Forecasting of Monash Solar Farms

Alisha Memar, Sam Cropman and Priyom Sarkar

Faculty of Science, Monash University

ADS2002: Data challenges 4

24 October, 2021

Table of contents

Executive summary

One of the biggest upcoming challenges in the 21st century is tackling climate change. And to do so the recent use of renewable energy such as solar and wind has become essential in overcoming this issue. Thus, Monash University has set a target to have Net-zero commissions by 2030, having set up microgrid solar installations and a battery for energy storage. However, due to some limitations with renewable energy, only being available when the sun is shining or the wind is blowing, often there is a misalignment between production and consumption. Thus, the importance of forecasting both electric consumption and production is key in order to produce an optimal battery schedule that will minimize the cost of electricity. So in this project, the main objective is to produce forecasts for the building consumption and solar production, after which these forecasts may be used for developing an optimal battery schedule.

The data provided to us in this project was a tsf file that contains time series data of six solar panels and six buildings with varying start dates. With this data, there were some major complications that negatively impact the data quality and therefore could have impacted our modeling. To remedy this issue, various data cleaning and preprocessing techniques were utilized in both Python and R to ensure that the data was ready for modeling.

Moreover, there was also an additional data set that was obtained. This data was sourced from the BOM, providing the weather conditions from several weather stations from a variety of days. Similar to the data time-series data that was provided, there were also issues with data quality, however, not to the same extent. To rectify these issues, we utilized Python to clean and preprocess this particular data.

Before delving into our modelling, time-series analysis and exploratory data analysis were conducted to build familiarity with our data. It was found that there was some sort of weak trend occurring with each of the building time-series data, where there seems to be a reduction in building consumption in the more recent years. Also, we also observed some degree of seasonality present in both building consumption and solar production. Additionally, we also found there was some correlation between the weather variables and

solar production. We did this by using a correlation matrix to highlight this. What we found was that there was a strong positive correlation between 'solar exposure' and solar production.

After processing and examining the data, there were a couple of forecasting models that were shortlisted. These models include a seasonal naive model, where the forecasts produced are essentially the last observed value that has occurred during that season and served as our benchmark model. The next model we examined was the ARIMA (Autoregressive-Integrated-Moving average) model. This model produces forecasts by using its past variables to predict the future, (the AR term), and incorporates the past forecast errors as well (the MA term) along with ensuring that the time-series data is stationary. In addition, to the ARIMA model, a SARIMA model was also looked at. The SARIMA model is simply an ARIMA model with a seasonality component. Lastly, the final model we used was an STL decomposition model. Using the Loess method for estimating non-linear relationships we were able to create a model that considers a much higher degree of seasonality and are able to identify which factors influence the time-series data. To incorporate these models, we utilised the 'fpp3' package that was available in R to determine the best model for each time series data and used it to develop the forecasts.

After developing the models, we decided to test the accuracy of the models by comparing MASE (mean average scaled errors) scores. By doing this, we would be able to determine which model will be used to forecast the month of October. What we found from the accuracy scores was that the STL decomposition model performed the best, yielding the lowest MASE score, due to its ability to handle complex seasonality. With regards to the ARIMA model, there were some issues with the automated function where the optimal model would become computationally heavy and therefore result in a worse accuracy score, even performing worse than the seasonal naive model. Using the lowest overall accuracy score, the STL model was used to produce forecasts for the month of October for both buildings and solar panels. Although unable to compare our forecasts, by being able to visualise the forecasts we can observe that they follow a similar pattern to the months prior for the buildings. However, with the solar panels, it was observed that there were no zero values forecasted which are strange considering that it was abundant throughout the time series.

Overall, this project demonstrated that an STL model proved to be the most effective when forecasting for models that had sub-daily seasonality. However, this model also had its complications when forecasting solar production, suggesting that an alternative model should have been investigated, one that takes a multivariate approach.

Introduction

In an effort to achieve net-zero emissions by 2030, Monash University has taken the initiative to install solar panels around buildings in an effort to use a renewable energy source. One of the major complications of using solar panels is that there is a certain amount of time in which the energy can be produced and used for building electricity. Thus, a way of mitigating this issue is utilising a battery schedule to provide power to the buildings when solar production is unavailable. To provide the most optimal battery schedule, it is important to know the amount of energy the buildings consumed and energy produced by the solar panels in an effort to minimize the cost of electricity. Thus, the key objective of this project is to forecast both solar production and building consumption for the month of October, which these forecasts may be used for developing an optimal battery schedule. To produce this forecast, a variety of data cleaning and preprocessing needed to take place to ensure that the data was ready for forecasting. After we investigated the different models that may be used to produce forecasts and compare the accuracy between the models.

Data Preprocessing and Cleaning

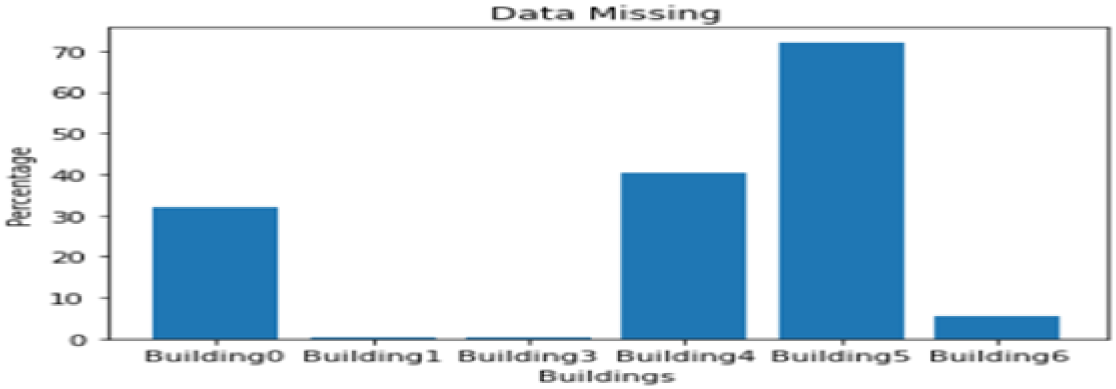
Data quality

There were two datasets that were made available to us, the first being a .tsf file containing the 15 minutely energy demand and production of buildings and solar panels. Along with this data, the file also contained information with regards to the length of each time series and whether there were any missing values, which was indicated by a "?" symbol. Importing this data was made easy due to the scripts that were provided to convert the .tsf file into a pandas' data frame or tsibble. Using the r-script available to us, we were successfully able to convert the .tsf file into tsibble which was then exported as a csv, ready to be read in as a pandas data frame. Despite the script being available to convert the tsf file into a pandas' data frame, utilizing the r-script was easier due to the familiarity with the language. Following the importation, some basic pre-processing was done. This included dropping unnecessary columns and rows using pandas' drop function. In addition, the column names were also renamed to a simplistic term to make the data easier to interpret. Next, the time column was changed into a DateTime format and indexed which make it easier to plot and produce forecasts. Lastly, the data was split based on the name of the series allowing for each building and solar panel forecast to be produced.

With regards to the weather data, there were some issues trying to import the data since the script provided was not working giving a “403 forbidden error”. To overcome this issue, the csv files that was available for each weather station and weather statistic had to be downloaded from the BOM website and then merged. The process of merging was tedious since I created a new csv file that simply copied and pasted the data from the other files. After successfully creating the csv file it was read into a pandas data frame. Like the energy demand data, the time column was also indexed and any missing values (which was towards the end of data) was dropped.

Handling of NA Values and Outliers

When cleaning data, one of the important aspects is to ensure that there are NA values. To check for these missing values, we first see the percentage of data that was missing from each respective data frame that was created. This was done by creating a new data frame entirely which was showed the percentage missing from each time series and allowed for visualization of where the missing values were occurring from. As we can see from figure 1, we can see that the series with the most missing values include building 0, 4 and 5.



