# Final Report: Solar Group 8

By Steven, Scott, Ellie, Khoi

---

## Executive summary

This project is based on the Monash campus and aims to explore the use of solar energy in powering the buildings and creating the most optimal building schedule to reduce the cost of using electricity as power is either drawn from the solar panels or the local electrical grid. Using historical data from the past few years of 6 Monash buildings each with solar panels, modelling techniques are used to predict the month of October 2020 in hopes of using this to create an optimal timetable of when and where to draw power from throughout the day. Throughout this project, problems that arose from the data mainly consisted of a substantial amount of missing data which had to be removed and having to test new modelling techniques. The modelling phase for the buildings required the data to be resampled into a daily time interval which compromised the aim to predict in 15 minute intervals due to the modelling technique being overly expensive to run in such a short time interval. While we were able to draw insight at when lectures and events should be held to minimise energy cost, an algorithm was not implemented to create an optimal timetable. However, two models were made to model building and solar data that resulted in reasonable results.

## Introduction

This project is based on the energy consumption and production of Monash's buildings and solar panels. The goal of this project is to be able to model the energy consumption and energy production for the month of October 2020 given the data, as well as creating an optimal timetable of lectures / events based on our predicted month of October values.

The datasets that were provided include:

- A Tsf file that contains the energy consumption of 6 buildings and the energy production of 6 solar panels on the Monash campus in kW/h
- Weather data sourced from the Australian Bureau of Meteorology
- Energy prices sourced from the Australian Energy Market Operator

Due to the nature of the data, it is expected that the weather data would have a strong relationship with the solar panel data as the weather would dictate the amount of solar energy that can be produced. Therefore, they were explored together during this project.

The modelling for this project is based on predicting a month's worth of observations based on historical data in 15 minute intervals as the building and solar data is segmented in 15 minute intervals. This comes with its own challenges as modelling in a short time interval will lead to the model either picking up a lot of noise and overfit or not being able to pick up the trends and underfit as it will be explained later. It was later found that using expensive models such as SARIMA was so computationally expensive that the data had to be resampled into a daily time frame rather than 15 minute time frame in order to model for the month of October 2020.

# Data Quality

## Imported Dataset problem

The first problem in this project involved having a data frame that could not be used immediately due to its irregular format.

| | series_name | start_timestamp | series_value |
|---|---|---|---|
| 0 | Building0 | 2016-07-03 21:30:00 | [283.8, 283.8, 283.8, 606.0, 606.0, 606.0, 606... |
| 1 | Building1 | 2019-01-09 23:15:00 | [8.1, 15.7, 22.8, 32.7, 8.1, 16.5, 24.7, 34.5,... |
| 2 | Building3 | 2016-03-01 04:15:00 | [1321.0, 1321.0, 1321.0, 1321.0, 1293.0, 1293.... |
| 3 | Building4 | 2019-07-03 04:45:00 | [2.0, NaN, 1.0, 2.0, NaN, 2.0, NaN, NaN, 2.0, ... |
| 4 | Building5 | 2019-07-25 23:00:00 | [30.0, 31.0, 24.0, 34.0, 30.0, 31.0, 26.0, 33.... |
| 5 | Building6 | 2019-07-25 01:45:00 | [36.8, 34.6, 34.6, 36.2, 36.2, 35.2, 35.2, 35.... |
| 6 | Solar0 | 2020-04-25 14:00:00 | [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ... |
| 7 | Solar1 | 2018-12-31 13:00:00 | [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ... |
| 8 | Solar2 | 2019-06-05 14:00:00 | [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ... |
| 9 | Solar3 | 2019-06-05 14:00:00 | [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ... |
| 10 | Solar4 | 2019-06-05 14:00:00 | [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ... |
| 11 | Solar5 | 2019-01-15 13:00:00 | [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ... |

Fig 1: Imported dataset

Converting the data from a tsf file into a data frame resulted in the buildings and solar panels being rows rather than columns, as well as the observations being contained inside an array. This was a problem as data techniques could not be used on such a data frame and so the dataset had to be manipulated in order to acquire a workable dataset. Upon further investigation of the imported dataset, it can be seen that building4 has Nan values as well as the initial observations for all the solar panels being 0, indicating that at the beginning of the time period, the solar panels were not generating energy. It can also be noted that there is no building 2 however, there is a solar 2. Therefore, throughout the project, it is assumed that building 3 corresponds to solar 2, building 4 corresponds to solar 3 ect. Furthermore, the times were in UTC, which also required the datetime to be converted to AEDT in order to get the correct timezone.

## Solar Panel data problem

Within the solar panel segment of the dataset, there were an abundance of missing values which can be viewed visually.

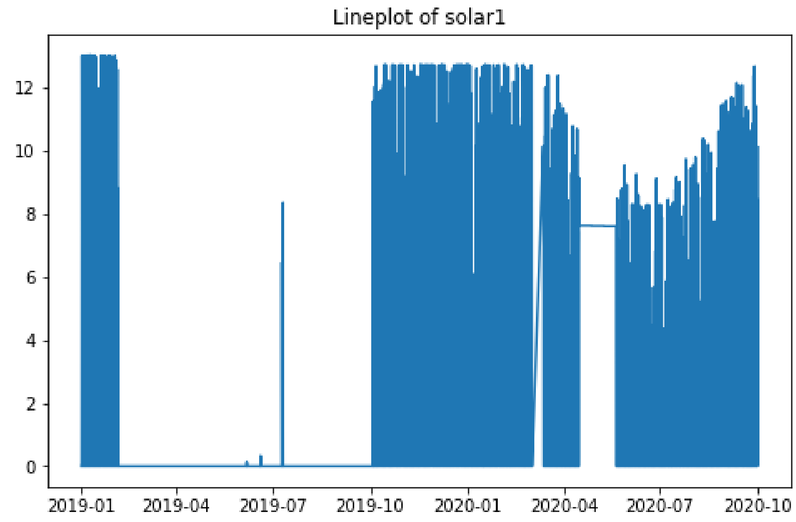


Fig 2: Line plot of solar panel 1

Taking a line plot of only solar panel 1, there are long periods of time where solar panel 1 is not operating or recording 0 within the 2 years worth of data despite it not being during lockdown times. The other solar panels suffer from the same problem which could be detrimental when modelling as the algorithms would attempt to learn from these long periods of empty observations and so they are removed during the modelling stage.

