Machine Learning for Total Solar Irradiance

Julia Barth, Giulio Carassai, Ludovica Cattaneo, Quentin Guilhot
Data Science Lab – Final report

1 Abstract

Solar radiation profoundly influences Earth's climate and plays a pivotal role in diverse fields, including climate studies, space weather predictions, and satellite communications. This project focuses on refining Total Solar Irradiance (TSI) measurements, a key parameter reflecting solar energy, for two radiometers DARA and CLARA. On one hand, we address the challenges in DARA's data quality by employing Deep Learning to identify potential disturbance sources. On the other, CLARA presents missing data during periods of satellite inactivity, prompting the exploration of reconstruction for the TSI. The goal is to improve TSI estimations and gaining insights into instrument effects to potentially benefiting future satellite missions. We employ various Deep Learning and XGBoost models to reconstruct the DARA TSI and then leverage Shapley values to provide a feature analysis for Deep Learning models. For CLARA, we perform multivariate time series prediction with LSTMs, analyzing the impact of different feature configurations on the results.

2 Introduction

Solar radiation is a fundamental factor influencing various aspects of Earth's climate system, ranging from climate studies to space weather predictions and satellite communications. At the forefront of monitoring solar radiation is the measurement of Total Solar Irradiance (TSI), representing the solar energy received at the top of the Earth's atmosphere per unit of time and area. TSI serves as a critical parameter, offering insights into the dynamics of our climate and aiding in the understanding of space weather phenomena. PMOD/WRC is a research institution specializing in solar physics, climate modeling, and irradiation measurements and it is actively engaged in TSI data collection, operating radiometers on satellites, such as the CLARA (Compact and Light-weight Absolute RAdiometer) radiometer on NorSat-1 since 2017 and the DARA (Davos Absolute RAdiometer) radiometer on FY-3E since 2021.[1][2]

With a general goal of refining TSI measurements, this research project tackles two distinct challenges posed by the data collected by DARA and CLARA; from now on we will use the radiometers' names to identify the respective datasets. The DARA data quality suffers from the inherent noise and disturbances present due to various reasons as well as instrumental noise. By leveraging housekeeping features, including temperature excursions and open-close shutter instability, the project seeks to discern the intricate interactions contributing to measurement artifacts.

For this first task, we have three TSI measurements: cavity A which collects just background "noise", B which gives us the main TSI measurement, and C - a backup for B. For this data we have been asked to evaluate the influence of housekeeping features on the TSI time series using Deep Learning to overcome the human bias present in the preexisting evaluation methods. The ultimate goal is to enhance the stability and quality of the irradiance measurements, ushering in a data-driven approach that is as unbiased as possible and going beyond manual time series analysis and cleaning. We decided to use different Neural Network architectures to predict the Total Solar Irradiance from the

remaining features and, to obtain a feature ranking for these intrinsically difficult to explain Neural Networks, use Shapley values as a post-hoc explanation tool.

On the other side, CLARA poses a different challenge which can be summarised as "gap filling". While the satellite experienced periods of inactivity, the housekeeping data is still accessible for these times, although the quality of it differs from the data collected when the radiometer is fully operational. The research investigates the feasibility of reconstructing the TSI during inactive periods solely using housekeeping data and the TSI time series itself when available. Thus, we performed multivariate time series forecasting with diverse deep learning models – in particular LSTMs – to reconstruct TSI values for gaps bigger than one day on a 15-minute rate. However, initial attempts reveal potential biases in the housekeeping data due to the absence of radiometric recordings during these periods. This discrepancy prompts further exploration into the accuracy and reliability of housekeeping data in the absence of proper radiometric adjustments.

The significance of these tasks extends beyond immediate improvements in TSI measurements. Successful completion of the project clearly offers enhanced TSI estimations, reduced noise in measurements, and valuable insights into the effects of instruments and measuring environments, but the research also provides a flexible and scalable tool for future satellite missions, establishing a foundation for advancing our understanding of solar irradiance.

3 Problem statement and technical background

The challenge at hand involved predicting Level2 data – Total Solar Irradiance – from Level1 data – various surrounding measures – within the context of a solar radiometer system. The motivation for the initial challenge lays in the need to have an accurate TSI measure, clean from influences found in the initial Level1 measurements. More in detail, while Level2 data represented the fully calibrated and normalized TSI, corrected for various known influences, the Level1 dataset included housekeeping data, encompassing temperature measurements related to different components of the instrument, as well as sensor measurements indicating the orientation of the instrument towards the sun.

The uncertainties associated with the data process were highlighted, emphasizing challenges related to temperature variations, outliers, and instrument artifacts and the provided data had been cleaned manually and by hand. This clearly exposes it to human bias and from this bias comes the first part of our challenge: to create a software that detects which housekeeping features have a strong influence in the final product so that it is possible to go back to the instruments and fix it. The program will reduce the room for human error in the cleaning process of the signal and at the same time make the process itself easier and quicker for the people involved.

We were encouraged to explore the provided data, understand the corrections applied, and develop models that robustly addressed the intricacies of solar radiometer measurements and to ultimately build our program using Neural Networks for both of the challenged illustrated in the previous section and given the time series nature of the data, we naturally explored the possibility of applying recurrent structures to the challenge.

Traditional Recurrent Neural Networks (RNNs) [7] often encounteres difficulties and exhibits instability during training due to their known hightened vanishing gradient problem. This problem

arises when the gradients of the loss function with respect to the network's parameters become extremely small as they are backpropagated through time during training. As a result, the network struggled to update its weights effectively, especially for long sequences, and faild to capture long-term dependencies in the data.

The vanishing gradient problem is particularly problematic in RNNs because the same set of weights is applied to each element in the sequence, leading to a rapid decrease in gradient magnitude as it is propagated backward through time. Consequently, the network struggles to learn dependencies that are separated by many time steps.

Long Short-Term Memory networks (LSTMs) [5] were introduced as an improvement over traditional RNNs to address the vanishing gradient problem. LSTMs utilize a more complex structure, including memory cells and gating mechanisms, which allows them to selectively remember or forget information over long sequences. This architecture enables LSTMs to capture dependencies across extended time horizons and mitigates the instability issues encountered by vanilla RNNs.

GRUs [4] are an alternative to LSTM which addesses the same issue of vainishing gradients present in vanilla RNNs. They have a simpler structure compared to LSTM since they combine the input and forget gates into a single update gate and merge the cell state and hidden state. Furthermore they may converge faster during training due to this reduced complexity and having an overall smaller number of parameters.

4 Methods/Approach

4.1 DARA

4.1.1 Preprocessing

The dataset, which spanned from August 18, 2021 to July 27, 2023, included the 3 irradiance measures and a spectrum of features such as temperatures, sensor orientations etc. and about 30 features were discussed with the challenge givers and were therefore considered for the task. As anticipates, for DARA the Irradiance was recorded via one main channel (Irradiance B) and one back-up cavity (Irradiance C) and additionally, noise was recorded in Irradiance A. As irradiance B is the central component, we focused our efforts on it for the scope of this project.

The initial generation of a working dataset, involved navigating through petabytes of information, encapsulated in fits files – a standard format used for space measurements. To manage the size of the files, we adopted a sequential processing approach. We converted the time from Julian Day to decimal year formats and aggregated the data on a minute level (facilitated by averaging) since the data was recorded up to nanoseconds. This resulted in a dataset comprising 0.95 million datapoints.