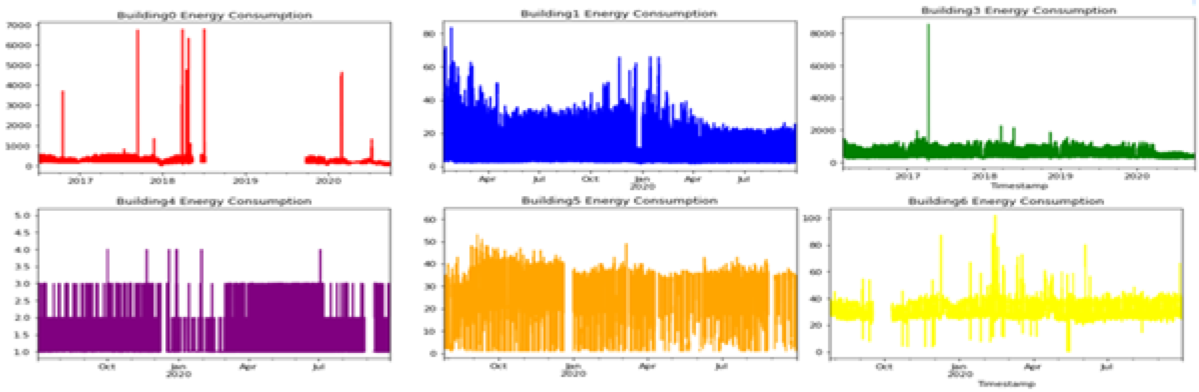


Figure I: Percentage of Data Missing and Original Time-Series Data

In addition, from figure 2 we can observe that are large gaps of data missing from a specific period, and this is prevalent in those with a large percentage of data missing. The reason to why these visualizations are important is because they provide an insight into how we can handle the missing values. In addition, to finding where the gaps occur in the data, we can also see the extreme outliers that may be present within each data frame. And as you can see from figure I the two buildings with extreme outliers are building 0 and 3. For these outliers, we simply removed them from the data and turn them into NA values. The way in which the outliers are removed was by using the quantile function. What this function does is essentially find the highest threshold value in which all other values may lie in a certain quantile. In this case, using 0.99 quantiles would retain all the necessary values and remove any extreme outliers. From figure II we can see the new time-series when the outliers have been removed and by removing these outliers it is expected that when testing the forecasting model that it would increase the accuracy scores.

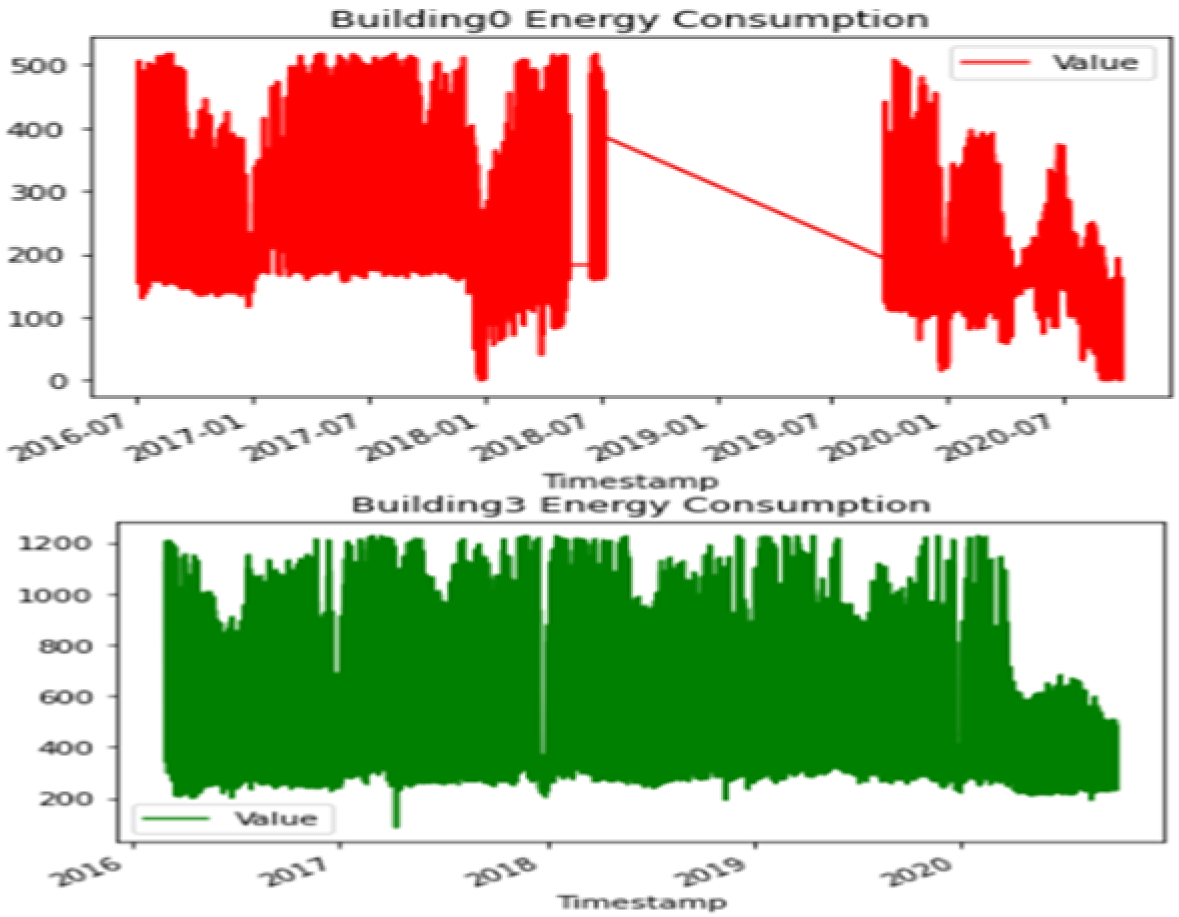


Figure II: Time-Series data of Building 0 and 3 outliers removed

Based on the percentage missing and the data that was available each way was utilized for different time series. For instance, building 1 and 3, the time series had an extremely small percentage of data missing, and to fill the NA values, linear interpolation was used. The way linear interpolation was executed was by using the interpolate function available with the pandas' package. With regards to building 4 and 5, the time series with the most missing values, the way we decided to handle the missing values is by using random sample imputation. Random sample imputation is the process of using the values present within the data set and assigning each missing value with a random value from the data. This method was chosen due to the high percentage of data that was missing and was done by using the apply function and np.where to find and replace NA values. In addition, with regards to building 0 and 6, we firstly condensed the time series by omitting values prior to the major gaps in the time series. After, like a previous method we used linear interpolation to fill the remaining missing values. Despite building 0 having a large amount of data missing and omitting those values to the large gap of missing data, the idea of condensing the time series was a valid approach since that specific time series had data that ranged from 2016. Thus, condensing the time series would still generate accurate forecasts since we still have a large amount of data available.

Exploratory Data Analysis

Before conducting any sort of modeling, time-series analysis was conducted to get the general idea of any trends, seasonality, and cycles that may be present in some of the datasets. In addition, some analysis with regards to the correlation between the different weather variables and the time-series data was also conducted.

Time Series Analysis

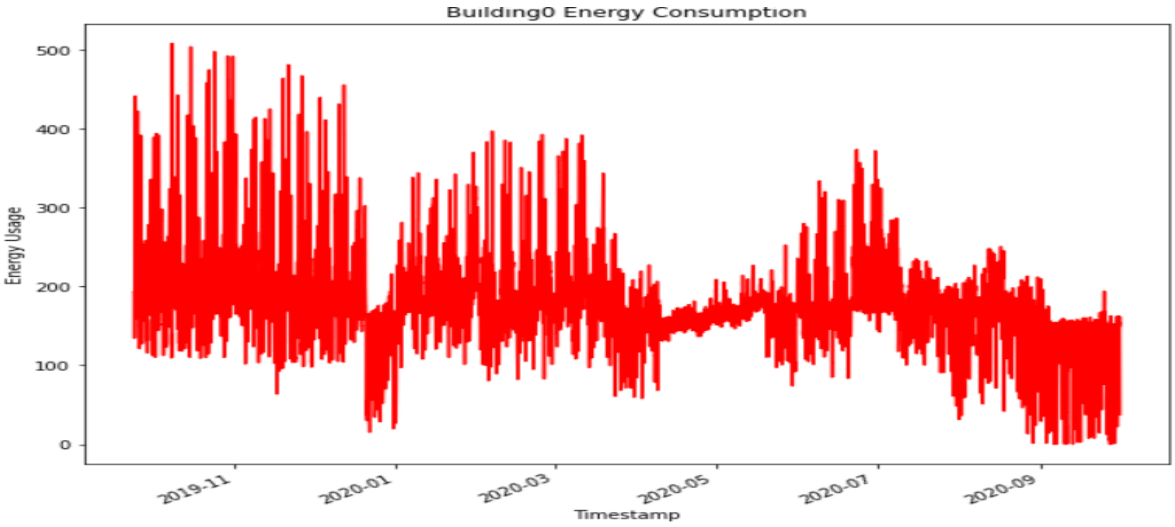


Figure III: Building 0 Time-Series Data

First and foremost we have Building 0. Figure III shows a time series plot containing the energy consumption over the months of November in 2019, January 2020, March 2020, May 2020, July 2020 and September 2020 for Building 0. The vertical red lines tell the time at which the energy has been used. The vertical axis is the amount of energy consumed. We can see the maximum energy consumption occurred in a few spikes in November 2019. We can see the outlier during the end of 2019 close to January 2020. There is a short downwards trend that does not display the overall downwards trend. The data in Figure III shows from approximately September 2020- September 2020 there is less energy consumption in Building 0. Overall in the course of that 10 months, we can see upwards spikes which may continue in the months continuing. However as the end of September is showing less energy usage, we can expect the same for the month of October 2020.

