



# Short-term nonlinear autoregressive photovoltaic power forecasting using statistical learning approaches and in-situ observations

Ayoub Fentis<sup>1</sup> · Lhoussain Bahatti<sup>1</sup> · Mohamed Tabaa<sup>2</sup> · Mohammed Mestari<sup>1</sup>

Received: 13 March 2018 / Accepted: 24 December 2018 / Published online: 8 January 2019  
© The Author(s) 2019

## Abstract

Due to the low total cost of production, Photovoltaic energy constitutes an important part of the renewable energy installed in the world. However, photovoltaic energy is volatile in nature because it depends on weather conditions, which makes the integration, control and exploitation of this type of energy difficult for grid operators. In the traditional grid architecture, system operators have accumulated enough experience that enables them to determine how much operating reserves are required to maintain system reliability based on statistical tools. Still, with the introduction of renewable energy (wind and photovoltaic), the grid structure has changed, and to maintain grid stability, it is becoming fundamental to know renewable energy state and production that can be combined with other less variable and more predictable sources to satisfy the energy demand. Therefore, renewable energy forecasting is a straightforward way to integrate safely this kind of energy into the current electric grid, especially photovoltaic power forecasting, which is still at a relative infancy stage compared to wind power forecasting, which has reached a relatively mature stage. The goal of this work is to present, first, a short-term offline forecasting model that uses only in-situ (local) collected data. Also, the performances of several pure non-linear auto-regressive models are investigated against those of non-linear auto-regressive models with exogenous inputs. For this purpose, two well-known statistical learning techniques, namely Feed Forward Neural Network and Least Square Support Vector Regression, have been used. To test the performance of the models, the results obtained are compared with those of a benchmark model. In this paper, we used the persistent model as well as a multivariate polynomial regression model as benchmark.

**Keywords** Photovoltaic · Power forecasting · Renewable energy integration · Statistical learning techniques · Support vector regression · Feed forward neural network

## Abbreviations

### List of symbols

$J(w, b, e)$	Empirical risk functional in the feature space
$N$	Number of training instances
$b$	Bias
$w$	Separation margin
$\gamma$	Regularization parameter of LSSVM
$e$	Vector of errors $e = [e_1, \dots, e_N]$
$x$	Training data $x = [x_1, \dots, x_N]$

$y$	Vector of labels with $y = [y_1, \dots, y_N]$
$1_v$	Vector of ones
$w^T w$	Inner product of $w$
$\alpha_i$	Lagrange multipliers
$\mathcal{L}(w, b, e, \alpha)$	Lagrangian objective function
$\Omega$	Kernel matrix
$K$	Bimodal kernel function
$\varphi$	Nonlinear mapping from input to feature space
$W$	Vector of FFNN weights $W = [w_1, \dots, w_p]$
$x_o$	Vector of computed values of output neurons
$\hat{x}_o$	Vector of required values of output neurons

✉ Ayoub Fentis  
fentis.ayoub@gmail.com

<sup>1</sup> Laboratory of Signals, Distributed Systems and Artificial Intelligence (SSDIA), ENSET Mohammedia, Mohammedia, Morocco

<sup>2</sup> Pluridisciplinary Laboratory of Research and Innovation (LPRI), EMSI Casablanca, Casablanca, Morocco

## Abbreviations

PV	PhotoVoltaic
AR	Auto-regressive
ARX	Auto-regressive with exogenous input
NAR	Nonlinear auto-regressive

NARX	Nonlinear auto-regressive with exogenous input
LSSVR	Least squares support vector regression
LSSVM	Least squares support vector machine
FFNN	Feed forward neural network
MPR	Multivariate polynomial regression
MLP	MultiLayer perceptron
MAE	Mean absolute error
MBE	Mean bias error
MSE	Mean squared error
RMSE	Root mean squared error
$R^2$	Coefficient of determination
SMO	Sequential minimal optimization
RBF	Radial basis function
NN	Neural network
LM	Lavenberg–Marquardt
$NMAE_{\%}$	Normalized mean absolute error
$NMAE_p$	Relative sample mean $NMAE_{\%}$
$S_p^2$	Sample variance
CI	Confidence intervals
ME	Margin error
$Irr_t$	Global horizontal solar irradiation at time $t$
$Tc_t$	Photovoltaic module temperature at time $t$
$P_t$	PV power generated at time $t$
$MOD_i$	$i^{th}$ Model

## Introduction

Photovoltaic (PV) power has proven to be one of the promising renewable energies in the recent years. This field has witnessed a significant increase in the value of investments; the production capacity reached 227 GW in 2015 compared to 5.1 GW in 2005. But with the emergence of renewable energy as a necessary alternative to the fossil one, new challenges have emerged, which requires both producers and managers to change control methods, distribution methods and all the related logistics. The main challenge is still the safe integration of renewable energy in the actual grid; it is challenging due to the volatile and uncertain nature of renewable power caused generally by weather conditions. In the traditional grid management, the grid operator must maintain the balance between supply and demand at all times to avoid security grid problems and economic losses. The grid operator uses a planning to ensure that power plants produce the right amount of electricity at the right time to meet consistently and reliably electric demand. Nowadays, the energy mixture has changed giving more place to renewable sources, which changed the structure of power grid and all traditional control and scheduling procedures. Recently photovoltaic power has begun to gain some place over other renewables; this is due to the lower total cost of production. But from grid management point of view, solar generation

variability caused generally by clouds can make it more difficult for the grid operator to predict how much additional electric generation will be required to ensure the balance between supply and demand. For that reason, renewable power forecasting imposes itself as a key solution to efficiently handle renewable energy in power grid and must be properly accounted for in the complex decision-making processes required to balance supply and demand in the power system.

Nowadays renewable power forecast is a key activity for a number of reasons. It is used for monitoring the performance of the plant, detecting anomalies and faults, making reliable dispatching plans for the grid operators, helping operation and maintenance scheduling... etc.

In the last years many research works have tried to handle the problem of PV power forecasting. The two main challenges of PV forecasting (this is also the cause of the poor penetration rate of PV systems) are variability and uncertainty, namely because the output of PV modules shows variability at all-time scales and the fact that this variability itself is difficult to predict, this fact subsequently makes the PV time series difficult to predict as shown in [36]. According to the state of the art, PV power forecasting models can be divided into three types: physical models, statistical models and hybrid models; Fig. 1. Physical models [14] are mathematical models based on a physical analysis of the process being studied; this model can contain a limit number of adjustable parameters, which have a physical meaning: in the case of photovoltaic, physical modeling uses mathematical equations that describe all the physical phenomena that govern PV conversion. Statistical models are used when there is not enough knowledge and information about the process and the parameters that influence it. Statistical modeling includes time series [1] and statistical learning models. Time series modeling aim to collect and study the past observations of a time series to fit a model which describes their internal structure (such as autocorrelation, trend or seasonality), the developed model is then used to forecast future values of the series, among the most used models we mention the AR, ARX, ARIMA models. A statistical learning model, called also black box model, is established from a set of measured variables  $X_k$  (inputs) and a set of measurements  $Y_k$  (outputs). We suppose that there is a relation between the  $X_k$  and the  $Y_k$ , and we try to determine a mathematical form of this relation. We say that we try to establish a model of the process from the available measures. Among statistical Learning tools, artificial neural network is the most used technique due to its performance proven over time. In the case of PV power forecasting different Neural Network architectures were used with a multiple choice of input parameters, among them we can cite the Elman Neural Network (ENN), generalized regression neural network (GRNN) [27], radial basis function neural network

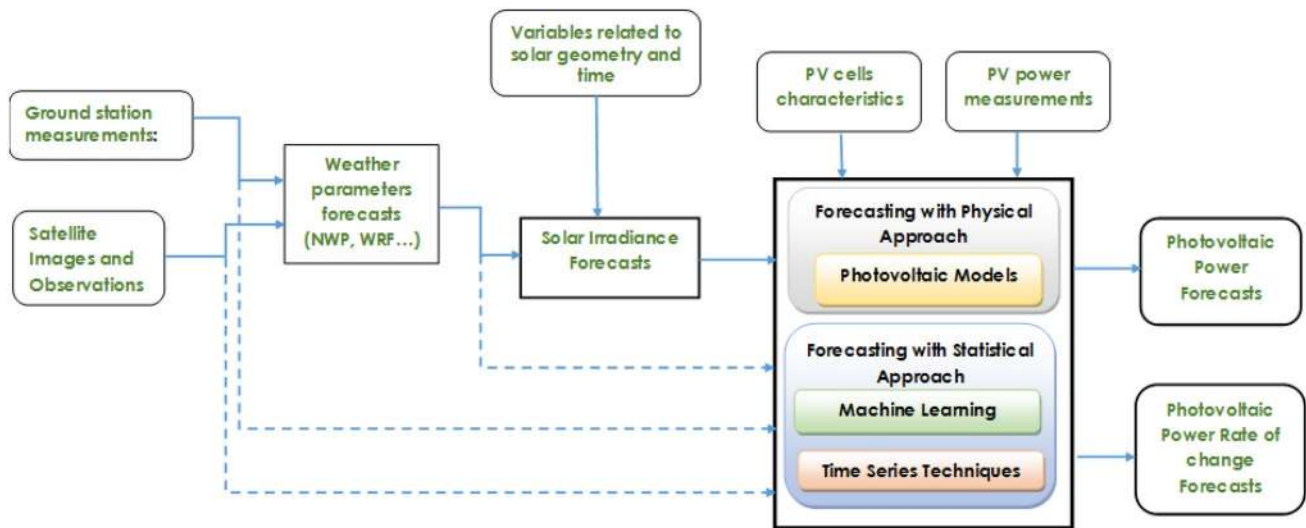


Fig. 1 Photovoltaic forecasting approaches

(RBFNN), dynamic recurrent neural network (DRNN) [20] and the feed-forward neural network (FFNN), which, in most cases, give the best results [20, 27]. In the same category as neural networks, another statistical learning technique named Support Vector machine (SVM) is beginning to gain success due to its generalization skills approved in several case studies; it has also been used in solar power forecasting in many studies, it was used for classification [29, 30] and regression [7, 36]. Hybrid models are a class of models that can be constructed from any combination of physical and statistical models; they can be a combination between physical and statistical approaches [34] or purely statistical like combining SOM and RBFNN [3], or SOM, SVR and fuzzy inference [35], or wavelet transform and RBFNN [17] and so on.

The choice of the appropriate technique depends on several parameters, in general there is no fixed rule to choose the technique to use. According to the current state of the art [26], the choice of the technique depends more on the horizon, so physical models are used for medium term, statistical models for very-short and short terms and hybrid models for medium and long term. Still, we must report that input parameters are also a very important factor that can change the final results; different collections of inputs were used in the literature. Research has shown that the main variables influencing PV power are global horizontal irradiation (GHI) at PV generator surface [30], plate temperature [30] and aerosol index [16], but this does not exclude the other parameters like Numerical Weather Predictions (NWP) [20, 27], Meteorological measurements made in ground stations, measurement of GHI and cloud coverage by Satellite [27, 30], PV power measurements [7, 27], Variables related to

solar geometry and time (zenith angle, light duration) [30]... etc.

