Spring 2024 Forecasting Class Competition

Master Forecasters

2024-04-28

Contents

Name	Major Contribution
Cory Petersen	Modeling
Laila Saleh	Data Transforms and Visualizations
Daniel Moore	.Rmd organization

Abstract

We used a Time Series Linear Regression Model (TSLM) for our final prediction of the Plane of Array Irradiance (POA) $\left(\frac{W}{m^2}\right)$. The TSLM outperformed the Seasonal Naive and Seasonal Mean benchmark models as well as more complex models such as Seasonal Auto-regressive Integrated Moving Average (SARIMA), SARIMA with exogenous variables (SARIMA-X) and Long-Shortterm Memory recurrent network (LSTM). While we found the TSLM could reliably predict the moving average throughout the day, neither it nor any other model could predict the randomness of clouds on days which weren't totally clear or totally overcast. These findings are significant because they highlight how simple models can still provide great insights and forecasts, outperforming much more complex models with many more parameters. This model could be deployed on a cheap System on a Chip (SOC) computer and used to make real time predictions and provide local control to the solar panel array. Deployment of distributed energy resources (DERs) with localized autonomy is an important step in building a Smart Grid that is inexpensive to operate while providing more efficiency and resilience.

1 Purpose

The purpose of this project is to predict the Plane of Array (POA) Irradiance $\left(\frac{W}{m^2}\right)$ for measurements made at the Rutgers University Energy Lab at Richard Weeks Hall in 10-minute increments for the next 12-hours. The POA has been measured by a pyranometer with the same orientation as the solar array. This measurement is critical for modeling the performance of a Photo-Voltaic (PV) system. Predicting future POA allows operators to optimize DERs.

2 Data Exploration

The first step is to gather all data and preporcess it to gain insights about patterns and relationships.

2.1 Loading the Data

The data is provided in 10-minute increments from June 1, 2023 to August 2, 2023 with the following measurements:

- DATE_TIME: Date/time information
- AIRTEMP: Air temperature (C)
- RH_AVG: Humidity (%)
- DEWPT: Dew point temperature (C)
- WS: Wind speed $\left(\frac{m}{s}\right)$
- GHI: Global Horizontal Irradiance $\left(\frac{W}{m^2}\right)$ measured from a horizontal pyranometer mounted on a sun tracker
- DNI: Direct Normal Irradiance $\left(\frac{W}{m^2}\right)$ measured from a horizontal pyranometer mounted on a sun tracker
- DIFF: Diffuse Irradiance $\left(\frac{W}{m^2}\right)$ measured from a horizontal pyranometer mounted on a sun tracker
- POA: Plane-of-Array Irradiance $\left(\frac{W}{m^2}\right)$ measured from a pyranometer that has the exact same tilting

The table below shows a few observations getting close to sunset (20:31) on July 7th, 2023.

DATE_TIME	AIRTEMP	RH_AVG	DEWPT	WS	GHI	DNI	DIFF	POA
2023-07-05 17:10:00	32.92	46.69	20.02	0.932	111.5	0	111.5	120.5
2023-07-05 17:20:00	32.65	50.36	21.01	1.988	104.8	0	104.8	111.4
2023-07-05 17:30:00	31.48	61.72	23.28	2.144	96.0	0	96.0	103.9
2023-07-05 17:40:00	31.00	63.82	23.36	1.870	69.9	0	69.9	81.8

2.2 Loading External Data

Predicting the POA is tantamount to predicting how sunny it is. We have also obtained historic, hourly-weather data for the time period from Visual Crossing, a company that provides weather data for enterprise. At a given time t, the data is treated as historic for time before t and forecasted weather for time after t. This is a reasonable approach for the scope of this project as day-ahead hourly weather forecasts are very accurate and we are only using basic weather data. If deployed, the model could be easily modified to train on true forecasts and we do not expect a significant change in the model's performance.

datetime	temp	dew	humidity	precip	precipprob	winddir	cloudcover	visibility
2023-06-05 23:00:00	61.7	42.0	48.39	0	0	328	40.9	7.6
2023-06-06 00:00:00	59.9	42.0	51.53	0	0	325	21.8	8.5
2023-06-06 01:00:00	58.1	42.8	56.58	0	0	335	19.4	8.7
2023-06-06 02:00:00	58.0	42.1	55.53	0	0	338	19.4	8.7

We combined the hourly weather data and 10-minute interval irradiance data by applying the same weather for a given hour to all irradiance data in that hour.

