Solar radiation forecasting using attention-based temporal convolutional network

Damilola OLAWOYIN-YUSSUF[1,a], Mohamad MOHANDES[1,b*], Bo LIU[1,c], and Shafiqur REHMAN[1,d]

[1]Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Academic Belt Road, Dhahran, 31261, Eastern Region, Saudi Arabia

[a]g202110610@kfupm.edu.sa, [b]mohandes@kfupm.edu.sa, [c]boliu@kfupm.edu.sa, [d]srehman@kfupm.edu.sa

Keywords: Global Solar Radiation, Temporal Convolutional Network (TCN), LSTM

Abstract. Solar energy, an inexhaustible and pristine power source, harbors the capability to mitigate the emissions of greenhouse gases and the dependency on fossil fuels, thereby playing a pivotal role in the conservation of our ecosystem. Nevertheless, the process of harnessing solar energy from sunlight is subject to the capricious characteristics of weather conditions, which include variables such as the density of cloud cover, levels of atmospheric moisture, and fluctuations in temperature. Hence, the task of prognosticating solar radiation holds significant importance for the strategic planning and efficient management of solar power systems. The current machine-learning methods for predicting global solar radiation make use of recurrent networks. One major downside of recurrent-based models is that they are exposed to vanishing gradients and stagnant performance over longer available input sequences. The model showcased is an attention-fueled Temporal Convolutional Network (TCN) intertwined with Convolutional Neural Network (CNN). The suggested method merges the advantages of the feature extraction proficiencies of a TCN and the aggregation capabilities of a CNN. The method has been tested for up to 24 hours of future time sequence prediction and it has been noted that its performance is unmatched.

Introduction

Owing to the unpredictable characteristics of weather conditions, the production of solar energy harnessed from sunlight cannot be pre-established with absolute certainty. Many methods for predicting solar radiation have been employed because it is critical in many industries, including solar energy production, agriculture, and weather forecasting[1]. Accurate predictions of solar radiation help these industries to plan and make informed decisions. Historically, the prediction of solar radiation was reliant on physical models that considered a multitude of factors. These included the position of the sun, the angle of incidence, the extent of cloud cover, and the prevailing atmospheric conditions [1]. These models require significant expertise and resources to develop, and the quality and availability of input data can limit their accuracy. Furthermore, they may be unable to capture the complex relationships between input variables and output predictions [1]. In recent years, there has been a growing interest in using machine learning (ML) algorithms to improve solar radiation prediction [2]. ML algorithms can model complex relationships between input and output by learning from the provided data. An ML technique that has an effective handle on getting better performance with big data is deep learning because of its ability to incorporate large datasets and interpret complicated correlations between variables in a versatile, trainable way [3]. Deep learning techniques like multi-layered perceptron [4], convolutional neural networks [5], and recurrent neural network [6, 7] have been used to explore predicting global solar radiation. While these methods have no doubt produced encouraging results, they do so with some challenges. Firstly, some of the methods used in predicting solar radiation fail to provide an estimation over multiple time steps. Multiple-time step prediction provides the advantage of being used in early warning applications and predictive planning. Secondly, although other models based on recurrent networks like Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and gated recurrent units (GRUs) have the capability of storing significant information and prediction over multiple time steps, they do suffer from degradation when making predictions over multiple time steps.

In response to these challenges, a multi-time step prediction architecture is proposed using a two-stage approach. The first stage extracts feature from the input dataset by using an attention mechanism with a temporal convolutional network (TCN) backbone. The first stage acts as an encoder as it outputs a fixed-length representation of the input. The second stage acts as a decoder using the fixed length representation to predict the desired output. It does so by leveraging a CNN and a dense layer to forecast solar radiation for multiple time steps.

