Toward Better PV Panel’s Output Power Prediction; a Module Based on Nonlinear Autoregressive Neural Network with Exogenous Inputs

Emad Natsheh and Sufyan Samara *

Department of Computer Engineering, An-Najah National University, Omar Ibn Al-Khattab St., PO Box 7, Nablus, Palestine
* Correspondence: sufyan_sa@najah.edu

Received: 27 July 2019; Accepted: 2 September 2019; Published: 4 September 2019

Abstract: Much work has been carried out for modeling the output power of photovoltaic panels. Using artificial neural networks (ANNs), one could efficiently model the output power of heterogeneous photovoltaic (HPV) panels. However, due to the existing different types of artificial neural network implementations, it has become hard to choose the best approach to use for a specific application. This raises the need for studies that develop models using the different neural networks types and compare the efficiency of these different types for that specific application. In this work, two neural network types, namely, the nonlinear autoregressive network with exogenous inputs (NARX) and the deep feed-forward (DFF) neural network, have been developed and compared for modeling the maximum output power of HPV panels. Both neural networks have four exogenous inputs and two outputs. Matlab/Simulink is used in evaluating the proposed two models under a variety of atmospheric conditions. A comprehensive evaluation, including a Diebold-Mariano (DM) test, is applied to verify the ability of the proposed networks. Moreover, the work further investigates the two developed neural networks using their actual implementation on a low-cost microcontroller. Both neural networks have performed very well; however, the NARX model performance is much better compared with DFF. Using the NARX network, a prediction of PV output power could be obtained, with half the execution time required to obtain the same prediction with the DFF neural network, and with accuracy of ±0.18 W.

Keywords: renewable energy; modeling; heterogeneous photovoltaic panel; artificial neural network; low-cost microcontroller; modeling; time series prediction

1. Introduction

In the last decade, solar energy has been of great interest to investors, governments, energy operators, and international organizations. This is because of its multiple environmental and economic benefits. To generate electricity from solar energy, one can either use solar thermal or photovoltaic (PV) panels. However, the PV panel’s systems have gained global acceptance and are now playing an important part in providing sustainable clean energy [1]. Consequently, over the past few years, there have been tremendous growths in the PV panel’s usage all over the globe. This can be clearly induced from Figure 1, in which the x-axis characterizes the year, and where the y-axis characterizes the amount of power produced from PV panels in Gigawatts (GW). As illustrated in Figure 1, the global capacity of power produced from PV panel’s installations increased in two years (2006–2008) from 6.7 GW to 16.0 GW and increase continuously to reach 100.5 GW in the next four years (2008–2012). In total, the installed PV panels’ capacity has increased all over the globe from 6.7 GW to 404 GW in the last 12 years. This continuous growth is anticipated to progress at a similar or even at a higher rate in the
forthcoming years. This can be related to the technological benefit of the PV panels. In summary; from the year 2006 to 2017, we see an exponential increase in the amount of power produced from the PV panels [2]. This exponential growth of the PV panel’s power production, all over the globe, is proportional to the exponential increase in the number of PV panels used to produce such power.

![Worldwide growth of photovoltaics](image)

*Figure 1.* The worldwide growth of total photovoltaic (PV) systems installed from 2000 to 2017.

However, the power generated from a PV panel depends on a number of uncertain meteorological aspects such as atmospheric temperature, PV panel temperature, and solar irradiance. Thus, the accurate prediction of PV panel’s output power production is highly difficult. The unforeseen PV panel’s power output creates different negative effects on the quality, the stability, and the reliability of an electrical grid system. These negative effects hinder any possibility for effective distribution scheduling and/or management of the generated power [3,4]. On the other hand, precise forecasting of the PV panel’s output power production can improve the reliability of the system. This could be carried out to keep an effective schedule that would maintain the power quality and stabilization with a grid with secure operation [5].

As a result, a precise prediction of PV power production is of great importance and imposes a rich research area with many challenges. This can be clearly induced by the amount of research that has been carried out to forecast solar power and PV panel’s output power through different perspectives. Various methods have been implemented, including numerical weather prediction [2,6–8], statistical [2,9–11], and artificial neural networks (ANNs) [2,12–16]. Several studies have illustrated that ANNs are more appropriate when compared to other techniques; especially when a complicated and non-linear connection occurs between the data without any prior presumption [17–19].

