BŞEÜ Fen Bilimleri Dergisi
9(1), 340-356, 2022

BSEU Journal of Science
https://doi.org/10.35193/bseufbd.1037563

e-ISSN: 2458-7575 (https://dergipark.org.tr/tr/pub/bseufbd)

Araştırma Makalesi - Research Article

Prediction and Analysis of Weather Parameters with Global Horizontal Solar Irradiance Using LSTM-CNN Based Deep Learning Technique

LSTM-CNN Tabanlı Derin Öğrenme Tekniği Kullanılarak Küresel Yatay Güneş Radyasyonu ile Hava Durumu Parametrelerinin Tahmini ve Analizi

Sercan Yalçın[1*], Münür Sacit Herdem[2]

Geliş / Received: 16/12/2021 ***Revize / Revised: 03/06/2022*** ***Kabul / Accepted: 08/06/2022***

ABSTRACT

Predicting global horizontal solar irradiance (GHSI) as well as important climate parameters plays an important role in energy management and resource planning of photovoltaic panels. To further benefit from solar energy, it is necessary to obtain information regarding future values by frequently analyzing and predicting such time series parameter data. Hence, predicting long-term solar irradiance data is a challenging task. For these purposes, in this work, a hybrid method, with modeling of Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) deep neural networks, is proposed to ensure the most accurate prediction of such data. The GHSI as well as temperature, relative humidity, and wind speed data obtained in the Jordan valley are used in the forecasting methodology. In the CNN block of the proposed deep architecture, the input parameters are passed through the convolution, pooling, and flattening layers, and the outputs are forwarded to the LSTM data input. With this method, it is aimed to make more effective and accurate estimations. The proposed method has been compared according to Root Mean Square Error (RMSE), Mean Absolute Deviation Error (MADE), and Mean Absolute Percentage Error (MAPE) error performance criteria in order to reveal the difference from other methods. The proposed method produces superior results compared to other algorithms, especially in GHSI estimation.

Keywords- Global Solar Irradiance, Deep Neural Networks, Long Short Term Memory, Convolutional Neural Network

ÖZ

Önemli iklim parametrelerinin yanı sıra küresel yatay güneş ışınımının (GHSI) tahmin edilmesi, fotovoltaik panellerin enerji yönetimi ve kaynak planlamasında önemli bir rol oynamaktadır. Güneş enerjisinden daha fazla yararlanmak için, bu tür zaman serisi parametre verilerinin sıklıkla analiz edilmesi ve tahmin edilmesi yoluyla gelecek değerler hakkında bilgi elde edilmesi gerekmektedir. Bu nedenle, uzun vadeli güneş ışınımı verilerini tahmin etmek zorlu bir iştir. Bu amaçlarla, bu çalışmada, bu tür verilerin en doğru tahminini sağlamak için Uzun Kısa Süreli Bellek (LSTM) ve Evrişimsel Sinir Ağı (CNN) derin sinir ağlarının modellenmesi ile hibrit bir yöntem önerilmiştir. Ürdün vadisinde elde edilen GHSI ve sıcaklık, bağıl nem ve rüzgâr hızı verileri tahmin metodolojisinde kullanılır. Önerilen derin mimarinin CNN bloğunda, giriş parametreleri evrişim, havuzlama ve düzleştirme katmanlarından geçirilir ve çıkışlar LSTM veri girişine iletilir. Bu yöntemle daha etkin ve doğru

[1*]Corresponding author contact: svancin@adiyaman.edu.tr (https://orcid.org/0000-0003-1420-2490)
Department of Computer Engineering, Adıyaman University, Faculty of Engineering, Adıyaman, Turkey
[2]Contact: herdem@adiyaman.edu.tr (https://orcid.org/0000-0003-0079-0041)
Department of Mechanical Engineering, Adıyaman University, Faculty of Engineering, Adıyaman, Turkey

tahminler yapılması hedeflenmektedir. Önerilen yöntem, diğer yöntemlerden farkını ortaya çıkarmak için RMSE, MADE ve MAPE hata performans kriterlerine göre karşılaştırılmıştır. Önerilen yöntem, özellikle GHSI tahmininde diğer algoritmalara göre daha üstün sonuçlar vermektedir.

Anahtar Kelimeler- Küresel Güneş Işınımı, Derin Sinir Ağları, Uzun Kısa Süreli Bellek, Evrişimli Sinir Ağı

I. INTRODUCTION

There are serious undesired effects of climate change to our world and society. Thus, it is important to reach the Paris Climate Agreement goals. Increasing implementation of solar energy to use it for different applications is crucial in order to reach the Paris Climate Agreement goals. Solar energy can be used for various applications including: (1) producing electricity for remote communities [1], (2) rooftop electricity production [2], (3) large-scale power generation, and (4) green-hydrogen production [4].

One of the most critical challenges with regards to the implementation of solar energy usage is its fluctuating nature. Therefore, weather prediction and forecasting of solar radiation are two important applications for effective integration of solar energy into the current energy infrastructure. Artificial Intelligence (AI) techniques are important tools to predict weather and solar radiation [5]. There have been numerous published studies regarding prediction of weather and solar radiation in the recent years.

Scher and Messori implemented deep learning with artificial convolutional neural networks to predict weather forecast uncertainty [6]. They used past weather data to train their machine learning models. Chantry et al. presented an outline related to the advantages, challenges and motivations of applying machine learning techniques to improve prediction of weather and climate [7]. Schultz et al. prepared a paper to discuss whether deep learning approaches can replace existing numerical weather models [8]. Moosavi et al. showed how machine learning techniques help to quantify uncertainty in numerical weather prediction models [9]. Fouilloy et al. applied eleven statistical and machine learning tools to predict hourly solar irradiation [10]. The tools used in [10] are listed as: (1) persistence, (2) smart persistence, (3) auto regressive moving average (ARMA), (4) Multilayer perceptron (MLP), (5) Regression tree (RT), (6) Boosted RT, (7) Bagged RT, (8) Pruned RT, (9) Random Forest, (10) Gaussian process, and (11) Support vector regression (SVR). Yagli et al. evaluated 68 machine learning algorithms for hourly solar forecasting by using three sky conditions, seven locations, and five climate zones in the USA [11]. They used R caret package to train all models. Feng et al. applied and compared various empirical and machine learning models to predict global solar radiation [12]. They used only air temperature as inputs to develop their machine learning and empirical models. Zhou et al. prepared a comprehensive review on global solar radiation prediction with machine learning models [13]. They reviewed 232 papers to cover all important aspects of machine learning models to predict global solar radiation.

Analyzing global horizontal irradiance (GHSI) data based on important weather parameters such as air temperature, relative humidity, and wind speed will be an interesting approach for using photovoltaic solar plants [14,15]. Accordingly, it is necessary to examine the effects and correlation of temperature on GHSI and to put forward energy management plans accordingly [16,17]. Therefore, making long-term predictions in GHSI and other climate data is a challenging task. In this study, we aim to predict climate and GHSI data by using LSTM and CNN deep learning methods together. In this study, we propose a Long Short-Term Memory (LSTM)-CNN deep network method to estimate the climate parameters of temperature, relative humidity, and wind speed with GHSI data. The data used in this study is the data from 2017 obtained in the Jordan Valley. In the CNN block, the input data is passed through layers of convolution, pooling, and flattening. After that, the data is trained, and optimal features are extracted in the LSTM block. Estimation results are created as a result of comparing the output data with the actual data. Based on these estimation data, the proposed method was compared with other methods by RMSE, MADE, and MAPE metrics.