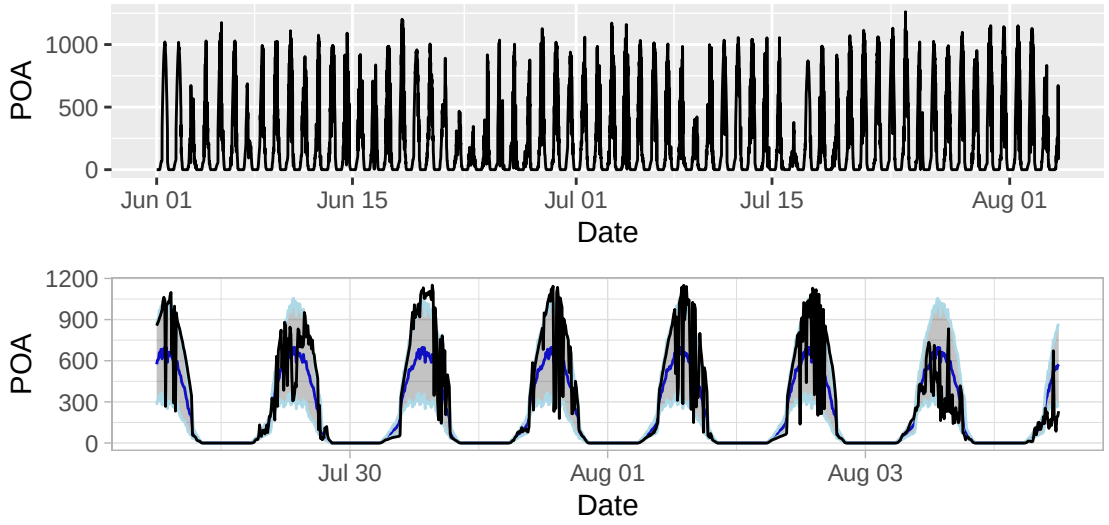
2.3 Data Visualizations

Data visualizations allow us to understand relationships amongst the data and the temporal nature of the features. Below we provide the figures which offer the most insight into why we chose or rejected certain models and selected certain hyperparameters.

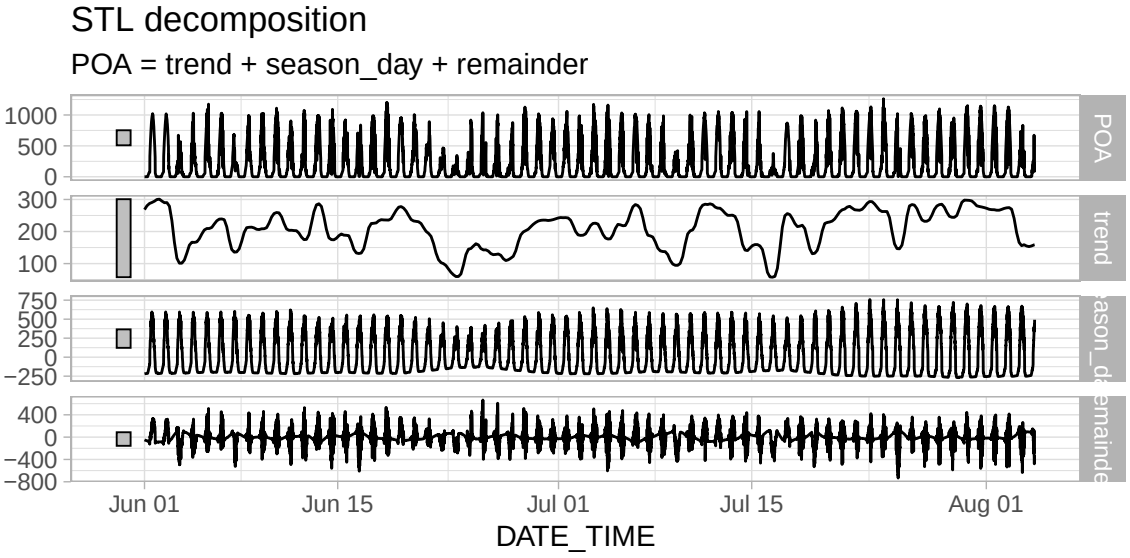
2.3.1 POA Time Series

Below we look at our target variable time series over the entire dataset. Naturally, the POA is highest around noon and zero at night and it generally follows a predictable pattern. However, we also observe many sudden drops in POA followed by a quick recovery. These changes are likely due to cloud cover and are very unpredictable. As will be shown, we are able to account for these changing conditions to a certain degree, but of course the randomness of clouds will always be a challenge.

The daily mean and standard deviation show a consistent pattern throughout the day and the polar plot accentuates this. The mean, upper, and lower bounds from concentric circles connected at the origin and all pulled to their maxima at the time of the sun's zenith.