In addition, because of the excellent outcomes achieved in actual applications [20–23], ANNs are chosen to be of the most popular models. One great benefit of the ANN model is its ability to model non-linear data associations. The authors in [24,25] employed a neural network to energy consumption approximation. Deb et al. [26] discussed a procedure to predict the energy consumption of day time load of cooling institutional buildings. Paoli et al. [27] developed a multi-layer perceptron for solar radiation forecasting. Mandal et al. [16] presented a PV system power output one-hour-ahead forecasting by combining radial basis and wavelet transform function neural network. Saberian et al. [28] used
both the feed-forward neural network (FFNN) and the general regression neural network (GRNN) to forecast the power output of photovoltaic cells. They concluded that FFNN outperformed GRNN.

All previously cited studies [12–28] modeled the output power of photovoltaic panels using static neural networks techniques (e.g., FFNN). In actual life applications, there are many non-linear systems, such as the PV panels, whose behavior varies according to their contemporary state. Therefore, in the past few years, researchers have developed new dynamic forecasting techniques that improve the accuracy and stability of model prediction [29]. Among these techniques are the dynamic recurrent neural network (RNN) and the nonlinear autoregressive network with exogenous inputs (NARX) [30,31]. Wunsch et al. [32] applied the NARX to obtain groundwater level forecasts for several wells in southwest Germany. In [33], convolutional neural network combined with a recurrent neural network (RNN) is used for automatic lip-reading recognition. Ahmad et al. [34] used an autoregressive RNN with exogenous inputs and weather variables to provide a day-ahead forecast of hourly solar irradiance in New Zealand. In [35], RNN combined with a wavelet neural network (forming a diagonal recurrent wavelet neural network) is presented to perform day-ahead solar irradiance forecasting. As opposed to recurrent networks, NARX have a lower number of feedback connections, which comes only from the output neuron rather than from hidden states. Hence, NARX has lower implementation complexity compared with RNN, especially when it comes to hardware implementation. Moreover, there are not many studies showing the effect of using such techniques in the PV output power forecasting field.

Nevertheless, the field of PV panel’s output power prediction is an open field and much work is required toward obtaining better and more accurate techniques. In the field of PV panel’s and solar energy, the area of ANN has yet many benefits to offer. Due to many ANN models and variations, one needs to explore the different kinds of ANN and their customization to reveal the benefits ANN can offer to accommodate different criteria. Among these criteria, we can point out, finding small size ANN to fit a low-cost embedded system, providing for more accuracy, and the support for heterogeneity in terms of different geolocation also manufacturing characteristics.

Hence, this study focuses on the output power modeling of heterogeneous PV panels using the NARX neural network. Moreover, a neural network module based on deep feed-forward (static network) is developed for comparison and investigation purposes of the proposed network. The deep feed-forward (DFF) has gained a lot of attention in recent years due to flexibility and accuracy in different application. In the end, the work presents a comparison between the running time of both networks implementations on a low-cost microcontroller. This implementation allows the proposed work to be used as a submodule in real-time PV panel’s monitoring system.

2. Artificial Neural Networks Topology

In this work, two promising neural network architectures have been developed for modeling the output power of HPV. The first model is the DFF neural network, and the second is the NARX neural network. The two models developed to tackle different manufactured PV panels. The following subsections describe the topology and the training algorithm used to implement these models.

2.1. DFF Neural Network

A DFF network, also called a Multilayer Perceptron (MLP), is a neural network structure that is used in many applications. It is applied to both discrete and discontinuous problems [36,37].

MLPs consist of neurons, which are structured into layers. The input layer and the output layer are the first layer and the last layer, respectively. The hidden layers are the ones in between the first layer and the last layer.

There is a specific function for each layer in the MLP. The input signals are accepted through the input layer and redistributed to all the hidden layer neurons. The hidden layer stimulus patterns are propagated to the output layer, which produces the entire network output pattern. The hidden layer
neurons detect the input patterns’ hidden features. The output layer uses these features to calculate the final output. Figure 2 shows the topology of the developed DFF neural network.