Related Work

The application of ML techniques has brought about significant advancements in the study of solar radiation [8]. ML, a smaller category under artificial intelligence, revolves around the principle of instructing algorithms to identify patterns and formulate predictions predicated on data. These algorithms, with their ability to learn and adapt, offer a more efficient and precise approach to data analysis compared to traditional methods. [9] proposed a novel data preprocessing approach that aims to reduce forecasting errors, which are often associated with traditional prediction methods such as Markov chains or k-Nearest Neighbors (KNN). They engineered an enhanced multi-layer perceptron (MLP) model, incorporating three neurons within the concealed layer. This model demonstrated the capacity to yield predictions that were on par, if not surpassing, those generated by techniques such as Bayesian inference, Markov chains, and the KNN algorithm. Xing et al. introduced an innovative hybrid stack autoencoder LSTM (SAELSTM) architecture, specifically de-signed for predicting daily global solar radiation (GSR) [10]. This architecture harnesses the power of deep learning and incorporates a feature selection technique grounded on Manta Ray Foraging Optimization (MRFO). The utilization of this architecture in the context of GSR forecasting is further elaborated in the work of Ghimire et al. [11]. The deep learning hybrid SAELSTM model outperformed other models and persistence methods in simulations in terms of accuracy. The model generates intervals for high-quality solar energy predictions with a high likelihood of coverage and minimal interval errors. The study found that deep learning models, such as Bidirectional LSTM [12], perform better than traditional ML for forecasting daily GSR models. In a study conducted by Alizamir et al., wavelet transformation was utilized to break down different meteorological parameters to predict daily solar radiation [7]. The decomposed signals were then used as input into an LSTM recurrent network. While this approach improved network performance, it also increased the number of input parameters needed, thereby increasing the complexity of the optimization process. In another study, [13] employed CNN and an amalgamation of CNN and LSTM to predict monthly radiation at multiple steps. The study inferred that CNN outperformed other models such as MLP, LSTM, GRU, and CNN-LSTM. However, it's important to note that the receptive fields of CNN do not consider the sequence progression of time series data, which could limit its effectiveness in certain applications. Ghimire et al. predicted solar radiation by selecting features using a random forest recursive feature elimination [5]. The convolutional neural network extracted features which were then fed as input to a multilayer perceptron to generate a predicted output. However, using a multilayer perceptron for prediction limited the model's capability of predicting global solar radiation over multiple time steps. This highlights the need for models that can effectively handle time series data and make accurate predictions over multiple time steps.

Methodology

Symbol Definitions and Issue Formulation:

Consider an exogenous series, denoted as $X = (X_1,\ X_2, \dots ,\ X_n)$ and $X \in \mathbb{R}^{T\times n}$ where n represents the number of features and T signifies the time steps. The $i - th$ exogenous series, expressed in terms of time steps, can be represented as $X_i = (X_i^1,\ X_i^2, \dots ,\ X_i^T)$ or $X_i \in \mathbb{R}^T$. The objective of a time series prediction network is to train a function that, given a specific set of previous time series features $X = (X_1,\ X_2, \dots ,\ X_n)$ and their corresponding outputs within that time steps $\hat{Y} = \hat{y}^{T+1}, \hat{y}^{T+2}, \dots, \hat{y}^{T+k}$ where $Y \in \mathbb{R}^k$. This can be mathematically expressed as:

$$\hat{y}^{T+1}, \hat{y}^{T+2}, \dots, \hat{y}^{T+k} = F(X^1, X^2, \dots, X^T, Y) \tag{1}$$

In this equation, the function $F(.)$ is the function whose parameters are learnable. This means that the function can adapt and improve its performance based on the data it is trained on, thereby enhancing the accuracy of the predictions it makes.

Model:

The design of the proposed model as outlined in Figure 1 takes a series of driving input sequences as its input. These sequences are then passed through an LSTM block, which acts like a translator, converting the input sequences into a form that the model can understand better; this is known as embedding. The decision to use recurrent layers for extracting embeddings was inspired by the work of Gugulothu et al., where GRUs were utilized to generate embeddings for decoding multiple sequences in a multivariate time series network [14].

