

AI-based gap filling of missing irradiance data for CLARA and DARA

Luca Baumann¹

Supervisors: Dr. Wolfgang Finsterle², Dr. Jean-Philippe Montillet², Dr. Magit Haberreiter²

¹ ETH Zurich, Department of Computer Science, Zurich, Switzerland

² Physikalisch Meteorologisches Observatorium Davos and World Radiation Center (PMOD/WRC), Davos, Switzerland

Dated: June 17, 2025

Abstract

Total Solar Irradiance (TSI) plays a vital role in understanding Earth's energy balance. The PMOD/WRC currently operates three active radiometers, including CLARA and DARA, which provide continuous irradiance measurements. However, data gaps can occur due to issues such as sensor malfunctions, and these missing values cannot be recovered retrospectively. This project explores the use of Artificial Intelligence (AI), specifically unidirectional and bidirectional Long Short-Term Memory (LSTM) networks and time series transformers, to reconstruct missing irradiance data. Experimental results demonstrate that both LSTM models perform effectively on irradiance data from CLARA and DARA.

1 Introduction

Total Solar Irradiance (TSI) is an important parameter in understanding and predicting the Earth's climate system. It plays a crucial role in determining the planet's energy balance as the net between the incoming solar- and outgoing electromagnetic radiation at the top of the Earth's atmosphere. PMOD/WRC has built and operates three active radiometers: the PMO6 radiometer as part of the VIGRO instrument on board the SOHO mission, launched in December 1995 (still active); CLARA on board the Norwegian NorSat-1 satellite, in orbit since July 2017; and DARA on board the Chinese FY-3E satellite, launched in July 2021. The instruments provide data that allow scientists to calculate TSI, with CLARA also measuring the terrestrial long-wave outgoing radiation. This project aims to explore the potential of Artificial Intelligence (AI) in performing time-consuming operations more consistently and efficiently, potentially reducing cognitive bias from human operators.

1.1 Project Goals

The main goals of this project were to:

- debug the code started by students of ETH Zurich as part of the Data Science Lab [Barth et al. \(2023\)](#)
- re-implement lost code parts
- investigate and improve its performance for gap filling of TSI measurements using the instrumental housekeeping data
- investigate the applicability for CLARA OLR measurements

1.2 Paper Organization

The next section discusses the necessary background with the most important findings of the project of the ETH Zurich students and what was already available before the project. Additionally, the used models and the most important hyperparameters are discussed. Section 3 discusses the investigated methods for filling gaps using the newly implemented code parts. The results are presented in section 4 and analyzed in section 5. Section 6 summarizes the most important findings of this work and suggests potential improvements and suggestions for further research.

2 Background

This project was started by students of ETH Zurich as part of the Data Science Lab. Unfortunately, most of their code was lost and the preserved fragments had many bugs. The code is shared in this Github Repository. [Barth et al. \(2023\)](#)

2.1 Existing part

The following Python files already existed:

- Debug.ipynb
- models.py
- preprocessing.py
- read_combine.py
- utils.py

2.1.1 Debug.ipynb This is a helper file to debug a simplified version of the read_combine.py file. Since the read_combine.py file works now it is essentially redundant and could be deleted.

2.1.2 models.py This file is responsible for model training. Initially, it only supported training of unidirectional LSTM models for forecasting.

2.1.3 preprocessing.py This file handles data outliers, generates correlation matrices and plots for the housekeeping data. Subsequently, it splits the dataset into training and test dataframes.

2.1.4 read_combine.py This file takes the fits files as input, combines Level1A and Level2A features and outputs a pickle file with the combined data.

2.1.5 utils.py This file contains a helper function called read_file which reads in the column names stored in a txt file.

2.2 Redesigned part

2.2.1 Gap Filling The most important redesigned part in the code is the gap filling functionality. To support a train-/test-split with training on artificial gaps, two additional helper functions in the utils.py file were implemented. These functions are cre-

ate_gap_train_test_split and get_mask. The exact technical details of how to use the functions and what the inputs and outputs are described in the function description in the utils.py file.

2.2.2 Reproducing Results

Results which could be reproduced with the previous code after debugging were:

- Correlation plots of housekeeping features
- DARA validation data plot (Figure 14 from DS Lab report)
- Forecasting of TSI

The exact loss values of the report could not be retrieved, since the exact hyperparameters were not stated in the report and also because the seeds for reproducibility were only set for pytorch.

2.3 New part

2.3.1 Models

To compare the performance of the LSTM model with other models, a bidirectional LSTM (BILSTM), and a neural network (NN) model were implemented. Additionally, a temporal convolutional network (TCN) [Bai et al. \(2018\)](#) and a time series transformer (PatchTST) [Nie et al. \(2023\)](#) model were used.

2.3.2 Preprocessing

Additional preprocessing functionality was provided in the file postprocessing.py. Since measurements for multiple days in the dataset are completely missing, those days were interpolated using the day before and after the gap. By using mean, median or more sophisticated interpolation techniques, the missing features could be filled.

2.3.3 Evaluation

The file predict.py allows for using a previously trained model for further evaluation. Further evaluation includes longer or shorter gaps, less or more data to reconstruct the gap, computation of the train loss, plotting of the predictions on the train dataset, and predictions on the validation set for housekeeping data (if available) or with mean/median imputation.

2.3.4 CLARA

To ensure that the code also runs for CLARA data multiple adaptions had to be performed. The files read_combine.py, preprocessing.py, postprocessing.py, models.py and predict.py had to be adapted to be able to work with the CLARA data. Additionally, the file transform_CLARA_data.py was implemented to combine the housekeeping data with the CLARA OLR data and extract the necessary features for training.

2.4 Hyperparameters

2.4.1 Hidden Size

Describes the dimensionality of the hidden state of the LSTM. For a hidden size of 128 we have $h \in \mathbb{R}^{128}$.

2.4.2 Learning Rate (LR)

The gradient always points in the direction of the steepest ascent. To minimize the loss, the weight update requires a negative sign (LR: α): $w_{t+1} = w_t - \alpha \nabla w_t$

2.4.3 Epochs

In a single epoch a machine learning model processes every datapoint in the dataset once. This processing occurs in batches. For each batch, the loss is computed and backpropagated to update the model parameters. A machine learning model generally requires multiple epochs to train a model.

2.4.4 Dropout

Dropout is a regularization technique in machine learning. It can help the model avoid overfitting the training

data.

2.4.5 Layers

In the case of LSTMs the amount of layers describes how many LSTMs are stacked sequentially.

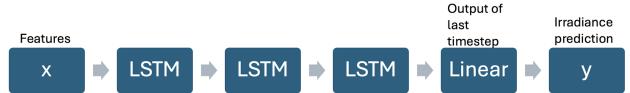


Figure 1. Illustration of the model architecture of a 3-stacked LSTM. The input features are fed through three LSTMs before the final linear layer projects all features on the output.

2.4.6 Window

Instead of using just a single feature row to predict the target, multiple rows can be used by increasing the window hyperparameter. For a window of 16 this means that all the feature rows from timestep 1 to 16 are used to make the first target prediction at timestep 16. Subsequently, all rows from 1 to 17 are used for the target at 17 and so on.

3 Methods

3.1 Models for one day gap and one month data

A first approach consisted in predicting an artificial gap of one day with one month of data for training. One month of data means that 15 days before and after the gap were taken into consideration. The idea behind this approach was to use as few data as possible, without loss in prediction accuracy, to have a minimal amount of training time.

3.2 Models for 2021-2023 data

As a next step, the model was trained on multiple artificial gaps of lengths of one to six days. With this approach, the model was forced to generalize well for each of those gaps and not only a single gap of fixed length.

3.3 Models for 2021-2024 data

For this approach, the training idea was similar as for the one from 2021-2023. However, additional data could allow the model to generalize better, assuming that the new data is representative and diverse.

3.4 CLARA model

The main goal for the CLARA model was to preprocess all the data for model training and ensure that training works. A first result is shown in the respective result section of chapter 4.

4 Results

4.1 Reproduction of Results

4.1.1 Correlation plot of housekeeping data

The plot is shown in Figure 2.

4.1.2 DARA validation data plot

For the configuration from Table 1 the plot is shown in Figure 3.

4.1.3 Forecasting of TSI

For the configuration from Table 1 the plot is shown in Figure 4.

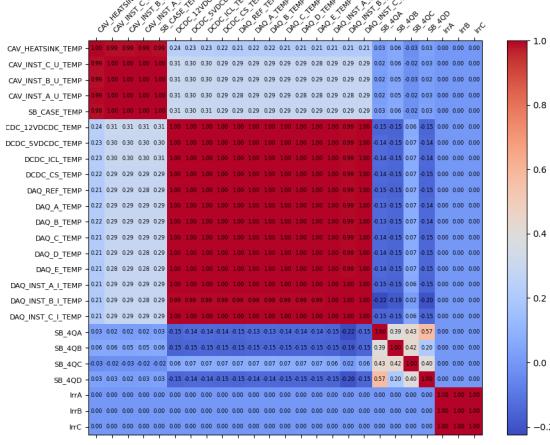


Figure 2. Correlation plot of all considered housekeeping features for DARA.