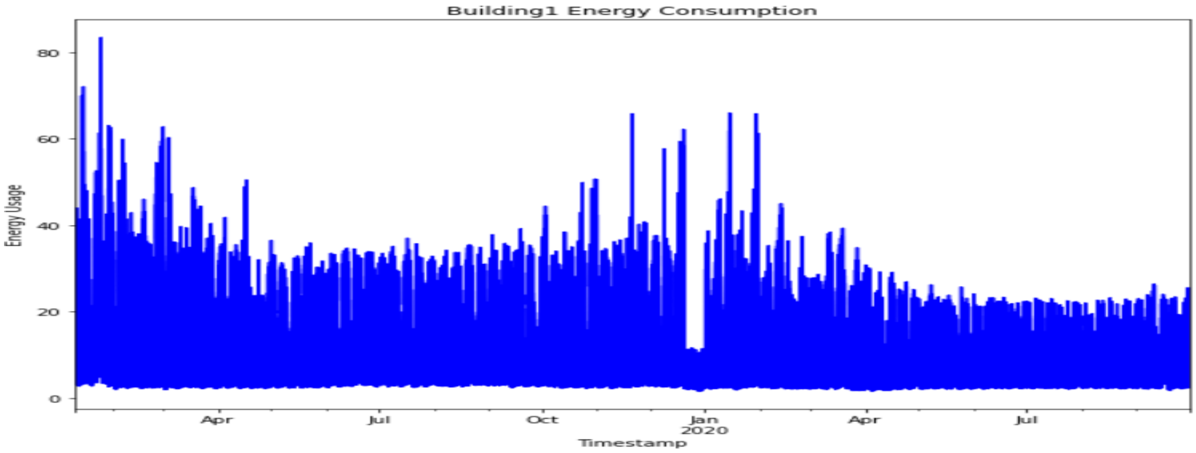


Figure IV: Time-Series data for Building 1

Figure IV presents a time series plot containing the energy consumption over the months of February 2019 until September 2020 for Building 1. The vertical blue lines tell the time at which the energy has been used. The vertical axis is the amount of energy consumed. We can see the maximum energy consumption occurred in 3 spikes from mid-November 2019 until February 2020. The outliers are present in a few uneven spikes from Jan 2019 until May 2019. Additionally, there is a dramatic decrease in usage in January 2020. Despite the outliers, there is a slow downwards trend. The data in Figure IV shows from approximately April 2020- September 2020 there is less energy consumption in Building 1. As the end of September is showing less energy usage, we can expect the same for the month of October 2020.

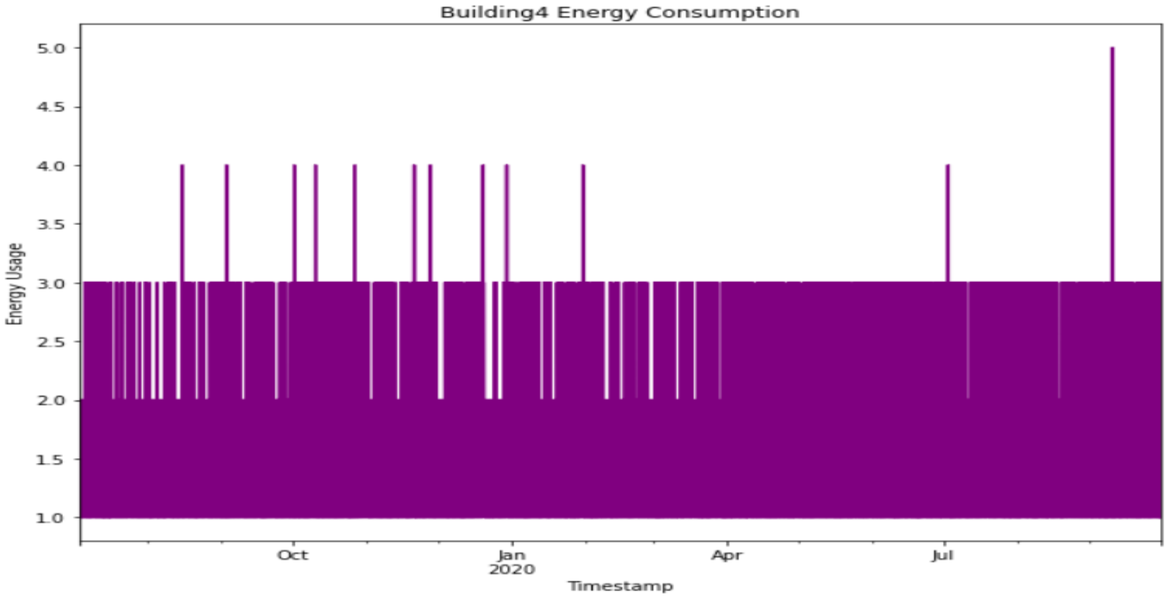


Figure V: Time-Series data building 4

Figure V presents a time series plot containing the energy consumption over the months of August in 2019 until September 2020 for Building 4. The vertical purple lines tell the time at which the energy has been used. The vertical axis is the amount of energy consumed. We can see the maximum energy consumption occurred in 2 spikes in June 2020 and August 2020. The mean energy consumed is 3.0 and occurs at most times during those time periods. However, some months had higher spikes or lower. The trend from September shows that there is a high chance for the month of October the energy consumption will be around 3.0 as a few spikes have already occurred in the previous month.

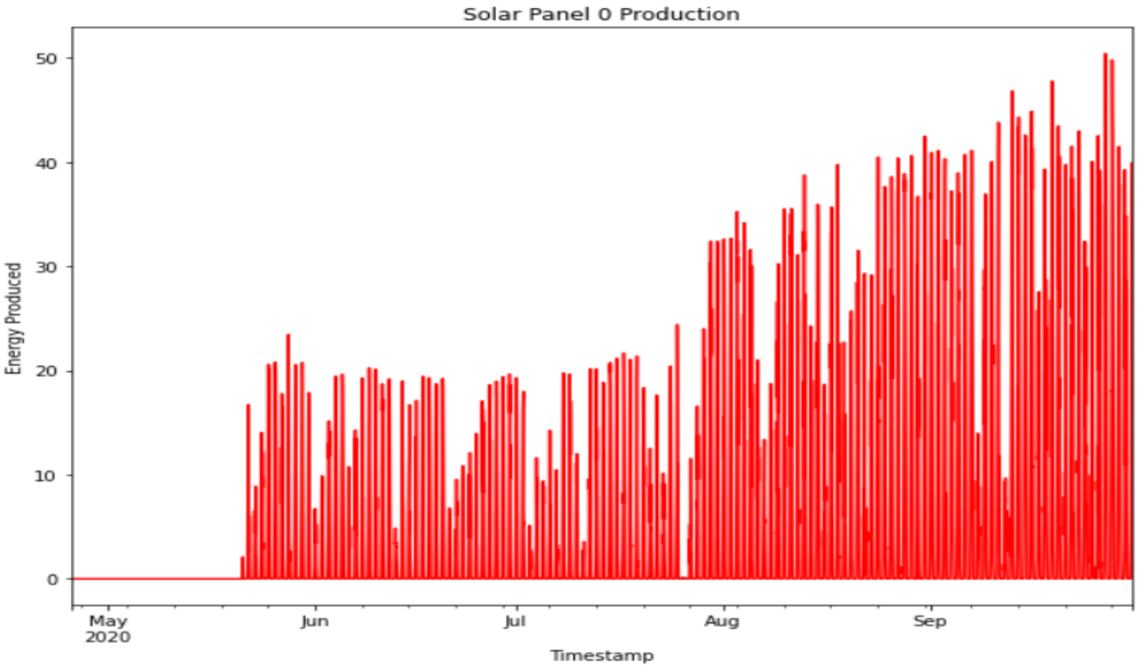


Figure VI: Time-Series data Solar Panel 0

Additionally, we had a look at the energy generation of the solar panels for each building. As shown in figure VI solar panel 0 had a drastic increase in energy generation. The vertical red lines represent the energy production for the months of May 2020 until January 2021. We can see that the maximum amount of energy produced occurred towards the end of 2020. For the month of October 2020, there is also a high production rate. This is controversial as discussed previously, for building 0 there was a low energy usage towards the month of October 2020.