## Building data problem

Similarly with the solar panel observations, the building observations were not fully complete as they all contained missing values.

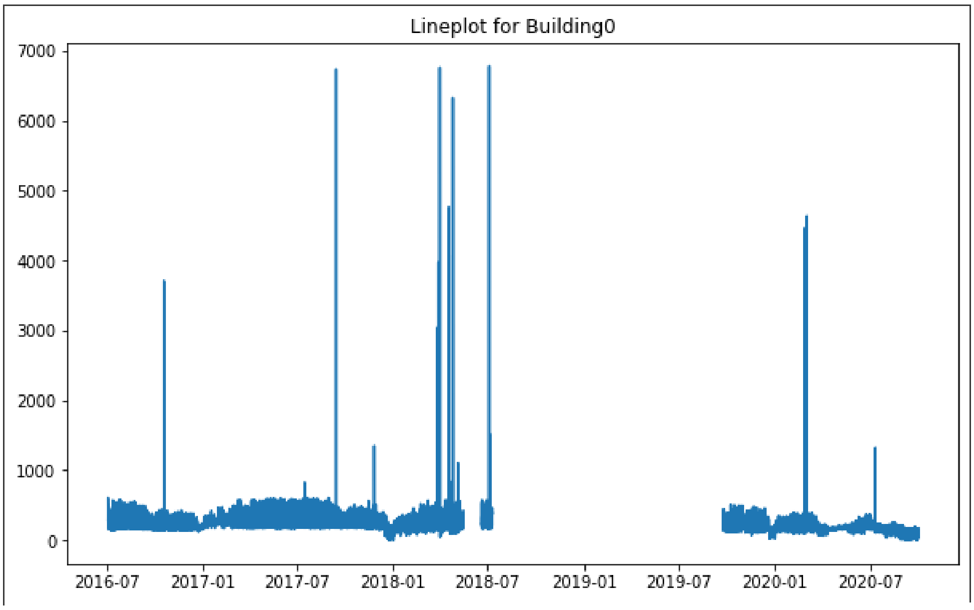


Fig 3: Line plot of building 0

```
Building0    47404
dtype: int64
Building1    88
dtype: int64
Building3    561
dtype: int64
Building4    17661
dtype: int64
Building5    29992
dtype: int64
Building6    2255
dtype: int64
```

Fig 4: How much data is missing for each building

Exploring the amount of data missing of each building, building 0 is missing the most while building 1 is missing the least. For the building modelling, the Nan values will be removed as some models cannot take Nan values. During this process however, it was found that the missing values were not recognised as numpy missing values but as strings in the dataset, and so this required changing the missing values from strings to numpy Nan values.

## Descriptive statistics for solar panel and building observations:

Below is the graph of boxplots from the solar panels:

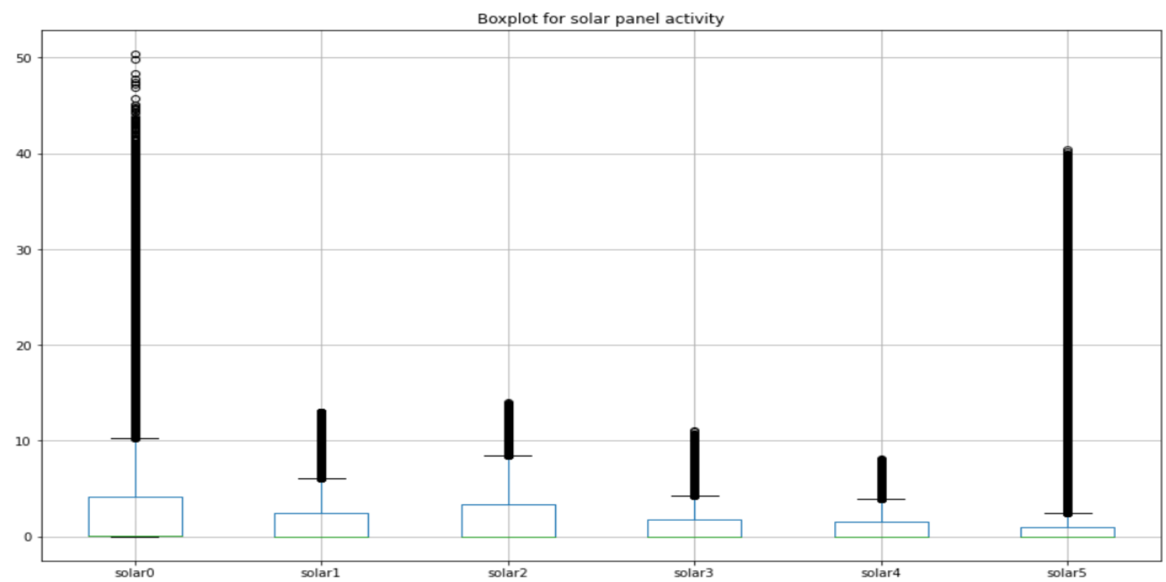


Fig 5: Box plot for solar panels values

From the graph above we can see that the median of buildings is at position 0. This makes sense since solar panels don't always work. Furthermore, there are data points that are separate from the rest and these data can be considered as outliers. Therefore, we need to discard this unhelpful data because it can cause errors in prediction.

