


Performance Evaluation of Neural Network-Based Short-Term Solar Irradiation Forecasts

Simon Liebermann ¹, YoungSeok Hwang ² and Stephan Schlüter ³*  0000-0001-5816-3337

¹ University of Ulm, Germany; simon.liebermann@uni-ulm.de

² Department of Climate Change, Kyungpook National University, Daegu, Korea; poi012345@naver.com

³ Faculty of Mathematics, Natural and Economic Sciences, University of Applied Sciences Ulm, Germany; stephan.schluter@thu.de

* Correspondence: stephan.schluter@thu.de

Abstract: Due to the globally increasing share of renewable energy sources like wind and solar power, precise forecasts for weather data are becoming more and more important. For forecasting hourly solar irradiation values, a long-short term (LSTM) neural network, a convolutional neural network (CNN), and combinations of both are benchmarked against each other. The naive forecast is included as a baseline. In order to improve forecasting precision, the extraterrestrial irradiation of the upcoming day is added as indicator for the maximum possible irradiation level. Besides, various locations across Europe are tested to analyze the models' performance under different climate conditions. Forecasts up to 24h in advance are generated and compared using different goodness of fit (GoF) measures. Thereby, GoF results are analyzed and interpreted in detail. As expected, the error of all models increases with rising forecasting horizon. Over all test stations it shows that combining an LSTM network with a CNN yields the best performance. However, regarding the chosen GoF measures, differences to the alternative approaches are fairly small. The hybrid model's advantage lies not in the improved GoF regarding an alternative simpler version but in its universality: Contrary to a LSTM or a CNN it produces very good results in all tested weather scenarios.

Keywords: Neural Network; Solar Irradiation; Time Series Forecasting; LSTM; CNN

MSC: 62M45

JEL Classification: C45;C53;C58

Citation: Liebermann, S.; Hwang, Y.S.; Schlüter, S. Neural Network-Based Solar Irradiation Forecasting. *Energies* **2021**, *1*, 0. <https://doi.org/>

Received:

Accepted:

Published:

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

Copyright: © 2021 by the authors. Submitted to *Energies* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The sun has been object of interest since the beginning of scientific research. Hence its movement over the year is well established and we can pretty precisely compute the maximum possible solar irradiation for any spot on earth. Michalsky [1] derived a set of formulas which allow to identify the current solar position with an error of $\pm 1^\circ$. The solar position allows compute the extraterrestrial irradiation for a certain GPS coordinate on earth. Bird's clear sky model ([2]), for example, uses this information to calculate the solar irradiation under clear sky conditions with an error of only $\pm 5\%$. However, when it comes to cloudy conditions, both estimation and forecasts of solar irradiation on the ground are far more complicated, but not less relevant – especially today. There are worldwide efforts to increase the use of solar power. However, if not installed in a desert and in absence of considerable battery power, solar panels are a highly volatile source of energy causing significant grid stabilization efforts. Nevertheless, the world and especially central European countries like Germany rely on it for transforming their current fossil-fuel dominated energy mix. Hence, there is a significant need especially for good short-term solar power forecasts. Power plant dispatching heavily relies on

such estimates to compute how much reserve power plant capacity needs to be activated on short notice. Besides new concepts like integrated energy, where different sectors (i.e. heating and driving) are connected, can benefit from better solar power forecasts.

High-quality solar irradiation forecasts are the major input factor deciding about the reliability of estimated future solar power amounts. Thereby, obviously, location matters. Forecasting models perform comparably better in areas with much sunshine and less clouds (Southern Europe) than in areas where clouds and/or precipitation is more likely [3] [4]. Due to the above described demand there is already an extensive literature on forecasting solar irradiation and/or solar power (see Section 2), whereby lately the focus was on methods that involve artificial neural networks (ANN). This concept, in most cases simply referred to as neural networks (NN), was invented in the 1940s and has been applied manifold like in Apple's Siri, for pattern recognition, etc. It has also been used for time series forecasting since 1996 [5]. In the context of solar irradiation forecasting we found more than 200 articles concerning with NN-based irradiation forecasts, whereby there are numerous suggestions to combine different NN setups with each other. Lately, various authors like Kreuzer et al. [6] or Wang et al. [7], for example, started to combine a convolutional neural network (CNN) with a long-short term (LSTM) neural network in order to incorporate interdependencies in time and between different climate data like wind, temperature, and irradiation. On the one side solar irradiation is more deterministic than temperature, as we have a clearly described annual and daily pattern. On the other side, solar irradiation on the ground depends on factors like the cloud coverage, which is both an autoregressive and a highly volatile stochastic process. Wang et al. [7] successfully use the above described combination to forecast global solar irradiation on the ground as, in their article, the method outperforms various NN benchmarks. However, as in [6], the authors' focus was rather on the method itself than on analyzing its performance in detail. Wang et al. [7], for example, apply the model to data from Alice Springs, Australia, which is an extremely sunny region. As noted above models perform worse in areas with less sunshine, as weather is less stable there. Lorenz [3] computed a forecasting error for Southern Europe between 20% and 35% whereas in central Europe of up to 60%! Hence, a good performance in a sunny area is not a strong argument for a forecasting method. It would be valuable information to know when and why a model performs good and if the comparably high complexity of combining a LSTM and CNN model really pays off regarding forecasting precision.

This paper aims to fill the gap. Different models like a LSTM network, a CNN and combinations of them are tested for various benchmark locations across Europe, ranging from sunny Almeria in Spain to Rovaniemi in Finland, located close to the Arctic circle. Results are analyzed in detail using various goodness of fit (GoF) measures like the root mean square error (RMSE) or the mean absolute error (MAE). Errors are checked for bias and skewness, for example. This detailed analysis allows to test the model's robustness under different climate conditions. As auxiliary variables for the forecast we add the extraterrestrial global irradiation and other meteorological data such as temperature and rainfall. We produce hourly, i.e. short-term forecasts for global irradiation on the ground for up to 24h in advance.

Overall, as a result, we see that errors increase with the forecasting horizon. This makes sense as a longer horizons offers more chance for changing weather. Albeit the combination of LSTM and CNN performs better than other NN benchmarks on the long run, it is outperformed for all locations for horizons up to 5h in advance. Besides, for Ulm, Hull and Rovaniemi, even for forecasting horizons larger than 5h the comparative advantage regarding precision does not justify the additional complexity. Interestingly, for Almeria, the sunniest candidate, combining CNN and LSTM network does make a change, as for the other locations, a comparably simple and well-established LSTM shows a reasonable performance. In Almeria it does not. Hence, the strength of the hybrid model shows in its universality. Above that extraterrestrial irradiation doesn't really

improve forecasting quality. We also see that all tested models are unbiased - except for the CNN, which on average overestimates irradiation a bit. Forecasting errors, which are computed as are slightly skewed to the left, meaning that if we underestimate irradiation, the risk of significantly missing the real value is larger than when overestimating.