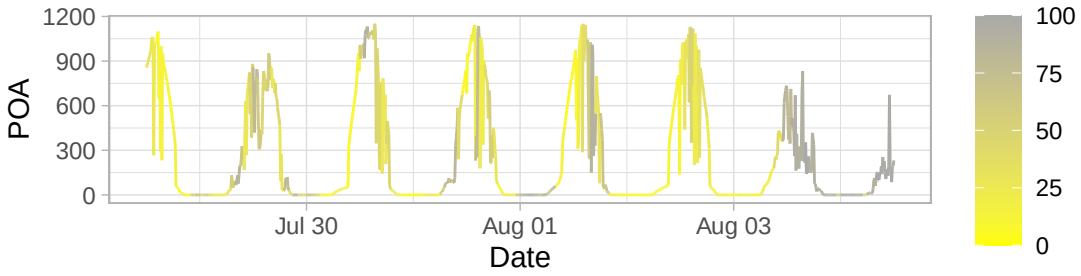


The time series decomposition into trend, seasonal, and remaining highlights how clear the daily cycle is, but also how variable the trend is. Note the height bars on the left which put the scale differences into perspective. The remainder is on the same order as the original data, indicating a lot of variability.



2.3.2 POA vs Time and Cloud Cover

The plots below show the impact of cloud cover on the POA. The first plot gives the last 7 days and it is clear how the last two days in particular appear cloudy and the POA is lower, accordingly. The bottom plots below show the seasonality on a linear and polar scale. We observe how during daylight times, the POA is generally high unless it is cloudy. Similarly the polar plot gives a clear indication of the typical cycle and what happens when clouds are present.

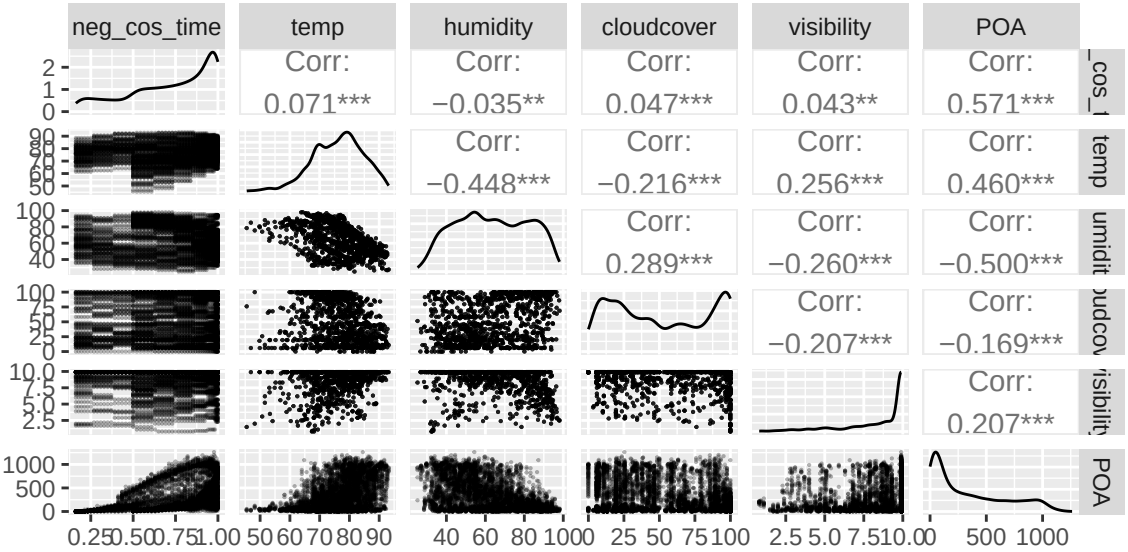


Compared to the previous seasonal plots, the main difference is the actual POA tends to be more binary and doesn't spend much time in the middle of the day at half irradiance. This misses the notion that days are either sunny or cloudy with either full or low irradiance, or if the day is not on the extremes of cloud cover, the POA randomly fluctuates from full to low irradiance and back. This describes all days reasonably well, but it doesn't describe any one day particularly well.

2.3.3 Statistical Plots

Finally, we look at correlations amongst variables. We have transformed the time of day by computing the cosine of the time with a 24-hour period, halving it, and adding one. This results in a feature which is 0 at midnight, increases to 1 at noon, and then decreases back to 0.

We want to first examine covariance amongst variables which we will do with a correlation pairplot. We will modify the dataset to only look at the times between 5:00 AM and 9:00 PM because we know the POA will be 0 outside of this. This ensures we are seeing actual covariance amongst the variables during the time we are concerned with. We are not including environmental observations from the original data as we have more complete weather data available which includes the same information. Through our own covariance analysis and prior knowledge about the process, we selected the following six features for further review.