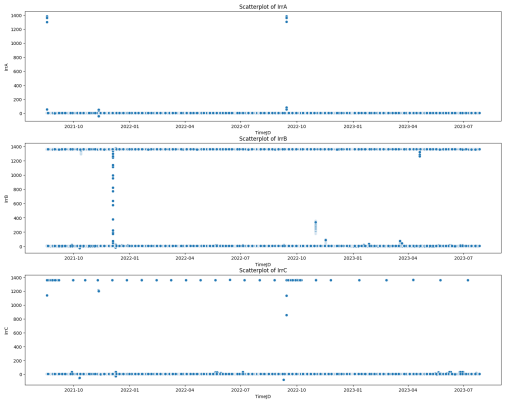
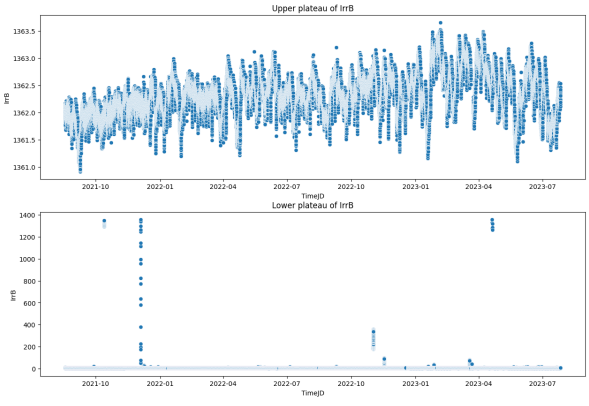


Figure 1: Raw DARA irradiances

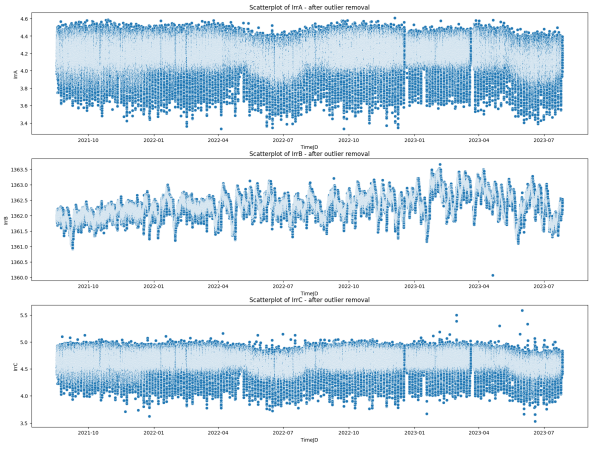
Addressing outliers was a crucial step in enhancing data quality and selecting only relevant measurements. The

outlier removal process was split two distinct categories: extreme outliers attributed to instrumental noise (e.g. initializations, ...), which were identified and eliminated directly and outliers arising from the shutter being closed during the measurement.

The latter were systematically removed using a threshold, leveraging the distinctive value ranges for closed and open shutter configurations (around 0 for closed shutter and 1360 for open shutter) (see fig. 2a). The dataset exhibited minimal missing values, primarily concerning Irradiance B (IrrB). As the missing values accounted for only 3% of the data set, the corresponding entries were not taken into account.



(a) Open and closed shutter configuration.



(b) The three irradiances after removing outliers.

Subsequent analyses included sanity checks e.g. negative values for features with positive domain, checking for consistent data types, detecting nulls and NA related values, as well as distribution assessments, and a thorough examination of feature correlations. An essential outcome of the analysis was the identification of distinct feature groups based on correlation patterns.

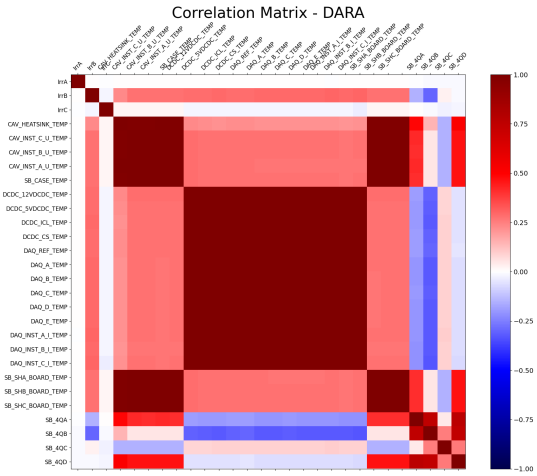


Figure 3: Correlation plot for DARA

These included Temperature Groups, comprising CAV_INST, Cavity heatsink (CAV_HEATSINK), SB_Case, and SB_SH[A/B/C]_Board Temperatures; a DAQ and DCDC Temperatures Group; and a sensor orientation Group consisting of the SB_4Q[A/B].

Moreover, feature correlation analysis revealed noteworthy patterns within Irradiance B (see fig. 3), exhibiting slightly negative correlation with SB_4QB (-0.302399) (and SB_4QA), and positive correlation with the temperatures in particular DAQ_INST _C_I_TEMP (0.300757) and DAQ_INST _B_I_TEMP (0.296035). Note, that DAQ_INST_B_I_TEMP is the temperature with the highest correlation with Irradiance B among all of the features relevant for IrrB. In contrast, Irradiance A and C displayed no significant correlation with other features. The latter was to be expected, as Irradiance A recorded noise while for IrrC this might be due to a lot of noise still being in the signal.

Lastly, we performed a scaling of the data to ensure uniformity and compatibility of the different features which is vital for mitigating bias and preparing the dataset for the subsequent deep learning model evaluations.

4.1.2 Feature selection and engineering

In constructing our machine learning models for the DARA dataset, our approach to feature consideration deviated from conventional feature selection methods. Rather than adhering to typical feature selection techniques, we intentionally retained all available features to maintain a comprehensive view of the dataset, allowing for a detailed exploration of potential traces and dependencies. This decision stemmed from our overarching objective: to unravel and understand the intricate relationships between housekeeping features and Total Solar Irradiance (TSI) measurements.

Unlike models such as random forests or XGBoost, neural networks can be sensitive to correlated features. Our rationale for preserving correlated features, a practice typically discouraged in neural network applications, is to be aligned with the goals of our analysis. Removing these correlations can oversimplify the complex relationships within the data while embracing the correlated features allows the models to capture more subtle dependencies, providing a more realistic representation of the underlying dynamics.

Furthermore, we pursued a nuanced exploration by conducting analyses with different feature configurations. Our models were evaluated using three distinct setups: one involving all available features, a second focused solely on features relevant to Irradiance B. These configurations allowed us to assess how specific sets of features contributed to TSI predictions, shedding light on the significance of individual features and their impact on model performance.

To maintain an unbiased and exploratory stance, we did not introduce additional features through feature engineering. While common practice might involve incorporating time-based features or other engineered variables, we intentionally avoided such enhancements to prevent introducing potential biases and to maintain a clear focus on the inherent relationships between the existing features and TSI measurements. However, for completeness, we will also provide the results of the models on additional time specific features due to their impressive results to illustrate and give an outlook on the opportunities further deep learning models could achieve on the DARA dataset.

4.1.3 Irradiance prediction

In our pursuit of Total Solar Irradiance prediction for the DARA task, we explored a selection of machine learning models. Since our task was to leverage Deep Learning, we considered different models of the Neural Network family: fully connected Neural Networks (FNN), Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU) and, as they were the most promising, various architectures of stacked Long Short-Term Memory networks (LSTM). Additionally, we trained XGBoost [3] as a benchmark since it is a tree based model and therefore naturally ranks features based on their importance in the output and shows competitive results on similar tasks.

It's crucial to note that despite the temporal nature of the data, this task deviated from typical time-series prediction scenarios since our interest is in ranking the housekeeping features. The features served as inputs, and the output was Irradiance B, making it a supervised learning regression task with tabular data.

We want to make a note on the usage of RNNs and LSTMs with tabular data. While our task did not involve predicting the next value in a time series, the recurrent architectures can still capture patterns and relationships within the time-related features. This is especially relevant when dealing with time-series data, as is the case with the TSI measurements and housekeeping features. These recurrent architectures have the ability to learn a hierarchical representations of input data. This can be beneficial in capturing intricate relationships between different groups of features and TSI measurements. In a typical recurrent setup, the model is feed with sequential data over time, where each element in the sequence corresponds to a timestep. However, in our scenario, we did not use sequencing and fed the models a row in a table. This corresponds with sequences of features instead of sequences over time. Each sequence represented a set of features at a given time point, allowing the model to capture dependencies and patterns in the input features rather than their temporal order. This approach is suitable for multivariate time series tasks where the focus is on understanding the relationships among different features at specific time instances rather than capturing temporal dependencies in a sequential manner which is the task for this section.

For model evaluation, a random train-test split with a 20% test set was employed, using Mean Squared Error (MSE) as the evaluation metric. The Mean Squared Error (MSE) is a common metric used in regression tasks, including time series prediction, to measure the average squared difference between predicted and actual values. We will refer to the MSE as either rescaled or scaled. The scaled MSE is the score obtained directly from the models output which takes place in the scaled environment. The rescaled MSE is calculated after rescaling the output back into its original domain. By that we can compare the models globally, while the scaled scores are only comparable if the same scaling was performed. Furthermore, to ensure model generalization and avoid overfitting, learning curves of training and testing were recorded and analyzed. The reconstruction of the models was displayed to further check visually the quality of the various predictions.