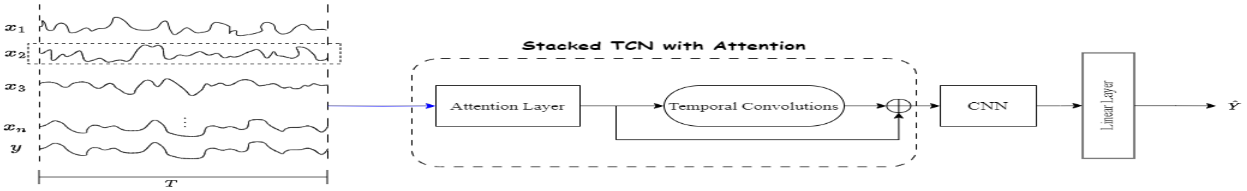

Figure 1 Graphical illustration of the proposed model architecture. It consists of an input layer (the blue line leading to the attention block signifies an LSTM layer used for embedding input), a stacked attention based TCN, a CNN block for merging the TCN output, and finally a linear output layer.

The output gate of an LSTM is expressed as:

$$o_n = F_{emb}(.) = \sigma(W_o * X_n + U_o * h_{n-1} + b_o) \tag{2}$$

X_n: the input vector at time n

h_n: the hidden state vector at time n

W : input-to-hidden weight matrix

b: hidden layer bias vector

σ: sigmoid activation functions.

From equation 2, the output o_n gives the temporal input embedding. The temporal input can be rewritten as:

$$x_{emb,i} = F_{emb}(X_i^1, \quad X_i^2, \dots , \quad X_i^T) \tag{3}$$

The embedded input is then fed into a feature extraction network. This network is made up of an attention block stacked on top of a temporal convolution network.

Attention Block:

The Attention Block [15], works by first obtaining an attention weight vector from the provided input. This input could be represented as $x_{emb,i} = F_{emb}\left(X_{emb,i}^1,\ X_{emb,i}^2, \dots ,\ X_{emb,i}^T\right)$. The attention block helps the model focus on the most important parts of the input. It's postulated that the embeddings possess the same dimensional attributes as the input, albeit this is typically not the scenario. The attention weight vectors are calculated using the following equations:

$$u_i = W_u^T x_{emb,i} + b_u \tag{4}$$

Where $W_u \in \mathbb{R}^{T\times 1}$, and $b_u \in \mathbb{R}$ are parameters to be learned. These attention weight vectors are then normalized using a SoftMax function to ensure they all sum to unity. The normalization SoftMax functions can be expressed as:

$$\mu_i^t = \frac{\exp(u_i^t)}{\sum_{t=1}^{T} \exp(u_i^t)} \tag{5}$$

where $t \in [1, T]$. These normalized softmax values represent the distribution of the input that should be paid attention to. The attention output, defined by the function F_{att} $(\cdot)$, can be calculated by multiplying the normalized SoftMax by the input:

$$x_{att,i} = \mu_i^t \cdot x_{emb,i} \tag{6}$$

Extracting Features with TCN:

In sequence modelling, recurrent networks such as RNN and its variants have been traditionally used until Bai et al. introduced the concept of using generic convolutional networks for sequence modelling tasks. This new approach, called Temporal Convolutional Networks (TCN), outperformed the LSTM [16]. The structure of the TCN is a simple modification of the conventional CNN. The TCN uses causal convolutions, which ensure that the model's prediction at a given time does not depend on future values of the input and makes them much faster to train compared to recurrent models as they do not have recurrent connections. A further enhancement of the causal convolution, known as dilated causal convolution, allows convolution over a wider window by skipping some input values. The receptive field of the dilated causal convolution is much wider than that of the causal convolution, making it more efficient. Figure 2 shows the dilated causal convolutions for different levels of dilation. Given the input $x \in \mathbb{R}^T$ and a filter f : {0, . . . , α, . . . , m − 1} of size m, the dilation convolution operator on α within the sequence can be defined as:

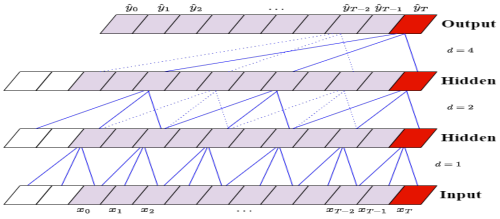

Figure 2 TCN Architecture

$$F_{TCN}(\alpha) = \sum_{j=0}^{m-1} f(j) \cdot x^{(\alpha - d\cdot j)} \tag{7}$$

Where d is the dilation factor, and $\alpha - d \cdot j$ explains the orientation of the past.