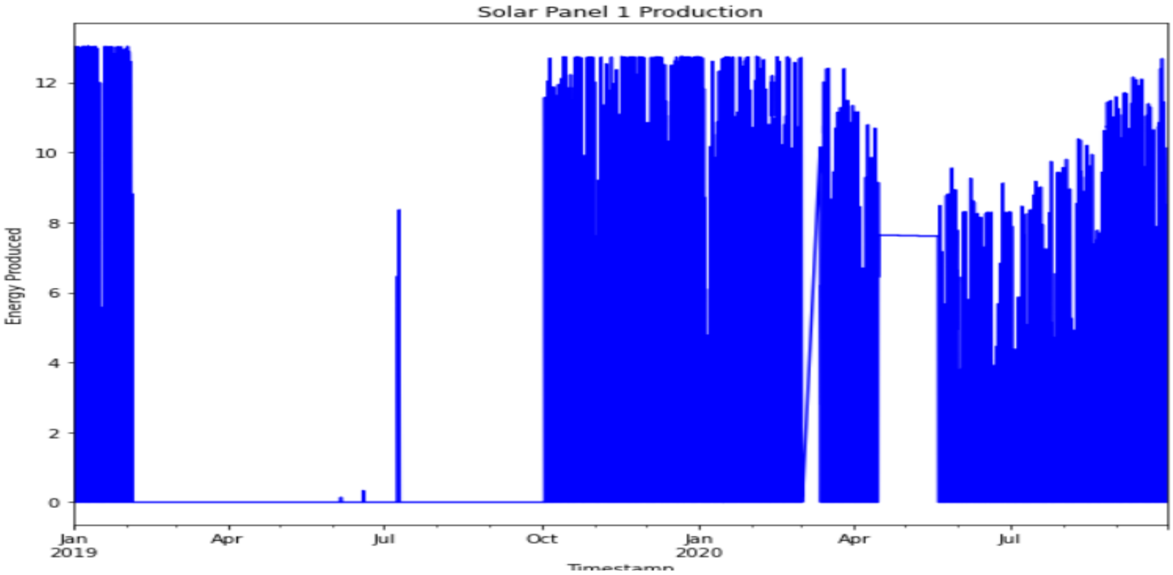


Figure VII: Time-Series data Solar Panel 1

Figure VII presents a time series plot containing the solar panel production over the months of January 2019 until September 2020 for Solar Panel 1. The vertical blue lines tells the time at which the energy has been produced. The vertical axis is the amount of energy produced. We can see the maximum energy production occurred from October 2019 until April 2020. Higher production can be a result of warmer weather. Moreover, there is a decrease in production in May- June 2020. However, the data in Figure VII shows from September 2020 there is an increase in energy production. As a result, it can be expected that the energy consumption of building 1 will also increase.

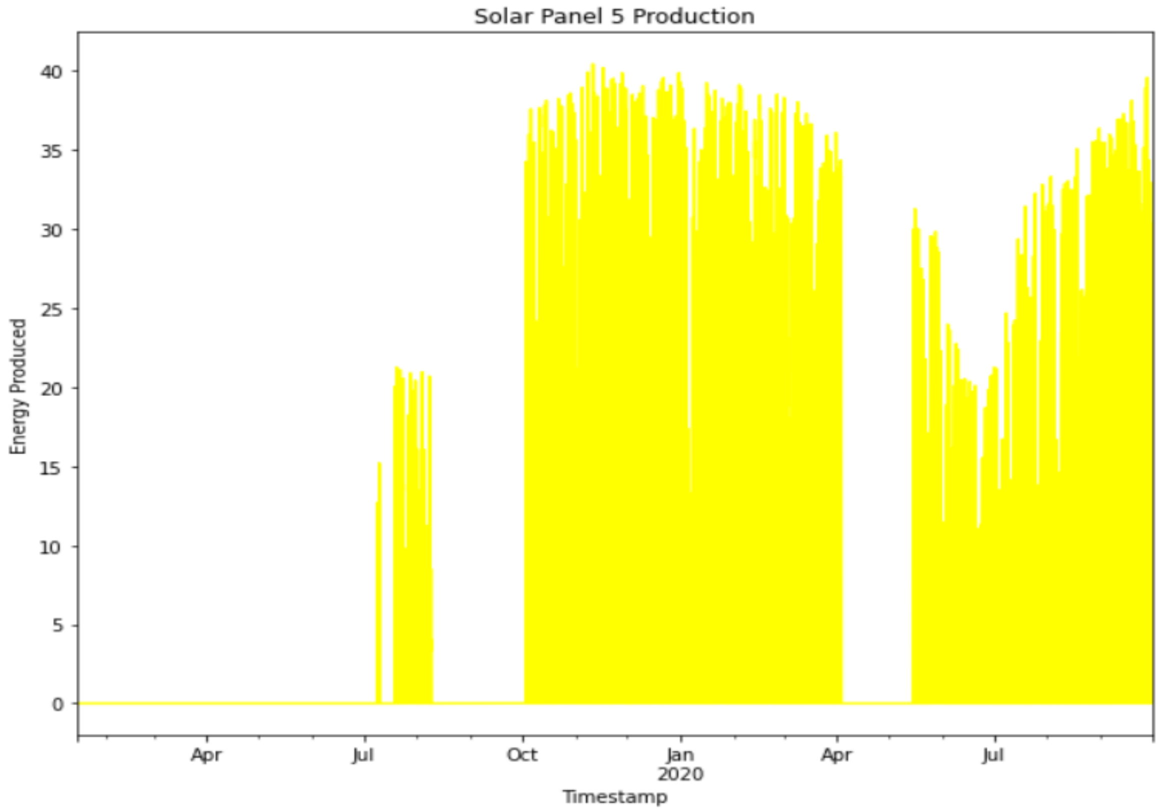


Figure VIII: Time-Series data Solar Panel 5

Figure VIII presents a time series plot containing the solar panel production over the months of February 2019 until September 2020 for Solar Panel 5. The vertical yellow lines tell the time at which the energy has been produced. The vertical axis is the amount of energy produced. We can see the maximum energy production occurred from October 2019 until April 2020 like the previous solar panel. There are time periods of very low or no energy production as seen in the months of April 2020 and June 2020. There is a decrease in production in May- June 2020. From June- July the low production can be a result of the colder weather going into winter. However, the data in Figure VIII shows from August 2020 there is an increase in energy production. Furthermore, it can be expected that the energy consumption of building 5 will also increase in October 2020.

Overall, what we observed between all the time-series data available was that there was some degree of seasonality that was present. The type of seasonality present is additive, also we can see some sort of trend within some time-series data.

Data Analysis

Before observing the correlation between the time series data and weather variables, we firstly resampled the time series data into a daily format getting the accumulated total consumption and production for that day. This was done to better capture the relationship between the variables.

