

## Article

# Short-Term Solar Irradiance Prediction Based on Adaptive Extreme Learning Machine and Weather Data

Ahmad Alzahrani 

Department of Electrical Engineering, College of Engineering, Najran University, Najran 11001, Saudi Arabia; asalzahrani@nu.edu.sa

**Abstract:** Concerns over fossil fuels and depletable energy sources have motivated renewable energy sources utilization, such as solar photovoltaic (PV) power. Utilities have started penetrating the existing primary grid with renewable energy sources. However, penetrating the grid with photovoltaic energy sources degrades the stability of the whole system because photovoltaic power depends on solar irradiance, which is highly intermittent. This paper proposes a prediction method for non-stationary solar irradiance. The proposed method uses an adaptive extreme learning machine. The extreme learning machine method uses approximated sigmoid and hyper-tangent functions to ensure faster computational time and more straightforward microcontroller implementation. The proposed method is analyzed using the hourly weather data from a specific site at Najran University. The data are preprocessed, trained, tested, and validated. Several evaluation metrics, such as the root mean square error, mean square error, and mean absolute error, are used to evaluate and compare the proposed method with other recently introduced approaches. The results show that the proposed method can be used to predict solar irradiance with high accuracy, as the mean square error is 0.1727. The proposed approach is implemented using a solar irradiance sensor made of a PV cell, a temperature sensor, and a low-cost microcontroller.

**Keywords:** solar; irradiance; prediction; machine learning; adaptive ELM; Najran; GHI



**Citation:** Alzahrani, A. Short-Term Solar Irradiance Prediction Based on Adaptive Extreme Learning Machine and Weather Data. *Sensors* **2022**, *22*, 8218. <https://doi.org/10.3390/s22218218>

Academic Editors: Kyandoghere Kyamakya, Fadi Al-Machot, Ahmad Haj Mosa, Jean Chamberlain Chedjou, Zhong Li and Antoine Bagula

Received: 28 September 2022

Accepted: 20 October 2022

Published: 27 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the author. 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 (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Renewable energy sources have the potential to satisfy the future energy demand because they are sustainable, clean, and cost effective. A study suggests that the path toward 100% of energy integration into the main grid is feasible in 2050 [1]. Several countries have already built large-scale renewable energy projects to gain economic advantages and reduce carbon footprint per capita [2]. The 300 MW Skaka solar PV project is launched in the first phase of the National Renewable Energy Program (NREP) in Saudi Arabia [3]. The project provides power to 44,000 houses and reduces carbon emissions by 606,000 tons/yr [4]. The second phase of the NREP includes seven solar PV projects with a total capacity of about 2.98 GW, and the fourth phase of NREP includes four solar PV projects with a full capacity of 1.2 GW. The Saudi Arabian 2030 vision aims to increase and diversify energy sources.

Solar PV energy is suitable for Saudi Arabia because of the high solar radiation and long average and peak hours [5]. However, residential solar PV in Saudi Arabia does not contribute significantly to the total energy supply as in Germany. The Electricity and Cogeneration Regulatory Authority (ECRA) introduced regulations for consumers who are eligible to install grid-tied PV systems [6], where all bill calculations and a net metering scheme are illustrated [7]. In a grid-tied PV system, the PV panels feed the electrical loads when solar irradiance is available. During nights and when no solar irradiance is available, the electric loads are fed by the main grid [8]. The connection of the PV panel to the main grid impacts the voltage stability of the system and leads to disturbances [9]. Therefore, several solutions have been introduced to mitigate the impact of penetrating the main

grid [10], such as a meter data management system [11], smart transformers [12], smart inverters [13], an outage management system [14], and solar irradiance prediction [15]. Both grid-tied and off-grid PV systems can be improved if information about the future values of solar irradiance is accurately predicted [16]. That is, the charge controller will have enough information and enough time to decide to charge or discharge a battery or to run critical loads [17].

Several solar irradiance prediction methods have been introduced in the literature [18]. These methods are categorized based on either the prediction time horizon or the input data type [19]. So, based on the prediction time horizon, the prediction types are short term (e.g., five minutes ahead) or long term (e.g., one week ahead) [20]. The input of the predictors is historical data [21], meteorological data [22], or sky images [23]. Prediction solar irradiance models are classified into physical, statistical, and empirical models. In physical models, the GHI is modeled as a function of the tilt angle, DNI, or the DHI [24]. However, the error in simple physical model estimation is significant during cloudy days [25]. The empirical prediction models can be found in the literature and are often used to estimate monthly solar irradiance. The most widely used approach is a sunshine-based predictive model [26].

Statistical models are the most widely used for short-term solar irradiance [27]. Common statistical methods are autoregressive moving average (ARMA) [28], Hidden Markov Models (HMM) [29], and autoregressive integrated moving average (ARIMA) [30]. In these approaches, the Yule–Walker method is used to identify the coefficients [31], and before applying this approach, the time series should be tested for stationarity [32], which may be a disadvantage. Support vector regression (SVR) is the most extensively used approach to predict solar PV power [33]. The SVR algorithm is a supervised learning system. SVM is an effective technique for classification and prediction. SVRs are designed to construct decision boundaries using the notion of decision planes [34]. Although it has been employed in prediction, SVM may be incapable of extracting the time series' long-term correlation or the extremely short-term components. Prediction based on an artificial neural network (ANN) [35] is used to relate the temperature and previous GHI values to predict the next value. Other studies use the convolutional neural network (CNN) by analyzing the spatiotemporal correlation of solar irradiance [36]. Most neural network-based approaches need training and backpropagation to update the weight and decrease the mean square error.

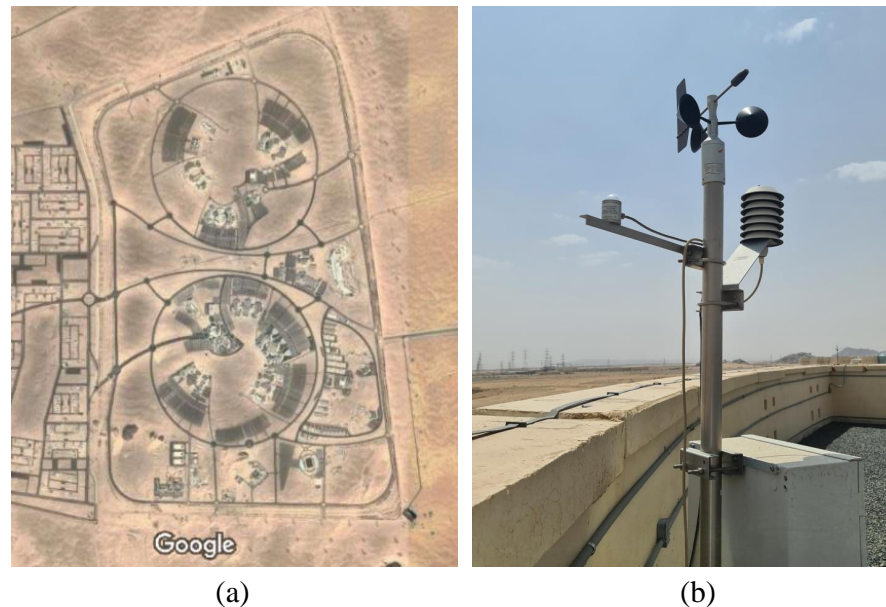
Therefore, new approaches, such as extreme learning machine, have been introduced to reduce the computation time and provide accurate results. However, the extreme learning machine does not update the output weight in each iteration. Therefore, an online update to output weight is needed. This paper presents an adaptive ELM approach implementation to predict solar irradiance and how to update its weight online. The proposed method differs from the original ELM in weight update and a buffer for the input values. The main contributions of this paper are as follows:

- It proposes two different approaches to solar irradiance prediction. These two approaches can predict irradiance with high accuracy and relatively less computational time.
- It presents a effective method for online adaptation of the output weight of the ELM method, which has less computational time.
- The developed models are trained, tested, and validated using local data with a 15 min/sample resolution.
- Implementation and testing of the adaptive ELM approach are carried out on a low-cost microcontroller.

The rest of this paper is structured and organized as follows: Section 2 presents site location and weather information, Section 3 presents some details about the theoretical analysis of the proposed approaches, and Section 4 presents the prediction process and the possibility of implementing low-cost hardware, and Section 5 presents the results of the proposed approach. Finally, the conclusion and future work are presented in Section 6.