Below is the graph of boxplots for the energy comsumption of 6 buildings :

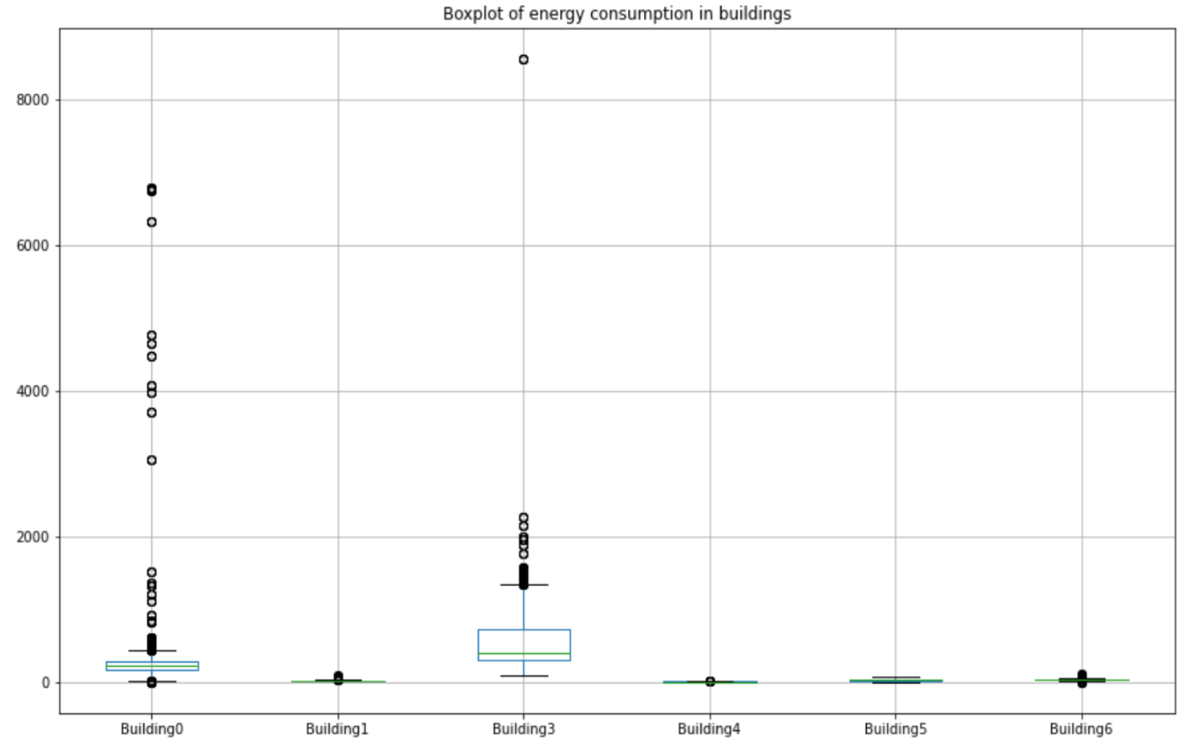


Fig 6: Boxplot for building values

Based on this graph we can see that buildings 1,4,5 and 6 use very little energy and almost no use. Meanwhile, buildings 0 and 3 consume more energy than the remaining buildings. In addition, we also need to remove the outliers that appear in this graph as we did with solar panels.

We also got a heatmap to show the correlation between variables that affects solar energy consumption.

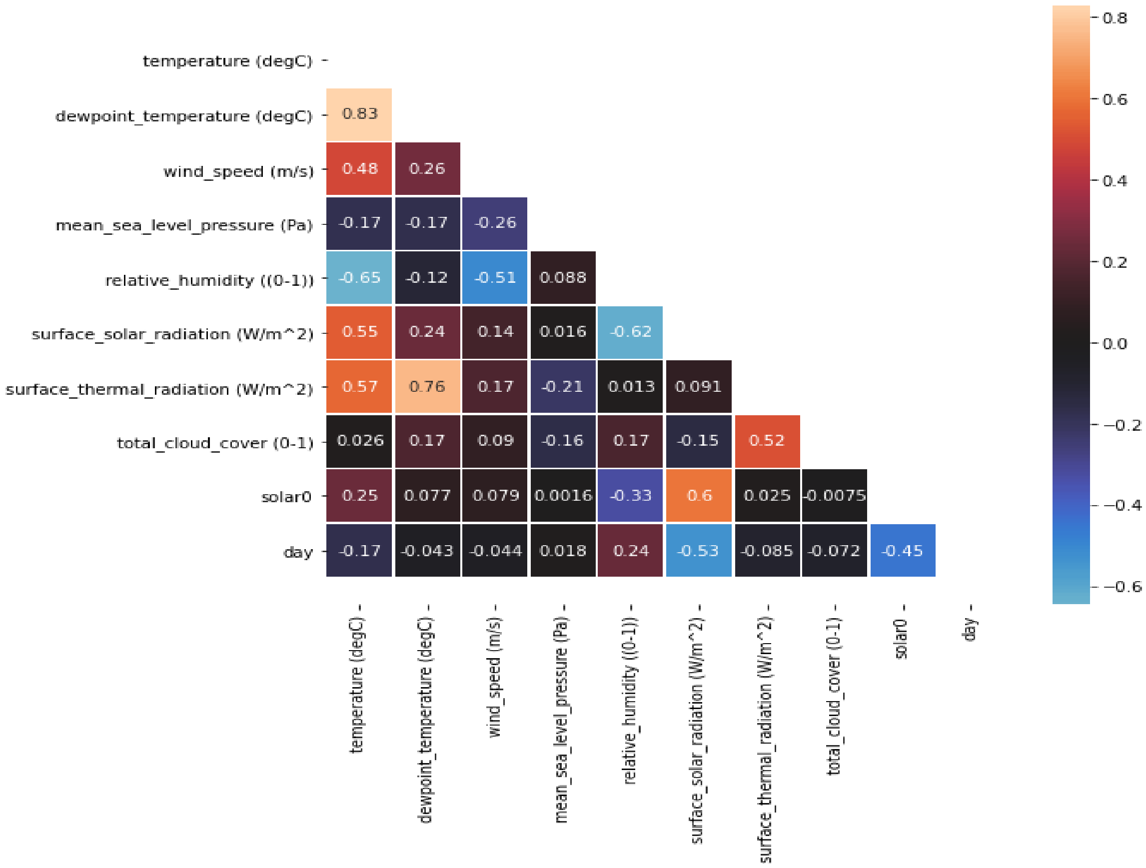


Fig 7: Heatmap of the correlations for variables

From the heatmap above we can see, the maximum value of correlation is 0.83. So the relationship between dewpoint_temperature and temperature is a strong, positive, linear relationship. Therefore, these two quantities have a close connection and have a great influence on the consumption of solar energy. Besides, the lowest value of correlation is 0.0016. This number is quite close to 0 and it indicates no relationship between Solar0 and mean_sea_level_pressure. Therefore, these two quantities do not have much influence on the consumption of solar energy.