Table 1

Model configuration with added time features.

Abbreviations: Mod.(Model), Hid. (Hidden Size), LR (Learning Rate), Eps (Epochs), Drop. (Dropout), Lay. (Layers), Win. (Window)

Mod.	Hid.	LR	Eps	Drop.	Lay.	Win.
LSTM	128	$3 \cdot 10^{-3}$	100	0	3	16

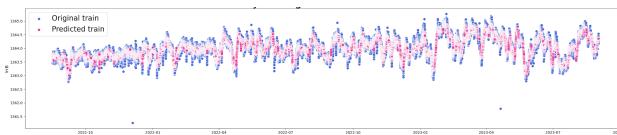


Figure 3. DARA validation data plot. The blue points represent the training data and the pink points show the model's predictions for these points.

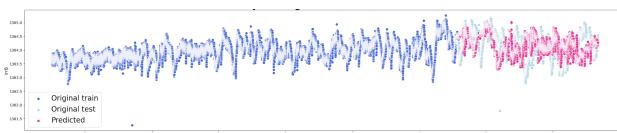


Figure 4. Forecast of Irradiance for DARA. The blue points represent the training data, the light blue points show the measured test data and the pink points describe the model's predictions for these points.

4.2 Models for one day gap and one month data

Table 2

Model configurations for best models without added time features.

Abbreviations: Mod.(Model), Hid. (Hidden Size), LR (Learning Rate), Eps (Epochs), Drop. (Dropout), Lay. (Layers), Win. (Window)

Mod.	Hid.	LR	Eps	Drop.	Lay.	Win.
LSTM	128	$3 \cdot 10^{-3}$	125	0	1	2
BILSTM	32	$3 \cdot 10^{-4}$	100	0	1	16
PatchTST	128	$3 \cdot 10^{-3}$	100	0	5	16

Table 3

Model losses and training time.

Model	Train Loss	Test Loss	Train Time (HH:MM:SS)
LSTM	0.0122	0.0093	00:00:10
BILSTM	0.0119	0.0093	00:00:39
PatchTST	0.0352	0.0353	00:53:53

4.2.1 LSTM For the configuration of Table 2 the plot is shown in Figure 5 and the losses are shown in Table 3.

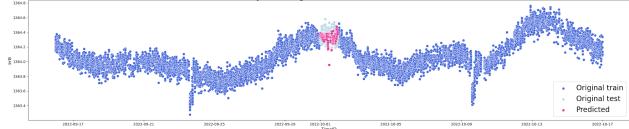


Figure 5. Forecast of the LSTM model for an artificial gap of one day and one month training data. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

4.2.2 BILSTM For the configuration of Table 2 the plot is shown in Figure 6 and the losses are shown in Table 3.

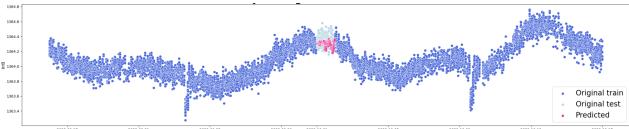


Figure 6. Forecast of the BILSTM model for an artificial gap of one day and one month training data. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

4.2.3 PatchTST Nie et al. (2023) For the configuration of Table 2 the plot is shown in Figure 7 and the losses are shown in Table 3.

4.3 Models for 2021-2023 data evaluated on housekeeping data

4.3.1 LSTM For the configuration in Table 4 the total losses are shown in Table 5 and the losses for each gap are shown in Table 6. The predictions for L1 to L6 are visualized in figures Figure 8 to Figure 13 respectively.

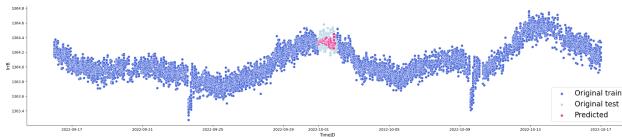


Figure 7. Forecast of the PatchTST model for an artificial gap of one day and one month training data. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

Table 4

Model configurations for best models with added time features.
Abbreviations: Mod.(Model), Hid. (Hidden Size), LR (Learning Rate), Eps (Epochs), Drop. (Dropout), Lay. (Layers), Win. (Window)

Mod.	Hid.	LR	Eps	Drop.	Lay.	Win.
LSTM	32	$3 \cdot 10^{-3}$	60	0	1	16
BILSTM	128	$3 \cdot 10^{-3}$	80	0	2	16

Table 5

Model losses and training time for the models in Table 4.

Model	Train Loss	Test Loss	Train Time (HH:MM:SS)
LSTM	0.0085	0.0213	00:07:05
BILSTM	0.0056	0.0218	01:39:46

Table 6

Model losses for each gap for the models in Table 4. Abbreviations: Mod. (Model), Li (Loss on gap i for $i \in \{1, \dots, 6\}$). Info: Gap 1 is from 2021-10-27 to 2021-10-27, Gap 2 is from 2022-03-05 to 2022-03-06, Gap 3 is from 2022-08-17 to 2022-08-22, Gap 4 is from 2023-01-03 to 2023-01-05, Gap 5 is from 2023-05-27 to 2023-05-31, Gap 6 is from 2023-10-23 to 2023-10-26.

Mod.	L1	L2	L3	L4	L5	L6
LSTM	0.0069	0.0114	0.0225	0.0124	0.0523	0.0130
BILSTM	0.0120	0.0103	0.0406	0.0110	0.0419	0.0103

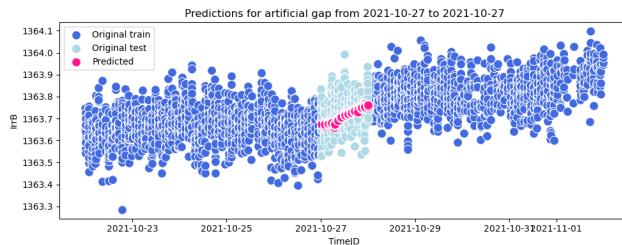


Figure 8. Forecast of the LSTM model for an artificial gap of 1 day at 2021-10-27. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

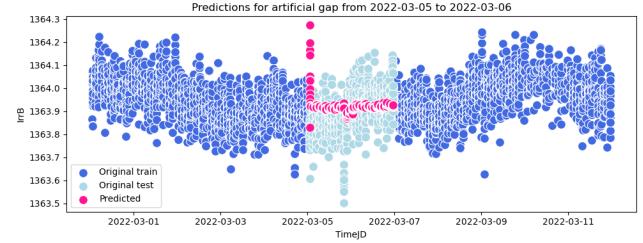


Figure 9. Forecast of the LSTM model for an artificial gap of 2 days from 2022-03-25 to 2022-03-06. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

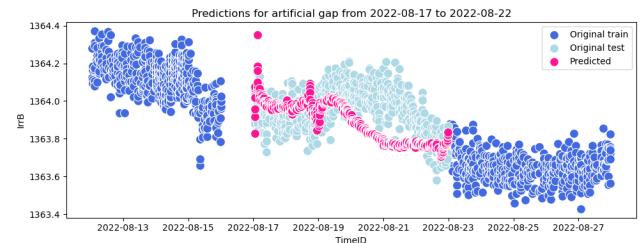


Figure 10. Forecast of the LSTM model for an artificial gap of 6 days from 2022-08-17 to 2022-08-22. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

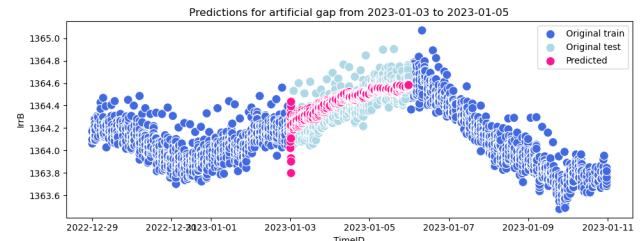


Figure 11. Forecast of the LSTM model for an artificial gap of 3 days from 2023-01-03 to 2023-01-05. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

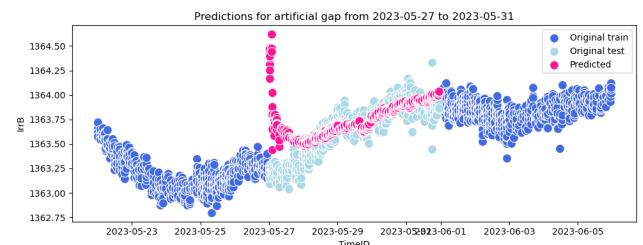


Figure 12. Forecast of the LSTM model for an artificial gap of 5 days from 2023-05-27 to 2023-05-31. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

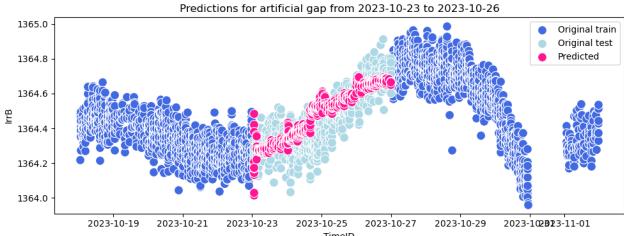


Figure 13. Forecast of the LSTM model for an artificial of gap 4 days from 2023-10-23 to 2023-10-26. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

4.3.2 BILSTM For the configuration in Table 4 the total losses are shown in Table 5 and the losses for each gap are shown in Table 6. The predictions for L1 to L6 are visualized in figures Figure 14 and Figure 19 respectively.