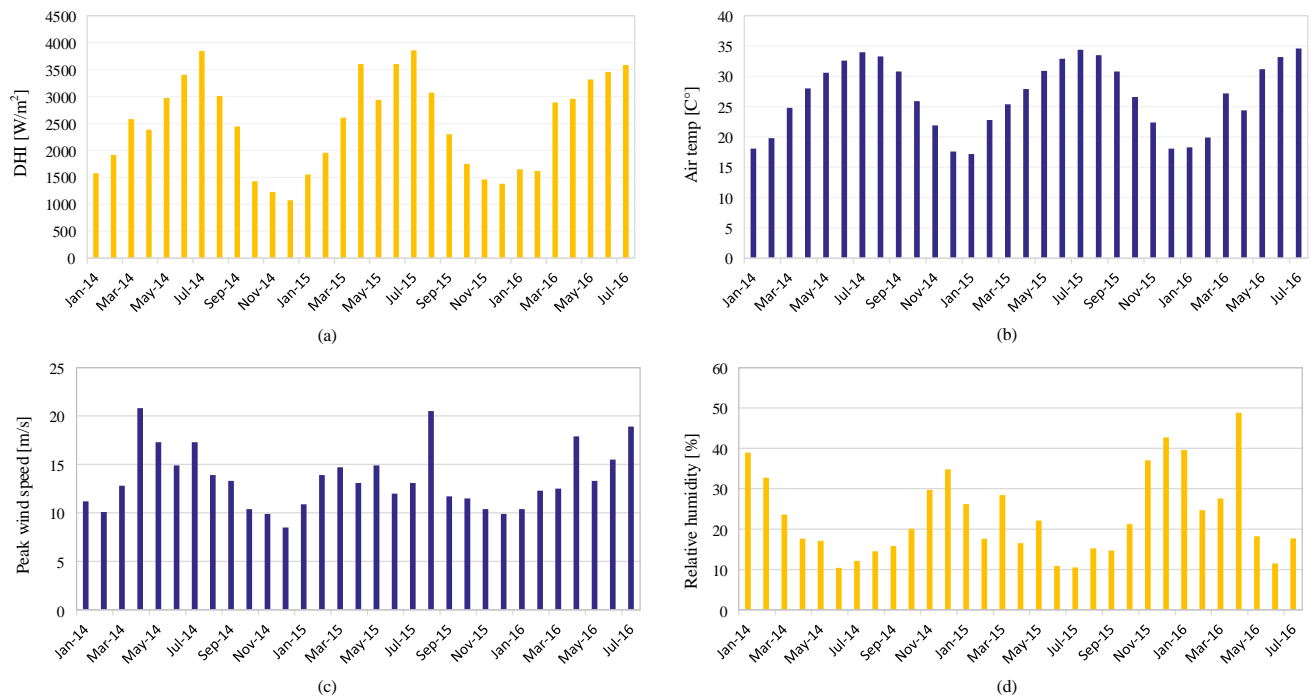
## 2. Site Location and Data Acquisition

The site location used in this paper is Najran University in the southern part of the Kingdom of Saudi Arabia. The University is located at latitude 17.63228 North, longitude 44.53735 East, and elevation of 1290 m above sea level. Figure 1 shows the location of the Najran University and the weather station. The city's climate and university surroundings are considered continental and relatively dry. Several data sources were found on the government website or weather websites. The data in this paper were obtained from Open data by K.A. Care, which contains monthly samples from 2014 to mid-2016. The hourly weather data can also be obtained from the National Solar Radiation Database (NSRDB) [37].

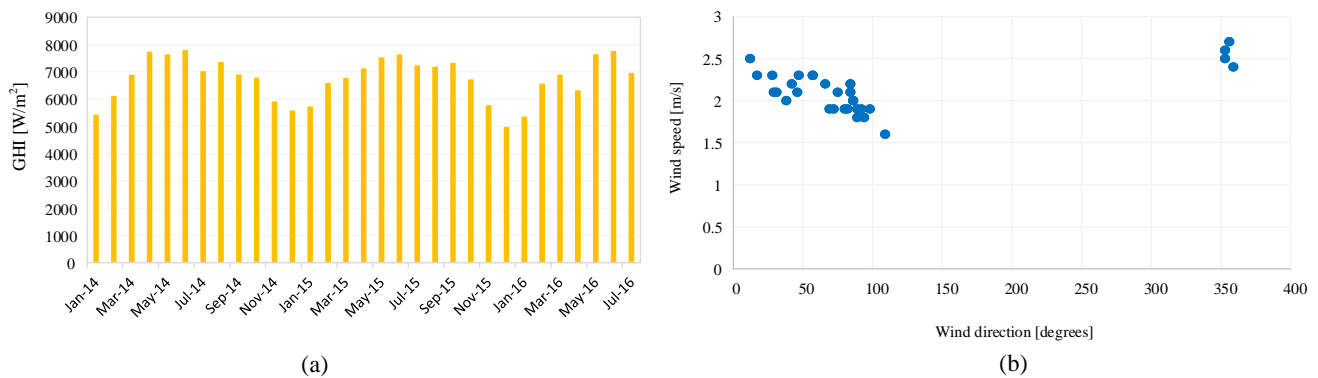


**Figure 1.** Site location used in this study. (a) Najran University; (b) Solar and wind weather station.

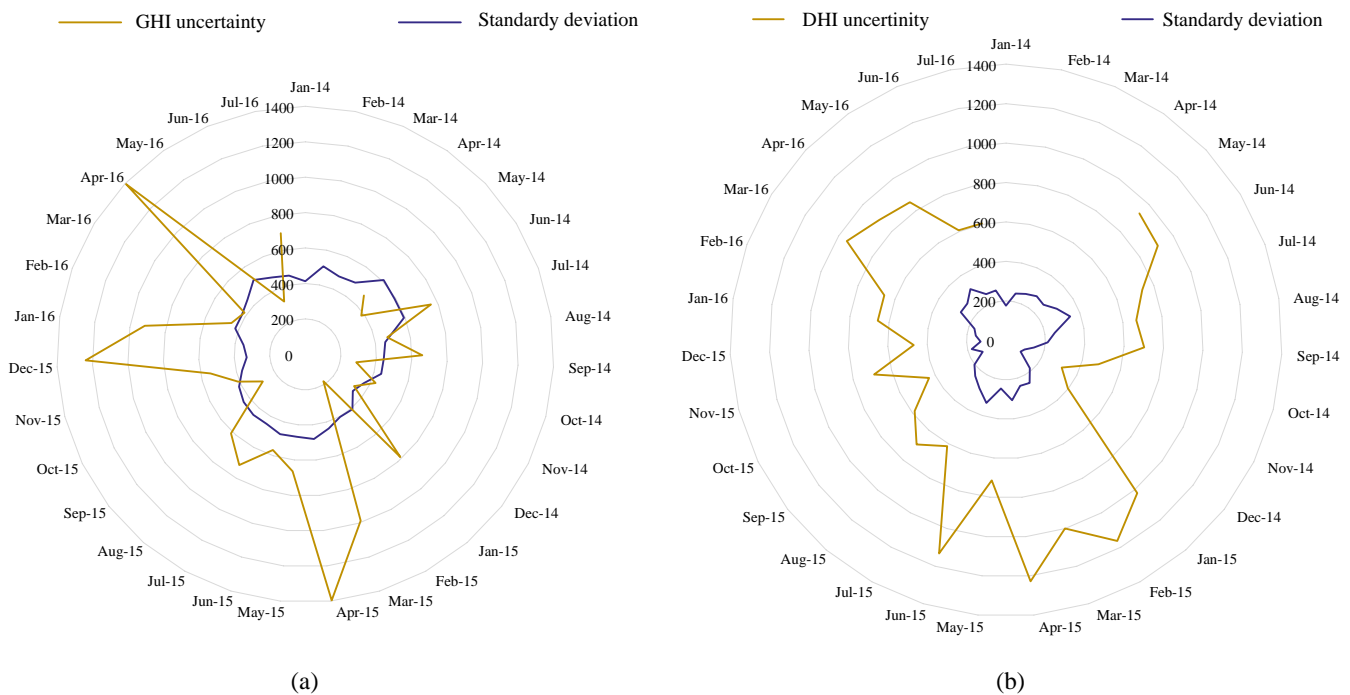
Figure 2a shows the monthly average of the DHI, which indicates the amount of solar irradiance that does not arrive directly on the PV panel. The temperature of the air is shown in Figure 2b. The monthly average is between 15 C and 35 degrees. The peak wind speed is the highest in April and August of every year, as depicted in Figure 2c. The relative humidity data prove the continental climate, which is mostly less than 30%, as shown in Figure 2d. Although several parameters are included in the dataset, the most critical parameters in this study are the GHI, temperature, humidity, and wind speed. Global horizontal irradiance (GHI) is the total amount of solar radiation falling on a surface horizontal to the ground. Najran city has one of the highest monthly averages of GHI, as shown in Figure 3a. The average value of wind speed is shown in Figure 3b. The uncertainty and standard deviation of both GHI and DHI are shown in Figure 4.



**Figure 2.** Monthly weather and diffused irradiance data of the Najran university. (a) The diffused horizontal irradiance. (b) Air temperature. (c) Peak wind speed. (d) Relative humidity.



**Figure 3.** Normalization energy sources data at the Najran university. (a) The global horizontal irradiance in Wh/m<sup>2</sup>. (b) The average wind speed in [m/s] vs. wind direction.