In this work we will combine the characteristics of time series models and statistical learning models in order to forecast the short term photovoltaic power. This combination is beneficial since it allows to merge the simplicity of time series models and the non-linear character of black box models, the result of this fusion is a nonlinear time series model. This study will allow us to; First, assess the performance of two supervised machine learning techniques for intraday PV power forecasting: Feed Forward Neural Network (FFNN) and least squares support vector machine regression (LSSVR). Second, it attempts to study the influence and sufficiency of in-situ collected data as input parameters to the developed models. For this purpose, we compared the performances of several models in order to find the best off-line model for PV power forecasting; by off-line we designate a model capable of giving accurate short-term forecasts without the need of weather forecasts. This is interesting because the majority of existing models use meteorological parameters to forecast PV power, especially forecasted parameters obtained from numerical weather prediction (NWP) systems like, solar irradiation forecasts [1, 36], ambient temperature forecasts [8], humidity [14, 27], cloud index [4, 30], wind speed [3, 16] and probability of precipitation [13, 35]. The problem is that access to NWP information is not given to everyone at any time, especially isolated installations. For this purpose, off-line models that use only local collected information to forecast PV power are of a great importance for grid operators as well as for individuals who do not have access to weather data and forecasts. To rectify the performances of our models we will compare their performances

with an usual benchmark models, the persistent model and multivariate polynomial model.

This paper is structured as follows. Section 3 gives a brief look at related works; in Sect. 4, we introduce the used statistical learning techniques. Section 6 presents models development strategy, while result analysis and discussion are presented in Sect. 7, and finally a conclusion.

## Related work

Photovoltaic power forecasting increasingly attracts the attention of researchers. In the last few years, several PV power forecasting models have been developed. In [1] an ARX model was used to forecast 6 h ahead PV power output using historical PV power output and forecast irradiation as inputs for the model. In same perspective, [4] uses a recurrent neural network to forecast 24-h ahead PV power using also Historical PV power and forecast temperature. In [17], wavelet transformation and radial basis function neural network (RBFNN) were combined to generate a one hour ahead PV Power forecast; the RBFNN inputs included past PV power output, irradiation and temperature. The authors in [27] adopt a hybrid modeling approach by applying stepwise regression to select meteorological parameters that are strongly correlated with solar power; these variables were used to construct an FFNN model for 24-h ahead PV power forecasting. This model outperforms five other ones. The authors underline that average solar irradiation and average humidity are the two most significant parameters to forecast PV power output. In [3] the authors analyses the performance of a 24-h ahead PV power forecasting tool based on multilayer perceptron (MLP) neural network trained with error back propagation (EBP) procedure; three types of inputs were used: weather forecasts provided by meteorological services, geographical coordinates of site, date and time to determine the correct position. They propose a procedure to validate the correctness of data and highlight that the method performance is strictly related to the historical data pre-process step and to the weather forecasting accuracy.

Another interesting approach based on weather type classification and similar day detection to forecast PV power for a horizon up to one day is used in [8] where authors use a Recurrent Neural Network with structural elements for 24-h ahead PV power output forecasting. The inputs used include clear sky irradiation and forecasting weather type for the forecast days. In [31] the historical power output is classified into several weather types using forecast irradiation, total cloud and low cloud cover as parameters selection; the authors uses RBFNN as a technique to produce PV Power forecasts with 24-h ahead horizon. In [12] forecasts of high, medium and low temperatures are used to classify historical PV power output into three weather types. After that, three

feed forward neural networks (FFNN) were employed to generate 24-h ahead forecasts. In [35], the authors present a hybrid method to forecast 1-day ahead PV power output; the proposed method comprises three stages: data classification stage, training stage and forecasting stage. The classification stage is developed using self organizing map (SOM) and learning vector quantization (LVQ); the objective is to classify the historical PV power data into five weather types according to the verbal weather forecast of the TCWB (Taiwan Center Weather Bureau). In the second stage, support vector regression (SVR) is used to construct five forecasting models—one for each weather type. In the last stage, a fuzzy inference algorithm is used to select an appropriate forecasting model to achieve more accurate results. The work presented in [13] proposes a hybrid model for one-day ahead hourly PV power forecasting; this work is an extension of [35]. The proposed method comprises three stages: data classification stage, training stage and forecasting updating stage. The classification stage is developed using Fuzzy K-Means clustering algorithm; the objective is to classify the historical PV power data into five weather types according to the verbal weather forecast of the TCWB (Taiwan Center Weather Bureau). In the second stage, RBFNN is used to construct five forecasting models, one for each weather type, and a fuzzy inference algorithm is used to select an appropriate forecasting model. In the last stage, the forecasts are updated every 3 h to cope with the possible fluctuation of PV power.

As can be seen from this brief state of the art, the majority of existing models use predicted inputs to forecast PV power, especially inputs obtained from NWP systems. Access to NWP information is not given to everyone at any time, especially for the Africa region. For this reason, off-line models that use only past information to forecast PV power are of a great importance. From this perspective, the goal of this work is to present, first, a short-term off-line forecasting model that uses only in-situ collected data. Also, the performances of several pure non-linear auto-regressive models are investigated against those of non-linear auto-regressive models with exogenous inputs. As such, two well-known statistical learning techniques, namely feed forward neural network (FFNN) and least square support vector machine (LSSVR), have been used.

## Statistical learning techniques

### Least squares support vector regression

The least squares support vector machine algorithm (LSSVM) is an improved version of the classical support vector machine (SVM) used to solve classification problems. Due to equality type constraints in the formulation, the solution will be obtained by solving a set of linear equations, instead of

quadratic programming for classical SVM. Vapnik’s SVM formulation [5] was modified in [31] into the following LSSVM optimization problem underlying non-linear LSSVM training formulated by:

$$\min_{w,b,e} J(w, b, e) = \frac{1}{2}w^T w + \frac{\gamma}{2} \sum_{i=1}^N e_i^2. \tag{1}$$

Subject to the equality constraints

$$y_i [w^T \varphi(x_i) + b] = 1 - e_i. \tag{2}$$

This formulation consists of equality instead of inequality constraints and takes into account a squared error with regularization term similar to ridge regression. The solution is obtained after constructing the Lagrangian:

$$\mathcal{L}(w, b, e, \alpha) = J(w, b, e) - \sum_{i=1}^N \alpha_i \{y_i [w^T \varphi(x_i) + b] - 1 + e_i\}, \tag{3}$$

where  $\alpha_i \in \mathbb{R}$  are lagrange multipliers that are always positive, from the conditions for optimality, one obtains the Karush–Kuhn–Tucker (KKT) system:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \rightarrow w = \sum_{i=1}^N \alpha_i y_i \varphi(x_i) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow \sum_{i=1}^N \alpha_i y_i = 0 \\ \frac{\partial \mathcal{L}}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \rightarrow y_i [w^T \varphi(x_i) + b] - 1 + e_i = 0 \end{cases} \tag{4}$$

Note that sparseness is lost, which is clear from the condition  $\alpha_i = \gamma e_i$ . As in standard SVM, we calculate neither  $w$  nor  $\varphi(x_i)$ . Therefore, we eliminate  $w$  and  $e$  yielding according to [31].

$$\begin{bmatrix} 0 \\ y \end{bmatrix} \begin{vmatrix} \frac{1}{\Omega + \gamma^{-1} I} \end{vmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5}$$

with  $y = [y_1, \dots, y_N]$ ,  $1_v = [1, \dots, 1]$ ,  $e = [e_1, \dots, e_N]$  and  $\alpha = [\alpha_1, \dots, \alpha_N]$ . Mercer’s condition is applied within the  $\Omega$  matrix.

$$\Omega_{ij} = y_i y_j \varphi(x_i)^T \varphi(x_j) = y_i y_j K(x_i, x_j). \tag{6}$$

For the kernel function  $K(., .)$ , here again, one typically has the following choices:

$$\begin{cases} \text{Linear Kernel : } K(x_i, x_j) = x_i^T x_j \\ \text{Polynomial Kernel : } K(x_i, x_j) = \left(1 + \frac{x_i^T x_j}{c}\right)^d \\ \text{RBF Kernel : } K(x_i, x_j) = \exp\left\{-\frac{\|x_j - x_i\|_2^2}{\sigma^2}\right\} \\ \text{MLP Kernel : } K(x_i, x_j) = \tanh(\kappa x_i^T x_j + \theta) \end{cases} \tag{7}$$

where  $d, c, \sigma$  and  $\theta$  are constants. In the case of least squares support vector regression (LSSVR), some changes in LSSVM formulation will take place. In this case we try to find the best regression function in the form:

$$y(x) = w^T \varphi(x) + b, \tag{8}$$

with  $x \in \mathbb{R}^n, y \in \mathbb{R}$ . Given a training set  $\{x_i, y_i\}_{i=1}^N$ , in this case the optimization problem is given by:

$$\min_{w,b,e} J(w, b, e) = \frac{1}{2}w^T w + \frac{\gamma}{2} \sum_{i=1}^N e_i^2. \tag{9}$$

Subject to the equality constraints

$$y_i = w^T \varphi(x_i) + b + e_i. \tag{10}$$

The resulting dual problem in the case of regression will be:

$$\begin{bmatrix} 0 \\ 1_v \end{bmatrix} \begin{vmatrix} \frac{1}{\Omega + \gamma^{-1} I} \end{vmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \tag{11}$$

with  $\Omega_{ij} = \varphi(x_i)^T \varphi(x_j)$ . The final model will be:

$$y(x) = \sum_{i=1}^N \alpha_i K(x_i, x) + b, \tag{12}$$

with  $\alpha_i = \gamma e_i$ .