The Temporal Convolutional Network (TCN) block, as depicted in figure 1, is a composite of several components. These encompass a dilated causal convolution, weight standardization, a Rectified Linear Unit (ReLU) activation function, and dropout strata incorporated to augment the resilience of the network. The extent of the TCNs could potentially cause the vanishing gradient problem. This is a difficulty encountered during the training of artificial neural networks with gradient-based learning methods and backpropagation. To mitigate this, a skip or residual connection has been incorporated. In a residual block, as described by He et al. [17], there's a pathway that leads us through a series of transformations, denoted as F_o. The results of these transformations are then seamlessly integrated with the block's original input, x. The output after a residual connection, O_{res}, is given by the equation:

$$O_{res} = \sigma\big(x + F_o(x)\big) \tag{8}$$

Here, σ represents the activation function. This function introduces non-linearity into the output of a neuron. This non-linearity helps the network learn from the error so that the model can classify inputs that are not linearly separable.

Aggregating extracted features with CNN:

The output from the stack of attention TCN block is of the form $\mathbb{R}^{N\times T\times L}$ where N is the batch size, T is the number of the input sequence, and L is the number of channels in each TCN layer. There exists a requirement to transform the output of this Temporal Convolutional Network (TCN) into a vector of dimension $\mathbb{R}^{N \times 1 \times K}$, where K signifies the number of anticipated output sequences. To achieve an output of this dimension two different approaches can be followed: flattening the last two dimensions and using a linear layer, or a convolutional layer. In this study, the path of using a convolutional layer was followed because of the added advantage of reducing the number of parameters needed for computation. If a linear layer was used, the number of parameters required would be of dimension $T \times L \times K$ while when a 1-dimensional convolutional layer is used the number of weights required would be $(L \times m) + (L \times K) = L \times (m + K)$, where m is the size of kernel used in the convolutional layer. The $L \times K$ accounts for the number of weights needed in the linear layer for the output to be in the right dimension after the convolutional layer. The convolution operation over an input X can be expressed mathematically as:

$$O_{conv} = \sigma_{lr}(W_{conv}X + B_{conv}) \tag{9}$$

where σ_{lr} is the leaky rectified linear unit (leakyReLU) activation function, W_{conv} is the convolutional weight, and B_{conv} is the convolutional bias. The dense layer utilized after the CNN is an affine amalgamation of the output after the convolution layer and a certain bias, devoid of an activation layer.

Experimental data description, training settings, and evaluation metrics

The empirical dataset employed in this investigation, amassed in the northern region of Saudi Arabia, comprises 75133 data entries encapsulating variables such as solar radiation, atmospheric temperature, relative humidity, velocity and direction of wind, and precipitation, extending over a period from 2012 to 2021. However, it's important to note that there are gaps in the dataset between October 3rd, 2019, and January 1st, 2020, as well as between March 29th, 2020, and June 4th, 2020. Given the sequential nature of the dataset, the data was not shuffled when loaded and was divided into training, validation, and testing portions. The split was done in a 70:15:15 ratio, ensuring a substantial amount of data for each phase of the model development and evaluation process.

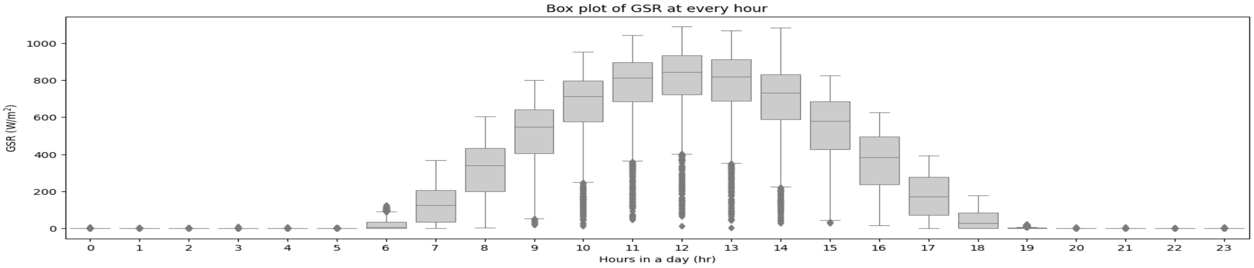

Figure 3 Box plots of GSR at every hour of the day.

