CAPSTONE PROJECT BY TEAM 2

A DATA SCIENCE APPROACH TO MAXIMISE THE PROFITABILITY OF A BATTERY IN THE ENERGY MARKET

Armaan Ravi Eshwar (z5180316), Ahsan Virani (z5188798), Justin Buhagiar (z5208846), Gonzalo Enrique (z5206518), William Thackeray (z5264709)

School of Mathematics and Statistics
UNSW Sydney

November 2020

Submitted in partial fulfilment of the requirements of the capstone course DATA3001

Plagiarism statement

I declare that this thesis is my own work, except where acknowledged, and has not been submitted for academic credit elsewhere.

I acknowledge that the assessor of this thesis may, for the purpose of assessing it:

- Reproduce it and provide a copy to another member of the University; and/or,
- Communicate a copy of it to a plagiarism checking service (which may then retain a copy of it on its database for the purpose of future plagiarism checking).

I certify that I have read and understood the University Rules in respect of Student Academic Misconduct, and am aware of any potential plagiarism penalties which may apply.

By signing this declaration I am agreeing to the statements and conditions above.

Signed: Armaan Ravi Eshwar Date: 21/11/2020

Signed: Gonzalo Ovalles Date: 21/11/2020

Signed: Ahsan Muhammad Date: 21/11/2020

Signed: Justin Buhagiar Date: 21/11/2020

Signed: William Thackeray Date:21/11/2020

Acknowledgements

Thank you to UNSW Associate Professor Pierre Lafaye de Micheaux and Senior Lecturer Michele De Nadai for their support.

Thankyou to Oliver Nunn, Senior Economist at the Australian Energy Market Commission and to Jack Simpson for their guidance.

21/11/2020.

Abstract

The Australian Energy Market Commission (AEMC) has recently implemented the use of new battery storage systems as a contribution to the movement towards renewable energy. Having a relationship between batteries and other sources of renewable energy generation, predominantly solar and wind, allows energy to be discharged at times of peak high demand and stored during times of low demand. Predictions of energy demand and energy market prices are needed to maximise the profits of using such a system. Profits are defined as the revenue obtained from discharging energy in moments of high energy demand and the costs of charging the battery when there is low demand. The aim of this report is to produce a forecasting model that generates accurate predictions of energy prices in the South Australian region which can then be linearly optimised to maximise the profit of a battery over a 6-month period. The results of the optimisation model are promising as 44.54% of the maximum attainable profit available in the market was achieved by the model.

Contents

Chapter 1
Introduction

The development and use of batteries as a renewable energy source is of increasing interest internationally and domestically. Providing another source of renewable energy alongside wind and solar power can lead to decreasing costs and substantial profits. The Australian Energy Market Commission (AEMC) has recently been implementing this new type of battery storage technology. Combining this new technology with already established renewable energy sources such as wind and solar allows energy to be discharged at times of peak high demand and stored during times of low demand instantly and efficiently. Due to the energy prices being so volatile as a result of biddings, demand and supply of the market and energy costs it is beneficial to have such a diversified type's of technology that can dispatch energy on command. The aim of this report is to produce a predictive model that forecasts the energy prices over a period of 6 months which can then be used to maximise the profit to be made under the constraints of battery technology. The predictive model refined in this report is categorised as type of multilayered neural networking model. Our model uses South Australian data from January to June of 2019 for model testing. Linear optimisation is used to maximise the profit. The reason we chose the South Australian region specifically was because our contact at the AEMC mentioned it would be the most interesting region to predict due to the greater use of renewable energy in that state.

Chapter 2

Literature Review

In his report into harnessing big data and data science across the energy sector Maurice R. Greenberg explored data growth within the energy sector, its applications and its uses in respect to data science techniques. He suggests the potential, latent within these large and diversified data sets, to create more efficient predictions and automation processes. The article suggests that "machine learning approaches can use historical wind and solar data to improve forecasts" as well as many other optimisation problems within mechanical infrastructure. [1] Professor George Yarrow, in his 2014 study into energy-only wholesale electricity markets, examines the nature of pricing in industry. He surveys the inelasticity of demand and stability of supply conditions. It can be deduced, from his study, that if demand is relatively insensitive to price, making a prediction about energy usage within a time series should be based predominantly on seasonality and human activity. [2]

An article written by students at Griffith University under the supervision of Hui Li and Zilong Yang for the optimisation of energy storage within microgrids, taking into account renewable sources such as solar and wind, helps contextualise a practical application using data assets in the energy sector . It theorises charge, discharge and critical load scenarios to help define algorithms for aiding battery size selection. The study endorses an economic based approach to the battery sizing problem in order to gain "the maximum benefit at minimum cost by the means of peak saving and energy saving techniques during peak times". [3] Alexander Kies, in his examination of battery life degradation in the electric vehicle industry, highlights the importance of finding an optimal time to charge and recharge, given particular cycles of traffic, battery specifications and run life. Following his examination, he concludes that appropriate battery usage aids a "reduction of approximately 8% in overall cost". [4] LianLian Jiang, and Guoqiang Hu, distinguished researchers of the Singapore School of Electrical and Electronic Engineering and Nanyang Technological University, Singapore, respectively, attempt to forecast electricity prices of the Australian market in the Victoria (VIC) region and Singapore market. They propose a "Long-Short term" recurrent neural network model. It is characterised by its "ability to bridge long time lags of inputs and remembering the historical trend information in time series". The methodology is meticulously refined and comparatively "out performs four popular forecasting methods and provides up to 47.3 percent improvement in the average daily MAPE for the VIC market" [5]

Chapter 3
Material and Methods

3.1 Software

Initial regression analysis was performed using R. Data cleaning, transformations and the deep learning model was run using Python and Gurobi, a python based software which performed the optimisation.

3.2 Description of the Data

Historic data from 2009 to 2019 was provided by the AEMC administration and extracted from SQL servers in CSV format. The data sets contained 50 columns of data points and more than 500'000 rows. The pricing data originates from states Victoria, Tasmania, New South Wales and South Austrlia. Further data was provided including solar and wind for further inputs in the deep learning model.

3.3 Pre-processing Steps

The data included rigorous descriptive variables in conjunction to the bidding procedures and most of that data was discarded. Price, demand and time were the primary variables of interest.

3.4 Data Cleaning

Main variables were extracted using Python. Month, day, and time were separated to create additional seasonal predictor variables. A function was created to replace any price outliers that are significantly over or under the upper and lower quantiles with 75% and 25% quartile values respectively.

3.5 Assumptions

The data provided was assumed to be accurate and reliable in accordance with the standards at AEMC and within the framework of open source, non confidential data.