As illustrated in Figure 2, the developed DFF model is composed of four layers. The input layer accepts the panel’s temperature, the solar radiation, and the panel’s Short Circuit Current (SCC) and Open Circuit Voltage (OCV) at 1 kW/m² and 25 °C. Each layer of the two hidden layers is composed of five neurons. Each of these neurons uses an activation function, which is the hyperbolic tangent sigmoid transfer function. The output layer consists of two nodes, which calculate the optimum operating voltage and the optimum current that correspond to the inputs at the input layer. Nodes of the output layer use a linear activation function.

Thus, the actual output of the neurons in the computational layers can be represented as in Equations (1)–(3):

$$H_{f_j} = \frac{2}{1 + e^{-2 \sum_{i=1}^{4} x_i w_{ij} - \theta_j}} - 1$$

$$H_{s_k} = \frac{2}{1 + e^{-2 \sum_{l=1}^{5} H_{f_l} w_{kl} - \theta_k}} - 1$$

$$Y_{o_l} = \sum_{k=1}^{5} H_{s_k} w_{kl} - \theta_l$$

where $H_{f_j}$ is the $j$th node output in the first hidden layer, $\theta_j$ is the threshold applied to the neuron $j$, $w_{ij}$ is attained through the neural network training stage and it represents the weight between $i$th node in the input layer and the $j$th node in the first hidden layer, $x_i$ is the value of input $i$, $H_{s_k}$ is the $k$th node output in the second hidden layer, $\theta_k$ is the threshold applied to the neuron $k$, $w_{kl}$ is obtained during the training phase of the neural network and it represents the weight between the $j$th node in the first hidden layer and $k$th node in the second hidden layer, $Y_{o_l}$ is the $l$th node output in the output layer, $\theta_l$ is the threshold applied to the neuron $l$, and $w_{lj}$ is obtained during the training phase of the neural network and it represents the weight between the $k$th node in the second hidden layer and $l$th node in the output layer.

2.2. NARX Neural Network

NARX is a neural network with recurrent dynamic behavior [38–40] that has been effectively used for time series problems estimation. The main difference between MLP and NARX is that NARX...
allows a weighted feedback connection between layers of neurons. This allows the NARX models to consider lagged (past period) values of variables, which make them suitable for time series analysis. Alternatively, although in the literature, there are many approaches for time series analysis, such as the autoregressive integrated moving average approach and the autoregressive moving average approach; these approaches cannot cope with nonlinear problems [41,42]. NARX, however, can efficiently be used for modelling time series with non-linear behavior.

The NARX neural network can be constructed into one of two architectures [31], the parallel architecture (namely close-loop architecture), and the series-parallel architecture (namely, open-loop architecture), see Figure 3.

This study uses the open-loop architecture. This is because of the pure feed-forward network architecture and the past accurate values availability of the time series. The NARX network behavior, in compliance with the time series forecast, can be mathematically modeled by the equation shown in Equation (4) [31]:

\[
y(t) = f(y(t-1), y(t-2), \ldots y(t-n_y), x(t-1), x(t-2), \ldots, x(t-n_x))
\]  

(4)

This equation defines in what way a NARX network is used to forecast a data series \(y(t)\) value. The equation makes use of the previous values of the \(y\) series and another external series \(x\). The \(f()\) unknown function is a mapping function, and it is the purpose of the training phase of the network to find an approximation of this mapping function. This is done through an optimization process of the neuron bias and the network weights.

As mentioned above, the NARX is a model with non-linear behavior that has the ability to approximate forthcoming values of time series using its former calculated values and some exterior data. In the presented work, NARX is used to estimate the values of the optimum operating voltage and current. As inputs, the NARX uses four inputs with exterior data at time \(t-1\). In addition, the NARX uses two time-series inputs that represent its previous outputs of the estimated optimum operating voltage and current at time \(t-1\). Figure 4 shows the developed NARX network discussed in this work.