## Aggregated cost data

In order to create an optimal battery and lecture schedule, the aggregated cost of electricity is required. The historical electricity price data from October 2020 was retrieved from the Australian Energy Market Operator. The data consisted of the total energy demand and the aggregated price every fifteen minutes for the whole of October (as shown in figure 6).

| Date | SETTLEMENTDATE | TOTALDEMAND | RRP |
|---|---|---|---|
| **2020-10-30 00:30:00** | 30/10/2020 0:30 | 4324.78 | 47.32 |
| **2020-10-30 01:00:00** | 30/10/2020 1:00 | 4222.34 | 43.99 |
| **2020-10-30 01:30:00** | 30/10/2020 1:30 | 4120.86 | 42.96 |
| **2020-10-30 02:00:00** | 30/10/2020 2:00 | 3964.18 | 45.49 |
| **2020-10-30 02:30:00** | 30/10/2020 2:30 | 3842.67 | 41.42 |

Figure 8: Aggregated electricity price data frame

The data frame was then indexed only have the electricity price data for the 30th of October for analysis on what the electricity price on a weekday in October would look like.

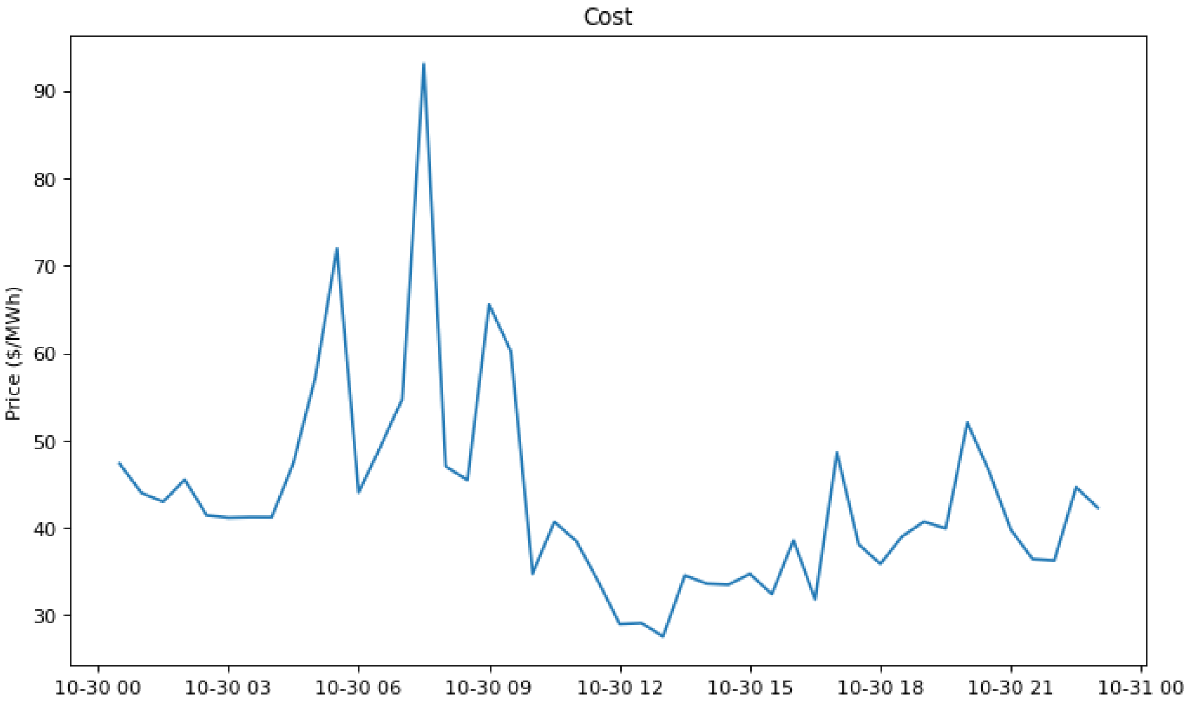


Figure 9: 30th October 2020 electricity price data

In figure 7, it can be seen that the electricity price is at its lowest during the middle of the day. This is expected because the most solar is being produced, which is a cheap source of power, therefore lowering prices. The price is overall higher overnight which is expected as there is no solar production at this time. It can be observed that there is a big spike in price in the morning at 5am, 7am and 9-10am. This is expected to be due to when everyone is waking up so there is a sudden increase in demand for electricity and not enough getting produced, resulting in the cost increasing.

# Model Development

## Energy consumption

To produce an optimal lecture schedule, the energy demand must be modelled and predicted for October 2020. With the buildings data cleaned and organised, we can further look into the energy consumption trends to get a better understanding at how we would predict future values. Figure 10 shows the energy consumption of building 3 since 2016. Due to an extensive amount of data, trends and patterns are not clear, therefore, looking at the energy consumption of September 2020 (see figure 11), there are clear trends and seasonality where a pattern occurs within each day as energy consumption rises during mid day then drops at night. Furthermore, a weekly seasonality can be shown where energy consumption decreases during weekends when compared to weekdays. These trends are consistent during the month of September 2020. Figure 12 gives a closer look at the weekly fluctuations of energy usage.

```
plt.plot(df3)
plt.title("Building 3's energy consumption")
plt.xlabel("Year")
plt.ylabel("Energy consumption (kW)");
```

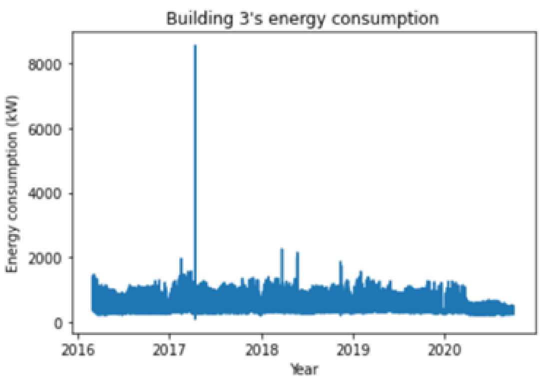


Figure 10: building 3's energy consumption

```
sept2020=df3['2020-09-01 00:00:00':'2020-09-30 23:45:00']
plt.xticks(rotation=45)
plt.title("Building 3's energy consumption Sept 2020")
plt.xlabel("Date")
plt.ylabel("Energy consumption (kW)")
plt.plot(sept2020);
```