3.6 Modelling Methods

We utilised a neural network framework to predict our prices. We ended up with 2 different variations of our model, with a final model being decided by experimenting with differ hyper-paramter values.

Chapter 4

Exploratory Data Analysis

The initial datasets provided from the AEMC were csv files containing the following variables:

SETTLEMENT DATE: The current date and time period.

TOTAL DEMAND: The demand for energy

RRP5MIN: The price of electricity for the 5 minute period

WIND: Renewable energy provided by wind turbines

SOLAR: Renewable energy provided by solar panels

Demand and renewable energy features were combined into one new variable labeled RESIDUAL DEMAND.

The following equation is used in this process:

RESIDUAL DEMAND = TOTAL DEMAND - WIND - SOLAR

RESIDUAL DEMAND encapsulates the demand that has not been met by renewable energy sources. This feature was included as demand purely for electricity. As the data is in the form of a time series, it is important that the features are stationary. Generally, models assume independence between points. For this to occur, the mean has to be roughly constant and the variance low.

Figure 4.1. Prices over February 2018

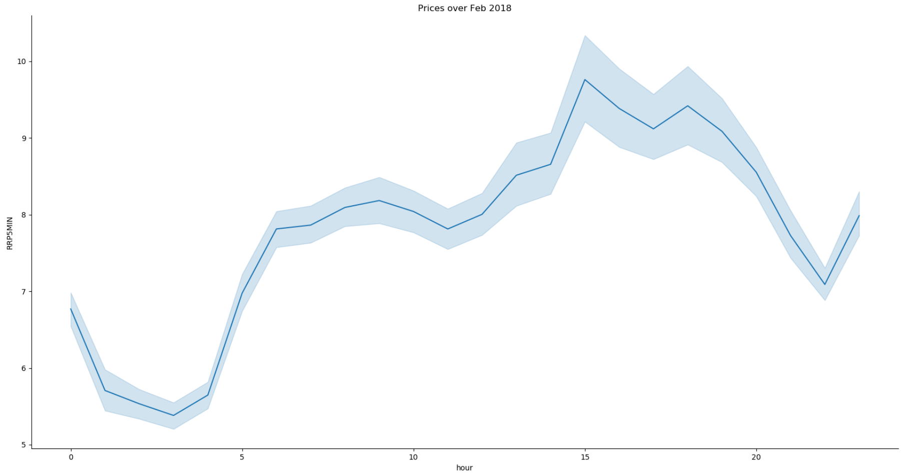


Figure 4.1 depicts the price variation on average in February of 2018. The curve explains the mean price for each hour of the day and shaded regions indicate the 95 percent confidence interval on the hourly price. It is sufficient to state there is an unstable mean and heteroscedastic variance. Due to the eratic nature of pricing in the energy market it is important that we create a feature that reflects the trend in price in a stationary manner.

Figure 4.1.2 Time series Output

```
THE SERIES IS STATIONARY
------------------
  Test Statistics
------------------
ADF Statistic: -24.380617
p-value: 0.000000
Critical Values:
        1%: -3.430
        5%: -2.862
        10%: -2.567
THE SERIES IS STATIONARY
------------------
  Test Statistics
------------------
ADF Statistic: -18.665321
p-value: 0.000000
Critical Values:
        1%: -3.430
        5%: -2.862
        10%: -2.567
```

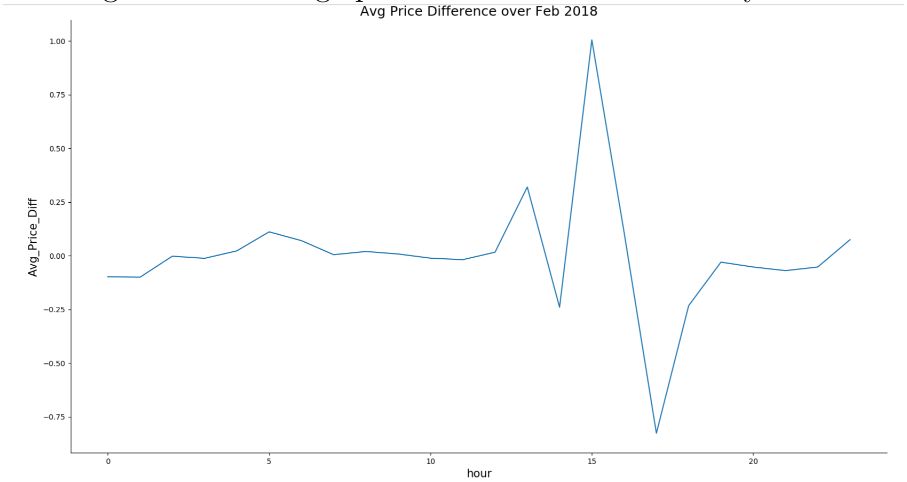
An Augmented Dickey Fuller test was conducted to test the stationarity of our price time series. The null hypothesis states that there is a unit root i.e. the series is non-stationary, whereas the alternate hypothesis states that the series is stationary. The p-value below 0.05 gives evidence against the null, hence we conclude that the series is indeed stationary. However, looking at the plot of price over a shorter time frame indicates non-stationarity as the mean is non-constant and the variance exhibits heteroskedasticity. One advantage of using LSTM is that it is not very sensitive to the unit root issue. Hence, we will first-difference the series for safety.

The difference transform formula below is applied:

$$\Delta y(t) = y(t) - y(t-1)$$

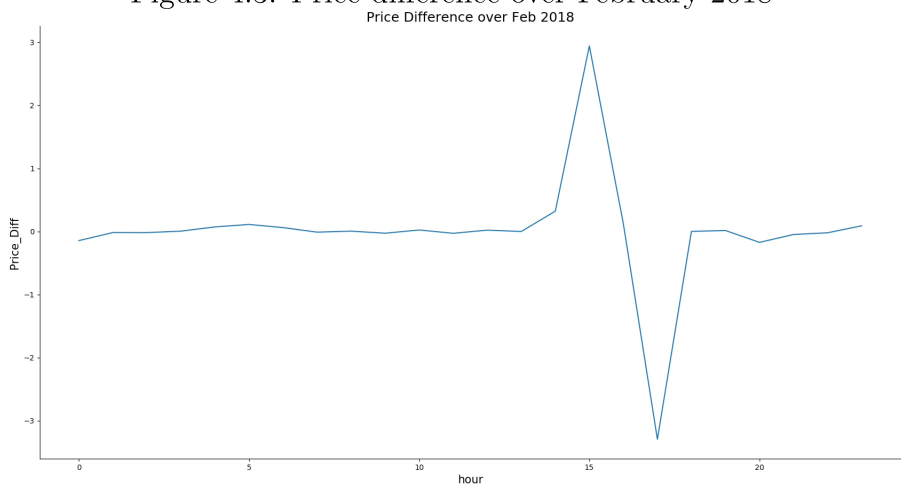
This method obtains the difference between two successive time periods in order to stabilize the mean and eliminate trend and seasonality. Two variables were created that employ this transformation. In one, the average price for a given hour is obtained, followed by the differences in those average prices. Figure 4.2 displays these average price differences over the time period of February 2018.