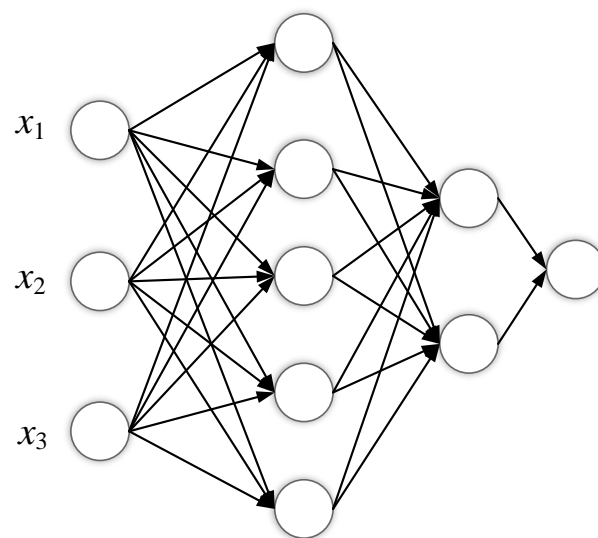
**Figure 4.** The standard deviation and uncertainty of the monthly (a) GHI (b) DHI.

### 3. Theoretical Illustration of Solar Irradiance Prediction Approaches

This section presents the theoretical background and mathematical formulation of the proposed prediction approaches.

#### 3.1. Extreme Learning Machine

One major disadvantage of the feedforward backpropagation algorithm is its slow learning speed. Therefore, the extreme learning machine has been introduced to solve computation and speed issues. The ELM was introduced in 2006 by Huang [38] and was utilized in several applications ranging from speech recognition to image processing. Figure 5 shows a basic ELM network, where ELM is a feedforward neural network that has a single random hidden neurons layer. The main idea of this method is to find the weight vector that maps the output  $y$  to the transformed input  $h$ . The reason that ELM converges faster than the backpropagation algorithm is because of randomly hidden neurons. At the same time, ELM can have a better generalization and avoid overfitting issues. According to Bartlett's theory, generalization performance is better when the training error is smaller. Therefore, ELM can reach the slightest training error and can run extremely quickly [9].



Input layer    Hidden layer    Output layer

**Figure 5.** Single-layer extreme learning machine.

#### Random Hidden Nodes for SLFNs

Let us suppose that we have  $N$  arbitrary number of samples  $(x_i, y_i)$  where  $x$  is the input vector and  $y_i$  is the output vector. The SLFNs with  $N$  hidden neurons can be mathematically modeled as follows:

$$\sum_{i=1}^N \beta_i g_i(x_j) = \sum_{i=1}^N \beta_i g(w_i \cdot x_i + b_i) = y_j \quad (1)$$

where  $w$  is the weight vector that connects the neurons in the hidden neuron with input neurons, and  $\beta$  is the weight vector that connects the hidden neuron layer to the output layer. The term  $(w_i, x_i)$  represents the inner product  $w_i$  and  $x_i$ . The  $N$  samples can be approximated with the zero error means that makes the sum of the difference between the predicted and actual value of the output equals zero, which is given by

$$\sum_{j=1}^N \|y_{predicted} - y_{actual}\| = 0 \quad (2)$$

The ELM algorithm is a straightforward and efficient way to train the single hidden-layer neural network [39,40]. The algorithm of the ELM is illustrated in Algorithm 1.

---

#### **Algorithm 1** ELM algorithm

---

- 1: Given training data samples  $(X_i, t_i)$  and an activation function
  - 2: Assign random input bias and weight  $(b, w)$
  - 3: Compute the output of the hidden layer
  - 4: Compute the weight matrix  $\beta$
- 

#### 3.2. Adaptive Extreme Learning Machine

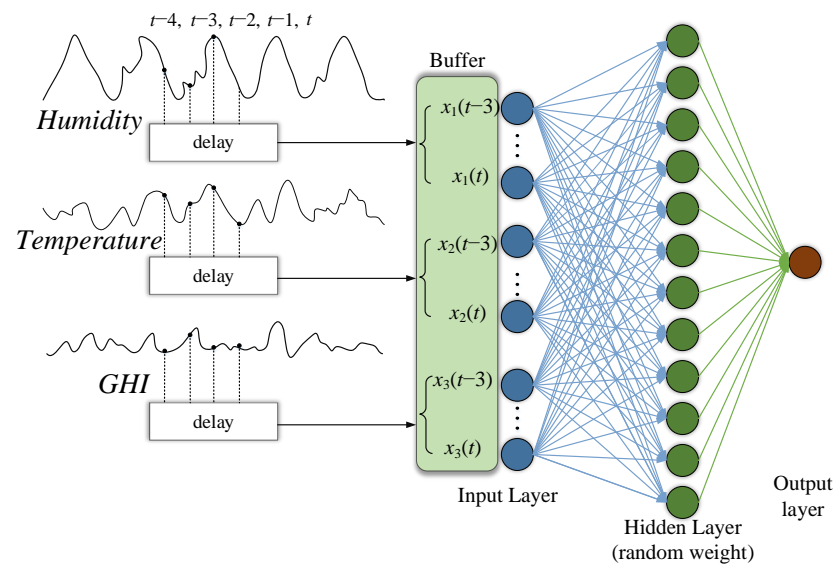
As mentioned previously, the simple aim for ELM is to calculate the output layer  $\beta$  that transforms the output of the hidden layer  $C$  to output  $Y$ . The figure shows the ELM with a buffer contains several previous samples from the input parameters. Next, we show the input

$$\mathbf{x} = [\text{GHI Temp humidity}]_{n \times z} \quad (3)$$



where  $z$  is the number of input variables and  $n$  is the buffer size. The number of including previous samples determines the buffer size. The input layer is multiplied by a weight vector filled with random values, which is usually sampled from Gaussian random noise. Figure 6 shows the adaptive ELM used to predict the time series. The multiplication is mapped to the hidden layer through an activation function  $G(\mathbf{a}_i, \mathbf{x})$

$$G(\mathbf{a}_i, \mathbf{x}) = \sigma([GHI \text{ Temp humidity}]_{n \times z} \begin{bmatrix} \text{Normal} - \text{dist}(\mu = 0, \sigma^2 = 1) \\ \vdots \\ \text{Normal} - \text{dist}(\mu = 0, \sigma^2 = 1) \end{bmatrix}_{z \times 1}). \quad (4)$$



**Figure 6.** Extreme Learning Machine with time series input and sliding windows.

The output from Equation (4) is transformed by hidden layer  $\mathbf{C}$ , which is formulated as follows:

$$\mathbf{C}(\mathbf{a}, \mathbf{x}) = \begin{bmatrix} G(\mathbf{a}_1, \mathbf{x}_1) & \dots & G(\mathbf{a}_{\tilde{N}}, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ G(\mathbf{a}_1, \mathbf{x}_n) & \dots & G(\mathbf{a}_{\tilde{N}}, \mathbf{x}_n) \end{bmatrix}_{n \times \tilde{N}} \quad (5)$$

The matrix  $\mathbf{C}$  is the output of the hidden neurons. The output  $\mathbf{Y}$  is calculated by multiplying  $\mathbf{C}$  with the output weight vector  $\beta$ , as follows:

$$\mathbf{Y} = \mathbf{C}\beta \quad (6)$$

The aim is to predict the accurate value of solar irradiance, so the output weight matrix needs to be determined. The output weight matrix can be solved using the least square solution and can also be extended by a diagonal weight matrix.

$$\beta = (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \mathbf{Y} = (\mathbf{C}^\top \mathbf{W} \mathbf{C})^{-1} \mathbf{C}^\top \mathbf{W} \mathbf{Y}. \quad (7)$$

In light of newly generated sample data pairs, dividing the solution into offline and online components would make it possible to reduce the need for processing power and storage space. In order to do this, the matrices are first decomposed into an offline part and an online part, which can be represented by:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{\text{OF}} \\ \mathbf{C}_{\text{ON}} \end{bmatrix}_{n \times \tilde{N}} \quad \mathbf{W} = \begin{bmatrix} \mathbf{W}_{\text{OF}} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{\text{ON}} \end{bmatrix}_{n \times N} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_{\text{OF}} \\ \mathbf{Y}_{\text{ON}} \end{bmatrix}_{n \times 1}. \quad (8)$$

The derivation for the recursive least square method is quite similar to that of the feedforward ELM, except it includes an additional weight to inversion part  $\mathbf{C}^\top \mathbf{W} \mathbf{C}$ . The inversion part is also denoted offline and online by  $\mathbf{C}^\top \mathbf{W} \mathbf{C}$ , respectively. Therefore, the inverted part of Equation (7) is simplified to:

$$\begin{aligned} \mathbf{K}_{\text{ON}} &= \mathbf{C}^\top \mathbf{W} \mathbf{C} = \begin{bmatrix} \mathbf{C}_{\text{OF}}^\top & \mathbf{C}_{\text{ON}}^\top \end{bmatrix} \begin{bmatrix} \mathbf{W}_{\text{OF}} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{\text{ON}} \end{bmatrix} \begin{bmatrix} \mathbf{C}_{\text{OF}} \\ \mathbf{C}_{\text{ON}} \end{bmatrix} \\ &= \mathbf{C}_{\text{OF}}^\top \mathbf{W}_{\text{OF}} \mathbf{C}_{\text{OF}} + \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} \mathbf{C}_{\text{ON}} = \mathbf{K}_{\text{OF}} + \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} \mathbf{C}_{\text{ON}}. \end{aligned} \quad (9)$$