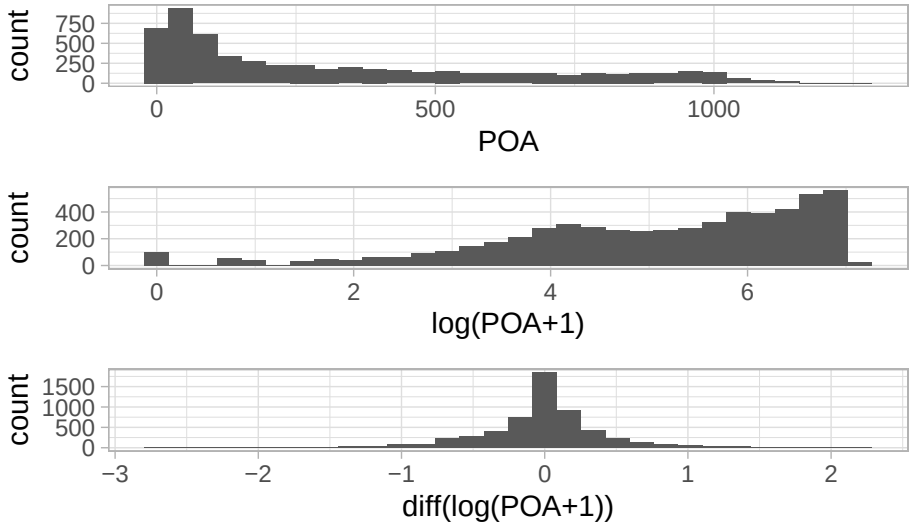
Looking at the pairplot above, we can observe the correlations by looking at the scatter and density plots. There are no particularly strong predictors for POA, and we know that weather phenomena have dynamic

components which all interact with each other. Cloud cover and visibility, however, are not entirely involved in those dynamics so we expect they will provide the best additional prediction information.

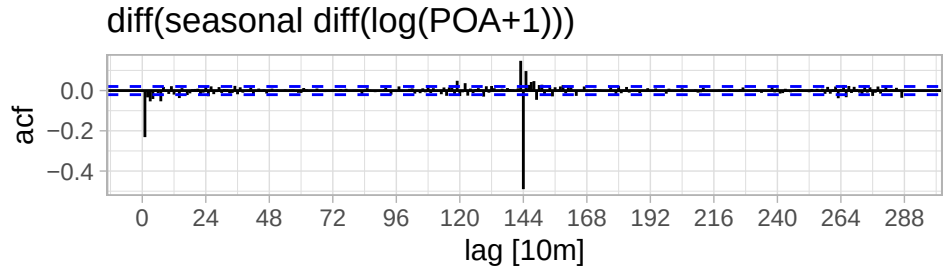
2.3.4 Data Transformations

We transform the data so it becomes stationary and satisfies the model assumptions of the time-series models that we employ in the subsequent sections.

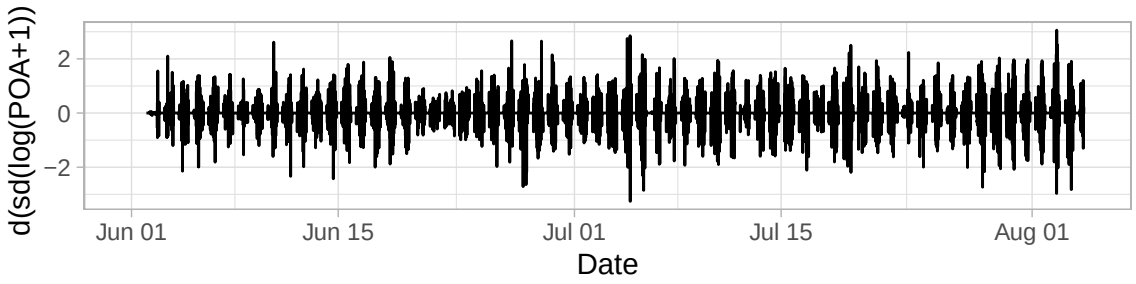
2.3.4.1 Achieve a Normal Distribution We transformed the POA measurements using a $log(x+1)$ transform and differenced it to achieve a normal distribution. The plots below show the resulting distributions. The majority of values are 0 because there is no irradiance at night, and $log(0)$ is undefined. Adding 1 to all POA measurements avoids undefined values. The plots below are for 5:00 AM to 9:00 PM.



2.3.4.2 Time Series Stationarity We use a combination of seasonal and non-seasonal differencing so that the resulting transform resembles white noise. Ultimately, we found that a non-seasonal difference of the seasonal difference of the log transform achieved the desired stationarity. The significant lags can be incorporated directly into the SARIMA models.



We will use this information when building time-series models by using appropriate hyperparameters for the seasonal and nonseasonal difference, autoregressive lags, and moving average lags. The plots below show how the POA time-series ends as white noise after the transforms and differencing.