The remainder of the paper is as follows. Section 2 presents the literature review. Section 3 presents the proposed method with constructing the LSTM-CNN based deep neural network. Section 4 presents the experimental results of the study. We conclude the study by mentioning the future plans in Section 5.

II. LITERATURE REVIEW

Recently, meteorological satellites are also used for the optimization of all solar fields. Cano et al. (1987) carried out one of the earliest studies on solar radiation estimations using the pixels of satellite images (cloud index (n)) taken by the weather satellite [18]. The name is the same HELIOSAT Method. This method or its modified versions are widely used by many research teams [19,20] Apart from the global solar radiation, the estimation of diffuse solar radiation components has also been very important for photovoltaic (PV) modules. Rusen et al. (2020) studied to compare and evaluate the validity of global and diffuse estimation methods for nine locations in Turkey. The satellite-based estimation methods (HELIOSAT, Meteonom, and PVGIS) mentioned in the literature were used to estimate global and diffuse radiation. These selected methods were statistically tested with accurate ground measured data. For HELIOSAT, the global radiation is varying −0.02 and 0.03, and 0.010 and 0.035, respectively, Relative MBE and Relative RMSE [21]. In addition, it is known that bright sunshine hours have been measured for more than 100 years all over the world. The estimation of the parameters extraction of the solar models and global solar radiation can be found using this bright sunshine hour and satellite parameters for a long time period. This coupled (hybrid) method approach is a combination of both ground and satellite data [22]. In this method, the satellite images and ground measurements of bright sunshine hours are taken into account to estimate the global solar radiation for photovoltaic (PV) modules. Angstrom models and ground measured daily global solar irradiation by using regressions and error analyses. In a work [23], the current version of the satellite-based HELIOSAT method and ground-based linear Ångström–Prescott type relations are used in combination.

III. PROPOSED METHOD

In this study, we propose an LSTM and CNN based deep learning model to predict weather parameters with GHSI. In the proposed LSTM-CNN model, the number of input neurons and layers, the number of hidden layers and the convolution layer are designed for making predictions of the weather and GHSI parameter data.

Deep Neural Networks (DNNs) are learning networks inspired by the interaction and communication of neurons with layers in the human brain. DNNS are able to process the observed data in the form of digital vectors, including images, audio, text and/or time series [24,25]. DNNs are complex networks consisting of many neurons interacting with one another. A DNN is a network architecture that can operate in parallel, consisting of multiple input, hidden, and output layers [26]. Figure 1 shows a DNN model structure.

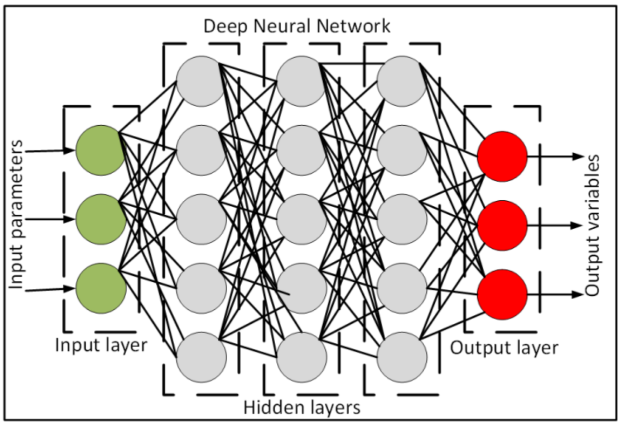

Figure 1. The conventional DNN model structure

Recurrent Neural Networks (RNNs) are developed versions of DNNs, one output of which is passed to the input of a feed-forward network, taking into consideration past values. The past output is calculated and included in the next input creating learning functional data. The output data consists of an iterative series of iterations that facilitate learning without memorization [27-29].

RNNs, like feed-forward DNNs, do not simply process data inputs and transfer them to the next state. At the same time, some components learned in their internal memory are not forgotten [30]. Moreover, LSTM networks are RNNs in which RNNs have been developed and their layers and the way their neurons work are different. In this way, the past learned data in the memory is prevented from being forgotten and makes it easier to remember when necessary. In addition, the disappearing gradient problem is also eliminated. An LSTM is often used to process, classify, and estimate time series data with unknown time delays [31]. There are three gates in the internal structure of an LSTM network: input, forget and output gates [32]. An LSTM model architecture is presented in Figure 2.

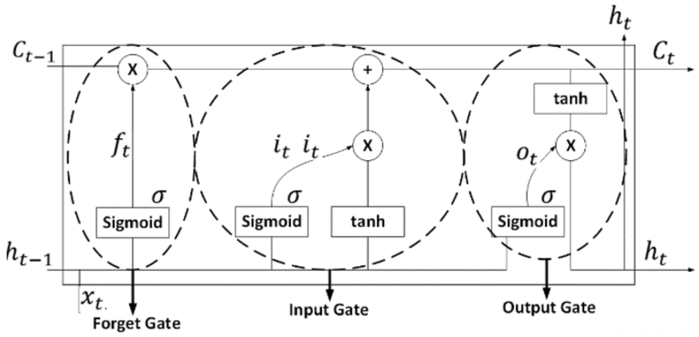

Figure 2. An LSTM architecture block

The input gate determines which values must be given from the input to replace the memory in the LSTM. The input gate i_t is defined as Equation (1).

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i) \tag{1}$$

where x_t is the input at time t, h_{t-1} is the output at time $(t-1)$, W_i and b_i are weight vector and bias of the input gate, respectively Also, sigmoid function σ decides which of 0 and 1 will pass.

In forget gate, it is determined which unlearned data will be forgotten from the block. This decision is made by the sigmoid function. This function evaluates both the previous h_{t-1} state and the content input x_t. The expression of the forget gate f_t is given as Equation (2).

where C_t is the cell state, W_f and b_f are weight vector and bias for forget gate, respectively.