As illustrated in Figure 4, the first layer has six nodes of inputs. These are the panel temperature, the solar radiation, the panel OCV, the panel SCC, and the two output values previously attained from the time series. In the middle (hidden) layer there are five nodes with hyperbolic tangent sigmoid transfer activation function. The last layer consists of two nodes, which calculate the two outputs of the network. These are the optimum operating current and voltage. A linear activation function is used in the nodes of the last layer.
2.3. Training Algorithm

Many training methods have been developed for neural networks \[43,44\]; these include the Gradient Descent training approach, the Resilient Back-propagation training approach, the Levenberg-Marquardt (LM) approach, and the Gauss-Newton training approach. The LM training approach has been selected for training both of the developed topologies (see Figures 3 and 4). The LM training method is less sensitive to local converges, and therefore, it provides a better learning training approach. Moreover, it provides a balanced tradeoff between the stability and training speed \[43\].

The LM training approach was developed to estimate the derivative of the second-order without the particular Hessian matrix computation requirement. Instead, it only requires the Jacobian matrix, also the gradient vector.

Equation (5) is used to calculate the loss function gradient vector when the performance function resembles the mean squared error (MSE). Using the Jacobian matrix we can estimate the Hessian matrix as shown in Equation (6) \[43\]:

\[
\nabla f = J^T e \quad \text{(5)}
\]

\[
H \approx J^T J + \mu I \quad \text{(6)}
\]

In Equations (5) and (6), \(e\) denotes the error vector, \(I\) represents the identity matrix, the combination coefficient \(\mu\) is always greater than zero, and the Jacobian matrix \(J\) can be defined as in Equation (7):

\[
J_{i,j} = \frac{\partial e_i}{\partial w_j}, \quad \text{where } j = 1, \ldots, m \text{ and } i = 1, \ldots, n \quad \text{(7)}
\]

Equations (5) and (6) can be combined to form the update rule \(\Delta w\) shown in Equation (8) \[43\]:

\[
\Delta w = (J^T J + \mu I)^{-1} J^T e \quad \text{(8)}
\]

The state diagram for the training algorithm is shown in Figure 5. The training process begins by calculating the loss (MSE), the Hessian approximation, and the gradient. The training objective is to minimize the loss (MSE) as much as possible. This is done by tuning the parameter (\(\mu\)) of the learning rate as follows:

- \(\mu\) is multiplied by 10 when the current epoch MSE exceeds the previous value.
• $\mu$ is multiplied by 0.1 when the MSE is equal or less to the previous one.

$\mu$ is multiplied by 10 when the current epoch MSE exceeds the previous value.

2.4. Dataset Collection

Meteorological data consisting of solar irradiance and temperature were gathered in real-time from An-Najah Energy Research Center (ERC) [45]. The equipment used to collect the dataset consists of a temperature sensor to measure the PV panel body temperature (namely, the sensor WE710 with $\pm0.25^\circ C$ precision), and a high accuracy solar radiation sensor (namely, the sensor WE300 with $\pm1\%$ precision).

Despite the precision of the used sensors, the data captured are sometimes unreliable, noisy, or incomplete. This is primarily due to a sensor error percentage or noise in connections. Hence, to overcome the noise in the obtained data, a moving average filter [46] was used, see Figure 6. The meteorological data were collected on an hourly based manner in the period from February 2018 to July 2018. Anomalies in data due to a broken or unfunctional device/sensor were eliminated.

The targeted output voltage ($V_r$) and current ($I_r$) are collected by running a MATLAB code that analyses the validated PV model output characteristics [47–50]. The dynamic behaviour of the proposed PV model has been validated previously by the author for different types of PV modules under variety of solar radiation and temperature conditions. High agreement between the outcomes of the model and the experimental data was achieved. Results and validation of the module can be found in the author previous work published in [50].