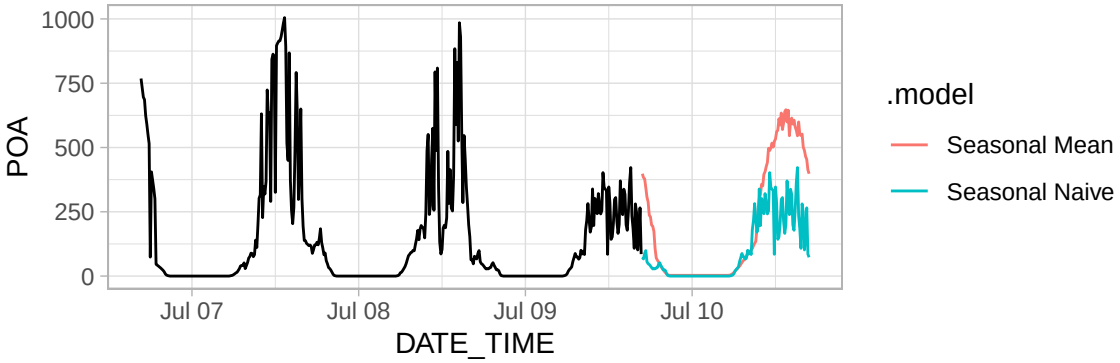


3 Models

In this section we define benchmark and candidate models for comparison. We used 60% of the available data for training and the remaining 40% as test data to evaluate the models on unseen data.

3.1 Benchmark Models

We set benchmarks using simple models for comparison with the more complex models we propose. We can immediately eliminate models which do not account for seasonality such as naive, drift, and mean. Instead, we use Seasonal Naive which just uses the value from the same time the day before and Seasonal Mean which has averaged the POA at that time for the entire dataset. These are both reasonable approaches which will provide a good indication of whether the more complicated models are producing commensurately better forecasts. Seasonal Naive is sensible as we observed in earlier plots that a given day is likely to look similar to the day before. Seasonal Mean is also sensible because we observed that data is equally distributed about the mean at each observation.



Looking at the plots above, we have reasonable forecasts. Here we can see that our Seasonal Mean is outperforming the Seasonal Naive model in the Mean Absolute Error (MAE) metric and significantly more so in the Root Mean Sqaured Error (RMSE). This indicates that the Seasonal Naive Model's error is often associated with outliers.

Model	MAE	RMSE	Type
Seasonal Mean	85.59	139.86	Benchmark
Seasonal Naive	98.06	209.70	Benchmark

3.2 Candidate Models

We provide a brief overview of more complex models employed, assumptions, and justifciations of their use in the following subsections. We then build, fit, and examine the models in the next section.

3.2.1 Time Series Linear Model

We tested two TSLMs, one with only one exogneous variable, cloud cover, and one with both cloud cover and visibility as exogenous variables. We limited the exogenous features to just these two because we know the other weather phenomena such as temperature, humidty, and dew point are all driven by the daily cycle of the sun. Using this knowledge helps narrow our scope to keep the model simpler and prevent overfitting and unnecessarily complex models.

3.2.2 SARIMA

Revisiting the ACF for the differenced, seasonally differenced log transform of the POA we can pick out appropriate parameters for the SARIMA model. We have already determined that the non-seasonal and seasonal differencing parameters should be$d = D = 1$. We also know that the amount of sunlight at time t is dependent on conditions just one moment ago. The ACF plot confirms this with the first lags being significant so we will set the non-seasonal autoregressive parameter to $p = 1$. We do not want to include a non-seasonal moving average term because the POA should not be returning to a mean. It typically goes from high POA to low without really settling around the mean so $q = 0$. We do however, want a seasonal moving average to capture the fact that one day is often very similar to the day before it, so $Q = 1$. We will set $P = 0$ because there is not a seasonal autoregressive component to the POA. Ultimately we will be using a $SARIMA(1, 1, 0)(0, 1, 1)_{144}$ with 144 representing the lags for one day.

3.2.3 SARIMA-X

Similar to the TSLM, we will use two SARIMA-X's - one with just cloud cover and one with both cloud cover and visiblity as exogenous variables. The reasoning for the hyperparmeters for the SARIMA models still holds, so the SARIMA part will also be $SARIMA(1, 1, 0)(0, 1, 1)_{144}$. By incorporating more information, we would only use a SARIMA-X if it perfomrs much better than the SARIMA.