$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f) \tag{2}$$

When determining the output h_t of the output gate o_t in LSTM, input x_t and memory are used to determine the output h_t. The output gate o_t is given as Equation (3).

$$o_t = \sigma(W_o . [h_{t-1}, x_t] + b_o \tag{3}$$

where W_o and b_f are weight vector and bias for output gate, respectively.

$$\hat{C}_t = tanh(W_C . [h_{t-1}, x_t] + b_C) \tag{4}$$

$\hat{C}_t$ is the memory cell state at time t, shown in Equation (4). Here, W_C and b_C are weight vector and bias the memory cell, respectively. Then, $\hat{C}_t$ memory cell returns a number between 0 and 1 for each number in the previous cell state C_{t-1}, as given in Eq.(5).

$$C_t = C_{t-1} * f_t + i_t * \hat{C}_t \tag{5}$$

Also, the $tanh$ function of C_t assigns a weight value to the values passed from -1 to 1 to determine its priority, it is multiplied by the sigmoid output o_t as given in Equation (6).

$$h_t = o_t * tanh(C_t) \tag{6}$$

CNNs are deep feed forward neural networks. The model of a CNN includes a convolution layer, a pool layer, and a fully connected layer [33]. Then the model runs an activation function to address the classification or estimation problem. In the convolution layer, the feature maps of the previous layer are convoluted with a convolution kernel. The resulting output feature maps are processed by an enable function [34]. The formulation of the convolution layer can be represented by Equation (7).

$$x_j^L = f(\sum_{i \in m_j} x_i^{L-1} * k_{ij}^L + b_j^L) \tag{7}$$

where $x_j^L$and x_i^{L-1}denote the jth and ith output feature maps of the Lth and $(L-1)$th layers, respectively. Moreover, m_j denotes the input maps selection, k_{ij}^Ldenotes the weights between the ith input map and the jth output map, $*$ denotes the operation of the convolution, b_j^L denotes the bias factor, and $f()$ represents the rectified linear unit activation function (ReLU) [35,36]. The pooling layer is used to minimize the number of CNN parameters. The pooling layer is determined by computing the average or maximum pooling value of a given area in a feature map. The formulation of the pooling layer is given as Equation (8).

$$x_j^L = f(\partial_j^L down(x_j^{L-1}) + b_j^L) \quad (8)$$

where $down()$ represents the max pooling subsampling function, x_j^{L-1}denotes the jth input feature map of the Lth layer, and b_j^L is the bias factor. As a result, measurements of the convolutional and pooling layers are performed. The fully connected layer then computes the final output vector as given in Equation (9).

$$x^L = f(w^L x^{L-1} + b^L) \quad (9)$$

where x^L represents the final output vector, x^{L-1} represents the input vector, w^L represents the weights between the $(L-1)$th layer and the Lth layer, and b^L represents the bias factor.

Input local connections are defined by convolutional layers. However, the pool layer reduces the computation corresponding to the target variable. Increasing the number of convolutional layers $N_c$uses many cores to generate the same number of invariant feature maps. These operations are executed for data from previous layers. Although the degree of information extraction is deeper, the number of neurons kN_i,i∈N_c in each convolution layer and the size of the kernels kS_i,i∈N_c are important for detail. Therefore, the feature expression with a scale range is more precise. The pooling layer after the convolution layer succeeds in minimizing the input size by subsampling defined by pS_i,i∈N_c. Feature maps are obtained and then integrated into a fully connected network [37].

We can explain our proposed LSTM-CNN based prediction algorithm in 6 steps.

Step (1): Input data: We input the required data for training of LSTM-CNN.

Step (2): Data standardization: A method with *z*-score standardization is applied to standardize the input data. This is required for training the model better. This is shown in the following Equations (10) and (11).

$$y_i = \frac{x_i - x'}{s} \quad (10)$$

$$x_i = y_i * s + x' \quad (11)$$

where y_i is the standardized value, x_i is the input data, x' and s are average and standard deviation of the input data, respectively.

Step (3): Network initialization: We initialize the LSTM-CNN layer’s weights and biases.

Step (4): Calculation of the CNN layer: The input data are operated in the convolution, pooling, and flattening layers in the CNN layer. This way we can execute the input data feature extraction and obtain the output value.

Step (5): LSTM layer configuration: For effective climate parameter estimation, it is required to select appropriate input data. In this study, we design the LSTM layer and the number of neurons adaptive to our study. In the proposed model, 4-$level$ is the LSTM learning model with N-1, N-2, N-3, and N-4 input neurons, and LSTM layers with 10 hidden units. In this case, all four input signals are utilized for training. The output h_t can be expressed as in Equation (12).

$$h_t \leftarrow 4 - L_i\,(x_j) - 1 \quad (12)$$

In the proposed hybrid model, an LSTM model with four layers and ten hidden units is proposed, x_j= 10, where j=1 and L_i= 4, where i =1. In this model, the output node is set to 1. In the input layer, four different input nodes are defined as temperature, relative humidity, wind speed, and GSHI values. While modeling the LSTM network, the number of layers and the number of hidden units are chosen adaptively. After the layers and neurons are determined, the number of layers i is assigned from 1 to k. Next, the i is set to the fixed value where the error metrics are minimized. The neuron number j is assigned from 1 to m for i =1. These operations are also valid for $k-1$ layers.

LSTM block cells eliminate the possibility of memorization in the data that can occur as a result of long-term learning. The number of LSTM cells is defined according to how the proposed model predicts the climate parameters. Batch size is an important parameter that affects the training efficiency of the LSTM neural network, which determines the number of samples to be detected over the deep network before the internal parameters of the LSTM model are updated. The data is divided into groups of several small blocks and at the same time the neural network is trained for multiple stages due to the fact that these networks work faster with mini-groups. On the other hand, with a very small batch size, the parameter value becomes harder to estimate and requires more memory space in the LSTM when a larger batch size is assigned in the network.

Step (6): Output data: At this stage, the estimation of the output data is made, and the error calculation is executed. We analyze the proposed model based on the RMSE, MADE, and MAPE metrics in this work. The RMSE usually measures the differences between samples or all data values forecasted by a model or an estimator and the observed values. The RMSE of forecasted values x'_t for t times a regression's dependent variable x_t with variables observed over T sample times, is computed for T different samples. The RMSE calculation is given as Equation (13). The lower the RMSE value, the more successful the data estimation is expected. Therefore, when the high performance of the proposed models is achieved, the RMSE values should be considerably lower. The closer the RMSE, MADE, and MAPE values are to 0, the better predictive performance will be [37].

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(x_t - x'_t)^2}{T}} \tag{13}$$

MADE prevents the problem where positive and negative errors in estimation dampen each other. The calculation of the MADE is presented as given in Equation (14).

$$MADE = \frac{1}{T}\sum_{t=1}^{T}|x_t - x'_t| \tag{14}$$

MAPE computes the deviation between the predicted and the observed data at the same time, it is based on the correlation between the error and the true value that show the prediction accuracy. The formulation of the MAPE is presented as given in Equation (15).

$$MAPE = \frac{100}{T}\sum_{t=1}^{T}\left|\frac{x_t - x'_t}{x_t}\right| \tag{15}$$

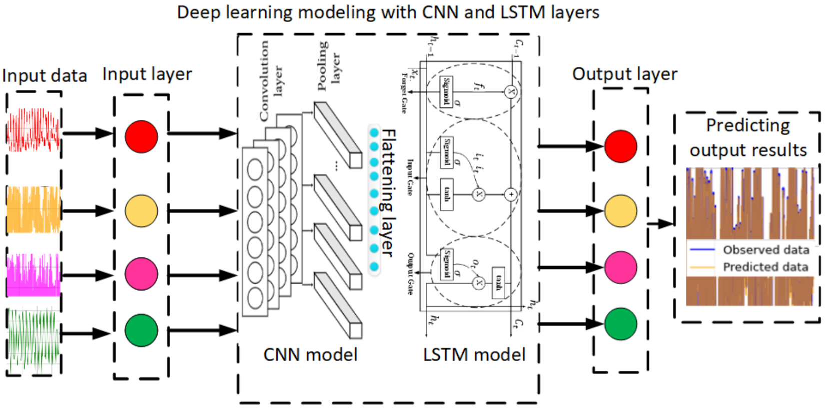

Figure 3. The block diagram of the proposed scheme

Figure 3 shows the block diagram of the proposed scheme. Neurons in the input layer are used for input data for temperature, relative humidity, wind speed, and GHSI. It is imporantant to note that the inpu is rendered from the CNN model and then in the flattening layer before being transferred to the LSTM block. The deep learning model is modeled by a combination of LSTM and CNN architecture, and the data goes through the training and testing steps. Finally, the prediction results are calculated from the output layer.