### Feed forward neural network

A neural network is a black box that directly learns the internal relations of an unknown system, without guessing functions for describing cause-and-effect relationships. It has been widely used in PV power forecasting [16, 17, 20, 27]. Neural network has the power of a universal approximator [12, 21], i.e., it can realize an arbitrary mapping of one vector space onto another vector space. The main advantage of neural networks is that they are able to use some a priori unknown information hidden in data (but they are not able to extract it). The process of ‘capturing’ the unknown information is called ‘learning’ or the training of neural network. In mathematical formalism to learn means to adjust the weight coefficients in such a way that some conditions are fulfilled [32]. To define a neural network first we introduce the static linear model defined as:

$$g(x, w) = \sum_{i=1}^P w_i f_i(x), \tag{13}$$

where the vector  $w$  is the vector of the parameters of the model, and where the functions  $f(x)$  are non-parameterized functions for the variable  $x$ . The networks of neurons are included in the category of the nonlinear models in their

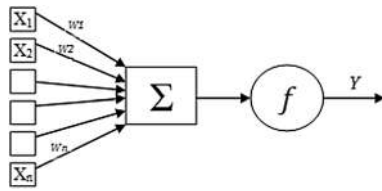


Fig. 2 A simple neuron presentation

parameters. The most current form of static network of neurons is a simple extension of the previous relation:

$$g(x, w) = \sum_{i=1}^P w_i f_i(x, w'), \tag{14}$$

where  $f_i(x, w')$  are parameterized functions, called 'neuron' which is presented in Fig. 2. A neuron is a nonlinear function, parameterized, with limited values. The variables on which the neuron operates are often indicated under the term of inputs of the neuron, and the value of the function under the term of output [19, 22]. The parameters  $w_i$  are indicated under the name of 'weights' or 'synaptic weights', because of the biological inspiration of the neural networks. The output of neuron is a nonlinear function of a combination of the variables  $x_i$  weighted by the parameters  $w_i$ . The parameter  $w_0$  is a constant term called 'bias'. The function  $f$  is called the 'activation function'. The output of a neuron has as equation:

$$y = f \left[ w_0 + \sum_{i=1}^n w_i x_i \right]. \tag{15}$$

A neuron realizes a nonlinear function. The advantage of the neurons lies in the properties which resulting from their association in networks, i.e., of the composition of the nonlinear functions fulfilled by each neuron. There is a large variety of topologies for this kind of networks; nevertheless, the most used topology is the so-called multi-layer perceptron (MLP), whose example is represented in Fig. 3. In this type of neural networks, the first layer is called 'the input layer', and the last layer is called 'output layer'. The layers between are hidden layers; this network carries out  $N_c$  algebraic functions of  $N$  variable of the network.

The MLP is mathematically represented by the expression:

$$g(x, w) = \sum_{i=1}^{N_c} \left[ w_{N_c+1,i} f \left( \sum_{j=1}^n w_{ij} x_j + w_{i0} \right) \right] + w_{N_c+1,0} \tag{16}$$

$$= w_2 f(W_1 x),$$

where  $x$  is the vector of variables (of dimension  $n + 1$ ),  $w_2$  is the vector of weights of the second layer (of  $N_{c+1}$  dimension)

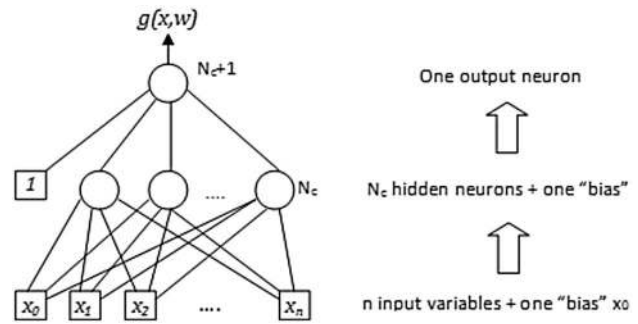


Fig. 3 Multi-layer perceptron neural network architecture

and  $W_1$  is the matrix of weights of the first layer (of dimension  $(N_c + 1, n + 1)$ ). By convention, the parameter  $w_{ij}$  designates the weight between neuron  $j$  towards neuron  $i$ . The model  $g(x, w)$  is a linear function of the parameters of the last layer, and it is a nonlinear function of the parameters of the first layer of connections.

In this paper we used a feed forward neural network (FFNN), which is a type of MLP, with a network of neurons not buckled. In the FFNN, information circulates from the inputs towards the outputs without 'feedback'. It can be represented by an acyclic graph whose nodes are neurons and the edges 'connections' between them. Training the network means to adjust its parameters. There exist two main types of training process: supervised and unsupervised training. In our case, we used a supervised learning process which is based upon the variation of the threshold coefficients  $w_{i0}$  'bias' and weight coefficients  $w_{ij}$  to minimize the sum of the squared error. This objective is accomplished by minimizing of the objective function:

$$E = \sum_o \frac{1}{2} (x_o - \hat{x}_o)^2, \tag{17}$$

where  $x_o$  and  $\hat{x}_o$  are vectors composed of the computed and required values of the output neurons and summation runs over all output neurons  $o$  [12]. The training mode begins with arbitrary values of the weights; the network uses a training algorithm and a set of training data to adjust the weights in the direction that reduces the error, until achieving the optimal set of values. The hope is that the neural network so designed will generalize. A network is said to generalize well when the network learns to correctly associate input patterns to output patterns, even for input–output patterns never used in training stage [12].

## Platform and statistical metrics

### Platform and data

The data used in this research work is collected from a hybrid platform located at the Moroccan School of

**Table 1** PV cell characteristics

Parameters	Description
Manufacturer	Voltec Solar
Max power voltage	30.36 V
Max power	255.36 Wp
Max power current	8.40 A

**Table 2** Inverter characteristics

Parameters	Description
Manufacturer	SMA
Start voltage	140 V
Max voltage	600 V
Max current	12 A
Power	1600 W
Frequency	50 Hz

Engineering Sciences in Casablanca, Morocco (latitude = 33.5415060 and longitude = -7.6735389). The platform is composed from Photovoltaic and wind installations, the photovoltaic installation is a 3.2kw rooftop plant, with an inclination of 40° and facing south, the PV plant consists of 12 modules from the constructor Voltec Solar (six mono-crystalline and six poly-crystalline), each six modules are connected to an SMA SUNNY-BOY inverter. The two SMA SUNNY-BOY inverters are connected to an inverter/charger MultiPlus manufactured by Victron Energy, the inverter/charger MultiPlus allow us to control the charge/discharge of a 3 kw battery park as well as controlling the injection into the grid. The wind installation is composed from a 2.4 kw vertical axis wind turbine (Skystream 3.7) and a 2.5 kw Darrieus wind turbine (apple-wind AW). Also, the platform contains a small meteorological station based on the SMA sunny SensorBox that measures global horizontal irradiance GHI, ambient and module temperature as well as wind speed. The meteorological parameters are recorded every 15 min; all measurements are stored via an SMA WEbBox. A detailed description of the platform is given in [9, 10, 33]. The characteristics of the PV plant are presented in Tables 1 and 2.

The used database consists of 6 months records, from 01 July to 31 December 2014, the records of 5 months, from July to October; contain missing data, while the records of December are intact. To deal with the problem of missing data we use a filling gap procedure. In general there is no definitive guide to replacing missing data in time-series [2]. In the case of photovoltaic time serie, choosing the appropriate method depends on different factors such as length of existing data, availability of reliable meteorological data and climate of the location [24]. The conventional interpolation stills the most used methods [2,

23, 28] because of their simplicity, but they are not always the most efficient. Meany other methods are presented in literature, such as regression, ARIMA, Spline, polynomial fitting [2]. Or more sophisticated method like adaptive interpolation schemes (AISs) [2], temperature based approach (TBA), singular spectrum analysis (SSA), and statistically adjusted solar radiation (SASR) methods [24] or special methods like METSTAT (meteorological/statistical) solar radiation model [18]. Also statistical learning approach can be used in this context as [15] where authors adopt Support Vector Machine (SVM) to obtain a nonlinear weather-type classifier based on humidity and temperature as input variables, the SVM is used to choose the days with the same season type, authors impute a missing value by the average over a specific set of those similar days. In this work we used the conventional interpolation method to fill gaps in solar and PV data, we choose this method because of its simplicity and also because gap length do not exceed 3 h.

### Statistical metrics

To evaluate the model accuracy, we must choose the right performance metrics because modeling is an iterative process, which consists of going back and forth between the output of the model and the desired value. Measuring forecasting error is important to validate the model, so it is necessary to use performance criteria that measure how close outputs (forecasts) are to the eventual outcomes. For this purpose, well-known statistical metrics are used, the mean absolute error (MAE), mean bias error (MBE), mean squared error (MSE), root mean squared error (RMSE) and R-square error, also called coefficient of determination ( $R^2$ ). These metrics are defined as.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (18)$$

$$\text{MBE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i) \quad (19)$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (20)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (21)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (\bar{y} - y_i)^2} \quad (22)$$



## Models development

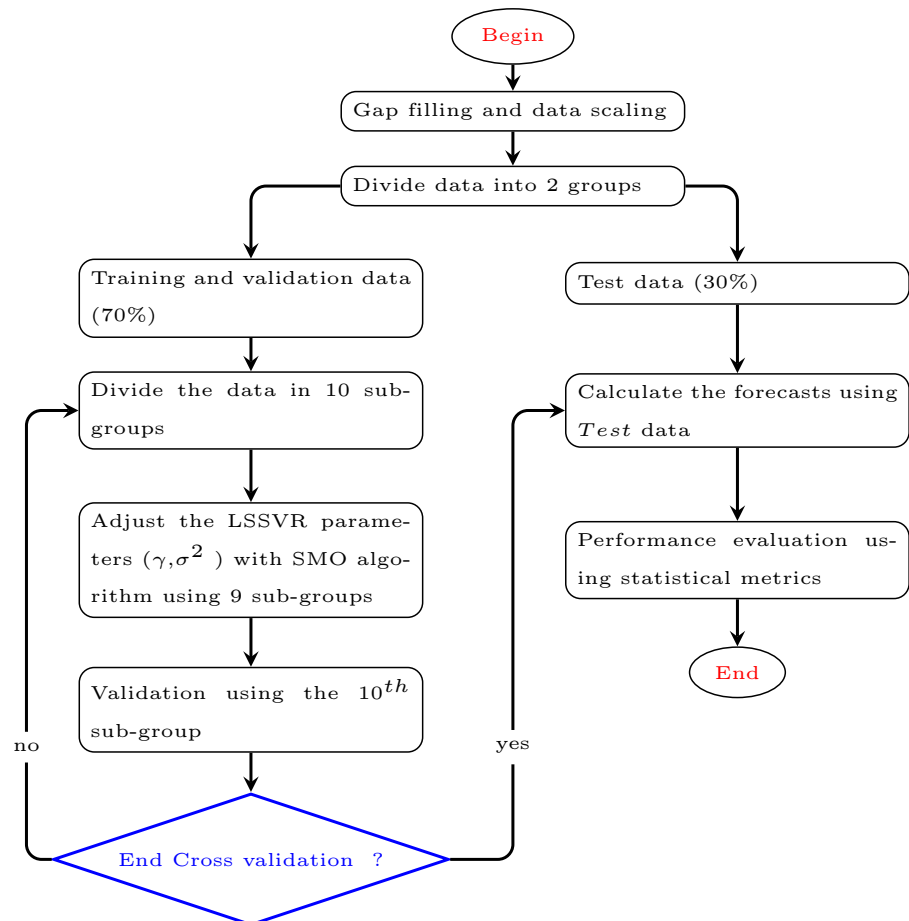
### Training procedures

In this work, we adopted a learning procedure composed from three stages, a pre-processing stage, training and validation stage and finally a test stage. In our case, the pre-processing stage is constituted by a gap filling procedure using linear interpolation method and a scaling procedure which consists of scaling data between zero and one. In the training stage we tried to find the best settings of the FFNN and LSSVR algorithms; for this purpose, we used a general work-flow composed from a training algorithm combined with 10-folds cross validation procedure based on the mean square error (MSE) as a judgment criterion. In the test stage, we use data that have not been used in the training stage to test the model performances. This procedure was adapted for the different algorithms and for all models. So for LSSVR, in the training and validation stage, we used the sequential minimal optimization (SMO) algorithm to find the parameters of radial basis function (RBF), used as kernel function, as well as the parameters  $\gamma$  from (1) and  $\sigma^2$  from (9). The

best model with the best parameters is used to calculate output forecasts. In the test stage the algorithm is fed with new data; the estimated outputs are compared with real outputs, and performance metrics are calculated to evaluate model accuracy. The best model will be the one that will give us the minimum forecasting error, Fig. 4 resume the used procedure. All simulations were done in the Matlab2015b environment; also, we used the standard librarie LS-SVMlab [25]. The obtained results are discussed in Sect. 7.