Similarly, the non-inverted part of Equation (7) can be simplified to:

$$\begin{aligned} \mathbf{C}^\top \mathbf{W} \mathbf{Y} &= \begin{bmatrix} \mathbf{C}_{\text{OF}}^\top & \mathbf{C}_{\text{ON}}^\top \end{bmatrix} \begin{bmatrix} \mathbf{W}_{\text{OF}} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{\text{ON}} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_{\text{OF}} \\ \mathbf{Y}_{\text{ON}} \end{bmatrix} = \mathbf{C}_{\text{OF}}^\top \mathbf{W}_{\text{OF}} \mathbf{Y}_{\text{OF}} + \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} \mathbf{Y}_{\text{ON}} \\ &= \mathbf{K}_{\text{OF}} \mathbf{K}_{\text{OF}}^{-1} \mathbf{C}_{\text{OF}}^\top \mathbf{W}_{\text{OF}} \mathbf{Y}_{\text{OF}} + \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} \mathbf{Y}_{\text{ON}} \\ &= \mathbf{K}_{\text{OF}} \boldsymbol{\beta}_{\text{OF}} + \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} \mathbf{Y}_{\text{ON}} \\ &= (\mathbf{K}_{\text{ON}} - \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} \mathbf{C}_{\text{ON}}) \boldsymbol{\beta}_{\text{OF}} + \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} \mathbf{Y}_{\text{ON}} \\ &= \mathbf{K}_{\text{ON}} \boldsymbol{\beta}_{\text{OF}} - \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} \mathbf{C}_{\text{ON}} \boldsymbol{\beta}_{\text{OF}} + \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} \mathbf{Y}_{\text{ON}}. \end{aligned} \quad (10)$$

Using Equation (10) to obtain the full online weight

$$\begin{aligned} \boldsymbol{\beta}_{\text{ON}} &= (\mathbf{K}_{\text{ON}}^{-1}) (\mathbf{C}^\top \mathbf{W} \mathbf{Y}) = \boldsymbol{\beta}_{\text{OF}} - \mathbf{K}_{\text{ON}}^{-1} \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} \mathbf{C}_{\text{ON}} \boldsymbol{\beta}_{\text{OF}} + \mathbf{K}_{\text{ON}}^{-1} \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} \mathbf{Y}_{\text{ON}} \\ &= \boldsymbol{\beta}_{\text{OF}} + \mathbf{K}_{\text{ON}}^{-1} \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} (\mathbf{Y}_{\text{ON}} - \mathbf{C}_{\text{ON}} \boldsymbol{\beta}_{\text{OF}}) \end{aligned} \quad (11)$$

The online weight can be calculated without the requirement for the offline dataset. The  $\tilde{N} \times \tilde{N}$  sized  $\mathbf{K}$  inverse can be computationally expensive. In order to reduce the computations, a smaller-sized buffer for the previous value of the solar irradiance is used. One can let  $\mathbf{P} = \mathbf{K}^{-1}$

$$\mathbf{P}_{\text{OF}} = \mathbf{K}_{\text{OF}}^{-1} = (\mathbf{C}_{\text{OF}}^\top \mathbf{W}_{\text{OF}} \mathbf{C}_{\text{OF}})^{-1} \quad (12)$$

$$\mathbf{P}_{\text{ON}} = \mathbf{K}_{\text{ON}}^{-1} = (\mathbf{P}_{\text{OF}}^{-1} + \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} \mathbf{C}_{\text{ON}})^{-1} \quad (13)$$

and apply the Woodbury formula on Equation (13) to get the following:

$$\mathbf{P}_{\text{ON}} = \mathbf{P}_{\text{OF}} - \mathbf{P}_{\text{OF}} \mathbf{C}_{\text{ON}}^\top (\mathbf{W}_{\text{ON}}^{-1} + \mathbf{C}_{\text{ON}} \mathbf{P}_{\text{OF}} \mathbf{C}_{\text{ON}}^\top)^{-1} \mathbf{C}_{\text{ON}} \mathbf{P}_{\text{OF}}. \quad (14)$$

More simplification can be applied to the presented algorithm, but first, the term the  $\mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}}$  part is added to leaves the term of Equation (11) to Equation (14) and distributes  $\mathbf{C}_{\text{ON}}^\top$  to get the following equation

$$\mathbf{P}_{\text{ON}} \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} = \left[ \mathbf{P}_{\text{OF}} \mathbf{C}_{\text{ON}}^\top - \mathbf{P}_{\text{OF}} \mathbf{C}_{\text{ON}}^\top (\mathbf{W}_{\text{ON}}^{-1} + \mathbf{C}_{\text{ON}} \mathbf{P}_{\text{OF}} \mathbf{C}_{\text{ON}}^\top)^{-1} \cdot \mathbf{C}_{\text{ON}} \mathbf{P}_{\text{OF}} \mathbf{C}_{\text{ON}}^\top \right] \mathbf{W}_{\text{ON}}. \quad (15)$$

Equation (15) is clarified by substituting  $\mathbf{A} = \mathbf{P}_{\text{OF}} \mathbf{C}_{\text{ON}}^\top$ ,  $\mathbf{B} = \mathbf{C}_{\text{ON}} \mathbf{A}$  and then distributing  $\mathbf{W}_{\text{ON}}$  to obtain the following equation:

$$\mathbf{P}_{\text{ON}} \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} = \mathbf{A} \left[ \mathbf{W}_{\text{ON}} - (\mathbf{W}_{\text{ON}}^{-1} + \mathbf{B})^{-1} \mathbf{B} \mathbf{W}_{\text{ON}} \right]. \quad (16)$$

After further simplification, the following equation

$$\mathbf{P}_{\text{ON}} \mathbf{C}_{\text{ON}}^\top \mathbf{W}_{\text{ON}} = \mathbf{A} (\mathbf{W}_{\text{ON}}^{-1} + \mathbf{B})^{-1}. \quad (17)$$



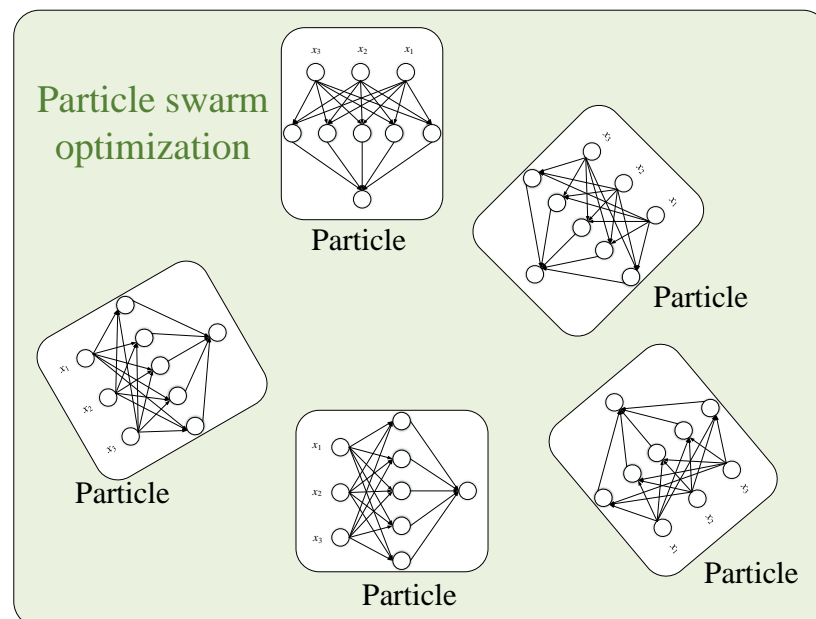
The inverse relation between  $\mathbf{P}_{\text{ON}}$  and  $\mathbf{K}_{\text{ON}}^{-1}$  is known. Therefore,  $\mathbf{P}_{\text{ON}}\mathbf{C}_{\text{ON}}^{\top}\mathbf{W}_{\text{ON}} = \mathbf{K}_{\text{ON}}^{-1}\mathbf{C}_{\text{ON}}^{\top}\mathbf{W}_{\text{ON}}$ . The online output weight can be solved by substituting Equation (17) into Equation (11). The summary of the offline and online computation is illustrated in Table 1