Three different photovoltaic panels were used to obtain data; these are the Astronergy-CHSM6610P panel, the Sharp’s-NUS0E3E panel, and Lorentz mono-crystalline panel. Standard testing conditions used when recording the PV panel’s manufacturer specifications. Table 1 shows the used PV panel’s manufacturer specifications.
For the proposed neural networks to be modeled, the collected data were separated into three parts: testing (25%), training (60%), and validation (15%). To have the same range of values, a normalization process [49] is done on all data to be in the range $(-1$ to $+1)$ using Equation (9):

$$A = (B - B_{\text{min}}) \left[ \frac{A_{\text{max}} - A_{\text{min}}}{B_{\text{max}} - B_{\text{min}}} \right] + A_{\text{min}}$$  \hspace{1cm} (9)$$

where the normalized value is denoted by $A$, and $B$ denotes the non-normalized value. When the result is obtained, it becomes normal after using Equation (10). Table 2 shows typical examples of the used datasets.

$$B = (A - A_{\text{min}}) \left[ \frac{B_{\text{max}} - B_{\text{min}}}{A_{\text{max}} - A_{\text{min}}} \right] + B_{\text{min}}$$  \hspace{1cm} (10)
Table 2. Typical random examples of the training sets.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>NDP</th>
<th>DP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>−0.0315</td>
<td>485</td>
</tr>
<tr>
<td>2nd</td>
<td>-0.1859</td>
<td>410</td>
</tr>
<tr>
<td>3rd</td>
<td>0.1256</td>
<td>565</td>
</tr>
<tr>
<td>4th</td>
<td>−0.3467</td>
<td>330</td>
</tr>
<tr>
<td>5th</td>
<td>0.01507</td>
<td>510</td>
</tr>
<tr>
<td>6th</td>
<td>0.21608</td>
<td>610</td>
</tr>
</tbody>
</table>

G (W/m²)  Tc (°C)  OCV a  SCC a  Vr (V)  Ir (A)
−0.33 20 0.9326 8.27 0.7510 0.1131
36.88 1.0 30.44 30.44 30.44 30.44
1.0 20 0.9326 8.27 0.7510 0.1131

3. Results and Discussion

3.1. Neural Network Structure

Several trials have been performed to determine the neural network configuration. Table 3 presents some important parameters for the selected configurations. The decisions have been made based on two imperatives attributes which are: the number of nodes in hidden layers, and the choice of each layer activation function.

Table 3. Attributes of the proposed neural network model’s structures. MSE: mean squared error.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>NARX Structure</th>
<th>DFF Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of hidden layers</td>
<td>Number of hidden layers</td>
<td>2</td>
</tr>
<tr>
<td>Normalization interval of dataset</td>
<td>[-1,1]</td>
<td>[-1,1]</td>
</tr>
<tr>
<td>Tapped delay line (TDL)</td>
<td>1 b</td>
<td>TDL</td>
</tr>
<tr>
<td>Error</td>
<td>MSE</td>
<td>MSE</td>
</tr>
<tr>
<td>Training approach</td>
<td>Levenberg-Marquardt</td>
<td>Levenberg-Marquardt</td>
</tr>
</tbody>
</table>

Choosing each layer activation function is proven to be an important issue. Moreover, previous studies showed that multilayer network with linear output neurons and sigmoid hidden neurons are arbitrarily well-fitted to map multi-dimensional problems [36,49]. Hence, in this study, the activation function of the hyperbolic tangent is implemented for the hidden layers, also the linear activation function is implemented for the output layer. There is an advantage, especially when using the Levenberg-Marquardt training approach, to use the hyperbolic tangent function. The fact that this function is derivable, gives the advantage for easier neural network weights tuning.

The last important parameter, which must be studied to complete the structures of the neural networks, is the number of nodes in each hidden layer. Table 4 compares the best MSE results to various hidden layer’s number of neurons for both NARX and DFF network.

Table 4. Best-obtained results vs. the hidden layer’s number of neurons.