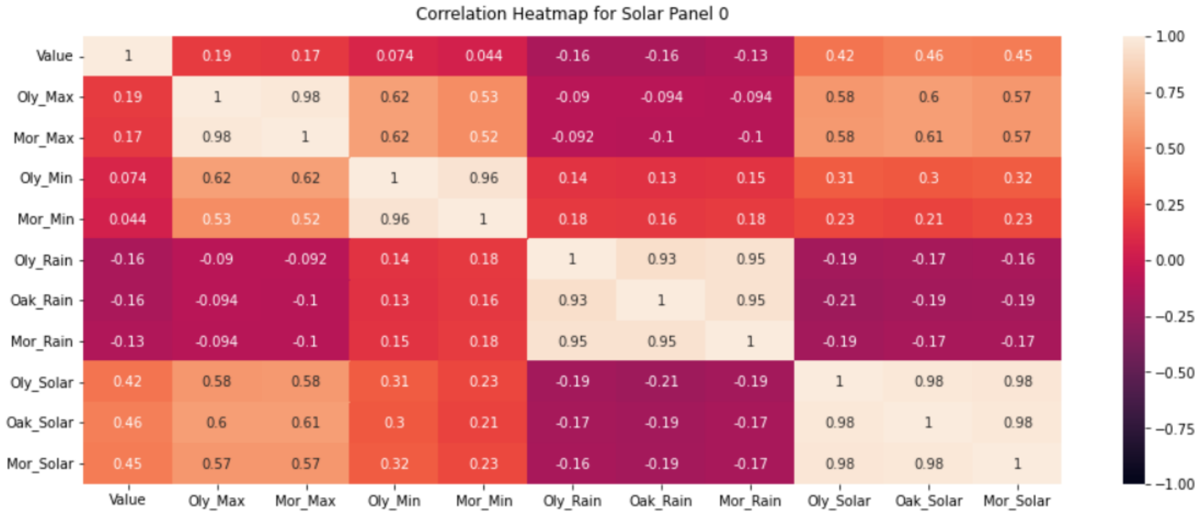


Figure XI: Correlation Heatmap for Solar Panel 0

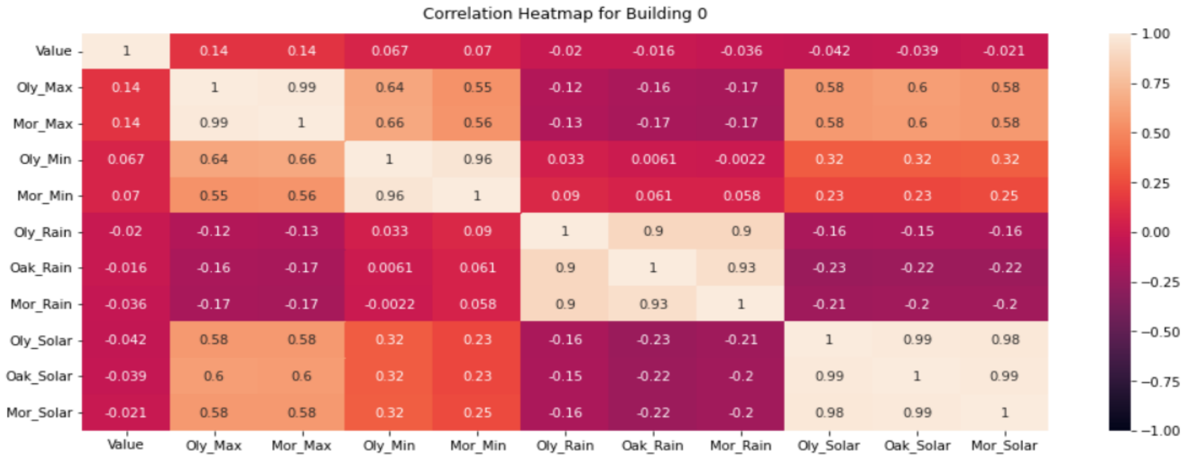


Figure X: Correlation Heatmap for Building 0

The correlation heatmap in Figure XI, illustrates the correlation between each variable. We are mainly interested in the correlation between the value and each weather variable. As we can see solar exposure has a positive correlation with solar panel production. This suggests that we can utilise solar exposure in our forecasting models if we choose to do multivariate forecasting. With regards to figure X, we can see that there is any correlation between

Overall, there is some positive correlation between solar exposure and solar panel production. Whereas, for the majority of the buildings there was weak to no correlation between the weather variables and energy consumption. Thus, this would suggest that we can take a univariate forecasting approach with the building consumption, while the solar exposure can take a multivariate approach.

Forecasting Models

Seasonal Naive

One of the models we first investigated was the seasonal naive model. This model essentially served as a benchmark model, to which our other models were compared. The way in which this model works is by forecasting future values that would be equal to the last observed value from the same season of the year or month. For instance, it will look at the previous data for October in previous years and make predictions based on that. The way this model was implemented by using the SNAIVE() function that is available in R.

ARIMA Model

The ARIMA model is a statistical analysis model that uses time series data to forecast future trends. ARIMA stands for Autoregressive Integrated Moving Average, and the model predicts future values based on past values and trends[1]. The ARIMA model requires time series data, which is a series of data points indexed in time order. It reflects the concepts of serial correlation, where future data points are assumed to be influenced based on past data points, recorded in a historical time series. An ARIMA model can be broken down into its 3 components.

- Autoregression (AR)
- Integrated (I)
- Moving Average (MA).

‘Autoregression’ refers to a model that shows a changing variable that regresses on its own lagged, or prior, values[1]. Any model that is autoregressive works under the premise that past values influence current values, and that they can be used to forecast a variable using a combination of past values of the same variable[2].

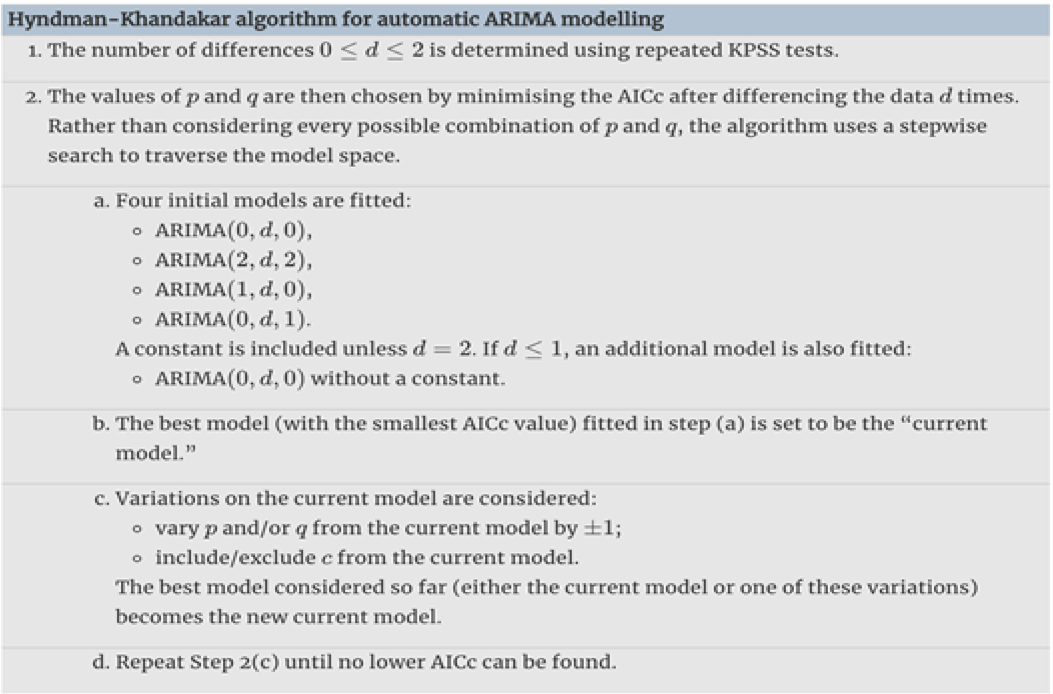
'Integrated' refers to the differencing of raw observations to allow for the time series to become stationary[1]. Seasonal differencing was required to make the data stationary, and the integrated aspect of ARIMA uses the difference between data values and the previous value rather than the value itself to build predictions.

'Moving Average' is a common technical analysis tool, often used when analysing stock prices, but can be applied in instances that involve trends. The ARIMA model takes this into account, as it is a calculation used to mitigate small, short-term fluctuations. It is used in the ARIMA model to create a series of averages of different subsets of the full data set, smoothing out the data over a specified time range. It does this by softening the small discrepancies and anomalies and incorporating the dependency between an observation and a residual error from a moving average model applied to prior observations[1].