Figure 4.2. Average price difference over February 2018



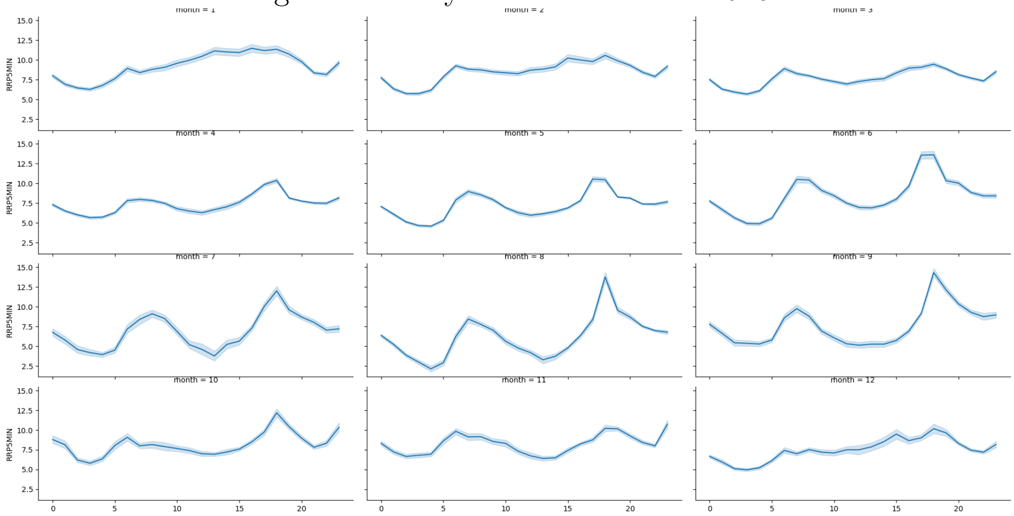
The other variable is simply the difference in price for every 5 minute interval, which can be seen in Figure 4.3.

Figure 4.3. Price difference over February 2018



It is interesting to note that while the mean and variance seem to be relatively constant, there is a massive peak from 3-5pm seen in both Figure 4.2 and 4.3. Due to the nature of how prices are decided every 5 minutes via a bidding system, this can lead to exponential price increases or decreases in a short amount of time, even when transformations have been implemented to make the data stationary. This can be seen throughout the year. Figure 4.4 below shows the trend of prices in day for each month of the year 2018.

Figure 4.4. Day Trend in months 2018



As can be seen from Figure 4.4, while in the first and last three months there seems to be relative stability in price, with a few peaks. However, these peaks are much more pronounced from the months April to September. This could indicate that during the colder periods of the year price of energy increases compared to summer. A common trend appears throughout the year, namely that during the hours from 5am-10am and 3pm-6pm there are large spikes in price. A possibility for this is due to the increased demand in energy during these time periods.
The relationship between the change in average price and the residual demand is displayed in Figure 4.5.

Figure 4.5. Price Vs Demand March

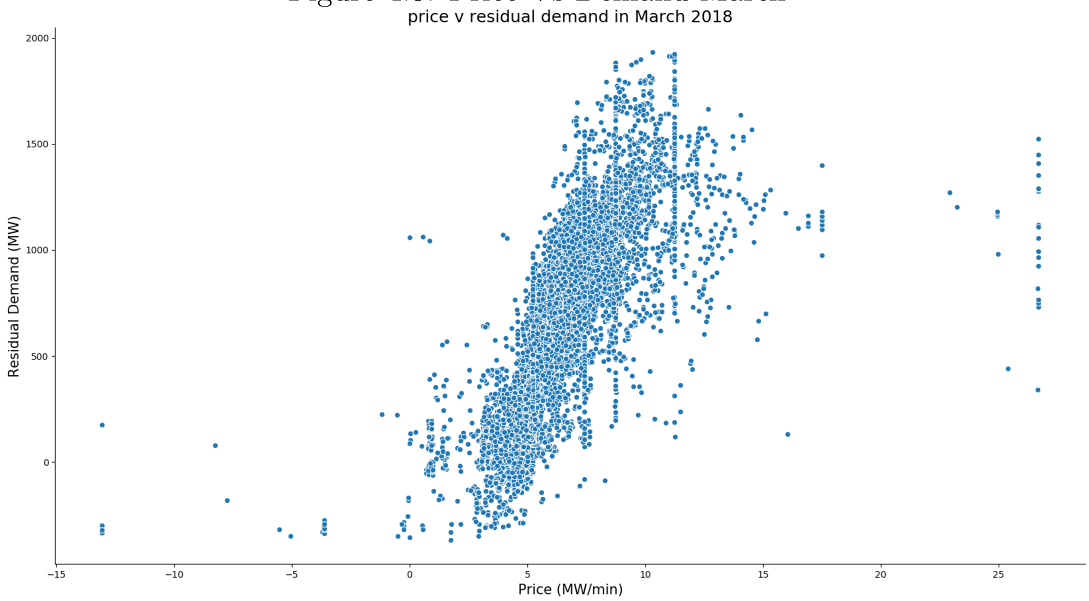


Figure 4.5 suggests that an increase in average price increases residual demand. This indicates a strong positive correlation between residual demand and the change in price of electricity.

Some other patterns discovered were the variations of electricity price according to public holidays, weekends and business hours. The following visualisations help to illustrate these relationships.

Figure 4.6. Distribution Of Prices

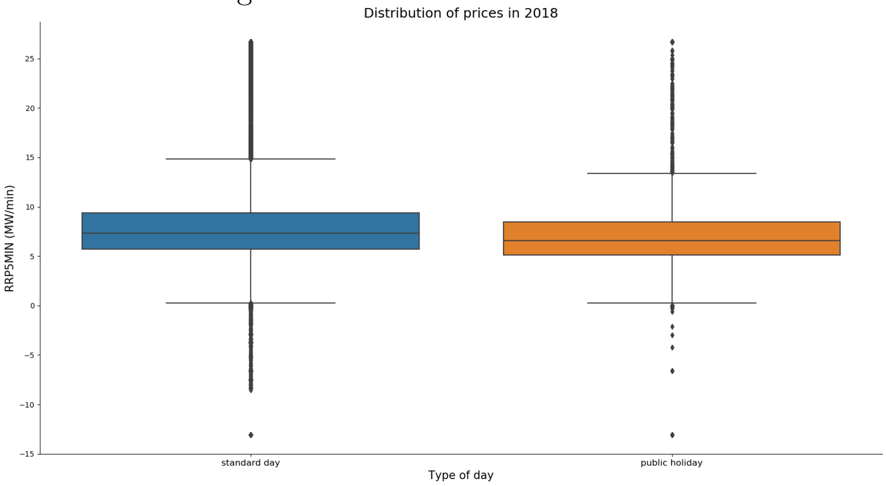


Figure 4.6 shows a distribution of prices every 5 minutes on both standard days and public holidays. Due to the way prices are decided, there are a large amount of outliers. However it can be deduced that the median price is higher on standard days than public holidays.