The paper is structured as follows: In Section 2 we comment on existing literature and why we see the necessity to add another concept. Section 3 provides an explanation of neural networks: the idea of the concept, methods to train the NN, and possible variations that can be used for forecasting. Purpose is to allow readers that are not familiar with the concept to gain a certain insight in order to apply it to practical data.

2. Different Concepts to Compute Solar Irradiation Forecasts

There are different types of irradiation forecasting models, whereby the choice especially depends on the forecasting horizon of interest. Physical models, so called numerical weather forecasts, mostly try to identify the interactions of meteorological factors like wind, solar irradiation etc. For the forecast itself no historical data is needed, however one requires substantial information about the location and the local interaction of individual weather data [8,9]. Normally those methods are not used for very short-term forecasts [10]. If applied in this context, all-sky cameras and/or satellite images are involved [11,12]. Depending on the forecasting horizons, methods like autoregressive models or NNs might be involved as well to produce forecasts. There are various models available and literature has shown that quality can be improved by combining a few methods in the context of ensemble forecasts [13–16]. Hassani et al. [17], for example, combined 12 models to compute forecasts.

In contrary to numerical weather forecasts, stochastic models mainly focus on identifying deterministic patterns in historical data in order to produce forecasts. The simplest approach is the naive forecast, i.e. the rough assumption that the weather tomorrow is more or less the weather of today. In formulas $\hat{X}_{t+k} = X_{t+k-24}$, $k = 1, 2, 3, \dots, t$, whereby X_t is an observation at time instance t . Despite being rather daring, the assumption that structures are persistent has worked fairly well in many applications like temperature or energy demand forecasts. Alternatively, especially for forecasts up to 24h the seasonal autoregressive moving average model (SARIMA) [18] is used. However, in case of hourly solar irradiation data, SARIMA might face quite a few problems as the time series shows numerous instationarities. This is why different forms of NNs become more and more popular in irradiation forecasting. They combine the promise of identifying and processing complex interdependencies between different meteorological data and the fact that they are still possible to work on a stand-alone application. As mentioned above We found more than 200 publication (Q4 2020) about using machine learning techniques for irradiation forecasts and we cannot mention them all. Instead we try to give an overview over the range of potential models. Most involve neural networks, but there are also a few approaches based on support vector machines [19–21]. Abdel-Nasser [22] applied a recurrent LSTM model to forecast photovoltaic power, and one can find more and more complex approaches as mentioned above. Wang et al. [7] used a combination of LSTM and CNNs to forecast irradiation. Benali et al. [24] combined random forests with a NN. Random forests are part of supervised learning where combinations of decision trees are trained. Using this concept the authors hope to catch anomalies in the data. Ozoegwu [25] combined an autoregressive model with NNs. Pereira et al. [26] use NNs to improve forecasting precision of other algorithms. Gensler et al. [27], again, apply deep learning strategies to forecast solar power. Lima et al. [28] compare different techniques with each other and propose a deep learning strategy that is combined with a forecast based on portfolio theory, a concept from finance. They apply their method to data from Brasil and Spain and measure for hourly data in Spain a fairly low average forecasting error of only 0.36%. However, again, Spain is a country with a high number of sunshine hours over the year, so these numbers are to be treated with care.

Before explaining the models used in this article in detail we present a brief introduction to neural networks in Section 3.

3. An Introduction to Neural Networks

Back in 1996 Hill et al. [5] compared NN-based time series forecasts to classic statistical methods. They found that NNs outperformed the traditional methods regarding the analyzed data sets. As a consequence and with increasing computational power and data availability, interest and focus of more and more scientists was drawn towards NN techniques. Lately, especially CNNs introduced by LeCun et al. (1998) [29] and LSTM networks introduced by Hochreiter (1997) [30] showed a promising performance. This, in turn, motivated researchers to combine both approaches [6,7,31]. In our case study we consider all models mentioned here, i.e. a LSTM networks (see Section 3.2), a CNN (see Section 3.3 as well as thy hybrid (see Section 3.4).

3.1. Artificial Neural Networks

This section is mainly based on Zou et al. [32] to which we also refer for a more detailed introduction. Another good source of information is given by Aggarwal [33]. Here, the relevant features are given. ANNs are computing systems inspired by the biological (real-world) neural networks that constitute animal brains. Accordingly, an ANN is based on a collection of nodes called artificial neurons. These neurons are connected with weights and store biases and activation functions. Depending on the input different neurons will be activated and lead to a certain output. In order to explain the network's functionality, we focus on the three core components: node character, network topology and learning.

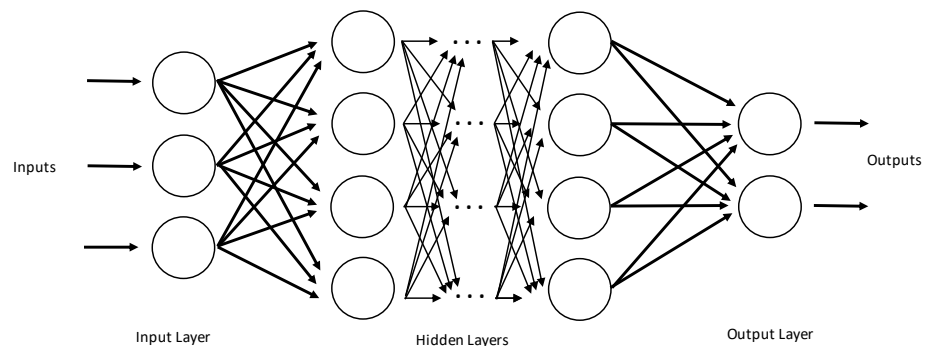


Figure 1. Simple artificial feedforward neural network

The **node character** provides information about the signal processing of the node, i.e. the number of inputs and outputs associated with the node, the weight associated with each input and output, and the activation function. The weighted sum of the input neurons is sent to an activation function, which generates the output. This process is shown in Figure 2. The most commonly used activation functions are the rectifier linear unit (ReLU), the tangens hyperbolicus (tanh) and the sigmoid function (S), whose formulas are displayed in Eq. (1) [34].

$$\text{ReLU}(x) = \max(x, 0) \quad \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad S(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

The neurons are organized in layers. Beside the input and output layers there is the possibility to include a certain number of hidden layers. Designing the **network topology** determines the number of nodes on each layer, the number of layers in the network, and the connections between the nodes. These factors are usually set by intuition or experience from other related work and then optimized through experiments.