**Table 1.** Summary of the offline and online computations.

Stage	Equations
Initial Training (offline)	$\mathbf{P}_{\text{OF}} = \left[ \left( \mathbf{C}_{\text{OF}}^{\top}\mathbf{W}_{\text{OF}}\mathbf{C}_{\text{OF}} \right)^{-1} \right]_{\tilde{N} \times \tilde{N}}$ $\boldsymbol{\beta}_{\text{OF}} = \left[ \mathbf{P}_{\text{OF}}\mathbf{C}_{\text{OF}}^{\top}\mathbf{W}_{\text{OF}}\mathbf{Y}_{\text{OF}} \right]_{\tilde{N} \times 1}$
Online Adaptive mode	$\mathbf{A} = \mathbf{P}_{\text{OF}}\mathbf{C}_{\text{ON}}^{\top}, \mathbf{B} = \mathbf{C}_{\text{ON}}\mathbf{A}$ $\boldsymbol{\beta}_{\text{ON}} = \boldsymbol{\beta}_{\text{OF}} + \mathbf{A} \left( \mathbf{W}_{\text{ON}}^{-1} + \mathbf{B} \right)^{-1} (\mathbf{Y}_{\text{ON}} - \mathbf{C}_{\text{ON}}\boldsymbol{\beta}_{\text{OF}})$
Online Prediction	$GHI = \mathbf{Y}_{n+1} = \mathbf{C}(\mathbf{a}, \mathbf{x}_{n+1})\boldsymbol{\beta}_{\text{ON}}$

### 3.3. Feed Forward Neural Network Based Particle Optimization

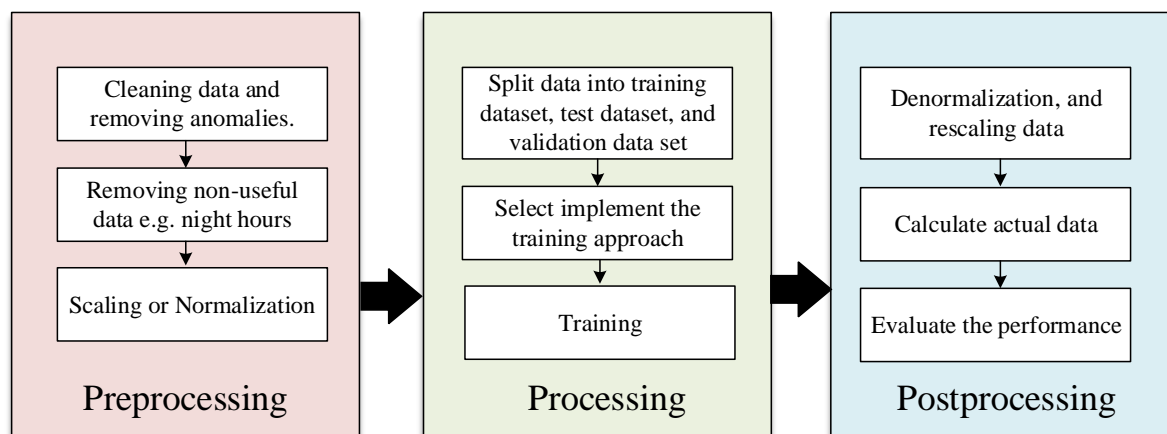
This is similar to the conventional particle swarm optimization algorithm. However, the particles are the neural network's output, as shown in Figure 7. The algorithm completes the search when the optimal weight is calculated. A similar approach is used in energy-management systems with the Internet of Things (IoTs) [41], and in the prediction of the composite behavior [42].



**Figure 7.** Feed forward neural network trained with PSO.

## 4. Prediction Methodology

The prediction process is divided into three stages, as shown in Figure 8. The first stage is preprocessing, where the data are prepared; the second stage is about training a machine learning model. The last stage is the post-processing and the performance evaluation.



**Figure 8.** The prediction methodology, which consists of three stages.

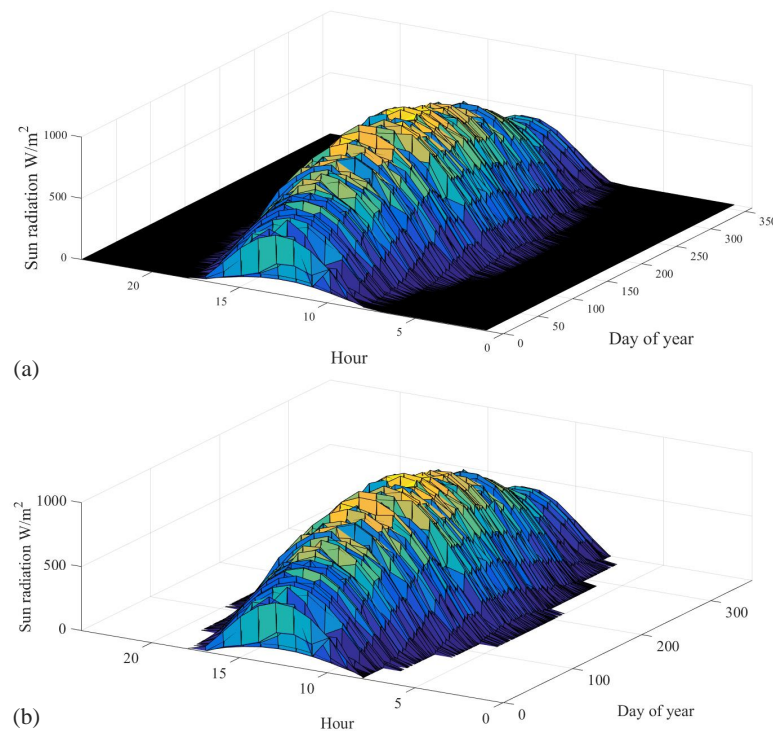
#### 4.1. Data Preprocessing and Data Cleaning

The dataset represents three years of weather samples; each year contains 17,520 samples. Raw data often contain errors and misreadings, which require preprocessing before they can be used in model training. The first step is to clean the data from anomalous sensor readings and missing data points with interpolated values. An example of incorrect readings is when the GHI has negative values, where the minimum value of solar irradiance is zero watts per square meter. Advanced techniques can also be used to clean the data, such as incorporating maximum temperature and humidity change per day. After cleaning the data, only valuable data and features are used. The useful data are the data points that start from 6 a.m. to 8 p.m., only during the day. The season affects the sunrise and sunset time, and the same goes for the peak hours. Figure 9a illustrates the irradiance before removing irradiance, and Figure 9b shows the irradiance after getting rid of non-useful irradiance data.

After cleaning the data and extracting valuable information, normalization is applied to scale all data between two values. Normalization allows the model to be trained faster, helps us avoid being trapped in local minima, and makes the gradient-based algorithms treat all features equally. The normalization used in this study is between zero and one, and the following equation is used to normalize the data.

$$\check{GHI} = \frac{GHI - GHI_{\min}}{GHI_{\max} - GHI_{\min}} \quad (18)$$

where  $GHI_{\min}$  and  $GHI_{\max}$  are the minimum and the maximum values of the original GHI dataset, respectively, and  $GHI$  and  $\check{GHI}$  are the normalized and the denormalized value of the solar irradiance, respectively. Similarly, the temperature and humidity follow the same normalization process.



**Figure 9.** The data cleaning process (a) the data before eliminating night hours (b) the data after eliminating night hours.

#### 4.2. Processing Stage

After preprocessing, the data are split into training, testing, and validation sets. The training takes a major portion of the data, accounting for about 70% of the dataset. The testing and validation account for 15% each. The adaptive ELM model begins by populating the weight matrices with random numbers. The random numbers are used in high precision to avoid the singularity. Then, the initial hidden layer matrix  $C$  is calculated based on previous samples. Then, the buffer size is selected. It is usually selected based on trial and error. Then, the computation initial training is completed using the equations listed in Table 1. The online adaptive mode only needs initial training once, and there is no need for previous results after that. The output weight matrix in online mode is calculated, and then the next value of GHI can be calculated. Note that implementation of this method on low-cost hardware depends on the complexity of the activation functions.

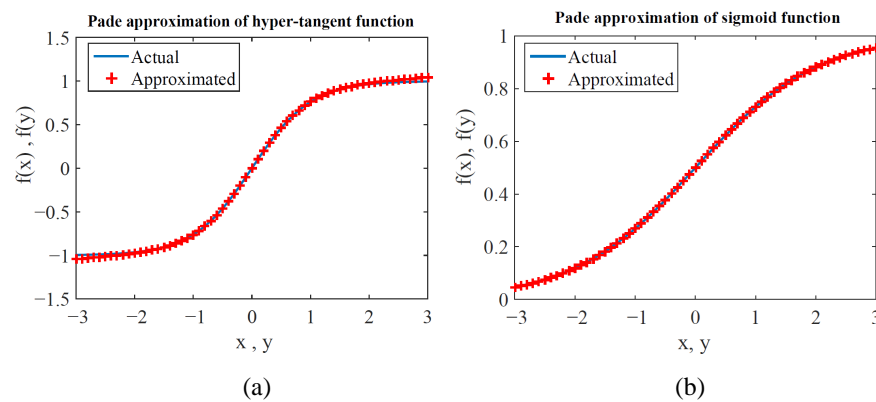
The activation function brings a degree of nonlinearity to the data, and several types of activation functions use the exponent function. The most common are logistic function and hyperbolic tangent activation function, and they are given in (19) and (20), respectively. Note that some microcontrollers and digital signal processing devices will be significantly slowed if assigned to calculate the exponential function. Therefore, an approximation of the activation function can be used in this case to speed up the computation. Using Padé approximation, the logistic and hyperbolic tangent are given in (21) and (22), respectively. Figure 10 approximated functions fit well for the range from  $[-1, 1]$  and where the activation function is used to form the network [43].

