0.1080**	**0.7400**	**0.5809 V**	**0.4990 V**	**0.1213**	
Frequency (183 ms)	FFNN	0.9987	0.0041 Hz	0.0026 Hz	0.0052	**0.9989**	**0.0036 Hz**	**0.0026 Hz**	**0.0052**	
	ENN	**0.9988**	**0.0038 Hz**	**0.0030 Hz**	**0.0061**	0.9977	0.0049 Hz	0.0037 Hz	0.0074	
Voltage V_{1-2} (timescale 1 s)	FFNN	0.9225	0.7591 V	0.5808 V	0.1407	0.9260	0.7404 V	0.5612 V	0.1360	
	ENN	**0.9507**	**0.5907 V**	**0.4873 V**	**0.1180**	**0.9486**	**0.6023 V**	**0.5009 V**	**0.1213**	**3-step ahead**
Frequency (timescale 1 s)	FFNN	**0.9970**	**0.0048 Hz**	**0.0037 Hz**	**0.0075**	**0.9940**	**0.0069 Hz**	**0.0051 Hz**	**0.0103**	
	ENN	0.9751	0.0137 Hz	0.0108 Hz	0.0215	0.9920	0.0078 Hz	0.0060 Hz	0.0120	
Voltage V_{1-2} (timescale 10 s)	FFNN	**0.9144**	**0.8009 V**	**0.6160 V**	**0.1492**	0.9000	0.8609 V	0.6429 V	0.1557	
	ENN	0.9000	0.8600 V	0.6841 V	0.1657	**0.9300**	**0.7001 V**	**0.5510 V**	**0.1335**	
Frequency (timescale 10 s)	FFNN	**0.9562**	**0.0181 Hz**	**0.0139 Hz**	**0.0278**	0.9251	0.0236 Hz	0.0179 Hz	0.0357	
	ENN	0.9119	0.0257 Hz	0.0193 Hz	0.0386	**0.9463**	**0.0203 Hz**	**0.0152 Hz**	**0.0304**	
Voltage V_{1-2} (timescale 60 s)	FFNN	**0.8443**	**1.0089 V**	**0.7242 V**	**0.1754**	0.7247	1.6104 V	1.3428 V	0.3258	
	ENN	0.8200	1.0838 V	0.8875 V	0.2150	**0.8476**	**0.9999 V**	**0.8136 V**	**0.1971**	**2-step ahead**
Frequency (timescale 60 s)	FFNN	**0.8500**	**0.0343 Hz**	**0.0260 Hz**	**0.0512**	**0.8167**	**0.0361 Hz**	**0.0287 Hz**	**0.0574**	
	ENN	0.7000	0.0499 Hz	0.0392 Hz	0.0784	0.6363	0.0562 Hz	0.0432 Hz	0.0864	

Table 6
Errors metrics calculated for the one and multi-step forecasters with four different time horizons (period: May).

Parameter / Forecaster		One-step ahead prediction				Multi-step ahead prediction				
		R	RMSE	MAE	MAPE (%)	R	RMSE	MAE	MAPE(%)	
Voltage V_{1-2} (183 ms)	FFNN	0.7716	0.8578 V	0.7179 V	0.1756	0.6000	0.8131 V	0.6333 V	0.1540	
	ENN	**0.8200**	**0.5631 V**	**0.4715 V**	**0.1152**	**0.7200**	**0.5597 V**	**0.4639 V**	**0.1135**	
Frequency (183 ms)	FFNN	**0.9977**	**0.0021 Hz**	**0.0017 Hz**	**0.0034**	**0.9987**	**0.0024 Hz**	**0.0019 Hz**	**0.0039**	
	ENN	0.9934	0.0036 Hz	0.0028 Hz	0.0055	0.9983	0.0028 Hz	0.0021 Hz	0.0043	
Voltage V_{1-2} (timescale 1 s)	FFNN	0.6600	1.0905 V	0.8618 V	0.2103	0.6500	0.9884 V	0.6393 V	0.1559	
	ENN	**0.8336**	**0.5559 V**	**0.4631 V**	**0.1130**	**0.8145**	**0.5772 V**	**0.4805 V**	**0.1174**	**3-step ahead**
Frequency (timescale 1 s)	FFNN	**0.9963**	**0.0039 Hz**	**0.0030 Hz**	**0.0061**	**0.9891**	**0.0059 Hz**	**0.0043 Hz**	**0.0086**	
	ENN	0.9646	0.0121 Hz	0.0092 Hz	0.0185	0.9827	0.0074 Hz	0.0056 Hz	0.0111	
Voltage V_{1-2} (timescale 10 s)	FFNN	**0.9000**	**0.8846 V**	**0.6734 V**	**0.1635**	0.9014	0.8805 V	0.6754 V	0.1641	
	ENN	0.9134	0.8268 V	0.6537 V	0.1589	**0.9360**	**0.6768 V**	**0.5369 V**	**0.1304**	
Frequency (timescale 10 s)	FFNN	**0.9450**	**0.0170 Hz**	**0.0135 Hz**	**0.0271**	0.9203	0.0204 Hz	0.0161 Hz	0.0321	
	ENN	0.9172	0.0208 Hz	0.0164 Hz	0.0327	**0.9424**	**0.0176 Hz**	**0.0140 Hz**	**0.0279**	
Voltage V_{1-2} (timescale 60 s)	FFNN	0.9042	0.8949 V	0.7231 V	0.1760	0.7000	1.2933 V	0.8755 V	0.2119	
	ENN	**0.9001**	**0.8833 V**	**0.6918 V**	**0.1683**	**0.9217**	**0.8108 V**	**0.6443 V**	**0.1567**	**2-step ahead**
Frequency (timescale 60 s)	FFNN	**0.8460**	**0.0282 Hz**	**0.0224 Hz**	**0.0449**	**0.8096**	**0.0309 Hz**	**0.0247 Hz**	**0.0494**	
	ENN	0.7000	0.0399 Hz	0.0318 Hz	0.0635	0.6400	0.0485 Hz	0.0382 Hz	0.0764	

Table 7
Errors metrics calculated for the forecasters using multi-inputs and four different time horizons.

Parameter Forecaster		One-step ahead prediction					Multi-step ahead prediction				
		R	RMSE (V)	MAE (V)	MAPE (%)		R	RMSE (V)	MAE (V)	MAPE(%)	
Voltage V_{1-2} (timescale 183 ms)	**1 input**	0.9484	0.6116	0.4049	0.0975	**1 input**	0.9537	0.7043	0.5480	0.1318	
	2 input	0.9545	0.5749	0.4236	0.1020	**2 input**	0.9669	0.5928	0.4645	0.1117	
	3 input	**0.9650**	**0.5058**	**0.3772**	**0.0908**	**3 input**	0.9709	0.5558	0.4373	0.1051	
	4 input	0.9640	0.5151	0.4130	0.0995	**4 input**	**0.9728**	**0.5383**	**0.4282**	**0.1030**	
Voltage V_{1-2} (timescale 1 s)	**1 input**	0.9148	0.7234	0.5773	0.1388	**1 input**	0.9181	0.7086	0.5334	0.1282	
	2 input	0.9366	0.6230	0.5129	0.1233	**2 input**	0.9401	0.6039	0.4865	0.1169	**3-step**
	3 input	0.9476	0.5671	0.4873	0.1171	**3 input**	0.9468	0.5714	0.4814	0.1157	**ahead**
	4 input	**0.9500**	**0.5619**	**0.4648**	**0.1116**	**4 input**	**0.9480**	**0.5694**	**0.4695**	**0.1128**	
Voltage V_{1-2} (timescale 10 s)	**1 input**	0.9512	0.7566	0.5450	0.1316	**1 input**	0.9391	0.8773	0. 6660	0. 1607	
	2 input	0.9612	0.6731	0.5167	0.1247	**2 input**	0.9535	0.7571	0.5951	0.1436	
	3 input	0.9629	0.6575	0.5103	0.1231	**3 input**	0.9540	0.7388	0.5662	0.1366	
	4 input	**0.9653**	**0.6490**	**0.5099**	**0.1230**	**4 input**	**0.9585**	**0.7003**	**0.5461**	**0.1318**	
Voltage V_{1-2} (timescale 60 s)	**1 input**	0.9378	0.8467	0.6065	0.1464	**1 input**	0.9166	1.4313	1.0777	0.2604	
	2 input	0.9433	0.8083	0.5963	0.1439	**2 input**	0.9291	1.2571	0.9366	0.2270	**2-step**
	3 input	**0.9468**	**0.7835**	**0.5946**	**0.1435**	**3 input**	0.9317	**1.1687**	**0.8612**	**0.2086**	**ahead**
	4 input	0.9455	0.8195	0.6231	0.1504	**4 input**	**0.9392**	1.2050	0.8946	0.2168	

voltage predictions have shown a good performance for all the considered time periods. Besides, for the time scales of 183 ms and 1 s, the ENNs models perform better in terms of error metrics than the FFNNs ones. Moreover, the 60 s voltage predictions using the FFNN and the ENN models have acceptable MAEs and MAPEs.

Figs. 7–9 show the comparisons between the measured and the predicted values of the grid voltage and frequency for the one-step ahead forecasters considering the different time periods (March, April and May) and horizons (183 ms, 1 s, 10 s and 60 s). The correlation between the measured and the forecasted quantities is good for both the online-trained and offline-trained forecasters.

5.2.2. Multi-step ahead forecasting

The error indices used to evaluate the performance of the multi-step forecasters are presented in the right side of Tables 5 and 6. All forecasters perform well, albeit slightly worse than the single-step forecasters, which is expected due to the greater uncertainty associated with multiple time step predictions.

The maximum MAPE is 0.32%, and the maximum RMSE is 1.61 V. Both correspond to the 60 s FFNN-voltage forecaster. The correlation coefficients in the range (0.6 – 0.99) are acceptable. As in the case of one-step forecasters, we notice that the forecast accuracy decreases with the time scale. The 183 ms and 1 s frequency forecasters (FFNNs and ENNs) show the best performance in term of accuracy (RMSE and MAE). For the four timescales, the ENNs voltage estimators give the best results in comparison to FFNNs forecasters (Tables 5 and 6). It can be seen also that more accurate prediction of frequency, for 10 s scale, is achieved with ENN model. Meanwhile, the FFNNs outperform the online-trained ENNs in the case of 183 ms, 1 s and 60 s frequency forecasts.

Figs. 10 and 11 show a very good correlation between the 3step frequency forecasts and the measured data. Fig. 11 also shows that for the time scales of 183 ms and 1 s, the FFNNs-based voltage predictors are the worst performers. Besides, Fig. 12 illustrates how the adopted ENNs and FFNNs have fit adequately the test data for the case of one-step voltage forecast (60 s timescale) and three-step frequency prediction (for 10 s timescale).

In summary, the results show that the predictions based on current information are better for the frequencies than the voltages. This is due to the random nature of the voltage profile that is more difficult to model, especially in the case of the offline-trained FFNNs. This is why the FFNN-based voltage forecasters need to be retrained in order to take into account the dynamic change in grid voltage profile. For this reason, in the next subsection we will show a study on the impact of the input variables on the voltage forecast accuracy.

5.3. Impact of the input variables on the voltage forecast accuracy

As shown in Fig. 13, in order to evaluate the performance of the FFNN-based voltage forecasters, we consider three different multi-input models for both the one and multistep ahead predictions. The input variables defined for each FFNN-based model are:

- For the first model, the actual and past observations: $V_{12}(k)$ and $V_{12}(k-1)$. The input–output relationship can be expressed as:

$$\begin{cases} \hat{V}_{12}(k+1) = f_{1S}^{1}\left(V_{12}(k), V_{12}(k-1)\right) \\ \left\{\hat{V}_{12}(k+1), \hat{V}_{12}(k+2), \hat{V}_{12}(k+3)\right\} = f_{MS}^{1}\left(V_{12}(k), V_{12}(k-1)\right) \end{cases} \tag{6}$$

  where f_{1S}^{1} and f_{MS}^{1} are the nonlinear approximation functions for the one step and the multistep predictors respectively.
- For the second model, we consider three points of the voltage trajectory $V_{12}(k)$, $V_{12}(k-1)$, and $V_{12}(k-2)$:

$$\begin{cases} \hat{V}_{12}(k+1) = f_{1S}^{2}\left(V_{12}(k), V_{12}(k-1), V_{12}(k-2)\right) \\ \left\{\hat{V}_{12}(k+1), \hat{V}_{12}(k+2), \hat{V}_{12}(k+3)\right\} = f_{MS}^{2}\left(V_{12}(k), V_{12}(k-1), V_{12}(k-2)\right) \end{cases} \tag{7}$$

  where f_{1S}^{2} and f_{MS}^{2} are the nonlinear approximation functions for the one step and the multistep predictors respectively. The knowledge of the two previous values of the voltage gives more information regarding the interval and the direction of the variation of the voltage.
- For the third model, the input variables are four: the actual three phase line-to-line voltages $V_{12}(k)$, $V_{13}(k)$, $V_{23}(k)$, and the past value $V_{12}(k-1)$:

$$\begin{cases} \hat{V}_{12}(k+1) = f_{1S}^{3}\left(V_{12}(k), V_{12}(k-1), V_{23}(k), V_{13}(k)\right) \\ \left\{\hat{V}_{12}(k+1), \hat{V}_{12}(k+2), \hat{V}_{12}(k+3)\right\} = f_{MS}^{3}\left(V_{12}(k), V_{12}(k-1), V_{23}(k), V_{13}(k)\right) \end{cases} \tag{8}$$

  where f_{1S}^{3} and f_{MS}^{3} are the nonlinear approximation functions for the one step and the multistep predictors respectively.

In order to compare the three models we used the same features and learning parameters used for the single-input FFNNs (listed in Table 2). The initialization of weights and biases is performed using the Nguyen Widrow method '*initnw*', while the '*mapminmax*'

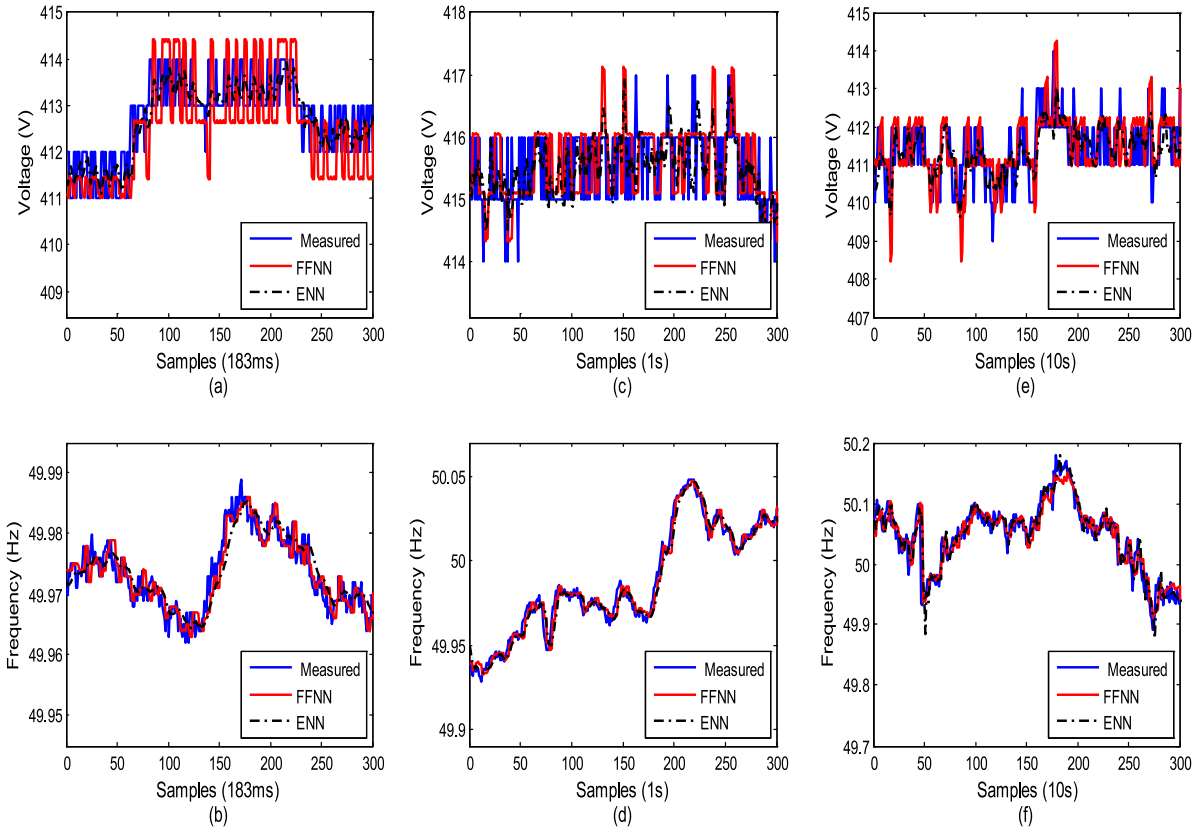


Fig. 10. Real-time simulation results for the three-step ahead forecasters with four time horizons (183 ms, 1 s, and 10 s, period: April).

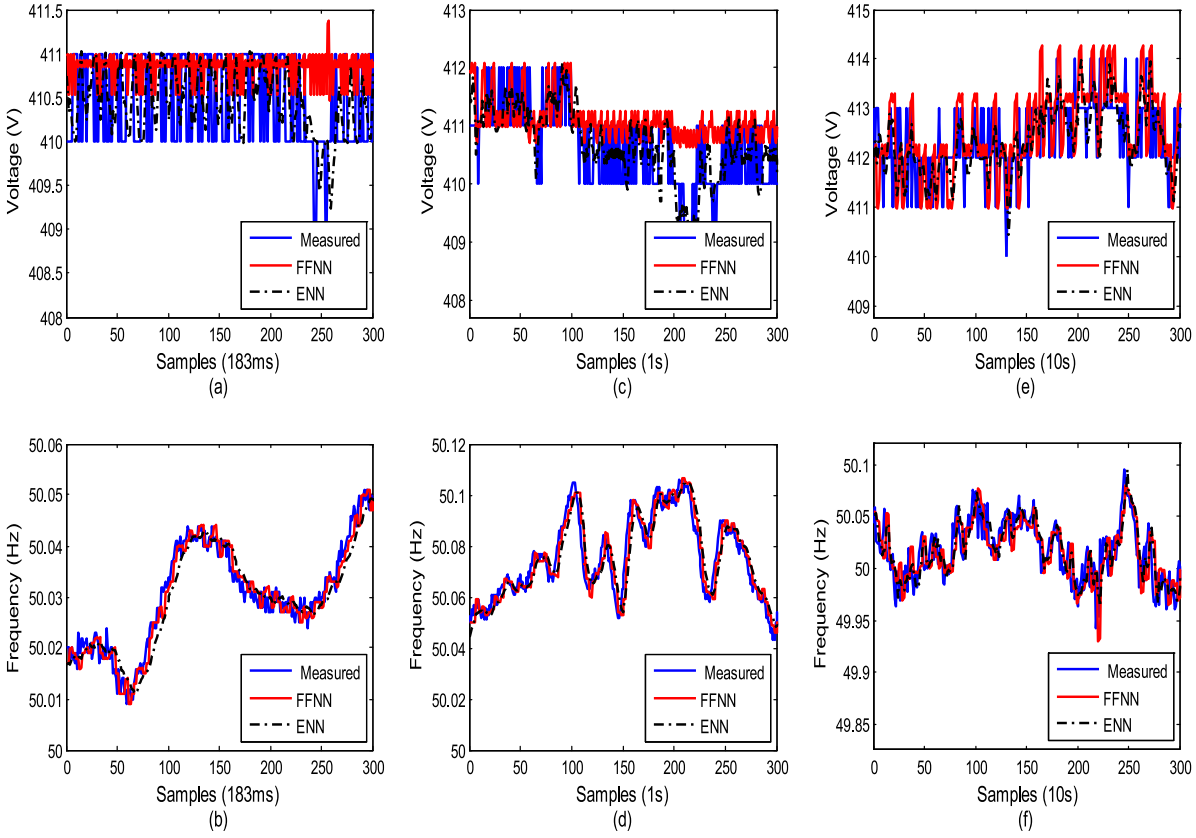


Fig. 11. Real-time simulation results for the three-step ahead forecasters with four time horizons (183 ms, 1 s, and 10 s, period: May).