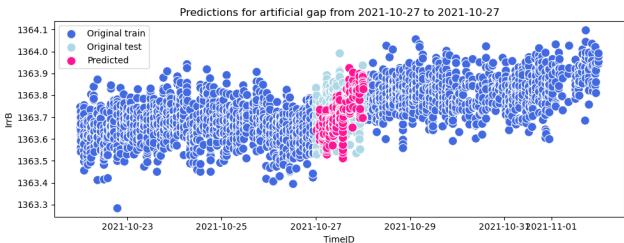


Figure 14. Forecast of the BILSTM model for an artificial gap of 1 day at 2021-10-27. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

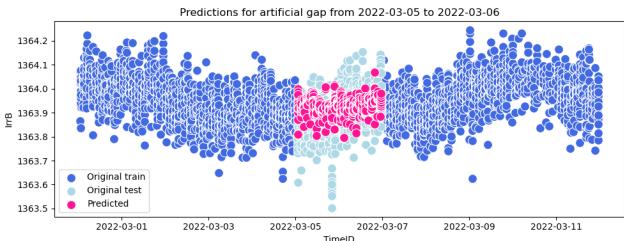


Figure 15. Forecast of the BILSTM model for an artificial gap of 2 days from 2022-03-05 to 2022-03-06. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

4.4 Models for 2021-2023 data evaluated with mean imputed features

Table 7

Model losses and training time for the models in Table 4.

Model	Train Loss	Test Loss	Train Time (HH:MM:SS)
LSTM	0.0085	0.0223	00:07:05
BILSTM	0.0056	0.0297	01:39:46

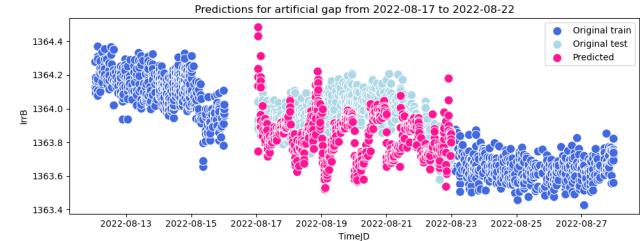


Figure 16. Forecast of the BILSTM model for an artificial gap of 6 days from 2022-08-17 to 2022-08-22. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

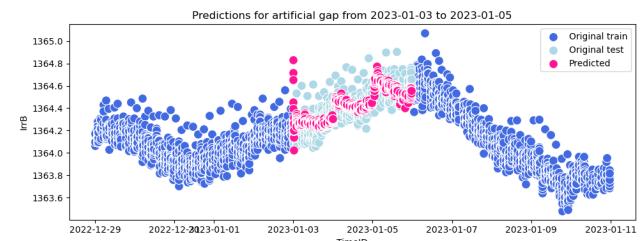


Figure 17. Forecast of the BILSTM model for an artificial gap of 3 days from 2023-01-03 to 2023-01-05. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

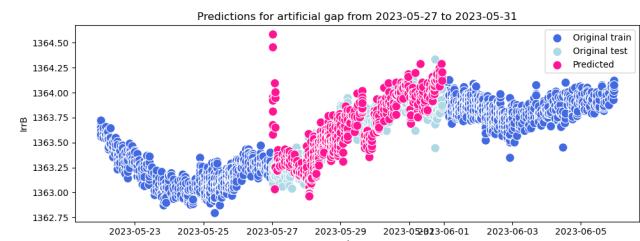


Figure 18. Forecast of the BILSTM model for an artificial gap of 5 days from 2023-05-27 to 2023-05-31. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

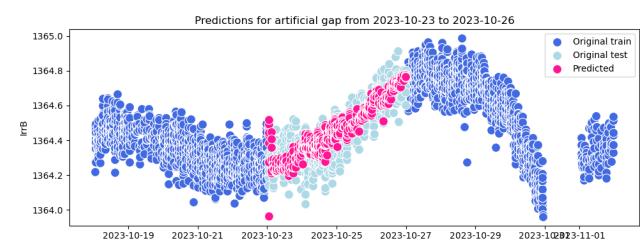


Figure 19. Forecast of the BILSTM model for an artificial gap of 4 days from 2023-10-23 to 2023-10-26. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

Table 8

Model losses for each gap for the models in Table 4. Abbreviations: Mod. (Model), Li (Loss on gap i for $i \in \{1, \dots, 6\}$). Info: Gap 1 is from 2021-10-27 to 2021-10-27, Gap 2 is from 2022-03-05 to 2022-03-06, Gap 3 is from 2022-08-17 to 2022-08-22, Gap 4 is from 2023-01-03 to 2023-01-05, Gap 5 is from 2023-05-27 to 2023-05-31, Gap 6 is from 2023-10-23 to 2023-10-26.

Mod.	L1	L2	L3	L4	L5	L6
LSTM	0.0091	0.0115	0.0170	0.0116	0.0489	0.0272
BILSTM	0.0127	0.0101	0.0378	0.0106	0.0827	0.0143

4.4.1 LSTM For the configuration in Table 4 the total losses are shown in Table 7 and the losses for each gap are shown in Table 8. The predictions for L1 to L6 are visualized in figures Figure 20 and Figure 25 respectively.

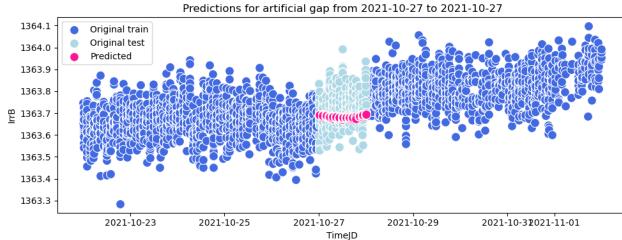


Figure 20. Forecast of the LSTM model for an artificial gap of 1 day at 2021-10-27. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

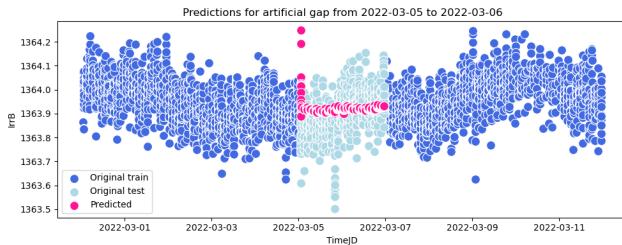


Figure 21. Forecast of the LSTM model for an artificial gap of 2 days from 2022-03-05 to 2022-03-06. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

4.4.2 BILSTM For the configuration in Table 4 the total losses are shown in Table 7 and the losses for each gap are shown in Table 8. The predictions for L1 to L6 are visualized in figures Figure 26 to Figure 31 respectively.

4.5 Models for 2021-2023 data evaluated with median imputed features

The plots for the different gaps only showed slight variations compared to the ones with mean imputed features and are not shown here.

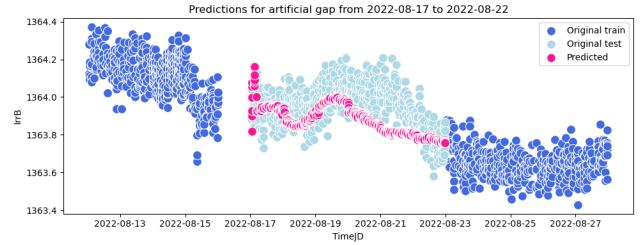


Figure 22. Forecast of the LSTM model for an artificial gap of 6 days from 2022-08-17 to 2022-08-22. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

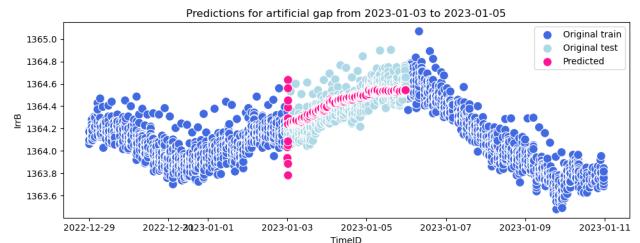


Figure 23. Forecast of the LSTM model for an artificial gap of 3 days from 2023-01-03 to 2023-01-05. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