The application of an ARIMA model required our data to be in a particular format. As aforementioned, the data must be stationary and must be a time series, as we are forecasting the variable of interest using a linear combination of past values of the variable[3]. A time series is considered to be stationary if its statistical properties do not depend on the time at which the series is observed[4]. As our dataset contains seasonality as well as trends, without manipulation it is not stationary, as, at different times of the year, seasonality would impact the values. The requirement for the ARIMA model to work with stationary data is that a stationary time series will have no predictable patterns in the long term, and so the model can use a linear combination of past data as an indicator of future values, without the impact of trends, or seasonality, and will have a somewhat constant variance[4].

Using differencing, we were able to make our data stationary, by computing the differences between consecutive recordings, which we used in the model instead of the original series values. Once the data is shown to have undergone some sort of differencing which is observed by the shape of the PACF and ACF plots, we are now required to fill in the ARIMA parameters of (P, D, Q). These terms refer to the Auto-regressive part, the amount of differencing required to make the data stationary, and the moving average part, respectively. The value of p, which is the autoregressive term, refers to the number of lags to be used as predictors. In addition, along with these parameters, a seasonal ARIMA model may also be adopted with the addition of three additional seasonal terms. This new SARIMA model is characterised by 6 terms, the same first 3 terms {p, d, q}, but also has a seasonal part {P, D, Q}$_M$. Where M is the number of observations per year. However, to manually input the parameters would be a tedious task and therefore the automated ARIMA model was used to determine the most optimal model for each time-series data.

The auto ARIMA function in R uses a variation of the Hyndman-Khandakar algorithm, which combines unit root tests, minimisation of the AIC (Akaike Information Criterion) and MLE (Maximum likelihood estimation) to obtain an ARIMA model. Below is a summary of the default behaviour of the auto ARIMA function that we chose to implement (Figure XI).



Hyndman-Khandakar algorithm for automatic ARIMA modelling

1. The number of differences $0 \leq d \leq 2$ is determined using repeated KPSS tests.
2. The values of p and q are then chosen by minimising the AICc after differencing the data d times. Rather than considering every possible combination of p and q, the algorithm uses a stepwise search to traverse the model space.
   a. Four initial models are fitted:
      - ARIMA$(0, d, 0)$,
      - ARIMA$(2, d, 2)$,
      - ARIMA$(1, d, 0)$,
      - ARIMA$(0, d, 1)$.

      A constant is included unless $d = 2$. If $d \leq 1$, an additional model is also fitted:
      - ARIMA$(0, d, 0)$ without a constant.

   b. The best model (with the smallest AICc value) fitted in step (a) is set to be the "current model."

   c. Variations on the current model are considered:
      - vary p and/or q from the current model by ± 1;
      - include/exclude c from the current model.

      The best model considered so far (either the current model or one of these variations) becomes the new current model.

   d. Repeat Step 2(c) until no lower AICc can be found.

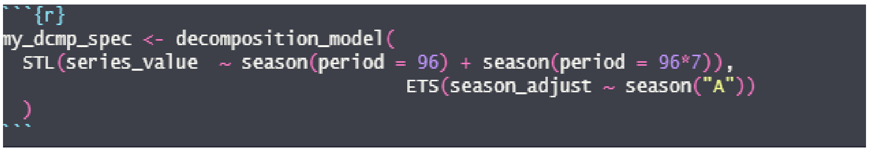
Figure XI: Auto ARIMA Summary

By using this automated procedure, we managed to reduce the time in creating a suitable model as there is belief that the most optimal ARIMA model would be selected.

STL Model

The STL (short for season and trend decomposition using Loess) is a decomposition technique, being able to break up a variety of seasonal patterns and trends to identify any strong components that may influence the time-series data. A major advantage of this forecasting technique is its ability to decompose the data into sub-daily seasonal patterns. Thus, for the case of our dataset, it is an extremely useful model to use due to the sub-daily seasonality that is present within the data. However, a limitation of using STL decomposition is that it cannot handle a time series that exhibits multiplicative seasonality. Although seeming to be a major complication, by simply using box-cox transformations the multiplicative seasonality can be changed into additive seasonality. However, in the case of our dataset, we observed that there was no multiplicative seasonality and therefore did not require any sort of transformation.

With regards to developing the STL model, we utilized R's decomposition_model() function to produce the model. Unlike the ARIMA model, where the parameters were automatically chosen, for this model, we had selected the parameters manually, ensuring that 15-minutely would be considered in our model.



~~~
```{r}
my_dcmp_spec <- decomposition_model(
  STL(series_value  ~ season(period = 96) + season(period = 96*7)),
                                       ETS(season_adjust ~ season("A"))
  )
```
~~~

Figure XII: STL Decomposition Model

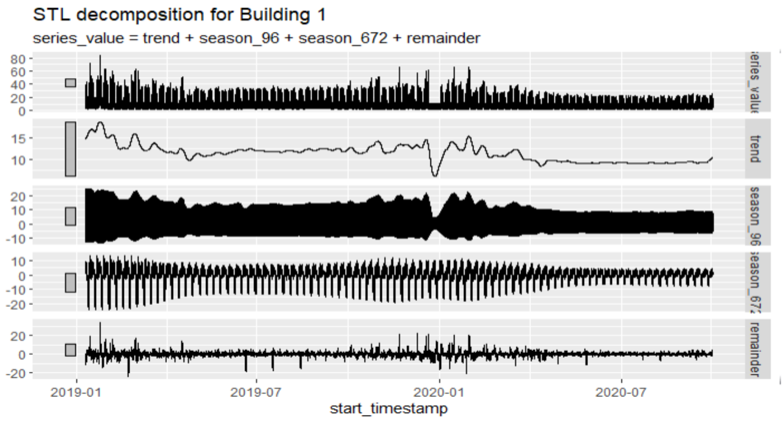


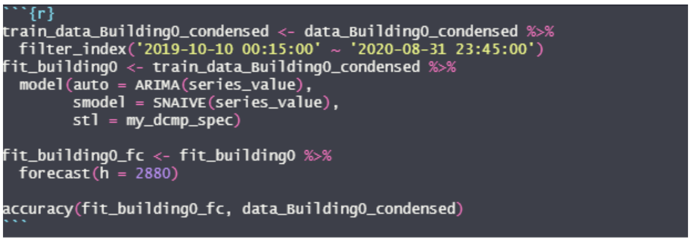
Figure XIII: STL Decomposition for Building 1

From the figure XII we can see that we have developed a model that takes into account the daily and weekly seasonality, (evident by the season parameters equating to 96 and 672 to represent the amount of 15 minutely intervals present within a day and a week). In addition, the ETS component of the model was utilized so that the model recognizes to use an additive seasonality when producing the forecasts. To demonstrate how important these seasonal components are, an STL decomposition plot was created. From the figure XIII, we can see there are two seasonal patterns, one a representation of the time of day and the other a representation of the time of the week. To assess the importance of these components, the grey vertical scales beside each plot would need to be examined since they represent the scale that certain components may have on the time-series data. For building 1 time-series data, we can observe that the daily and weekly seasonality have narrower bars which suggest that these components are significant. Whereas, the trend component is shown to have a wider grey bar and therefore would mean that the trend observed in this data is weak. This pattern of strong daily and weekly seasonality was also observed in the other time-series building data and solar production. Therefore, we used this STL model to forecast all the buildings and solar panels.