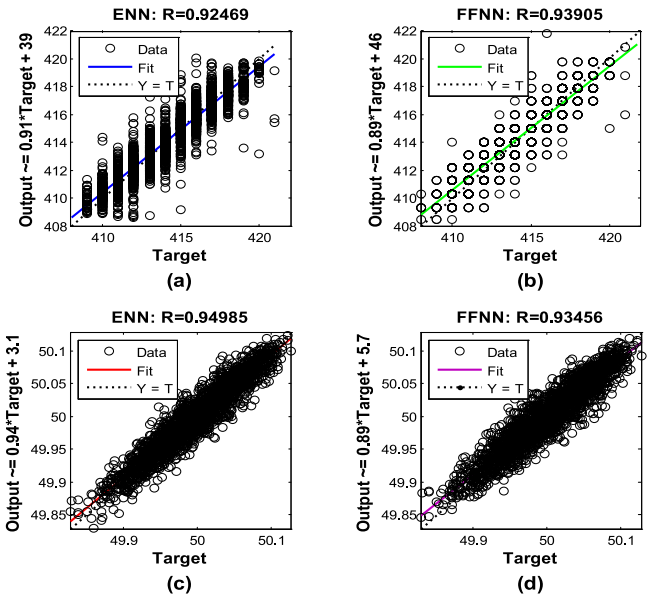


Fig. 12. Comparison between predicted and measured data for: (a) one-step ENN voltage forecaster (60 s), (b) one-step FFNN voltage forecaster (60 s), (c) 3-step ENN frequency forecaster (10 s), (d) 3-step FFNN frequency forecaster (10 s).

function is used for the input–output normalization. In order to obtain the optimal FFNN-based forecaster, the training is repeated 10 times for each case with a random initialization of the FFNN parameters. After the training process, the optimal multi-input FFNN forecasters are simulated for the same test dataset. The statistical errors calculated for each time scale are reported in Table 7.

The 3 and 4-input FFNN models show the best performance in terms of correlation coefficient and error metrics, with a significant improvement in comparison with the one-input FFNN forecasters. In the case of 1 s and 10 s one-step and 3-step ahead prediction, the 4-input FFNNs give the most accurate results with a correlation factor greater than 94% and a RMSE smaller than 0.71 V. The 3-input FFNNs work more precisely in the case of 60 s one-step and 2-step voltage forecasts. This means that the use of more information about the historical voltage profile, as well as current observations of three-phase voltage components, help the FFNN to learn more effectively the relationship between the inputs and outputs.

6. Conclusions

This paper proposes a method for very-short real-time forecast of grid voltage and frequency based on ANNs. This type of forecasters can be used in an advanced control schemes as well as in monitoring systems of distributed generators. To the best of our knowledge, this is the first work that proposes the multistep ahead prediction of grid quantities. The key benefit of the multistep forecasting is to prevent grid faults caused by random change of load demand or power generation.

In this study, a comparison between different types of offline-trained ANNs based forecast models (FFNNs, RBFNs and LSTMs) is performed. The performance of ANNs predictors have been checked through simulations for different time scales. It has been demonstrated that FFNNs and RBFNs based models can accurately predict the grid voltage and frequency with a big similarity in the results. The main difference between the developed forecast models is the architecture complexity of neural networks.

In comparison to other works in this area, the presented FFNN and ENN predictors have been validated in real-time, for different time scales (183 ms, 1 s, and 10 s), and using real-time measurements of grid voltage and frequency. The prediction accuracy of the developed one-step and multi-step ahead forecasters, implemented on a dSPACE controller, has been demonstrated for the scales of 183 ms, 1 s, 10 s, and 60 s. It has been seen that the frequency forecasters perform better than the voltage forecasters for all investigated ANNs. The best performance was from the 183 ms and 1 s frequency forecasters, which yield values of RMSE and MAE close to zero for both case of one step and three-step ahead prediction. The precision of the forecasters degrades as the time horizons becomes larger. In the case of voltage prediction, we found that the accuracy of offline trained FFNNs, using current observation only, needs to be improved. Thus, three different FFNN models are investigated with different combinations of input variables to study their impact on grid voltage forecast. We observed an improvement in the prediction accuracy with the three-input and four-input FFNN models.

The main advantages of the developed ANNs-based forecasters are related to the fact that they do not need any mathematical modeling of the physical system, they have a simple structure, and are easy to implement on low cost hardware. Moreover, the key benefit of the adaptive ENN is its ability to give correct and accurate results based on online learning rule, while avoiding the time consuming task of offline training. The performance of the developed forecasters can be enhanced, especially in the case of long time scales (i.e. 60 s), by using advanced training algorithms and considering other input variables (like active and reactive powers).

CRediT authorship contribution statement

N. Chettibi: Software, Visualization, Investigation, Validation, Writing - original draft, Writing - review & editing. **A. Massi Pavan:** Conceptualization, Methodology, Writing - original draft, Data curation, Writing - review & editing. **A. Mellit:** Conceptualization, Methodology, Writing - review & editing. **A.J. Forsyth:** Supervision, Conceptualization, Writing - review & editing. **R. Todd:** Supervision, Data curation, Writing - original draft, 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.

Acknowledgments

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) Multi-scale ANalysis for Facilities for Energy STorage (Manifest) project [EP/N032888/1].

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 program Interreg Italy-Croatia.

A. Mellit would like to thank the International Centre for Theoretical Physics (ICTP), Trieste (Italy) for providing the materials and the computer facilities used to develop some parts of the work presented in the paper.

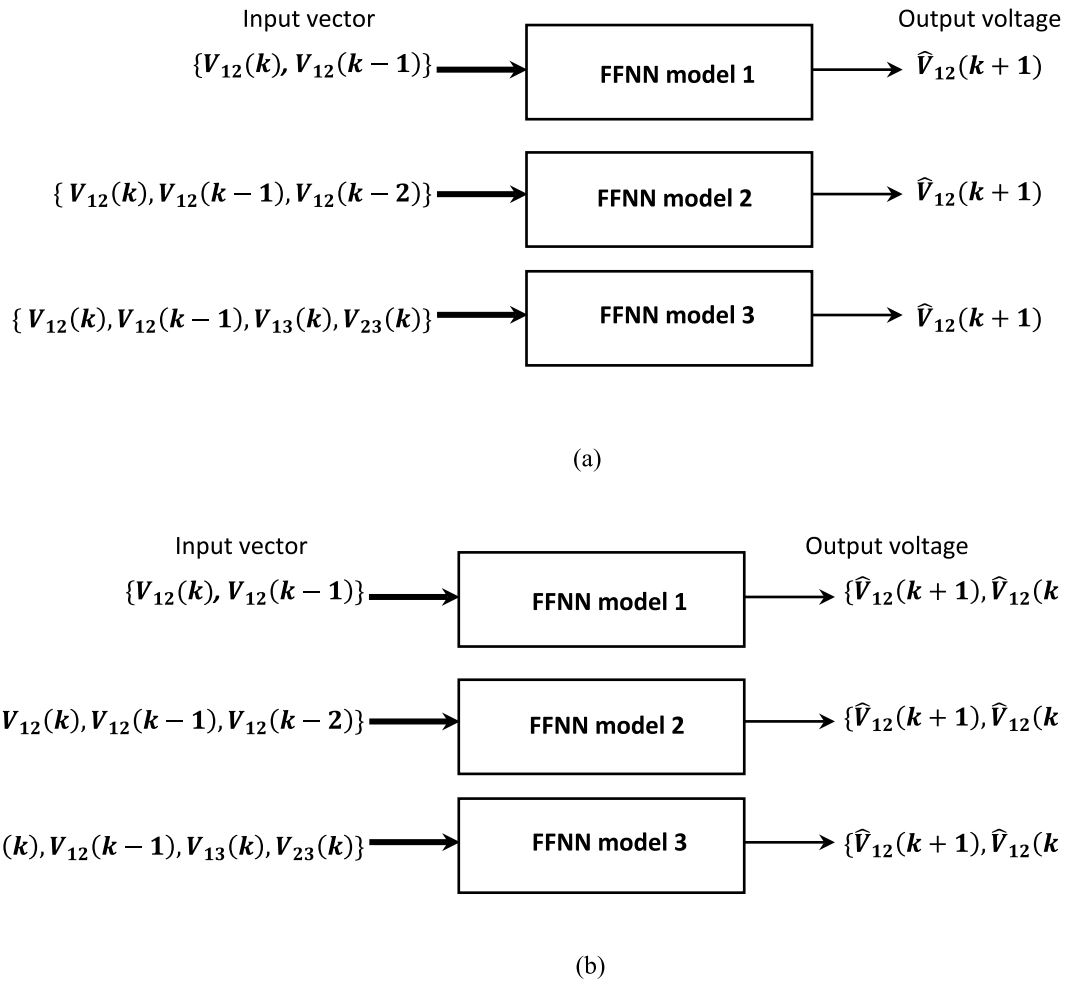


Fig. 13. Configuration of the multi-input FFNN models proposed for the voltage prediction: (a) one step ahead, (b) two-step ahead.

References

[1] IRENA Renewable capacity statistics, 2019, (accessed on 06 august 2019).
[2] A. Nespoli, E. Ogliari, S. Leva, A. Massi Pavan, A. Mellit, V. Lughi, A. Dolara, Day-ahead photovoltaic forecasting: a comparison among the most effective techniques, Energies 12 (2019) 1–15.
[3] H.-J. Lee, J.-C. Kim, S.-M. Cho, Optimal volt–var curve setting of a smart inverter for improving its performance in a distribution system, IEEE Access 8 (2020) 157931-157945, http://dx.doi.org/10.1109/ACCESS.2020.3019794.
[4] S. Paudyal, O. Ceylan, B.P. Bhattarai, K.S. Myers, Optimal coordinated EV charging with reactive power support in constrained distribution grids, in: IEEE Power & Energy Society General Meeting, USA, 2017.
[5] M.E. Khodayar, M. Barati, M. Shahidehpor, Integration of high reliability distribution system in microgrid operation, IEEE Trans. Smart Grids 3 (2012) 1997–2006.
[6] J. Kruse, B. Schäfer, D. Witthaut, Predictability of power grid frequency, IEEE Access 8 (2020) 149435–149446.
[7] F. Bignucolo, A. Cerretti, M. Coppo, A. Savio, R. Turri, Effects of energy storage grid code requirements on interface protection performances in low voltage networks, Energies 10 (2017) 387–407.
[8] M. Corti, G. Superti Furga, E. Tironi, A. Massi Pavan, G. Sulligoi, Intentional islanding in medium voltage distribution grids, in: the 16th International Conference on Environment and Electrical Engineering, EEEIC, 2016.
[9] J. and Ma, X. and Djouadi, S.M.. Dong, H. Li, Y.Liu, Frequency prediction of power systems in FNET based on state-space approach and uncertain basis functions, IEEE Trans. Power Syst. 29 (2014) 2602–2612.
[10] Y.J.A. Zhang, C. Zhao, W. Tang, S.H. Low, Profit-maximizing planning and control of battery energy storage systems for primary frequency control, IEEE Trans. Smart Grid 9 (2016) 712–723.
[11] F. Hafiz, M.A. Awal, A.R. De Queiroz, I. Husain, Real-time stochastic optimization of energy storage management using deep learning based forecasts for residential PV applications, IEEE Trans. Ind. Appl. 46 (2020) 2216–2226.
[12] M. Mousavi, M. Rayati, A.M. Ranjbar, Optimal operation of a virtual power plant in frequency constrained electricity market, IET Gen. Transm. Distrib. 13 (2018) 2123–2133.
[13] M. T.Alrifai, M. Zribi, M. Rayan, M.S. Mahmoud, On the control of time delay power systems', Int. J. Innovative Comput. Inf. Control 2013 (2013) 769–792.
[14] R. Dhand, G. Lee, G. Cole, Communication delay modelling and its impact on real-time distributed control systems, in: International Conference on Advanced Engineering Computing and Applications in Sciences, ADVCOMP, 2010.
[15] I. Jayawardene, G. K.Venayagamoorthy, Cellular computational extreme learning machine network based frequency predictions in a power system, in: 2017 IEEE International Joint Conference on Neural Networks, IJCNN, 2017, pp. 3377–3384.
[16] S. Kaur, Y. Pal Verma, L. S.Agrawa, Optimal generation scheduling in power system using frequency prediction through ANN under ABT environment, Front. Energy 7 (4) (2013) 468–478.
[17] S. Tripathi, S. De, Dynamic prediction of power line frequency for wide area monitoring and control, IEEE Trans. Ind. Inf. 14 (2017) 2837–2846.
[18] Y. Wei, G.K. Venayagamoorthy, A lite cellular generalized neuron network for frequency prediction of synchronous generators in a multimachine power system, in: IEEE International Joint Conference on Neural Networks, IJCNN, 2016, pp. 3085–3092.
[19] J. Zhou, Z. Wang, M. Chen, Z. Yang, W. Liu, Combined voltage forecasting method based on EMD-CNN for distribution networks with distributed PVs, in: 2019 IEEE Sustainable power and energy conference, iSPEC, Beijing, China, 2019.
[20] R. Dobbe, W. Van Westering, X.S. Liu, D. Arnold, D.S. Callaway, C. Tomlin, Linear single-and three-phase voltage forecasting and Bayesian state estimation with limited sensing, IEEE Trans. Power Syst. 35 (2020) 1674–1683.
[21] M. Pertl, P.J. Douglass, K. Heussen, K. Kok, Validation of a robust neural real-time voltage estimator for active distribution grids on field data, Electr. Power Syst. Res. 154 (2018) 182–192.

[22] M. Ferdowsi, A. Löwen, P. McKeever, A. Monti, F. Ponci, A. Benigni, New monitoring approach for distribution systems, in: IEEE International Instrumentation and Measurement Technology Conference, I2MTC, 2014.
[23] A. Massi Pavan, N. Chettibi, A. Mellit, T. Feehally, A.J. Forsyth, R. Todd, An ANN-based grid voltage and frequency forecaster, in: IET 9th International Conference on Power Electronics, Machines and Drives, 2018, pp. 1-6.
[24] Z.A. Arfeen, M. Kermadi, M.K. Azam, T.A. Siddiqui, Z.U. Akhtar, M. Ado, Md P. Abdullah, Insights and trends of optimal voltage-frequency control DG-based inverter for autonomous microgrid: State-of the art review, Int. Trans. Electr. Energy Syst. 30 (2020) e12555.
[25] H. Kanchev, D. Lu, F. Colas, V. Lazarov, B. Francois, Energy management and operational planning of a microgrid with a PV-based active generator for smart grid applications, IEEE Trans. Ind. Electron. 58 (10) (2011) 4583–4592.
[26] S. Chandra, P. Gaur, Pathak. D., Radial basis function neural network based maximum power point tracking for photovoltaic brushless DC motor connected water pumping system, Comput. Electr. Eng. 86 (2020) 106730.
[27] J. Brownlee, Deep learning for time series forecasting: Predict the future with MLPs, CNNs and LSTMs in python, in: Machine Learning Mastery, 2020, pp. 42–52.
[28] J. Kaushal, P. Basak, Power quality control based on voltage sag/swell, unbalancing, frequency, THD and power factor using artificial neural network in PV integrated AC microgrid, Sustain. Energy Grids Netw. 23 (2020) 100365.
[29] A. Mellit, A. Massi Pavan, E. Ogliari, S. Leva, V. Lughi, Advanced methods for photovoltaic output power forecasting: A review, Appl. Sci. 10 (487) (2020).
[30] T. Feehally, A.J. Forsyth, R. Todd, M.P. Foster, D. Gladwin, D.A. Stone, D. Strickland, Battery energy storage systems for the electricity grid: UK research facilities, in: 8th IET International Conference on Power Electronics, Machines and Drives, PEMD, April 2016, 2016.
[31] A. Andalib, F. Atry, Multi-step ahead forecasts for electricity prices using NARX: a new approach, a critical analysis of one-step ahead forecasts, Energy Convers. Manage. 50 (2009) 739–747.
[32] A. Gulli, S. Pal, Deep Learning with Keras: Implementing Deep Learning Models and Neural Networks with the Power of Python, Packt Publishing, UK, 2017.
[33] R.D. Brandt, F. Lin, Adaptive interaction and its application to neural networks, Inform. Sci. 121 (1999) 201–215.
[34] N. Chettibi, A. Mellit, G. Sulligoi, A. Massi Pavan, Adaptive neural network-based control of a hybrid AC/DC microgrid, IEEE Trans. Smart Grid 9 (2018) 1667–1679.
[35] S. Ben Taieb, G. Bontempi, A.F. Atiya, A. Sorjamaa, A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition', Expert Syst. Appl. 39 (2012) 7067–7083.

Sustainable Energy, Grids and Networks 31 (2022) 100769

Contents lists available at ScienceDirect

Sustainable Energy, Grids and Networks

journal homepage: www.elsevier.com/locate/segan

Real time Energy Management System of a photovoltaic based e-vehicle charging station using Explicit Model Predictive Control accounting for uncertainties

Ana Cabrera-Tobar [a,*], Alessandro Massi Pavan [b], Nicola Blasuttigh [b], Giovanni Petrone [a], Giovanni Spagnuolo [a]

[a] Department of Information and Electrical Engineering and Applied Mathematics, University of Salerno, Italy
[b] Department of Engineering and Architecture, and Center for Energy, Environment and Transport Giacomo Ciamician — University of Trieste, Italy

ARTICLE INFO

Article history:
Received 22 November 2021
Received in revised form 28 April 2022
Accepted 9 May 2022
Available online 21 May 2022

Keywords:
Optimization
Charging station
EV
Multiparametric
Uncertainties
Model predictive control
MPC
Explicit
Emissions

ABSTRACT

This paper proposes an Explicit Model Predictive Control (eMPC) for the energy management of an e-vehicle charging station fueled by a photovoltaic plant (PV), a battery energy storage system (BESS), and the electrical grid. The method computes offline an explicit solution of the MPC, which is stored and then used for real time control. Multiparametric programming is used to calculate the explicit solution by mapping the MPC laws as a function of uncertain parameters. In this paper, the uncertainties introduced into the multiparametric programming are the photovoltaic power production, the electricity price, the battery's state of charge, and the electric vehicle power consumption. Moreover, the environmental impact of the charging station operation is considered through the CO_2 emissions level. The explicit solution is computed for a specific range of uncertain parameters. Then, during the real-time control, their current values are measured to evaluate the control laws saved beforehand. The proposed approach, consisting of an offline MPC-based determination of the control laws followed by their online use, reduces the computational burden without affecting the MPC performance. Numerical simulations and experimental results confirm the eMPC's performance for the proposed application.

1. Introduction

The market for electric vehicles (EVs) is growing exponentially, with the total number of EVs on the road nearing 10 million in 2020 and expected to reach around 140 million by 2030. It will cause an increase in electricity demand of at least 2% of global electricity consumption by 2030, causing an increment of charging stations for home, work, and commercial purposes of up to 125 million [1].

The use of clean and sustainable energy is mandatory to keep CO_2 emissions at a low value. Consequently, the electrical system must face new challenges to supply the necessary power when requested by the EVs with the lowest possible environmental impact. Hence, suitable solutions are photovoltaic (PV) charging stations, including a battery energy storage system (BESS) connected to the electrical grid. The Energy Management System (EMS) of such stations must provide the energy requested to charge the vehicle with the minimum environmental impact and at the minimum price.

This intelligent charging should consider parameters such as the fluctuation of PV production, the EV charging time, the EV power consumption, the State of Charge (SoC) of the BESS, the electricity price, and the CO_2 emissions related to the energy drawn from the electrical grid. These parameters are commonly treated as certain when scheduling by deterministic optimization is developed, e.g., in [2]. However, in the real time operation of these charging stations, some of these parameters are forecasted with a degree of error, such as PV production, CO_2 emissions, and electricity price, or they are random, like the EV charging time, its arrival, and departure time. The uncertain nature of these parameters could lead to cumulative errors during the day, generating higher environmental and economic costs. Consequently, energy management techniques accounting for uncertainties appear to help improve the energy management quality.