4.1.4 Frequency domain evaluation

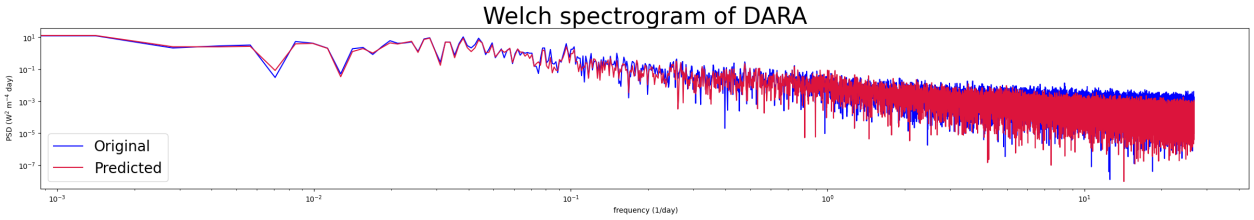


Figure 4: Overlapped periodgrams.

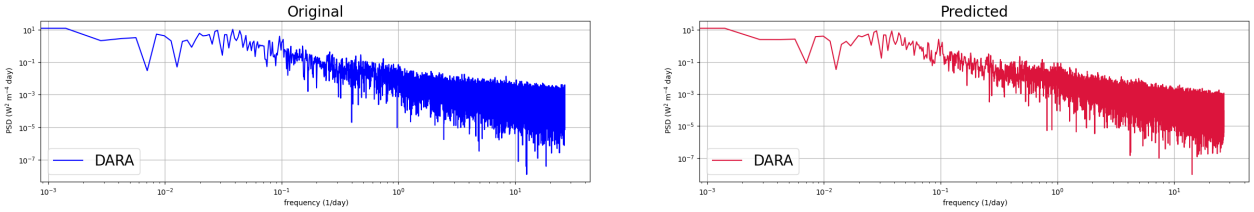


Figure 5: Periodgrams side by side.

To further verify the success of the reconstruction of the TSI from the housekeeping features, we evaluated its quality using a Welch periodogram [10]: a method for estimating the power spectral density (PSD) of a signal, which represents the distribution of signal power over different frequencies.

The periodogram is a common tool in signal processing and spectral analysis and it is less noisy in the estimated power spectra than a regular periodgram since the signal is divided into overlapping segments. A periodogram is computed for each segment and then averaged to obtain a smoother and more reliable estimate of the PSD.

The comparison of the periodgram of the validation part of original dataset with the reconstructed one from the 3-stacked LSTM with just the plain features is shown in fig. 4 and fig. 5. It is evident that the two signals exhibit very similar shapes and this implies that their frequency content and spectral characteristics are alike within the analyzed frequency range.

4.1.5 Feature ranking with Shapley values

In the realm of machine learning and feature ranking, Shapley values [8] offer a robust methodology to quantify the importance of individual features in influencing model predictions. The Shapley value for a particular feature is calculated as the average marginal contribution of that feature across all possible combinations of features, providing a fair allocation of importance. Mathematically, the Shapley value for feature i, denoted as ϕ_i, can be expressed as:

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)]$$

Here, N represents the set of all features, S is a subset of features excluding feature i, $f(S)$ denotes the model's prediction when considering only the features in subset S, and $f(S \cup \{i\})$ represents the prediction when including feature i in addition to S. The sum iterates over all possible subsets S of features, computing the marginal contribution of feature i to the model's prediction.

In the context of our feature-ranking project, leveraging Shapley values provides a rigorous and interpretable approach to assess the relative importance of features, offering valuable insights into their impact on model outcomes. We decided to use the respective explainers available in the library SHAP [6] both for our tree model (XGBoost) and the deep learning ones, looking at the summaries to get a qualitative and quantitative measure of the features' importance.

4.2 CLARA

4.2.1 Preprocessing

CLARA has been recording data on board the NorSat-1 nano-satellite since 2017 and we received data from the period between January 1, 2020, and November 7, 2023, which included 30 housekeeping features and an irradiance time series corresponding to Irradiance B. These housekeeping features cover a range of temperatures, voltages, currents, and sensor readings. The measurements were taken with an irregular rate and in the most dense areas recording times were saved on nanosecond basis. Therefore, we also aggregated the CLARA data set on a minute rate using averaging. This initial data aggregation process produced a dataset of 1.550.000 data points.

While Irradiance data is sporadic, housekeeping data maintains continuous recordings. We found 92.8% missing values for the irradiance data compared to the consistently recorded housekeeping data. These missing values are first of all due to the fact that even when the data is not missing the irradiance is only measured more or less every 15 minutes, but bigger gaps can result from

deactivation during the day or from general long-term shutdowns. The first case is not relevant for our task to reconstruct missing data, as the radiometer is not expected to record data continuously but only in certain settings, as it is for DARA. Thus, it is necessary to select relevant gaps and a moderate reconstruction rate. We created a separate dataset with a reduced time resolution, aggregating the data at 15-minute intervals – equal to the average rate of the recorded TSI. For the gap reconstruction, we only selected gaps larger than one day to avoid filling gaps where the radiometer naturally did not record and not to have just reconstructed points everywhere.

Regarding the preprocessing, we performed outlier removal on features and the target values. Initial outlier removal targeted severe anomalies, followed by the application of rolling median techniques to mitigate noise. We considered multiple techniques such as rolling mean or median, z-score selection and interquartile range outlier removal. Due to its resistance to extreme values and proficiency in capturing local patterns we settled for rolling median outlier removal since the dynamic CLARA TSI time series has a slight increasing trend. For the features, a z-score outlier removal approach provided more precise results.

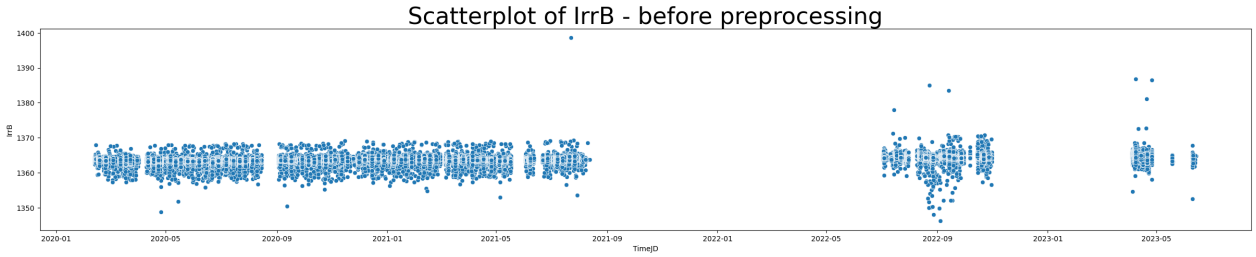
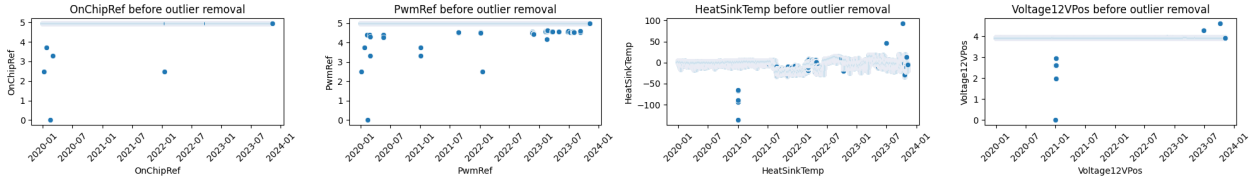


Figure 6: CLARA TSI time series before preprocessing

Z-score outlier removal Z-score outlier removal is a statistical technique used to identify and manage outliers in a dataset. The Z-score, or standard score, measures how far an individual data point deviates from the mean of the dataset, expressed in terms of standard deviations. The Z-score is calculated for each data point using the formula:

$$\text{Z-score} = \frac{x - \mu}{\sigma}$$

Here, x is the data point, μ is the mean of the dataset, and σ is the standard deviation. In practical terms, the Z-score provides a numerical representation of how unusual or extreme a data point is within the dataset. The process of Z-score outlier removal involves setting a threshold value (we used $\|Z\| > 8$) beyond which data points are considered outliers. This threshold indicates that any data point lying more than eight standard deviations away from the mean is flagged as an outlier.