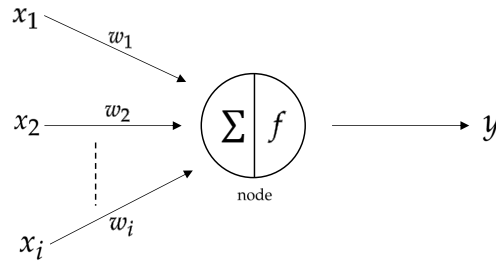


Figure 2. A basic model of a single node: x_i = inputs, w_i = weights, f = transfer function, y = output

176 Mainly there are two types of **learning**: supervised learning and unsupervised learning.
 177 Here we focus on supervised learning, meaning that a training set with inputs and
 178 corresponding label outputs is provided. For a feedforward network, as used here, a
 179 backpropagation mechanism is most commonly used as learning algorithm. During
 180 backpropagation the error between the network output and the label, which is also
 181 called loss, is minimized. A standard optimizer for this task is the Adaptive Moment
 182 Estimation (ADAM) [35], which we use as optimizer. Another important aspect of the
 183 learning procedure is the duration. The training process is divided into epochs. One
 184 epoch is over when all the inputs are processed. Since good accuracy is already achieved
 185 after a few epochs it is reasonable to implement an early stopping callback, i.e. to stop
 186 the training process if the accuracy has stopped to improve. This prevents overfitting
 187 and reduces training time.

188 3.2. Long Short-Term Memory Neural Networks

189 Long short-term memory is an artificial recurrent neural network (RNN) architec-
 190 ture to learn long-term dependence information and to deal with long time sequences.
 191 Such networks are well-suited for classifying or processing time series data as well as
 192 making predictions based on such time series data. The benefit of an LSTM network
 193 compared to traditional RNN is its internal memory unit and gate mechanism, which
 194 overcomes the gradient disappearance and gradient explosion problems of other NNs
 195 [30]. A LSTM cell consists of different gates and the corresponding vectors. Namely
 196 input gate i with output vector i_t , a forget gate f (output vector f_t), an update gate
 197 (output vector g_t) and an output gate o (output vector o_t). At each time step t , a LSTM
 198 maintains a hidden vector h_t and a memory vector c_t responsible for controlling state
 199 updates and outputs. The computations at time step t are defined as described in Eqs. (2
 200 - 4) [7].

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i); \quad f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

$$g_t = \tanh(W_g \cdot [h_{t-1}, x_t] + b_g); \quad o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes g_t; \quad h_t = o_t \otimes \tanh(c_t) \quad (4)$$

201 where σ denotes the sigmoid function (Eq. 1) and \otimes represents element wise multi-
 202 plication. The weight matrices W_i, W_f, W_g, W_o and biases b_i, b_f, b_g, b_o are determined in
 203 the course of the training process. The inputs of the four gates include the LSTM output
 204 value h_{t-1} at a former time step $t - 1$ and the input data x_t at a present time step. Figure
 205 3 shows the flow of the data in a single LSTM cell.

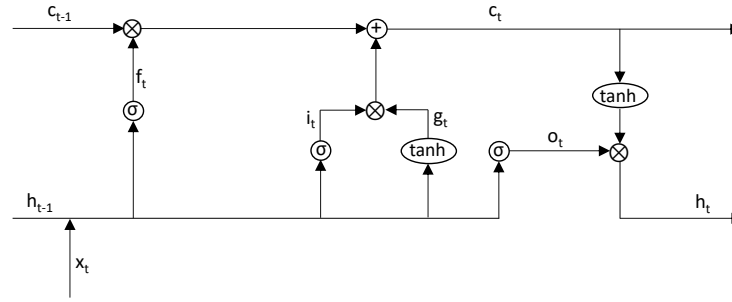


Figure 3. Data flow in a single LSTM cell

3.3. Convolutional Neural Networks

Convolutional neural networks (CNN) have been one of the most applied methods in the field of machine learning in the previous years. Initially developed for computer vision tasks, CNNs have shown superior performance in classification tasks [36] such as object recognition in images [37], speech recognition and modelling [38], and natural language processing [39]. Because of the great success of CNNs, they have been applied to other tasks, in particular time series forecasting. Promising results could be shown by Borovykh et al. [40] who applied CNNs mainly to financial data. Other applications are solar power forecasting and electricity load forecasting [41]. One significant advantage of CNN is that no manual feature extraction is required. CNNs can be applied to raw data and feature extraction is part of the training: inside the convolutional layers, filters for feature extraction are trained based on the data. Since the full pipeline structure from input data to network output is the result of the learning procedure, this methodology is called end-to-end learning [42]. Additionally, CNNs work well on noisy data, as in each subsequent layer the noise will be discarded [40]. A CNN mainly consists of a convolutional layer and a pooling layer. In the convolutional layer two techniques are used: local connectivity and weight sharing. Local connectivity means that each convolutional node is connected to a small subset of inputs (corresponding to a local region of the input and called a receptive field of the node) instead of all inputs as in the standard feedforward NNs. As before described CNNs can be used for multiple tasks. The biggest difference in the usage is the type of convolution which is used. Basically there are three types of convolutions [43]: For time-series data 1D convolution is used, 2D convolution is used for image data and 3D convolution is mostly used for 3D images like Magnetic Resonance Imaging data. The difference between those convolutions is the direction in which the kernel/filter moves. So in a 1D convolutional layer the kernel only moves in one direction, which is here the time. Whereas in a 2D convolution the kernel moves in horizontal and vertical direction. The advantages of a 1D convolution is that a relatively shallow architecture (i.e. small number of hidden layers and neurons) is enough to proceed time-series data whereas 2D convolution require deeper architecture to handle such tasks. This effects the training time and complexity. With few hidden layers and neurons the training time and the computational requirements are low [42]. For the kernel size in a 1D convolution only the width of the convolution matters, as it always processes all features. Therefore the kernel size is $\# \text{features} \times \text{convolution width}$. The calculations for the l -th time step of time series S and a kernel $K \in \mathbb{R}^{\beta \times \gamma}$ is defined by the $*$ operator in Eq. 5:

$$\hat{S}(l) = K * S(l) = \sum_{i=1}^{\beta} \sum_{j=1}^{\gamma} K(i, j) \cdot S_j(l + i - 1), \quad (5)$$

where S_i determines the i -th feature of S and \hat{S} the output of the convolution at time step l . Remark, the \hat{S} has no features anymore and that the number of time steps reduces by $\beta - 1$ (with a stride of 1).

The other major layer of a CNN are the pooling layers, which are used to make the outputs immutable for shifts in the input data. During the pooling process the input values become aggregated according the pooling size and pooling type. The most common type is maximum pooling where the biggest value of the pooling area is taken. In Figure 4 the operations of the layers are illustrated.