In this study, we considered the lags of Global Solar Radiation (GSR) as input to the model, to envisage the next one hour, six hours, 12 hours, and 24 hours respectively. The lags used are 24, 48, and 72 hours respectively, or in mathematical notation as T ∈ {24, 48, 72}. To understand the effect of the time of day on global solar radiation, a box plot showing the distribution of GSR at every hour of the day is presented in Figure 3. The time of day significantly affects global solar radiation, with some outliers observed. A comparison of the maximum values of the variables in our dataset reveals the need for input scaling. Various scaling methods like the standard scaler, robust scaler, and min-max scaler have been used in different studies [18]. The best scaling method for a dataset depends on the variable distribution in that dataset, which can be visualized using box plots. The robust scaler, which uses the interquartile range and median, is used for this problem. The robust scaler calculates the scaled value of an item V in a series of inputs as:

$$V_{scaled} = \frac{V_{original} - \tilde{\eta}}{IQR} \tag{10}$$

where the scaled value is V_{scaled}, the original value is $V_{original}$, $\tilde{\eta}$ represents the input median, and IQR is the input's interquartile range.

The forecast was made for the upcoming 1, 6, 12, and 24 hours, focusing on the variable k in equation 1, where k can be any of the values 1, 6, 12, or 24. This was done to test the theory of Temporal Convolutional Network (TCN) structure. The model's hidden size was determined to be 64 through a grid search, and this size was also used for the RNN and LSTM models. The model structure consists of 1 CNN layer and 6 TCN blocks each having a dropout rate of 0.2. An Adam optimizer with weight decay of 1×10^{-6} was used with a learning rate scheduler having a patience of 2, and a threshold of 0.01. All models underwent training under homogeneous conditions, commencing with a learning rate of 1×10^{-3} and persisting for 40 epochs within a Google Colab environment, utilizing the Torch library of Python-3. A v100 GPU with a 15GB RAM infrastructure was used for the entire experiment. To evaluate the models, three widely used metrics for time series prediction were employed: R-squared, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). Lower values of RMSE and MAE and a higher R-squared indicate a better model.

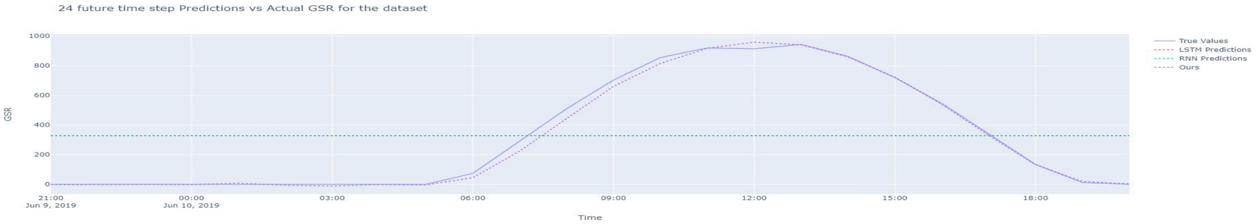

Figure 4 Prediction results for 24 future time steps.