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of Nodes/Neurons</th>
<th>MSE Train</th>
<th>MSE Test</th>
<th>Number of Nodes/Neurons</th>
<th>MSE Train</th>
<th>MSE Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>NARX Model</td>
<td>2</td>
<td>0.0365</td>
<td>0.0465</td>
<td>2 × 2</td>
<td>1.5618</td>
<td>2.9642</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.0313</td>
<td>0.0315</td>
<td>3 × 3</td>
<td>0.7764</td>
<td>0.8642</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.00964</td>
<td>0.010065</td>
<td>4 × 4</td>
<td>0.5831</td>
<td>0.6945</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.007589</td>
<td>0.006415</td>
<td>4 × 5</td>
<td>0.4816</td>
<td>0.4951</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.010832</td>
<td>0.009921</td>
<td>5 × 5</td>
<td>0.4167</td>
<td>0.4283</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>0.009412</td>
<td>0.007052</td>
<td>5 × 6</td>
<td>0.4576</td>
<td>0.4891</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0.008641</td>
<td>0.007731</td>
<td>6 × 5</td>
<td>0.4317</td>
<td>0.5172</td>
</tr>
</tbody>
</table>

b One input delay and one output delay.
As shown in Table 4 the best-obtained structure of the DFF model is when having five neurons in each hidden layer. The MSE for both training and testing with this structure are 0.4167 and 0.4283, respectively. However, using the NARX model the results were much better. Result shows that when having five neurons in one hidden layer, the MSE for both testing and training are 0.006415 and 0.007589, respectively.

3.2. Evaluation of the Proposed Network Models

Before starting the simulation, the dataset was divided into three parts: testing (1680 cases), validation (840 cases), and training (5880 cases). The training and validation sets are used during the training stage. The training dataset part is used to find the neural network bias and weights. The overfitting was minimized using the validation set. Figure 7, shows the results throughout the training and validation stages.

As shown in Figure 7, the measured error with the NARX model is much lower compared with the DFF model. Moreover, Figure 8 illustrates the function of error autocorrelation (for the NARX network) which describes the time correlation of the prediction errors. It can be seen from Figure 8 that...
there were significant prediction errors correlation, the only autocorrelation value outside the interval occurs at lag 0.

![Autocorrelation of Error 1](image)

**Figure 8.** NARX error autocorrelation (a) zoom out view (b) zoom-in view.

After the training stage is done, the testing dataset part is used to evaluate the model and test its performance. Figures 9–12, shows real time comparison between the real data of the networks models and the simulated results for two different manufacturing PV panels.

The results of these figures are obtained from the average hourly solar radiation and temperature distributions during April 2018 (see Figure 6). The comparison of NARX and DFF results demonstrate that NARX is more efficiently track the real time data compared with DFF.

To further verify the effectiveness and significance of the proposed models, we employed the Diebold-Mariano (DM) statistic [51]. The DM compares the forecast accuracy of two models. The MSE is adopted as the loss function to build the loss differential. The DM test results are exhibited in the following Table 5.

<table>
<thead>
<tr>
<th>Average Value</th>
<th>First Test-Sample Window</th>
<th>Second Test-Sample Window</th>
<th>Third Test-Sample Window</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(406)</td>
<td>(912)</td>
<td>(1680)</td>
</tr>
<tr>
<td>DM-test</td>
<td>1.6873</td>
<td>1.3783</td>
<td>-6.5539</td>
</tr>
<tr>
<td>p-value</td>
<td>0.0915</td>
<td>0.1681</td>
<td>&lt;0.00001</td>
</tr>
</tbody>
</table>

According to the DM test results presented in Table 5, the first two sample windows indicates that there is no significant different between the two forecasting models (p-value > 0.05); however, in the third test-sample window, the forecasting performance of the NARX model is better than the DFF model (p-value < 0.05).
Figure 9. Real time comparison using the Astronergy-CHSM6610P PV panel (a) between the actual power and simulated DFF network (b) DFF voltage and current errors.
Figure 10. Real time comparison using Sharp’s-NUS0E3E PV panel (a) between the actual power and simulated DFF network (b) DFF voltage and current errors.
Figure 11. Real time comparison using the Astronergy-CHSM6610P PV panel (a) between the actual power and simulated NARX network (b) NARX voltage and current errors.

Figure 11. Real time comparison using the Astronergy-CHSM6610P PV panel (a) between the actual power and simulated NARX network (b) NARX voltage and current errors.
Figure 12. Real time comparison using Sharp’s-NUS03E PV panel (a) between the actual power and simulated NARX network (b) NARX voltage and current errors.

Table 5. Results of the Diebold-Mariano (DM) test.