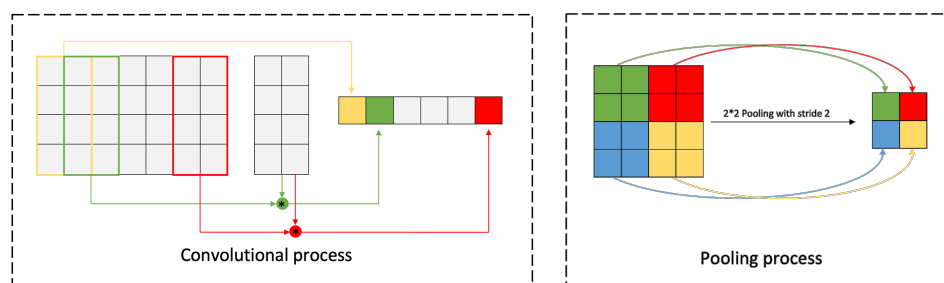


Figure 4. 1D-Convolutional process with kernel size 2 and stride on the left and pooling process with size 2*2 and stride 2 on the right

3.4. Hybrid models

As both LSTM and convolutional neural network are already widely used for forecasting purposes, various authors began to combine them to hybrid models with certain success **HIER EIN PAAR QUELLEN**, as their results were better than the forecasts of classic NN approaches [31]. The idea of hybrid models is that the LSTM network handles the temporal information of the historical data and the CNN the spatial information. Models differ in the sequence of LSTM and CNN step. The first version is to first insert data into the LSTM layer, whose output is then given to the convolution layer. We call this model short LSTMconv as Kreuzer et al. do [?]. **I.e. input data like wind, pressure, etc. are processed individually before being convoluted. If you do it the other way round, all input data are convoluted first. IST DA EIN VORTEIL BZW. BEGRÜNDUNG DAFÜR?? This model is labelled convLSTM**

4. Case Study

In the context of a case study we test various NN-based forecasting models at different locations across Europe in order to verify their performance under different climate conditions. The data sets and the required preprocessing steps are explained in Section 4.1; GoF measures are explained in Section 4.2, and results are given in Section 4.3. Almost all numerical computation we mainly use Python 3.7. Merely for calculating the maximum possible solar irradiation (MSI) we use R 3.6.3. The final model is set up with Tensorflow (v2.3.1), loading and managing the data is done with the pandas (v1.1.3) and numpy (v1.18.5) libraries. Visualizing the data is done with matplotlib (v3.3.1). For data scaling and error calculation the libraries scikit-learn (v0.23.2) and scipy (v1.5.2) are used. All source code is publicly available on GitHub via <https://github.com/schlueterTHU/SolarForecasting>.

4.1. Data Sets and Preprocessing

All data are downloaded from www.soda-pro.com. The corresponding raw data are measured by the United States National Aerospace Agency (NASA) and processed as MERRA-2 data set [44]. Thereby, MERRA stands for Modern-Era Retrospective analysis for Research and Applications and includes several meteorological indicators such as wind or temperature for any location worldwide and different time steps such as hourly or daily, for example. The spatial resolution is $0.625^\circ \times 0.5^\circ$. All data used in this case study are summarized in Table 1. Besides we include the MSI. As this value is due to the movement of earth around the sun it can be calculated with only a comparably small precision error. There is also a corresponding R package called *solarR* provided by [45].

Furthermore we add the difference between the global irradiation and the maximum irradiation. For all chosen locations, which are briefly characterized in Table 2, hourly data between 1st Jan, 2016 and 30th Nov, 2020 are obtained, which means we have 43,080 (multivariate) observations per city.

Table 1. Summary of the satellite data used in the network input.

Factor	Unit	Factor	Unit
Global irradiation	Wh/m ²	Temperature	°C
Relative Humidity	%	Pressure	mbar
Wind Speed	m/s	Rainfall	L/m ²

Table 2. List of Locations Considered in the Case Study.

City	Country	Description
Ulm	Germany	close to the Alps, continental climate
Almeria	Spain	hot and dry climate, large number of sunshine hours
Rovaniemi	Finland	close to arctic circle, cold and dark in winter
Hull	England	coastal climate with rapidly changing weather

As input for the NNs we use 24 time steps to predict the next 24 hours. To train the NNs we first split the data into a training, validation and test set. The training and validation sets are handed to the NNs, whereas the testing set is kept for evaluating the model's performance against real data (out of sample testing). In order to see how the NN performs for over the course of a whole year the test set contains data for one year, which means 8760 hours. The remaining data is split into 80% training data and 20% validation data. Apart from that, the time index is transformed with sine and cosine to catch the periodicity. This is done by fitting the day/year data to a sine/cosine oscillation by dividing the timestamp by the day/year. Then sine/cosine is applied and we obtain four variables, namely *day sine*, *day cosine*, *year sine* and *year cosine*. Moreover, for facilitating the NN data processing, we normalize all data to values between 0 and 1 using a MinMax Scaler. The architecture and training parameters of the networks can be found in appendix (A).

4.2. Goodness of Fit Measures

We compute two measures, namely the root mean square error and the mean absolute error, to evaluate the performance of the different forecasting models. The measures are calculated separately for each predicted hour: Given the i -th predicted global irradiation at the t -th hour $\hat{X}_i^{(t)}$ for the true global irradiation $X_i^{(t)}$, the RMSE for the hour t is defined as:

$$\text{RMSE}^{(t)} = \sqrt{\frac{1}{N} \sum_{i=1}^N (X_i^{(t)} - \hat{X}_i^{(t)})^2}.$$

where N is the number of samples, so in our case the size of the test set which is 8760. The MAE is defined analogously:

$$\text{MAE}^{(t)} = \frac{1}{N} \sum_{i=1}^N |X_i^{(t)} - \hat{X}_i^{(t)}|.$$

In order to compare the average error over all tested forecasting horizons we eventually compute the average for both RMSE and MAE:

$$\text{RMSE}_{\text{total}} = \frac{1}{T} \sum_{t=1}^T \text{RMSE}^{(t)}; \quad \text{MAE}_{\text{total}} = \frac{1}{T} \sum_{t=1}^T \text{MAE}^{(t)},$$

where T is the number of predicted time steps, so here $T = 24$. The unit for our case is always Wh/m^2 , as we compare global irradiation values. Furthermore, To compare also results from different locations, a measure that adjusts for the local solar irradiation amount is needed. As level adjustment the seasonal naive forecast at each location is used. We divide the models' errors by the error of the seasonal naive forecast (SN) and call them percentage RMSE (PRMSE) and percentage MAE (PMAE) respectively.

$$\text{PRMSE} = \frac{\text{RMSE}_{\text{model}}}{\text{RMSE}_{\text{SN}}}; \quad \text{PMAE} = \frac{\text{MAE}_{\text{model}}}{\text{MAE}_{\text{SN}}} \quad (6)$$

Note that night hours are removed before computing the above described error measures as those hours naturally reduce all models' errors without offering more insight.

4.3. Results and Performance Evaluation

Given all results we can say that merging a CNN with an LSTM model does indeed improve the forecasting performance. However, considering the additional complexity, it doesn't really pay off.

Before presenting the results for all tested locations we need to verify if and how adding the MSI to the set of input parameters makes sense. REgarding test results we only discuss Ulm here, as the results for the other locations are very similar: Adding the MSI to the input only slightly reduces forecasting errors as shown in Table 3. Alternatively, as the MSI can be calculated for any location and any time over the year, we might add future MSI values (up to 24h in advance) to the input set. Results from Table 3 show that errors are not significantly smaller either. The performance even decreases for some models (see convLSTM). Hence, we only use the factors from Table 1 and the MSI as input for all models.