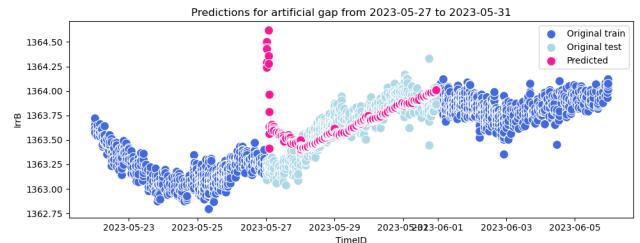


Figure 24. Forecast of the LSTM model for an artificial gap of 5 days from 2023-05-27 to 2023-05-31. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

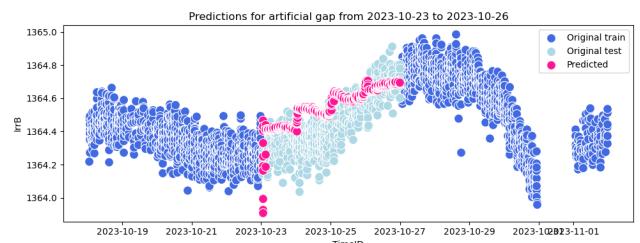


Figure 25. Forecast of the LSTM model for an artificial gap of 4 days from 2023-10-23 to 2023-10-26. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

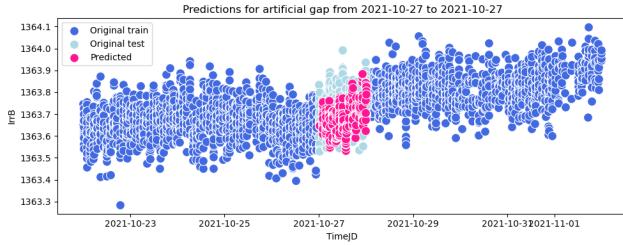


Figure 26. Forecast of the BiLSTM model for an artificial gap of 1 day at 2021-10-27. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

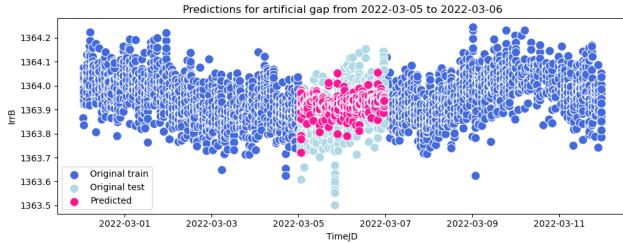


Figure 27. Forecast of the BiLSTM model for an artificial gap of 2 days from 2022-03-05 to 2022-03-06. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

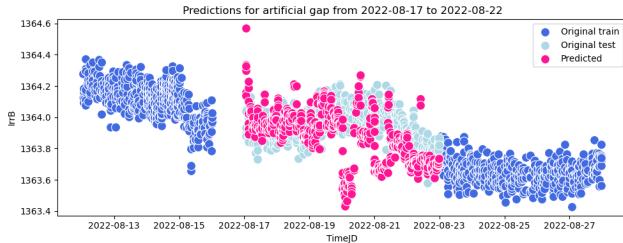


Figure 28. Forecast of the BiLSTM model for an artificial gap of 6 days from 2022-08-17 to 2022-08-22. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

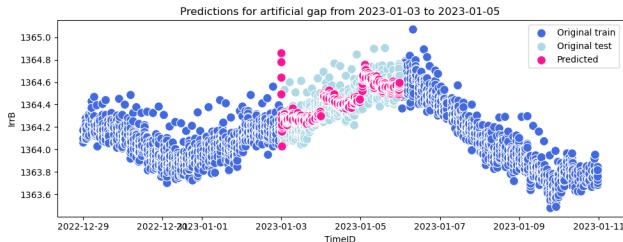


Figure 29. Forecast of the BiLSTM model for an artificial gap of 3 days from 2023-01-03 to 2023-01-05. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

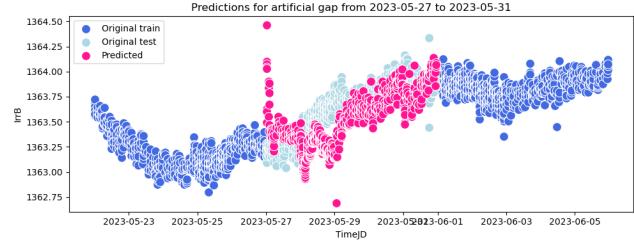


Figure 30. Forecast of the BiLSTM model for an artificial of 5 days 2023-05-27 to 2023-05-31. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

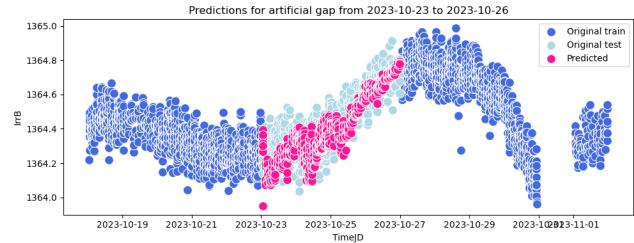


Figure 31. Forecast of the BiLSTM model for an artificial gap of 4 days from 2023-10-23 to 2023-10-26. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

Table 9
Model losses and training time for the models in Table 4.

Model	Train Loss	Test Loss	Train Time (HH:MM:SS)
LSTM	0.0085	0.0266	00:07:05
BiLSTM	0.0056	0.0370	01:39:46

Table 10

Model losses for each gap for the models in Table 4. Abbreviations: Mod. (Model), Li (Loss on gap i for $i \in \{1, \dots, 6\}$). Info: Gap 1 is from 2021-10-27 to 2021-10-27, Gap 2 is from 2022-03-05 to 2022-03-06, Gap 3 is from 2022-08-17 to 2022-08-22, Gap 4 is from 2023-01-03 to 2023-01-05, Gap 5 is from 2023-05-27 to 2023-05-31, Gap 6 is from 2023-10-23 to 2023-10-26.

Mod.	L1	L2	L3	L4	L5	L6
LSTM	0.0087	0.0115	0.0167	0.0116	0.0579	0.0414
BiLSTM	0.0152	0.0101	0.0340	0.0106	0.0905	0.0481

4.6 Predictions for BILSTM trained on 2021-2023 data on missing data

The results are shown in figures Figure 32 to 34.



Figure 32. Forecast of the BILSTM model for the missing data from 2021. The features were reconstructed using mean imputation. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

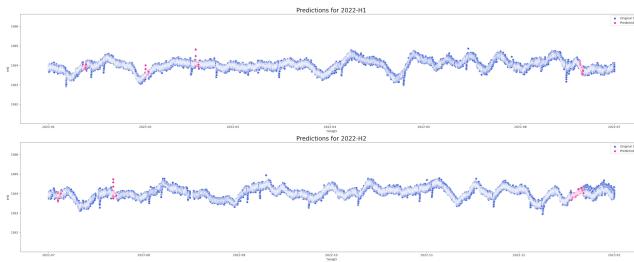


Figure 33. Forecast of the BILSTM model for the missing data from 2022. The features were reconstructed using mean imputation. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

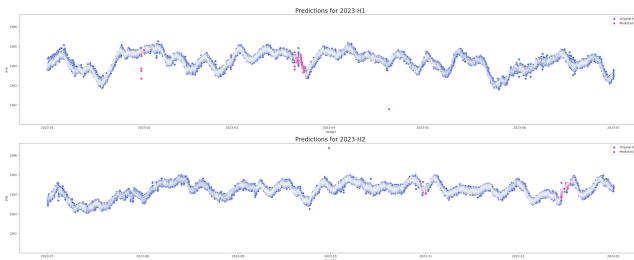


Figure 34. Forecast of the BILSTM model for the missing data from 2023. The features were reconstructed using mean imputation. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

4.7 Models for 2021-2024 data

Table 11

Model configurations for best models with added time features.
Abbreviations: Mod.(Model), Hid. (Hidden Size), LR (Learning Rate), Eps (Epochs), Drop. (Dropout), Lay. (Layers), Win. (Window)

Mod.	Hid.	LR	Eps	Drop.	Lay.	Win.
LSTM	128	$3 \cdot 10^{-4}$	100	0	2	16
BILSTM	128	$3 \cdot 10^{-3}$	100	0	3	16

Table 12
Model losses and training time for the models in Table 11.

Model	Train Loss	Test Loss	Train Time (HH:MM:SS)
LSTM	0.0054	0.0455	02:06:08
BILSTM	0.0088	0.0464	05:37:35

Table 13

Model losses for each gap for the models in Table 11. Abbreviations: Mod. (Model), Li (Loss on gap i for $i \in \{1, \dots, 6\}$). Info: Gap 1 is from 2021-11-29 to 2021-11-29, Gap 2 is from 2022-06-24 to 2022-06-25, Gap 3 is from 2023-01-07 to 2023-01-12, Gap 4 is from 2023-09-06 to 2023-09-08, Gap 5 is from 2024-02-13 to 2024-02-17, Gap 6 is from 2024-09-30 to 2024-10-03.