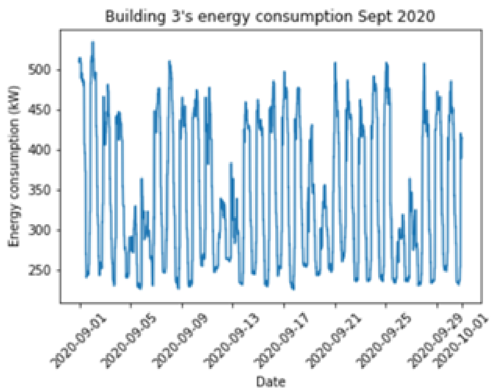


Figure 11: energy consumption September 2020

```
sept2020=df3['2020-09-01 00:00:00':'2020-09-07 23:45:00']
plt.xticks(rotation=45)
plt.title("Building 3's energy consumption first week of Sept")
plt.xlabel("Date")
plt.ylabel("Energy consumption (kW)")
plt.plot(sept2020);
```

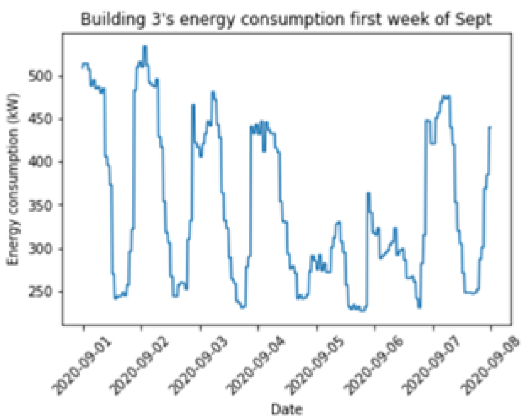


Figure 12: week 1 September 2020 energy consumption

To predict the energy consumption of October, time series forecasting would be needed as the time of day would be the main influence for modelling. SARIMA (Seasonal Auto-Regressive Integrated Moving Average) is a type of time series forecasting model. It is an updated version of ARIMA (Auto-Regressive Integrated Moving Average), where it further includes seasonality when finding a forecast of the time series, which is evident in our data.

Our data was given in 15-minute intervals, which is too extreme for SARIMA to handle, therefore, data was resampled and the daily energy consumption was calculated and this was the data used to model the forecast. Python has built in SARIMAX models that can be imported and will be used (see figure 13).

```
#import SARIMAX
from statsmodels.tsa.statespace.sarimax import SARIMAX
from pmdarima import auto_arima
from statsmodels.tsa.seasonal import seasonal_decompose
```

Figure 13: Importing SARIMA code

The SARIMAX function takes in seven parameters, three trend parameters (p, d, q) and four seasonal parameters (P, D, Q, m). The first trend parameter (p) is the trend autoregression order which predicts future values from past values. Next is the trend difference order (d) which is the number of differencing transformations in order to get a constant mean and variance. Last is the moving average order (q) which is the error order component or moving average terms. The (P, D, Q) parameters are the seasonal trends of (p, d, q) and ‘m’ is the number of periods in a single season.

The best (p, d, q) and (P, D, Q) values can be found using the built-in function ‘auto_arima’ which takes an input of the data frame and the number of seasonal periods (m). As the data was resampled into days, our seasonal period would be 7 for seven days of the week. The best parameters for building 3 were ‘order=(5,1,0)’ and ‘seasonal_order=(1,9,1)’ (see figure 14).

```
auto_arima(df3,m=7, trace=True).summary()
```

```
Best model:  ARIMA(5,1,0)(1,0,1)[7]
Total fit time: 405.658 seconds
```

SARIMAX Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | y | **No. Observations:** | 1671 |
| **Model:** | SARIMAX(5, 1, 0)x(1, 0, [1], 7) | **Log Likelihood** | -9066.688 |
| **Date:** | Thu, 06 Oct 2022 | **AIC** | 18149.377 |
| **Time:** | 21:32:52 | **BIC** | 18192.741 |
| **Sample:** | 0 | **HQIC** | 18165.445 |
| | - 1671 | | |
| **Covariance Type:** | opg | | |

Figure 14: code for auto_arima, gives the best ARIMA parameters

To train the model,the data will be split  into a training and testing set. All data points up until September 2020 were used to test against September 2020 (see figure 15). Then using the built in function ‘SARIMAX’ with the best parameters found in ‘auto_arima’ to predict september 2020 and october 2020.

```
train=df3[:'2020-08']
test=df3['2020-09']
model=SARIMAX(df3, order=(5,1,0), seasonal_order=(1,0,1,7))
res=model.fit()
start=len(train)
end=len(train)+len(test)-1
prediction=res.predict(start,end).rename("Prediction")
```

Figure 15: training and testing split of data, using SARIMAX with retrieved parameters

## Solar Production

Before beginning to model the solar production, two variables were added which were the hour of the day and if it was daytime or not. This was done under the assumption that solar production is cyclical. The data frame was also cut down to when solar production values started, excluding the first 3 years due to there being no data.

```
final = final[123632:160784]
final['hour']=final.index.strftime('%H')
final['day']=final.hour.apply(lambda x: 1 if x >= '07' and x<='18' else 0)
```

Figure 16: Indexing weather data to month of October 2020

The next step was creating six data frames for each solar panel to model each one separately. In order to address the possible missing values for each solar panel, it was decided to remove the 0 values if the period that the 0 values stretched over were greater than a day (97 zero entries in a row). This is to ensure that the model is not skewed and that the 0 values during the night when there is no energy production are not being removed.

In order to model the solar production for October 2020, weather for that month is required. A separate data frame 'test_w', was created which contains the weather for the month of October. This was done by subsetting the weather data using the following code.

```
mask = (test_w['index1'] >= '2020-10-01 00:00:00') & (test_w['index1'] <= '2020-10-31 23:45:00') # month of October
test_w = test_w.loc[mask]
```

Figure 17: Indexing weather data for month of October 2020

The variables for hour and day were also added using the same code used on the final data above.

## Feature selection

Before modelling, the feature importance of the weather variables was looked into to explore the effect of using only the most important variables on the model accuracy. This was done using the ANOVA score from SelectKBest.