Table 3. Comparison of different input factors for the NN. The table shows the RMSE values in Wh/m^2 . The results with smallest errors are marked in bold letters.

Model	RMSE			MAE		
	Input1	Input2	Input3	Input1	Input2	Input3
LSTM	98.16	97.71	98.95	68.70	68.41	69.93
convolutional	98.63	99.56	101.07	68.56	70.70	71.95
convLSTM	96.64	95.25	96.64	65.69	65.34	65.71
LSTMconv	97.23	97.35	99.52	67.37	66.75	69.76

Input1: factors from Table 1 + MSI and MSI of forecast horizon;

Input2: factors from Table 1 + MSI;

Input3: factors from Table 1;

Having settled the issue how to handle the MSI, we compute 24h ahead forecasts for all locations and different neural network models, namely the LSTM network, the CNN, and both hybrides convLSTM and LSTMconv. The forecasts are evaluated using the GoF measures from Section 4.2. Results for all four cities are displayed in Figure 5, where the MAE for each individual hour of the 24h ahead forecast is shown. The errors are thereby calculated based on the above described test set, i.e. for 8760h from 1st Dec, 2019 to 30th Nov, 2020.

From the graphic representation we can draw some major conclusions: First and foremost, exact for the seasonal naive forecast, MAE values of the different NN models differ only slightly. In the first few hours the convolutional and convLSTM network perform the best. The CNN is outperformed and overall the LSTMconv model seems to perform the best. The simple LSTM model has the worst performance, which is still close to the others though.

By comparing the cities we notice, that here the main difference is the the error level. In Rovaniemi, where we have big seasonal differences (no solar irradiation in winter, all day irradiation in summer), the networks perform best (MAE is roughly around 50 Wh/m²), whereas in Ulm with no specific seasonal pattern the networks produce the largest errors (MAE is roughly around 75 Wh/m²).

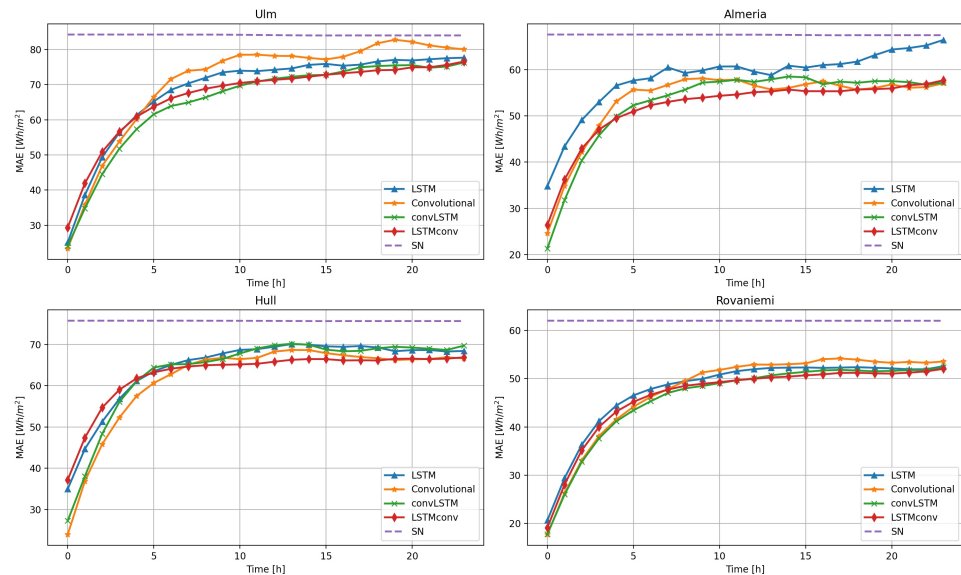


Figure 5. MAE for various neural network models: LSTM, Convolutional, convolutional LSTM (convLSTM) and LSTM convolutional (LSTMconv). As reference the seasonal naive forecast (SN) is shown.

The aggregated results for all models and exemplary for the cities Ulm and Almeria are shown in Table 4, whereby the best results are highlighted in bold letters. Values for Hull and Rovaniemi are given in Appendix A. Thereby, results from Table 4 just confirm the graphical observations. In general, combining an CNN with an LSTM model increases forecasting precision.

Table 4. Performance results of the different NN Models for the cites Ulm and Almeria.

Model	Ulm			Almeria		
	RMSE	MAE	PMAE	RMSE	MAE	PMAE
LSTM	97.71	68.41	81.35%	87.63	58.36	86.45%
convolutional	99.56	70.70	84.08%	85.08	53.28	78.93%
convLSTM	97.35	65.34	77.70%	85.50	52.78	78.19%
LSTMconv	95.25	66.75	79.37%	86.58	51.89	76.87%

For more insight into the models' performance we also consider distributional properties more in detail. Thereby we see that, except for the convolutional model, all models are more or less unbiased. The bias of the naive forecast is closest to zero, whereas the convolutional model's bias is positive. Hence, as the error is computed as true irradiation minus forecasted irradiation ($X_i^{(t)} - \hat{X}_i^{(t)}$), the convolutional model tends to underestimate irradiation levels. Besides, at all locations and for all considered forecasting horizons we find that model errors are negatively skewed. In Figure 7, where this fact is exemplary shown for Ulm, we see a quite uniform behavior except for the CNN, which is still skewed though. Negative skewness means that, when overestimating irradiation, the risk of missing the real value significantly is larger than when underestimating irradiation. The skewness is clearly visible in Figure 6 where the histogram of the six hour-ahead forecast for the LSTMconv model for Ulm is displayed.

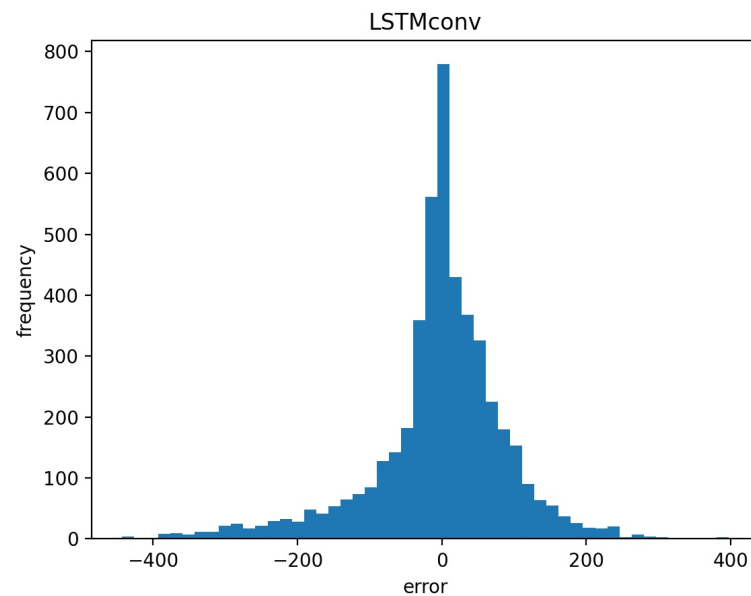


Figure 6. Histogram of the prediction errors for the LSTMconv model for six hour-ahead forecast in the city Ulm.