In the case of FFNN, training the neural network amounts to adjusting the synaptic weights  $w_i$  as well as the number of hidden layers and the number of neurons in each hidden layer without forgetting to choose the right activation function. In order to find the optimal value of the synaptic weights, we used the levenberg–Marquardt (LM) algorithm, which is an improvement of the classical gradient descent algorithm. For the number of hidden layers, we decided to use a single hidden layer FFNN since it is a universal approximator [12], therefore, the FFNN will consist of an input layer, a hidden layer and an output layer. The number of neurons in the input layer depends on the number of parameters used in each model, while the output layer consists of a single neuron with a linear activation function. To find the best number of neurons in the hidden

**Fig. 4** The procedure used in the case of LSSVR training





layer, a sensitivity analysis was performed using the procedure reported in [11].

The final parameter to find is the activation function, the choice of activation function is an important design issue, it is a vital part of neural network providing nonlinear mapping potential and help achieving fast convergence and good generalization performance. To choose the right activation function we assess the performance of the three most used activation functions in FFNN architecture, the radial basis function (RBF), the tangent sigmoid function (Tansig) and the logistic sigmoid function (Logsig), we do so for each developed model.

To carry out this simulation, we consider a number of neurons varying between 1 and 160, and we used the neural network MATLAB toolbox under Matlab2015b environment. The steps we followed are:

1. Choose the maximum number of neurons in the hidden layer “ $p$ ” ( $1 \leq p \leq 160$ ).
2. Initialize (init) the synaptic weights  $w_i$  randomly.
3. Train the FFNN with those settings using the levenberg-Marquardt algorithm and 10-folds cross-validation procedure.
4. Calculate the “ $n_t$ ” forecasts (estimations) obtained after training using validation data.
5. Calculate the Normalized Mean Absolute Error for each forecast  $NMAE_{\%}$ .

$$NMAE_{\%} = \frac{1}{N \cdot C} \sum_{i=1}^N |P_m - P_f| \cdot 100, \tag{23}$$

where  $P_m$  is the measured value of the output power and  $P_f$  the estimated one,  $C$  is the net capacity of the plant and  $N$  is the number of samples.

6. Repeat from step 2 for a chosen number of times (in this study we have repeated the initialization 100 times).
7. Calculate the relative sample mean  $\overline{NMAE_p}$  as an estimator of all possible  $NMAE_{\%}$  values.

$$\overline{NMAE_p} = \frac{1}{n_t} \sum_{i=1}^{n_t} NMAE_{i,p}, \tag{24}$$

where  $NMAE_{i,p}$  is the  $NMAE_{\%}$  calculated for the  $i$ -th trial performed by the FFNN with the  $p$ -th settings.

8. Calculate the sample variance  $S_p^2$  and the sample standard deviation  $S_p$ .

$$S_p^2 = \frac{1}{n_t - 1} \sum_{i=1}^{n_t} (NMAE_{i,p} - \overline{NMAE_p})^2 \tag{25}$$

$$S_p = \sqrt{\sum_{i=1}^{n_t} \frac{(NMAE_{i,p} - \overline{NMAE_p})^2}{n_t - 1}} \tag{26}$$

9. Constructs a confidence intervals (CI) helping the estimation of the unknown population mean  $\mu$  defined as:

$$CI = \overline{NMAE_p} \pm ME, \tag{27}$$

with ME a margin error defined as :

$$ME = t_{\frac{\alpha}{2}} \left( \frac{S_p}{\sqrt{n_t}} \right), \tag{28}$$

with  $t$  is set by the relative  $t$  student distribution according to the degree of freedom equal to  $n_t - 1$ .

10. After choosing the best settings according to  $NMAE_{\%}$  score, we retrain the FFNN and we use test data to evaluate his performances using statistical metrics presented in Sect. 5. Figure 5 resume the procedure used to train the FFNN.

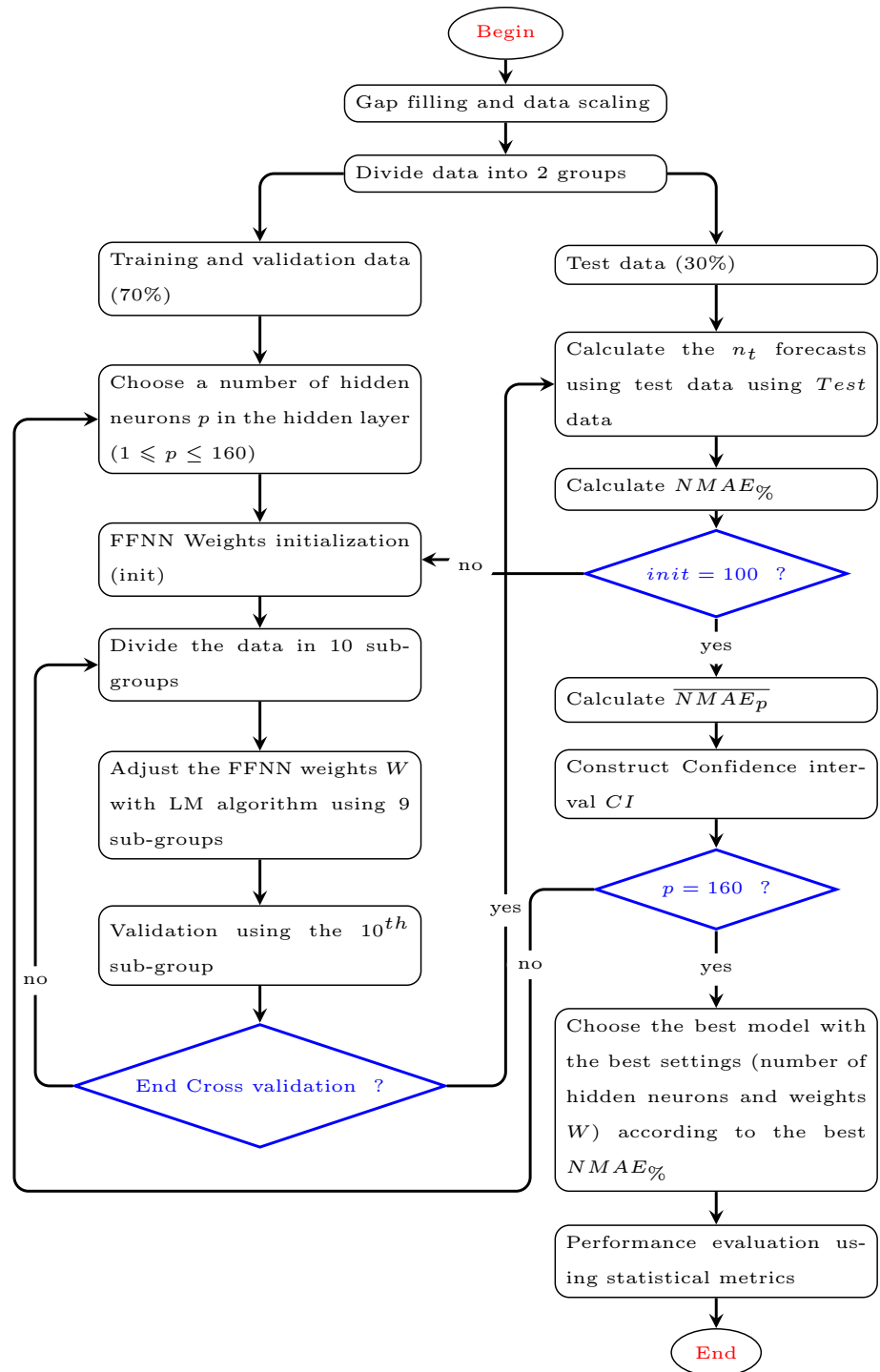
### Inputs selection

There are several statistical models that describe the photovoltaic phenomenon using different weather parameters that influence the conversion. According to the state of the art, the parameters that influence the most PV forecasting are horizontal solar irradiation (Irr) [6, 30], cell temperature (Tc) [6, 30], ambient temperature [6], and aerosol index [16]. In this work, the data used to forecast photovoltaic power are: past PV power generation ( $P$ ), past measured global horizontal solar irradiation (Irr) and past measured photovoltaic modules temperature (Tc), collected via the SMA WebBox. The characteristics of these data is that they are simple to collect locally and do not require a considerable investment. Mathematically, find a one step photovoltaic forecasting model is to find a function in the form:

$$P_{t+1} = f(X), \tag{29}$$

with  $X$  vector of input parameters, it can be a vector of exogenous parameters or a vector of pure auto-regressive parameters. This will give rise to two types of models, a non linear auto-regressive with exogenous inputs model and a pure non linear auto-regressive model. In this study, we focused on the choice of  $X$  and its influence on the accuracy of the model, (the vector  $X$  present the locally (in-situ) measured parameter). Also, we tested different combinations of the three locally measured parameters: the solar irradiation

**Fig. 5** The procedure used in the case of FFNN training



(Irr), the temperature of the cells ( $T_c$ ) as well as the PV power ( $P$ ). To compare the accuracy of the obtained models, statistical metrics were used. Also, to give more meaning to the results, we compared the performances of the models with two other statistical models used as a benchmark: the persistent model as well as a Multivariate Polynomial Regression model (MPR).

## Results analysis and discussion

In this section, we will discuss simulation results that describe the performances of several pure non-linear autoregressive models (NAR) against those of non-linear autoregressive models with exogenous inputs (NARX).