IV. EXPERIMENTAL RESULTS

To implement the applications of this study, we performed the following installations. This study is closely related to open-source software installations. We utilized the Python software that comes with Anaconda 3 distribution. Also, some Python scripts were compiled via PyCharm IDE.

A. Study Data

The dataset used in this study was taken from reference [38]. All data is publicly accessible, and the interested readers can access the data used in this study from reference [38]. The authors in [38] used calibrated, high accuracy sensors to collect the data. The data includes average ambient temperature (^oC), minimum ambient temperature (^oC), maximum ambient temperature (^oC), average wind speed (m/min), minimum wind speed (m/min), maximum wind speed (m/min), relative humidity (%), and solar irradiance (W/m^2).

We focus on the cities of Umman in Jordan for study area [38]. Figure 4 shows this area of Jordan taken from Google maps.

Figure4. The Jordan Valley from Google Maps

We used 37,610 of these data for training and the remaining 9,403 for testing in long-term forecasting.

B. Installing the Proposed LSTM-CNN Deep Network

We used a function to operate the LSTM-CNN Tensor Flow Keras model with layers. The model uses a neuron for the output layer because we are estimating a real-valued number. The DNN utilizes the ReLU activation function which can take its own value in the range of 0 to maximum value. Also, the proposed model uses the Adam optimizer. Table 1 shows some of the parameters used in the LSTM-CNN based proposed method.

Table 1. Parameters of the LSTM-CNN based proposed method

Parameters	**Value**
Number of hidden units in LSTM layer	64
Activation function of the LSTM layer	tanh
Convolution layer filters	32
Convolution layer kernel size	1
Activation function of the convolution layer	tanh
Convolution layer padding	same
Pooling layer pool size	1
Pooling layer padding	same
Activation function of the pooling layer	ReLu
Time step, embedding size	12
Batch size	16
Learning rate	0.002
Optimizer	Adam
Loss function	Mean absolute error
Number of Epochs	500

In this part of the study, we model the average temperature, relative humidity, wind speed, and GHSI data of Jordan Valley. We used 80% of the total 47,013 data for each climate parameter with GHSI for training and the remaining 20% for testing. We used the first 37,610 data in the data set for training and the remaining 9,403 values for testing.

C. Prediction of Temperature Data

In this section, the temperature data received from the Jordan Valley is analyzed. Figure 5 plots the temperature data for training and testing. We plot the training data in blue and the testing data in red. This distinction is marked by a black line. Temperature values are given in Celsius as in the data set. We understand that the temperature in Umman normally ranges from 5°C to 45°C throughout the year, rarely below 3°C and above 50°C. After running the proposed method, the predicted temperature data model is given in Figure 6. Predictive data model was obtained. Figure 7 shows the comparison of the observed and predicted temperature data. From these results, it is understood that with the proposed CNN-LSTM model, the observed and the predicted values produce very close results. In other words, the estimation error rate is minimal.