In the literature, techniques such as fuzzy, information gap decision theory, and robust optimization have been applied to manage EV charging stations by accounting for uncertain parameters. In [3], fuzzy optimization is applied, where the energy market price fluctuation, the EV's arrival and departure time, and the EV battery's state of charge are considered uncertainties. However, it does not include an electrical power generator employing renewable energy or a storage unit in this case. In

* Corresponding author.
E-mail address: anacabrera.t@gmail.com (A. Cabrera-Tobar).

https://doi.org/10.1016/j.segan.2022.100769
2352-4677/

Nomenclature

$\alpha_1, \alpha_2, \alpha_3$	Weight values to give priority to the control variables
Γ_1	Limit factor for electricity price (€)
Γ_2	Limit factor for CO_2 emissions (kg)
Θ	Space of uncertainties
θ	Vector of uncertainties
θ_{CO_2}	Uncertainty of CO_2 emissions (kg)
θ_{EV}	EV's demand uncertainty (W)
θ_{pv}	PV power production's uncertainty (W)
θ_p	Uncertainty of electricity price (€)
θ_{SoC}	SoC's uncertainty (%)
A, B, C	Coefficient matrices of the state space model
C_{max}	Maximum battery capacity (sW)
$CO_2(k)$	Instantaneous (CO_2) emissions (kg)
G, W, E	Coefficients of inequalities constraints
k	Time instant (s)
m	Number of state variables
n	Number of control variables
N_c	Control horizon
N_p	Prediction horizon
n_{bat}	Battery efficiency (%)
$p(k)$	Instantaneous electricity price (€)
$P_{bat}(k)$	Instantaneous battery power (W)
$P_{bat}^{min}, P_{bat}^{max}$	Lower and upper bounds of P_{bat} (W)
$P_{EV}(k)$	Instantaneous EV power (W)
$P_{grid}(k)$	Instantaneous grid power (W)
$P_{grid}^{min}, P_{grid}^{max}$	Lower and upper bounds of P_{grid} (W)
P_{inv}	Inverter's power (W)
$P_{pv}(k)$	Instantaneous PV power (W)
Q, H, c	Cost function coefficients with the corresponding dimensions
$SoC(k)$	Instantaneous state of charge of the battery (%)
SoC^{min}, SoC^{max}	Lower and upper bounds of BESS's SoC (%)
T_s	Sampling time (s)
$u(k)$	Control variables
$x(k)$	State space variable
CR	Critical regions
i	Number of CR

contrast, the system studied in [4] includes a PV generator, a storage unit, and a fuel cell. The algorithm used for optimization is information gap decision theory, where the uncertainty accounted for is the load. In [5], it is concluded that affine arithmetic and robust optimization approaches are helpful tools for keeping track of the parametric uncertainties, but at the price of a high computational cost and can result in costly, respectively. However, the downside of these techniques is that the schedule is done once for a certain period. Despite these are accounting for uncertainties, the schedule calculated cannot be changed in real time operation, which leads to suboptimal performance of the charging station.

Thus, Model Predictive Control (MPC) is a more suitable algorithm for real time operation by taking into account uncertainties such as disturbances and the future behavior of the system [6]. The control law is determined by optimizing the value of a suitable objective function by considering several constraints and a predictive model. The control law is computed at every sampling time for a particular horizon time by refreshing the state of the system, and the corresponding prediction values [7].

The MPC sampling time can range from several seconds to several hours. In microgrids that include PV systems and EVs, the performance of the control algorithm can be affected by the uncertainty of the solar irradiance level and by the EV's connection/disconnection time and its power demand. Thus, the smaller the sampling time, the smaller these uncertainties affect the control laws. Unfortunately, this may result in a higher computational burden and lower economic efficiency. For instance, in [8], the authors present a MPC for controlling a microgrid where the optimization algorithm is treated by a multi-integer linear programming (MILP). The MPC-MILP calculates the schedule of the energy flow for the different distributed generators every 15 min and predicts the system's state for one hour. The computation time takes 24 s, and it is repeated every 15 min on an Intel Core 2 Duo CPU, 3 GHz. The scheduling is based on the price forecast, power generation, and load, where no error is assumed. When a disturbance to the parameters occurs, the schedule is recalculated. MPC has been used jointly with other techniques to account for uncertainties. In [9], the use of a robust-MPC for a PV-BESS scheduling is proposed, with the economic incentive revenue being considered uncertain. The latter is modeled as a box uncertainty set to get the control law as a function of its robustness. However, the uncertainty set's limits negatively affect the optimal performance of the PV-BESS. The computational time is around 20 ms using a 4.3 GHz CPU. In [10], a MPC-based chance-constraints stochastic optimization is adopted to manage the power flow in a microgrid. The power generation forecast, the demand, and the EV's connection and disconnection times are the uncertain variables. Although the chance-constraints stochastic optimization benefits from the MPC, this technique significantly affects the computation time, so it is unsuitable for a sampling time lower than one hour.

Explicit MPC (eMPC) reduces the online optimization time and algorithm complexity. This method operates offline to create an explicit function that is a map of the optimal control laws depending on the uncertain parameters and on the active sets related to the given constraints [11]. It inherits the MPC advantages without requiring online optimization. Indeed, during the online operation, it only evaluates the explicit function calculated beforehand, thus offline, by using the actual values of the parameters considered uncertain in the offline MPC based optimization. This approach drastically reduces the online computation time compared to a classical MPC, as stated in [12]. For this purpose, Multi-Parametric Programming (MPP) is a mathematical tool that helps solve the explicit solution of the eMPC problem. The MPP allows creating the optimized control laws as functions of the uncertain parameters, the latter being modeled as bounded ranges of values, thus not as historical sequences of data.

The MPP has been widely used to implement eMPC approaches in batch scheduling, control, and optimization of process system engineering [13]. Instead, in energy, this approach has not been exploited. Few examples use this technique for EMS, accounting for uncertainties in the literature. In [14], MPP has been used to dispatch energy by minimizing a microgrid cost operation. The uncertainties affecting the load consumption, the wind and the PV power production forecasts were accounted for. The computation time for the real time operation is only 34.8 μs in contrast with 372 ms using a traditional online optimization. The offline optimization is carried out on a remote cloud platform, and the real time operation on a digital signal processor. In [15] the same technique is used to manage the energy schedule of a combined heat and power energy system. Uncertainty in a rolling horizon framework affects the demand and state of the power units and

the heating system. Unfortunately, this approach was not applied to a real system. In [16], an eMPC battery charging control based on eMPC was presented, but no parameters were considered uncertain.

The eMPC has three main advantages compared to other algorithms, (i) it allows having short computation times during the online application of the control rules, so that this phase can be implemented on hardware with low computing capabilities, (ii) it is easy to code, (iii) it accounts for uncertainties. Thus, it is suitable for real time energy management applied in nano and microgrids where the variation of renewable energy and load can affect the energy scheduling and its optimization.

In this paper, the eMPC is applied for the first time to a PV/BESS charging station by keeping into account energy, economic and environmental aspects. The proposed eMPC takes into account five uncertainties: (i) the PV power production, (ii) the grid energy price, (iii) the CO_2 emission level, (iv) the EV power consumption, and the (v) BESS's SoC. The eMPC is explicitly formulated by means of MPP. This paper explicitly pays attention to the use of the MPP as part of the mapping strategy when uncertainties are considered and how these specific parameters affect the optimal schedule. Furthermore, the eMPC is applied to a real PV/BESS charging station located at the University of Trieste (Italy)

The contributions of this paper can be summarized as follows:

1. The development of the problem formulation of a PV/BESS charging station suitable for the eMPC framework is introduced.
2. The inclusion of PV production, EV power consumption, CO_2 emissions, electricity price, and BESS's SoC as uncertainties in the MPP paradigm is discussed.
3. The scheduling of the energy flow based on the rolling horizon framework by using MPC and the evaluation of the mapping of control laws saved by the MPP at every time step is proposed.
4. The implementation of the eMPC in a real PV/BESS charging station in two different testbeds, one using a standard PC and the other using the dSPACE Scalexio system as a central controller, is demonstrated.

The remaining of this paper is organized as follows: Section 2 describes the structure of the charging station located at the University of Trieste (Italy). Then, Section 3 presents the model and the problem formulation in the eMPC framework. Section 4 explains the real time control. Afterwards, numerical simulations are described and presented in Section 5. Section 6 presents the experimental studies. Then, Section 7 discusses the advantages and disadvantages of using eMPC on this type of application. Finally, the conclusions are presented in Section 8.

2. Description of the charging station

The University of Trieste (Italy), thanks to the project MUSE [17], has recently installed a PV based charging station including a Sonnen inverter with an embedded battery. The system is connected to the electrical grid feeding the University campus (Fig. 1). The PV array consists of two strings with 7 PV modules each and has a nominal power of 3.9 kWp. The single-phase inverter Sonnen Hybrid 9.5/10 has a maximum power equal to 3.3 kW, and it converts the DC power from the PV array and the Battery to AC. The lithium iron phosphate 10 kWh battery embedded into the Sonnen inverter can deliver a maximum power equal to 3.3 kW.

Each of the two charging stations, one dedicated to EVs and the other to electric bikes, has two sockets. The EV and bike charging stations can have a maximum power of 22 kW and 440 W per socket, respectively. However, the station only charges a specific EV, a Nissan Leaf, for the current study. This car has a 40 kWh battery and an embedded 6.7 kW battery charger.

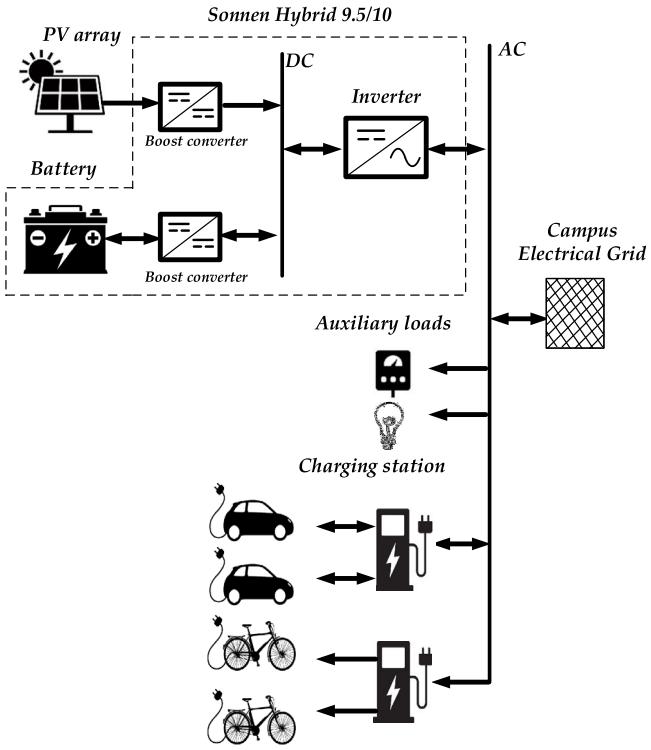


Fig. 1. Main components of the PV based charging station at the University of Trieste.

3. Modeling and problem formulation

To capture the dynamic of the charging station, state space model is commonly used specially for applications in eMPC. This model includes a vector of state variables ($x(k)$) and a vector of control variables ($u(k)$) and takes the following form:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k). \end{aligned} \tag{1}$$

A is the identity matrix, B depends on the coefficients of the control variables and C is the matrix identity so that the output $y(k)$ is equal to state variable $x(k)$ and k is the time step for a discrete state space model.

For the current application, the BESS's SoC ($SoC(k)$) is the state variable, while the battery charging/discharging power ($P_{bat}(k)$) and the power from/to the main grid ($P_{grid}(k)$) are the control variables ($u(k)$). Thus, the vectors appearing in the state space model are:

$$x(k) = [SoC(k)] \tag{2}$$

$$u(k) = \begin{bmatrix} P_{bat}(k) & P_{grid}(k) \end{bmatrix}. \tag{3}$$

The system's dynamic is based on the storage unit (BESS) and the balance equation of power of the charging station. In this model, the power fed into the grid is positive, and the one extracted from the grid is negative. In the case of the BESS, the power is positive when it is discharging and negative otherwise. The BESS's SoC depends on its previous state and the ratio between the current battery capacity and its maximum value C_{max}. The battery capacity varies with time and depends on its efficiency (n_{bat}), the current $P_{bat}(k)$ and the sampling time (T_s):

$$SoC(k+1) = SoC(k) - \frac{n_{bat} T_s}{C_{max}} P_{bat}(k). \tag{4}$$

In this model, the efficiency for charging and discharging is assumed to be equal for both cases and positive.

Then, the main equation that governs the charging station is the power balance which is the net sum between $P_{bat}(k)$, $P_{grid}(k)$, $P_{pv}(k)$, $P_{EV}(k)$ at the point of common coupling. The sum should be equal to zero at every time instant, thus:

$$P_{bat}(k) + P_{grid}(k) + P_{pv}(k) - P_{EV}(k) = 0. \tag{5}$$

3.1. Problem formulation

The main goal of the optimization problem is to reduce the CO_2 emissions and operational costs. This goal is gained by minimizing the grid power feeding the EV for a specific prediction horizon (N_p), thus maximizing the exploitation of the PV/BESS system. Moreover, the BESS should charge only when PV power is available. Thus, the objective function includes three terms: the power from the grid, the BESS, and finally, the power from the PV. The three weights (α_1, α_2, α_3) assign the priority to each power source. The objective function is expressed as follows:

$$\min J(k) = \sum_{k=1}^{N_p-1} \alpha_1 P_{grid}^2(t+k) + \alpha_2(P_{bat}^2(t+k)) + \alpha_3(P_{pv}^2(t+k)). \tag{6}$$

According to the strategy explained above, the highest priority is given to the group of the PV/BESS. Meanwhile, the lowest priority is given to the grid, as the main objective is to minimize the energy consumption from the grid. These values remain fixed during this study. The weight factors, α_1, α_2, and α_3 have been selected by using the guidelines presented in [18], and by testing different weight factors to check the system's performance. The testing of these weight factors was performed with the main goal of the grid energy minimization. On this, it is important to mention that the BESS should not be charged during night hours by the grid. During the day, the BESS can be charged by the available PV production. However, if the EV connects during the day and the BESS is not fully charged, the BESS and the EV could have power coming from the PV production. Thus, α_2 and α_3 are equal. For this application, α_1 is 0.016, (α_2) and (α_3) are set to 0.014. These values allow a good trade-off between minimizing the power provided by the grid and the MPC controller stability.

3.2. Constraints

The constraints are formulated by considering the safety limitations of the system. The *SoC* should be inside some safety limits in order to prevent the battery degradation. The maximum and minimum power limitations for the battery and the PV system are considered too:

$$\begin{aligned} SoC^{min} &\leq SoC(k) \leq SoC^{max}, \\ P_{grid}^{min} &\leq P_{grid}(k) \leq P_{grid}^{max}, \\ P_{bat}^{min} &\leq P_{bat}(k) \leq P_{bat}^{max}. \end{aligned} \tag{7}$$

In the real system considered in this paper for the experimental validation of the eMPC, the PV system, and the BESS are both managed by the same inverter (P_{inv}), so that the following additional constraints are accounted for:

$$\begin{aligned} P_{inv}^{min} &\leq P_{inv}(k) \leq P_{inv}^{max}, \\ P_{inv} &= P_{pv}(k) + P_{bat}(k). \end{aligned} \tag{8}$$

3.3. MPP formulation

Because the online solving of the previous problem formulation can be time-consuming in a real implementation, MPP is formulated in order to solve the problem offline over predefined ranges of parameters treated as uncertain.

In the current case, the cost objective function presented in Eq. (6) is translated to a MPP-Quadratic Programming (MPP-QP), which obeys the following expression:

$$\begin{aligned} J(\theta) = \min_{u \in \mathbb{R}} \quad & (Qu + H\theta + c)^T u \\ \text{s.t.} \quad & G(\theta)u \leq W + E\theta \\ & \theta \in \Theta \subset \mathbb{R}^q \\ & \theta_l^{min} \leq \theta_l \leq \theta_l^{max}, l = 1, \ldots, q. \end{aligned} \tag{9}$$

On this equation, (Θ) is the space of the uncertainty parameters, $Q \in \mathbb{R}^{(nxn)}$, $H \in \mathbb{R}^{(nxq)}$, $c \in \mathbb{R}^n$, $G \in \mathbb{R}^{(mxn)}$, $G \in \mathbb{R}^{(mxn)}$, $W \in \mathbb{R}^m$, and $E \in \mathbb{R}^{(mxq)}$. The objective function (6) is formulated according to the quadratic MPP function presented in (9), thus Q, H, c, G and W are computed from the matrices A, B and α. The mathematical transformation is explained in [19].

The solution to the problem is a piecewise affine function defined by the active sets constructed due to the constraints and the limits given by the uncertainties. Every function defines a polygon, known as a Critical Region (CR). By definition, in the CRs, the objective function relates to each of the uncertainties and their combination. Each CR gives the optimal value of the control variable vector ($u(\theta)$) when the uncertain parameters assume values falling in that CR [13].

3.4. Uncertainties

In MPP formulation, the uncertainties are modeled as bounds defining the uncertain ranges that could correspond to forecasting errors, variance, or minimum and maximum values. For an MPP formulation, the uncertainties should be those parameters that can be measured in real time. In the present application, these parameters are the PV production (θ_{pv}), the EVs consumption (θ_{EV}), the BESS's *SoC* (θ_{SoC}), parameters that come from the charging station. Furthermore, electricity price (θ_p) and CO_2 emissions (θ_{CO_2}) are also parameters that can be known in real time and are given by the Transmission System Operator (TSO). Thus, the vector of the uncertainty parameters is defined as:

$$\theta(k) = \left[\theta_{pv}(k), \theta_{EV}(k), \theta_{SoC}(k), \theta_p(k), \theta_{CO_2}(k)\right]. \tag{10}$$

For the current case, the uncertain parameters assume values within given bounds as follows:

$$\begin{aligned} P_{pv}^{min} &\leq \theta_{pv} \leq P_{pv}^{max}, \\ P_{EV}^{min} &\leq \theta_{EV} \leq P_{EV}^{max}, \\ SoC^{min} &\leq \theta_{SoC} \leq SoC^{max}, \\ Price^{min} &\leq \theta_p \leq Price^{max}, \\ CO_2^{min} &\leq \theta_{CO_2} \leq CO_2^{max}. \end{aligned} \tag{11}$$

In order to obtain a possible solution of the MPP for any given day, the ranges assumed for θ_{pv} and θ_{EV} go from 0 to the maximum PV power and the maximum EV load respectively. In the case of θ_{SoC}, the minimum and maximum values are the same as the limitations chosen in (7). For the limitations of price and CO_2 emissions, it is necessary to analyze the data given by the TSO from the previous years, in order to see which are the minimum and maximum possible value. In this study, the ranges chosen are wide enough in order that the mapping of the critical regions guarantee the solution for any given case, even for quick variations of the uncertainties.

Two further constraints are added concerning to the two last uncertain parameters. These constraints give a threshold regarding the CO_2 emissions and the electricity price. This threshold is limited by Γ_1 and Γ_2 for the electricity price and the CO_2 emissions, respectively. The lower the value of Γ_1 and Γ_2, the lower the chance to recharge the EV with power from the grid

when CO_2 emissions and electrical price are higher in the day. These two constraints are written as:

$$0 \leq \theta_p * P_{grid}(k) \leq \Gamma_1, \tag{12}$$

$$0 \leq \theta_{CO_2} * P_{grid}(k) \leq \Gamma_2. \tag{13}$$

3.5. MPP model solving

In literature, a few algorithms for multiparametric quadratic programming are available. They have three main objectives: (i) the creation of the CRs in the multidimensional space of the uncertain parameters, (ii) the calculation of the control law corresponding to each region, and (iii) the capability to enter into the corresponding CR once the actual value of the parameter that was considered as uncertain during the optimization process becomes known. Regarding the first objective, the partition of the problem in different CRs is based on computational geometry. This algorithm creates piecewise quadratic models into polyhedral regions limited by the given constraints, the inputs, and the space of states. The geometric computation algorithm divides the space for every time step of the corresponding prediction horizon. This approach is explained in detail in [20]. As for the second objective, a Parametric linear complementary programming is applied to every created CR. This repeats for the complete parameter space and it is performed offline. Finally, regarding the last goal, a Point location problem is used in order to find the CRs and the corresponding control law [21]. This third step operates on the real values of the parameters, thus it is effectively developed online. More information about these algorithms can be found in [12]. As for the offline optimization procedure, the algorithm steps are summarized as it follows:

1. Divide the parameters space in a range of finite values (i);
2. Select a parameter (θ);
3. Set the constraints and the initial state of variables;
4. Calculate the corresponding critical region (CR_i) for the specific (θ_i);
5. Solve the MPP quadratic problem to obtain the control law as a function of the parameter ($u(\theta_i)$);
6. Store the control function;
7. Repeat this for every time step into the prediction horizon;
8. Repeat the procedure until all the parameter space is analyzed.

Once the critical regions have been created for the entire parameter space, these are used at each time step during the real time operation, when the actual values of the parameters that were considered as uncertain in the offline optimization procedure are known. Thus, the CRs are evaluated as follows:

1. Measure the real values of θ;
2. Search the corresponding CR_i, such that $\theta \in CR_i$;
3. Calculate the control variables (u(θ)) for the complete N_p.

Fig. 2 shows an example of a bi-dimensional space of two uncertain parameters θ_1 and θ_2 in which two CRs, i.e., CR1 and CR2, have been generated through MPP. If, during the real-time operation of the system, it is $\theta_1 = a$ and $\theta_2 = b$, then the objective function value $J(a, b)$ falls within CR1, thus the system control variables are settled according to this. Instead, if $\theta_1 = a'$ and $\theta_2 = b'$, then the objective function value falls in CR2, so that the control variables are settled differently from the previous case.