Rolling median outlier removal Outlier removal using rolling mean or median is a technique employed in time series analysis to smooth the data and identify significant deviations from the overall trend. Unlike static mean or median calculations, which consider the entire dataset at once,

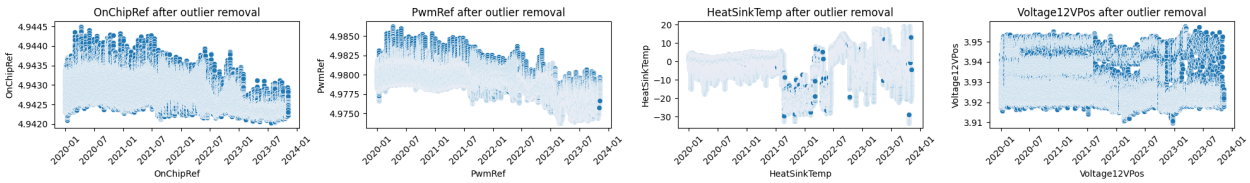


Figure 7: Comparison of selected features before and after applying Z-score outlier removal

rolling mean or median is computed over a moving window of consecutive data points. This approach helps capture local variations and patterns in the time series.
The rolling mean is calculated by averaging values within a specified window that slides along the time series. This smoothing process helps mitigate short-term fluctuations, highlighting the underlying trend. On the other hand, the rolling median, calculated similarly, is more resilient to extreme values and outliers compared to the mean.

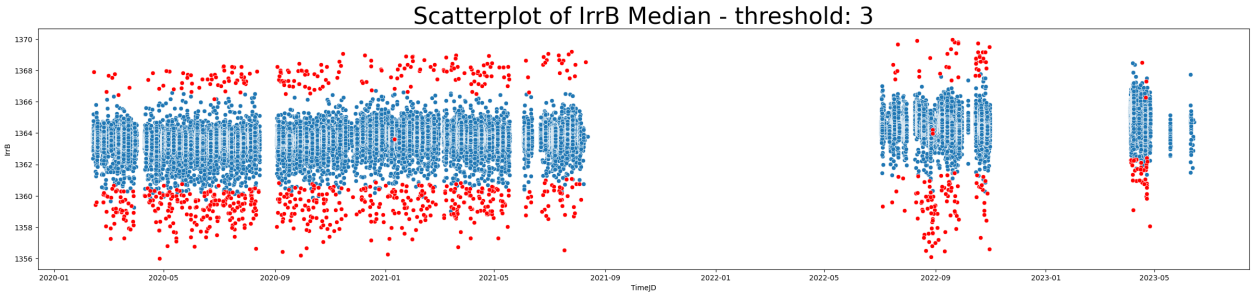


Figure 8: Results of the outlier removal with rolling median for Irradiance B

The rationale for using rolling mean or median in a time series with a trend lies in its ability to distinguish between short-term fluctuations and the overall trend. The choice between mean and median depends on the data's nature, with the rolling median often preferred when dealing with extreme values or outliers due to its robustness. The window size in this context determines the number of data points considered in each calculation. A larger window captures longer-term trends, while a smaller window accentuates shorter-term variations. In our specific case, we opted for a rolling median with a window size of 5 and a threshold of 3 for Z-score outlier removal.

4.2.2 Further preprocessing

Another preprocessing step involved normalizing the time series data. Normalization ensures that all values are on a consistent scale, enhancing the performance of models like Long Short-Term Memory networks (LSTMs). We employed a common method of scaling the data to values between 0 and 1 using the minimum and maximum values as we did for DARA.

We also performed additional analyses on the data as well as a correlation analysis to extract preliminary insights into the relationship of the target Irradiance

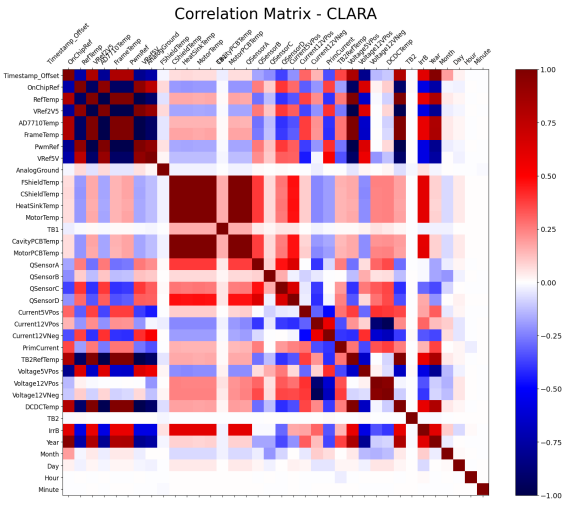


Figure 9: Results of correlation analysis for the CLARA data

with the housekeeping features and possible correlations between the features. FShieldTemp, HeatSinkTemp, and CShieldTemp have the three highest positive correlation values each score is above 0.6 (see 9). Motor-related temperatures follow, including MotorPCBTemp and MotorTemp, also displaying a high positive correlation.
Generally, all temperature-related features exhibit positive correlation scores (above 0.58) with the output, implying their importance on Irradiance B. The correlations with electrical parameters exhibit more diverse patterns. While Current12VPos, Current5VPos and QSensorD show moderate positive correlations, Current12VNeg, Voltage12VPos, and Voltage12VNeg have moderate negative scores. VRef2V5, OnChipRef, Voltage5VPos, PwmRef, and VRef5V are negative correlated with Irradiance C with scores above 0.57.

The close up on the correlation matrix in figure 11 (rearranged to feature groups) shows significant interdependencies among various housekeeping features in the CLARA dataset. The instrument temperature group in the right-lower corner shows almost perfect correlation with each other with scores above 0.999. The RefTemp, AD7710Temp, FrameTemp and DCDCTemp group in the left-upper corner also provides almost perfect correlation. There is also high positive correlation between the the features OnChipRef, VRef2V5, PwmRef, and VRef5V and strong negative correlation with the left-upper feature group. In particular, VRef2V5 and PwmRef are almost perfectly correlated.

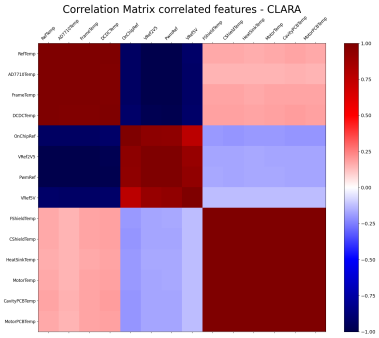


Figure 10: Correlation analysis for highly correlated feature groups

4.2.3 Feature selection and engineering

For the identified feature groups in the correlation analysis, the high correlation could significantly influence the performance of the deep learning models. Since the CLARA task is aiming to produce accurate reconstruction values and not to evaluate the features, we decided to perform feature selection based on the correlation results.

We analyze the performance in different configurations. In setting 'All correlated' we chose a representative for each group and removed the other features from the data set to eliminate as much correlation as possible. Moreover, we investigate the influence of correlated features on LSTMs. For that we introduce 'Setting 1' and 'Setting 2' which do not exclude all correleted features in a group. The difference in Setting 1 and 2 is to test how big the influence of feature selection is. In Setting 1, RefTemp is used, which is suffering from outliers that AD7710Temp does not have. In Setting 2, this RefTemp is replaced by the smoother AD7710Temp. We also decide to use DCDCTemp as a feature, as it has slight structural differences to the other group members. An overview of the feature selection of the different settings can be found in Table 1.