<table>
<thead>
<tr>
<th>Average Value</th>
<th>First Test-Sample Window (406)</th>
<th>Second Test-Sample Window (912)</th>
<th>Third Test-Sample Window (1680)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM-test</td>
<td>1.6873</td>
<td>1.3783</td>
<td>~6.5539</td>
</tr>
<tr>
<td>p-value</td>
<td>0.0915</td>
<td>0.1681</td>
<td>&lt;0.00001</td>
</tr>
</tbody>
</table>

*Significance level is 5%.*
3.3. Implementing the Two Networks on A Low-Cost Microcontroller

The aim of this work is to find an accurate forecasting neural network that is suitable for using as sub-module in PV monitoring, PV fault detection, or any PV system that requires PV output power prediction. To further investigate the two developed networks, both trained ANNs are loaded on the same hardware, namely the ATmega2560 8-bit low cost microcontroller. Both networks were implemented using the topologies shown in Figures 2 and 4. Equations (1)–(3) were used for implementing the neurons with weights \( w_{ij}, w_{jk}, \) and \( w_{kl} \) obtained from the training phase of the neural networks. The same data set is used as input on both implementations and each neural network output is recorded along with the execution time, see Table 6. As shown in Table 6, there is a difference between values obtained from MATLAB simulation and hardware implementation. This is because the hardware implementation on low cost microcontroller requires some rounding in values and requires some tradeoff between memory and value precision. However, we notice that the NARX network showed better performance in term of execution time, also better accuracy over the DFF network compared to the results obtained in simulation. Using NARX network an output power value is obtained with a higher accuracy of \( \pm 0.18 \) W and with half the execution time compared to the DFF network. Using the DFF network the same output power value calculation requires double the execution time of the NARX network and with lower accuracy of \( \pm 0.59 \) W.

### Table 6. Execution time and accuracy results from DFF and NARX networks hardware implementations on low cost microcontroller.

<table>
<thead>
<tr>
<th>#</th>
<th>G</th>
<th>Tc</th>
<th>OCV</th>
<th>SCC</th>
<th>DFF Simulation Output Power</th>
<th>NARX Simulation Output Power</th>
<th>DFF Hardware Implementation Output Power</th>
<th>NARX Hardware Implementation Output Power</th>
<th>Execution Time in Ms</th>
<th>Execution Time in Ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>135</td>
<td>5</td>
<td>36.88</td>
<td>8.27</td>
<td>27.45394878</td>
<td>28.74701286</td>
<td>29.3508</td>
<td>13.312</td>
<td>28.557</td>
<td>7.584</td>
</tr>
<tr>
<td>3</td>
<td>195</td>
<td>10</td>
<td>36.88</td>
<td>8.27</td>
<td>41.66625045</td>
<td>42.9396561</td>
<td>43.0836</td>
<td>13.264</td>
<td>42.9358</td>
<td>7.68</td>
</tr>
<tr>
<td>4</td>
<td>315</td>
<td>15</td>
<td>36.88</td>
<td>8.27</td>
<td>69.91714602</td>
<td>71.1672504</td>
<td>69.7662</td>
<td>13.168</td>
<td>71.185</td>
<td>7.712</td>
</tr>
<tr>
<td>5</td>
<td>430</td>
<td>20</td>
<td>36.88</td>
<td>8.27</td>
<td>95.0832446</td>
<td>96.29310936</td>
<td>94.5768</td>
<td>13.232</td>
<td>96.5568</td>
<td>7.616</td>
</tr>
<tr>
<td>8</td>
<td>195</td>
<td>10</td>
<td>30</td>
<td>8.37</td>
<td>30.05208908</td>
<td>30.98741522</td>
<td>29.1856</td>
<td>13.616</td>
<td>30.8238</td>
<td>7.648</td>
</tr>
<tr>
<td>9</td>
<td>315</td>
<td>15</td>
<td>30</td>
<td>8.37</td>
<td>52.92964254</td>
<td>53.88320892</td>
<td>52.269</td>
<td>13.568</td>
<td>53.664</td>
<td>7.648</td>
</tr>
<tr>
<td>10</td>
<td>430</td>
<td>20</td>
<td>30</td>
<td>8.37</td>
<td>74.6095383</td>
<td>75.55625954</td>
<td>74.2356</td>
<td>13.44</td>
<td>75.831</td>
<td>7.44</td>
</tr>
<tr>
<td>11</td>
<td>100</td>
<td>5</td>
<td>21</td>
<td>5.4</td>
<td>2.53894176</td>
<td>2.79904892</td>
<td>2.3488</td>
<td>12.64</td>
<td>2.6946</td>
<td>7.184</td>
</tr>
<tr>
<td>12</td>
<td>135</td>
<td>5</td>
<td>21</td>
<td>5.4</td>
<td>4.6246709</td>
<td>4.97279076</td>
<td>4.68</td>
<td>12.704</td>
<td>5.1604</td>
<td>7.168</td>
</tr>
<tr>
<td>15</td>
<td>430</td>
<td>20</td>
<td>21</td>
<td>5.4</td>
<td>29.38812465</td>
<td>29.82476539</td>
<td>29.232</td>
<td>12.592</td>
<td>30.082</td>
<td>7.408</td>
</tr>
</tbody>
</table>