For the current application, the quadratic MPP problem is solved by using the toolbox MPT 3.0 available in MATLAB, with an interface to YALMIP. This toolbox is specialized for parametric optimization, computational geometry and model predictive control. The main solver is Parametric Linear Complementary Programming (PLCP). Additionally, this toolbox permits saving the CRs defined by the control laws in C code, which can be used by other software or platforms.

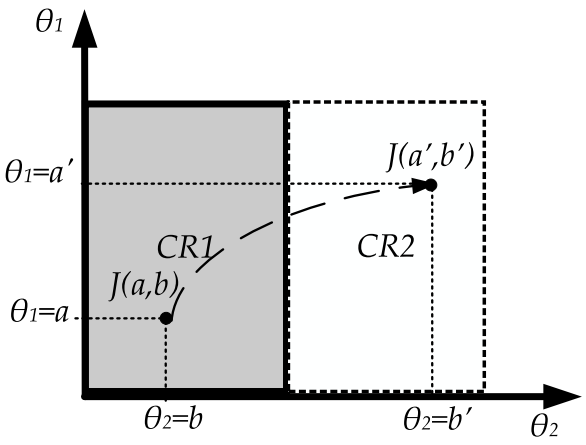


Fig. 2. Simple example of CRs in the space of the uncertain parameters.

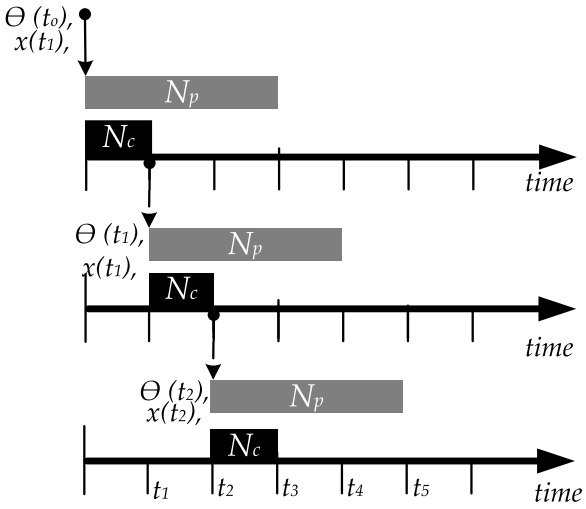


Fig. 3. Example of the eMPC operation (N_p: Prediction Horizon, N_c: Control Horizon).

4. Real time control using eMPC

In this section, the real time control is formulated based on eMPC. The common operation of this technique is to provide an optimal solution for the future, which yields in the current state, constraints, and the given dynamics of the system. In the specific framework of the charging station, this means that, at the current point in time (k), an optimal plan is formulated for a prediction horizon (N_p) based on the prediction of the response of the BESS, the current θ_{SoC}, and the real values of θ_{pv}, θ_{ev}, θ_{CO_2}, and θ_p. Then, the CRs are evaluated to search for the control law, taking into account the actual values of the uncertainties. The control variables (P_{bat} and P_{grid}) are chosen from these CRs for the next time step ($k + 1$) known as control horizon (N_c). The decision is implemented for this N_c. At the next step, the new values of the uncertainties and the current state of the battery are measured, and a new optimal schedule is planned. As a result, the horizon is shifted (see Fig. 3) at every sampling time. This rolling horizon approach helps to compensate for the optimal solution when new information is available, e.g., uncertainties.

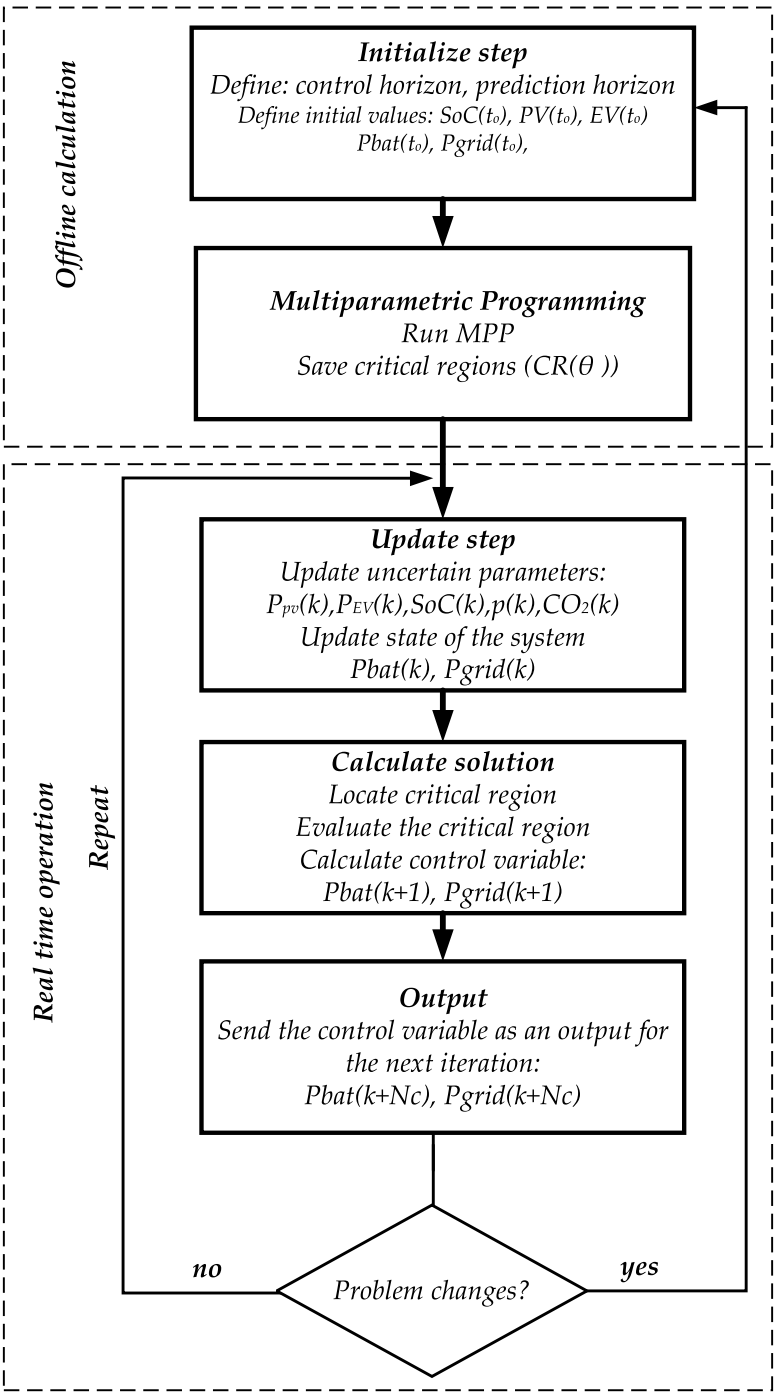


Fig. 4. eMPC framework.

In the current study, the offline mapping of the CRs occurs only once, and it is not repeated at any time of eMPC operation. If the problem formulation changes, e.g., because of the ranges of the uncertain parameters, the constraints, the horizon N_p and N_c change, then the CRs have to be recalculated. The operation of the eMPC is summarized in Fig. 4.

5. Numerical simulation

This section presents the simulation results of the proposed EMS based on eMPC. It first presents a basic case study to explain the operation of the proposed algorithm. Then, it presents the performance under different settings. The data for the numerical simulation is obtained from the PV/BESS based EV charging station at the University of Trieste. The main parameters of the charging station are summarized in Table 1, while those referring to the BESS are collected in Table 2. The data of PV production, EV power consumption, and the SoC of the BESS have been saved with a T_s of five minutes.

5.1. Case study description

This case study aims to show the way the proposed algorithm works. Thus, the set of uncertain parameters has been limited to θ_{PV}, θ_{EV}, and θ_{SoC}, so that the CRs can be visualized in three-dimensional space. The ranges of these uncertain parameters (11) are listed in Table 3. As for θ_{PV} and θ_{EV}, the upper bounds are

Table 1
Charging station's main parameters.

Parameters	Value	Units
Maximum PV power	3900	W
Maximum battery power	3300	W
Maximum inverter power	3300	W
EV charging power	6700	W

Table 2
BESS main parameters.

Parameters	Value	Units
Technology	Ion Lithium iron phosphate battery	
Normal capacity	10	kWh
Number of max cycles	10 000	cycle
Vdc	49	V
Efficiency	98	%
Warranty	10	years
Maximum power	3.3	kW

Table 3
Lower and upper bounds of the uncertainties for the numerical simulation.

Uncertainties	Min	Max	Units
θ_{PV}	0	3900	W
θ_{SoC}	10	90	%
θ_{EV}	0	6700	W
θ_{SoC}	10	90	%

Table 4
Vertices corresponding to CR2.

θ_{PV} (W)	θ_{EV} (W)	θ_{SoC} (%)
3300	0	90
0	0	90
3300	3300	90
3300	0	68.57

set by taking into account the rated power presented in Table 1. The upper and lower bounds of the θ_{SoC} range have been fixed by considering the safety limits of 10% and 90%, respectively.

The sampling time chosen for the model formulated in Section 3 is 5 min. In the objective function presented in Eq. (6), the weight values are: $\alpha_1 = 0.016$, $\alpha_2 = 0.014$, and $\alpha_3 = 0.014$. For the eMPC, the N_p is set to 12 time steps and N_c is equal to one time step. The time step corresponds to the sampling time, i.e., 5 min. For this case study, experimental data acquired on the 22nd of July 2020 have been considered.

5.2. MPP solution

MPP solves the optimization problem and determines four CRs (Fig. 5). These polyhedral regions in the three-dimensional space of the uncertain parameters correspond to 33 variables, 154 inequality constraints, 22 equality constraints. Additionally, 33 lower and upper bounds corresponding to each time step of the N_p have been considered.

As an example, it is worth noting that CR2 is a pyramid where the θ_{PV} varies in the range [0, 3300] W, the θ_{SoC} in the range [80, 90]%, and the θ_{EV} in the range [0, 4000] W. It has in total four vertices, taking the values presented in Table 4. This region is represented as follows:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ -0.0065 & 0.0065 & -0.9999 \end{bmatrix} \times \begin{bmatrix} \theta_{PV_k} \\ \theta_{EV_k} \\ \theta_{SoC_k} \\ u_{(k+1)} \end{bmatrix} \leq \begin{bmatrix} 3300 \\ 0 \\ 90 \\ -89 \end{bmatrix}$$

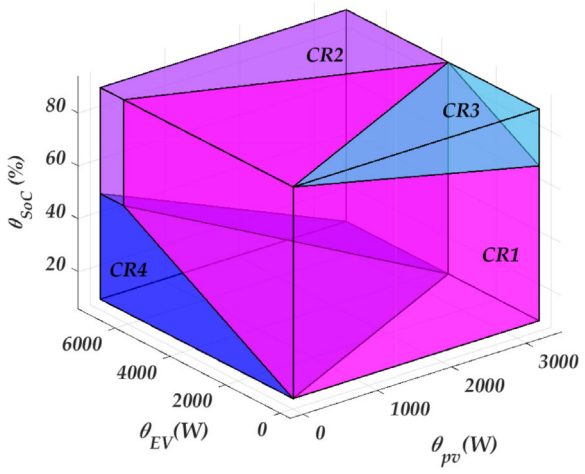


Fig. 5. Critical regions constructed by the MPP when θ_{PV}, θ_{EV} and θ_{SoC} are taken into account.

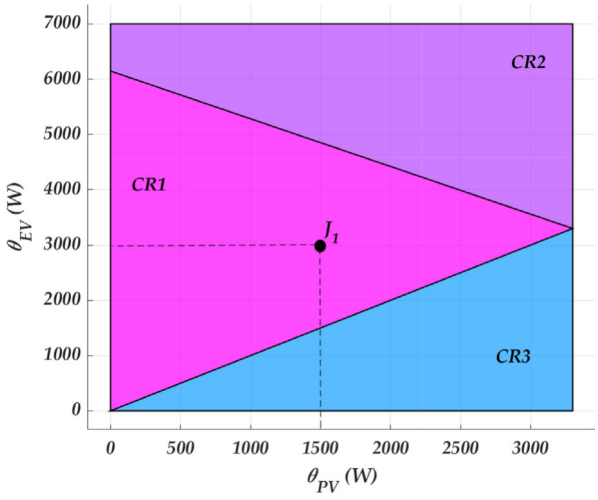


Fig. 6. Critical regions when θ_{SoC} is known and equal to 90%.

Every CR is associated with a specific set of control variables values and does not mesh up with other regions. The total solution is saved in a look-up table using a memory space of 9 kB.

5.3. eMPC performance

The eMPC moves in these CRs by reading the set of actual uncertain parameters values and by evaluating in which CR they fall to get the optimized values of the control variables. For instance, in the case $\theta_{SoC} = 90\%$, the solution falls inside one of the CRs that are visible in Fig. 5 at the top face of the parallelepiped. This plane is shown in Fig. 6, where are three critical regions that depend on two parameters: θ_{pv}, and θ_{EV}.

As an example of the eMPC operation through the CRs shown in Fig. 6, at the first time instant, with $\theta_{SoC} = 90\%$, $\theta_{pv} = 1500$ W and $\theta_{EV} = 3000$ W, the objective function takes the value J_1. The optimized schedule is determined for the N_p, and only the samples for the corresponding N_c are implemented. In this case, N_c is equal to the sampling time. As a result, the EV is fed from the grid for the next sampling time by 694 W, by 805 W from the BESS, and the PV array provides the remaining part.

In another instant, the current values of the uncertain parameters are again measured, and they are: $\theta_{SoC} = 11\%$, $\theta_{pv} = 1500$ W,

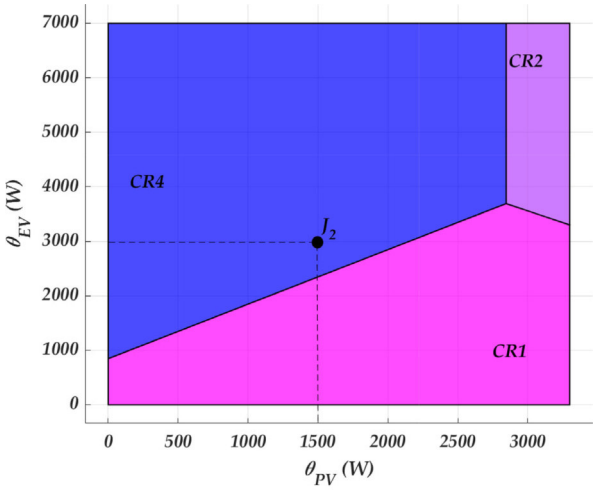


Fig. 7. Critical regions when θ_{SoC} is known and equal to 11%.

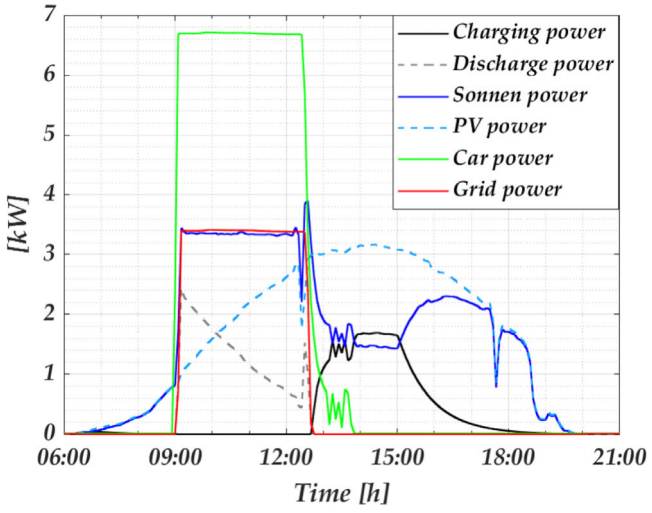


Fig. 8. Power profiles for the simulation test.

and $\theta_{EV} = 3000$ W, so J_2 is obtained. It falls in CR4, and it is in the plane illustrated in Fig. 7. For this value, the EV is fed from the grid by 1045 W, by 455 W from the battery and the remaining part is provided by the PV array.

The evaluation of the CRs with the current values of the parameters that were considered as uncertain during the MPP formulation is repeated every sampling time. The power profiles, computed as described above over the whole day, are shown in Fig. 8. In the day considered in this example, the EV connects for recharging early in the morning, at 9 am, when the PV production is still low and the BESS *SoC* is close to 90%. The algorithm calculates the optimized control values of P_{grid} and P_{bat}. As long as the PV array delivers more power, the battery reduces its contribution. The EV battery recharging terminates after three hours from its connection. Afterwards, the PV system recharges the BESS until the *SoC* approaches 90%. The maximum PV power is close to 3.1 kW and the BESS recharges in 5 h.

5.4. Effect of changing prediction horizon and number of parameters

The prediction horizon and the number of uncertain parameters affect the eMPC performance. This subsection examines their effect on the proposed study from the operational point of view.

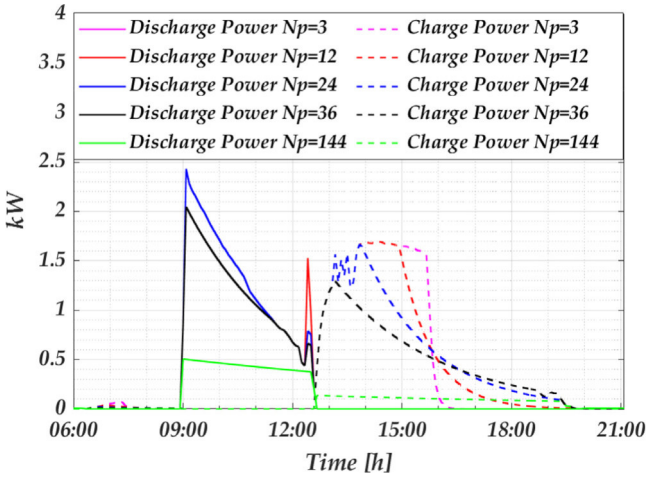


Fig. 9. Battery power profiles for different prediction horizon (N_p).

First, a number of different N_p values has been tested with the same number of uncertain parameters presented in Section 5.1. All the other parameters and the data set remains unchanged. N_p takes the following values: 3, 12, 24, 36, and 144, which correspond to 15 min, 1 h, 2 h, 3 h, and 12 h, respectively. Fig. 9 shows the charging and discharging power of the battery for each of the five N_p values considered. The EV is connected at 9 am; thus, the battery proceeds supplies power to the charging station. There is no difference among the performance for $N_p = 3$, $N_p = 12$, or $N_p = 24$. However, this power reduces slightly for the case of $N_p = 36$.

Instead, for $N_p = 144$, the battery supplies only $P_{bat} = 0.5$ kW. Additionally, at the end of the EV's connection, there is a sudden reduction of the PV production (Fig. 8), thus, the battery needs to compensate for the loss of power. In the case of a $N_p \leq 12$, the compensation for the quick variation is 1.2 kW. The compensation is less than 0.8 kW For $N_p = 24$ and $N_p = 36$. However, this compensation does not occur when $N_p = 144$. After the EV disconnection, the higher the N_p the smoother the charging power of the battery. For $N_p = 144$, the charging power is reduced as the battery has not been fully depleted from the first interaction with the EV. In any case, -charging or discharging-, the slow control is because the objective function is evaluated for a longer time. Although the battery shows a slow dynamic, its charging profile varies with the N_p value. A further analysis has been conducted by varying the number of uncertain parameters (θ) and thus the number of constraints. The test is developed for (a) $\theta = 3$ and (b) $\theta = 5$. The prediction horizon is the same for both studies and equal to $N_p = 12$. For $\theta = 3$, the main parameters and the study case are described in Section 5.1. For $\theta = 5$, adds to the previous study θ_{CO_2} and θ_p. Their bounds and the additional parameters are in Tables 5 and 6. The data set of PV production and EV consumption are the same as Section 5.1. The power profile of the both cases remains the same, and it does not present any variation under the performed parameters. This is because θ_{CO_2} and θ_p affects directly on the amount of power bought from the grid but not the one related to the battery. The variation of these parameters and its effect on the system are tested on the real time operation.

5.5. Evaluation of computational performance

The evaluation of the computational operation is performed by varying the number of uncertain parameters and the prediction horizon. As first, it is tested for $\theta = 3$ and a N_p that varies

Table 5
Lower and upper bounds of the uncertainties.

Uncertainties	Min	Max	Units
θ_{PV}	0	3900	W
θ_{SoC}	10	90	%
θ_{EV}	0	6700	W
θ_p	0.03	0.1	€
θ_{CO_2}	0.3	1	kg

Table 6
Main parameter's values.