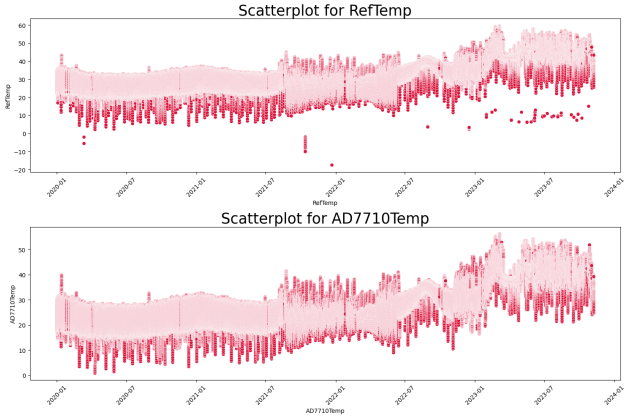


Figure 11: Comparison of RefTemp and AD7710Temp

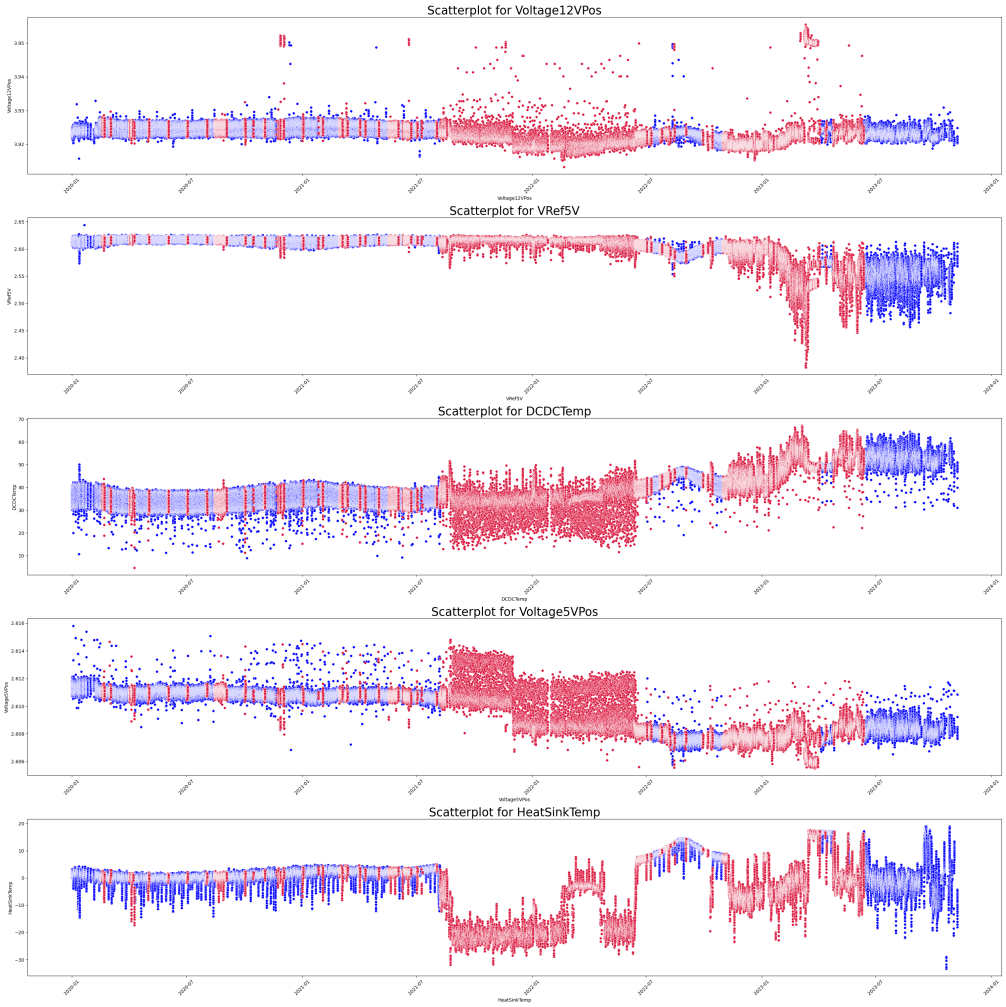


Figure 12: Irregularities in features of 15-minute dataset (blue: data with available TSI data; red: data with missing TSI data)

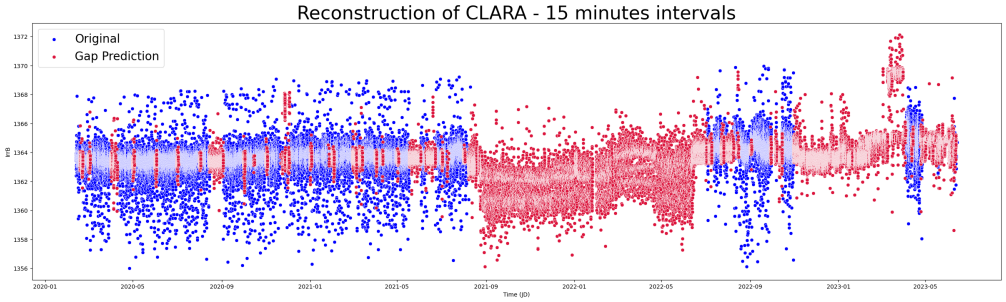


Figure 13: An LSTM result suffering from the biased input data. It exhibits the separated structure on the right of the Voltage and Current features, the dips from the HeatSinkTemp group and the inconsitent outlier structure from DCDCTemp and Voltage5Pos.

During the large gaps in the time series, the TSI measurements were not deliberately recorded. In order to provide useful measurements, the satellite and its instruments must be correctly positioned in relation to the sun. As the radiometers were not active, the satellite was not correctly aligned.

Housekeeping measurements during these periods are therefore biased. There are several features where anomalies in the measurements during this recording gap are noticeable, as can be seen in Figure 18. In order to remove the bias, we applied additional preprocessing on the test data to features Current12VPos, Current12VNeg, Voltage12VPos, Voltage12VNeg, as they exhibited a disconnected out-of-range structure in our model results (see Figure 13), which can be attributed to the anomaly in features listed above (see as an example Figure 18). To address this, we first imputed the outliers with the mean of the non-gap data. We then calculated the moving average, which was then applied only to the outliers in order to obtain a more nuanced alternative value for the outlier. Features VRef5V, Voltage5VPos, DCDTemp and feature group of HeatSinkTemp, as seen in Figure 18, had anomalies that require a more complex cleanup, which we recommend as a next step. In the scope of this project, the features were removed for the non-biased feature setting (referenced as 'W.o. Bias').
We trained models with the above configurations with and without additional time features since the goal of this task are an accurate prediction and thus feature engineering can be applied.

	W.o. Bias	Setting 1	Setting 2	All Correlated
Features excluded	FShieldTemp MotorTemp CShieldTemp CavityPCBTemp MotorPCBTemp HeatSinkTemp	FShieldTemp MotorTemp CShieldTemp CavityPCBTemp MotorPCBTemp	FShieldTemp MotorTemp CShieldTemp CavityPCBTemp MotorPCBTemp	FShieldTemp MotorTemp CShieldTemp CavityPCBTemp MotorPCBTemp
	DCDCTemp	FrameTemp AD7710Temp	FrameTemp RefTemp	FrameTemp RefTemp DCDCTemp
	VRef5V	VRef2V5	VRef2V5	VRef2V5 PwmRef VRef5V
	Voltage5VPos			
Features altered	Current12VPos Current12VNeg Voltage12VPos Voltage12VNeg			

Table 1: Feature Settings ordered by correlation feature groups and action performed on the features

4.2.4 Time Series Forecasting

We identified the task of reconstructing gaps in the CLARA dataset as a multivariate time-series prediction problem. Time series prediction involves utilizing statistical or machine learning models to forecast future values based on historical observations and for our specific application, it means predicting missing values in the time series of the irradiance using the housekeeping features and learning on the already available TSI.

To proceed, we split the time series data into training and validation sets. Wile the validation set

contains the actual values and we used it to evaluate the performance of the models, the test set is for this task the actual gap data for which we have no measure of how good our performance is. Although a typical split involves using the first 80-90% for training and the remaining 20-10% for validation to avoid data leakage, our task focused on reconstructing gaps rather than predicting future values therefore only evaluating on the very end would not be the most reliable measure. Given this distinction, we leveraged the entire dataset for training. This is particularly important as the time series shows a trend and the right side of the time series contains numerous larger gaps, therefore removing the right side of the time series for testing is not conform to the general setting where values "in the future", i.e. to the right of the gap, are available and provide valuable information for reconstruction. Therefore, we trained most of our models with a random split of training and validation (90% training and 10% validation). We denote from now on the data used for training as Original (or train), the data for validation as Original - Validation (or validation) and the models output as Prediction - Validation. The resampled data for the gaps is referred to as test set (and its model output as Gap Prediction in plots).