### Nonlinear auto-regressive with exogenous inputs models (NARX)

In time series modeling, a nonlinear autoregressive with exogenous inputs model (NARX) is a nonlinear autoregressive model which has exogenous inputs. This type of models relates the current value of output to both past values of the same output and current and past values of externally inputs that influence the output of interest. Such a model can be formulated as.

$$Y_t = F(Y_{t-1}, Y_{t-2}, \dots, Y_{t-N}; U_{t-1}, U_{t-2}, \dots, U_{t-N}) + \varepsilon_t, \quad (30)$$

where the function  $F$  is some nonlinear function, with  $Y$  is the variable of interest,  $U$  is the exogenous variable and  $\varepsilon_t$  a forecasting error term. In this study we used a combinations of three in-situ measured parameters: the global horizontal solar irradiation (Irr) and the temperature of PV modules (Tc) as exogenous inputs  $U$ , and the PV power ( $P$ ) as variable of interest  $Y$ . The first functions to evaluate are:

$$\text{MOD}_1 : P_{t+1} = F(\text{Irr}_t, \text{Tc}_t) + \varepsilon \quad (31)$$

$$\text{MOD}_2 : P_{t+1} = F(\text{Irr}_t, \text{Tc}_t, P_t) + \varepsilon \quad (32)$$

$$\text{MOD}_3 : P_{t+1} = F(\text{Irr}_t, \text{Tc}_t, P_{t-1}, P_t) + \varepsilon \quad (33)$$

$$\text{MOD}_4 : P_{t+1} = F(\text{Irr}_{t-1}, \text{Irr}_t, \text{Tc}_{t-1}, \text{Tc}_t, P_{t-1}, P_t) + \varepsilon \quad (34)$$

$$\text{MOD}_5 : P_{t+1} = F(\text{Irr}_t, \text{Tc}_t, P_{t-2}, P_{t-1}, P_t) + \varepsilon \quad (35)$$

$$\text{MOD}_6 : P_{t+1} = F(\text{Irr}_t, \text{Tc}_t, P_{t-3}, P_{t-2}, P_{t-1}, P_t) + \varepsilon, \quad (36)$$

We used the FFNN and LSSVR approaches to find the most accurate function  $F$  given in the Eqs. 31–36. The simulation results are presented here after.

#### Least square support vector regression

Using the procedure described in Sect. 6 and the sequential minimal optimization (SMO) algorithm to find the parameters of radial basis function (RBF) as well as the parameters  $\gamma$  and  $\sigma^2$ , the best founded parameters for LSSVR–NARX models are presented in Tables 3 and 4.

The obtained results show interesting characteristics. First, all the models give forecasts with sufficient precision.

**Table 3** LSSVR–NARX parameters

	$\gamma$	$\sigma^2$
MOD <sub>1</sub>	2.58	2.82
MOD <sub>2</sub>	48554.86	8.528
MOD <sub>3</sub>	261.023	11.99
MOD <sub>4</sub>	1051.57	42.43
MOD <sub>5</sub>	67117.98	118.51
MOD <sub>6</sub>	846.78	32.27

**Table 4** LSSVR–NARX models results

Model	MSE	$R^2$
MOD <sub>1</sub>	0.0124	0.8413
MOD <sub>2</sub>	0.0090	0.8846
MOD <sub>3</sub>	0.0087	0.8889
MOD <sub>4</sub>	0.0091	0.8837
MOD <sub>5</sub>	0.0089	0.8865
MOD <sub>6</sub>	0.0082	0.8956

The classification of the models gives us an idea about the influence of output parameters on the precision of the results. After the analysis of the simulation results, we found that the  $\text{Irr}_t$  and  $\text{Tc}_t$  data alone are not sufficient, since the model MOD<sub>1</sub> gives less precise results than the other models. Also, according to the results of the model MOD<sub>2</sub>, we have observed that the combination of the parameters  $\text{Irr}_t$ ,  $\text{Tc}_t$  and  $P_t$  gives better results than those of MOD<sub>1</sub>. The results obtained by the model MOD<sub>3</sub> confirm this observation, since the addition of the parameter  $P_{t-1}$  to the model MOD<sub>2</sub> increases the accuracy of forecasts, whereas the addition of the  $\text{Irr}_{t-1}$  and  $\text{Tc}_{t-1}$  parameters to MOD<sub>2</sub> (which gives the model MOD<sub>4</sub>) reduces the accuracy and gives results that are almost equivalent to those of MOD<sub>1</sub>. As a first remark, we can observe that the use of the  $\text{Irr}_t$ ,  $\text{Tc}_t$  parameters gives good results, but adding historical power data greatly improves model accuracy. To examine this hypothesis, we decided to create other two models in which we will increase the number of auto-regressive inputs by adding to the model MOD<sub>3</sub> the input  $P_{t-2}$  to create MOD<sub>5</sub>, and after that we added the input  $P_{t-3}$  to create MOD<sub>6</sub>. The two new models realize the functions 35 and 36.

According to the simulation results resumed in Table 4, it can be concluded that, effectively, adding the past PV power values  $P_{t-2}$  and  $P_{t-3}$  helps to increase the accuracy of the offline forecasting model. The comparison of the different models leads us to underline the importance of the parameters  $P$  in this kind of model. Figure 6 shows the graphical results of the models MOD<sub>1</sub> to MOD<sub>6</sub>. From the results of the LSSVR–NARX models, the conclusion of this subsection is that for offline short-term PV power forecasting, the most influential parameter is the past PV power ( $P_t, \dots, P_{t-i}$ ), while the parameters  $\text{Irr}_t$ ,  $\text{Tc}_t$  add precision to model forecasts. This is logical, since the parameters  $P_{t-i}$  implicitly contains information concerning the photovoltaic phenomenon, such as the effect of irradiation, temperature and even geographical parameters.

#### Feed-forward neural network

Training the FFNN amounts to adjusting the synaptic weights  $w_i$ , as well as finding the best activation function and the best number of neurons in the hidden layer; for this

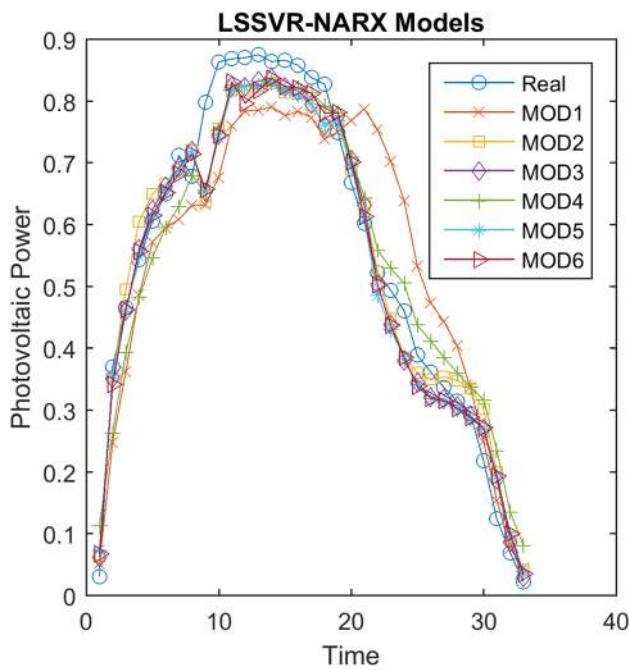


Fig. 6 PV power forecasts obtained by the LSSVR–NARX models

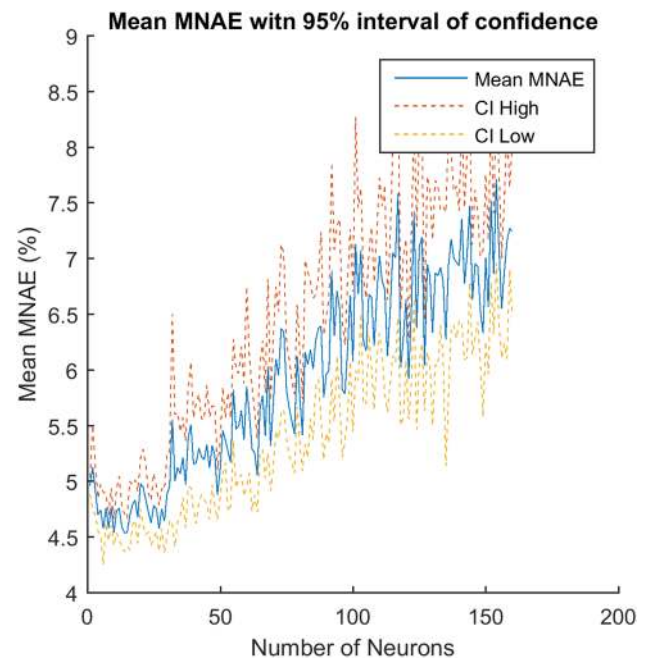


Fig. 7 Mean normalized mean absolute error  $\overline{\text{NMAE}}$  as function of number of neurons in the case of MOD<sub>5</sub>

purpose, we used the ANN Sizing procedure in a way that allowed us to find the best configuration for each model (best activation function and best number of hidden neurons). To do so, first, for one of the models, we choose an activation function and we use the sizing procedure to choose the best number of hidden neurons, after that we change the activation function and we repeat the same procedure to find the best number of hidden neurons and so on until finding the best activation function and best number of hidden neurons for all developed models, of course while using the specific input parameters of each model. After choosing the best number of neurons in the hidden layer, the FFNN was re-trained 100 times in such a way that each training is made with a synaptic weights  $w_i$  initialization different from the others. At the end of sizing procedure the FFNN that gives the minimum  $\text{NMAE}_{\%}$  is considered as the best model. This procedure gives a satisfactory results, so we use it to choose the best configuration of each one of the models. The obtained results are presented in Table 5. Also Fig. 7 shows an example of mean  $\text{NMAE}_{\%}$  evolution as well as its confidence interval (CI) during the training of the model MOD<sub>5</sub>.

After analyzing all FFNN–NARX models results presented in Table 6 and Fig. 8, almost the same remarks as those of the LSSVR–NARX are made; the Irr and Tc inputs alone are not enough to make accurate forecasting. The MOD<sub>1</sub> model gives the worst results with an  $\text{MSE} = 0.0124$ . The addition of the past PV power as inputs considerably improves the accuracy of the models, since in MOD<sub>2</sub> just the addition of the input  $P_{t-1}$ , alone, improved the MSE by almost

Table 5 ANN sizing procedure results for NARX models

Model	Best number of hidden neurons			Best $\text{NMAE}_{\%}$
	RBF	Logsig	Tansig	
MOD <sub>1</sub>	6	8	5	5.73%
MOD <sub>2</sub>	5	7	6	4.72%
MOD <sub>3</sub>	37	6	8	4.54%
MOD <sub>4</sub>	10	10	10	5.65%
MOD <sub>5</sub>	30	8	11	4.61%
MOD <sub>6</sub>	36	12	14	4.48%

Table 6 FFNN–NARX models Results for the three activation functions

Model	$R^2$ (%)		
	RBF	Logsig	Tansig
MOD <sub>1</sub>	84.39	84.66	<b>84.87</b>
MOD <sub>2</sub>	88.57	<b>89.10</b>	88.61
MOD <sub>3</sub>	89.36	86.47	<b>89.86</b>
MOD <sub>4</sub>	00.2	00.42	<b>87.89</b>
MOD <sub>5</sub>	88.2	88.42	<b>88.49</b>
MOD <sub>6</sub>	90.63	89.33	<b>90.73</b>

24.6% ( $\text{MSE} = 0.0089$ ). According to the results of the models MOD<sub>3</sub>, MOD<sub>3</sub> and MOD<sub>6</sub>, we observed that, the more we increase the number of used past PV power as input, the more we increases the forecasts accuracy. But MOD<sub>4</sub> constitutes an aberrant case; normally we tend to think that the more we increase the number of inputs, the more we