Mod.	L1	L2	L3	L4	L5	L6
LSTM	0.0104	0.0108	0.0418	0.0159	0.0439	0.0911
BILSTM	0.0143	0.0111	0.0307	0.0116	0.0609	0.0909

4.7.1 LSTM For the configuration in Table 11 the total losses are shown in Table 12 and the losses for each gap are shown in Table 13. The predictions for L1 to L6 are visualized in figures Figure 35 to Figure 40 respectively.

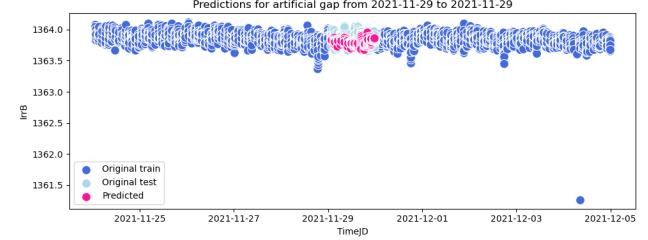


Figure 35. Forecast of the LSTM model for an artificial gap of 1 day at 2021-11-29. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

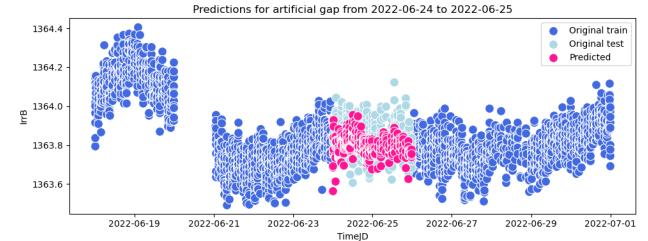


Figure 36. Forecast of the LSTM model for an artificial gap of 2 days from 2022-06-24 to 2022-06-25. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

4.7.2 BILSTM For the configuration in Table 11 the total losses are shown in Table 12 and the losses for each gap are shown in Table 13. The predictions for L1 to L6 are visualized in figures Figure 41 to Figure 46 respectively.

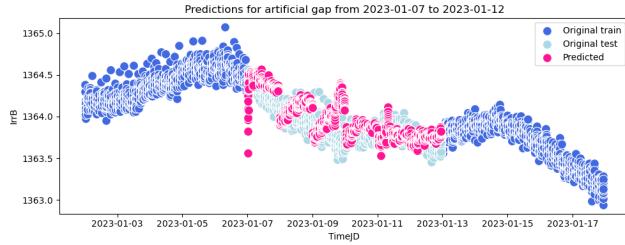


Figure 37. Forecast of the LSTM model for an artificial gap of 6 days from 2023-01-07 to 2023-01-12. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

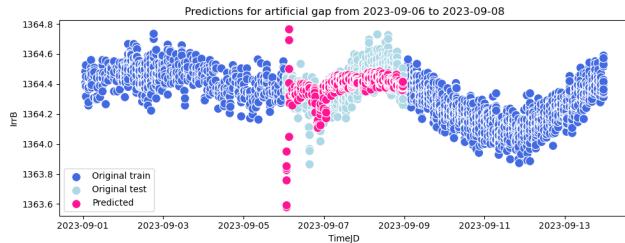


Figure 38. Forecast of the LSTM model for an artificial gap of 3 days from 2023-09-06 to 2023-09-08. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

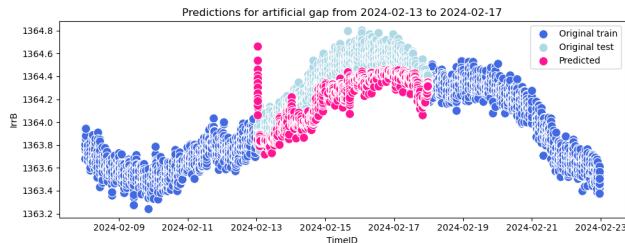


Figure 39. Forecast of the LSTM model for an artificial gap of 5 days from 2024-02-13 to 2024-02-17. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

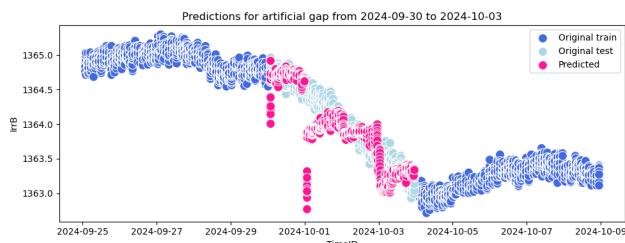


Figure 40. Forecast of the LSTM model for an artificial of 4 days from 2024-09-30 to 2024-10-03. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

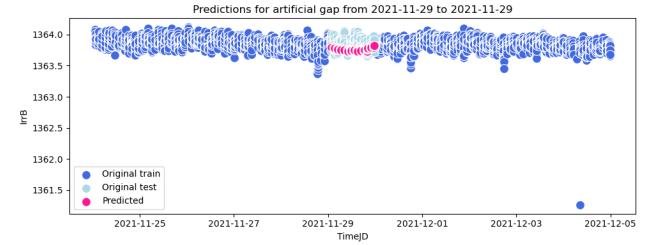


Figure 41. Forecast of the BiLSTM model for an artificial gap of 1 days at 2021-11-29. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

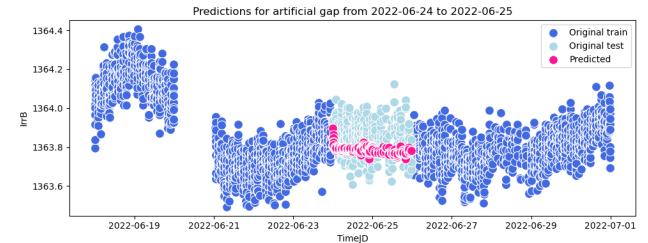


Figure 42. Forecast of the BiLSTM model for an artificial gap of 2 days from 2022-06-24 to 2022-06-25. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

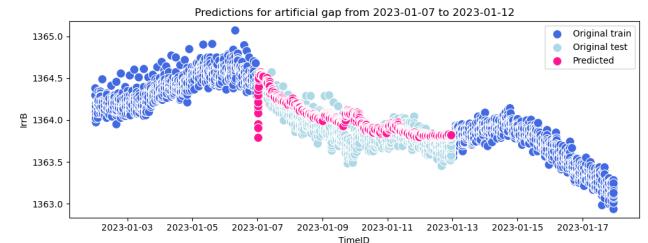


Figure 43. Forecast of the BiLSTM model for an artificial gap of 6 days from 2023-01-07 to 2023-01-12. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

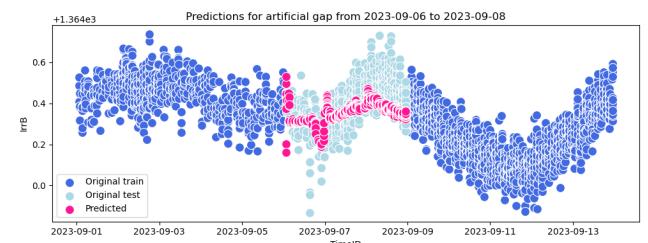


Figure 44. Forecast of the BiLSTM model for an artificial gap of 3 days from 2023-09-06 to 2023-09-08. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

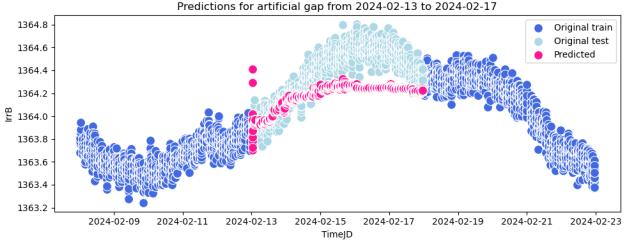


Figure 45. Forecast of the BILSTM model for an artificial gap of 5 days from 2024-02-13 to 2024-02-17. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

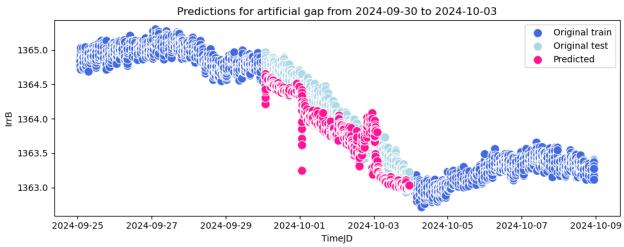


Figure 46. Forecast of the BILSTM model for an artificial gap of 4 days from 2024-09-30 to 2024-10-03. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

4.8 First CLARA Results

4.8.1 Data preparation The CLARA data with combined housekeeping features and OLR data were filtered for a solar zenith angle greater than 120. Subsequently, all CLARA radiance values greater than 1000 and smaller than -100 were classified as outliers and excluded. The data distribution is shown in Figure 47. After filtering outliers from the data, the data used for training is shown in Figure 48. For the filtering a rolling window of width 50 was taken and the median was computed. Values which deviated by more than 238 from this median value were classified as outliers and are shown in red in Figure 48.