Figure 4.7. Distribution Of Prices

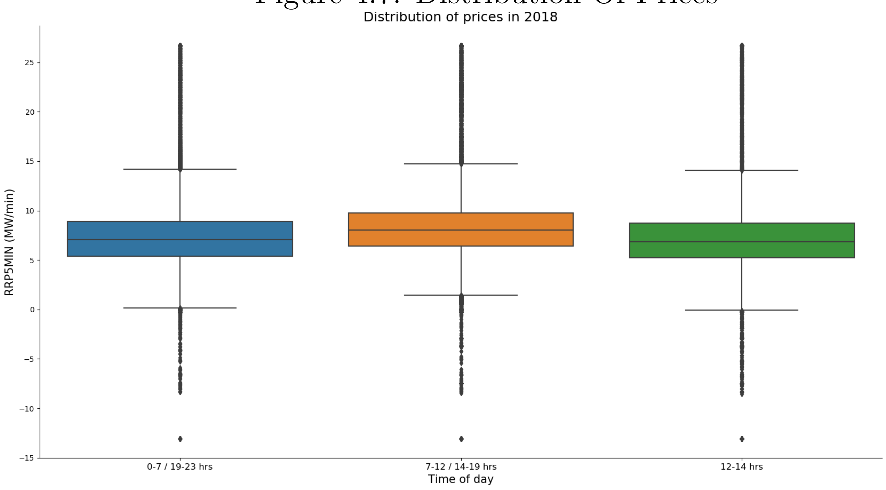
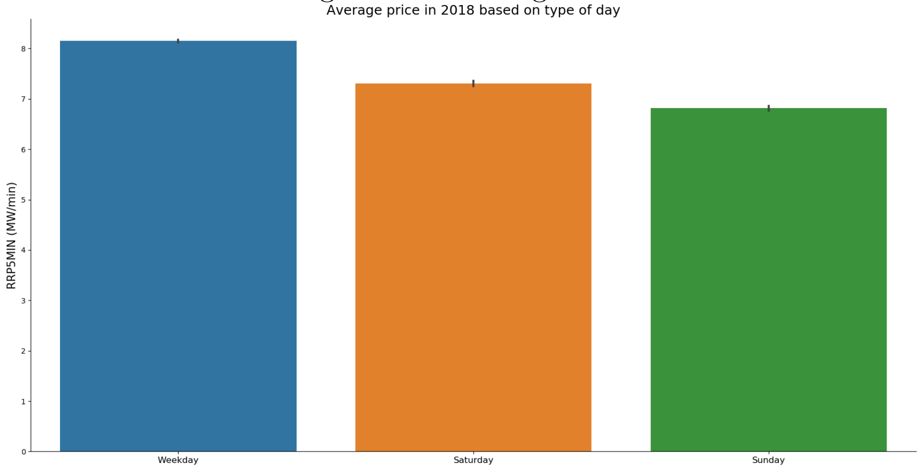


Figure 4.7 displays the distribution of prices based on the time of day. Again it can be seen that a definite increase in prices during the time periods of 7 am – 12 pm and 2pm – 7 pm occurs. This coincides with the price peaks seen in other figures.

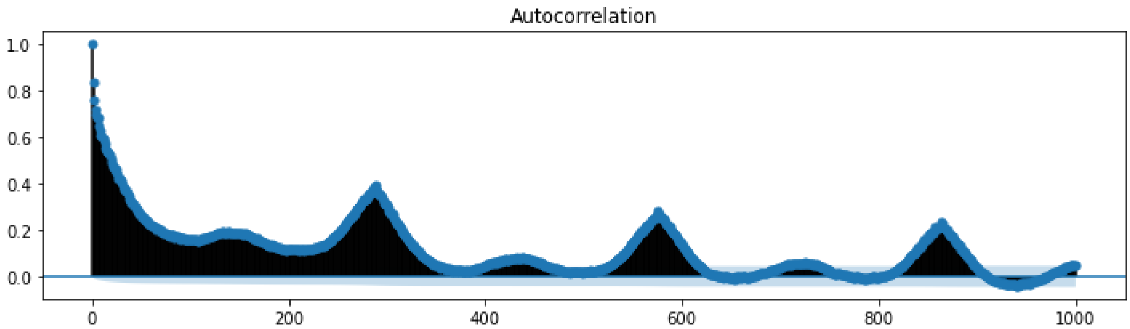
Figure 4.8. Average Price 2018



The mean price of electricity over every 5 minute period, based on weekdays and weekends, is seen in Figure 4.8. While weekdays have the higher prices for electricity when compared to weekends, it must be noted that Saturdays seem to have a slightly higher price than Sundays. These findings indicate that Saturdays and Sundays should be separately categorized.

Another area to explore is the autocorrelation of price. If high autocorrelation was discovered, it would be appropriate to incorporate previous time lags (each lag being 5 minutes) as features in the model. Figure 4.9 depicts the autocorrelation of prices in 2018 with 1000 lags.

Figure 4.9. AutoCorrelation



The first 200 lags prior to a current price point has a very high correlation, as seen in Figure 4.9. In addition, this is also the case for lags ranging from 255-325. These lag periods indicate that they would be useful features to include in our model, in order to predict the current price.

Chapter 5
Analysis and Results

5.1 Model Selection

Electricity markets are extremely volatile, exhibiting high seasonality, fluctuations and spikes, which makes it quite difficult to predict prices accurately. Our battery's performance depends on how well we forecast the prices, thus the algorithm must be able to model these factors. In literature, we can use the following methods for electricity price forecasting:

1. Simulation models
2. Multi-agent models
3. Statistical models
4. Computational intelligence models
5. Deep learning methods
6. Hybrid intelligent models

Simulation models require detailed system operation parameters and have high computational complexity. Multi-agent models, like game theory, are able to model the strategies of the market participants but focus more on qualitative issues rather than quantitative results. Statistical models include similar day methods: exponential smoothing methods (ESM), general additive models (GAM), regime-switching models (RSM) and jump-diffusion models (JDM).