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (19)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (20)$$

$$f(y)_{\text{sigmoid}} = \frac{120 + 60y + 12y^2 + y^3}{120 - 60y + 12y^2 - y^3} \quad (21)$$

$$f(y)_{\tanh} = \frac{y(y^2 + 15)}{6y^2 + 15} \quad (22)$$



**Figure 10.** Padé approximation of the activation function. (a) Tanh activation function. (b) Sigmoid function.

#### 4.3. Post Processing Stage

The first step after we found the predicted values was to denormalize the data. Data can be renormalized using the following equation

$$GHI = \left( \frac{G\tilde{H}I - G\tilde{H}I_{\min}}{G\tilde{H}I_{\max} - G\tilde{H}I_{\min}} \right) \quad (23)$$

where  $G\tilde{H}I_{\min}$  and  $G\tilde{H}I_{\max}$  are the minimum and the maximum values of the normalized GHI dataset, respectively. The GHI and  $G\tilde{H}I$  are the normalized and the denormalized value of the solar irradiance, respectively. Similarly, the temperature and humidity follow the same denormalization process. The data are checked to see if there are no outliers or negative values resulting from ill-conditioned matrices. The performance of a model can be evaluated using various methods, such as the mean absolute error (MAE), the mean squared error (MSE), and the root mean squared error (RMSE). They can be calculated using (24)–(26), respectively. In hardware implementation and testing, only MSE is used, which determines the quality of the predictor.

$$MAE = \frac{1}{n} \sum_{i=1}^n |GHI_{\text{predicted}}[i] - GHI_{\text{actual}}[i]| \quad (24)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (GHI_{\text{predicted}}[i] - GHI_{\text{actual}}[i])^2 \quad (25)$$

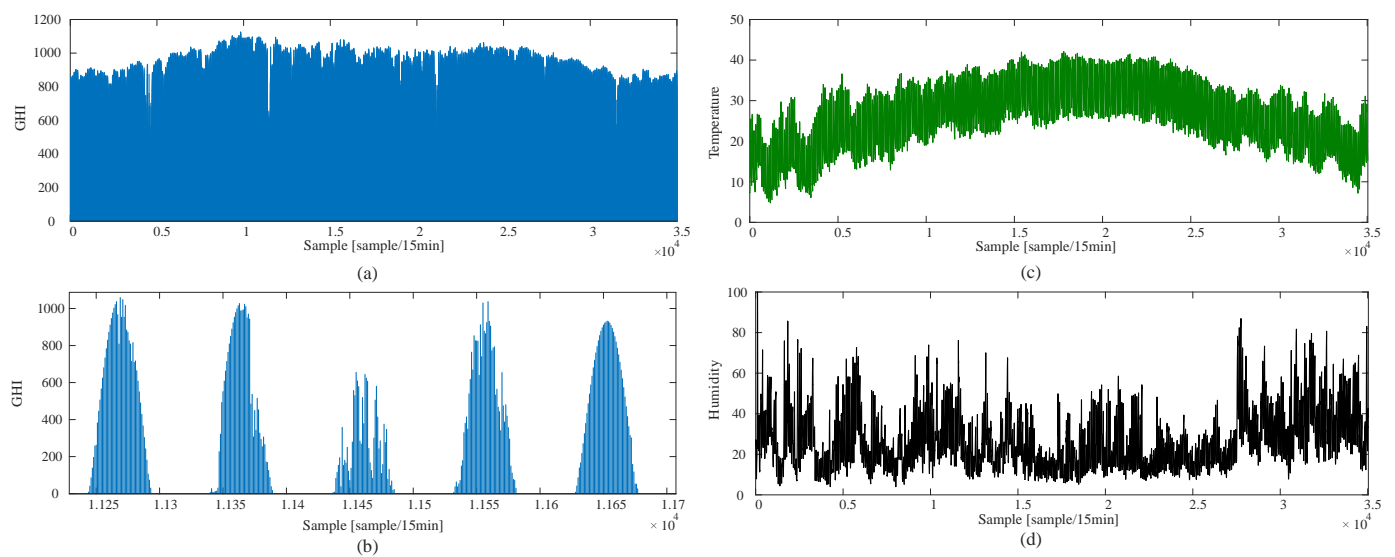
$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (GHI_{\text{predicted}}[i] - GHI_{\text{actual}}[i])^2} \quad (26)$$

where  $n$  is the number of samples,  $GHI_{\text{actual}}$  is the real values, and  $GHI_{\text{predicted}}$  is the output values of the model.

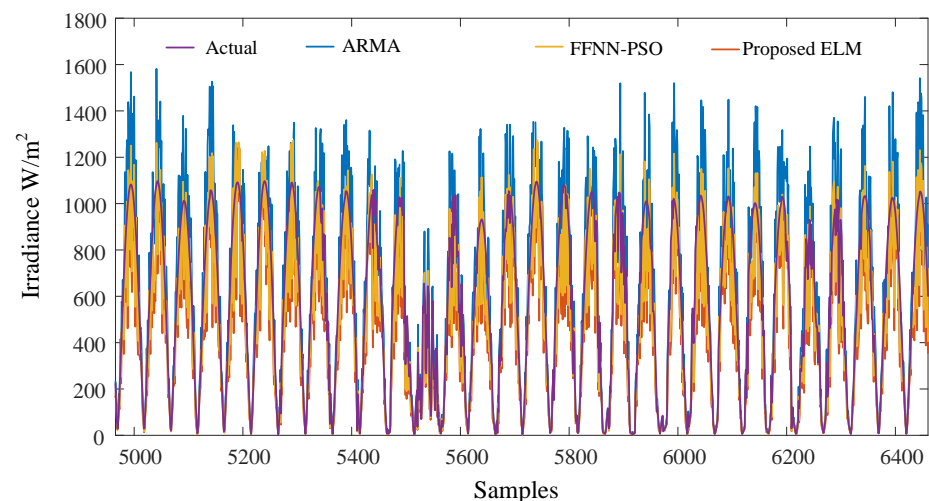
## 5. Results and Discussion

The proposed algorithm is compared to several commonly used approaches, such as ARMA and FFNN-based PSO. Figure 11 shows the samples of the data used in training and testing. Each sample was taken 15 min after the previous one. The irradiance is less than  $1150 \text{ W/m}^2$ , the maximum temperature is less than 43 degrees, and the humidity is mostly less than 50%. Figure 12 shows the overall performance of the algorithm. The ARMA has the fastest training time. However, it shows the worst performance, since the predicted output significantly differs from the actual values, especially during peak hours.

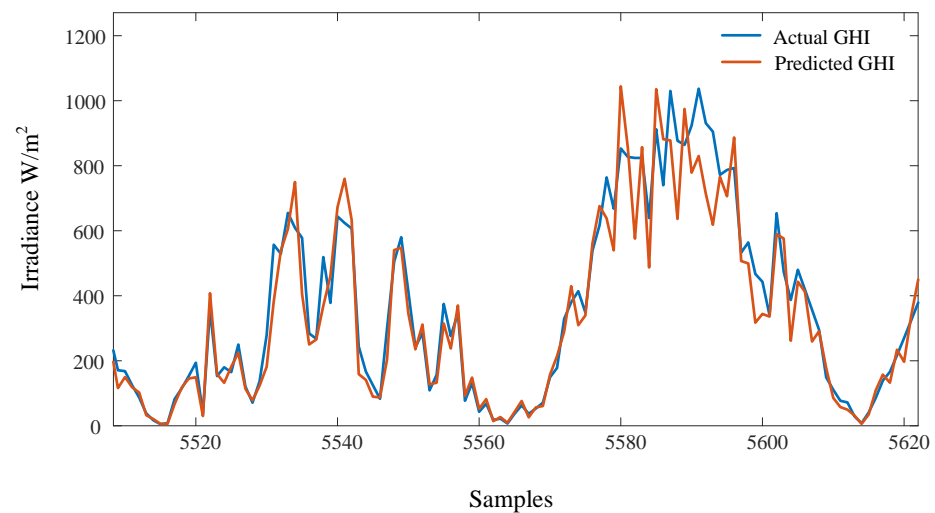
The FFNN-PSO is an enhanced performance over the ARMA, but it takes a very long time to train and test, and still, there are some significant mismatching results. The proposed approach produces better results and requires less computation time compared to the other techniques, as the predicted values are close to the actual values. The performance of the previously mentioned algorithms is compared in terms of MAE, MSE, and RMSE, as shown in Table 2. The ARMA approach has the worst performance as the MAE is 0.3124, MSE is 0.233, and RMSE is 0.4463. MAE of about 0.2675 indicates the FFNN-PSO performance, MSE of 0.1880, and 0.3684 RMSE. The proposed method performs slightly better, where the MAE is 0.2444, MSE is 0.1727, and RMSE is 0.3012. The proposed method's predicted samples are shown in Figure 13. The error, which is the absolute error, is shown in Figure 14.