Results

Methodology of Results

Before analyzing our results, we must delve into the methodology in which the results were determined. The two methods in which we used to assess our models were time-series cross-validation and training/testing splitting. Cross-validation is a procedure where there are a series of test-sets, each consisting of a single observation[5]. Using these test sets the forecast accuracy is determined by averaging them. On the other hand, training-test splitting is simply partitioning the data into two sets, a training set and a testing set. By splitting the time-series data into these sets, we are able to determine the forecasting accuracy, since the forecasts generated for the period being tested can be easily compared. Ideally, both methods of assessing accuracy would have been used but due to how computationally heavy it was to produce accuracy results via cross-validation, the method was unfortunately dropped.



~~~
```{r}
train_data_Building0_condensed <- data_Building0_condensed %>%
  filter_index('2019-10-10 00:15:00' ~ '2020-08-31 23:45:00')
fit_building0 <- train_data_Building0_condensed %>%
  model(auto = ARIMA(series_value),
        smodel = SNAIVE(series_value),
        stl = my_dcmp_spec)

fit_building0_fc <- fit_building0 %>%
  forecast(h = 2880)

accuracy(fit_building0_fc, data_Building0_condensed)
```
~~~

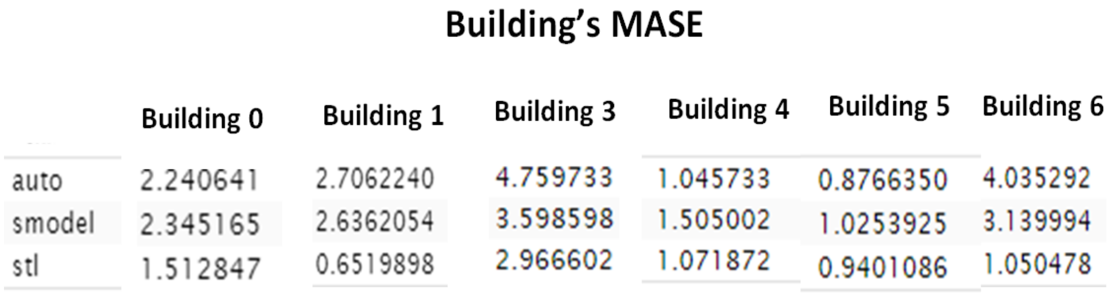
Figure XIV: Code for how training set was made

From figure XIV, we can see that we used the month of September as the test set and the rest of the time-series prior to that month were treated as the training set.

Assessing Accuracy

With regards to how the forecasting accuracy was assessed, we used the MASE (Mean-Absolute-Scaled-Error) to determine the accuracy of a model. The reason why this form of measurement was used was due to its ability to compare forecasting accuracy with time-series data that may have other units. Another accuracy measure that is commonly used is the MAPE (Mean-Absolute-Percentage-Error). Due to it being a unit free measure, it is often used to compare the accuracy of models with different units. However, there is a major complication that makes it difficult to assess the accuracy of our dataset. As previously mentioned, the solar panel production time series has a lot of null values, and this is to be expected given the way how solar panel production works. With the presence of null values in the data set, the MAPE would return an undetermined value. Thus, when measuring accuracy the MASE accuracy score was preferred to assess a model's performance.

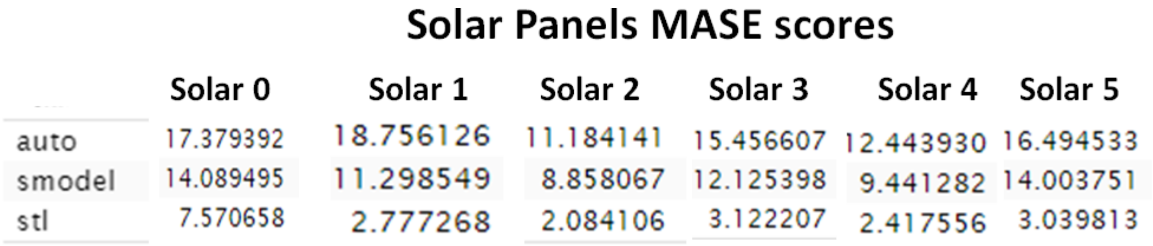
Determining The Optimal Model



Building's MASE

	Building 0	Building 1	Building 3	Building 4	Building 5	Building 6
auto	2.240641	2.7062240	4.759733	1.045733	0.8766350	4.035292
smodel	2.345165	2.6362054	3.598598	1.505002	1.0253925	3.139994
stl	1.512847	0.6519898	2.966602	1.071872	0.9401086	1.050478

Figure XV: Building MASE accuracy scores



Solar Panels MASE scores

	Solar 0	Solar 1	Solar 2	Solar 3	Solar 4	Solar 5
auto	17.379392	18.756126	11.184141	15.456607	12.443930	16.494533
smodel	14.089495	11.298549	8.858067	12.125398	9.441282	14.003751
stl	7.570658	2.777268	2.084106	3.122207	2.417556	3.039813

Figure XVI: Solar Panels MASE accuracy scores

From the results that we have yielded, we are able to determine that the STL decomposition model seemed to have performed the best overall, having the lowest MASE score for the majority of the buildings and solar panels. Following the STL decomposition model, the Seasonal Naive model yielded the best results followed by the automated ARIMA model. The results yielded were surprising because of how the automated ARIMA model performed. What these results may indicate is that the automated procedure did not produce a model which best captures the data. And upon further investigation, we have found that the automated procedure had only a max amount of lags of five that could be considered for each AR and MA parameter. This restriction set in place may suggest that the automated model was not designed to handle complex seasonalities, struggling to capture the sub-daily seasonality present within the data. In comparison, the STL model which, aforementioned, can handle the complex sub-daily seasonality present.

Although the overall performance of the STL model was by far the best, we observed that the MASE score for building 4 and 5 performed the best for the automated ARIMA model. This surprising result was likely due to having occurred due to the way missing values were handled during the pre-processing stage. As mentioned before, random sample imputation was used to fill the missing values and therefore it is likely that the degree of seasonality is reduced, also with random sample imputation, we are limiting the number of observed values in the training set since it is filling missing values with a limited range of values. Hence, that may be the reason why the automated ARIMA function is producing a better accuracy score.

Forecasting with the Optimal Model

From our findings, we used the STL model to produce forecasts for the month of October. Unfortunately, due to not having access to the actual consumption and production energy it is difficult to assess how accurate these forecasts are. Despite this, we can still visually assess the forecasts by plotting them with the original data set on which the models are based.

