Solar Irradiance Forecasting with Neural Networks

Lii Schmidt (2024)

Eindhoven, July 22, 2024

Contents

1 | Data

This project aims to predict solar irradiance using Long Short-Term Memory (LSTM) and Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) models. Preprocessing two data sets that contain different features stored per an interval of ten minutes and projecting solar irradiance for two hours ahead (12 data points). The two preprocessed datasets should be easily utilizable in both models in the notebook.

1.1 | First Dataset

The first dataset is derived from the sunshine and radiation dataset kis_tos_202403.gz from the Royal Netherlands Meteorological Institute (KNMI) Data Platform [2]. This dataset includes, among others, measurements of average solar radiation in J/cm^2 (Q) and sunshine duration (SQ) at 10-minute intervals for the entire month of March 2024.

1.1.1 | Preprocessing

The dataset includes multiple weather stations. Here, only data from the first weather station was selected to avoid multiple measurements for the same timestamp. This was done to ensure consistency and avoid the complexities of handling data from multiple sources.

Nighttime values (when solar irradiance is zero) were removed to enhance the prediction process, as only relevant data is fed to the model. Through data exploration, it was found that for the given month, the nighttime was from 5AM until 6PM. However, the timeframe can be different given a certain location or time of the year. This practice aligns with findings from Alzahrani *et al.* (2017), which suggest that eliminating irrelevant data points improves the performance of predictive models [4]. There were no rows with missing values in the dataset, but if a new dataset would be used, any rows with null values should be dropped to ensure data integrity.

Additionally, all numerical features were scaled using Standard Scaler, which transforms the data by subtracting the mean and scaling to unit variance, which results in a dataset where each feature has a mean of zero and a standard deviation of one. This is crucial for neural network models, as unscaled data can lead to issues where certain features dominate due to their larger scale, potentially causing the gradient descent algorithm to converge slowly or get stuck. Hence, it was done to ensure they are on a similar scale, which helps in improving the convergence of gradient descent during the training of neural networks

1.2 | Second Dataset

The second dataset comes from the KNMI daily weather data archive for March 2024 [1]. This dataset provides a broader range of meteorological variables, but after feature selection, only sunshine duration, time, relative atmospheric humidity, and temperature were kept as features.

1.2.1 | Preprocessing

Similar to the first dataset, data from the same weather station (station 215) was selected to maintain consistency. Nighttime values were removed to focus on periods with solar irradiance. This ensures that the model is trained only on relevant data, enhancing its predictive capability. Moreover, all numerical features were scaled, and the date field was converted to a datetime format for easier manipulation.

1.2.2 | Feature Selection

While the second dataset originally includes a broader range of features, it is important to note that an increase in the number of features does not always enhance predictive performance. To systematically evaluate the relevance of each feature, a Random Forest model was employed to calculate feature importances. This method helps identify which features contribute most significantly to the prediction of solar irradiance [13].

Random Forest is a robust ensemble learning method, and it is known for its efficiency in handling high-dimensional data and evaluating feature importance. By constructing multiple decision trees and averaging their predictions, Random Forest reduces overfitting and provides a robust measure of each feature's importance. This ensures that the features included in the final model are those that realistically improve predictive performance [12, 6].

Including too many features can lead to overfitting, where the model learns the noise in the training data rather than the underlying pattern, therefore degrading the model's performance on unseen data. Additionally, using too many features increases computational complexity, making the training process slower, more resource-intensive, and complicating the model interpretation and maintenance. Therefore, a balance must be found between including enough features to capture the patterns and excluding those that add noise or redundancy. This trade-off is significant in the context of solar irradiance forecasting, where the primary indicators of irradiance are relatively well understood [5].

Although feature importance scores provide a quantitative basis for feature selection, it is crucial to also consider insights from relevant literature. Features that might intuitively or quantitatively seem less important could still hold value based on domain-specific knowledge [10].

2 | Models

A neural network is a series of algorithms that attempt to identify underlying relationships in a set of data through a process that simulates the way the human brain operates. They consist of layers of connected nodes, or neurons, where each connection represents a weight adjusted during training. These models are capable of performing complex tasks like classification, regression, and pattern recognition by learning from data.

LSTM (Long Short-Term Memory) and CNN-LSTM (Convolutional Neural Network-LSTM) models are considered very effective for solar irradiance forecasting due to their ability to capture both temporal and spatial dependencies in data. LSTM networks are efficient at handling time series data, because they can learn long-term dependencies and avoid the vanishing gradient problem, which is crucial for accurately predicting solar irradiance that varies over time. CNNs, on the other hand, are great at identifying spatial features from data. Combining these two architectures in CNN-LSTM models builds on the strengths of both, CNNs derive meaningful spatial features from input data, which are then processed by LSTM layers to learn time-series patterns, resulting in improved forecasting accuracy [9, 16].