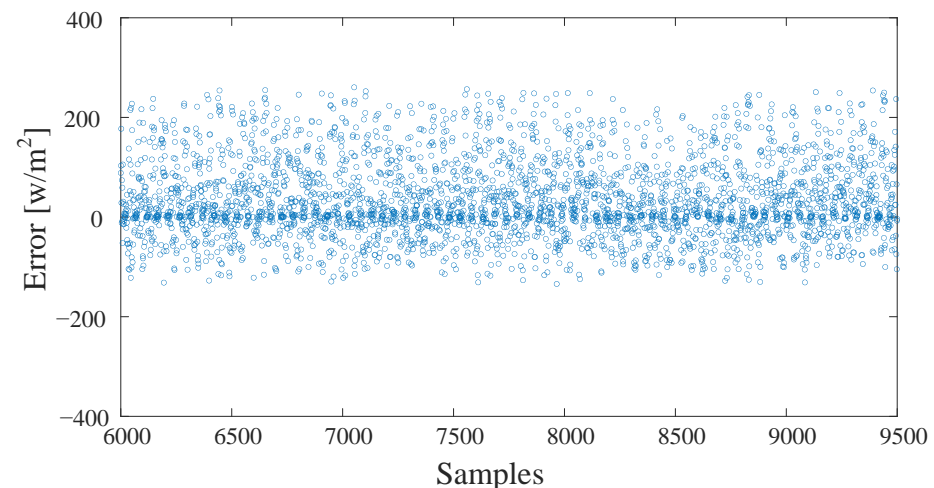
**Figure 11.** The data used in training and testing. (a) The GHI for a whole year. (b) The GHI for five working days. (c) The temperature in degree celsius. (d) Relative humidity in percentage.



**Figure 12.** The predicted samples of the ARMA, FNN-PSO, and the proposed method compared to the actual values of GHI.



**Figure 13.** The predicted samples of the proposed method compared to the actual values of GHI. The samples are recorded every 15 min.



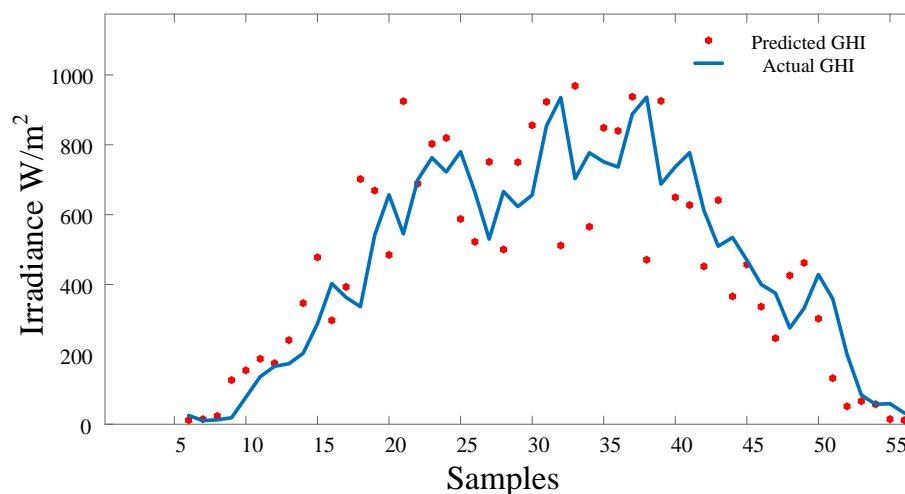
**Figure 14.** The error in samples prediction of the GHI, which is the difference between the proposed method's output and the GHI's actual values. Note that the samples are concentrated near 0, and the maximum error is less than  $250 \frac{W}{m^2}$ .

**Table 2.** Performance comparison between different algorithms.

Prediction Approach	MAE	MSE	RMSE
ARMA	0.3124	0.2133	0.4463
FFNN-PSO	0.2675	0.1880	0.3684
Proposed method	0.2444	0.1727	0.3012

The adaptive ELM and FFNN-PSO are used as predictive approaches in hardware implementation (Atmega328 and PC). The results are presented and compared to a linear regression algorithm, as listed in Table 3. Different prediction times were experimented with to show the correlation between the predicted value and the previous hours or time horizon. It can be noticed that prediction in shorter time horizons yields better results. Therefore, the trade-off between computation time and time horizon must be figured out to obtain the optimized results. The adaptive ELM is implemented in real time to examine the performance. Data were collected from 6:30 a.m. until 5:30 p.m. on 18 August 2022. Figure 15 are samples of the graphs of data taken at a resolution of a single data point per 15 min. Data were acquired on the south side of the Engineering College building at Najran

University. The results show the robust performance and ability to estimate the next value of solar irradiance.



**Figure 15.** The experimental results of GHI. The predicted results (dots) are read every 15 min and compared to the actual GHI (solid line).

**Table 3.** Comparison of the experimental performance

Method	Training Time	Training MSE	Testing Time	Testing MSE
Initial ELM	0.0252	-	0.0046	0.2511
Adaptive ELM	0.2884	-	0.0062	0.2459
1 h regression	0.0022	0.3262	0.0007	0.4257
2 h regression	0.0094	0.4144	0.0005	0.4282
1 h NN PSO	30.5079	0.2320	0.0912	0.2552
4 h NN PSO	43.9875	0.1679	0.0020	0.2024

## 6. Conclusions

This paper presented a prediction approach based on an adaptive extreme learning machine. The proposed algorithm works both offline and online, with reduced computational time and higher prediction accuracy. The dataset used in this study is based on a local site at Najran University. Several preprocessing steps were taken to ensure the training data are valuable, clean, and suitable for training predictive models. The offline performance of the proposed approach yields the lowest mean square error, about 0.1727, and the lowest mean absolute error, less than 0.25. Furthermore, the presented method can be implemented on hardware, which can be tested in less than 0.0062s, with a mean square error of about 0.2459. In the future, this method can be integrated into the whole energy management system or utilities to enhance the performance of the power grid and allow a higher level of renewable energy integration.

**Funding:** This work was funded by the General Research Funding program grant code (NU/-/SERC/10/557).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data could be shared upon request.

**Acknowledgments:** The author is thankful to the Deanship of Scientific Research at Najran University for funding this work under the General Research Funding program grant code (NU/-/SERC/10/557).

**Conflicts of Interest:** The author declares no conflict of interest.



## Abbreviations

The following abbreviations are used in this manuscript:

ELM	Extreme learning machine.
FFNN	Feedforward neural network.
GHI	Global horizontal irradiance.
DNI	Direct Normal Irradiance letter acronym.
PSO	Particle swarm optimization.
NSRDB	The National Solar Radiation Database.