Evaluation metrics centered on (Rescaled) Mean Squared Error (MSE), evaluated on the validation set is used to assess the performance of the models. Plots of prediction results were used for visual inspection, and the models were tested on the 15-minute gap data to evaluate their effectiveness in reconstructing missing values within the given time series. As before we also used the learning curves of training and validation to detect and prevent overfitting.

To solve the task of multivariate time series prediction, we primarily used Long Short-Term Memory networks (LSTMs) complemented by XGBoost as a benchmark. LSTMs were chosen for their ability to capture and learn dependencies in sequential data over longer time intervals. These networks excel at modeling long-range dependencies and are therefore well suited for time series predictions with complicated patterns and contextual relationships. The mechanism of memory cells in LSTMs allows them to selectively store and forget information, facilitating effective learning of temporal dependencies.

XGBoost, a gradient boosting algorithm, was used as a benchmark to provide a comparative basis for our deep learning models. While XGBoost was not designed for sequential data, it is a powerful algorithm known for its efficiency and effectiveness in capturing complex patterns in tabular data. In addition, XGBoost can be trained with less data than LSTMs, which require a large data set and a longer and more complex tuning process. The inclusion of XGBoost allowed us to assess how well the deep learning models performed against a strong, easy to train, and non-sequential baseline which already proved its effectiveness in the DARA section.

5 Results

5.1 DARA

From the results shown in table 2 it emerges that the models that could rely on all features for the prediction have consistently lower losses. The 3-stacked LSTM, whose prediction can be found in fig. 14 and fig. 15, is outperforming the other deep learning models (using all and using only B features). However, XGBoost, shown in fig. 16 has the overall best performance for both feature settings. In the corresponding figures 14, 15, and 16 it is evident that the XGBoost models can reconstruct the DARA TSI better. The LSTM plots show less coverage of the top of the TSI which is not observable in the XGBoost plot 16.

Models	Configurations			
	All features		B features	
	Rescaled MSE	Scaled MSE	Rescaled MSE	Scaled MSE
Neural Network	0.021 14	0.002 80	0.022 06	0.002 92
2-stacked LSTM	0.020 32	0.002 69	0.019 99	0.002 65
3-stacked LSTM	**0.018 54**	0.002 45	**0.019 01**	0.002 52
3-stacked GRU	0.019 48	0.002 58	0.020 32	0.002 69
RNN	0.030 24	0.004 01	0.032 44	0.003 99
XGBoost	**0.012 62**	0.001 67	**0.013 97**	0.001 85

Table 2: Performance overview

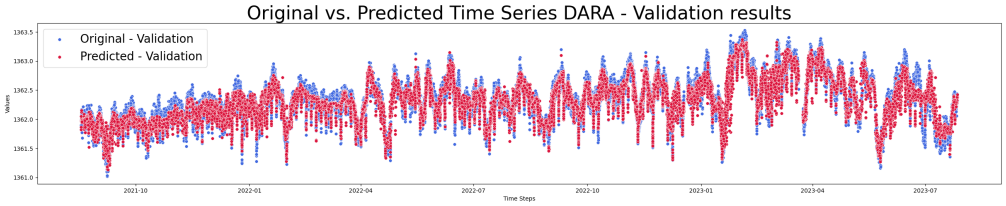


Figure 14: Validation data reconstructed from all features with LSTM.

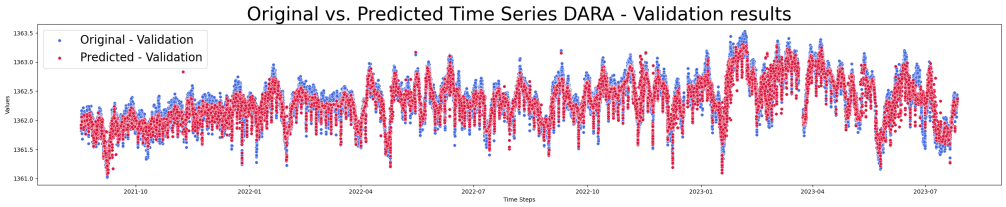


Figure 15: Validation data reconstructed from the B features with LSTM.

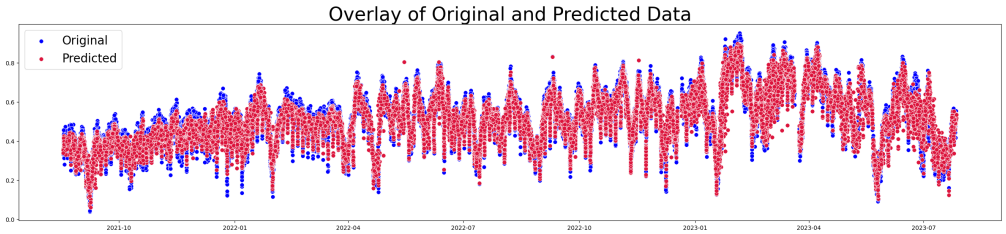
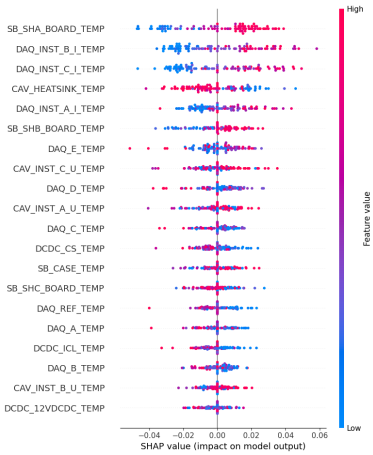


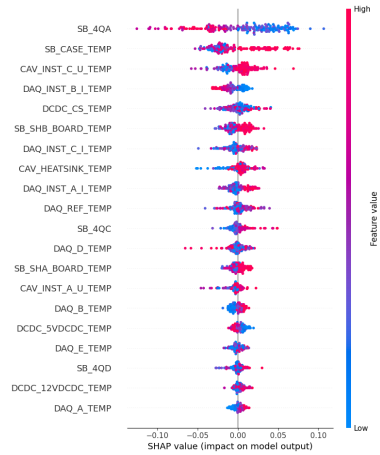
Figure 16: Validation data reconstructed from all features with XGBoost.

For the Shapley values, we noticed the same features scoring the highest Shapley values for most deep learning models. However, they do not match the ones obtained from XGBoost. These are reported in fig. 17a and fig. 17b.

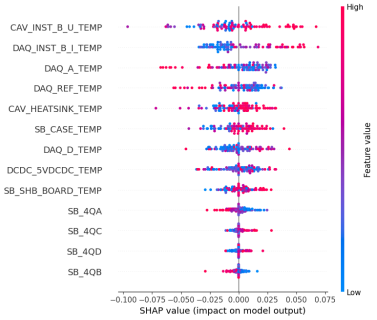
The highest ranking features for the neural models according to the Shapley values are SB_ SHA_ BOARD_TEMP, the DAQ_INST_TEMP group and CAV_HEATSINK_TEMP. For XGBoost on the other hand, the DAQ_INST_TEMP group achieved high Shapley values but SB_4QA was consistently ranked at the top, while it was always ranked in the Last 4 (as all the other sensors) for the network



(a) Shapley values from the 3-stacked LSTM with all features.



(b) Shapley values from XGBoost with all features.



(c) Shapley from the 3-stacked LSTM with B features.

models.

We also evaluated the feature importance only for selected B features since other features corresponding to measurements of the other cavities could bias the results. For example, SB_SHA_ BOARD_TEMP is actually not directly related to Irradiance B. For this setting CAV_INST_B_U_TEMP achieved high (if not even the highest) Shapley values in most models we trained as for example in fig. 17c. Followed by DAQ_INST_B_I_TEMP which matches with the analysis for all features. The Features DAQ_A_TEMP, DAQ_REF_TEMP and CAV_HEATSINK_TEMP were also located in the Top 5 for many models. The XGBoost chart remained unchanged.