Parameter	Value	Unit
Γ_1	2	€/W
Γ_2	2	kg/W
Np	12	
Nc	1	

from 3 to 144. Afterwards, the analysis is performed for $\theta = 5$ with the same variation of N_p as the case before. The number of regions created, variables, inequality and equality constraints are summarized in Table 7. Additionally, the computation time and the size of the file containing the data describing the CRs for each case are also shown. The increase of the number of parameters determines an increase of the number of regions. In this case, the number of CRs does not change with N_p. Instead, the computation time increases as the number of θ and N_p increase. On the same processor mentioned above, it can go from 0.15 s to 2400 s by only varying θ and N_p. However, for $\theta = 5$ and $N_p = 144$, the solution is infeasible after 12 h of simulation. As the number of parameters increases, the file's size where these values are stored also increases, but it keeps in the order of kB. These files can be used in any hardware or platform to do the real-time control. The larger file is obtained when the $N_p = 144$ with a $\theta = 3$, but it is only 84 kB. It can be easily saved and used by any low cost platform. To access the file in online optimization only takes 0.019 s. Thus, it did not change with any variation of N_p or the number of θ.

In conclusion, the eMPC parameters affect performance and the offline optimization computation time. Although the offline computation time increases with the number of parameters, the online use of the CRs resulting from the offline optimization of the eMPC always needs the same computation time. This is because the online procedure only searches for the specific control laws by inspecting the CRs according to the system's current status. This one represents the significant advantage of eMPC with respect to a simple MPC.

6. Experimental validation

Two testbeds were used to evaluate the proposed EMS strategy's control and computing performance. The first testbed (Testbed 1) is completely based on a Personal Computer (PC), while the second includes a dSPACE Scalexio (Testbed 2) (Fig. 10).

There are two purposes of this experimental validation. The first one is to overview the EMS's performance under the effect of five uncertain parameters. The uncertainties considered are θ_{pv}, θ_{EV}, θ_{SoC}, θ_{CO_2} and θ_p. The upper and lower bounds of these uncertain parameters (Eq. (11)) are listed in Table 5. The bounds for θ_p and θ_{CO_2} are fixed by considering the historical data of the electrical system in Italy, referring to the years 2020 and 2021 [22]. The second one is to overview the computing performance under different hardware.

In both testbeds, the CO_2 emissions are calculated by using the type of power generation given by ENTSO-E along the day. For comparison, the variation of the price and the CO_2 emissions

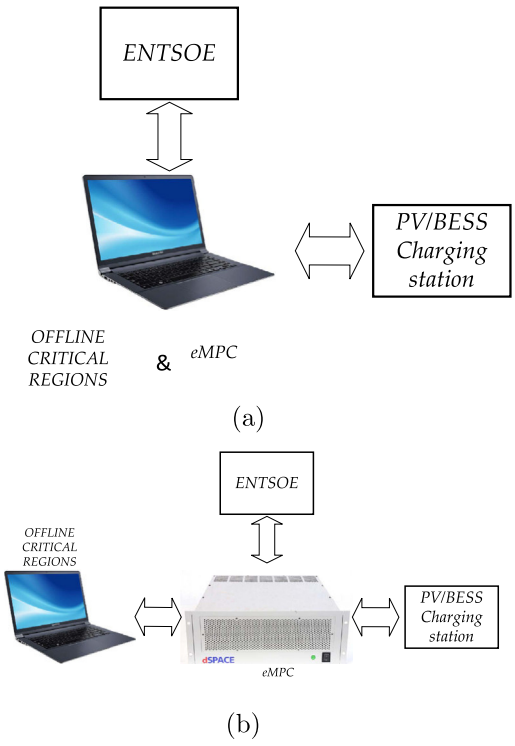


Fig. 10. Testbed layout (a) Testbed 1, (b) Testbed 2.

are assumed to be equal to the ones presented in Fig. 11 for all the tests.

Table 6 collects the used values of the factors and the constants involved in the optimization model. The results achieved by using the two testbeds are described in the following subsections.

6.1. Testbed 1

The PC used is an Intel i5-8500 processor with a frequency of 3 GHz and a memory RAM of 16 GB. The eMPC runs in Matlab/Simulink. It interacts with the charging station through an Ethernet protocol to set the control signals and to receive the actual P_{pv}, P_{EV}, and SoC values. It also receives from ENTSO-E the prices and type of power generation. In this case, the communication between the PC and the charging station occurs every 5 min, the sampling time. The control horizon is equal to the sampling time ($N_c = 5$ min) and the prediction horizon is fixed at three times the N_c, thus at $N_p = 15$ min.

The MPP is run offline to map the CRs. Then, the eMPC operates at every N_c by reading its inputs, accessing the CRs to fix the optimal values of the control variables, and finally sending these values of the control signals to the charging station. For each N_c, the eMPC takes 0.019 s to access the CRs and 2.5 s to read its actual input values and settle the optimal values of the control variables of the charging station. The MPP needs 0.34 s of computing time to determine the 4 CRs.

The power profiles referring to the 1st of July 2021 are shown in Fig. 12. The EV connects for recharging at 9 am. During the four hours of charging, the PV array and the BESS feed the charging station with their maximum power, i.e., 3.3 kW. Afterwards, when the EV is fully charged, the PV array recharges the BESS (40 to 90%). The rest of the power comes from the electrical grid.

In this result, it is important to notice that the PV/BESS has a slow response to the variations of PV production and load due to the chosen value of the sampling time and N_p. Phenomena

Table 7
Computational performance.

θ	Np	CR	Variables	Inequality constraints	Equality constraints	Time [s]	File size [kB]
3	3	4	6	28	4	0.15	2.76
3	12	4	33	154	22	0.605	9
3	36	4	105	490	70	5.15	25
3	144	3	429	2002	286	506.7	84
5	3	4	6	44	6	0.188	4
5	12	6	33	242	22	2.213	10
5	36	6	105	770	70	2400	33
5	144	Infeasible				43 200	–

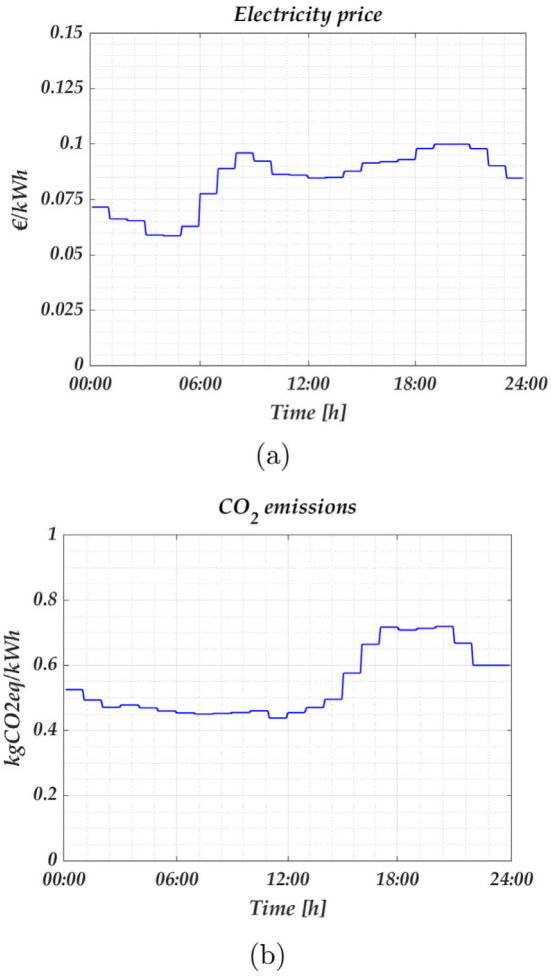


Fig. 11. One day data of (a) Electricity price, (b) CO_2 emissions.

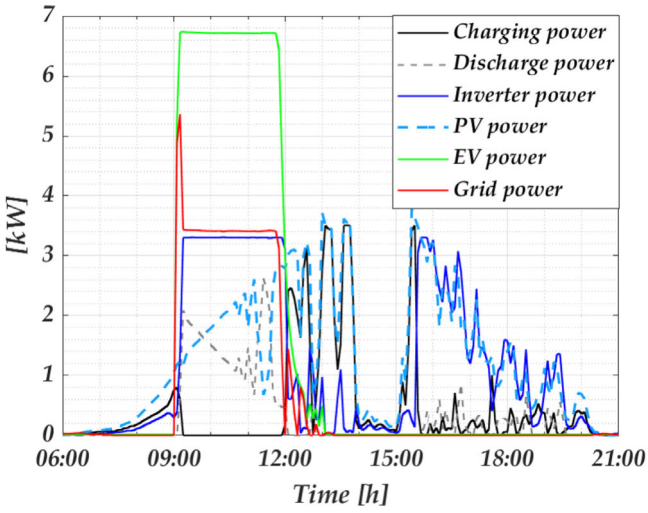


Fig. 12. Power profile for Testbed 1 (Day 1).

occurring on a shorter timescale, thus within two consecutive N_ps, receive a delayed counteraction from the EMS.

6.2. Testbed 2

The second testbed uses dSPACE for implementing reactive scheduling. The processor board is a DS 6001 equipped with an Intel i7-6820EQ processor with a frequency of 2.8 GHz. This processor has four cores and a memory of 4 GB RAM plus 8 GB of flash memory. The interface communication chosen for the current application is ETHERNET with a low latency interface.

In testbed 2, the dSPACE communicates with the charging station through the Ethernet protocol to receive the values of P_{pv}, P_{EV} and SoC at each sampling time and to set the control signals (P_{bat} and P_{grid}). Additionally, it communicates with ENTSO-E to get the prices and the type of power generation every hour. As for Testbed 1, the MPP is run offline to get the CRs which are saved in the dSPACE as a lookup table. The eMPC runs in the dSPACE platform and evaluates the CRs at every sampling time with the actual data. For testbed 2, the sampling time chosen is one second; thus $N_c = 1$ s, and the prediction horizon is $N_p = 3$ s.

Two days have been chosen to evaluate this configuration's performance. On day 1 (7 July 2021), the EV is connected for recharging in the morning. The PV/BESS and the grid supply power to the EV. During the day, when the PV production rises, the BESS reduces its contribution proportionally. As a result, the BESS's SoC drops from 90% to 42%, and it is recharged in the afternoon. On day 2, 8 July 2021, the EV was connected at 4 pm, during a day characterized by a large PV production variability. Unlike Testbed 1, thanks to the shorter N_c, the PV/BESS response is not delayed; thus the fast fluctuations of the PV production and the EV usage are accounted for properly. The SoC of the BESS decreases from 90% to 40%. The BESS is not recharging from the grid at night and remains at 40% until the next day.

6.3. Computing performance on both testbeds

Table 8 summarizes the computing times required by the two testbeds. In both cases, the mapping of the CRs is performed offline, on the PC, and only once. The eMPC in any testbeds takes less than one second as execution time. Thus, regardless of the testbed considered, this time remains the same. The most significant difference between the two platforms is the time needed to communicate with the charging station.

6.4. Analysis of cost and CO_2 emissions

To analyze the cost and the CO_2 emissions, different values for Γ_1 and Γ_2 are explored and evaluated using the data from day 1

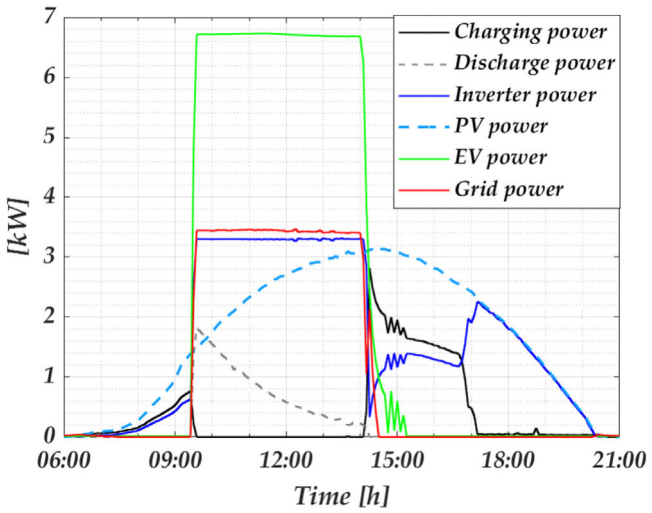


Fig. 13. Power profile for Testbed 2 (Day 1).

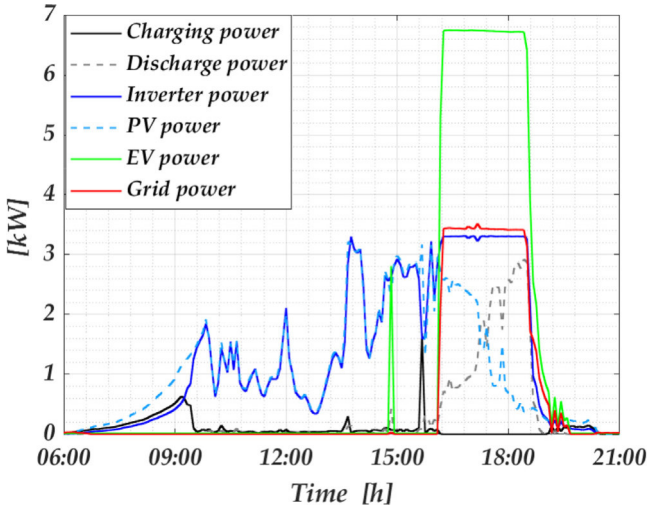


Fig. 14. Power profile for Testbed 2 (Day 2).

Table 8
Comparison between Testbed 1 and Testbed 2.

	Testbed 1	Testbed 2
MPP	0.34 s	0.34 s
eMPC	0.019 s	0.001 s
Online communication	2.5 s	1 s
Sampling time	300 s	1 s

and day 2. Three specific values are chosen: 1, 1.2, and 1.4 and it is assumed that $\Gamma_1 = \Gamma_2$. The results are summarized in Table 9.

The variation of Γ_1 and Γ_2 affects the charging station's response. If gamma is less than 1, the power from the grid will be limited, and only PV/BESS, thus not the grid, are allowed to supply power to the EV (3300 W). This lengthens the EV's charging time or the charging process terminates before the full charge is reached. If gamma is settled at 1.2, the grid delivers a maximum power of 1557 W from 9 am to 11 am on day 1, and 3300 W in the next 2 h as the electricity price is lower. On day 2, however, the charging station provides 5283 W, whereas the grid only provides 1983 W. This is owing to the fact that prices and CO_2 emissions are higher in the afternoon, limiting the amount of power supplied from the grid. In the case, Γ is greater than 1.4, the EV can be charged to its maximum capacity of 6900 W and be fully charged in roughly 4 h for any of the two days tested.

Table 9
Variation of CO_2 emissions and price by considering different values of Γ.

$\Gamma_1 = \Gamma_2$	Parameters	Day 1	Day 2
1	$Pgrid_{max}$ [W]	554	245
	Maximum power PV/BESS [W]	3300	3300
	CO_2 [kg]	0.6474	1.199
	Price [€]	0.1001	0.1707
	Number of critical regions	5	5
1.2	$Pgrid_{max}$ [W]	1557–3300 W	1983
	Maximum power PV/BESS [W]	3300	3300
	CO_2 [kg]	61.71	44.2121
	Price [€]	11.49	5.9743
	Number of critical regions	6	6
1.4	$Pgrid_{max}$ [W]	3700.00	3700.00
	Maximum power PV/BESS [W]	3300	3300
	CO_2 [kg]	91.06	75.8168
	Price [€]	17.12	10.23
	Number of critical regions	4	4

6.5. Comparison with traditional method

To show the advantages of the proposed eMPC, a comparison with a conventional MPC is proposed. The problem formulation is the same presented in Section 3 for the proposed eMPC. The constraints and limits of the parameters are also the same. The uncertainties are defined as part of the constraints with the same bounds. The MPC, in this case, reads at every sampling time the actual values of the PV power production, of the BESS *SoC*, of the EV power absorption, of the power price, and of the CO_2 emissions. Then, it runs the optimization algorithm and calculates the values of P_{bat} and P_{grid} to settle at every sampling time. Next, it sends the values of the control variables as outputs to the charging station. The values P_{bat} and P_{grid} calculated by the deterministic optimization are the same as the ones from the eMPC approach. The power profiles equal to the ones presented in Figs. 13 and 14. Instead, the computation time required by the optimization at every time step varies from 0.06 to 0.1 s, while the eMPC approach only requires 0.019 s to access the CRs at each sampling time (Table 8).

7. Discussion

In this paper, the application of the eMPC to an e-vehicle charging station based on PV/BESS by accounting for five main uncertainties is shown. The uncertainties affect: θ_{pv}, θ_{SoC}, θ_{EV}, θ_{CO_2}, θ_p. The main advantages by using this approach can be summarized as follows:

- The general problem formulation assembles to a common MPC, making it easier to code and to implement for any application. The MPT toolbox by MATLAB helps to transition from MPC to a MPP to create the CRs in offline mode.
- The uncertainties are modeled by fixing a range of real values for each of them. These bounds can be min/max operation values or forecast errors. In this application, it was chosen to use bounds characterized by min/max operation values. This comes in handy, especially for uncertainties that are difficult to forecast and for which no historical data is available. For instance, in the current application, the main uncertainty is the one related to the EV's connection, disconnection and power consumption. This uncertainty is modeled by a range of values between 0 and 7 kW. Also, the uncertainty of PV production, the price and CO_2 emissions are modeled in the same way. In the case of PV power, the boundaries allow considering sudden changes of solar irradiance during the day. For instance, in Trieste, the 2nd of July 2021, the PV production reduced to 0 in less than 1 min

in the afternoon due to a cloud passing. Because of this behavior, the limits for the PV production are set between 0 and 3.3 kW. The modeling of price and CO_2 emissions by using means of a couple of bounds allows considering a wide range of values.

- The main advantage of using this technique is the computation time in online mode. eMPC is based on two main aspects: creating the CRs in offline mode and the rolling control horizon in online mode. The second one, only evaluates the CRs saved beforehand. For the evaluation, it only takes 0.019 s to extract the control values from the CRs. Using other approaches as stochastic-MPC, reported in [10], that it takes 1.04 s, for a $N_p = 12$, $N_c = 3$. Plus, it only takes into account one uncertain parameter. In our case, the number of uncertainties is 5 and the evaluation is 55 times less than the cited example. Because the EV can come at any time, the optimization online has to be fast. Thus, calculating the control laws beforehand offers a great advantage for this type of application.
- The control laws can be exported in C, and python language, making them independent of MATLAB or the MPT toolbox for the real time control. This helps the porting of the real time control to any platform. Its use in FPGAs [23] and chips [14], has been reported, making eMPC suitable for low-cost platforms.

The main drawbacks of the proposed eMPC technique are summarized as follows:

- The offline computation time and memory usage increase with the problem's dimensions. The number of CRs increases with the number of parameters, constraints, and number of components. Thus, the offline computation time increases. Therefore, the proposed approach fits with low dimensional systems.
- The use of license software to construct the critical regions as MPT Toolbox and that runs under Matlab framework. The mathematical approach behind the construction of the CRs can be complex. Thus, the use of a toolbox or a programming tool is necessary. This can be a drawback in the case it is required to reconstruct the CRs.

After having analyzed the advantages and disadvantages of this approach, it can be stated that eMPC is a good solution for EMS in the case of nanogrids under the effect of various uncertainties. The current paper shows the eMPC applicability in the case of an e-vehicle charging station based on a PV/BESS. This solution can be reproduced for similar applications at the distribution level. It also offers a solution for distributed energy control, as it can provide optimized real-time control for a prosumer. This approach can be especially applicable for virtual power plants where two-stage optimization could be necessary, where an eMPC can develop the low-level optimization at the side of the prosumer.

8. Conclusions

This paper presents the use of eMPC in EV charging stations based on PV and BESS, taking into account uncertainties. The explicit formulation is based on the mapping of CRs by the use of MPP. The uncertainties considered are: PV production, SoC, electricity price, CO_2 emissions, and the EV consumption. The algorithm proposed was tested in two different testbeds, one using a commercial PC and the other using a dSPACE real-time platform.

The results showed that the eMPC is a good candidate for the real-time energy management as it can run offline while the results can be saved in a lookup table. This reduces the online optimization time when the actual values of the uncertainties are known. The eMPC should only evaluate the critical regions saved at the lookup-table and set the new control variables at every sampling time. The evaluation of the critical regions takes less than 1 s for any of the two platforms tested. Thus, eMPC could be used in a low cost platform to run real time energy management. However, the number of CRs for the current application can grow exponentially when the prediction horizon increases, affecting the computation time offline. Besides the electricity price, the proposed algorithm also takes into account the CO_2 emissions due to the mix energy from the grid as part of the uncertainties and constraints. Severe limitations in this account would increase the connection time of the EV to be fully charged.