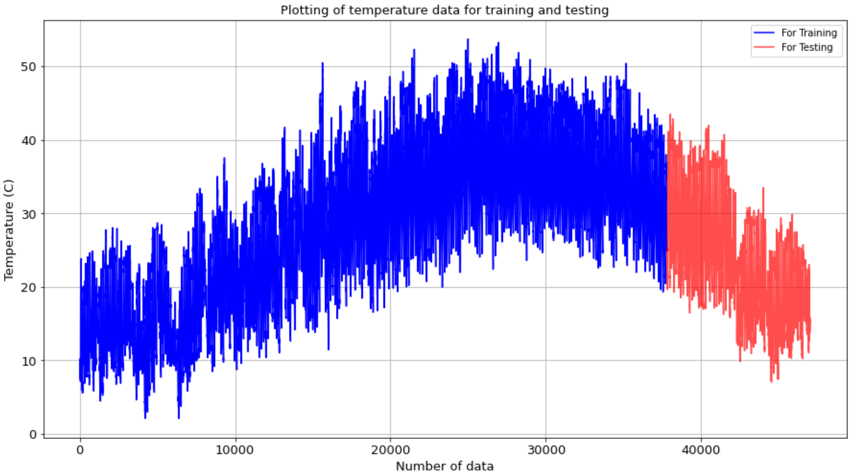

Figure 5. Plotting of temperature data for training and testing

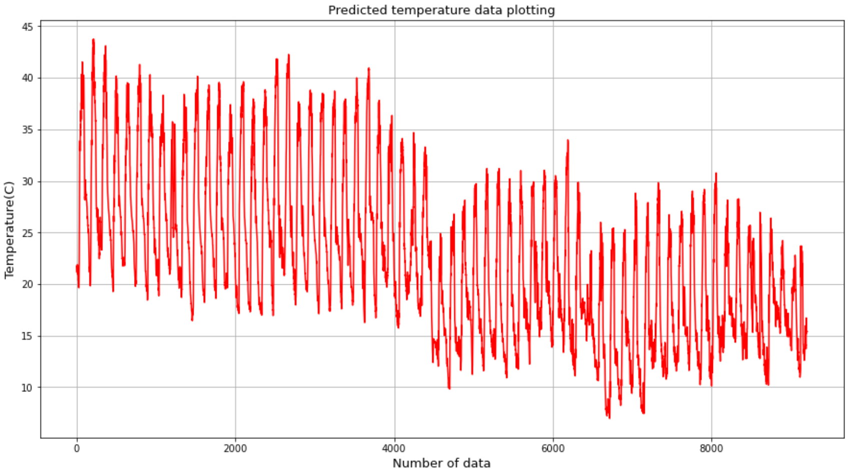

Figure 6. Predicted temperature data plotting

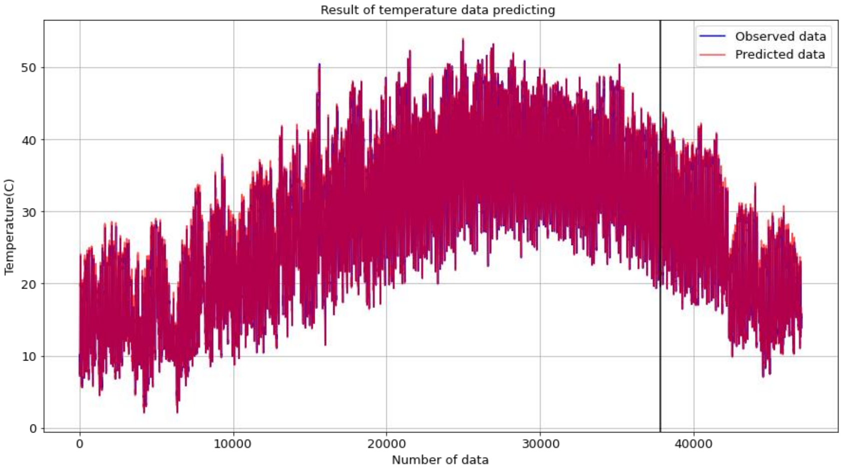

Figure 7. Observed and predicted temperature data comparison

D. Prediction of Relative Humidity Data

In this section, the relative humidity data received from the Jordan Valley is analyzed. Figure 8 shows the relative humidity data for training and testing. We plot the training data in blue and testing data in orange. Humidity values are given in percentage (%) as in the data set. The relative humidity in Umman normally ranges from 20% to 100% throughout the year and rarely falls below 20%. With the proposed method, the predicted humidity data model is given in Figure 9. Figure 10 illustrates the comparison of the observed and predicted relative humidity data. It appears that the forecast data model was well suited to the real data. The accuracy of the estimation is increased by decreasing the error rate. This resulted from the layer and neuron selection structure of the LSTM-CNN based proposed model and the training model.

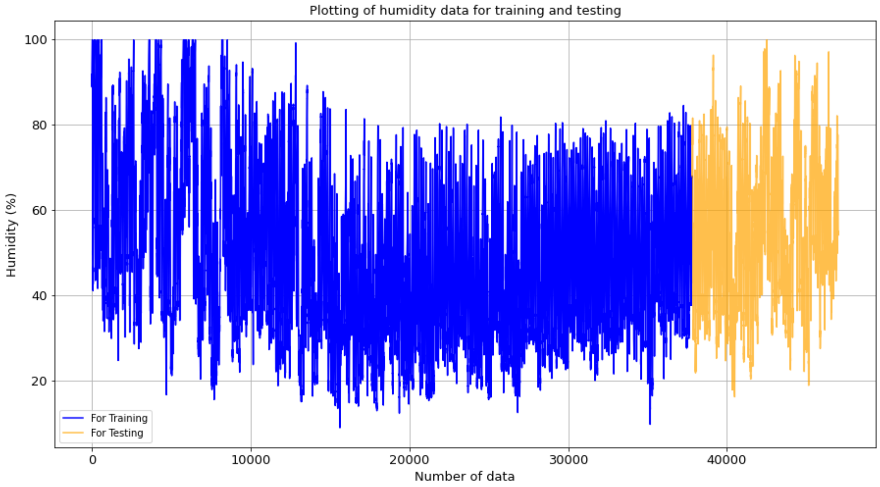

Figure 8. Plotting of humidity data for training and testing

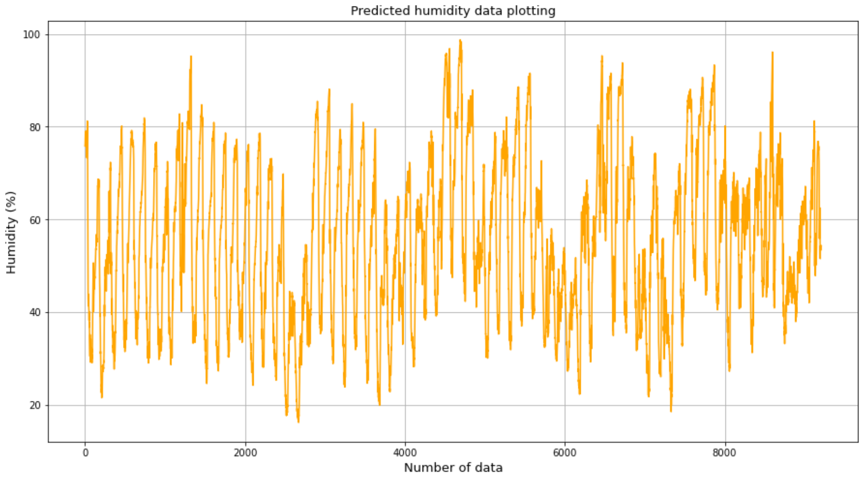

Figure 9. Predicted humidity data plotting

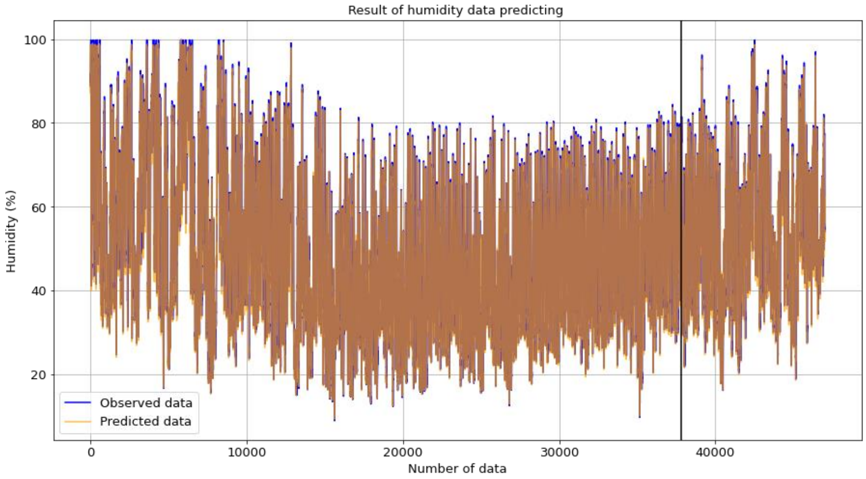

Figure 10. Observed and predicted humidity data comparison

E. Prediction of Wind Speed Data

In this section, the wind speed data received from the Jordan Valley is analyzed. As it is known, wind speed is significantly affected by the air temperature and pressure difference between two points. In the data set, wind speed is recorded in m/s. Figure 11 demonstrates the wind speed data for training and testing. We plot the training data in blue and testing data in magenta. As can be seen in Figure 11, the wind speed generally varies between 0 and 12 m/s throughout the year and rarely rises above 25 m/s. With the proposed method, the predicted wind speed data model is given in Figure 12. Figure 13 illustrates the comparison of the observed and predicted wind speed data. It is critical to note that some values in the prediction model fell below 0. This certainly reflects the errors present. Although there are prediction errors in some data points, it is observed that the estimation of results in general is quite successful.