4. Conclusions

This paper discusses a novel and a more accurate heterogeneous photovoltaic (HPV) output power modeling topology using nonlinear autoregressive network with exogenous inputs (NARX). The work compares the proposed NARX neural network structure with the commonly used deep feed-forward neural network (DFF) structure. Both structures have been used to model the maximum output power of HPV panels.

The training phases of the neural networks are performed periodically. Several simulations are carried out with different evaluation criteria. Both of neural networks have decent modeling performance; nevertheless, NARX has outperformed DFF with mean square error of 0.007589.

Both models are further investigated by implementing them on real hardware for performance and execution time evaluation. The nonlinear autoregressive network with exogenous inputs outperform the deep feed-forward neural network with both accuracy and performance. Using the NARX neural
network, a prediction of PV output power could be obtained with half the execution time required to obtain the same prediction with the deep feed-forward neural network and with an accuracy of ±0.18 W.

On the other hand, future research will need to be carried out on using this network model in PV fault detection. The NARX model has the ability to capture the nonlinear association of patterns between predictors, such as irradiance and temperature, to determine accurate MPP.

**Author Contributions:** Conceptualization, E.N. and S.S.; methodology, E.N. and S.S.; validation, E.N. and S.S.; formal analysis, E.N. and S.S.; investigation, E.N. and S.S.; resources, E.N. and S.S.; data curation, E.N. and S.S.; writing—original draft preparation, E.N. and S.S.; writing—review and editing, S.S.; visualization, E.N. and S.S.

**Acknowledgments:** We acknowledge the support of An Najah National University.

**Funding:** This research received no external funding.

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

**Abbreviations**
The following abbreviations are used in this manuscript:

- ANN: Artificial Neural Networks
- DFF: Deep Feed-Forward
- DM: Diebold-Mariano
- DP: Data Pattern
- FFNN: Feed-Forward Neural Network
- G: Solar radiation
- GRNN: General Regression Neural Network
- HPV: Heterogeneous Photovoltaic
- LM: Levenberg-Marquardt
- MLP: Multilayer Perceptron
- MSE: Mean Square Error
- NARX network: Nonlinear Autoregressive network with exogenous inputs
- NDP: Normalized Data Pattern
- OCV: Open Circuit Voltage
- PV: Photovoltaic
- p-Value: Probability Value
- RNN: Recurrent Neural Network
- SCC: Short Circuit Current
- Tc: Panel temperature
- TDL: Tapped Delay Line

**References**


32. Wunscha, A.; Liescha, T.; Brodab, S. Forecasting groundwater levels using nonlinear autoregressive networks with exogenous input (narsx). *J. Hydrol.* 2018, 567, 743–758. [CrossRef]
33. Lu, Y.; Li, H. Automatic Lip-Reading System Based on Deep Convolutional Neural Network and Attention-Based Long Short-Term Memory. *Appl. Sci.* 2019, 9, 1599. [CrossRef]


© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).