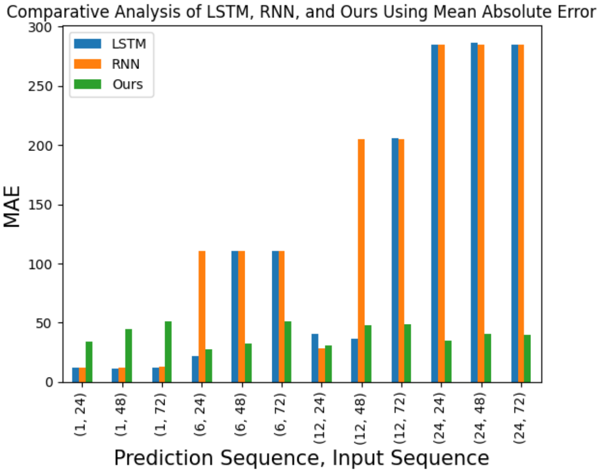

Figure 5 MAE Comparison

Results

The outcomes of applying the proposed model to predict global solar radiation over several time steps are detailed in Table 1 and Figure 5. It's plausible that employing exclusively recurrent architectures, such as LSTM and RNN, might culminate in optimal outcomes for a solitary temporal increment. However, it's also arguable that the model introduced here can confidently compete at that single time step. When predictions are made over multiple time steps like that shown in Figures 4, and regardless of the amount of historical time sequence data used as input, the performance of the recurrent models (i.e., LSTM and RNN) declines significantly. The extent to which they are less accurate compared to the proposed model is quite substantial. There were some unexpected performances when the RNN was predicting 12 future time steps, and the LSTM was predicting 6-time steps. Even under these conditions, the proposed model achieved performance measures close to the best possible outcomes. The consistent performance of this model, either close to the best or the best, attests to its balance and robustness.

Table 1. Juxtaposition of Different Methods with 1 and 6 Prediction Steps.

	Pred. Steps	1			6			12			24		
Mthds	Inp. Seq.	24	48	72	24	48	72	24	48	72	24	48	72
	Metric												
LSTM	RMSE	35.765	39.639	38.988	**55.651**	158.812	159.395	84.145	89.088	259.090	323.678	323.688	323.576
	MAE	**11.719**	**11.400**	**11.699**	**21.768**	110.315	110.644	40.773	**36.344**	206.024	284.396	286.481	284.912
	R-Sq.	0.988	0.985	0.986	**0.971**	0.735	0.733	0.932	0.926	0.345	0.032	0.031	0.032
RNN	RMSE	**30.437**	**32.067**	**33.161**	159.285	159.385	159.293	**67.259**	258.887	258.753	323.627	323.554	323.531
	MAE	12.195	12.0160	12.911	110.162	110.404	110.693	**27.966**	204.729	205.334	284.612	284.856	284.684
	R-Sq.	**0.991**	**0.990**	**0.990**	0.733	0.733	0.733	**0.957**	0.345	0.347	0.032	0.032	0.0319
Ours	RMSE	71.96	95.539	107.244	72.619	**72.770**	**87.462**	75.711	**87.858**	**81.709**	**79.112**	**81.334**	**91.994**
	MAE	33.81	44.961	51.358	27.406	**32.155**	**51.358**	30.793	47.847	**48.853**	**35.018**	**40.539**	**39.420**
	R-Sq.	0.952	0.915	0.893	0.951	**0.951**	**0.928**	0.946	**0.928**	**0.938**	**0.941**	**0.938**	**0.921**

Conclusion

This work suggested the application of an attention-fueled temporal convolutional network in conjunction with a convolutional neural network to predict global solar radiation (GSR). This approach is particularly effective when the accessible historical sequence of GSR spans durations of 24, 48, and 72 hours. We then compared the proposed model with other ML models used for GSR, including RNN and LSTM. The models were evaluated using RMSE, MAE, and R2 metrics. The empirical findings demonstrated that the suggested model exhibited superior performance compared to other models in most of the instances. Potential avenues for subsequent research could encompass broadening the temporal scope of the models to forecast one week or one month into the future and juxtaposing their performance with other established methodologies. Another direction is to investigate further feature generation and manipulation and to include other meteorological variables as input features for the models.