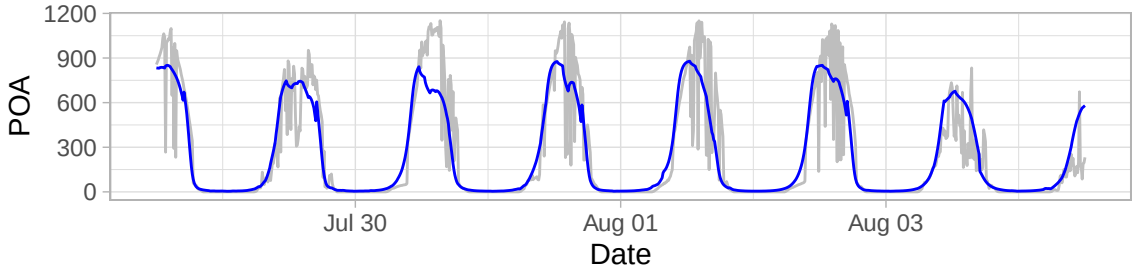
3.2.4 LSTM

Finally, we trained a recurrent neural network to compare against the conventional time series forecasting models. This was done using the Julia Programming Language's library for deep learning, Flux, with the following steps: - Select features: neg_cos_time, cloudcover, and visibility - Normalize all exogenous and target features. When using gradient methods for parameter estimation, it is important that variables be on a similar scale. We normalize because we know that the POA has a minimum of zero and we have observed the highest value achievable for our intents and purposes.

- 1. Batch the data. Training the LSTM is improved by providing it batches of data to evaluate the gradient of the loss function. This helps parameter adjustment be better for most of the data rather than just one or two observations.
- 2. Create the LSTM. We opted for a simple model for this task. For each observation, the first layer takes the 4 input variables at time, T (neg_cos_time, cloudcover, visibility, and POA) and outputs an 8-element vector. This is passed to a dense layer which takes the 8-element vector and outputs a single value and applies the sigmoid nonlinear activation function. This is the resultant (normalized) POA prediction.
- 3. Select the loss function. We used MAE as that is what we are optimizing for
- 4. Select the optimizer. We used Adam which is the most widespread optimizer as its "momentum" allows it to "roll past" local minima.
- 5. Iteratively train the model, adjusting the learning rate of the optimizer between training iterations. For this, it was fed batches of data of the 4 input variables as the "X" and the actual normalized POA at the next time step for each sample in the batch as the "y".

The special aspect of LSTMs compared to normal "feed-forward" neural networks is that they have a "state" which is retained between calls. When training, the first batch would be used to condition the LSTM, and

then training would begin from the second batch to the end. This was also important to keep in mind when using the LSTM to make forecasts. We called the LSTM on historic data from time, $t = 1$ to $t = T$ to set the state. From then on, we would pass it the neg_cos_time feature, the "forecasted" cloud cover and visibility, and its own last prediction until we reached the end of the forecast horizon, 1 day, and then this process was repeated for the next day.



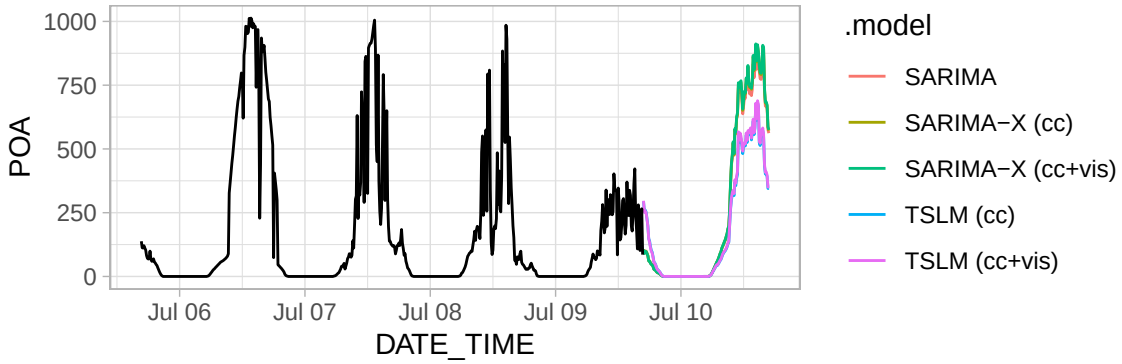
3.3 Model Performance

The table below shows the candidate models we considered:

Model
TSLM (cloud cover)
TSLM (cloud cover + visibility)
SARIMA$(1, 1, 0)(0, 1, 1)_{144}$
SARIMA-X$(1, 1, 0)(0, 1, 1)_{144}$ (cloud cover)
SARIMA-X$(1, 1, 0)(0, 1, 1)_{144}$ (cloud cover + visibility)
LongShort-term Memory

Finally, we look at the residuals for the the best models, TSLM and then the SARIMA below it.

Looking at the forecasts, we can confirm that the models are performing as intended. In the next section we will cover model evaluation and selection.



Immediately we notice there is practically no difference between the SARIMA and SARIMA-X models or between the two TSLM models. We will favor the SARIMA and TSLM (cloud cover) because their simplicity will likely result in better generalizations.