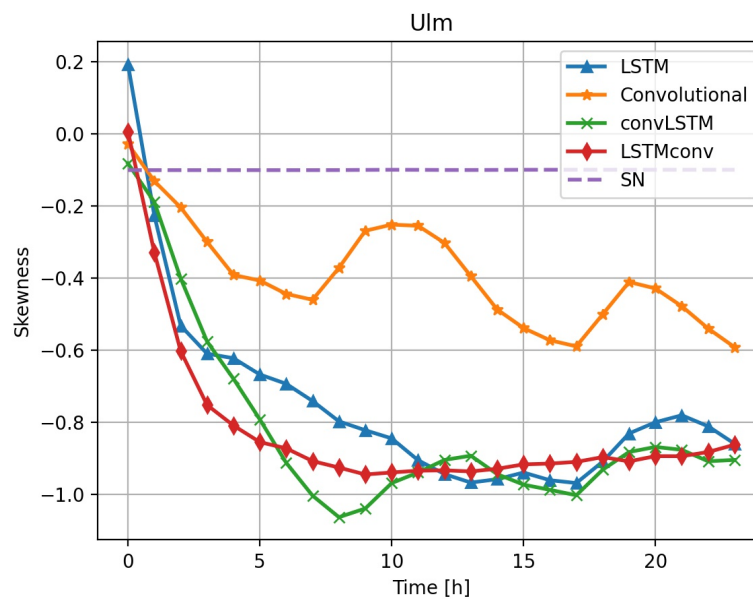


Figure 7. Skewness of the different NN models.

362 To visualize the models' performance for two specific situations we plot the fore-
 363 casting results of each model in Fig. 8. A 24h ahead forecast was computed for two days
 364 at 6 a.m. The right graphic shows the irradiation forecasts for 22nd Mar, 2020. Here
 365 we see that even though the irradiation on the previous day (input data) is on a low
 366 level all NNs are able to predict more or less the correct irradiation level. Nevertheless,
 367 the forecast is lower than the actual irradiation. On 16th Feb, 2020 (left plot) there is a
 368 irradiation drop in the morning, which was not incorporated by any NN.

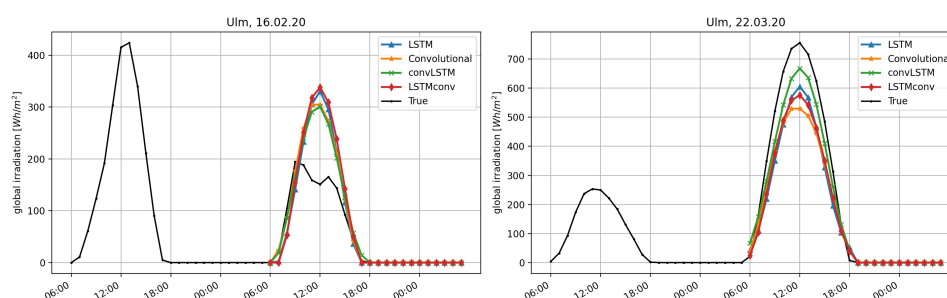


Figure 8. Global irradiation forecasts for Ulm on 16th Feb, 2020 (left) and 22nd Mar, 2020 (right)

Eventually we compare the MAE values to the mean global irradiation per hour for each city during the test sets' period. Thereby, for the mean, we only consider hours with positive global irradiation values. The results are shown in Table 5. This shows that – as expected – the NNs perform better in sunny regions like around Almeria than in regions with instable weather like Ulm or Hull.

Table 5. MAE of the LSTMconv network and the mean global irradiation per hour

	Ulm	Almeria	Hull	Rovaniemi
MAE	66.75	51.89	62.87	46.38
Mean	296.46	407.88	248.27	179.79
<u>MAE</u> Mean	22.5%	12.7%	25.3%	25.8%

To sum up our results: The LSTMconv neural network performs best for this specific forecasting task across all tested locations. However, the advantage is small – especially when considering that the training process of the neural networks appeared to be not deterministic (**DETERMINISTISCH ODER KONVERGENT??**). This means that the results vary – albeit very little – when training the same network multiple times. Furthermore the problem of the non-deterministic training process makes fine tuning of the neural networks quite hard. So, the LSTM seems like a good alternative, however its performance for sunny Almeria is comparably worse. The CNN, again, has problems with the continental climate of Ulm, but would be a feasible alternative for the other locations. The LSTMconv, again, shows as mentioned above the best performance for all locations. Hence, it is the best model not because it is the most complex one, but because it appears to be a rather universal approach producing fairly good results in all tested climate conditions.

In terms of computational time all networks are feasible for practical application as the training, which has to be done once a day or once a week, is done in less than 30 minutes on a regular laptop.

5. Conclusion

Lately, using neural networks for forecasting purposes has become very much fashion. This is proven by the wide range of literature about this topic. The motivation simply is that NNs show a comparably good performance and authors have tried to improve on these level by combining different types of NNs. In this article we analyse suitability for solar irradiation forecasting by comparing the performance of different NN models, namely a LSTM, a CNN and two hybrid versions. We consider short-term forecasts up to 24h ahead and test the models on four different locations in Europe to check the local climate's influence on the overall behavior. It shows that the hybrid versions, i.e. the combination of a CNN and a LSTM model, outperform the other models, but the advantage is not significant - given our data sets and the tested climate conditions. Except for Almeria, the sunniest location, a comparably simple LSTM performs very similar to a more complex combination of CNN and LSTM model.

However, the additional complexity does pay off, because even if the forecasting error is not considerable smaller compared to the other tested NNs, the hybrid models are more robust against changing climate conditions. The LSTM model has problems in sunny regions, the CNN for Ulm, i.e. a continental climate. The hybrid method's produce good forecasts in all scenarios. Eventually, there is still some research to be done. First, one might to check the models' performance at more locations across the planet, longer data sets might increase the results' reliability, although as we work with hourly values, one year should already guarantee significance of the results. Second, when analyzing more years in the test data set one could study seasonal differences.

Author Contributions: "Conceptualization, S.L., S.S. and YS.H.; methodology, S.L.; software, S.L.; validation, S.L. and S.S.; formal analysis, S.L. and S.S.; investigation, S.L. and YS.H.; resources, S.L.; data curation, YS.H.; writing—original draft preparation, S.L. and S.S.; writing—review and editing, S.L. and S.S.; visualization, S.L.; supervision, S.S.; project administration, S.S.. All authors have read and agreed to the published version of the manuscript.", please turn to the [CRediT taxonomy](#) for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: This research received no external funding.

Data Availability Statement: Data can be downloaded free of charge for research purposes via www.soda-pro.com.

Acknowledgments: The authors would like to thank David Kreuzer for his valuable comments about his Python codes.