Further research is necessary taking into account the forecast of PV production and the EV connection, which will lead to the variation of N_c and N_p along the day.

CRediT authorship contribution statement

Ana Cabrera-Tobar: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Alessandro Massi Pavan:** Conceptualization, Methodology, Funding acquisition, Project administration, Supervision, Writing – original draft, Writing – review & editing. **Nicola Blasuttigh:** Conceptualization, Software, Investigation, Writing – review & editing. **Giovanni Petrone:** Conceptualization, Methodology, Supervision, Writing – original draft, Writing – review, Project administration, Supervision & editing. **Giovanni Spagnuolo:** Conceptualization, Methodology, Supervision, Writing – original draft, Writing – review & editing, Supervision, Project administration.

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.

Acknowledgments

This work was supported 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, and 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 program Interreg Italy-Croatia. This work has been also supported by MIUR, Italy funds in the frame of PRIN 2017- Holistic approach to EneRgy-efficient smart nanOGRIDS (HEROGRIDS) project (2017WA5ZT3_003), and by FARB funds of the University of Salerno.

References

[1] Global EV Outlook 2020, Tech. rep., IEA, Paris, 2020, URL https://www.iea.org/reports/.
[2] M.M. Vazifeh, H. Zhang, P. Santi, C. Ratti, Optimizing the deployment of electric vehicle charging stations using pervasive mobility data, Transp. Res. A 121 (2019) 75–91, http://dx.doi.org/10.1016/j.tra.2019.01.002.
[3] S. Faddel, A.T. Al-Awami, M.A. Abido, Fuzzy optimization for the operation of electric vehicle parking lots, Electr. Power Syst. Res. 145 (2017) 166–174, http://dx.doi.org/10.1016/j.epsr.2017.01.008.
[4] S. Nojavan, M. Majidi, K. Zare, Performance improvement of a battery/PV/fuel cell/grid hybrid energy system considering load uncertainty modeling using IGDT, Energy Convers. Manage. 147 (2017) 29–39, http://dx.doi.org/10.1016/j.enconman.2017.05.039.

[5] A. Vaccaro, M. Petrelli, A. Berizzi, Robust optimization and affine arithmetic for microgrid scheduling under uncertainty, in: Proceedings - 2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe, EEEIC/I and CPS Europe 2019, Institute of Electrical and Electronics Engineers Inc., 2019, http://dx.doi.org/10.1109/EEEIC.2019.8783572.
[6] B. Carlos, G.-T. Félix, A.R. Miguel, Advances in Industrial Control Model Predictive Control of Microgrids, 2019, pp. 1–266.
[7] J. Hu, Y. Shan, J.M. Guerrero, A. Ioinovici, K.W. Chan, J. Rodriguez, Model predictive control of microgrids – An overview, Renew. Sustain. Energy Rev. 136 (2021) 110422, http://dx.doi.org/10.1016/J.RSER.2020.110422.
[8] A. Parisio, E. Rikos, L. Glielmo, A model predictive control approach to microgrid operation optimization, IEEE Trans. Control Syst. Technol. 22 (5) (2014) 1813–1827, http://dx.doi.org/10.1109/TCST.2013.2295737.
[9] J. Choi, J.I. Lee, I.W. Lee, S.W. Cha, Robust PV-BESS scheduling for a grid with incentive for forecast accuracy, IEEE Trans. Sustain. Energy 13 (1) (2022) 567–578, http://dx.doi.org/10.1109/TSTE.2021.3120451.
[10] A. Ravichandran, S. Sirouspour, P. Malysz, A. Emadi, A chance-constraints-based control strategy for microgrids with energy storage and integrated electric vehicles, IEEE Trans. Smart Grid 9 (1) (2018) 346–359, http://dx.doi.org/10.1109/TSG.2016.2552173.
[11] A. Alessio, A. Bemporad, A survey on explicit model predictive control, in: Lecture Notes in Control and Information Sciences, Springer, Berlin, Heidelberg, http://dx.doi.org/10.1007/978-3-642-01094-1_29, Ch. Nonlinear.
[12] I. Pappas, D. Kenefake, B. Burnak, S. Avraamidou, H.S. Ganesh, J. Katz, N.A. Diangelakis, E.N. Pistikopoulos, Multiparametric programming in process systems engineering: Recent developments and path forward, Front. Chem. Eng. (2021) 32, http://dx.doi.org/10.3389/FCENG.2020.620168.
[13] R. Oberdieck, N.A. Diangelakis, I. Nascu, M.M. Papathanasiou, M. Sun, S. Avraamidou, E.N. Pistikopoulos, On multi-parametric programming and its applications in process systems engineering, Chem. Eng. Res. Des. 116 (2016) 61–82, http://dx.doi.org/10.1016/j.cherd.2016.09.034.
[14] S. Wang, X. Wang, W. Wu, Cloud computing and local chip-based dynamic economic dispatch for microgrids, IEEE Trans. Smart Grid PP (c) (2020) 1, http://dx.doi.org/10.1109/tsg.2020.2983556.
[15] G.M. Kopanos, E.N. Pistikopoulos, Reactive scheduling by a multiparametric programming rolling horizon framework: A case of a network of combined heat and power units, Ind. Eng. Chem. Res. 53 (11) (2014) 4366–4386, http://dx.doi.org/10.1021/ie402393s.
[16] N. Tian, H. Fang, Y. Wang, Real-time optimal charging for lithium-ion batteries via explicit model predictive control, in: IEEE International Symposium on Industrial Electronics, Vol. 2019-June, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 2001–2006, http://dx.doi.org/10.1109/ISIE.2019.8781259.
[17] A.M. Pavan, V. Lughi, M. Scorrano, Total cost of ownership of electric vehicles using energy from a renewable-based microgrid, in: 2019 IEEE Milan PowerTech, PowerTech 2019, IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/PTC.2019.8810736.
[18] Model Predictive Control Toolbox Documentation - MathWorks Switzerland. URL https://ch.mathworks.com/help/mpc/.
[19] A. Bemporad, M. Morari, V. Dua, E.N. Pistikopoulos, Explicit solution of model predictive control via multiparametric quadratic programming, in: Proceedings of the American Control Conference, Vol. 2, IEEE, 2000, pp. 872–876, http://dx.doi.org/10.1109/acc.2000.876624.
[20] A. Alessio, A. Bemporad, A Survey on Explicit Model Predictive Control.
[21] M. Herceg, M. Kvasnica, C.N. Jones, M. Morari, Multi-parametric toolbox 3.0, in: 2013 European Control Conference, ECC 2013, 2013, pp. 502–510, http://dx.doi.org/10.23919/ecc.2013.6669862.
[22] ENTSO-E. URL https://www.entsoe.eu/.
[23] A. Gersnoviez, M. Brox, I. Baturone, High-speed and low-cost implementation of explicit model predictive controllers, IEEE Trans. Control Syst. Technol. 27 (2) (2019) 647–662, http://dx.doi.org/10.1109/TCST.2017.2775187.

sustainability

MDPI

Article

A Machine Learning and Internet of Things-Based Online Fault Diagnosis Method for Photovoltaic Arrays

Adel Mellit [1,*], Omar Herrak [1], Catalina Rus Casas [2] and Alessandro Massi Pavan [3]

[1] Renewable Energy Laboratory, Jijel University, Jijel 18000, Algeria; herrakomar@gmail.com
[2] Departamento de Ingeniería Electrónica y Automática, Universidad de Jaén, 23071 Jaén, Spain; crus@ujaen.es
[3] Department of Engineering and Architecture, Center for Energy, Environment and Transport Giacomo Ciamician, University of Trieste, 34127 Trieste, Italy; apavan@units.it
* Correspondence: adelmellit013@gmail.com

Abstract: In this paper, a novel fault detection and classification method for photovoltaic (PV) arrays is introduced. The method has been developed using a dataset of voltage and current measurements (I–V curves) which were collected from a small-scale PV system at the RELab, the University of Jijel (Algeria). Two different machine learning-based algorithms have been used in order to detect and classify the faults. An Internet of Things-based application has been used in order to send data to the cloud, while the machine learning codes have been run on a Raspberry Pi 4. A webpage which shows the results and informs the user about the state of the PV array has also been developed. The results show the ability and the feasibility of the developed method, which detects and classifies a number of faults and anomalies (e.g., the accumulation of dust on the PV module surface, permanent shading, the disconnection of a PV module, and the presence of a short-circuited bypass diode in a PV module) with a pretty good accuracy (98% for detection and 96% classification).

Keywords: photovoltaic array; machine learning; Internet of Things; fault detection; fault classification

Citation: Mellit, A.; Herrak, O.; Rus Casas, C.; Massi Pavan, A. A Machine Learning and Internet of Things-Based Online Fault Diagnosis Method for Photovoltaic Arrays. *Sustainability* **2021**, *13*, 13203. https://doi.org/10.3390/su132313203

Academic Editor: Manosh C. Paul

Received: 27 October 2021
Accepted: 25 November 2021
Published: 29 November 2021

1. Introduction

The International Energy Agency (IEA) reports that at the end of 2020, global photovoltaic (PV) capacity installations reached 760 GWp [1]. PV monitoring systems are indispensable for the reliable operation and maintenance activities of an impressive number of photovoltaic systems. Recently, with the advances in telecommunication technologies, the Internet of Things (IoT) technology represents a key solution for the design of remote PV monitoring systems [2]. However, a fault diagnosis technique should be embedded in the system in order to prevent and isolate any possible fault, which may compromise the normal operation of a PV installation [3].

With reference to the fault diagnosis methods, a good number of machine learning (ML) methods have been developed and presented in the literature [3,4]. These have demonstrated a good ability in the detection and the classification of both common faults, (e.g., open circuit, short circuit, partial shading, soiling, and degradation) and complex faults (e.g., multiple faults). In general, ML-based models are trained and tested by using measured or simulated data (which can be obtained by using MATLAB/Simulink, (Ver. 2018a, MathWorks: Natick, MA, USA). Only a limited number of works have been verified experimentally using a real-time implementation of the developed algorithms. A fault diagnosis system usually includes of a number of tasks such as the detection, the identification, the classification, and the localization of the faults.

In order to detect a fault occurring in a PV system, the simplest approach consists of a comparison between the measured and the predicted output powers. There are two different approaches which are usually used in order to estimate the PV power:

1. The first uses accurate mathematical models which can estimate the produced power as a function of some parameters, such as the solar irradiance (G), the air temperature

(T), and the cell temperature (Tc). These models can be implicit or explicit and are based on the parameters which can be found in the datasheet of the PV modules;

2. The second is based on a data-driven approach. In this case, a dataset of a number of parameters is used in order to estimate the produced power.

With reference to the identification and classification of the faults, these can be performed by solving a binary classification and multiclass problem using ML techniques [5–7] or conventional methods. These latter are known as conventional methods and are usually used in order to detect simple faults which are not associated with any other fault. Thus, the PV fault classification and identification consists in the solution of a multiclass problem.

In the literature, there are only a few recent works regarding the fault localization in PV systems [8–10], and this topic still represents an important challenge in the field, especially for large-scale PV plants [11].

A low-cost PV monitoring system based on the Internet of Things (IoT) technique has been developed in Ref. [12]. Here, the monitored data were the currents and the voltages of the PV array, as well as the environmental data (T and G). The monitoring system was developed using an Arduino Mega 2560 microcontroller (2005, the Interaction Design Institute Ivrea, Milano, Italy). A simple fault detection and identification procedure has been described in Ref. [13] for the detection and identification of four types of faults occurring in a PV string (permanent shading, soiling/deposit of dust, short-circuited PV modules, and disconnected PV modules). However, the microcontroller used in this study was not suitable for the fault classification based on the ML methods. This was due to the limited resources of the device. In this work, we aim at showing that a Raspberry Pi 4 microprocessor (2012, Raspberry Pi Foundation, Cambridge, UK) can overcome this type of problem.

In the literature, there are already a certain number of papers where the Raspberry Pi has been used for the development of an IoT-based technique for the monitoring of PV systems [14–17]. However, only in a small number of works has this microprocessor been used in the field of automatic PV fault diagnosis. Thus, the main contribution of this paper is the development of a fast and accurate method for the online automatic PV fault detection and classification. The developed method allows the experimental verification of the capability of ML algorithms to detect and classify the faults occurring in PV modules, and to monitor the operation of a PV array, thanks to the use of a webpage which informs the users about the state of their installations. The investigated faults are: the soiling/deposit of dust, permanent shading, the short-circuited bypass diode in PV modules, and disconnected PV modules.

Soiling mainly occurs due to environmental conditions, which depend on the location where the PV system is installed. Permanent shading is caused by the presence of buildings, trees, etc. Short-circuited bypass diodes may be caused by many factors, such as overheating, corrosion, manufacturing problems, bad connections, etc. Open-circuited PV modules are mainly due to the breaking down of panel-panel cables or connections, bypass diode issues, bad connections, etc.

The paper is organized as follows: Section 2 provides the description of the system and of the database used to develop the fault detection method. The proposed fault detection method based on machine leaning algorithms is described in Section 3, while the results and the discussion are reported in Section 4. Finally, the conclusion and perspectives are provided in Section 5.

2. Photovoltaic Array Description and Dataset

The PV array considered in this study consists of three parallel-connected PV modules installed at the University of Jijel (Algeria). Figure 1 shows the PV modules used in this study [12]. The considered faults are: (a) soiling/dust deposit; (b) permanent shading; (c) open-circuit (disconnected of one PV module); and (d) short-circuit (short circuited bypass diode in a PV module). The "Prova I-V tracer" (See Table A1) is used to collect the data (Figure 1e), while the faults are labeled manually. The data (I–V curves) were collected

during different climatic conditions and under normal and abnormal circumstances (faulty PV array). The PV module specifications are listed in Table 1.

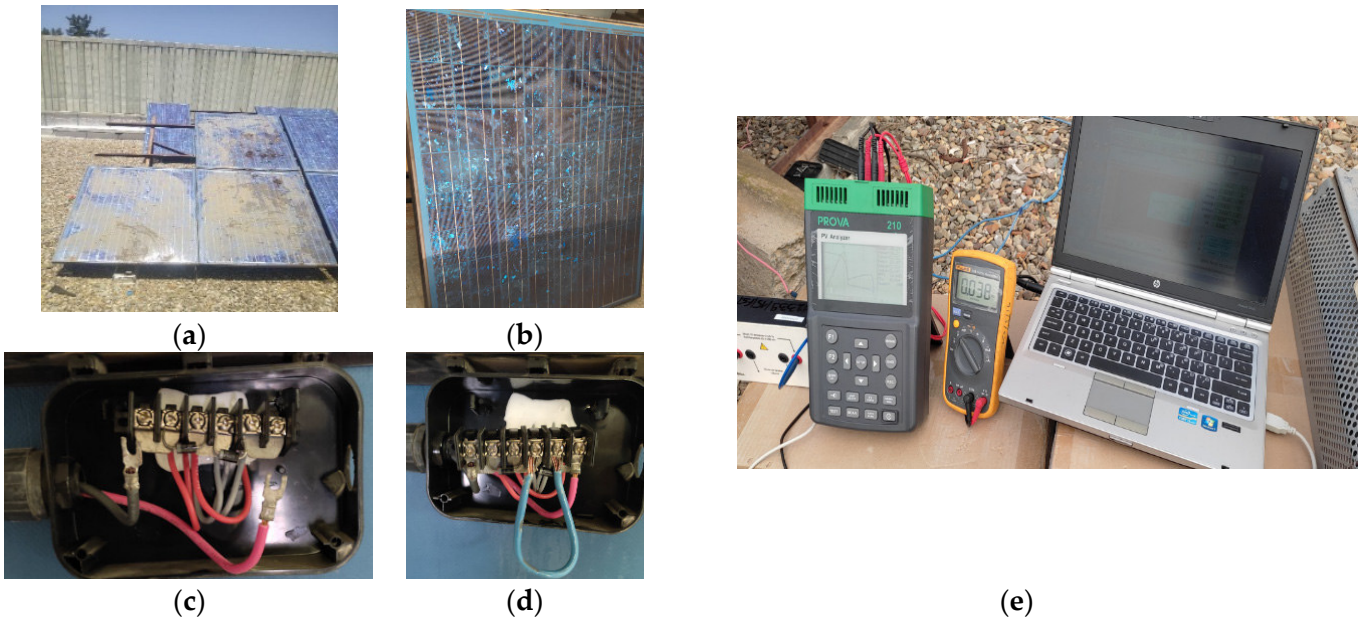


Figure 1. (**a**) Dust deposit on the PV modules surface, (**b**) shading (permeant shadow), (**c**) open circuited PV module (one PV module disconnected), (**d**) short-circuited bypass diode in a PV module, and (**e**) the used measurement instrument ("Prova 210 I–V tracer").

Table 1. PV module specifications.

Power (Pmp)	121.4 W
Maximum voltage (Vmp)	17.1 V
Maximum current (Imp)	7.11 A
Short-circuit current (Isc)	7.98 A
Open circuit voltage (Voc)	21.2 V
Maximum bypass diode current	8.1 A
Current Temperature Coefficient	+0.054%/°C
Voltage Temperature Coefficient %/°C	−0.35%/°C

Figure 2 shows an example of the I–V curves measured during the experiments. A number of measurements have been carried out under different working conditions. In total, we have collected 246 I–V curves from which we could extract the main features of the PV array, such as: the short circuit current (Isc), the open circuit voltage (Voc), the current at maximum power point (Imp), the voltage at the maximum power point (Vmp), the power at the maximum power point (Pmp), and the fill factor (FF). While preparing the dataset, missed data usually occur and they are automatically ignored from the dataset (CSV files), and redundant features have also been removed.

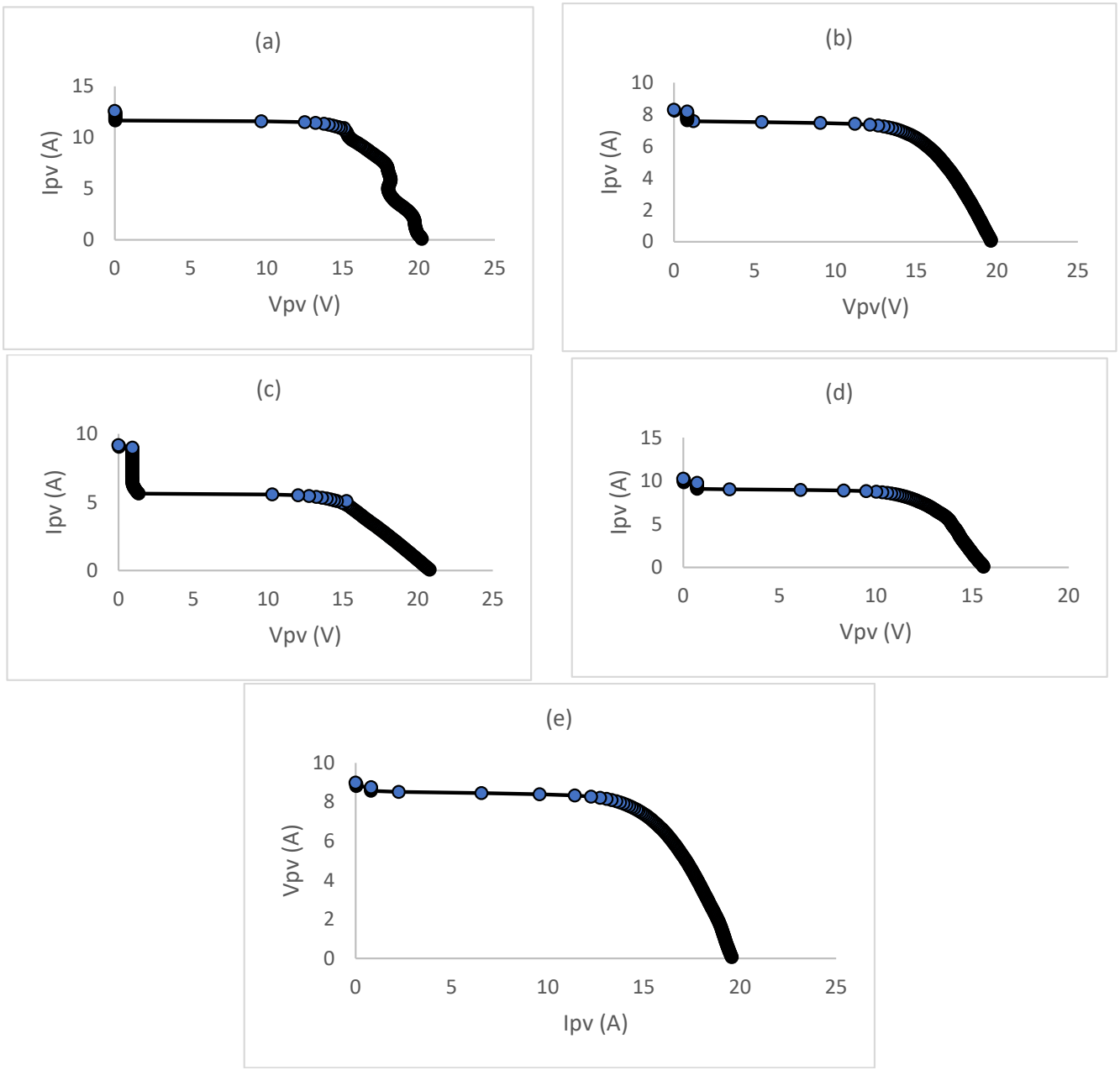


Figure 2. Some of the I-V curves measured for different faults under various working conditions: (**a**) dust deposit on the PV array surface; (**b**) open circuit (one PV module disconnected); (**c**) shading (permanent shadow); (**d**) short circuited (short-circuited bypass diode in one PV module); (**e**) normal operations (no fault).