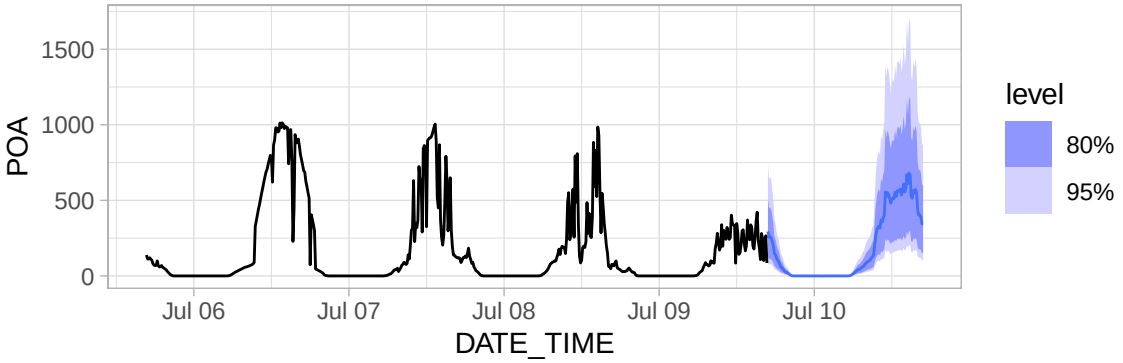
4 Model Evaluation

We will focus on the MAE as this is the metric we are seeking to optimize. For all the models, the $MAE = 85 \pm 13$, indicating that they are making similar forecasts. The two TSLMs are performing the best with the one including visibility negligibly better than the one that uses only cloud cover. We then

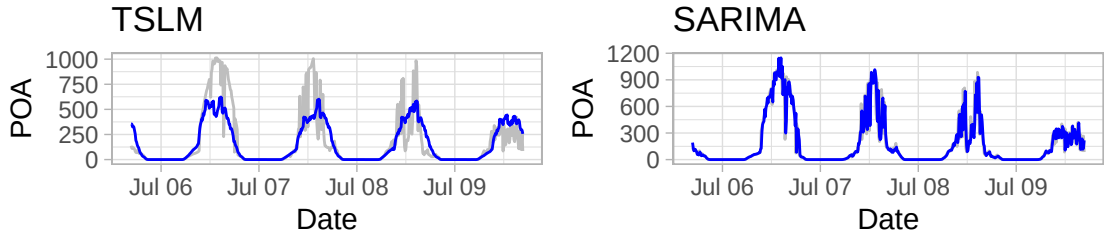
notice that all of the models beat the benchmarks, although the more complex LSTM, and SARIMA-Xs just barely so. Finally, it is interesting to note how the RMSE metric would reorder the models with TSLMs still dominating, but then the Seasonal Mean, SARIMA, and LSTM. This indicates that the SARIMA error may be typically more accurate, but has more volatile error.

Model	MAE	RMSE	Type
TSLM (cc+vis)	76.53	129.07	Candidate
TSLM (cc)	76.62	129.90	Candidate
SARIMA	78.28	148.50	Candidate
SARIMA-X (cc)	80.41	153.68	Candidate
SARIMA-X (cc+vis)	81.17	155.54	Candidate
LSTM	85.31	152.25	Candidate
Seasonal Mean	85.59	139.86	Benchmark
Seasonal Naive	98.06	209.70	Benchmark

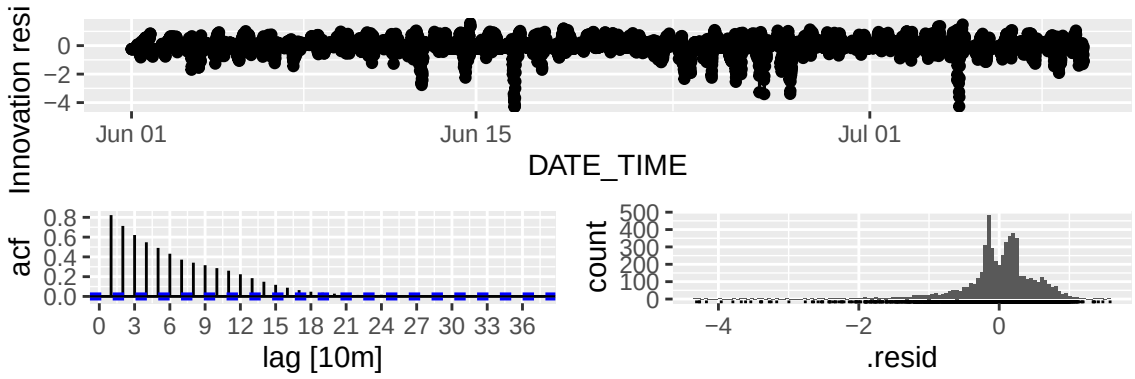
The plot and confidence intervals shown below for the TSLM (cc) are acceptable.