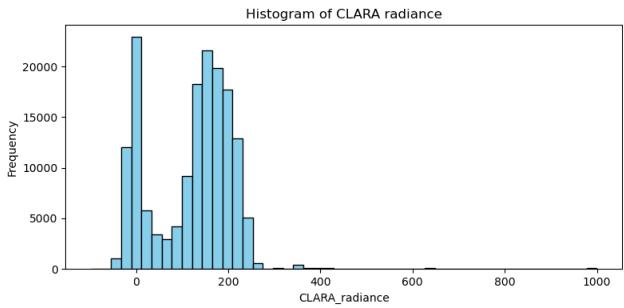


Figure 47. Histogram of the filtered CLARA radiance data.

4.9 First model

For the configuration in Table 14 the total losses are shown in Table 15 and the losses for each gap are shown in Table 16. The predictions for L1 to L6 are visualized in figures Figure 49 to Figure 54

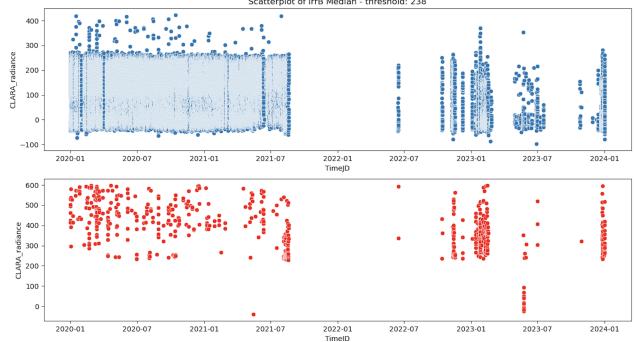


Figure 48. Visualization of the CLARA radiance data after median based filtering.

respectively.

Table 14

Model configurations for first model with added time features.

Abbreviations: Mod. (Model), Hid. (Hidden Size), LR (Learning Rate), Eps (Epochs), Drop. (Dropout), Lay. (Layers), Win. (Window)

Mod.	Hid.	LR	Eps	Drop.	Lay.	Win.
LSTM	128	$3 \cdot 10^{-3}$	35	0	1	16

Table 15

Model losses and training time for the model in Table 14.

Model	Train Loss	Test Loss	Train Time (HH:MM:SS)
LSTM	1306.1296	1215.8917	00:04:29

Table 16

Model losses for each gap for the model in Table 14. Abbreviations: Mod. (Model), Li (Loss on gap i for $i \in \{1, \dots, 6\}$). Info: Gap 1 is from 2020-03-01 to 2020-03-01, Gap 2 is from 2020-06-24 to 2020-06-25, Gap 3 is from 2020-10-11 to 2020-10-14, Gap 4 is from 2020-10-16 to 2020-10-17, Gap 5 is from 2021-01-28 to 2021-01-30, Gap 6 is from 2021-05-28 to 2021-06-01

Mod.	L1	L2	L3	L4	L5	L6
LSTM	912.47	1224.88	1302.54	939.00	1323.39	1304.69

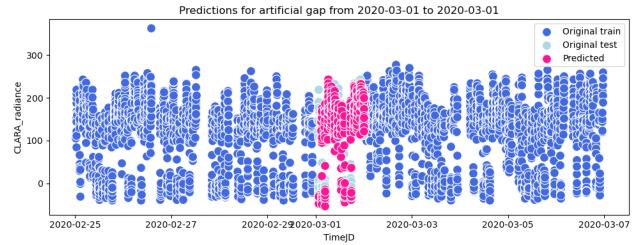


Figure 49. Forecast of the LSTM model for an artificial gap of 1 days at 2020-03-01. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

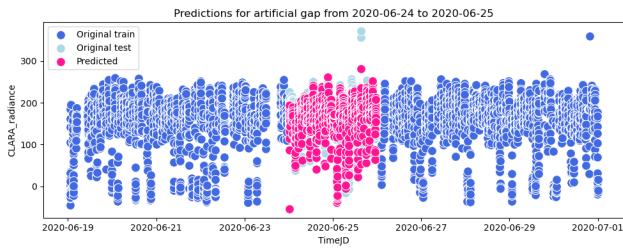


Figure 50. Forecast of the LSTM model for an artificial gap of 2 days from 2020-06-24 to 2020-06-25. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

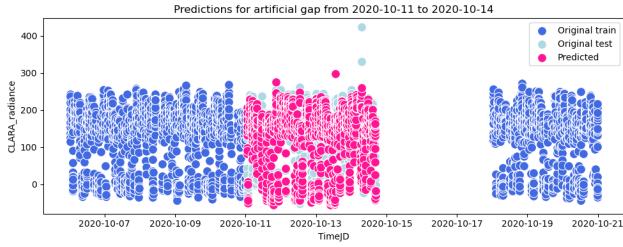


Figure 51. Forecast of the LSTM model for an artificial gap of 4 days from 2020-10-11 to 2020-10-14. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

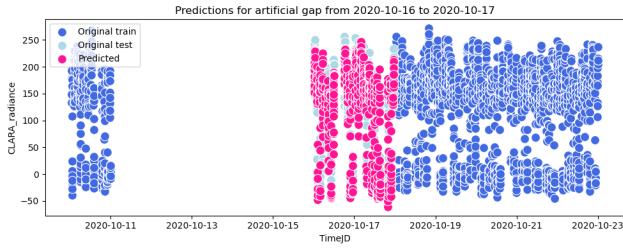


Figure 52. Forecast of the LSTM model for an artificial gap of 2 days from 2020-10-16 to 2020-10-17. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

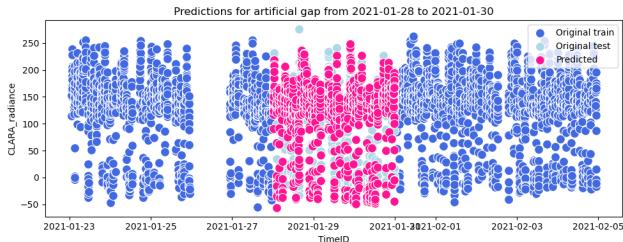


Figure 53. Forecast of the LSTM model for an artificial gap of 3 days from 2021-01-28 to 2021-01-30. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

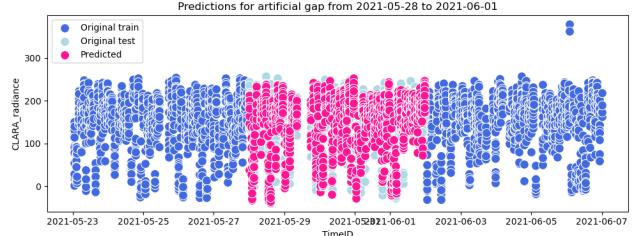


Figure 54. Forecast of the LSTM model for an artificial gap of 5 days from 2021-05-28 to 2021-06-01. The blue points represent the training data. The light blue points show the measured data for the artificial gap and the pink point show the model's prediction for these points.

5 Discussion

5.1 Reproduction of Results

5.1.1 Correlation plot The correlation plot from Figure 2 could be easily reconstructed after debugging the code. The results are, apart from the order of the features, identical to the plot in the report.

5.1.2 DARA validation data plot Figure 3 can be confusing. It visualizes the training data in blue and the prediction of the LSTM model for the training data in pink. However, the model has seen all this data during the training process. Hence, this plot cannot be seen as a validation of the model's prediction capability. Even a much simpler model could do the same. To measure the model's prediction capability the predictions on a held out validation or test dataset with artificial gaps should be evaluated. This analysis is done in the following subsection.

5.1.3 Irradiance forecasting Figure 4 shows that training models to forecast irradiance data is possible. However, this was not the focus of this project and to obtain meaningful results the model would have to be tuned appropriately.

5.2 Models for one day gap and one month data

Table 3 shows that both the LSTM and the BiLSTM models achieved a test loss of 0.0093 on held-out data, significantly outperforming the PatchTST model, which had a higher test loss of 0.0353. In terms of computational efficiency, the LSTM and BiLSTM required only 10 and 39 seconds to train, compared to over 53 minutes for PatchTST. Visually, Figures 5 to 7 show no big differences except for the single outlier in the prediction of the LSTM model in Figure 5.

While the lower test loss and faster training time favor the recurrent models, evaluating model generalization requires examining the gap between training and test loss. PatchTST demonstrates a very small relative increase (0.28%) from training to test loss, indicating strong generalization. In contrast, both LSTM and BiLSTM had test losses over 20% lower than their respective training losses. This is unusual and may be explained by the smaller size of the test dataset compared to the train dataset, which can introduce higher variance and potentially favorable evaluation results for the LSTM and BiLSTM.

Although some practitioners consider a test loss up to 50 percent higher than the training loss as acceptable, this is not a uni-

versally applicable rule. More importantly, a test loss significantly lower than the training loss may also indicate underfitting on the training data, which could be improved by using more complex models or by using more training data, as this also induces further variability.