## References

- Jacobson, M.Z.; Delucchi, M.A.; Bazouin, G.; Bauer, Z.A.F.; Heavey, C.C.; Fisher, E.; Morris, S.B.; Piekutowski, D.J.Y.; Vencill, T.A.; Yeskoo, T.W. 100% clean and renewable wind, water, and sunlight (WWS) all-sector energy roadmaps for the 50 United States. *Energy Environ. Sci.* **2015**, *8*, 2093–2117. [CrossRef]
- Shahsavari, A.; Akbari, M. Potential of solar energy in developing countries for reducing energy-related emissions. *Renew. Sustain. Energy Rev.* **2018**, *90*, 275–291. [CrossRef]
- Alnaser, W.; Alnaser, N. The impact of the rise of using solar energy in GCC countries. In *Renewable Energy and Sustainable Buildings*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 167–183.
- Sakaka Solar Power Plant Project. Available online: <https://www.vision2030.gov.sa/v2030/v2030-projects/sakaka-solar-power-plant/> (accessed on 9 September 2022).
- Crook, J.A.; Jones, L.A.; Forster, P.M.; Crook, R. Climate change impacts on future photovoltaic and concentrated solar power energy output. *Energy Environ. Sci.* **2011**, *4*, 3101–3109. [CrossRef]
- Imam, A.A.; Al-Turki, Y.A.; R., S.K. Techno-Economic Feasibility Assessment of Grid-Connected PV Systems for Residential Buildings in Saudi Arabia—A Case Study. *Sustainability* **2020**, *12*, 262. [CrossRef]
- ECRA. The Electricity & Cogeneration Regulatory Authority. 2014. Available online: <http://www.ecra.gov.sa/en-us/MediaCenter/doclib2/Pages/SubCategoryList.aspx?categoryID=4> (accessed on 9 September 2022).
- Kumar, A.; Gupta, N.; Gupta, V. A comprehensive review on grid-tied solar photovoltaic system. *J. Green Eng.* **2017**, *7*, 213–254. [CrossRef]
- Refaat, S.S.; Abu-Rub, H.; Sanfilippo, A.P.; Mohamed, A. Impact of grid-tied large-scale photovoltaic system on dynamic voltage stability of electric power grids. *IET Renew. Power Gener.* **2018**, *12*, 157–164. [CrossRef]
- Olowu, T.O.; Sundararajan, A.; Moghaddami, M.; Sarwat, A.I. Future challenges and mitigation methods for high photovoltaic penetration: A survey. *Energies* **2018**, *11*, 1782. [CrossRef]
- Khan, Z.A.; Jayaweera, D. Smart meter data based load forecasting and demand side management in distribution networks with embedded PV systems. *IEEE Access* **2019**, *8*, 2631–2644. [CrossRef]
- Manojkumar, R.; Kumar, C.; Ganguly, S.; Gooi, H.B.; Mekhilef, S. Voltage control using smart transformer via dynamic optimal setpoints and limit tolerance in a residential distribution network with pv sources. *IET Gener. Transm. Distrib.* **2020**, *14*, 5143–5151. [CrossRef]
- Smith, J.; Sunderman, W.; Dugan, R.; Seal, B. Smart inverter volt/var control functions for high penetration of PV on distribution systems. In Proceedings of the 2011 IEEE/PES Power Systems Conference and Exposition, Phoenix, AZ, USA, 20–23 March 2011; pp. 1–6.
- Bletterie, B.; Kadam, S.; Pitz, R.; Abart, A. Optimisation of LV networks with high photovoltaic penetration—Balancing the grid with smart meters. In Proceedings of the 2013 IEEE Grenoble Conference, Grenoble, France, 16–20 June 2013; IEEE: New York, NY, USA, 2013; pp. 1–6.
- Bueso, M.C.; Paredes-Parra, J.M.; Mateo-Aroca, A.; Molina-García, A. Sensitive Parameter Analysis for Solar Irradiance Short-Term Forecasting: Application to LoRa-Based Monitoring Technology. *Sensors* **2022**, *22*, 1499. [CrossRef]
- Dissawa, L.H.; Godaliyadda, R.I.; Ekanayake, P.B.; Agalgaonkar, A.P.; Robinson, D.; Ekanayake, J.B.; Perera, S. Sky Image-Based Localized, Short-Term Solar Irradiance Forecasting for Multiple PV Sites via Cloud Motion Tracking. *Int. J. Photoenergy* **2021**, *2021*, 9973010. [CrossRef]
- Bot, K.; Ruano, A.; Ruano, M.d.G. Short-Term Forecasting Photovoltaic Solar Power for Home Energy Management Systems. *Inventions* **2021**, *6*, 12. [CrossRef]
- Diagne, M.; David, M.; Lauret, P.; Bolland, J.; Schmutz, N. Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renew. Sustain. Energy Rev.* **2013**, *27*, 65–76. [CrossRef]
- Sreekumar, S.; Bhakar, R. Solar Power Prediction Models: Classification Based on Time Horizon, Input, Output and Application. In Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 11–12 July 2018; pp. 67–71. [CrossRef]
- Zhang, H.T.; Xu, F.Y.; Zhou, L. Artificial neural network for load forecasting in smart grid. In Proceedings of the 2010 International Conference on Machine Learning and Cybernetics, Qingdao, China, 11–14 July 2010; Volume 6; pp. 3200–3205. [CrossRef]

21. Alfari, F.; Alzahrani, A.; Kimball, J.W. Stochastic model for PV sensor array data. In Proceedings of the 2014 International Conference on Renewable Energy Research and Application (ICRERA), Milwaukee, WI, USA, 19–22 October 2014; pp. 798–803. [\[CrossRef\]](#)
22. Alzahrani, A.; Kimball, J.; Dagli, C. Predicting Solar Irradiance Using Time Series Neural Networks. *Procedia Comput. Sci.* **2014**, *36*, 623–628. [\[CrossRef\]](#)
23. Sobri, S.; Koochi-Kamali, S.; Rahim, N.A. Solar photovoltaic generation forecasting methods: A review. *Energy Convers. Manag.* **2018**, *156*, 459–497. [\[CrossRef\]](#)
24. Voyant, C.; Nutton, G.; Kalogirou, S.; Nivet, M.L.; Paoli, C.; Motte, F.; Fouilloy, A. Machine learning methods for solar radiation forecasting: A review. *Renew. Energy* **2017**, *105*, 569–582. [\[CrossRef\]](#)
25. Fu, P.; Xie, Y.; Weng, Q.; Myint, S.; Meacham-Hensold, K.; Bernacchi, C. A physical model-based method for retrieving urban land surface temperatures under cloudy conditions. *Remote. Sens. Environ.* **2019**, *230*, 111191. [\[CrossRef\]](#)
26. Naserpour, S.; Zolfaghari, H.; Zeaiean Firouzabadi, P. Calibration and evaluation of sunshine-based empirical models for estimating daily solar radiation in Iran. *Sustain. Energy Technol. Assess.* **2020**, *42*, 100855. [\[CrossRef\]](#)
27. Sudharshan, K.; Naveen, C.; Vishnuram, P.; Krishna Rao Kasagani, D.V.S.; Nastasi, B. Systematic Review on Impact of Different Irradiance Forecasting Techniques for Solar Energy Prediction. *Energies* **2022**, *15*. [\[CrossRef\]](#)
28. David, M.; Ramahatana, F.; Trombe, P.J.; Lauret, P. Probabilistic forecasting of the solar irradiance with recursive ARMA and GARCH models. *Sol. Energy* **2016**, *133*, 55–72. [\[CrossRef\]](#)
29. Eniola, V.; Suriwong, T.; Sirisamphanwong, C.; Ungchittrakool, K. Hour-ahead forecasting of photovoltaic power output based on hidden Markov model and genetic algorithm. *Int. J. Renew. Energy. Res* **2019**, *9*, 933–943.
30. Jaihuni, M.; Basak, J.K.; Khan, F.; Okyere, F.G.; Arulmozhi, E.; Bhujel, A.; Park, J.; Hyun, L.D.; Kim, H.T. A partially amended hybrid bi-GRU—ARIMA model (PAHM) for predicting solar irradiance in short and very-short terms. *Energies* **2020**, *13*, 435. [\[CrossRef\]](#)
31. Sanfilippo, A.; Pomares, L.; Perez-Astudillo, D.; Mohandes, N.; Bachour, D. Optimal selection of training datasets for solar nowcasting models. In Proceedings of the 32nd European Photovoltaic Solar Energy Conference and Exhibition, Munich, Germany, 20–24 June 2016; pp. 1482–1484.
32. Ji, W.; Chee, K.C. Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN. *Sol. Energy* **2011**, *85*, 808–817. [\[CrossRef\]](#)
33. Alfadda, A.; Adhikari, R.; Kuzlu, M.; Rahman, S. Hour-ahead solar PV power forecasting using SVR based approach. In Proceedings of the 2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 23–26 April 2017; pp. 1–5.
34. Awad, M.; Khanna, R. Support vector regression. In *Efficient Learning Machines*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 67–80.
35. Wang, F.; Mi, Z.; Su, S.; Zhao, H. Short-Term Solar Irradiance Forecasting Model Based on Artificial Neural Network Using Statistical Feature Parameters. *Energies* **2012**, *5*, 1355–1370. [\[CrossRef\]](#)
36. Jeon, H.J.; Choi, M.W.; Lee, O.J. Day-Ahead Hourly Solar Irradiance Forecasting Based on Multi-Attributed Spatio-Temporal Graph Convolutional Network. *Sensors* **2022**, *22*. [\[CrossRef\]](#)
37. Wilcox, S. *National Solar Radiation Database 1991–2005 Update: User's Manual*; Technical Report; National Renewable Energy Lab. (NREL): Golden, CO, USA, 2007.
38. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [\[CrossRef\]](#)
39. Huang, G.B.; Zhou, H.; Ding, X.; Zhang, R. Extreme Learning Machine for Regression and Multiclass Classification. In *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*; IEEE: New York, NY, USA, 2012; Volume 42, pp. 513–529. [\[CrossRef\]](#)
40. y. Liang, N.; b. Huang, G.; Saratchandran, P.; Sundararajan, N. A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. *IEEE Trans. Neural Netw.* **2006**, *17*, 1411–1423. [\[CrossRef\]](#)
41. Pawar, P.; TarunKumar, M.; K, P.V. An IoT based Intelligent Smart Energy Management System with accurate forecasting and load strategy for renewable generation. *Measurement* **2020**, *152*, 107187. [\[CrossRef\]](#)
42. Fadlallah, S.O.; Anderson, T.N.; Nates, R.J. Artificial neural network–particle swarm optimization (ANN-PSO) approach for behaviour prediction and structural optimization of lightweight sandwich composite heliostats. *Arab. J. Sci. Eng.* **2021**, *46*, 12721–12742. [\[CrossRef\]](#)
43. Vaughan, A.; Bohac, S.V. An Extreme Learning Machine Approach to Predicting Near Chaotic HCCI Combustion Phasing in Real-Time. *arXiv* **2013**, arXiv:1310.3567.