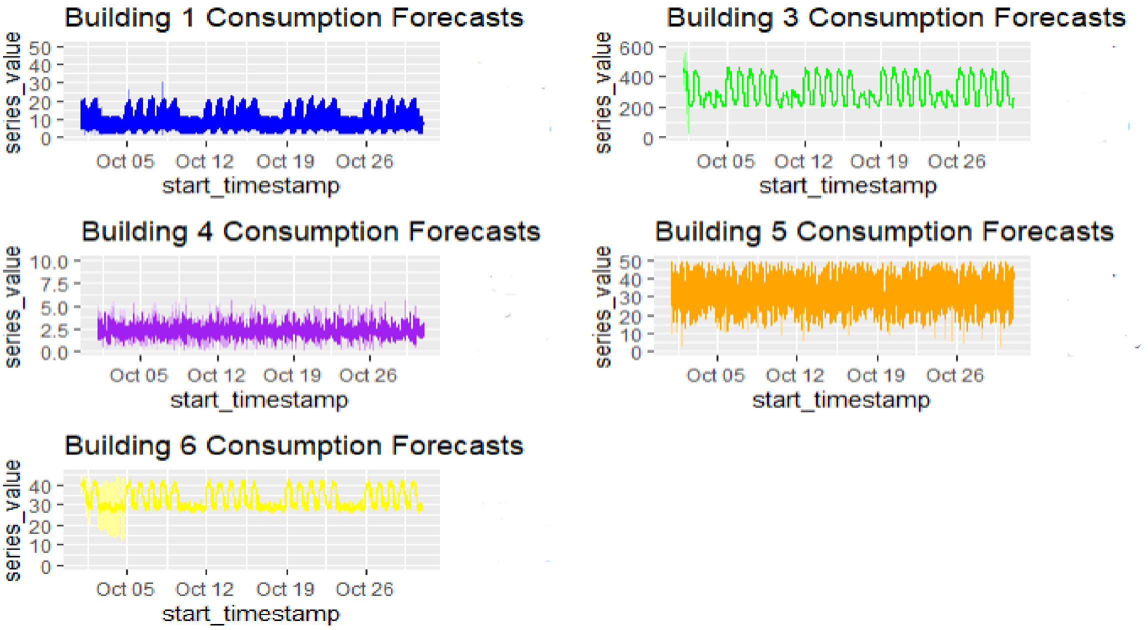


Figure XVII: Building Forecasts month of October

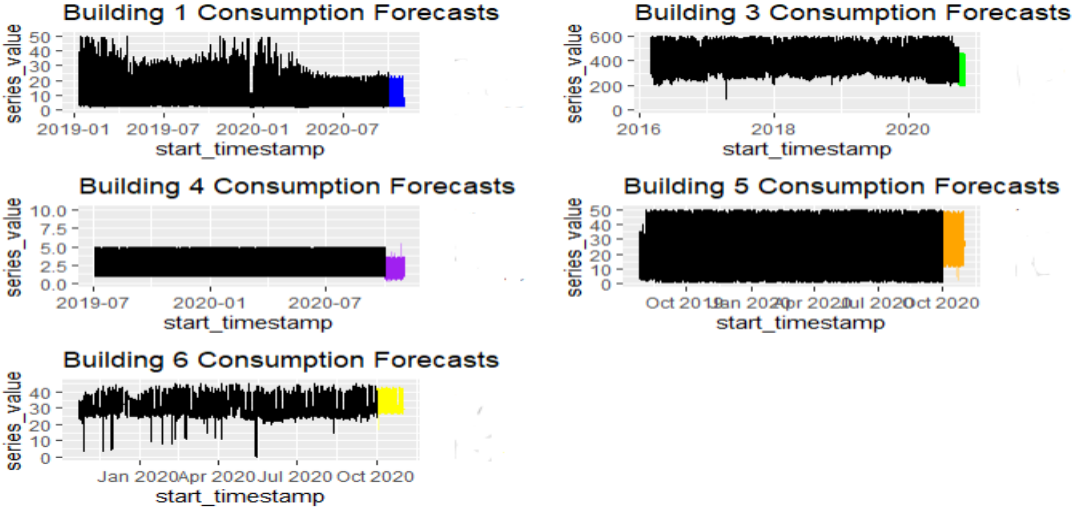


Figure XVIII: Building Forecasts with full data set

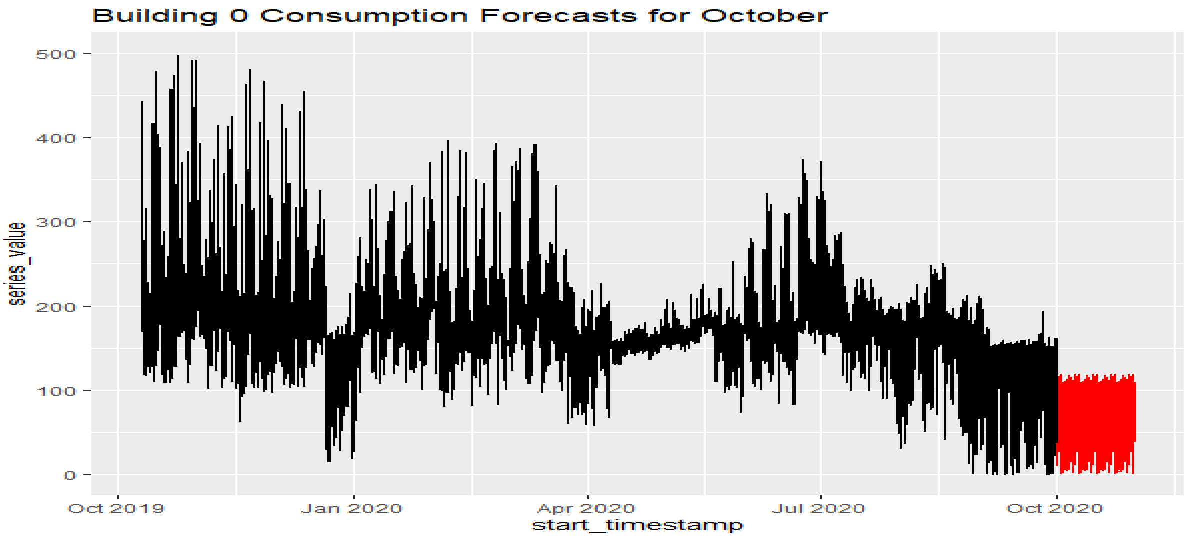


Figure XIX: Building 0 forecasts with full data set

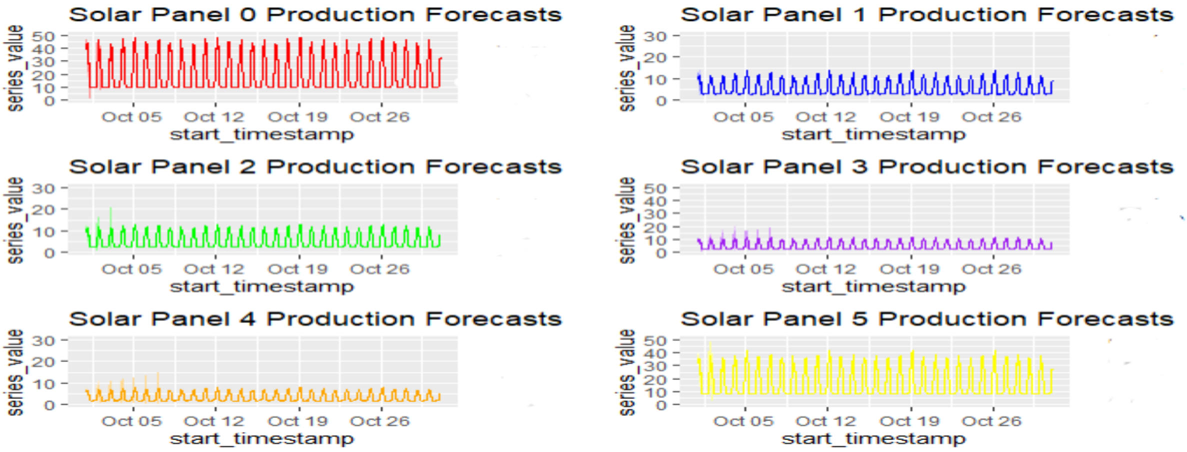


Figure XX: Solar Panel Forecasts month of October

From figure XVII, we can see that the forecasts follow a sinusoidal type pattern having up and down fluctuations. In addition, from figure XVIII we can see that the model also follows a pattern similar to the latest observations. With regards to building 0 forecasts, the forecasts produced were inaccurate given that negative forecasts were produced. This is likely to have occurred due to following a downward trend and no boundary set prevents it from going below zero. A way in which this issue could have been resolved was by making modifications to the time series data such as providing a logarithmic transformation. From figure XX we can see the solar production forecasts. Similar to the building forecasts, it also follows a sinusoidal pattern with no upward or downward trend. A flaw with the forecasts was that they failed to produce forecast values of zero. This is likely to occur due to the forecast function in R which would not allow for zero values to be forecasted.

Conclusion

In essence, the quality of the data made it difficult to immediately produce any sort of forecasts for the month of October due to the missing values present and outliers within the data. However, with the proper data cleaning and processing techniques, the time-series data was ready for manipulation and modelling.

The time-series and data analysis that was conducted provided invaluable information about the data set. Illustrating any trends and seasonalities that may be present within the data which would be useful when determining a model. In addition, the correlation matrix also provided insights as to what variables may impact the forecasting of solar production and building consumption.

From the analysis, three models were shortlisted. These models were a seasonal naive model where it served as the benchmark model for which other models were compared to. An automated ARIMA model was also used due to the high complexity in which the forecasts are generally are derived from and being the typical model utilised for any forecasting problem. Lastly, the STL decomposition model was investigated due to its robust ability to handle complex seasonality.

With regards to assessing the accuracy of the models, the STL model performed the best due to, aforementioned, its ability to handle sub-daily data. The automated ARIMA's model performance was initially underwhelming but after further investigation, it is understood as to why these branch of model underperformed due to the restrictions that are in place for the automated procedure. In addition, the actual forecast that were derived from the STL model are peculiar at times sometimes producing negative forecasts and was unable to produce forecasts that are zero values.

In the future, different types of forecasting models should have been investigated and incorporated. Some of the models that should have been investigated would be beyond the scope of traditional forecasting and utilised machine learning models such ensemble forecasting to produce forecasts. In addition, it would have also been ideal to use the weather dataset as well to forecast the solar production. This is because, as previously discovered, there was some correlation between some of the weather variables and solar production.

References

[1] https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

[2] https://www.investopedia.com/terms/a/autoregressive.asp

[3] https://otexts.com/fpp3/AR.html

[4] https://otexts.com/fpp3/stationarity.html#fn14

[5] https://otexts.com/fpp3/tscv.html

[6] https://www.machinelearningplus.com/time-series/arima-model-time-se