3. The Proposed Method

The block diagram of the proposed method is shown in Figure 3. There are three main blocks:

Block #1. This block contains the PV array together with the sensors used to measure the PV current, the PV voltage, the solar irradiance, and the cell temperature every thirty minutes.

Block #2. This block comprises the controller, which is based on a Python (version 3.8, Python Software Foundation: Wilmington, DE, USA) code implemented into the Raspberry Pi 4. The code contains the fault detection program, which is based on a ML method (decision tree) [18], an explicit I-V model with features extraction

parameters [19], and a fault classification method based on an ensemble method (random forest) [18].

Block #3. This block includes the webpage application which has been designed for the remote visualization of the stored data, and to notify the users about the status of the PV array.

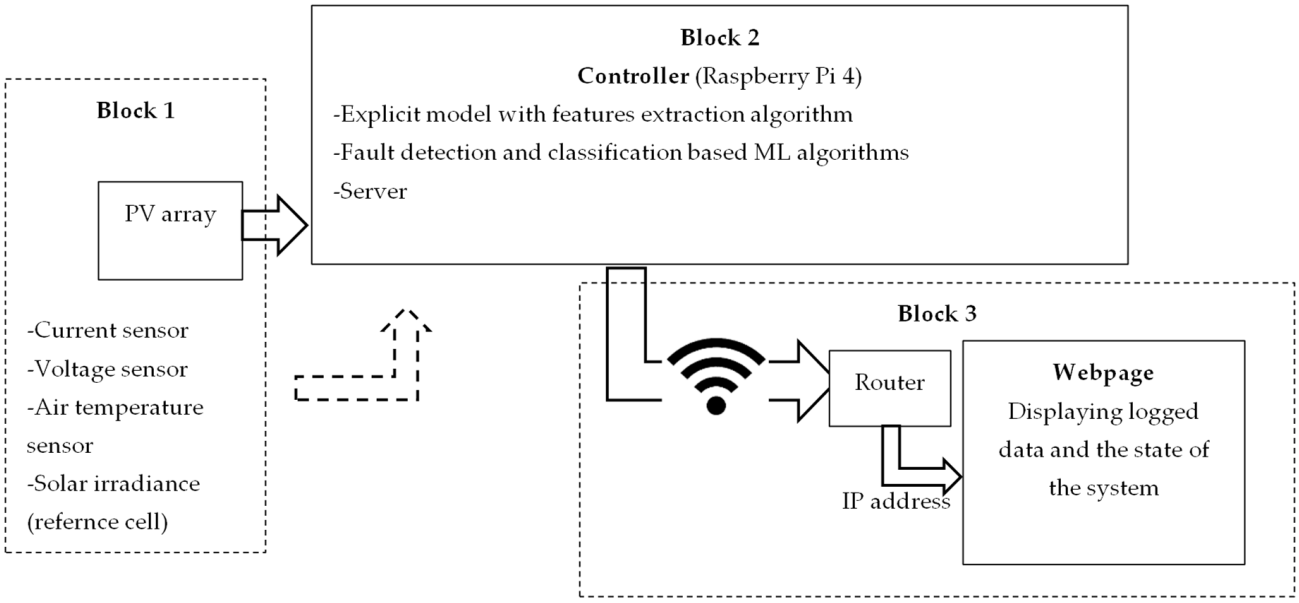


Figure 3. Block diagram of the proposed method.

The code performs four main steps which can be described as follows:

Step 1: import the libraries and the functions from Python.
Step 2: load the dataset and split it into the training and the testing subsets.
Step 3: select the ML algorithm and use the k-fold cross validation technique.
Step 4: fit the ML model and predict the results.

A large number of ML and ensemble techniques (e.g., Naïve Bayes, decision tree, k-nearest neighbors, random forest, neural networks, boosting, bagging, CatBoost, LightGBM, XGboost, etc.) are available in the literature. In this study, a decision tree (DT) algorithm has been used for the detection of faults, while a random forest algorithm (RF) has been used for their classification.

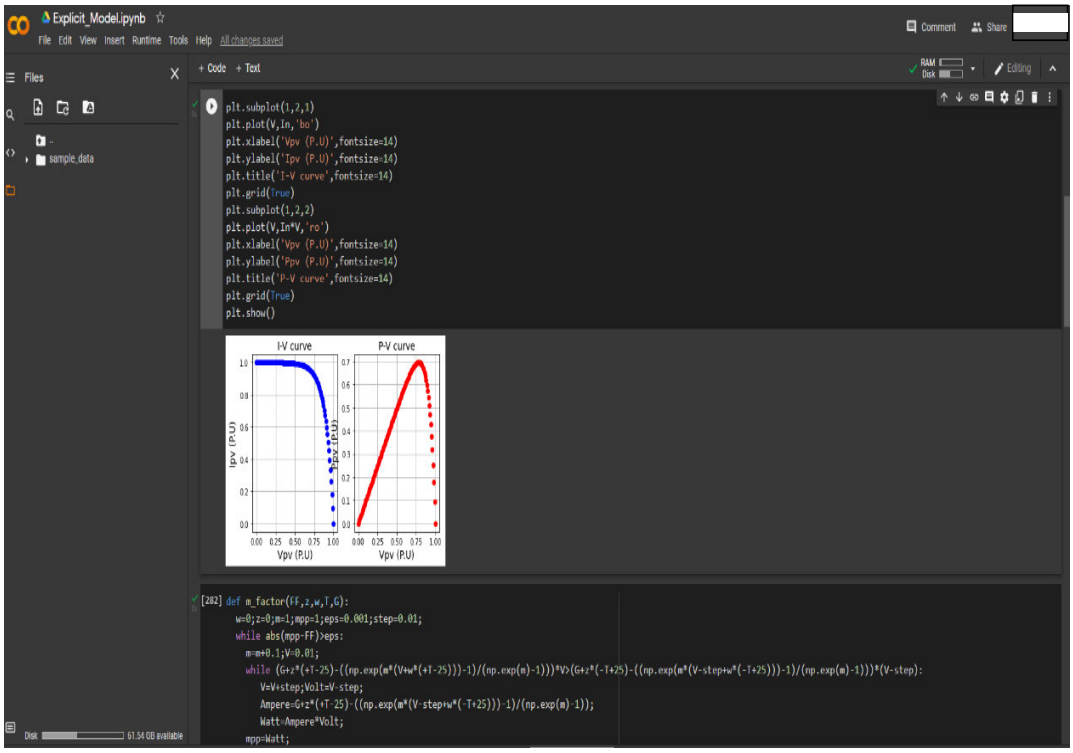
It is assumed that only one fault at time can happen during the measurement process. Multiple faults and faults with similar symptoms are not considered in this study.

With reference to Figure 4, the I-V explicit model described in Ref. [19] is used in order to estimate the I-V curve based on the values of G and T, while a simple algorithm is employed to extract the main features of the I-V curve (Voc, Isc, Vmp, Imp, Pmp, and FF).

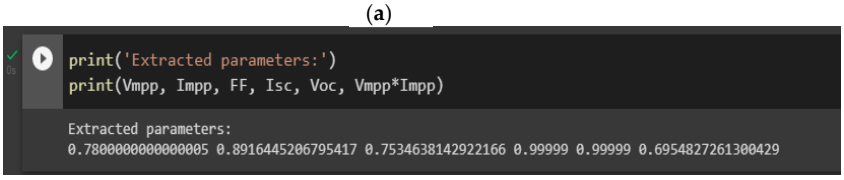
The detection of faults in PV arrays can be modeled as a binary classification problem. Among the different available ML methods, the decision tree algorithm has been chosen in order to detect the faulty PV module because of its simplicity. Concerning the PV fault classification, which is a multiclass classification problem, an ensemble learning method has been chosen, the random forest algorithm.

3.1. Programming Language

With reference to Figure 5, the fault detection and classification models are implemented online using Google Colab https://colab.research.google.com (accessed on 23 May 2018), a cloud platform that provides Jupyter netbook https://jupyter.org/ (accessed on 1 February 2015) services. Google Drive https://www.google.com/drive/ (accessed on 1 February 2015) was used in order to read the dataset.



(**a**)



(**b**)

Figure 4. (**a**) An example of the I-V and P-V curves obtained using the explicit model, (**b**) extracted features Vmp, Imp, FF, Isc, Voc, and Pmp.

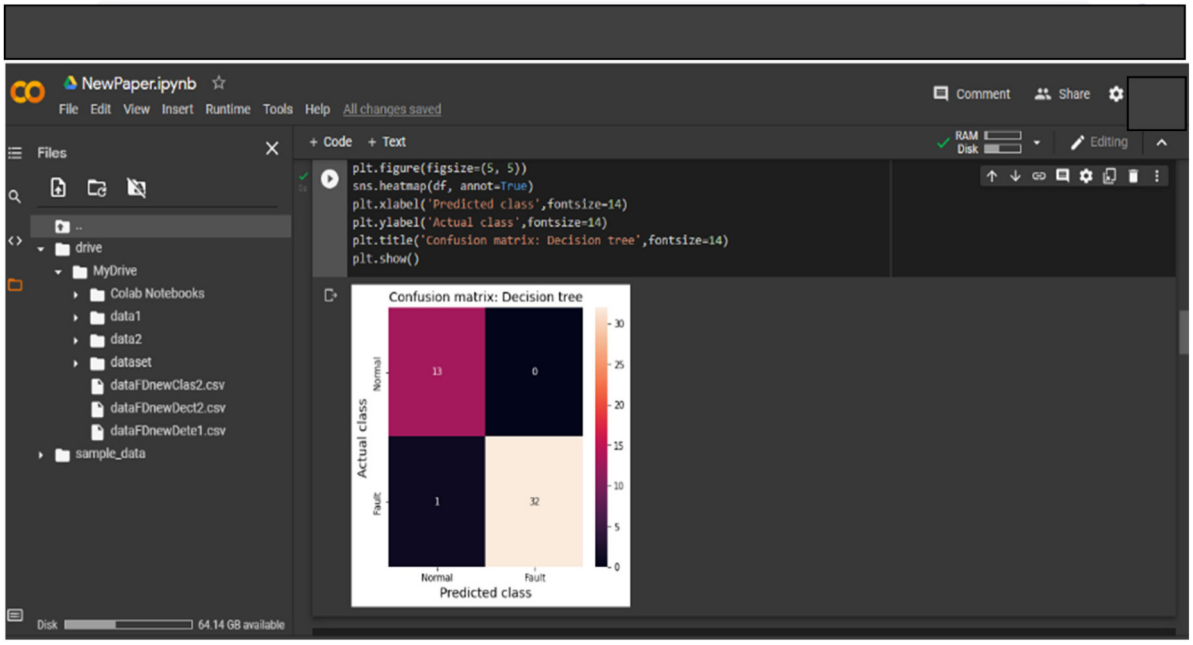


Figure 5. Google Colab interface used for developing and running the proposed method.

As an example, Appendix B shows the main functions used to develop both classifiers.

3.2. Performance Metrics

In order to evaluate the performance of the developed classifiers, the well-known confusion matrix (CM) method has been used in order to calculate the accuracy, the precision, the sensitivity, the F-score, and the misclassification rate, which are defined as follows:

$$\text{Accuracy } (\%) = \frac{\sum_i \text{CM}(i,i)}{\sum_i \sum_j \text{CM}(i,j)} \times 100 \quad (1)$$

$$\text{Precision}_i(\%) = \frac{\text{CM}(i,i)}{\sum_j \text{CM}(j,i)} \times 100 \quad (2)$$

$$\text{Sensitivity}_i(\%) = \frac{\text{CM}(i,i)}{\sum_j \text{CM}(i,j)} \times 100 \quad (3)$$

$$F1 - \text{score } (\%) = 2\frac{(\text{sensitivity} \times \text{Recall})}{(\text{sensitivity} + \text{Precision})} \times 100 \quad (4)$$

$$\text{Misclassified rate } (\%) = 100 - \text{ accuracy} \quad (5)$$

4. Experimental Implementation and Results

A K-fold cross validation technique has been used to resample the dataset without replacement. The advantage of this technique is that each example is used both for the training and for the validation exactly once. This yields a lower variance estimate of the model performance than the holdout method. Each classifier (DT and RF) accepts as input G, T, and the extract I-V features (Isc, Voc, Imp, Vmp, Pmp, and FF).

In order to develop the classifiers (binary classification for the detection and multiclass classification for the fault classification), a number of experiments have been carried out by tuning the hyper-parameters for both the classifiers' decision tree and the random forest algorithms. The adjusted random forest parameters are: max_depth, max_features, n_estimators, and random state. The adjusted decision tree parameters are: criterion, max_depth, and random_state.

A cross validation method has been used to evaluate both classifiers. This is a resampling procedure used to evaluate ML models on a limited data sample [20].

4.1. Fault Detection Performance

The cross-validation accuracy scores for K = 10 are given in Figure 6.

The error metrics are listed in Table 2. The accuracy is 98% and the misclassified rate is 2%. With reference to the confusion matrix shown in Figure 7, in the first row, thirteen normal cases have been classified correctly, while in the second row, thirty-two are correctly classified and only one fault is incorrectly classified as a normal case.

Table 2. Error metrics: precision, recall, F1-score, accuracy, and misclassified rate for the fault detection method using the decision tree learning algorithm (with tuned parameters).

ML Classifier	**Tuned Parameters: Criterion = gini, max_depth = 3 and random_state = 40**				
	Precision (%)	**Sensitivity (%)**	**F1-Core (%)**	**Classification Accuracy (%)**	**Misclassified Rate (%)**
Normal	93	100	95	98	2
Fault	100	97	98		

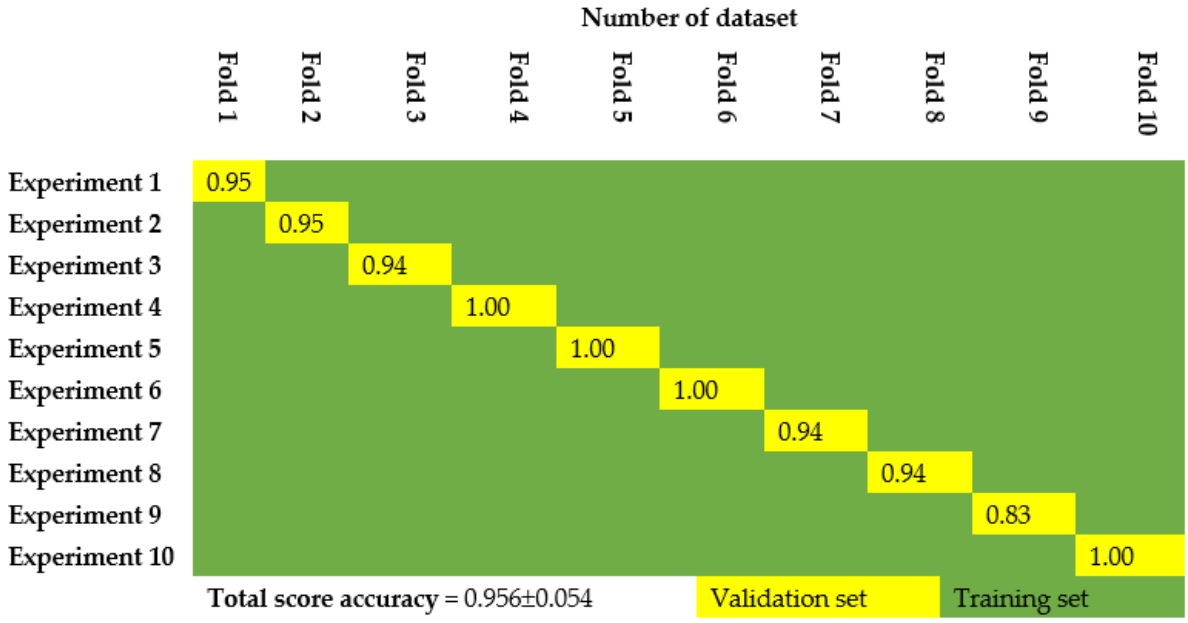


Figure 6. Cross validation accuracy scores for K = 10 (fault detection).

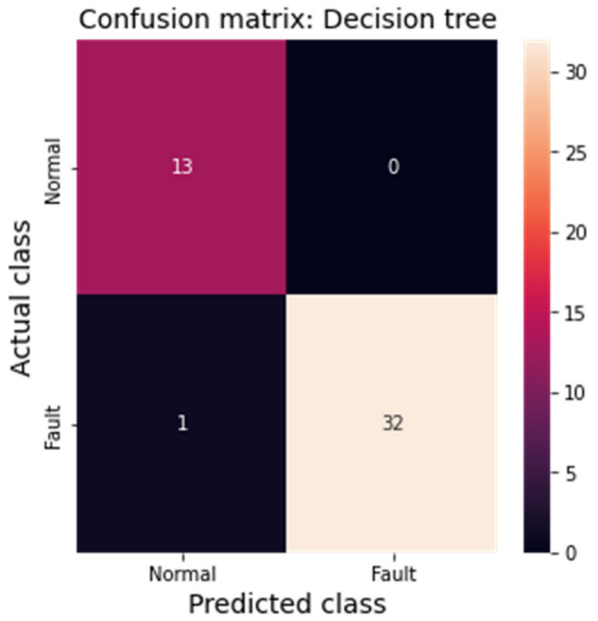


Figure 7. Confusion matrix: fault detection using the decision tree algorithm.

The decision tree results are shown in Figure 8.

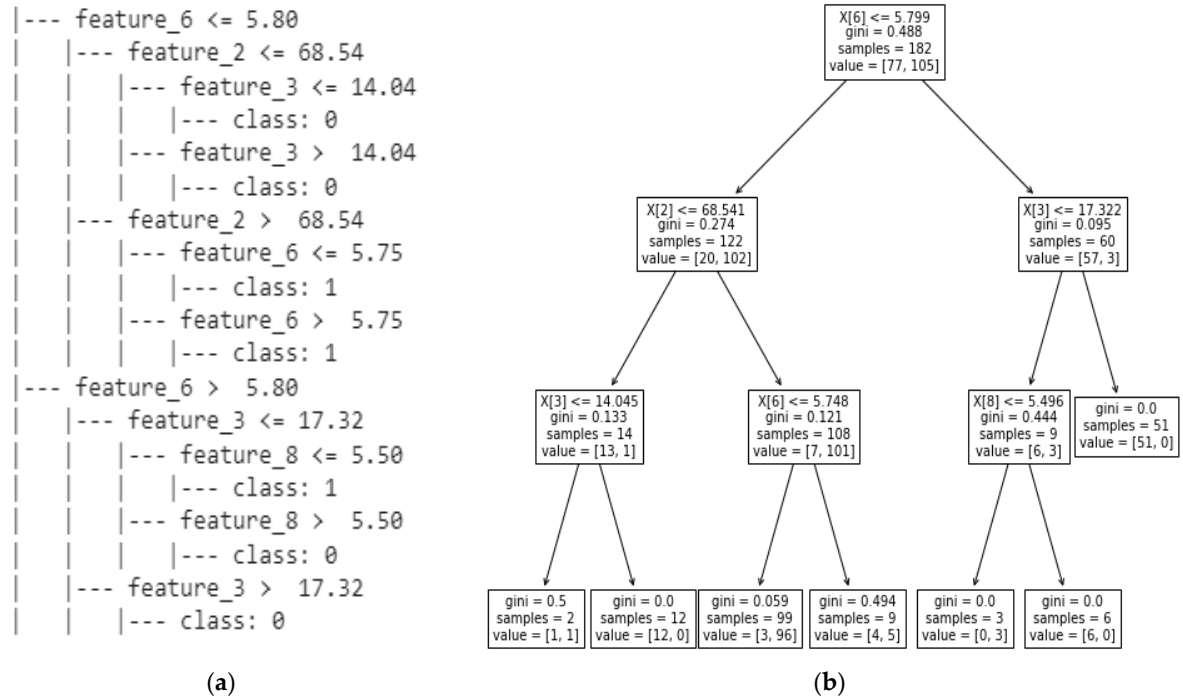


Figure 8. Decision tree model: (**a**) decision tree text and (**b**) decision tree diagram of the model.