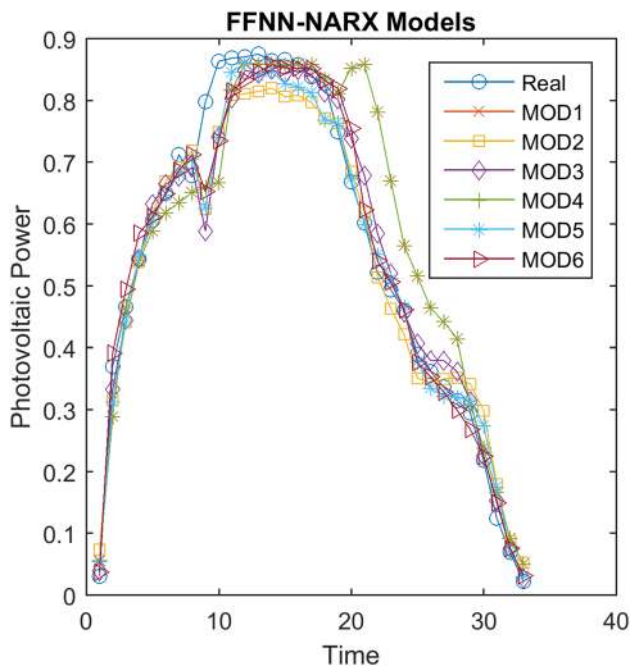


Fig. 8 PV power forecasts obtained by the FFNN–NARX models

add precision to our models. Here the MOD<sub>4</sub> is just the MOD<sub>3</sub> to which has been added the  $Irr_{t-1}$  and  $Tc_{t-1}$  inputs. We expected that MOD<sub>4</sub> will outperform the MOD<sub>3</sub> model, but the simulations show the opposite, with an increase in the forecast error by 20.2% with an MSE = 0.0099. Therefore, the results obtained are consistent with those of the LSSVR–NARX, except that the FFNN–NARX models demonstrates a slight superiority over the LSSVR–NARX models in 75% of the cases. For the FFNN–NARX models, the Tangent sigmoid activation function gives the best results in 83% of the cases, with a coefficient of determination  $R^2 = 90.73\%$  given by MOD<sub>6</sub>.

### Nonlinear autoregressive models

In this section, in order to choose the best offline PV power forecasting model and to analyze the influence of locally collected parameters on model accuracy, we investigate the results obtained from four pure nonlinear auto-regressive models (NAR). In time series modeling, the nonlinear autoregressive model specifies that the output variable non-linearly depends on its own previous values and on a forecasting error term. This type of models can be formulated as:

$$Y_t = F(Y_{t-1}, Y_{t-2}, \dots, Y_{t-N}) + \varepsilon_t. \tag{37}$$

To realize those models, we used only the past PV power values as input. In this case, we try to realize a pure nonlinear auto-regressive model without exogenous parameters. To find the best function  $F$ , we used, as for the NARX

models, the LSSVR and FFNN approaches. To conduct this study, we have developed four models, MOD<sub>7</sub>, MOD<sub>8</sub>, MOD<sub>9</sub> and MOD<sub>10</sub>, which perform the following functions, respectively:

$$MOD_7 : P_{t+1} = f(P_t) + \varepsilon, \tag{38}$$

$$MOD_8 : P_{t+1} = f(P_{t-1}, P_t) + \varepsilon, \tag{39}$$

$$MOD_9 : P_{t+1} = f(P_{t-2}, P_{t-1}, P_t) + \varepsilon, \tag{40}$$

$$MOD_{10} : P_{t+1} = f(P_{t-3}, P_{t-2}, P_{t-1}, P_t) + \varepsilon, \tag{41}$$

here also the  $\varepsilon$  term designates the forecasting error; it has different values for the different models.

### Least square support vector regression

This subsection presents the obtained simulation results using the LSSVR algorithm. After using the sequential minimal optimization (SMO), the best founded  $\gamma$  and  $\sigma^2$  parameters for the LSSVR–NAR models are presented in Table 7. Also, the simulation results are resumed in Table 8.

In the previous subsection, it was concluded that the past PV generation is the most important parameter for short-term PV power forecasting model. The best NARX model, until now, is the model MOD<sub>6</sub>, which realizes the function given by (36). The results resumed in Table 8 and presented in Fig. 9 allow us to make the following remarks. First, all LSSVR–NAR models give good results; also, we observe that the more we add past PV values as inputs, the more we increase the forecasts accuracy. The comparisons between LSSVR–NAR models and LSSVR–NARX models revealed that LSSVR–NAR models give clearly better results. For example, the model MOD<sub>7</sub> that takes into account the current power  $P_t$  as input gives a satisfactory result with an MSE = 0.0092, a result that is better than those of MOD<sub>1</sub>. From the accuracy point

Table 7 LSSVR–NAR best parameters

	$\gamma$	$\sigma^2$
MOD <sub>7</sub>	25664.51	13.758
MOD <sub>8</sub>	80630.94	34.44
MOD <sub>9</sub>	15.95	3.70
MOD <sub>10</sub>	5.8885	3.1766

Table 8 LSSVR–NAR models results

Model	MSE	$R^2$
MOD <sub>7</sub>	0.0092	0.8827
MOD <sub>8</sub>	0.0081	0.8971
MOD <sub>9</sub>	0.0072	0.9074
MOD <sub>10</sub>	0.0066	0.9152

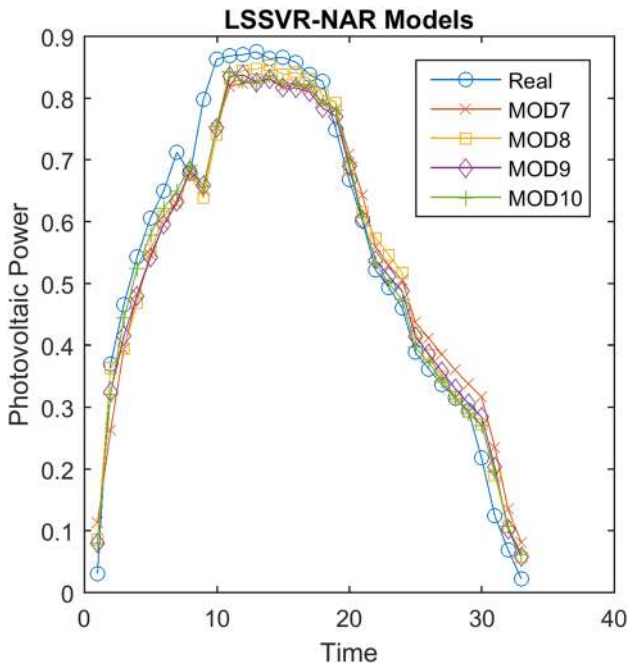


Fig. 9 Forecasting results of LSSVR–NAR models

of view, we notice that all LSSVR–NAR models give better results than the model MOD<sub>6</sub> (for example MOD<sub>8</sub> with only two inputs gives better results than MOD<sub>6</sub> that use six inputs). We also note that the more we add the past PV power terms, the more we increase the accuracy of the forecasts obtained. As a result, the model MOD<sub>10</sub> that realizes the function given by the equation (41) gives the best results; Fig. 10 shows the PV power forecasts obtained by this model. These results are interesting from a practical point of view, since the use of this type of model does not generate an important cost add: most of the used converters can record those parameters and their recuperation are done in an easy and safe way. As a conclusion, the comparison between the different models leads us to underline the importance of past PV power as input parameters in offline short-term PV power forecasting. Table 9 resume all LSSVR based models results.

**Feed-forward neural network**

As in the case of the FFNN–NARX models, the FFNN–NAR models were trained using the same ANN Sizing procedure; the Table 10 summarizes the results obtained by giving the best number of neurons in the hidden layer as well as the corresponding NMAE<sub>%</sub>.

The results obtained by the FFNN–NAR models are resumed in Table 11 and presented in Fig. 11, the results

Table 9 LSSVR all results

	MAE	MBE	MSE	RMSE	R <sup>2</sup>
MOD <sub>1</sub>	0.0865	− 0.0240	0.0124	0.111	0.8413
MOD <sub>2</sub>	0.0647	0.0093	0.0090	0.0948	0.8846
MOD <sub>3</sub>	0.0651	0.0106	0.0087	0.0932	0.8889
MOD <sub>4</sub>	0.0721	0.0012	0.0091	0.0953	0.8837
MOD <sub>5</sub>	0.0666	0.0114	0.0089	0.0943	0.8865
MOD <sub>6</sub>	0.0619	0.0046	0.0082	0.0905	0.8956
MOD <sub>7</sub>	0.0730	0.0005	0.0092	0.0959	0.8827
MOD <sub>8</sub>	0.0675	− 0.0020	0.0081	0.09	0.8971
MOD <sub>9</sub>	0.0622	− 0.0025	0.0072	0.0848	0.9074
MOD <sub>10</sub>	0.0561	− 0.0029	0.0066	0.0812	0.9152

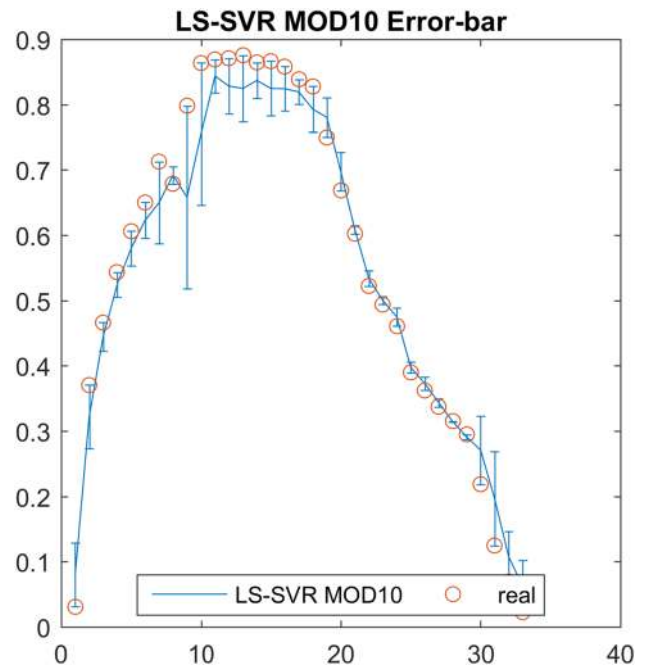


Fig. 10 a PV power forecasts obtained by the model LSSVR–MOD<sub>10</sub> with error bars

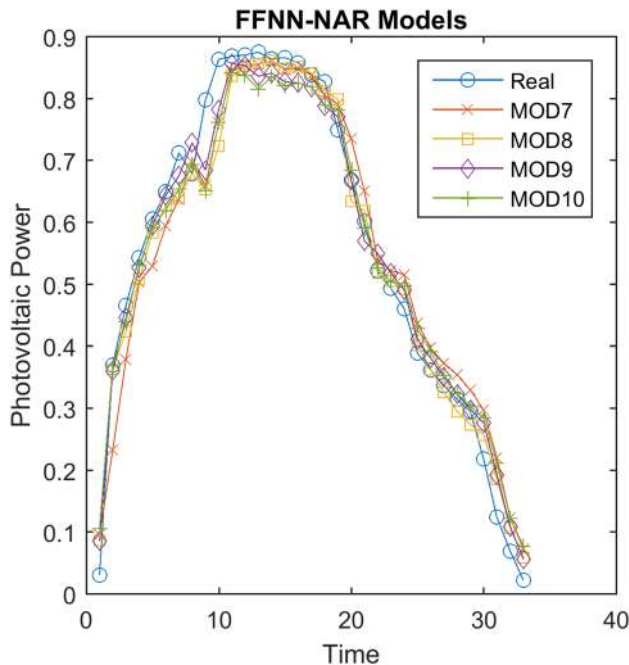
Table 10 ANN sizing procedure results for NAR models