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

Sample Availability: For data samples please contact the authors.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial neural network
CNN	Convolutional neural network
LSTM	Long-short term memory (neural network)
MAE	Mean absolute error
MSI	Maximum solar irradiation
NN	Neural network
PRMSE	Percentage root mean square error
PMAE	Percentage mean absolute error
RMSE	Root mean square error
RNN	Recurrent neural networks
SARIMA	Seasonal autoregressive integrated moving average (model)
SN	Seasonal naive

Appendix A.

Table A1. Performance results of the different NN Models for the cities Hull and Rovaniemi.

Model	Hull			Rovaniemi		
	RMSE	MAE	PMAE	RMSE	MAE	PMAE
LSTM	91.50	64.39	85.06%	72.33	47,65	76,83%
convolutional	92.82	61.59	81.36%	74.21	47,56	76,68%
convLSTM	91.23	63.54	83.94%	71.25	46,00	74,16%
LSTMconv	91.09	62.87	83.05%	71.59	46,38	74,78%

Table A2. Architecture of LSTM network

category	parameter	value	description
LSTM Layer	units	32	amount of LSTM cells
Dense Layer	units	24	amount of hours to predict
training	loss	MSE	Mean squared error
training	optimizer	ADAM Optimizer	
training	learning rate	0.0001	learning rate of optimizer

Table A3. Architecture of convolutional network

category	parameter	value	description
Conv1D Layer	filters	64	amount of filters
Conv1D Layer	kernel size	3	size of a filter
MaxPool1D Layer	pool size	2	size of MaxPool kernel
Conv1D Layer	filters	64	amount of filters
Conv1D Layer	kernel size	2	size of a filter
MaxPool1D Layer	pool size	2	size of MaxPool kernel
Flatten Layer			
Dense Layer	units	24	amount of hours to predict
training	loss	MSE	Mean squared error
training	optimizer	ADAM Optimizer	
training	learning rate	0.001	learning rate of optimizer

Table A4. Architecture of convolutional LSTM network

category	parameter	value	description
Conv1D Layer	filters	256	amount of filters
Conv1D Layer	kernel size	3	size of a filter
Conv1D Layer	filters	256	amount of filters
Conv1D Layer	kernel size	2	size of a filter
MaxPool1D Layer	pool size	2	size of MaxPool kernel
LSTM Layer	units	32	amount of LSTM cells
Dense Layer	units	24	amount of hours to predict
training	loss	MSE	Mean squared error
training	optimizer	ADAM Optimizer	
training	learning rate	0.0001	learning rate of optimizer

Table A5. Architecture of LSTM convolutional network

category	parameter	value	description
LSTM Layer	units	32	amount of LSTM cells
Conv1D Layer	filters	256	amount of filters
Conv1D Layer	kernel size	3	size of a filter
MaxPool1D Layer	pool size	2	size of MaxPool kernel
Flatten Layer			
Dense Layer	units	24	amount of hours to predict
training	loss	MSE	Mean squared error
training	optimizer	ADAM Optimizer	
training	learning rate	0.00001	learning rate of optimizer

References

1. Michalsky J.J. The Astronomical Almanac's algorithm for approximate solar position (1950–2050). *Solar Energy*, **1988**, *40*(3), 227–235.
2. Bird, R. E.; Hulstrom, R. L. A Simplified Clear Sky Model for Direct and Diffuse Insolation on Horizontal Surfaces. *NREL Technical Report*, **1981**, <https://www.nrel.gov/docs/legosti/old/761.pdf> (last accessed on 28 Jan 2021).
3. Lorenz, E.; Hurka, J.; Heinemann, D.; Beyer, H.. Irradiance forecasting for the power prediction of grid-connected photovoltaic systems. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **2009**, *2*(1), 2–10.
4. Müller, S.; Remund, J. Advances in radiation forecast based on regional weather models MM5 and WRF. Proceedings of the 25th European Photovoltaic Solar Energy Conference, Valencia, Spain, **2010**.
5. Hill, T.; O'Connor, M.; Remus, W. Neural Network Models for Time Series Forecasts. *Management Science*, **1996**, *47*(7), 1082–1092.
6. Kreuzer, D.; Munz, M.; Schlüter, S. Short-term temperature forecasts using a convolutional neural network — An application to different weather stations in Germany. *Machine Learning with Applications*, **2020**, *2*, 100007.
7. Wang, K.; Qi, X.; Liu, H. Photovoltaic power forecasting based LSTM-Convolutional Network. *Energy*, **2019**, *189*.
8. Bakker, K.; Whan, K.; Knap, W.; Schmeits, M. Comparison of statistical post-processing methods for probabilistic NWP forecasts of solar radiation. *Solar Energy*, **2019**, *191*.
9. Yeom, J.-M.; Deo R.-C.; Adamowski J.F.; Chae T.; Kim, D.-S.; Han, K.-S.; Kim, D.-Y. Exploring solar and wind energy resources in North Korea with COMS MI geostationary satellite data coupled with numerical weather prediction reanalysis variables. *Renewable and Sustainable Energy Reviews*, **2020**, *119*, 109570.
10. Murata, A.; Ohtake, H.; Oozeki, T. Modeling of uncertainty of solar irradiance forecasts on numerical weather predictions with the estimation of multiple confidence intervals. *Renewable Energy*, **2018**, *Issue 117*, 193–201.
11. Perez, R. et al., 2010. Improving the Performance of Satellite-to-Irradiance Models using the Satellite's Infrared Sensors. Proceedings of the ASSES Annual Conference 2010 in Phoenix, Arizona, USA, **2010**.
12. Pelland, S.; Galanis, G.; Kallos, G. Solar and photovoltaic forecasting through post-processing of the Global Environmental Multiscale numerical weather prediction model. *Progress in Photovoltaics: Research and Application* **2013**, *21*(3), 284–296.
13. Ernst, B.; Reyer, F.; Vanzetta, J. Wind power and photovoltaic prediction tools for balancing and grid operation. Proceedings of the CIGRE/IEEE PES joint symposium integration of wide-scale renewable resources into the power delivery system, Calgary, Alberta, Canada **2009**.
14. Chaouachi, A.; Kamel, R.; Nagasaka, K. Neural network ensemble-based solar power generation short-term forecasting. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **2010**, *14*(1), 69–75.
15. Möller, A.; Groß, J. Probabilistic temperature forecasting based on an ensemble autoregressive modification. *Quarterly Journal of the Royal Meteorological Society*, **2016**, *142*, 1385–1394.
16. Kann, A.; Haiden, T.; Wittmann, C. Combining 2-m temperature nowcasting and short range ensemble forecasting. *Nonlinear Processes in Geophysics*, **2011**, *18*, 903–910.
17. Hassani, H.; Silva, E. S.; Gupta, R.; Das, S. Predicting global temperature anomaly: A definitive investigation using an ensemble of twelve competing forecasting models. *Physica A: Statistical Mechanics and its Applications*, **2018**, *509*, 121–139.
18. McNeil, A. J.; Frey, R.; Embrechts, P. Quantitative risk management: Concepts, techniques and tools, revised ed., pages 125–157, Princeton University Press: Princeton, U.S.A., 2015.
19. Lin, G.-Q.; Li, L.-L.; Tseng M.L.; Liu H.-M.; Yuan, D.-D.; Tan R.-R. An improved moth-flame optimization algorithm for support vector machine prediction of photovoltaic power generation. *Journal of Cleaner Production*, **2020**, *253*, 119966.
20. VanDeventer, W.; Jamei, E.; Thirunavukkarasu, G.S.; Seyedmahmoudian, M.; Soon T.K.; Horan, B.; Mekhilef, S.; Stojcevski, A. Short-term PV power forecasting using hybrid GASVM technique. *Renewable Energy*, **2019**, *140*, 367–379.
21. Zendehboudi, A.; Baseer, M.; Saidur, R. Application of support vector machine models for forecasting solar and wind energy resources: a review. *Journal of Cleaner Production*, **2018**, *199*, 272–285.