4.2. Fault Classification Performance

The cross-validation accuracy scores for K = 10 is given in Figure 9.

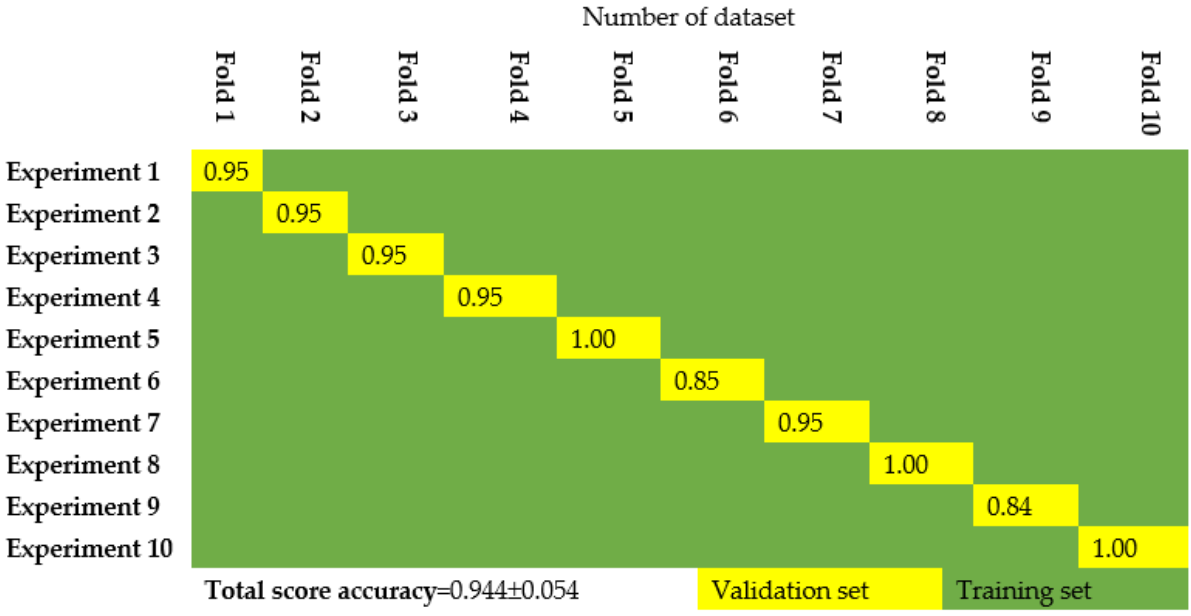


Figure 9. Cross validation accuracy scores for K = 10 (fault classification).

The error metrics are calculated and listed in Table 3. The accuracy is 96% and the misclassified rate is 4%. With reference to the confusion matrix plotted in Figure 10, in the first row, which refers to the fault class 0, nine samples are correctly classified and only one element is not; the sensitivity is 9/10 (90%). In the second row, all the eight samples are correctly classified, and the sensitivity is 8/8 (100%). In the third row, 20 samples are correctly classified, while one is misclassified into class 1; the sensitivity is 20/21 (95%).

With reference to the last row, all eleven samples are correctly classified, and the sensitivity is 11/11 (100%).

Table 3. Error metrics: precision, recall, F1-score, accuracy, and misclassified rate for fault classification using random forest ensemble learning algorithm (with tuned parameters).

ML Classifier Parameters	max_dept h = 5, max_features = sqrt, n_estimators = 300, and random_state = 42				
Fault Classes	**Precision (%)**	**Sensitivity (%)**	**F1-Score (%)**	**Classification Accuracy (%)**	**Misclassified Rate (%)**
Class {0}: dust deposit on the PV array surface	100	90	95	96	4
Class {1}: short-circuited bypass diode in one PV module	80	100	89		
Class {2}: permanent shadow	100	95	98		
Class {3}: disconnected one PV module	100	100	100		

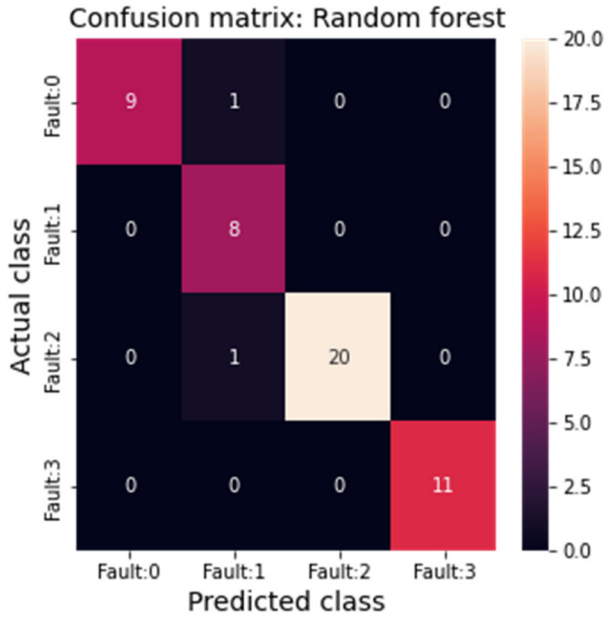


Figure 10. Confusion matrix: fault classification using random forest algorithm.

With reference to the first column of the matrix, all the nine samples are correctly classified, and the precision is 9/9 (100%). In the second column, eight samples are correctly classified, while two are misclassified into class 0 and class 1 respectively; the precision is 8/10 (80%). In the third and the last columns, all the twenty and eleven samples are well classified so that the precisions are 20/20 (100%) and 11/11 (100%), respectively. The F1-score ranges between 89% and 100%. Some of the faults (e.g., class 3, class 2, and class 0) are very well classified, having a precision of 100%. Nevertheless, class 1 faults are characterized by a precision of 80%, and that is acceptable. Globally, the results listed in Table 4 are quite satisfactory.

Table 4. Component specification and costs.

Components	Specification	Cost (Dollars)
Raspberry Pi 4	Cortex-A72 (ARMv8), 4 Go	85
Current sensor ACS712	30 A	6
Voltage sensor	25 V	4
Temperature sensor Type K	Max6675 −20 °C + 80 °C	5
Solar irradiance (Silicon irradiance sensor)	0–1200 W/m^2	50

4.3. Experimental Implementation

Once the detection and classification algorithms have been verified, these have been implemented into the Raspberry Pi 4 for a real time verification. The WiFi module embedded into the Raspberry Pi 4 has been used in order to send data to the cloud.

The following steps represent the main procedure implemented into the microprocessor.

Step 1: read the data (G, T, Ipv, and Vpv) by the Raspberry Pi 4.
Step 2: call the explicit model to estimate the I-V curve.
Step 3: call the features extraction algorithm to calculate Isc, Voc, Imp, Vmp, FF, and Pmp.
Step 4: call the fault detection procedure, and based on the calculated features, display the results on the webpage and go to **Step 1** if the PV system works properly, otherwise go to the next **Step 5**.
Step 5: call the fault classification procedure and identify the nature of the fault.
Step 6: display the results on the webpage and notify the user by indicating the type of fault.

Figure 11 shows the basic experimental setup of the prototype where the Raspberry Pi 4 is used.

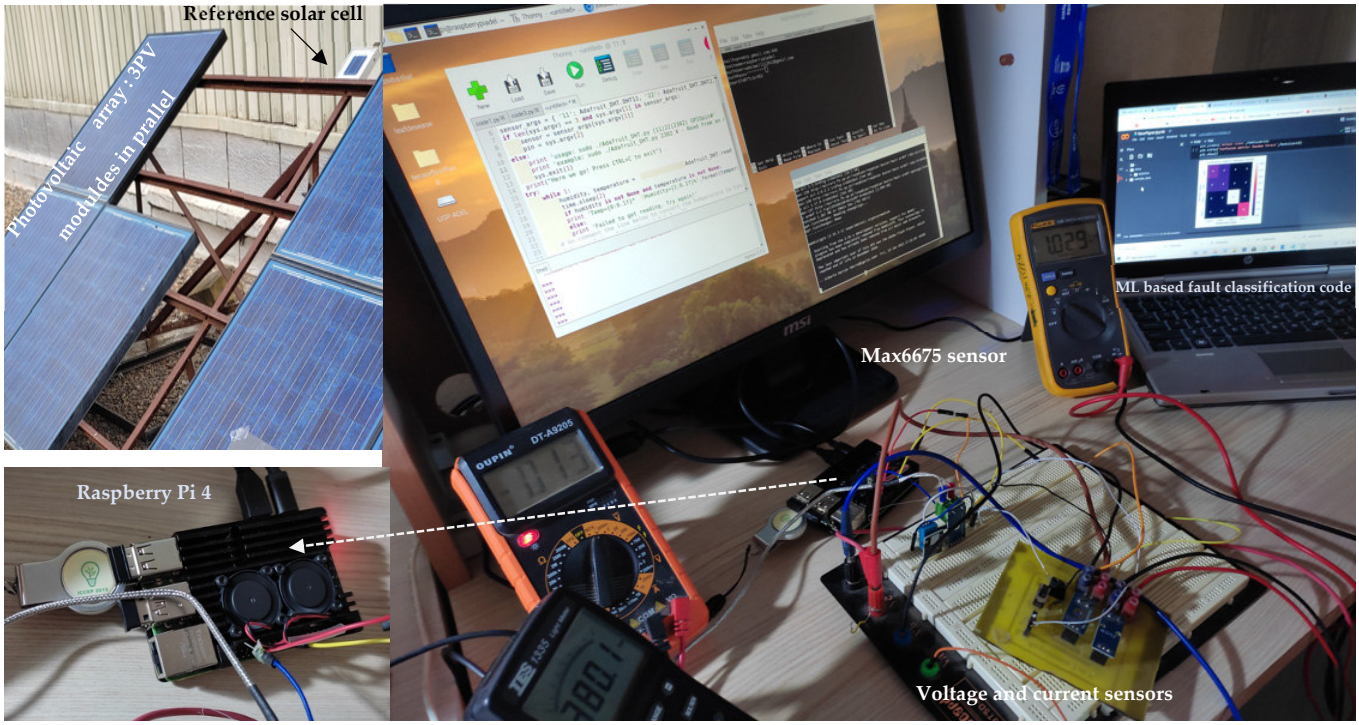


Figure 11. Experimental setup at the laboratory level: Raspberry Pi 4, sensors, PV modules, and data-acquisition system.

The specification and the cost of the main components used in the experiment are listed in Table 4. The total cost is USD 150.

In order to test the method, artificial shading has been created on the field by partially covering one of the PV modules. Figure 12 shows the corresponding I-V curve, the extracted parameters, the current, the voltage, the cell temperature, and the solar irradiance. The state of the system is also displayed in the webpage. The results show clearly that the method is able to detect and correctly classify the fault that occurred on the investigated PV array (type of fault: class #2).

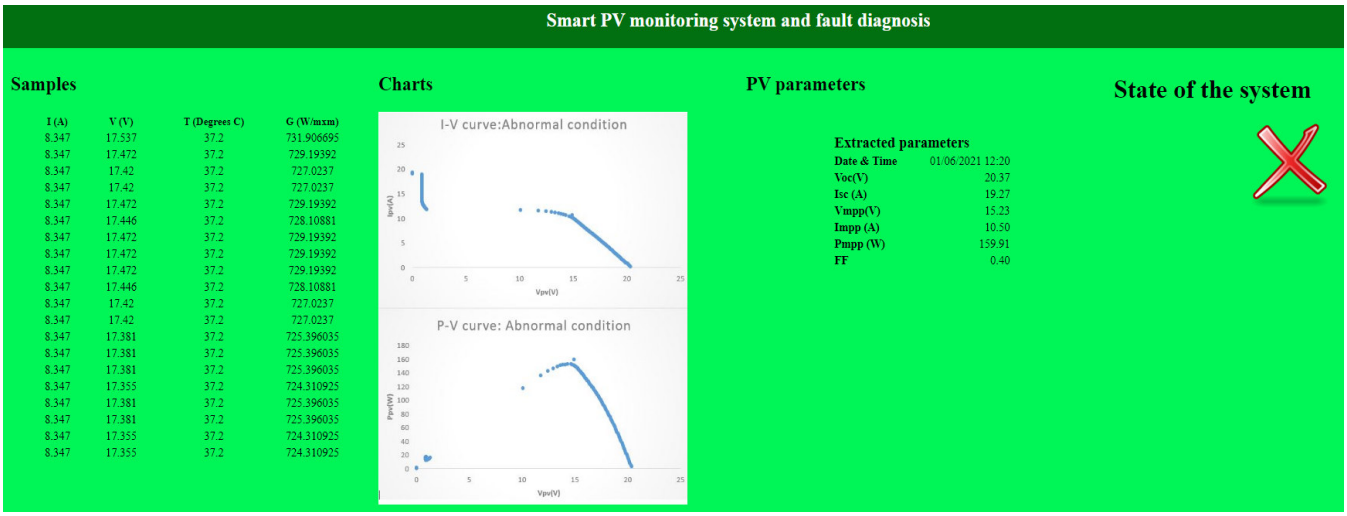


Figure 12. The webpage with the monitored data and the state of the system (https://solar-system.w3spaces.com/ (accessed on 20 May 2021)).

5. Conclusions

In this work, a machine learning-based fault diagnosis method for photovoltaic arrays has been developed and experimentally verified. A Python code, including the decision tree, the random forest, the explicit model of the I–V curve, and the extraction parameters algorithms were written to a Raspberry Pi 4 microprocessor suitable for real-time applications.

The experimental results showed the feasibility of the developed method to detect and classify the common faults occurring in PV arrays. The designed prototype can rapidly detect and classify the examined faults with a good accuracy (98% for the detection and 96% for the classification).

It should be pointed out that the data should be periodically updated in order to keep the classifier working effectively and avoiding false alarms. In addition, multiple PV faults have not been considered in this study, which remain an open challenge.

This work can be further improved and extended for fault detection in photovoltaic systems, including DC-DC converters, batteries, and DC-AC inverters. The webpage designed could be also enhanced by posting more information about the PV installation, as well as to notify users by e-mail or SMS.

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

Funding: A. Massi Pavan acknowledges 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 program Interreg Italy-Croatia. A. Mellit acknowledges financial support provided by the DGRSDT, Algiers, (Algeria) socio-economic project: ''Realization of a smart prototype for fault diagnosis of photovoltaic systems (7/2019)".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Adel Mellit acknowledges the ICTP for their support throughout the Simons Associate Program.

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

Appendix A

Table A1. Prova I-V tracer specifications.

Range (60 V/12A)	Resolution	Accuracy
DC Voltage Measurement		
0–10 V	0.001 V	±1% ± (1% of Vopen ± 0.1 V)
10–60 V	0.01 V	±1% ± (1% of Vopen ± 0.1V)
DC Current Measurement		
0.01–10 A	1 mA	±1% ± (1% of Ishort ± 9 mA)
10–12 A	1 mA	±1% ± (1% of Ishort ± 9 mA)

Appendix B

```
Main code (DT and RF functions)
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.pipeline import make_pipeline
In_train, In_test, Out_train, Out_test = train_test_split(InD, OutD,
test_size=0.20,random_state=42)
    Classifier_RF_Model = make_pipeline(StandardScaler(),
RandomForestClassifier(n_estimators=300, max_depth=5))
    #Classifier_DT_Model = make_pipeline(StandardScaler(),
DecisionTreeClassifier(max_depth=5, random_state=40))
classifier.fit(In_train, Out_train)
Out_pred = classifier.predict(In_test)
```

References

1. Snapshot of Global PV Markets. 2021 Report IEA-PVPS T1-39. 2021. Available online: https://iea-pvps.org/snapshot-reports/snapshot-2021/ (accessed on 25 April 2021).
2. Adhya, S.; Saha, D.; Das, A.; Jana, J.; Saha, H. An IoT Based Smart Solar Photovoltaic Remote Monitoring and Control Unit. In Proceedings of the 2nd International Conference on Control, Instrumentation, Energy & Communication (CIEC), Kolkata, India, 28–30 January 2016; pp. 432–436.
3. Mellit, A.; Kalogirou, S. Artificial intelligence and internet of things to improve efficacy of diagnosis and remote sensing of solar photovoltaic systems: Challenges, recommendations and future directions. *Renew. Sustain. Energy Rev.* **2021**, *143*, 110889. [CrossRef]
4. Tina, G.M.; Ventura, C.; Ferlito, S.; De Vito, S. A State-of-Art-Review on Machine-Learning Based Methods for PV. *Appl. Sci.* **2021**, *11*, 7550. [CrossRef]
5. Tchoketch_Kebir, S.; Cheggaga, N.; Ilinca, A.; Boulouma, S. An Efficient Neural Network-Based Method for Diagnosing Faults of PV Array. *Sustainability* **2021**, *13*, 6194. [CrossRef]
6. Samara, S.; Natsheh, E. Intelligent PV panels fault diagnosis method based on NARX network and linguistic fuzzy rule-based systems. *Sustainability* **2020**, *12*, 2011. [CrossRef]
7. Barker, C.; Cipkar, S.; Lavigne, T.; Watson, C.; Azzouz, M. Real-Time Nuisance Fault Detection in Photovoltaic Generation Systems Using a Fine Tree Classifier. *Sustainability* **2021**, *13*, 2235. [CrossRef]
8. Mellit, A. Recent Applications of Artificial Intelligence in Fault Diagnosis of Photovoltaic Systems. In *A Practical Guide for Advanced Methods in Solar Photovoltaic Systems*; Springer: Cham, Switzerland, 2020; Volume 128, pp. 257–271. [CrossRef]
9. Mehmood, A.; Sher, H.A.; Murtaza, A.F.; Al Haddad, K. Fault detection, Classification and Localization Algorithm for Photovoltaic Array. *IEEE Trans. Energy Convers.* **2021**, *70*, 1–12. [CrossRef]
10. Dhoke, A.; Sharma, R.; Saha, T.K. An approach for fault detection and location in solar PV systems. *Sol. Energy* **2019**, *194*, 197–208. [CrossRef]
11. Mellit, A.; Tina, G.M.; Kalogirou, S.A. Fault detection and diagnosis methods for photovoltaic systems: A review. *Renew. Sustain. Energy Rev.* **2018**, *91*, 1–17. [CrossRef]
12. Hamied, A.; Mellit, A.; Zoulid, M.A.; Birouk, R. IoT-based experimental prototype for monitoring of photovoltaic arrays. In Proceedings of the 2018 International Conference on Applied Smart Systems (ICASS), Medea, Algeria, 24–25 November 2018; pp. 1–5. [CrossRef]
13. Mellit, A.; Hamied, A.; Lughi, V.; Pavan, A.M. A low-cost monitoring and fault detection system for stand-alone photovoltaic systems using IoT technique. In *ELECTRIMACS*; Springer: Cham, Switzerland, 2020; pp. 349–358. [CrossRef]

14. Badave, P.M.; Karthikeyan, B.; Badave, S.M.; Mahajan, S.B.; Sanjeevikuma, P.; Gill, G.S. Health monitoring system of solar photovoltaic panel: An internet of things application. In *Advances in Smart Grid and Renewable Energy*; Springer: Singapore, 2018; pp. 347–355. [CrossRef]
15. Pereira, R.I.; Dupont, I.M.; Carvalho, P.C.; Jucá, S.C. IoT embedded linux system based on Raspberry Pi applied to real-time cloud monitoring of a decentralized photovoltaic plant. *Measurement* **2018**, *114*, 286–297. [CrossRef]
16. Paredes-Parra, J.M.; Mateo-Aroca, A.; Silvente-Niñirola, G.; Bueso, M.C.; Molina-García, Á. PV module monitoring system based on low-cost solutions: Wireless raspberry application and assessment. *Energies* **2018**, *11*, 3051. [CrossRef]
17. Priharti, W.; Rosmawati, A.F.K.; Wibawa, I.P.D. IoT based photovoltaic monitoring system application. *J. Phys. Conf. Ser.* **2019**, *1367*, 012069. [CrossRef]
18. Shalev-Shwartz, S.; Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*; Cambridge University Press: England, UK, 2014; Available online: https://www.cambridge.org/9781107057135 (accessed on 12 January 2014).
19. Pavan, A.M.; Mellit, A.; Lughi, V. Explicit empirical model for general photovoltaic devices: Experimental validation at maximum power point. *Sol. Energy* **2014**, *101*, 105–116. [CrossRef]
20. Brownlee, J. A Gentle Introduction to k-Fold Cross-Validation. 2018. Available online: https://machinelearningmastery.com/k-fold-cross-validation/ (accessed on 23 May 2018).