5.2 CLARA

Comparison of models with and without time features When comparing the models with and without time features, it is obvious that the models with time features dominate the models without in every configuration Table 3 and 4. One factor that contributes to this result is the presence of a slight upward trend in the Irradiance B time series. This trend is consistent with the upward trend in the time-related features. The relevance of these features were already revealed in our correlation analysis (see 9) which showed correlation between irradiance B and Year, Month and Day. These correlations suggest that certain temporal factors play a role in influencing the outcome variables. However, it is crucial to note that the learning process exhibits periodic patterns, as can be seen in Figure A. This discrepancy emphasizes that while time features contribute to improved predictions, they can also lead to biases such as those observed in the periodic patterns during the learning process.

Comparison of the different feature configurations When comparing the results of the correlation configuration. We found that all features followed by selection 2 performs the best and models without correlated features the worst. Although deep learning models can suffer from correlated features and it is best-practice to remove correlated features, this effect is not evident in our results (see 6.0.2 for further details). We expected setting 2 to perform slightly better than setting 1 due to the outliers that can affect the scaling and training of the models. It also shows, that it is relevant which features are picked as representative.

The models without biased features perform slightly worse in terms of loss than the other configurations,

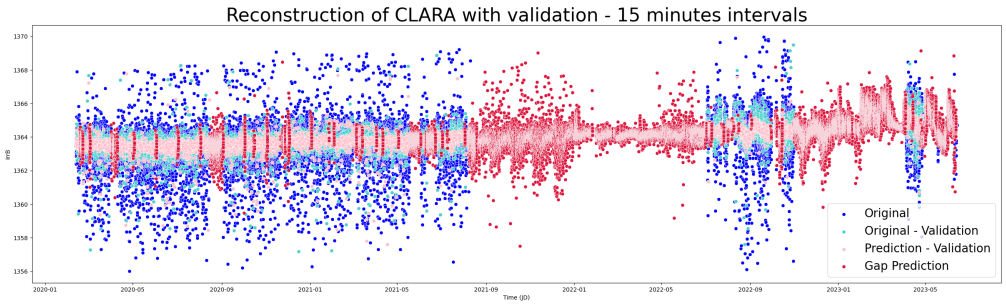


Figure 18: Example of a time features biased reconstruction LSTM output including train, validation data as well as validation and test prediction; a periodic pattern can be seen in the predicted validation and test set.

Models	Configurations				
	All	W.o. Bias	Correlation		
			Setting 1	Setting 2	All correlated
LSTM	**0.2679**	0.2859	0.2789	0.2763	0.2884
XGBoost	0.2935	0.3093	0.3016	0.3003	0.3093

Table 3: Models and metrics

Models	Configurations with time features				
	All	W.o. Bias	Correlation		
			Setting 1	Setting 2	All correlated
LSTM	**0.2504**	0.2630	0.2586	0.2539	0.2721
XGBoost	0.2716	0.2743	0.2768	0.2767	0.2777

Table 4: Models and metrics

which is to be expected to some extent, as the set of heat sink features provides important general information (in DARA, these features were one of the most important in the Shapley value evaluation). However, in the plotting results as in Figure 19, the floating structure on the right side of the large gap in 2023 is not present. The unbiased models also did not suffer from the dips in the large gaps, strong dispersion in the largest gap or a disproportionate rising trend in the larger gap from the right. Thus, we were able to show that the disturbance in the results comes from the anomalies in the test data set.

Comparison of LSTMs and XGBoost LSTMs consistently demonstrated better predictive performance compared to XGBoost. This can be attributed to LSTM's ability to capture intricate temporal dependencies and patterns in the multivariate time series data, which is crucial for this task. LSTMs were designed for handling sequential time series predictions. The best performing XGBoost models prediction results can be found in Figure 20. It does not match the expected shape of the Irradiance and suffers from irregularities in the shape of the prediction time series. We can conclude, that LSTM models are a valid and performant choice for this task.

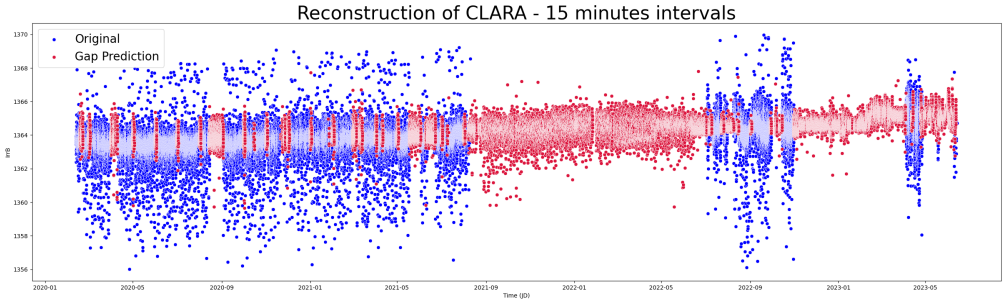


Figure 19: The prediction result of the unbiased feature setting (without time features)

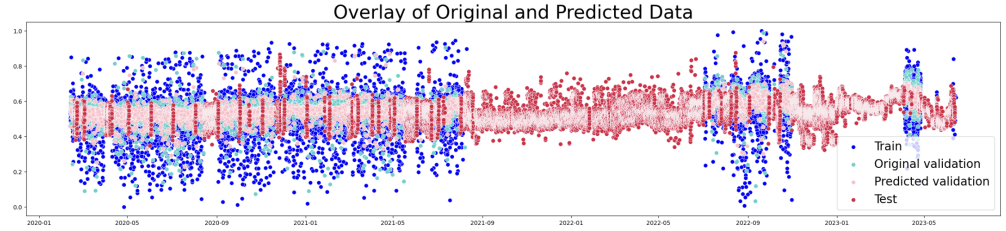


Figure 20: The prediction result of the best XGBoost model (all features including time features)

Visual evaluation The reconstruction plot in Figure 21 shows the result of the best performing model using all features. The results of the unbiased feature setting (see Figure 19) seem a bit too smooth. However, anomalies (due to distortions) can be observed in the plot of the best model that do not match the expected course of the time series. Thus, one needs to find a trade-off between both configurations. We will discuss the slight inconsistency of visual explanation and loss in 6.0.2.

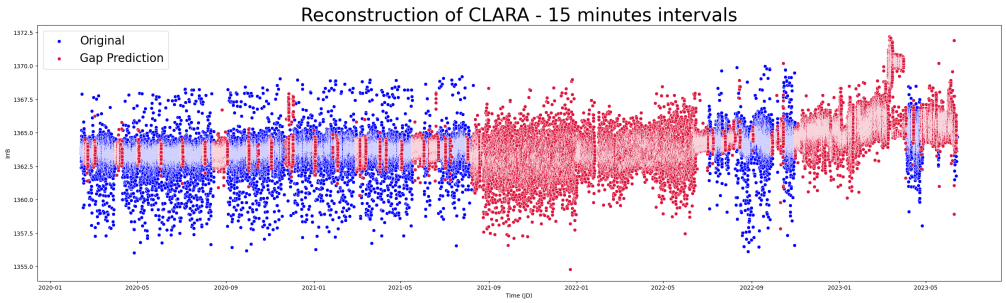


Figure 21: The prediction result of the best performing model (using all features incl. time features)

6 Discussion

6.0.1 DARA

XGBoost vs. Deep Learning As seen in the results, XGBoost performed better than LSTMs or any other deep learning model. XGBoost is designed to excel in tabular data scenarios, where the input features are structured in rows and columns. LSTMs on the other hand were designed for sequential data (time series prediction). The lack of temporal information may cause the model to not extract enough information to be competitive to XGBoost. XGBoost is able to extract temporal-related information even without explicit time sequences and would perform a time series prediction in the same way as in this setting (with success as seen in CLARA). However, in this

setting LSTMs have the capability to implicitly capture complex interactions and dependencies between features since they are treated as sequential time steps. The model can learn to associate patterns across different features in a way that NNs struggle to do so. RNNs showed instable training probably due to sensible learning rates and vanishing gradients and thus performed not as well.

Including time features We have also fed the LSTMs with time features (year, month, day, hour, minute). The results were significantly better than those without feature engineering. While the best LSTM model in our analysis above achieves a rescaled MSE of 0.01854, an LSTM with time features can reach a score of 0,00482. This is reasonable since LSTMs in this task are used on the feature axis and thus lack temporal information. Since the underlying structure is a time series which carries a lot of temporal patterns and dependencies, adding time features provides the model with vital information for the reconstruction.