Another argument for the transformer-based model is the capability to better capture long-term trends compared to the LSTM based models which can also be beneficial for gaps of more consecutive days.

5.3 Models for 2021-2023 data evaluated on housekeeping data

Adding time features information for the year, month, day, hour and minute to the housekeeping features significantly improved the performance of the models. This aligns with the findings of the ETH students as part of their DS Lab project Barth et al. (2023).

Table 5 shows that the LSTM had a slightly lower loss of 0.0213 compared to the 0.0218 for the BILSTM on held-out test data. In terms of training time, the LSTM required only 7 minutes compared to the 1 hour and 39 minutes for the BILSTM. Both of these factors favor the LSTM over the BILSTM. However, Figures 8 to 13 show that the predictions of the LSTM closely align with the mean of the data and the model is not able to accurately capture the complex patterns of the underlying time series even if the test loss is smaller than the one of the BILSTM. On the other hand, Figures 14 to 19 show that BILSTM is better at capturing the complexity of the underlying time series. This suggests that lower test loss on its own does not imply better performance on unseen data. The model needs to have a certain complexity to be able to capture the underlying patterns of the time series efficiently.

A potential limitation of the current models is the presence of outliers at the beginning of their prediction sequences. These anomalies are particularly evident in Figures 9 to 12 and 16 to 18. A potential explanation for this behavior might be attributed to the window-based processing of the data. Specifically, during inference, the initial windows of the validation set are used to generate predictions, but these inputs lack access to the internal cell states of the (Bi)LSTM that were established during training. As a result the model starts from a cold state, leading to less accurate predictions at the sequence boundaries. A possible mitigation strategy is to include the final *window* – 1 data points from the training set as a prefix to the validation set. This would allow the model to warm up and carry forward the contextual information into the prediction phase. However, such an approach risks introducing data leakage because it would compromise the strict separation between training and validation data.

Figure 18 illustrates that even though the BILSTM model is generally more effective at capturing the complexity of the underlying time series compared to the LSTM model, its predictions occasionally converge toward the mean of the data. A potential explanation for this behavior might lie in the loss function used. Specifically, mean squared error (MSE) was used during both training and inference. Hence, in regions where the model is highly uncertain about the true values, it tends to adopt a more conservative strategy by predicting values close to the mean. This minimizes the loss, as large deviations from the true values would incur a significantly higher penalty under the MSE criterion. A potential alternative loss function would be the Huber loss, which is a smooth alternative to the mean absolute error loss and combines

both advantages. With this loss the model would be less sensitive to outliers than with MSE such that it is less incentivized to always predict the mean in regions of large uncertainty.

5.4 Models for 2021-2023 data evaluated with mean imputed features

When evaluating the models with mean imputed features, and comparing the results from Table 5 to Table 7, both the test losses of the LSTM (Table 5: 0.0213 to Table 7: 0.0223) and the BILSTM (Table 5: 0.0218 to Table 7: 0.0297) increase. This is important because features for the missing data (no housekeeping features available) were reconstructed in the same way. Hence, in terms of loss values the results are expected to have a higher loss as when housekeeping features were available. However, this is not necessarily a bad sign because the previous subsection showed that a LSTM model had a lower test loss than the BILSTM model, but it was not able to accurately capture the underlying pattern of the time series.

By comparing the Figures 8 to 19 without mean imputed features to Figures 20 to 31 with mean imputed features only four of the six figure pairs show large visually noticeable deviations. When comparing Figure 8 to Figure 20 the increasing trend of the prediction in Figure 8 is replaced by a constant prediction in Figure 20. Subsequently, comparing Figure 10 to Figure 22, the fluctuation around 2022-08-19 in Figure 10 is replaced by a smoother prediction in Figure 22. Additionally, the high starting values in Figure 10 are replaced by starting values closer to the mean in Figure 22. By comparing Figure 13 to Figure 25 it is evident that the predictions of Figure 13 better capture the underlying time series than the predictions in Figure 25. The predictions in Figure 25 are mostly too high and are not able to capture the radiation band accurately. Finally, when comparing Figure 16 to Figure 28 the wild fluctuations in Figure 16 are replaced by more scattered prediction in Figure 28 which are better capturing the underlying patterns of the time series.

The previous results can also directly be observed in the loss values of Tables 6 and Table 8. For the predictions of the LSTM model in Figures 8 and Figure 20, the losses increase from 0.0069 with housekeeping features to 0.0091 with mean imputed features. For the same models predictions in Figures 10 and Figure 22, the loss decreases from 0.0225 to 0.0170 when replacing the housekeeping features with mean imputed features. Again for the LSTMs predictions in Figures 13 and Figure 25 the losses increase from 0.0130 to 0.0272 when replacing the housekeeping features with mean imputed features. For the BILSTM models predictions in Figures 16 and Figure 28 the losses decrease from 0.0406 to 0.0378 when replacing the housekeeping features with mean imputed features. Most of the other losses closely align such that no big visual differences are present. A remarkable exception is the prediction of the BILSTM model shown in Figures 18 and 30. The loss value in Figure 18 was 0.0419 and the one in Figure 30 was 0.0827. While the plots show that the predictions with housekeeping features are better around the date 2023-05-28, no other big visual differences are present such that the magnitude in loss deviation is a bit surprising. To explain this finding, further analysis would be required. A suitable starting point to investigate this finding would be to plot the residuals of the predictions and find a way to mitigate this problem.

5.5 Models for 2021-2023 data evaluated with median imputed features

When evaluating the models with median imputed features, and comparing the results from Table 5 and Table 7 to Table 11, both the test losses of the LSTM (Table 5: 0.0213 and Table 7: 0.0223 to Table 11): 0.0266 and the BiLSTM (Table 5: 0.0218 and Table 7: 0.0297 to Table 11) increase again. This suggests that mean imputation of the features is better suited for our problem than median imputation of the features.

As mentioned before, visually, the predictions with mean and median imputation showed only slight deviations. This can also be seen when comparing the results per gap from Table 8 to Table 12. The largest deviations for both the predictions of the LSTM and BiLSTM model are on gap 6 where the losses with median imputation are almost twice as large as for the LSTM model with mean imputation and more than three times bigger for the BiLSTM model with median imputation. This again suggests that it is preferable to use mean imputed features over median imputed features for prediction making.

5.6 Predictions of BiLSTM trained on 2021-2023 data for missing data gaps

Figures 32 to 34 show that the BiLSTM model with mean imputed features is able to accurately reconstruct the missing DARA data. The LSTM model was also able to reproduce the values, however, as already mentioned before, the predictions of the LSTM model were not able to accurately capture the complex underlying pattern of the time series and rather converged to the mean prediction.

5.7 Models for 2021-2024 data

Table 9 shows that the LSTM model (0.0455) has slightly lower losses on unseen data than the BiLSTM model (0.0464). Additionally, the LSTM model had a training time of two hours and six minutes which is less than half of the five hours and 37 minutes of the BiLSTM. This aligns with the findings for the models trained on the 2021-2023 data.

Figures 35 to 46 demonstrate that generally, the LSTM model is better at capturing the underlying patterns of the time series and the predictions of the BiLSTM model often align with the mean of the data. This finding is surprising because for the 2021-2023 data it was exactly the other way around. However, this might be explained by the fact that there was not enough time to try out the same amount of configurations for the BiLSTM model as for the LSTM model due to the higher training time. Hence, in future work the tuning of the BiLSTM model could still be improved.

Similar to the findings of the 2021-2023 trained models, many of the predictions in Figures 35 to 46 have severe prediction outliers in the beginning. To mitigate this problem a similar strategy as discussed before can be used.

Table 10 shows that the losses for gap 1, 2, and 4 were similar for both the LSTM and BiLSTM. However, as discussed before, visually the LSTM model better captured the underlying pattern of the time series data. Additionally, on gaps 3, 5 and 6, both models had significantly higher losses than on the other gaps. The predictions for the LSTM model are visualized in Figures 37, 39 and 40 and the ones of the BiLSTM model in Figures 43, 45 and 46. For the LSTM model, it can be observed in Figures 37 and 40

that the predictions exhibit noticeable jumps during periods when the radiation levels are decreasing. A potential reason for this behavior is the model's delayed adaption to rapid temporal changes, which can occur when the input sequence does not provide sufficient historical context. This issue could potentially be mitigated by increasing the window size of the model, allowing it to access a broader temporal context and better capture the dynamics of the sharp decreases in radiation. Alternatively, a TCN model [Bai et al. \(2018\)](#) or time series transformer model [Nie et al. \(2023\)](#) could be used to solve this problem since both of these models are better suited in capturing long range dependencies in the data. Similarly, Figures 39, 43, 45 and 46 show that the model either underpredicts or overpredicts the true values of the radiation data. This suggests that the models struggle to capture the highly non-linear and abrupt patterns of the underlying data. To mitigate this problem, the complexity of the models could be further increased by increasing the hidden size of the model and if necessary also the number of LSTM layers.