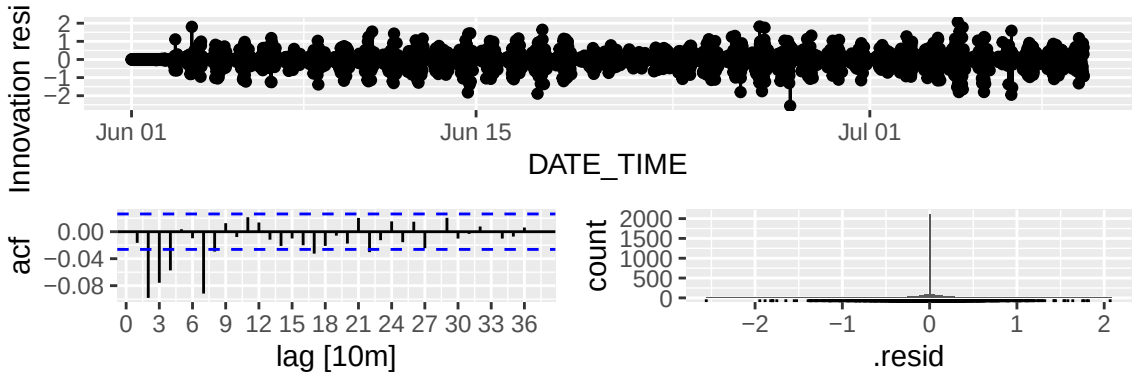
Looking at the training predictions for the TSLM and the SARIMA below, we can see the TSLM generally captures the moving average while the SARIMA is nearly a perfect fit. This explains why the TSLM testing error is better because the SARIMA has overfit the training data.



Finally, we look at the residuals for the the best models, TSLM and then the SARIMA below it.



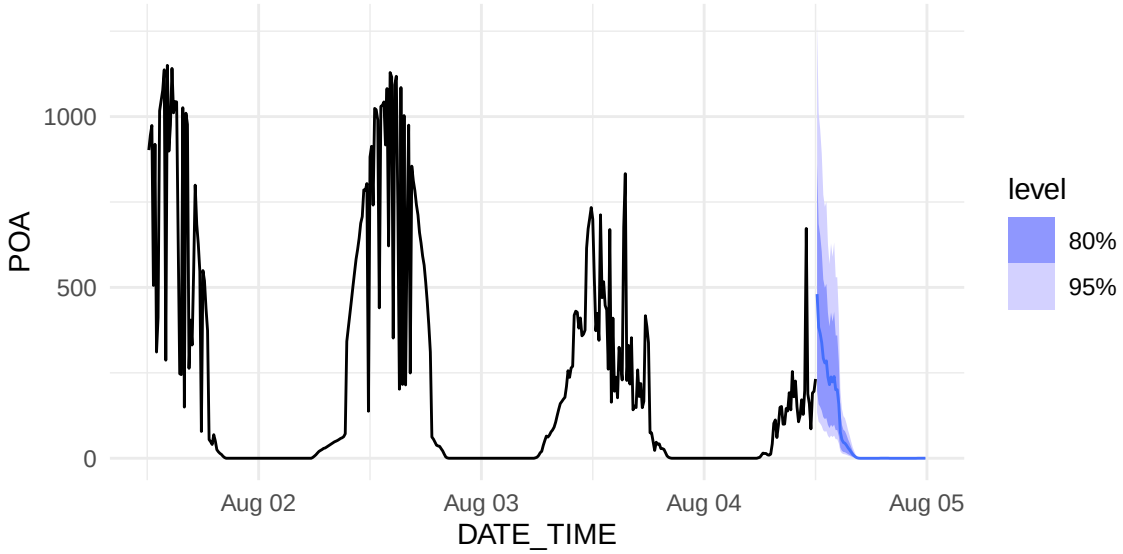
And the SARIMA:



We can see that the SARIMA model is has correlation with several of its first 10-lags. This can explain the wildly large confidence intervals and indicates that the ideal parameters have not been found, if they exist. The TSLM also has correlation with its lags, but that is to be expected as there is not an auto-regressive term in the TSLM that is supposed to reduce that. The residual error histogram shows we are close to normal distribution. The two peaks are easily understood as the binary effect of either the sun is out or it isn't.

5 Final Predicitions and Conclusions

Now that we have selected our model, we will train it on the last data available before making predictions on the final test data. Lookng at the final forecast, it appears reasonable as it looks like the past two days had a lot of cloud cover.



The major takeaway from this study is that it is important to incorporate practical knowledge of the system and then find data which will provide a model with that information. Without this, the TSLM model would not have been effective. The other aspect that proved to be a surprise is how effective very basic techniques can be in systems such as these which are very cyclical. It is estimated that the specific times of sudden drops and raises in the POA measurement could only be predicted if very detailed and computationally prohibitive numerical weather models were used to simulate the skies.