The similar-day method is simple, but has low accuracy when large variations exist in price time series. ESMs perform better than the normal moving average method but it is difficult to determine the parameter of smoothing factors. GAM is a flexible nonlinear model and provides better estimation accuracy than conventional linear models. However, it works best for trends that are steady and systematic.

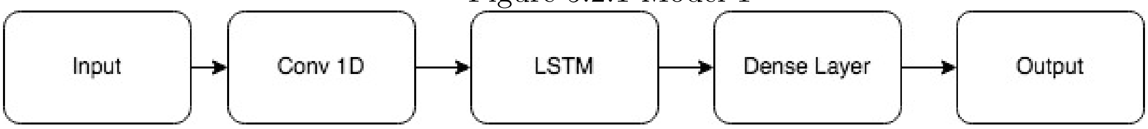
RSMs and JDMs have the ability to take into account the asymmetry of the time series and even can simulate large spikes although require accurate model parameters which are usually hard to acquire. Time series based models have been widely used to predict the electricity price, such as autoregressive integrated moving average (ARIMA) and generalized autoregressive conditional heteroskedastic (GARCH) but are problematic when rapid variations and high frequency changes in the prices occur.

Traditional neural networks have gradient disappearance and gradient explosion problems in practical applications. We propose an improved Long Short Term Memory (LSTM) neural network, which can solve the long-distance dependence problem that RNN's can't handle, thereby making full use of historical information, and having stronger adaptability in time series data analysis. Due to the sequential nature of our data and how the price at a given moment is dependent on the information from previous time periods, it is important to have a model that incorporates past dependencies. An LSTM does this by retaining key information from prior points in the sequence, which will help our model make more accurate price predictions.

To further improve forecasting results, we propose a two-way neural network. The first neural network would be a multi-layer perceptron to model daily trend and seasonality using categorical features such as hour of the day, day of the week, month of the sample, public holiday and business hours. The second network would be a Convolution (1D)-LSTM combination, making use of 335 lags of differenced average price, differenced price, and residual demand. CNN is a type of feed-forward artificial neural network, which works very well to identify simple patterns with the data. A 1D CNN in particular is very effective to extract significant features from shorter segments of the overall data and where the location of the feature within the segment is not of high relevance. Both these NNs would be concatenated to map to 288 intervals forecast i.e. one day ahead.

5.2 Model 1, The Convolution 1D-LSTM combination

Figure 5.2.1 Model 1



The first model is a hybrid model, combining Convolution 1D layer with LSTM, which maps to a fully connected output dense layer with 288 intervals to predict - A day ahead forecast at 5 minutes granularity. We begin by passing our inputs, 250 lags of price and residual demand, to the 1D Convolution layer, which creates a convolution kernel that passes over a single spatial (or temporal) dimension to produce a tensor of outputs. This model is capable of capturing both the short term and long term patterns in the data.

In the Convolution 1D layer, the kernel size parameter specifies the size of the convolutional window, the strides parameter specifies the shift size of the convolution window, and the padding parameter is useful when modeling temporal data. By setting kernel size and strides to 335, we cover all the serially correlated historic data in one convolution step. Padding of valid means the input is not zero-padded, so the output of the convolution will be smaller than the dimensions of the original image, thus avoiding overfit by ignoring the noise in the data. The activation function is set to sigmoid and number of neurons to 128 after trial and error.

An LSTM layer has three gates: an input gate which determines whether or not to let the new input in, a forget gate which deletes information that is not important and an output gate which decides what information to output. This architecture gives our LSTM model the ability to maintain previous information. The output of our LSTM depends not only on the current inputs, but also on inputs received in previous sequence entries. We will make use of the lags of the data to infer a long term trend to predict the day ahead electricity price. It makes sense because we established high serial correlation with the historic prices in the data analysis section. The activation function is set to sigmoid and number of neurons to 64 after trial and error.

Training Model 1:

Years 2017 and 2018 are used for training and we will test our predictions on 2019 data. With the above inputs, the input data to the model is set to (729, 250, 2), which represents samples, number of time steps, number of input features). The fully connected output dense layer will predict prices for 288 intervals of 5 minutes granularity, thus having a dimension of (729, 288), which means (number of samples, number of output features).

Compiling the Model:

We chose 'MSE' as our loss function, which calculated as the average of the squared differences between the predicted and actual values. Taking 'MSE' as our loss function means that the model will get penalised more for making larger mistakes. Because of this large penalty, our model might end up picking up on noise and ignoring the major patterns, thus causing overfit. Therefore, we will also test mean absolute error (MAE) as a loss function when we fine tune hyper parameters.

For the optimizer, we will use the 'Adam' optimization algorithm, which can be used instead of the classical stochastic gradient descent procedure to update network weights iteratively based on training data. The method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters. Finally, we will pick batch size to be 32 and train over 160 epochs. The validation loss graph on training epochs converges to a constant loss as shown in model performance in figure 5.2.3, hence stopping at 160 epochs will avoid overtrain.