- 489 22. Abdel-Nasser, M.; Mahmoud, K. Accurate photovoltaic power forecasting models using
490 deep lstm-rnn. *Neural Computing and Applications*, **2019**, *31*, 2727–2740.
- 491 23. Wang K.; Qi X. Liu H. Photovoltaic power forecasting based LSTM-Convolutional Network.
492 *Renewable Energy* **2009**, *189*, 116225.
- 493 24. Benali, L.; Notton, G.; Fouilloy A.; Voyant C.; Dizene, R. Solar radiation forecasting using
494 artificial neural network and random forest methods: application to normal beam, horizontal
495 diffuse and global components. *Renewable Energy*, **2019**, *132*, 871–884.
- 496 25. Ozoegwu, C. G. Artificial neural network forecast of monthly mean daily global solar
497 radiation of selected locations based on time series and month number. *Journal of Cleaner
498 Production*, **2019**, *216*, 1–13.
- 499 26. Pereira, S.; Canhoto, P.; Salgado, R.; Costa, M. J. Development of an ANN based corrective
500 algorithm of the operational ECMWF global horizontal irradiation forecasts. *Solar Energy*,
501 **2019** *185*, 387–405.
- 502 27. Gensler, A.; Henze, J.; Sick, B.; Raabe, N. Deep Learning for solar power forecasting - an
503 approach using AutoEncoder and LSTM Neural Networks. Proceedings of the 2016 IEEE
504 International Conference on Systems, Man and Cybernetics (SMC) in Budapest, Hungary,
505 **2016**.
- 506 28. Lima, M. A. F.; Carvalho, P. C.; Fernandes-Ramirez, L. M.; Braga, A. P. Improving solar
507 forecasting using Deep Learning and Portfolio Theory. *Energy*, **2020**, *195*.
- 508 29. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document
509 recognition. *Proceedings of the IEEE*, **1998**, *86*, 2278–2324.
- 510 30. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation*, **1997**, *9*,
511 1735–1780.
- 512 31. He, Z.; Zhou, J.; Dai, H. N.; Wang, H. Gold price forecast based on LSTM-CNN model. Pro-
513 ceedings of the 17th IEEE International Conference on Dependable, Autonomic and Secure
514 Computing, IEEE 17th International Conference on Pervasive Intelligence and Computing,
515 IEEE 5th International Conference on Cloud and Big Data Computing, 4th Cyber Science in
516 Fukuoka, Japan, 1046–1053, 2019.
- 517 32. Zou, J.; Han, Y.; So, S.-S. Overview over artificial networks. In *Artificial Neural Networks
518 Methods and Applications*; Livingston D.J.; Humana Press, Totowa, New Jersey, U.S.A., 2009;
519 12–23
- 520 33. Aggarwal C. C. *Neural Networks and Deep Learning*, 1st ed.; Springer International Publishing:
521 Springer, Cham, 2018.
- 522 34. Ohn, I.; Kim, Y. Smooth Function Approximation by Deep Neural Networks with General
523 Activation Functions. *Entropy* **2019**, *21*, 627. <https://doi.org/10.3390/e21070627>
- 524 35. Kingma, D. P.; Ba, J. L. Adam: A method for stochastic optimization. Proceedings of the 3rd
525 International Conference on Learning Representations in San Diego, CA, USA, 1–15, **2015**.
- 526 36. Karim, F.; Majumdar, S.; Darabi, H.; Chen, S. LSTM Fully Convolutional Networks for Time
527 Series Classification. *IEEE Access*, **2017**, *6*, 1662–1669.
- 528 37. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional
529 Neural Networks. Proceedings of the International Conference on Neural Information
530 Processing Systems in Lake Tahoe, Nevada, USA, **2012**.
- 531 38. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.; Mohamed, A.; Jaitly, N.; Senior, A.; Vanhoucke, V.;
532 Nguyen, P.; Sainath, T.N.; Kingsbury, B. Deep neural networks for acoustic modelling in
533 speech recognition. *IEEE Signal Processing Magazine*, **2012**, *29*(6), 82–97.
- 534 39. Jozefowicz, R.; Vinyals, O.; Schuster, M.; Shazeer, N.; Wu, Y. Exploring the limits of language
535 modeling. **2016**, URL: <https://arxiv.org/pdf/1602.02410.pdf> (accessed on 28 Jan 2021).
- 536 40. Borovykh, A.; Bohte, S.; Oosterlee, C. W. Conditional time series forecasting with convolu-
537 tional neural networks. Proceedings of the International Conference on Artificial Neural
538 Networks (ICANN).
- 539 41. Koprinska, I.; Wu, D.; Wang, Z. Convolutional Neural Networks for Energy Time Series
540 Forecasting. Proceedings of the International Joint Conference on Neural Networks, 2018-
541 July.
- 542 42. Kiranyaz, S.; Avci, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D. J. 1D convolutional neu-
543 ral networks and applications - A survey. **2019**, URL: <https://arxiv.org/pdf/1905.03554.pdf>
544 (accessed on 15 Nov 2020)
- 545 43. Understanding 1D and 3D Convolution Neural Network | Keras. Available online:
546 <https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610>
547 (accessed on 15 Nov 2020).

-
- 548 44. Global Modeling and Assimilation Office (GMAO) (2015), MERRA-2 tavg1_2d_slv_Nx: 2d,1-
549 Hourly,Time-Averaged,Single-Level,Assimilation,Single-Level Diagnostics V5.12.4, Green-
550 belt, MD, USA, Goddard Earth Sciences Data and Information Services Center (GES DISC),
551 Accessed [Data Access Date] DOI:10.5067/VJAFPLI1CSIV
- 552 45. Perpiñán. solaR: Solar Radiation and Photovoltaic Systems with R. *Journal of Statistical*
553 *Software*, **2012**, 50(9), 1 - 32. URL = <http://www.jstatsoft.org/v50/i09/>
- 554 46. Lee, S.; Lee, Y.-S.; Son, Y. Forecasting daily temperatures with different time interval data
555 using deep neural networks. *Applied Sciences*, **2020**, 10, 1609.