References

[1] Makade, R.G., Chakrabarti, S., Jamil, B.: Development of global solar radiation models: A comprehensive review and statistical analysis for indian regions. Journal of Cleaner Production 293, 126208 (2021) https://doi.org/10.1016/j.jclepro.2021.126208

[2] Solano, E.S., Dehghanian, P., Affonso, C.M.: Solar radiation forecasting using machine learning and ensemble feature selection. Energies 15(19), 7049 (2022). https://doi.org/10.3390/en15197049

[3] Ngiam, K.Y., Khor, W.: Big data and machine learning algorithms for health-care delivery. The Lancet Oncology 20(5), 262–273 (2019). https://doi.org/10.1016/S1470-2045(19)30149-4

[4] Rehman, S., Mohandes, M.: Artificial neural network estimation of global solar radiation using air temperature and relative humidity. Energy policy 36(2), 571– 576 (2008). https://doi.org/10.1016/j.enpol.2007.09.033

[5] Ghimire, S., Nguyen-Huy, T., Prasad, R., Deo, R.C., Casillas-Perez, D., SalcedoSanz, S., Bhandari, B.: Hybrid convolutional neural network-multilayer perceptron model for solar radiation prediction. Cognitive Computation 15(2), 645–671 (2023). https://doi.org/10.1007/s12559-022-10070-y

[6] Pang, Z., Niu, F., O'Neill, Z.: Solar radiation prediction using recurrent neural network and artificial neural network: A case study with comparisons. Renewable Energy 156, 279–289 (2020). https://doi.org/10.1016/j.renene.2020.04.042

[7] Alizamir, M., Shiri, J., Fard, A.F., Kim, S., Gorgij, A.D., Heddam, S., Singh, V.P.: Improving the accuracy of daily solar radiation prediction by climatic data using an efficient hybrid deep learning model: Long short-term memory (lstm) network coupled with wavelet transform. Engineering Applications of Artificial Intelligence 123, 106199 (2023). https://doi.org/10.1016/j.engappai.2023.106199

[8] Ali-Ou-Salah, H., Oukarfi, B., Bahani, K., Moujabbir, M.: A new hybrid model for hourly solar radiation forecasting using daily classification technique and machine learning algorithms. Mathematical Problems in Engineering 2021, 1–12 (2021). https://doi.org/10.1155/2021/6692626

[9] Paoli, C., Voyant, C., Muselli, M., Nivet, M.-L.: Forecasting of preprocessed daily solar radiation time series using neural networks. Solar energy 84(12), 2146–2160 (2010). https://doi.org/10.1016/j.solener.2010.08.011

[10] Xing, X., Li, Z., Xu, T., Shu, L., Hu, B., Xu, X.: Sae+ lstm: A new framework for emotion recognition from multi-channel eeg. Frontiers in neurorobotics 13, 37 (2019). https://doi.org/10.3389/fnbot.2019.00037

[11] Ghimire, S., Deo, R.C., Wang, H., Al-Musaylh, M.S., Casillas-P´erez, D., SalcedoSanz, S.: Stacked lstm sequence-to-sequence autoencoder with feature selection for daily solar radiation prediction: a review and new modeling results. Energies 15(3), 1061 (2022). https://doi.org/10.3390/en15031061

[12] Liu, G., Guo, J.: Bidirectional lstm with attention mechanism and convolutional layer for text classification. Neurocomputing 337, 325–338 (2019). https://doi.org/10.1016/j.neucom.2019.01.078

[13] Azizi, N., Yaghoubirad, M., Farajollahi, M., Ahmadi, A.: Deep learning based long-term global solar irradiance and temperature forecasting using time series with multi-step multivariate output. Renewable Energy 206, 135–147 (2023). https://doi.org/10.1016/j.renene.2023.01.102

[14] Gugulothu, N., Tv, V., Malhotra, P., Vig, L., Agarwal, P., Shroff, G.: Predicting remaining useful life using time series embeddings based on recurrent neural networks. arXiv preprint arXiv:1709.01073 (2017)

[15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems 30 (2017)

[16] Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)

[17] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90

[18] Balabaeva, K., Kovalchuk, S.: Comparison of temporal and non-temporal features effect on machine learning models quality and interpretability for chronic heart failure patients. Procedia Computer Science 156, 87–96 (2019). https://doi.org/10.1016/j.procs.2019.08.183