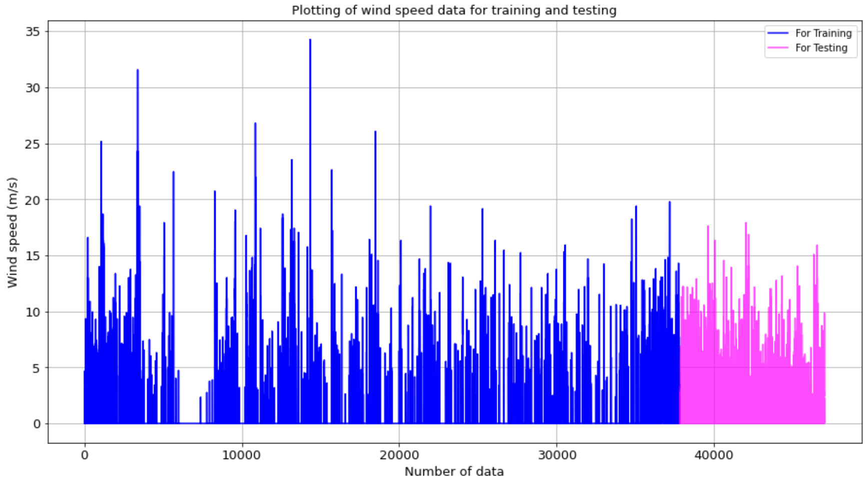

Figure 11. Plotting of wind speed data for training and testing

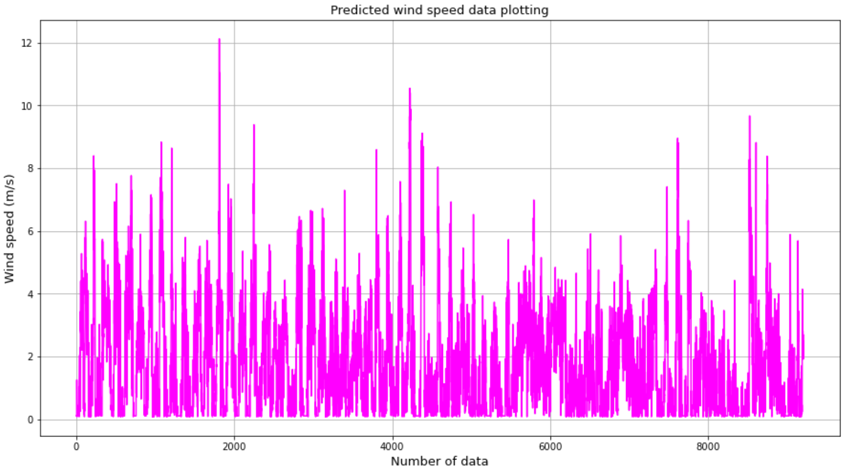

Figure 12. Predicted wind speed data plotting

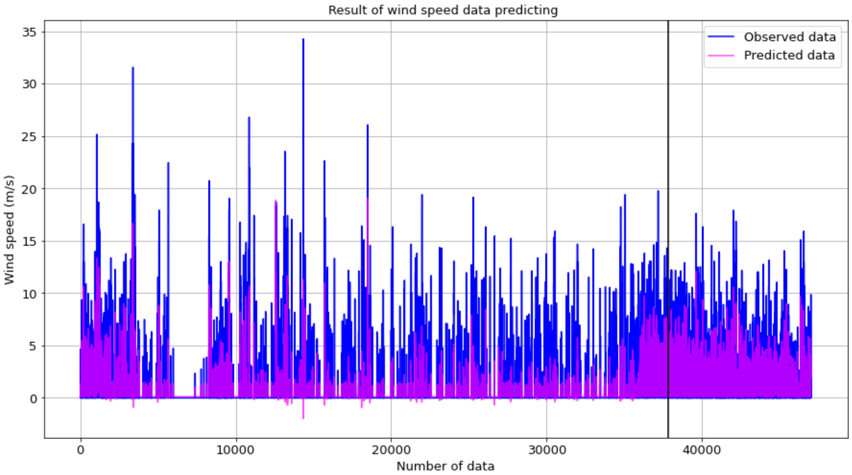

Figure 13. Observed and predicted wind speed data comparison

F. Prediction of Global Horizontal Irradiance Data

The clarity index is defined as the ratio between the values of the spherical radiation at ground level G on the horizontal surface and the $GHSI$ of extraterrestrial spherical radiation [39]. The clarity index consists of the difference between the mean value of the daily maximum temperatures (T_{max}) and the mean value of the daily minimum temperatures (T_{min}) and can be estimated using Equation (16).

$$\frac{G}{GHSI} = K\Delta T^{0.5} \tag{16}$$

where $\Delta T = T_{max} - T_{min}$. The empirical parameter K was fixed at 0.17 for arid and semiarid climates, whereas latterly proposed the use of K = 0.16 and 0.19 for interior and coastal regions, respectively. In the data set, GHSI is recorded in W/m^2. Figure 14 shows the GHSI data for training and testing. We plot the training data in blue and testing data in green. As seen in Figure 14, the GHSI generally varies between 0 and 1040 W/m^2 throughout the year and rarely rising above 1200 W/m^2. With the proposed method, the predicted GHSI data model is given in Figure 15. Figure 16 illustrates the comparison of the observed and predicted GHSI data. The boundary between training and testing splits is denoted by the vertical black line. There are some obvious errors in the model forecasting. These may be corrected with suitable hyperparameter tuning operations.