Model	Best number of hidden neurons	Best NMAE <sub>%</sub>
MOD <sub>7</sub>	8	4.80%
MOD <sub>8</sub>	9	4.31%
MOD <sub>9</sub>	6	4.21%
MOD <sub>10</sub>	7	4.08%

join and rectify, again, those announced in the case of LSSVR–NAR case. All NAR models demonstrate superiority compared to NARX models. Again, the FFNN algorithm

**Table 11** FFNN–NAR models Results

Model	$R^2$ (%)		
	RBF	Logsig	Tansig
MOD <sub>7</sub>	88.49	88.30	<b>88.52</b>
MOD <sub>8</sub>	90.15	<b>90.75</b>	90.59
MOD <sub>9</sub>	<b>92.03</b>	91.37	91.26
MOD <sub>10</sub>	<b>91.32</b>	91.11	91.20



**Fig. 11** PV power forecasts obtained by the FFNN–NAR models

**Table 12** FFNN all results

	MAE	MBE	MSE	RMSE	$R^2$
MOD <sub>1</sub>	0.0792	−0.0127	0.0118	0.1088	0.8487
MOD <sub>2</sub>	0.0648	0.0061	0.0089	0.0944	0.8861
MOD <sub>3</sub>	0.0588	0.0006	0.0079	0.0891	0.8986
MOD <sub>4</sub>	0.0721	0.00712	0.0093	0.0964	0.8789
MOD <sub>5</sub>	0.0619	0.0135	0.0090	0.0949	0.8849
MOD <sub>6</sub>	0.0540	0.0041	0.0073	0.0852	0.9073
MOD <sub>7</sub>	0.0686	0.0034	0.0090	0.0948	0.8852
MOD <sub>8</sub>	0.0574	−0.0004	0.0074	0.0858	0.9059
MOD <sub>9</sub>	0.0564	0.0052	0.0068	0.0827	0.9126
MOD <sub>10</sub>	0.0582	−0.0012	0.0069	0.0830	0.9120

results slightly surpasses the results obtained by the LSSVR, but if we take into account the execution time things will change, since for the FFNN the procedure consumes a very important time (almost 17 min) to find the best parameters of an FFNN model, while the LSSVR trained with SMO

**Table 13** MPR and persistent versus LSSVR and FFNN

Metrics	MSE			
	LSSVR	FFNN	MPR	Persistent
MOD <sub>1</sub>	0.0124	0.0118	0.0128	
MOD <sub>2</sub>	0.0090	0.0089	0.0089	
MOD <sub>3</sub>	0.0087	0.0079	0.0090	
MOD <sub>4</sub>	0.0091	0.0093	0.0095	0.0092
MOD <sub>5</sub>	0.0089	0.0090	0.0093	
MOD <sub>6</sub>	0.0082	0.0073	0.0091	
MOD <sub>7</sub>	0.0092	0.0090	0.0091	
MOD <sub>8</sub>	0.0081	0.0074	0.0088	
MOD <sub>9</sub>	0.0072	0.0068	0.0081	
MOD <sub>10</sub>	0.0066	0.0069	0.0080	

consumes only 1 min and 20 s to find the best parameters, which is much lower than the time consumed by the FFNN, Table 12 resume all FFNN based models results. Now for the FFNN–NAR activation function, results demonstrate that radial basis activation function gives the best results in 50% of cases, with a coefficient of determination  $R^2 = 92.03\%$  given by MOD<sub>8</sub>. Also, in all models, the NAR models outperform the NARX models.

**Benchmark models**

In this subsection, we present the results obtained by the two statistical benchmark models: the multivariate polynomial regression model and persistent model. The persistent model is regarded as a naive predictor (today equals tomorrow), and it is the most cost-effective forecasting model which assumes that the conditions will not change; as a result, the PV power at time  $t + 1$  will be equal to those at time  $t$ . In spite of its simplicity, it provides a good benchmark against more sophisticated models and still the most popular reference model in short-term PV power forecasting. On the other hand, the Multivariate Polynomial Regression Model (MPR) is a more sophisticated model. It is an extension of the ordinary polynomial regression, in which the relationship between the input variables  $x$  and the output variable  $y$  is modeled as an  $n^{\text{th}}$  degree polynomial in  $x$ . Equation (42) presents an example of second order multiple polynomial regression:

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \beta_{11}x_1^2 + \beta_{22}x_2^2 + \beta_{12}x_1x_2 + \epsilon. \quad (42)$$

This can again be represented in Matrix form as:

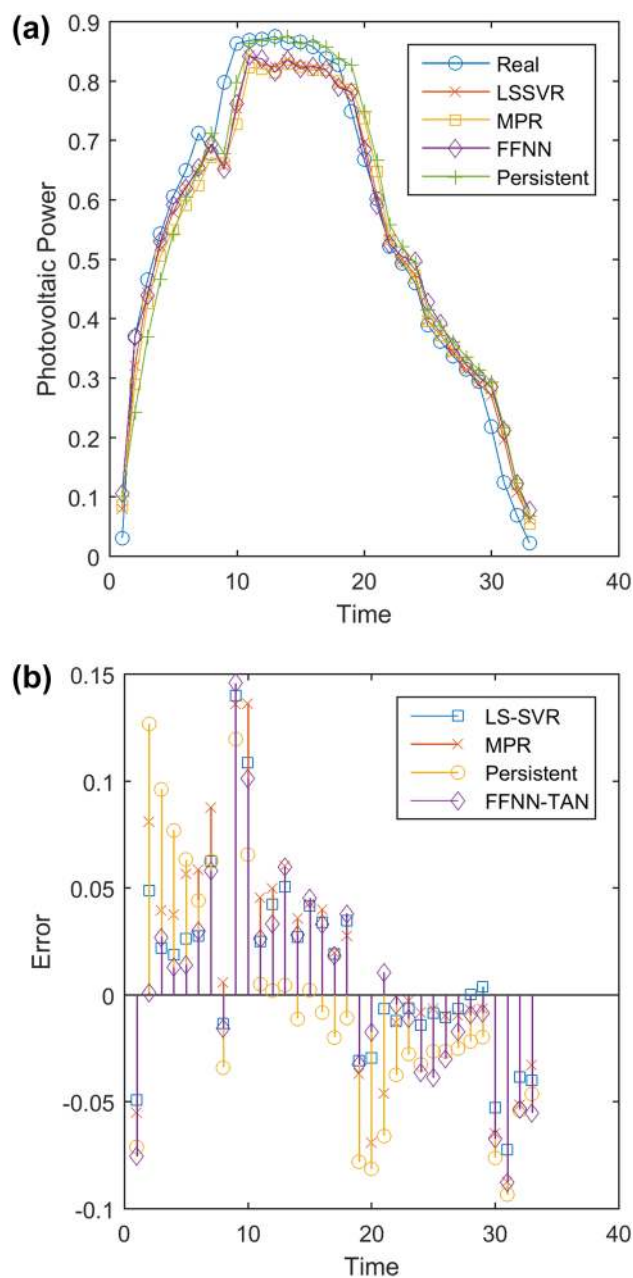
$$Y = \beta X + \epsilon, \quad (43)$$

where  $\beta$  is matrix of weights,  $X$  is matrix of input parameters and  $Y$  is the output. The two models are used as a benchmark to check the performance of the developed models, and subsequently it demonstrated the effectiveness of these models in the short-term PV power forecast. The

simulation results are given in Table 13. It is observed that the MPR model gives results as close as those of the FFNN and LSSVR models; while the persistent model gives an  $MSE = 0.0092$ . We will use the persistent model performance, with an  $MSE = 0.0092$ , to help us identify the models that deserve to be used, since they have to give more interesting results than those of the persistent model as they are more complex. Thus, the models that can be described as interesting are those that outperform the persistent model. According to this constraint, we observe that the LSSVR and FFNN are ranked first because their results outperform those of the persistent model except the model LSSVR-MOD<sub>1</sub> and FFNN-MOD<sub>1</sub>. The MPR model also surpasses the persistent model in 70% of cases, but it remains less efficient than the LSSVR and FFNN. A simple comparison of the results shows the superiority of the two approaches, FFNN and LSSVR, in almost all models. Until now the FFNN is the best, of course with very high calculation time. However, the MPR and the LSSVR consume almost the same time of calculation but with a superiority of the LSSVR. Another observation is that the persistent model gives excellent results in the stable weather conditions (clear day) as can be seen in Fig. 12, whereas nonlinear models such as LSSVR and FFNN are more efficient under unstable weather conditions, Fig. 13, which present the results obtained in the case of a cloudy day, demonstrate this observation. So according to this comparison we can conclude that the models proposed for the PV forecast show a superiority over the benchmark models, especially the NAR models. Those results are very interesting knowing the importance of short-term forecasts in the integration of photovoltaic sources in the energy mix and to guarantee the grid stability.

## Conclusion

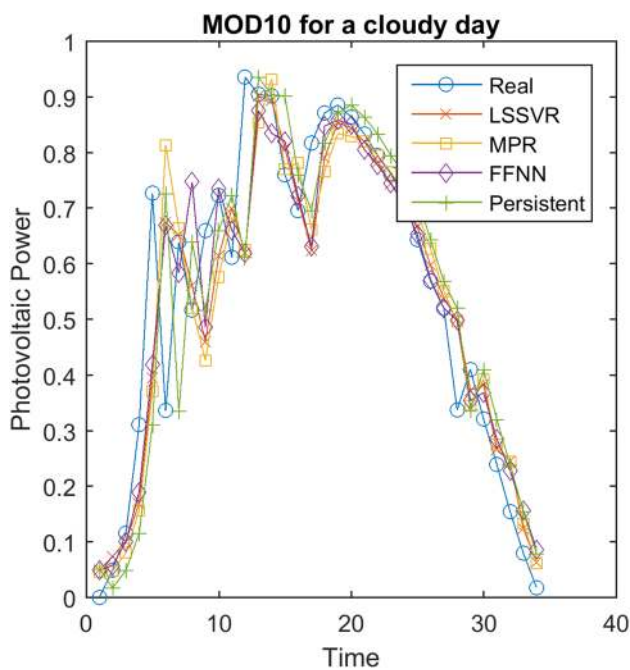
In the present contribution offline models have been proposed that allows us to forecast the short-term PV power using only information collected from local monitoring system, i.e., without the need of weather forecasts. The offline models are interesting for grid operators as well as for individuals because the majority of existing PV power forecasting models uses NWP, the issue is that access to NWP information is not given to everyone, especially for isolated installations. In the aim of studying the behavior of each model and each algorithm, we combined the simplicity of time series models (AR and ARX) and the non-linearity of statistical learning models (FFNN and LSSVR), also we used a different combination of collected data in the aim to analyze the influence of different locally collected data on forecasts accuracy.