Figure 22 provides the corresponding reconstruction plot. One can observe that the model is able to recover the validation set almost fully. In the Shapley value evaluation the time feature Year, Day, and Month (in this order) were listed as the three most important features.

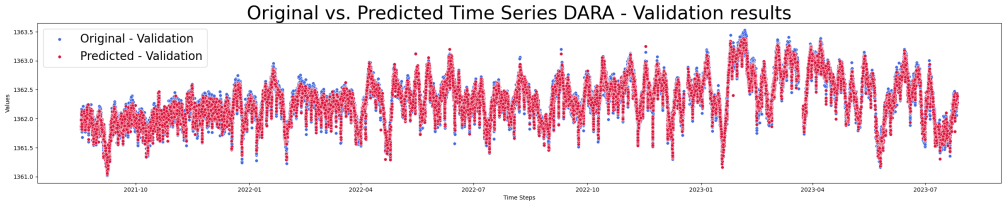


Figure 22: Reconstruction performance of LSTM with time features

Including Irradiance A and Irradiance C
We also explored the impact of including Irradiance A and B in the training of our model, assessing whether these variables contribute to improved results and are essential for irradiance reconstruction (Shapley evaluation). The best score of 0.01839 was again achieved by a 3-stacked LSTM and thus slightly improving the performance. However, this can also be an effect of the added information (as we have also seen in the CLARA part, the LSTMs are 'data-hungry' and react well on additional features even if they carry little information).

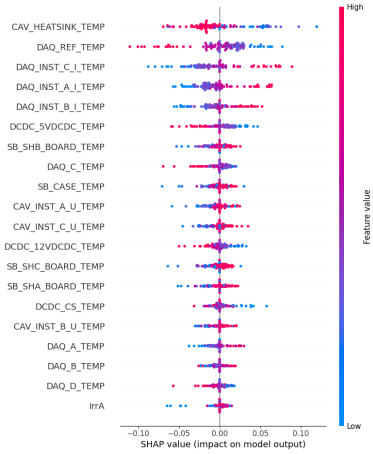
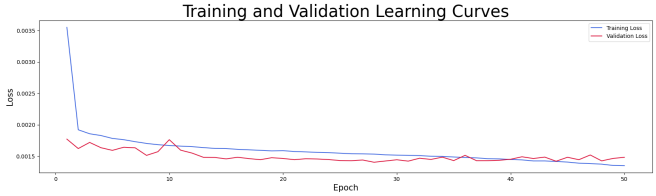


Figure 23: Shapley values for a LSTM model with Irradiance A and C

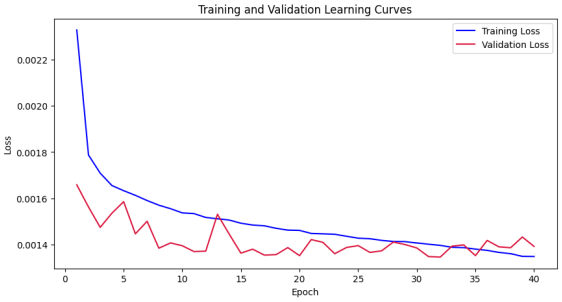
The Shapley value evaluation indicated that both Irradiances A and C had limited relevance for Irradiance B (the results can be found in Figure 23). Moreover, Irradiance A (the 'noise' Irradiance) even scored higher shapley values than Irradiance C. This result is somewhat surprising, as Irradiance C is treated as a backup for Irradiance B and Irradiance A captures noise. However, it should be noted that Irradiance B was cleaned and processed. The noise in Irradiance C is probably too large to provide relevant information for the reconstruction of Irradiance B. In the correlation analysis (see 3), we were also able to determine that Irradiance B is not correlated with the other two Irradiances which matches the Shapley evaluation.

6.0.2 CLARA

Convergence behaviour of Clara When comparing both model configurations, the models with time series features converged faster than the models without time features. While a convergence phase for the latter often needed 35-45 epochs, the time feature models converged usually around epoch 25 epochs as can be seen in the Figures in 24a and 24b. Time features can reduce the ambiguity in the learning task by providing additional cues about the temporal structure of the data. This reduction in ambiguity allows the model to focus on relevant patterns and relationships, leading to quicker convergence during training.



(a) Typical learning curves for models without time features: convergence phase around 35-40 epochs



(b) Typical learning curves for models with time features: convergence phase around 25 epochs

Inconsistencies in evaluation and task We would also like to point out that the lowest validation losses (convergence point) sometimes occurred before the reconstruction graph reached an expected shape. Thus, the better visual reconstruction results were obtained when the mode was slightly overfitting. There were also multiple cases were the validation losses were low but the prediction plots were very bad (an example with a competitive score of 0.2597 (with time features) can be found in Figure 25). In general, it should be noted that the gap data differ structurally in many features, so that a well-performing model on the validation data provides a good orientation, but is not perfectly transferable. The difference between validation and test data is larger than in general machine learning cases, which is why we base our evaluation not only on the statistical results, but also on the graphs and the training behavior.

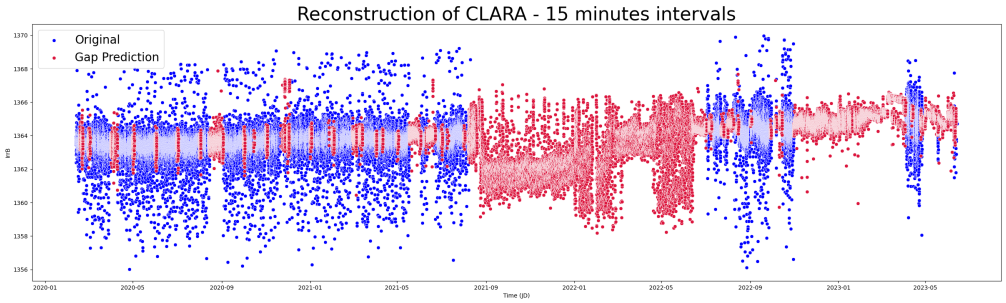


Figure 25: Poor prediction result of a model with good score: rescaled MSE of 0.2597 (with time features)

LSTMs and correlated features LSTMs are not expected to always perform well on correlated features.[9] However, the models with additional features, even if they are highly correlated, outperform the uncorrelated feature models in our setting. This suggest an interesting nuance in the impact of correlated features on deep learning models on this specific dataset. Reasons for this effect could be that the correlated features may have non-linear dependencies that LSTMs can effectively model. On one side, LSTMs are 'data-hungry' and capable of learning complex relationships. So they

could benefit from multiple correlated features which provide complementary (and slightly different) information. On the other side, LSTMs inherently performs a form of implicit feature selection due to their different gates. This adaptability allows the model to handle correlated features without explicit feature selection to some extend.

7 Conclusion

In this project, we used Deep Learning to refine Total Solar Irradiance (TSI) measurements for the DARA and CLARA radiometers with the support of XGBoost. Our focus was on one side on gaining insights into the underlying feature/instrumental effects to improve data quality and on the other side to reconstructing missing data.

Using Deep Learning for evaluating feature traces on the DARA TSI lead to consistent results over models of the deep learning family and we were able to extract important features for Irradiance B. However, XGBoost outperformed the deep learning models since the data was tabular, correlated and complex which is challenging for some deep learning models. Its evaluation with Shapley values deviated from the Deep Learning models.

For CLARA we can record a partial success in data reconstruction due to low quality of the reconstruction input. For the gaps when the measurement environments were operational and unbiased housekeeping features were recorded, predicting reasonable values solely on supporting housekeeping features is possible with LSTMs. In order to optain results even for the biased gaps further models and preprocessing needs to be applied. This opens promising outlooks for future research. Specifically, it paves the way for exploring the prediction of Total Solar Irradiance using advanced machine learning techniques and cost-effective instruments.

References

[1] Esa: Dara description.

[2] Pmod wrc press release of clara, 2017.

[3] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[4] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[6] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[7] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. volume 2, pages 1045–1048, 01 2010.

[8] L. S. Shapley. *17. A Value for n-Person Games*, pages 307–318. Princeton University Press, Princeton, 1953.

[9] Huaiyu Wan, Shengnan Guo, Kang Yin, Xiaohui Liang, and Youfang Lin. Cts-lstm: Lstm-based neural networks for correlatedtime series prediction. *Knowledge-Based Systems*, 191:105239, 2020.

[10] P. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967.