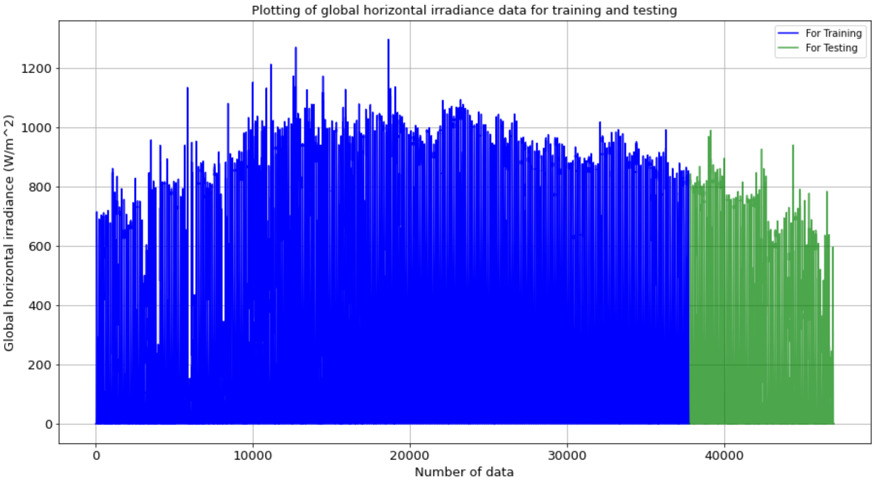

Figure 14. Plotting of the GHSI data for training and testing

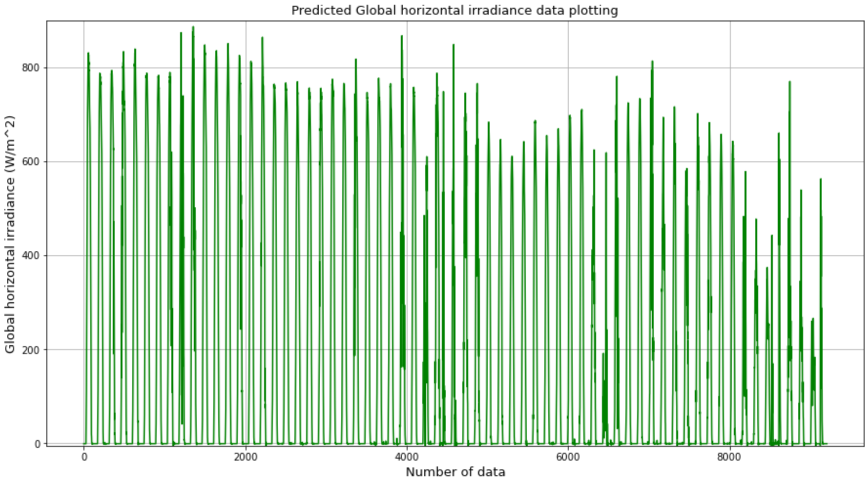

Figure 15. Predicted GHSI data plotting

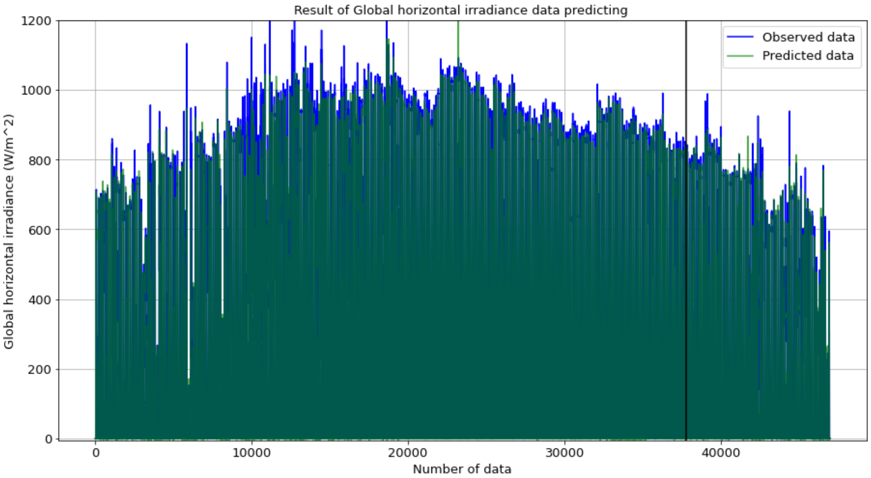

Figure 16. Observed and predicted GHSI data comparison

The proposed LSTM-CNN model has been compared in performance with standard RNN, standard LSTM, and LSTM-GRU models. The algorithmic complexity of the proposed model is slightly higher than other methods due to the CNN layers. However, accurate and timely estimation of parameter values avoids this complexity. Table 2 presents the performance results according to RMSE, MADE, and MAPE values for all climatic parameters of Umman in Jordan. From Table 2, we observed that with the proposed LSTM-CNN model, the RMSE, MADE, and MAPE values for temperature are 0.87, 0.65, and 0.17, for relative humidity 1.38, 0.93, and 0.47, for wind speed 3.34,1.59, and 0.84, and for GHSI 0.61, 0.34, and 0.02, respectively. That is, when compared to other methods, minimal errors were achieved with the proposed methodology. It is important to note that as the number of epochs increases, the RMSE and other metrics decrease and the prediction performance improves. However, it is clear that further refinement and optimization are required for this climate parameter data.

Finally, the RMSE, MADE, and MAPE values obtained are lower than the other LSTM and RNN derivatives compared, and when deep network algorithms are used, it shows that there is a minimal difference between the observed values and the predicted values when estimating the climate parameters with GHSI variables. From these results, we understood that high achievement is shown in climate parameter and GHSI estimations. The reason why the results are successful and give the least errors is that we designed the LSTM-CNN network modeling in accordance with current neural network technologies for the proposed approach.

Table 2. Comparison of the error values in each climate parameter

Weather parameters with global horizontal irradiance variables	**Error metrics**	**Models**			
		Standard RNN	**Standard LSTM**	**LSTM-GRU**	**LSTM-CNN based proposed method**
Temperature	RMSE	6.45	4.82	3.38	**0.87**
	MADE	4.37	2.42	1.87	**0.65**
	MAPE	2.32	1.07	0.82	**0.17**
Relative humidity	RMSE	8.42	5.28	4.28	**1.38**
	MADE	7.65	3.84	2.27	**0.93**
	MAPE	3.53	1.71	1.16	**0.47**
Wind speed	RMSE	11.48	8.51	5.64	**3.34**
	MADE	8.34	4.62	2.13	**1.59**
	MAPE	3.85	2.38	1.28	**0.84**
Global horizontal solar irradiance (GHSI)	RMSE	5.12	4.62	3.14	**0.61**
	MADE	4.38	2.29	1.53	**0.34**
	MAPE	2.23	0.85	0.54	**0.02**

V. CONCLUSION

In this paper, we propose an LSTM-CNN deep learning framework to predict the GHSI with weather hyperparameters. On the CNN side of the proposed model, the input data is processed with neurons in convolution, pooling, and flattening layers, respectively. After that, the training data is presented with LSTM input weight values. After passing through unique training and testing stages with LSTM input, forget and output layers, the inputs are transferred to the output layer. Estimation results are published by comparing the predicted values obtained with the calculated data. The proposed algorithm has been implemented in Python coding language and has been compared with existing LSTM and CNN based algorithms according to their RMSE, MADE, and MAPE performances. As can be seen from the results, the proposed method demonstrates the best performance with the least amount of error rates. In future studies, we can propose deep learning models that can further reduce the error rate for GHSI and energy battery level estimation.

REFERENCES

[1] Arriaga, M., Cañizares, C. A., & Kazerani, M. (2014). Northern lights: Access to electricity in Canada's northern and remote communities. *IEEE Power and Energy Magazine*, 12(4), 50-59.

[2] Assouline, D., Mohajeri, N., & Scartezzini, J. L. (2018). Large-scale rooftop solar photovoltaic technical potential estimation using Random Forests. *Applied energy*, 217, 189-211.

[3] Cabrera, P., Carta, J. A., Lund, H., & Thellufsen, J. Z. (2021). Large-scale optimal integration of wind and solar photovoltaic power in water-energy systems on islands. *Energy Conversion and Management*, 235, 113982.

[4] Herdem, M. S., Mazzeo, D., Matera, N., Wen, J. Z., Nathwani, J., & Hong, Z. (2020). Simulation and modeling of a combined biomass gasification-solar photovoltaic hydrogen production system for methanol synthesis via carbon dioxide hydrogenation. *Energy Conversion and Management*, 219, 113045.

[5] Ahmad, T., Zhang, D., Huang, C., Zhang, H., Dai, N., Song, Y., & Chen, H. (2021). Artificial intelligence in sustainable energy industry: Status Quo, challenges and opportunities. *Journal of Cleaner Production*, 125834.

[6] Scher, S., & Messori, G. (2018). Predicting weather forecast uncertainty with machine learning. *Quarterly Journal of the Royal Meteorological Society*, 144(717), 2830-2841.

[7] Chantry, M., Christensen, H., Dueben, P., & Palmer, T. (2021). Opportunities and challenges for machine learning in weather and climate modelling: hard, medium and soft AI.