Alternatives to LSTM and CNN-LSTM models include machine learning methods such as Support Vector Machines (SVM), Recurrent Neural Networks (RNN), and individual Convolutional Neural Networks (CNN). However, these have certain limitations - SVMs and RNNs often have difficulty dealing with large-scale time series data due to their simpler architectures and inability to capture long-term dependencies effectively. Standalone CNNs, while powerful in extracting spatial features, are not inherently designed to handle time-series data without modification. Comparative studies have demonstrated that hybrid models like CNN-LSTM outperform these traditional approaches by providing superior prediction accuracy and robustness under various climatic conditions [7, 11].

Therefore, both LSTM and CNN-LSTM models were utilized. They were trained using the first dataset. The model takes the input sequence worth 24 or 48 hours (details in the notebook) on a rolling basis and outputs the 2-hour window by making one prediction at a time.

2.1 | LSTM

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to address the limitations of traditional RNNs, particularly the vanishing gradient problem. LSTMs are capable of learning long-term dependencies and are well-suited for sequence prediction problems due to their ability to remember information for extended periods.

LSTMs have a structure that includes a cell state and various gates to control the flow of information. Firstly, there is a cell state, which acts as a memory that carries information across time steps. Then an input gate determines how much of the new information should be stored in the cell state. Forget gate decides what information to discard from the cell state, and output gate determines the information to be passed on to the next hidden state. Given their ability to capture temporal dependencies, LSTMs are frequently used in time series forecasting.

Padding sequences is a process used to ensure that all input sequences in a batch have the same length. In time series forecasting and sequence modelling, input sequences can vary in length, but neural network models like LSTMs require inputs of uniform size to process them in parallel. Padding involves adding extra values (typically zeros) to shorter sequences to match the length of the longest sequence in the batch. This uniformity allows for efficient batch processing and helps leverage GPU acceleration.

Creating sequences is important in time series prediction because it helps the model learn temporal dependencies in the data. In this context, sequences represent chunks of historical data points used to predict future values. For example, in sequences of 24 hours of data (with 10-minute intervals), the model can learn patterns such as daily cycles and trends that are crucial for accurate forecasting. These sequences serve as the input (features) and output (target) pairs for the model, so it understands how past observations influence the future predictions.

2.1.1 | Implementation

In the notebook, the data is preprocessed and reshaped into sequences. This involves converting the dataset into a supervised learning format by creating input-output pairs for the model.

The LSTM model is defined with appropriate layers, given that due to processing power limitations, iterative hyperparameter tuning was not feasible. The number of LSTM units and the depth of the network are crucial hyperparameters. The model is compiled with a loss function and optimizer. Mean Squared Error (MSE) is used for regression tasks, and the Adam optimizer is selected for its efficiency. The model is trained using the prepared sequences. The number of epochs and batch size are also important hyperparameters that influence the training process.

The hyperparameters that needed to be specified were epochs, batch size, LSTM units, optimiser, filters, and kernel size in CNN. 50 epochs were chosen to balance between sufficient training and avoiding overfitting. A batch size of 32 was selected to balance between training speed and convergence stability. The Adam optimizer was chosen for its adaptive learning rate capabilities and efficient performance. 50 units were used to capture sufficient temporal dependencies without making the model too complex. 64 filters with a kernel size of 3 were used to extract meaningful spatial features from the data.

2.2 | CNN-LSTM

The CNN-LSTM model combines convolutional layers to capture spatial dependencies and LSTM layers to capture temporal dependencies. This combination is powerful for time series forecasting, especially when dealing with spatial-temporal data like solar irradiance.

2.2.1 | Implementation

The implementation of CNN-LSTM is similar to LSTM, but the data is reshaped to fit the CNN-LSTM architecture. The CNN-LSTM model is defined, combining convolutional layers with LSTM layers. Then, the model is compiled and trained similarly to the LSTM model.

3 | Further Refinements

The desired visualizations were not feasible because the neural networks overwrite the predictions during each iteration. However, it might be possible to plot the predictions by feeding input sequences to the trained rolling window model in a different rolling window approach. The model referred to as model1_1 in the notebook generates one prediction at a time, iteratively producing 12 predictions to cover a two-hour forecast window. Additionally, there is an alternative model in the extras section with the output length set to 12, which should output 12 data points in a single prediction. Nonetheless, it is impractical to have a model that predicts 12 data points one at a time, saves each output, and then uses a rolling window approach as the model1_1 does. Ideally, the plots could be created using a rolling window approach for the saved one-by-one prediction model, feeding input sequences in a rolling window manner. While this should be achievable in theory, I was unable to implement it in practice.

Due to limitations in processing power and time efficiency, the current model was trained using data from only one month. While this was necessary to manage computational constraints, training the model on a full year of data from the same location could significantly enhance its generalizability. By incorporating more comprehensive temporal data, the model would likely improve its performance across different seasons and weather conditions, providing more robust predictions. This is supported by research indicating that extensive data collection over longer periods can lead to more accurate solar irradiance forecasts [3].