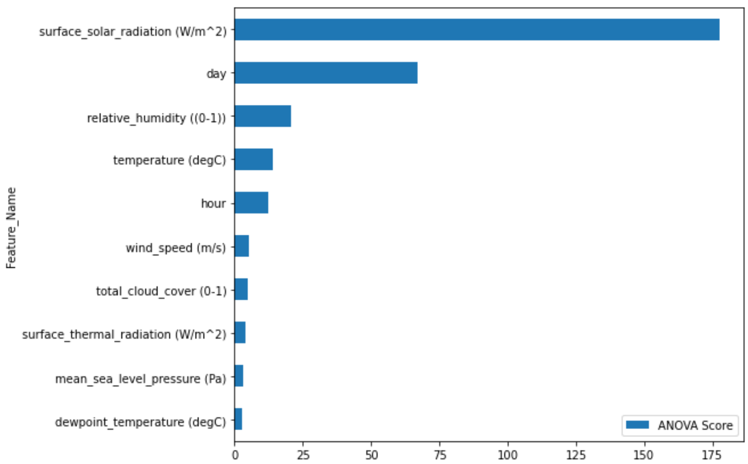


Figure 18: Feature importance

From the bar graph (figure 10) of the ANOVA score, it can be observed that surface solar radiation is by far the most important variable, with weather it is day or night following with just below half the score. All the other variables have a lot less importance, with a more gradual decrease in importance between each one. The model used for the modelling process of the solar production was random forest regression. Therefore, the random forest regression model was used to test what number of most important features gave the best accuracy.

Before modelling for each solar panel, the effect of using different numbers of k best variables was explored using solar4 data to find the optimal number of weather variables to use for the model. To compare the accuracy of each model, we elected to use the adjusted r squared value instead of the r squared value. This is because the r squared value will always increase as a variable is added to the model, where the adjusted r squared score takes into account the number of variables in the model, adjusting the accuracy accordingly. The relative adjusted r squared values were calculated using the following loop.

```
def num_var_used():
    scores = [] # array of all acuracies using k-best features
    for i in range(1,10): # looping through each number of k best features
        features_anova = SelectKBest(k=i).fit_transform(features, label)
        X_train, X_test, y_train, y_test = train_test_split(features_anova, label, test_size=0.2)
        rfr = RandomForestRegressor(n_estimators=100)
        rfr.fit(X_train,y_train)
        y_pred = rfr.predict(X_test)
        R2 = r2_score(y_test, y_pred) # calculating the R^2 value
        scores.append(round(1-(1-R2)*(len(X_train)-1)/(len(X_train)-i-1),3)) # adding adjusted R^2 value to scores
    print(scores)

num_var_used()

[0.645, 0.685, 0.761, 0.803, 0.869, 0.9, 0.923, 0.933, 0.942]
```

Figure 19: Modelling for SelectKBest variables

From the results in figure 11, it can be seen that even using the adjusted r squared, the accuracy of the model increases as each feature is added to the model. The best result was achieved when every weather variable was used. For the future modelling of each solar panel, it was decided to use all features going off the assumption made from the adjusted r squared results.

## Modelling using random forest regression

The solar production and whether data that is being used for the solar production model is continuous. Therefore, the model we began with was random forest regression to model the solar production using the weather data.

The initial modelling was done using all weather predictors and all solar data, with gaps of data that extended over a period of time greater than a day removed. The modelling process was repeated for each solar panel to get a predicted solar production for October 2020 and the accuracies. This was done by looping through each of the solar panel data frames. Each solar panel was split into its own train and test set. This was done because each solar data frame had different periods of missing data, resulting in there being some different dates for each solar panel. The training data set's were then fitted to a random forest regressor model using 100 n_estimators. This was done using the following for loop,

```python
solar_models = []
accuracies = []

panels = [solar0, solar1, solar2, solar3, solar4, solar5]
panel_name = ['solar0', 'solar1', 'solar2', 'solar3', 'solar4', 'solar5']

for i in range(len(panels)):

    panel = panels[i]
    panel_name_ = panel_name[i]

    features = panel.drop([panel_name_], axis=1)
    label = panel[panel_name_]

    X_train, X_test, y_train, y_test = train_test_split(features, label, test_size=0.2, random_state=0)
    rfr = RandomForestRegressor(n_estimators=100, random_state=0)
    rfr.fit(X_train,y_train)
    solar_models.append(rfr.predict(test_w))

    y_pred = rfr.predict(X_test)
    accuracies.append(round(r2_score(y_test, y_pred), 3))
```

storing the accuracies in a list 'accuracies'

# Results

## Energy consumption

The SARIMA model was fairly accurate at depicting the overall trends of energy consumption in September for buildings 1, 3, and 6. Figure 20b, 20c, and 20f shows the graphs of energy consumption of building 0, 1, and 3 respectively for September 2020, as well as the model prediction for September 2020 and October 2020. Looking at the predicted line up against the actual values, it shows the increasing of energy consumption over the weekdays, and decreasing on the weekend. All four weeks of October for building 1 have identical energy consumption, whereas building 3 and 6 have a slight decrease throughout the month.

The model has difficulty finding the seasonality trends for building 0 (see figure 20a) and building 4 (see figure 20d) as evident by the major differences with the blue and orange line. This could be due to weak seasonalities and patterns, proving that the energy consumption for the particular building is inconsistent with time. However, looking closer at building 4, the energy consumption throughout September sits at 1.3kW which is significantly lower than other buildings which can show that that energy consumption is little, thus October 2020 building 4 prediction is constantly 1.3kW.

Building 5 has missing days in its data which is a problem for SARIMA modelling due to its sensitivity to missing values. Therefore, the indexing in training and testing within SARIMAX did not match up, and October 2020 was not predicted (see figure 20e). Perhaps for future references, further data manipulation could be done by interpolating the data to fill in the missing values, allowing for building 5's October 2020 prediction to be made.