[8] Schultz, M. G., Betancourt, C., Gong, B., Kleinert, F., Langguth, M., Leufen, L. H., ... & Stadtler, S. (2021). Can deep learning beat numerical weather prediction?. *Philosophical Transactions of the Royal Society A*, 379(2194), 20200097.

[9] Moosavi, A., Rao, V., & Sandu, A. (2021). Machine learning based algorithms for uncertainty quantification in numerical weather prediction models. *Journal of Computational Science*, 50, 101295.

[10] Fouilloy, A., Voyant, C., Notton, G., Motte, F., Paoli, C., Nivet, M. L., ... & Duchaud, J. L. (2018). Solar irradiation prediction with machine learning: Forecasting models selection method depending on weather variability. *Energy*, 165, 620-629.

[11] Yagli, G. M., Yang, D., & Srinivasan, D. (2019). Automatic hourly solar forecasting using machine learning models. *Renewable and Sustainable Energy Reviews*, 105, 487-498.

[12] Feng, Y., Gong, D., Zhang, Q., Jiang, S., Zhao, L., & Cui, N. (2019). Evaluation of temperature-based machine learning and empirical models for predicting daily global solar radiation. *Energy Conversion and Management*, 198, 111780.

[13] Zhou, Y., Liu, Y., Wang, D., Liu, X., & Wang, Y. (2021). A review on global solar radiation prediction with machine learning models in a comprehensive perspective. *Energy Conversion and Management*, 235, 113960.

[14] Prieto, J.I., Martínez-García, J.C., & García, D. (2009). Correlation between global solar irradiation and air temperature in Asturias, Spain, *Solar Energy*, 83(7),1076-1085.

[15] Malakar, S., Goswami, S., Ganguli, B. et al. (2021). Designing a long short-term network for short-term forecasting of global horizontal irradiance. *SN Applied Sciences*, 3, 477.

[16] Jalali, S. M. J., Ahmadian, S., Kavousi-Fard, A., Khosravi, A. & Nahavandi, S. (2021). Automated Deep CNN-LSTM Architecture Design for Solar Irradiance Forecasting, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 1-12, 10.1109/TSMC.2021.3093519.

[17] Zang, H., Liu, L., Sun, L., Cheng, L., Wei, Z., & Sun. G. (2020). Short-term global horizontal irradiance forecasting based on a hybrid CNN-LSTM model with spatiotemporal correlations, *Renewable Energy*, 160, 26-41.

[18] Cano, D. et al. (1987). A method for the determination of the global solar radiation from meteorological satellites data. *Solar Energy, Elsevier*, 37(1), 31-39.

[19] Rusen, S.E. (2018). Modeling and Analysis of Global and Diffuse Solar Irradiation Components Using the Satellite Estimation Method of HELIOSAT, *CMES-Computer Modeling in Engineering & Sciences*, 115 (3), 327-343.

[20] Rusen, S.E. (2018). Performance evaluation of a coupled method for the estimation of daily global solar radiation on a horizontal surface, *Atmósfera*, 31(4), 347-354.

[21] Rusen S.E. & Konuralp, A. (2020). Quality control of diffuse solar radiation component with satellite-based estimation methods, *Renewable Energy, Elsevier*, 145(C), 1772-1779.

[22] Rusen S.E., Hammer A. & Akinoglu B.G. (2013). Coupling satellite images with surface measurements of bright sunshine hours to estimate daily solar irradiation on horizontal surface, *Renewable Energy, Elsevier*, 55(C), 212-219.

[23] Rusen S.E., Hammer A. & Akinoglu B.G. (2013). Estimation of daily global solar irradiation by coupling ground measurements of bright sunshine hours to satellite imagery, *Energy, Elsevier*, 58(C), 417-425.

[24] Sarker, I.H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, 2, 420.

[25] Greff, K., Srivastava, R. K., Koutnık, J., Steunebrink, B.R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey, *Transactions on Neural Networks and learning systems*, 1-12.

[26] Kim, H., Ham, Y. G., Joo, Y. S. & Son, S. W. (2021). Deep learning for bias correction of MJO prediction. *Nature Communications*, 12, 3087.

[27] Hanab, J. M., Ang, Y. Q. Malkawi, A., & Samuelson, H. W. (2021). Using recurrent neural networks for localized weather prediction with combined use of public airport data and on-site measurements, *Building and Environment*, 192, 107601.

[28] Wang, K., Ma, C., Qiaoa, Y., Lua, X., Hao, W., & Dong, S. (2021). A hybrid deep learning model with 1DCNN-LSTM-Attention networks for short-term traffic flow prediction, *Physica A: Statistical Mechanics and its Applications*, 583, 126293.

[29] Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M.A. (2020). Multi-Sequence LSTM-RNN Deep Learning and Metaheuristics for Electric Load Forecasting, *Energies*, 13, 391.

[30] Hoang, D. T., Yang, Pr. L., Cuong, L. D. P., Trung, P. D., Tu, N. H., Truong, L. V. , Hien, T. T., & Nha, V. T. (2020). Weather prediction based on LSTM model implemented AWS Machine Learning Platform. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, 8(5), 283-290.

[31] Pei, J., Deng, L., Song, S., et al. (2019). Towards artificial general intelligence with hybrid tianjic chip architecture, *Nature*, 572 (7767), 106-111.

[32] Gundu, V., & Simon, S. P. (2021). PSO–LSTM for short term forecast of heterogeneous time series electricity price signals, *Journal of Ambient Intelligence and Humanized Computing*, 12, 2375–2385.

[33] Liu, W., Wang, Z., Zeng, N., Alsaadi, F. E., & Liu, X. (2021). A PSO-based deep learning approach to classifying patients from emergency departments, *International Journal of Machine Learning and Cybernetics*,12, 1939–1948.

[34] Shao, B., Li, M., Zhao, Y. & Bian, G. (2019). Nickel Price Forecast Based on the LSTM Neural Network Optimized by the Improved PSO Algorithm, *Mathematical Problems in Engineering*, Article ID 1934796, 15 pages.

[35] Ju, Y. Sun, G.Y. Chen, Q.H. Zhang, M. Zhu, H.X. & Rehman, M.U. (2019). A model combining convolutional neural network and Light GBM algorithm for ultrashort-term wind power forecasting, *IEEE Access* 7 28309e28318.

[36] Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-Based Model to Forecast Stock Prices. *Complexity*, vol. 2020, Article ID 6622927, 10 pages.

[37] Gensler, A., Henze, J., Sick, B. & Raabe, N. (2016). Deep learning for solar power forecasting—An approach using autoencoder and LSTM neural networks, in Proc. *IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Budapest, Hungary, 2016, 2858–2865.

[38] Dalalaa, Z., Al-Addous, M., Alawneha, F. & Class, C.B. (2020). Environmental data set for the design and analysis of the Photovoltaic system in the Jordan Valley, *Data in Brief*, 31, 105794.

[39] Richardson, C.W. (1985). Weather simulation for crop management models. *Trans. ASAE*, 28(5), 1602–1606.