Figure 5.2.2 Model 1 Summary

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 3, 64)             36928
_________________________________________________________________
lstm_1 (LSTM)                (None, 3, 50)             23000
_________________________________________________________________
dropout_1 (Dropout)          (None, 3, 50)             0
_________________________________________________________________
lstm_2 (LSTM)                (None, 50)                20200
_________________________________________________________________
dropout_2 (Dropout)          (None, 50)                0
_________________________________________________________________
dense_1 (Dense)              (None, 288)               14688
=================================================================
Total params: 94,816
Trainable params: 94,816
Non-trainable params: 0
_________________________________________________________________
```

Figure 5.2.3. Training and validation error

Figure 5.2.4 Model 1 Results over 1 week

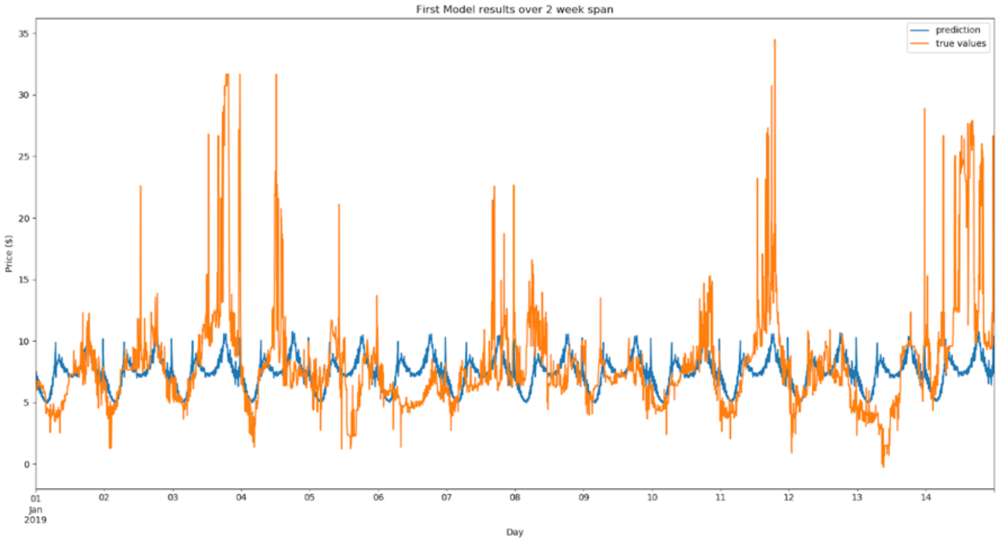
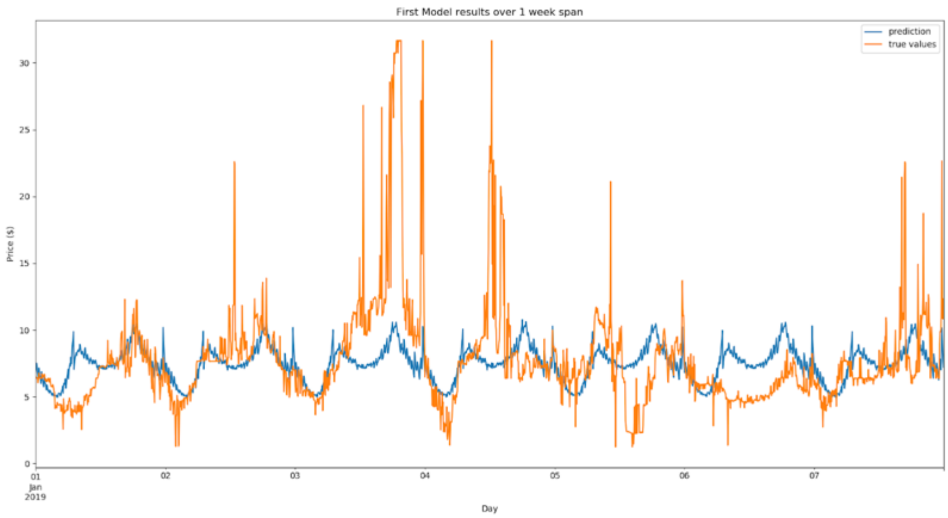


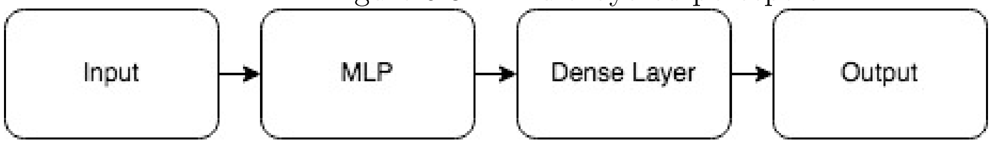
Figure 5.2.5 Model 1 Results over 2 weeks



From Figures 5.2.4 and 5.2.5 we can see that our predictions capture the general trend of prices in the electricity market. However during the early hours of the day there appears to be a mismatch between the trend of the actual price and predicted price. The model also does not capture the price trend at 12pm as there is a massive spike in prices. Unfortunately due to the erratic nature of prices in the electricity market, it is difficult to accurately capture these spikes in price as our model will generally view these as outliers.

5.3 Model 2: MLP + Model 1

Figure 5.3.1. Multilayered perceptron

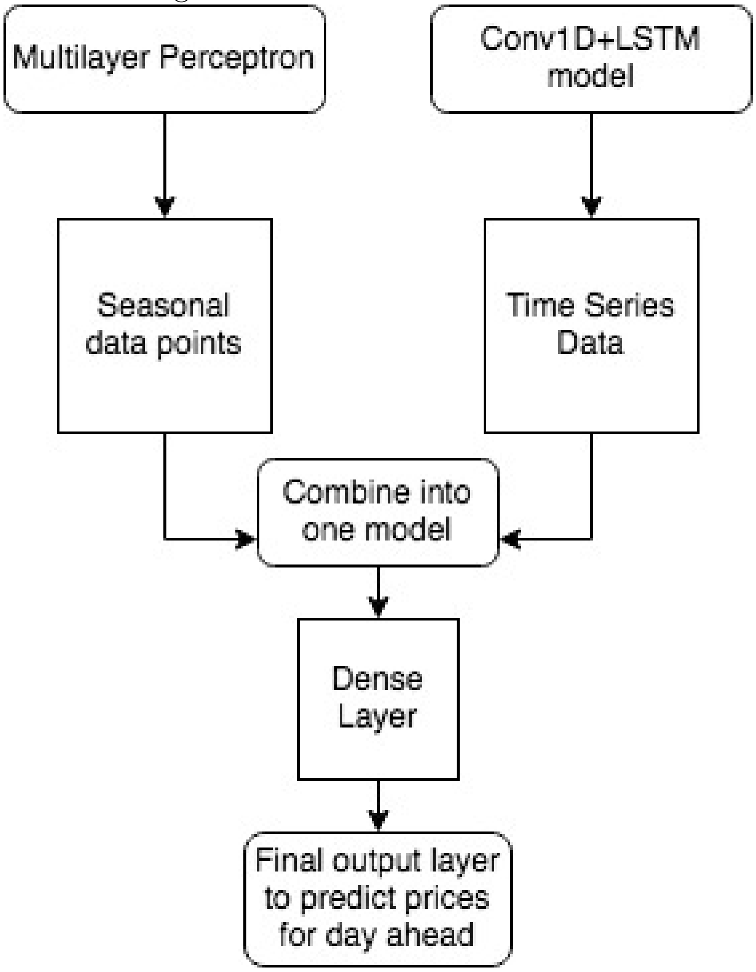


In model 2, the same Convolution 1D-LSTM model is used for Model 1, but with 335 serially correlated lags of differenced price, differenced average price, and residual demand. These features are carefully selected after intense data exploration and were discussed in the data exploration section. Model 2 also comprises a Multilayer Perceptron (MLP) in addition to the revised version of model 1 that we dicussed above. The reason for utilising an MLP is to harness the accuracy that additional data can provide for the prediction of future prices. The MLP model that we designed takes into account the short and long term trend and seasonality. The seasonal features for every sample include month, hour of the week, a dummy variable for weekend days, business hours and public holidays. The dummy variable used for weekend days ascribes Sunday a value of 2, Saturday a value of 1 and any other day a value of 0. As discussed in our exploratory data analysis there was a significant difference between prices on a Sunday and Saturday. The dummy variable for business hours ascribes a value of 2 if the time is between 12-2pm, a value of 1 if it is between 7am-7pm and a value of 0 otherwise. The value of 2 is ascribed to the lunch time period as electricity consumption increases significantly.

Training Model 2:

The MLP model will have two dense layers, one with 64 neurons and the other with 32 neurons. The size of the neurons is chosen after trial and error. RELU is used as our activation function.
With the above inputs, the input data to the model is set to (730, 335, 3) for the enhanced convolution 1D-LSTM model and (730, 5) for the MLP model. The rest remains the same as Model 1 training.

Figure 5.3.2. Concatenated Model



As shown in the diagram above, Figure 5.3.2, we concatenate both, the convolution1D-LSTM network and the MLP network into a bigger neural network. This bigger network will span over an 8 dimensional vector (3 dimensions from conv1D+LSTM and 5 dimensions from MLP) and it would further map to a 32 neuron dense layer and a fully connected dense layer that map to the output layer, which predicts a day ahead. The rest remains the same as Model 1.

Figure 5.3.3. Model 2 Results over 1 week

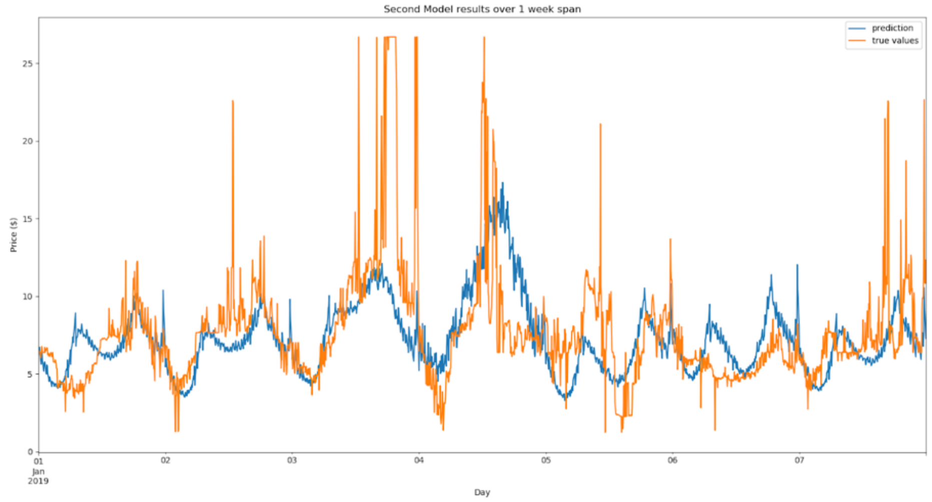