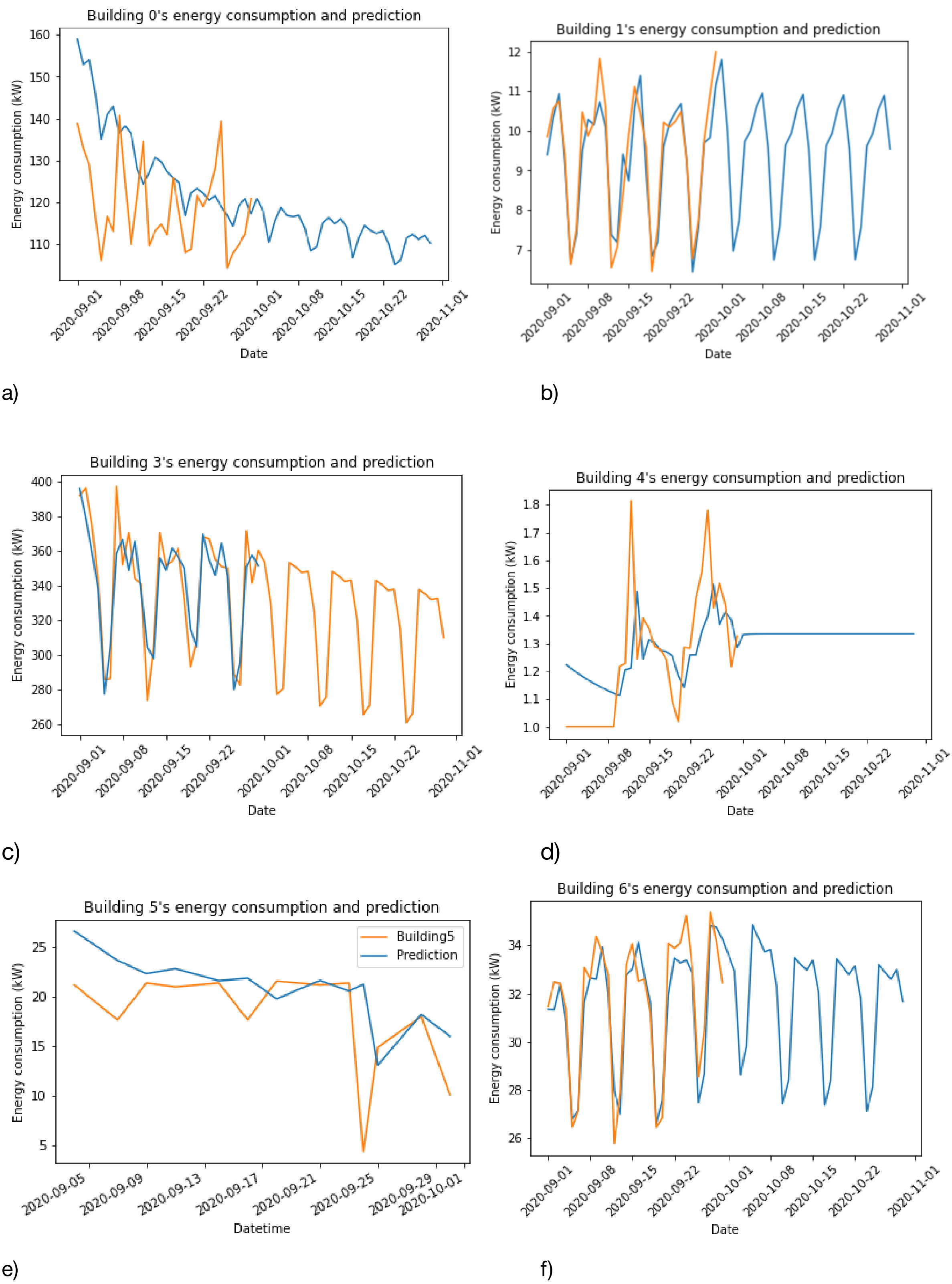


Figure 20: Building energy prediction and consumption of September and October 2020

## Solar Production

Once the solar panel data frames were fitted to the random forest regression model, the following accuracies were given;

```
accuracies
```

```
[0.91, 0.912, 0.912, 0.87, 0.912, 0.925]
```

Figure 21: Accuracies for initial model

From figure 13, it can be seen from the accuracy results that the models for each panel gave very good results, with all except one panel giving an accuracy of above 90%. This suggests that the random forest regression model is able to give an accurate prediction of the solar production

Upon further investigation, it is found that solar0 was missing data up to the big gap in solar production data. This prompted me to investigate modelling the solar production using solar data from after the big gap in missing data for each solar panel. The date at which the data resumed was the 21st May 2020. It was decided to model using data from when the solar data for solar0 began. This was done using the same process as above, subsetting the data to begin at the 21st May 2020 onwards.

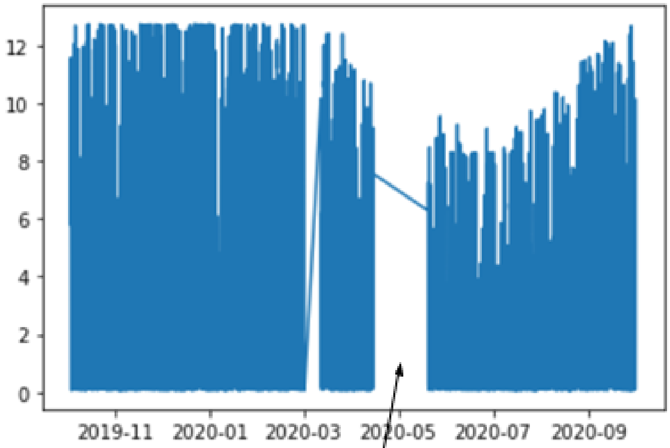


Figure 22: Big gap in solar data



The new data frames were then modelled using the same process used before. This gave the following accuracies.

```
accuracies
```

```
[0.915, 0.914, 0.927, 0.926, 0.918, 0.904]
```

Figure 23: Model accuracies using data after big gap of missing data

From figure 15,it can be seen that the accuracies of the models for each panel using the solar production data after the big gap of missing data were better for every panel except solar5, with all accuracies being above 90%. This means that the above 90% of the variation in solar production is explained by the predictors used in the model. It was concluded that the best model for the solar data is derived when using the solar production data from after the big gap of missing data. This was made under the assumption that the data after the big gap in missing data is the most complete, with minimal missing data and no big gaps of missing data.

The solar production for the month of October 2020 was then modelled for each solar panel using the models trained off the solar production data after the 21st May 2020. This gave the following prediction of solar production for the month of October 2020.