To enhance the efficiency of training and model performance, feature selection algorithms can be employed. These feature selection algorithms systematically select the most relevant features, reducing the dimensionality of the dataset and improving both training speed and model accuracy. Feature selection is particularly important when dealing with large datasets where many features may be redundant or irrelevant, in this context, the second dataset. For instance, the use of Conditional Mutual Information (CMI) and Gaussian Process Regression (GPR) for feature selection in solar irradiance prediction has been shown to significantly improve model accuracy by reducing redundancy in the feature set [8].

Iterative feature selection could also be beneficial when utilizing additional datasets. This method involves repeatedly selecting features and retraining the model to progressively improve performance. Iterative selection helps in refining the feature set to include only those variables that contribute the most to the predictive power of the model, thereby optimizing the use of available data. Dynamic feature selection using techniques like deep reinforcement learning has been shown to adaptively alter selected features based on changing weather patterns, enhancing prediction accuracy [14].

Detailed hyperparameter tuning was not feasible due to computational constraints. Techniques such as Grid Search or Randomized Search, which systematically explore a wide range of hyperparameters, require substantial processing power and time. Given these limitations, hyperparameters were selected based on domain knowledge and prior research to balance model complexity and performance. Future work could focus on leveraging more advanced tuning techniques and larger computational resources to further refine the model and enhance its predictive accuracy. Employing advanced optimization methods, such as Bayesian optimization, has been shown to significantly improve the performance of machine learning models for solar irradiance prediction [15].

4 | References

[1] KNMI - Daggegevens van het weer in Nederland — knmi.nl. https://www.knmi.nl/nederland-nu/klimatologie/daggegevens. [Accessed 20-07-2024].

[2] Sunshine and radiation - sunshine and radiation at a 10 minute interval - KNMI Data Platform — dataplatform.knmi.nl. https://dataplatform.knmi.nl/dataset/zonneschijnduur-en-straling-1-0. [Accessed 10-07-2024].

[3] Majid Almaraashi. Investigating the impact of feature selection on the prediction of solar radiation in different locations in saudi arabia. *Appl. Soft Comput.*, 66:250–263, 2018.

[4] Ahmad Alzahrani, Pourya Shamsi, Cihan Dagli, and Mehdi Ferdowsi. Solar irradiance forecasting using deep neural networks. *Procedia Computer Science*, 114:304–313, 2017.

[5] L. Benali, G. Notton, A. Fouilloy, C. Voyant, and R. Dizène. Solar radiation forecasting using artificial neural network and random forest methods: Application to normal beam, horizontal diffuse and global components. *Renewable Energy*, 2019.

[6] M. Chaibi, El Mahjoub Benghoulam, Lhoussaine Tarik, M. Berrada, and A. El Hmaidi. Machine learning models based on random forest feature selection and bayesian optimization for predicting daily global solar radiation. *International Journal of Renewable Energy Development*, 2021.

[7] Bixuan Gao, Xiaoqiao Huang, Junsheng Shi, Yonghang Tai, and Jun Zhang. Hourly forecasting of solar irradiance based on ceemdan and multi-strategy cnn-lstm neural networks. *Renewable Energy*, 162:1665–1683, 2020.

[8] Nantian Huang, Ruiqing Li, Lin Lin, Zhiyong Yu, and G. Cai. Low redundancy feature selection of short term solar irradiance prediction using conditional mutual information and gauss process regression. *Sustainability*, 2018.

[9] Xiaoqiao Huang, Qiong Li, Yonghang Tai, Chen Zaiqing, Jun Zhang, Junsheng Shi, Bixuan Gao, and Wuming Liu. Hybrid deep neural model for hourly solar irradiance forecasting. *Renewable Energy*, 171:1041–1060, 2021.

[10] S. Karasu and Aytaç Altan. Recognition model for solar radiation time series based on random forest with feature selection approach. *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, pages 8–11, 2019.

[11] P. Kumari and Durga Toshniwal. Long short term memory–convolutional neural network based deep hybrid approach for solar irradiance forecasting. *Applied Energy*, 295:117061, 2021.

[12] Junho Lee, Wen Wang, F. Harrou, and Ying Sun. Reliable solar irradiance prediction using ensemble learning-based models: A comparative study. *Energy Conversion and Management*, 208:112582, 2020.

[13] J. Liu, M. Cao, D. Bai, and R. Zhang. Solar radiation prediction based on random forest of feature-extraction. *IOP Conference Series: Materials Science and Engineering*, 658, 2019.

[14] C. Lyu, S. Eftekharnejad, S. Basumallik, and Chongfang Xu. Dynamic feature selection for solar irradiance forecasting based on deep reinforcement learning. *IEEE Transactions on Industry Applications*, 59:533–543, 2023.

[15] Saman Soleymani and Shima Mohammadzadeh. Comparative analysis of machine learning algorithms for solar irradiance forecasting in smart grids. *ArXiv*, abs/2310.13791, 2023.

[16] Haixiang Zang, L. Ling, Li Sun, Chen Lilin, Zhi nong Wei, and Guo qiang Sun. Short-term global horizontal irradiance forecasting based on a hybrid cnn-lstm model with spatiotemporal correlations. *Renewable Energy*, 160:26–41, 2020.