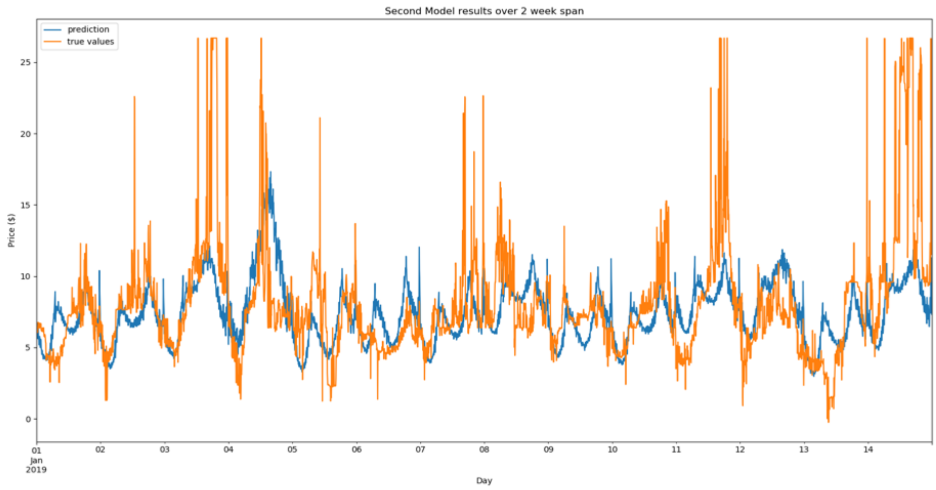


Figure 5.3.4. Model 2 Results over 2 weeks



Compared to Model 1 this model does a better job at predicting the price peaks. However there is a trade off. When feeding these prices to our model however it actually makes less profit compared to Model 1. A possible reason for this is because this model is trying to capture the price peaks it may lead to having these price outliers influence the model. Secondly, the univarite model 1 was validated during training to ensure that the best model is chosen, which retains its performance on the test sample. However, this validation was difficult to carry out for the multivariate model 2. We reckon that fixing this issue would improve our model performance by 5%-10%, hence motivating us to further explore model 2.

With the structure of the model being constant it still needs to ensure that the neural network model is optimized to reduce the error between our predicted price and the actual price. Some of the ways which error has been measured in this model are: Mean Absolute Error, Mean Squared Error, and Root Mean

Squared error. Some of the key hyper parameters that can impact our performance of the neural network are the following:

Number of neurons in each layer:

This hyper-parameter determines the ability for the model to learn complex functions. A general rule of thumb is that the number of neurons in each hidden layer should be in between the size of the input layer and the size of the output layer.

While increasing the number of neurons can theoretically allow our model to learn more complexities of our function, it can also increase the time it takes to train and produce predictions. Due to the models time sensitivity with 5 minute price prediction intervals, it is important to ensure performance is not hindered.

The drop out rate:

This value indicates the probability of dropping the output of a given neuron. Due to how a neural network works, it is constantly aggregating the outputs of all the neurons in the network. Because of this it can sometimes lead to overfitting the training data, rather than generalizing it. The parameter aims to reduce that by randomly ignoring the outputs of certain neurons. The higher the value, the higher the probability a neuron's output is ignored.

The activation function:

This is the function that determines the output of each neuron. After the inputs are fed in, each neuron produces a linear combination of our features with a certain weight. This output is then aggregated and passed into an activation function which introduces some non-linearity in our output. This is a key part of a neural network as it allows it to learn non linear functions. The two activation functions considered in our model testing is the ReLU function and the sigmoid function.

ReLU: $\mathrm{f(x)} = \max(0, \mathrm{x})$

Sigmoid : $\mathrm{f(x)} = (\mathrm{e}^x)/((e^x) + 1)$

The loss function:

This is the function that teaches the neural network how accurate the predictions are. It aims to minimize the loss produced from this function. The two functions considered are the mean absolute error and the mean squared error functions.

$$\text{Mean Absolute Error:} \qquad \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i|$$

n = number of data points, Y_i = observed values, $\hat{Y}_i$ = predicted values

$$\text{Mean Squared Error:} \qquad \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2$$

y_i= prediction x_i= true value n = total number of data points

It is important to note that the MSE loss function has a much greater penalization of errors compared to the MAE loss function, due to the fact that the errors are squared. This can impact the way the model handles the backpropagation when learning the weights in the network.

There can be a variety of models created by experimenting with how the parameters are assigned. The performances of each model can then be evaluated based on the MAE, RMSE, and MSE. After tweaking the hyper parameters, this was identified as the final model which was used to predict prices:

Figure 5.3.5. Final Model Summary