**Fig. 12** **a** PV power forecasts obtained by all best models and **b** forecasting error

During the simulations, it was observed that the FFNN gives different results each time the simulation is repeated, which is due to the problem of initialization, whereas the LSSVR gives a unique solution, which constitutes the optimal one (As long as there is sufficient training database). To improve the FFNN results and to avoid over-fitting and local-minima problems during FFNN learning a sizing procedure have been proposed, by using the ANN sizing procedure we observe that the performance of FFNN models have improved. The choice of the right activation function





**Fig. 13** PV power forecasts obtained by all best models in a cloudy day

is an important design issue, according to simulation results we found that tangent sigmoid function (Tansig) gives the best results in 70% of the cases, and even if logistic sigmoid (Logsig) and radial basis (RBF) functions outperforms the Tansig in 30% of the cases, the Tansig still the best activation function according the global performances. Prior to training the proposed models, the data used was subjected to a pre-processing procedure which consists of filling the gaps using a linear interpolation method and scaling data between zero and one.

The comparison between all FFNN based models and LSSVR based models indicate that the FFNN algorithm slightly outperforms the LSSVR algorithm. But if we take into account the execution time things will change, since for the FFNN the procedure consumes a very important time (almost 17 min) to find the best parameters of an FFNN model, while the LSSVR trained with SMO consumes only 1 min and 20 s to find the best parameters, which is much lower than time consumed by the FFNN.

To test the performance of the proposed models, the results obtained are compared with those of the persistent model as well as a multivariate polynomial regression model (MPR) as benchmark. Comparison demonstrate the superiority of FFNN and LSSVR against MPR and persistent models, simulation results indicates that the FFNN–MOD<sub>9</sub> with RBF activation function and LSSVR–MOD<sub>10</sub> give the best results and outperform all other models with an MSE = 0.0065 and MSE = 0.0069, respectively. Also results

of the persistent technique and statistical techniques (MPR, FFNN and LSSVR) offer evidence regarding the advantage of using non-linear forecasting models over a trivial forecast. To not underestimate the persistent model, we underline that the persistent model gives excellent results in the stable weather conditions (clear day), whereas nonlinear models (in addition to stable conditions) are more efficient under unstable weather conditions.

The comparison of our results with other works will not be fair, since the data used and the weather conditions change from one country to another and from one installation to another. what we can do is to compare our main findings with another work, to do so we choose [1] as main reference. According to simulation results it was observed that the NAR models give better results than NARX models. These results seem contradictory with those of [1] in which ARX models outperform the AR models, the difference is that authors in [1] uses NWP of global solar irradiation as input for NARX model, in our work we try to avoid the use of NWP parameters and using only locally collected data. Moreover, the present contribution demonstrates that the use of past photovoltaic power production as input improves the accuracy of forecasting models, and the use of past generated power data only is enough to have an accurate and acceptable short-term PV power forecasts. This result confirm the findings of [1] where authors indicates that solar power is most important input for making forecasts of horizon shorter than 2 h.

We must report that, the length of data used in this work does not allow the proposed models to adapt to all types of weather conditions, this will cause a decrease in performance of our models especially in the case of overcasting days. Also, to increase forecast horizon the use of only local collected data is not sufficient, in this case the use of weather forecasts will be an obligation. Those issues will be resolved in future works.

## Compliance with ethical standards

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Bacher, P., Madsen, H., Nielsen, H.A.: Online short-term solar power forecasting. *Solar Energy* **83**(10), 1772–1783 (2009)
- Brooks, M.J., von Backström, T.W., van Dyk, E.E.: Performance characteristics of a perforated shadow band in the presence of cloud. *Solar Energy* **139**, 533–546 (2016)
- Chen, C., Duan, S., Cai, T., Liu, B.: Online 24-h solar power forecasting based on weather type classification using artificial neural network. *Solar Energy* **85**(11), 2856–2870 (2011)
- Chupong, C., Plangklang, B.: Forecasting power output of pv grid connected system in thailand without using solar radiation measurement. *Energy Procedia* **9**, 230–237 (2011)
- Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995)
- De Giorgi, M., Congedo, P., Malvoni, M.: Photovoltaic power forecasting using statistical methods: impact of weather data. *IET Sci. Meas. Technol.* **8**(3), 90–97 (2014)
- De Leone, R., Pietrini, M., Giovannelli, A.: Photovoltaic energy production forecast using support vector regression. *Neural Comput. Appl.* **26**(8), 1955–1962 (2015)
- Ding, M., Wang, L., Bi, R.: An ann-based approach for forecasting the power output of photovoltaic system. *Procedia Environ. Sci.* **11**, 1308–1315 (2011)
- Fentis, A., Bahatti, L., Mestari, M., Chouri, B.: Short-term solar power forecasting using support vector regression and feed-forward nn. In: *New Circuits and Systems Conference (NEWCAS)*. In: 2017 15th IEEE International, 405–408. IEEE, (2017)
- Fentis, A., Bahatti, L., Mestari, M., Tabaa, M., Jarrou, A., Chouri, B.: Short-term pv power forecasting using support vector regression and local monitoring data. In: *Renewable and Sustainable Energy Conference (IRSEC)*. In: 2016 International, 1092–1097. IEEE, (2016)
- Grimaccia, F., Leva, S., Mussetta, M., Ogliari, E.: Ann sizing procedure for the day-ahead output power forecast of a pv plant. *Appl. Sci.* **7**(6), 622 (2017)
- Hornik, K.: Approximation capabilities of multilayer feedforward networks. *Neural Netw.* **4**(2), 251–257 (1991)
- Huang, C.-M., Chen, S.-J., Yang, S.-P., Kuo, C.-J.: One-day-ahead hourly forecasting for photovoltaic power generation using an intelligent method with weather-based forecasting models. *IET Gener. Trans. Distrib.* **9**(14), 1874–1882 (2015)
- Huang, Y., Lu, J., Liu, C., Xu, X., Wang, W., Zhou, X.: Comparative study of power forecasting methods for pv stations. In: *Power System Technology (POWERCON)*, 2010 International Conference on, 1–6. IEEE, (2010)
- Layanun, V., Suksamornsorn, S., Songsiri, J.: Missing-data imputation for solar irradiance forecasting in thailand. In: *Society of Instrument and Control Engineers of Japan (SICE)*, 2017 56th Annual Conference of the, 1234–1239. IEEE, (2017)
- Liu, J., Fang, W., Zhang, X., Yang, C.: An improved photovoltaic power forecasting model with the assistance of aerosol index data. *IEEE Trans. Sustain. Energy* **6**(2), 434–442 (2015)
- Mandal, P., Madhira, S.T.S., Meng, J., Pineda, R.L., et al.: Forecasting power output of solar photovoltaic system using wavelet transform and artificial intelligence techniques. *Procedia Comput. Sci.* **12**, 332–337 (2012)
- Marion, W., George, R.: Calculation of solar radiation using a methodology with worldwide potential. *Solar Energy* **71**(4), 275–283 (2001)
- McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5**(4), 115–133 (1943)
- Mellit, A., Pavan, A.M., Lughfi, V.: Short-term forecasting of power production in a large-scale photovoltaic plant. *Solar Energy* **105**, 401–413 (2014)
- Mestari, M., Benzirar, M., Saber, N., Khouil, M.: Solving non-linear equality constrained multiobjective optimization problems using neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(10), 2500–2520 (2015)
- Minski, M.L., Papert, S.A.: *Perceptrons: an introduction to computational geometry*. MIT Press, Cambridge (1969)
- Moreno-Tejera, S., Silva-Pérez, M., Lillo-Bravo, I., Ramírez-Santigosa, L.: Solar resource assessment in seville, spain. Statistical characterisation of solar radiation at different time resolutions. *Solar Energy* **132**, 430–441 (2016)
- Ogunsola, O.T., Song, L.: Restoration of long-term missing gaps in solar radiation. *Energy Build.* **82**, 580–591 (2014)
- Pelckmans, K., Suykens, J.A., Van Gestel, T., De Brabanter, J., Lukas, L., Hamers, B., De Moor, B., Vandewalle, J.: *Ls-svmlab toolbox users guide*. *Pattern Recogn. Lett.* **24**(2003), 659–675 (2003)
- Pelland, S., Remund, J., Kleissl, J., Oozeki, T., De Brabandere, K.: Photovoltaic and solar forecasting: state of the art. *IEA PVPS Task* **14**, 1–36 (2013)
- Ramsami, P., Oree, V.: A hybrid method for forecasting the energy output of photovoltaic systems. *Energy Convers Manag.* **95**, 406–413 (2015)
- Schwandt, M., Chhatbar, K., Meyer, R., Fross, K., Mitra, I., Vashistha, R., Giridhar, G., Gomathinayagam, S., Kumar, A.: Development and test of gap filling procedures for solar radiation data of the indian srma measurement network. *Energy Procedia* **57**, 1100–1109 (2014)
- Shi, J., Lee, W.-J., Liu, Y., Yang, Y., Wang, P.: Forecasting power output of photovoltaic systems based on weather classification and support vector machines. *IEEE Trans. Ind. Appl.* **48**(3), 1064–1069 (2012)
- Silva Fonseca, J.G., Oozeki, T., Takashima, T., Koshimizu, G., Uchida, Y., Ogimoto, K.: Use of support vector regression and numerically predicted cloudiness to forecast power output of a photovoltaic power plant in kitakyushu, japan. *Prog. Photovolt. Res. Appl.* **20**(7), 874–882 (2012)
- Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. *Neural Process. Lett.* **9**(3), 293–300 (1999)
- Svozil, D., Kvasnicka, V., Pospichal, J.: Introduction to multi-layer feed-forward neural networks. *Chemom. Intell. Lab. Syst.* **39**(1), 43–62 (1997)
- Tabaa, M., Dandache, A., Alami, K.: Hybrid renewable energy installation for research and innovation: Case of casablanca city in morocco. In: *New Circuits and Systems Conference (NEWCAS)*, 2017 15th IEEE International, 389–392. IEEE, (2017)
- Tao, C., Shanxu, D., Changsong, C.: Forecasting power output for grid-connected photovoltaic power system without using solar radiation measurement. In: *Power Electronics for Distributed Generation Systems (PEDG)*, 2010 2nd IEEE International Symposium on, 773–777. IEEE, (2010)
- Yang, H.-T., Huang, C.-M., Huang, Y.-C., Pai, Y.-S.: A weather-based hybrid method for 1-day ahead hourly forecasting of pv power output. *IEEE Trans. Sustain. Energy* **5**(3), 917–926 (2014)
- Zhang, Y., Beaudin, M., Taheri, R., Zareipour, H., Wood, D.: Day-ahead power output forecasting for small-scale solar photovoltaic electricity generators. *IEEE Trans. Smart Grid* **6**(5), 2253–2262 (2015)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