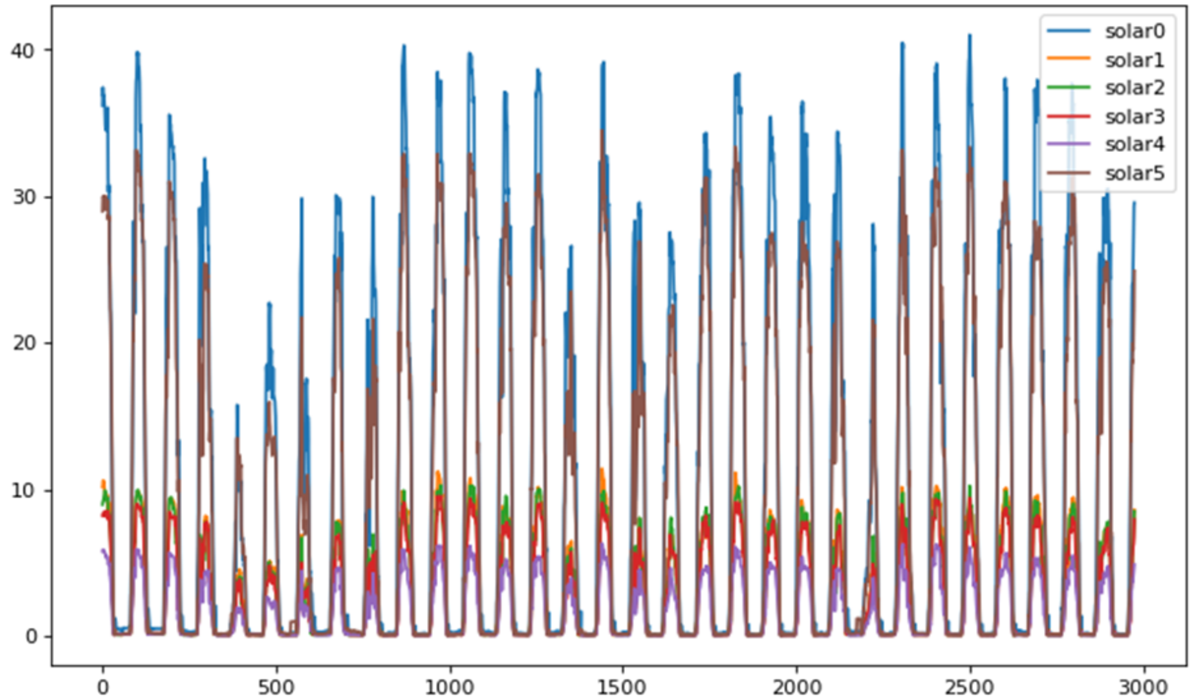

Figure 24: Estimated solar production for month of October 2020

From figure 16, it can be seen that there is no solar production predicted overnight, with the amount of solar production increasing in the morning, hitting its peak in the middle of the day, then decreasing back to zero at the end of the day. The maximum solar production during the middle of the day is expected as the middle of the day is when the sun is at its highest point and there is no sunlight during night, explaining why there is no solar production overnight.

It can be seen in figure 16 that there are a few predictions of minimal solar production during the night. This could be due to some inaccuracies in the solar production data, resulting in the occasional solar reading during the night, having an effect on the models predictions. This could also be due to the use of approximate estimations of times for night and day, resulting in the early predictions of solar production before sunrise and sunset.

Overall, the solar production model for the month of October using random forest regression gives an accurate estimate of the solar production in October 2020. Some further ways to improve the model could be to explore the adding more seasonal features like seasons and the optimisation of when it is daytime and night time. Some more measurements like RMSE could have been used to help find the optimal model.

## Schedule

### Lecture timetable

For the lecture timetable, it would be most optimal to spread the lectures out across the weekdays of the week during the middle of the day. This is because electricity is the cheapest at this time, therefore it is the most cost efficient to run them at this time. The most optimal time to run the majority of the lectures is from 12-1pm (as per figure 7). However, not all lectures can fit into this time. The time 1-3pm is the next cheapest time period (as per figure 7), so any other lectures would be optimally run during those times.

In order to further optimise the lecture schedule, investigation into the building consumption and how to optimally spread out the consumption in an cost effective way could help to reduce costs. Creating a model using the future consumption predictions, solar production and the energy price would give a more precise lecture schedule.

### Battery Schedule

On initial investigation, there was found to be only two batteries, one for building one and one for building three. The battery for building one has an efficiency of 0.85 and the battery for building three has an efficiency of 0.6. From the efficiencies, it can be concluded that energy stored during the middle of the day and used over the night and during the peaks in energy price would be cost effective.

When analysing the energy consumption of the buildings at Monash, it can be seen that the buildings are still consuming the energy at night. This would make the batteries incredibly useful during these times when power is more expensive, as seen in figure 7. If the batteries were charged during the day on weekdays, it is possible for the stored electricity to be used during the night to subside the cost of electricity.

The optimal battery schedule would be to charge the batteries during the middle of the day when energy is at its cheapest and the solar production is at its highest, therefore it would be cheapest to do so. Then the stored energy would be optimally used either overnight or in the morning when energy prices peak. The optimal time could be discovered based on the amount the battery is able to store and how quickly it can output its energy. Further investigation would need to be done to determine these times, through models or further visual analysis. An optimal schedule has not been determined for the weekend. Investigation into the energy price on the weekend could be done to create an optimal schedule for the weekend.

## Conclusion

While the initial goal of this project was to develop a timetable of the month of October 2020 energy consumption in 15 minute intervals, this was not possible using SARIMA so instead the predictions are measured on a daily time frame due to the model being computationally expensive.

While we were able to draw insight at when lectures and events should be held to minimise energy cost, an algorithm was not implemented to create an optimal timetable.

To optimise cost savings for lectures, we should extend classes to weekdays and the appropriate time is midday. Specifically, 12pm-1pm is the cheapest and if it is difficult to fit all the classes at this time, opening classes from 1-3pm is also a next saving option.

To save fees on recharging the battery, we need to have a reasonable schedule to charge the battery. Based on the above analysis, we can conclude that the right time to charge the battery is in the middle of the day because the energy at this time is cheapest, we can save a fee on energy usage. In addition, in the middle of the day, solar production is the most common so we can also take advantage of this time to get a lot of solar energy. Then, we use the charged energy for the battery to use in the morning and overnight when the energy price is quite expensive.