5.8 First CLARA model

Figure 47 shows that most of the CLARA data is distributed around two bands. The first band is centered around 0 and the second band is centered around 180. This is also visible in Figures 49 to 54.

The model described in Table 14 is just a first try and would have to be tuned appropriately. The number of epochs were reduced to 35 for faster training, but similarly to the DARA models 100 epochs might be an appropriate starting point for further tuning. Additionally, the number of layers were reduced to one. For machine learning it is generally always beneficial to start with a simple model (only one layer and a moderate hidden size) and then to steadily increase first the hidden size and if necessary the number of layers if further model complexity is required to capture the nonlinearity in the input data. More details to improve the model are provided in the following subsection.

Tables 15 and 16 show that both the train and test losses for the CLARA data are much larger than the ones for the DARA data. The reason for this large deviation lies in the range of input data. The DARA irradiance band has a thickness of roughly 0.4 radian points while the thickness of the CLARA data is almost 400 (from roughly 300 to roughly -100). This can lead to large loss values with the mean squared error loss since for example a ground truth value of 250 and a prediction of 150 results in a residual error of $(250 - 150)^2 = 10000$.

Visually, Figures 49 to 54 show that the model is able to accurately capture the patterns of the underlying time series. This is a promising first model and highlights the potential of AI-based gap filling with CLARA data.

An important point which has not been tested so far is the mean imputation of the test dataframe (postprocessing.py). Potentially, the code in this part would still have to be adapted to allow for mean- or median-based imputation of the input features to predict the missing data. Additionally, Figure 48 demonstrated that the second part of the time series data has much less data points with housekeeping features. In this region it might be beneficial to also artificially enhance the amount of datapoints using one of the previously discussed imputation variants.

5.9 Recommendation for future work

5.9.1 General Due to the high training times for the transformer-based model (over 24 hours), no suitable configuration was found. Additionally, there was insufficient time to train the temporal convolutional network [Bai et al. \(2018\)](#). However, as both of these models are better suited for longer gaps in the data, this is a suitable starting point for further research on both the CLARA and DARA datasets. Moreover, these models can also help mitigate the problem of jumps in the forecasts during abrupt decay of the irradiance data.

For this project the ideal model configuration from the project of the ETH students was taken as a baseline and further tuning was done around this baseline. However, since much of the code was lost it is not evident that the given hyperparameters were the optimal ones. Additionally, when changing the amount of training data or the data set the previous analysis showed that the ideal configurations changes. Hence, it is fundamental to appropriately tune the models for the required task. When training a machine learning model one usually starts with a simple model (for example one LSTM layer and a moderate amount of hidden neurons). If additional model complexity is required to accurately capture the nonlinearity of the data, usually the number of neurons in case of a neural network or the hidden size in case of an LSTM based model are increased first, before potential additional layers are added. When the number of layers are increased, the hidden size should be reduced again to a moderate size, before increasing it again if necessary. The number of epochs should be chosen appropriately such that the train losses no longer change as much as in the beginning. This is the region in which the losses converge. However, training losses are expected to further decrease when training is continued, but training the model too long leads to overfitting. Additionally, if the test losses are much larger than the train losses regularization might mitigate this problem. One way to achieve this is to increase the dropout parameter during model training. Usually, this value should not be chosen larger than 0.5. Another alternative is to include L2-regularization which adds an additional penalty term to the loss, forcing the norm of the weight vector to be small and hence reducing the complexity of the model. L2 regularization can easily be implemented by using the `weight_decay` parameter in the optimizer of the `models.py` file. Finally, the learning rate and window size need to be chosen appropriately. Choosing the learning rate too high can lead to divergence of the weights and choosing it too low leads to slow training progress. Hence, the optimal learning rate is found balancing both of these aspects. In practice, the optimal learning rate usually lies between 10^{-2} and 10^{-5} . In time series analysis also the window size is an important parameter. However, too high windows lead to long processing and hence training time and too small windows prevent the model from learning long term dependencies. Hence, also here a trade-off has to be made.

5.9.2 DARA The analysis of the DARA data also showed that it is crucial to choose the models complexity appropriately, because too simple models tend to converge to the mean prediction, while more complex models tend to be better at capturing the complexity of the underlying time series data. Hence, to further improve the LSTM model trained on the 2021-2023 data and the BiLSTM model trained on the 2021-2024 data, increasing first the hidden size and if necessary also the amount of layers might be beneficial.

Additionally, to further prevent the convergence to the mean in regions with larger gaps and higher uncertainty, the use of alternative loss functions like the Huber loss might be beneficial. In the case in which this does not help, the implementation and investigation of a Bayesian model could make sense. The goal of using the Bayesian model is to predict a mean function and a standard deviation function over time such that for every time point the radiation band can be reconstructed.

To mitigate the outlier problem at the beginning of the predictions of the LSTM and BiLSTM models, the last $window - 1$ datapoints of the train data could additionally be included to the validation set. This would have to be implemented in the `create_gap_train_test_split` function in the `utils.py` file.

To mitigate the problem of the jumps of the LSTM model trained on the 2021-2024 data experimenting with larger window sizes can be beneficial. Additionally, this might also help to capture other nonlinearities in the irradiance data.

Finally, to further investigate large gap losses, the residual plots of the predictions could be investigated, such that the time steps with large deviations can be further analyzed and hopefully improved.

5.9.3 CLARA The most important next steps for the CLARA data are appropriate model tuning and evaluation. For the tuning the ideal procedure is described in detail in the general recommendations section. For the evaluation, the `postprocessing.py` file should be checked and potentially adapted for the CLARA data such that the large gaps can be filled. Additionally, in the second half of the CLARA data additional timepoints can be included to ensure a more homogeneous prediction. Subsequently, further improvements as discussed in the DARA section could be implemented if necessary.

6 Conclusion

The available code of the students from ETH Zurich was debugged and many lost code parts were re-implemented. Hence, the correlation plots (Figure 2), the DARA validation plot (Figure 3) and the forecast plot of the irradiance for DARA (Figure 4) could be reconstructed. However, since optimal parameters for most models were not stated and since seeds to ensure reproducibility for certain packages were not set, the loss values still deviate from the findings of the ETH students during the DS Lab. Additionally, due to time restrictions, many previous data analysis steps including Shapely values were not re-implemented. However, this could be done in further research.

Figures 33 and 34 demonstrate that a suitably trained and tuned BiLSTM model is able to fill the gaps in the DARA data. Hereby, mean imputed features showed better performance than median imputed features. Furthermore, including time features information (year, month, day, hour, minute) significantly improved the model performance and helped prevent jumps in the predictions between train and test datasets. Importantly, there is not one set of ideal hyperparameters to solve the gap filling task, but the optimal configuration always depends on the dataset. Additionally, Figures 49 to 54 show some promising first results for irradiance prediction with CLARA data.

All results could still be improved in further research. To prevent the LSTM-based models from predicting the mean, the complexity of the models should be increased. Additionally, experi-

menting with other loss functions like the Huber loss, which punishes regions of large uncertainty less heavily, could also help mitigating this problem. Alternatively, a Bayesian model predicting the mean and standard deviation function for each time step could be implemented to more accurately reconstruct the irradiance bands. To mitigate the problem of the outliers in the beginning of the forecast region, the last $window - 1$ datapoints of the train dataset might be prepended to the validation and test dataset. This should help to initialize the hidden state of the LSTM during inference appropriately such that outliers can be prevented. To further mitigate the problem with the jumps during abrupt decays of the irradiance data, larger window sizes could be experimented with to provide the model with additional contextual information. Finally, residual plots might provide additional information for large loss values and could be further investigated. Furthermore, other models could also be investigated. The time series transformer showed promising results on the one day gap with one month of training data. Its close alignment of train and test loss combined with its capability to capture long-range dependencies due to the used attention mechanism make this model a strong contender to improve the results. Additionally, the implemented temporal convolutional networks has similar benefits in capturing long range dependencies over the LSTM models.

Acknowledgements

This project was part of a Data Science Internship at PMOD/WRC and received support by Dr. Wolfgang Finsterle, Dr. Margit Haberreiter and Dr. Jean-Philippe Montillet.

References

- Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun (2018). *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*. arXiv: 1803 . 01271 [cs.LG]. URL: <https://arxiv.org/abs/1803.01271>.
- Barth, Julia et al. (2023). *Machine Learning for Total Solar Irradiance*. URL: https://github.com/LudoCatt/TSI-copy/blob/main/TSI-Prediction/Documentation/Report_DARA-CLARA_23.pdf.
- Nie, Yuqi et al. (2023). *A Time Series is Worth 64 Words: Long-term Forecasting with Transformers*. arXiv: 2211 . 14730 [cs.LG]. URL: <https://arxiv.org/abs/2211.14730>.