```
Model: "model_2"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 335, 3)       0
__________________________________________________________________________________________________
conv1d_1 (Conv1D)               (None, 1, 64)        53824       input_1[0][0]
__________________________________________________________________________________________________
dense_1_input (InputLayer)      (None, 5)            0
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 1, 64)        0           conv1d_1[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 64)           384         dense_1_input[0][0]
__________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 64)           33024       dropout_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 32)           2080        dense_1[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 32)           2080        lstm_1[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 64)           0           dense_2[0][0]
                                                                 dense_3[0][0]
__________________________________________________________________________________________________
dense_4 (Dense)                 (None, 32)           2080        concatenate_1[0][0]
__________________________________________________________________________________________________
dense_5 (Dense)                 (None, 288)          9504        dense_4[0][0]
==================================================================================================
Total params: 102,976
Trainable params: 102,976
Non-trainable params: 0
__________________________________________________________________________________________________
```

Figure 5.3.6. Final Model Performance over 1 week

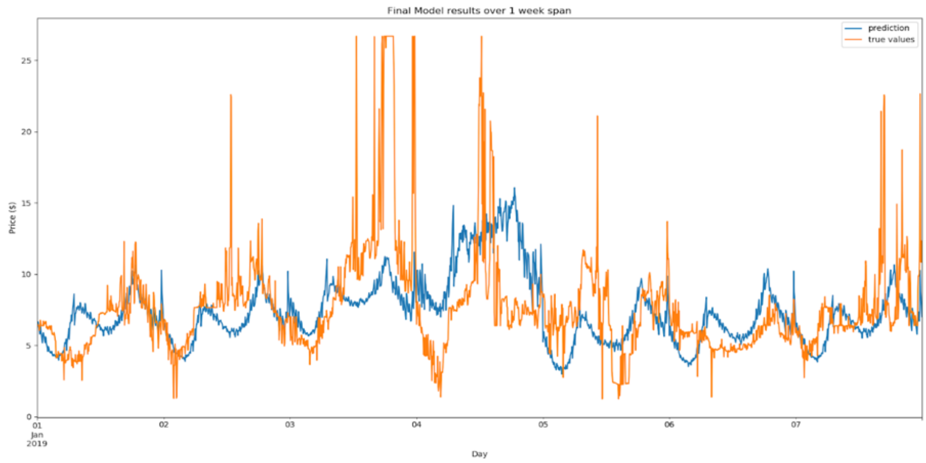
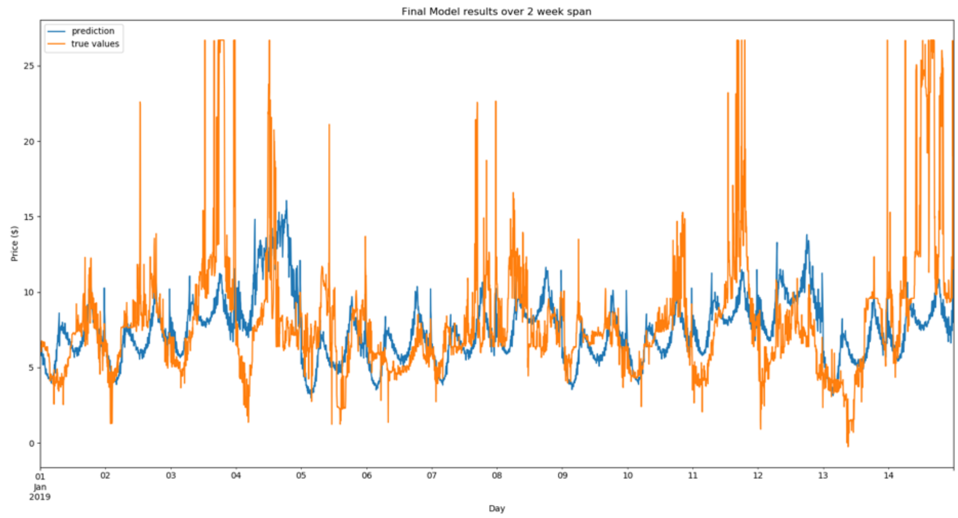


Figure 5.3.7. Final Model Performance over 2 weeks



This model incorporates the best of both of the previous models. It does a better job capturing the larger price peaks compared to model 1, and it also does a better job at predicting the general trend of prices similar to model 2, due to the optimal values of our hyperparameters.

5.4 Formulation of the Optimization Problem

A linear optimisation model was developed to maximise the profit produced by batteries in the Energy market. The variables and the descriptions are stated in table I.

The objective function can then be written as the maximisation of the sum of all the profits in 5-minute intervals from time 0 to time N.

Variable	Definition
X_i	Profit obtained by the battery at time i (\$)
C_i	The dispatch cost at time period i (\$)
R_i	The dispatch revenue at time period i (\$)
G_i	The total amount of dispatch energy generated at time i (MW)
D_i	The amount of energy discharged at time i (MW)
CH_i	The amount of energy charged at time i (MW)
P_i	The price for energy in interval i \$/(MWh)
E_i	The energy available in storage at time i(MWh)
B	Storage capacity of battery at time 0 (MWh)
MD	The maximum discharge rate (MW)
MC	The maximum charge rate (MW)
S	The storage capacity of the battery (MWh)
T	The Round Trip Efficiency of the battery (unitless)
N	The time horizon in 5-minute intervals

Table 5.1: Variables & definitions

$$\max \sum_{i=1}^{\mathbb{N}} X_i \qquad (1)$$

Where the profit X_i can be written as the difference between the dispatch revenue and the dispatch cost at time i.

$$X_i = R_i - C_i \qquad (2)$$

Therefore, the objective function can be rewritten as:

$$\max \sum_{i=1}^{\mathbb{N}} (R_i - C_i) \qquad (3)$$

The decision variables are given by the following expressions:

$$R_i = D_i P_i \qquad (4)$$

$$C_i = -CH_i P_i \qquad (5)$$

$$0 \leq D_i \leq MD \qquad (6)$$

$$MC \leq CH_i \leq 0 \qquad (7)$$

$$0 \leq E_i \leq S \qquad (8)$$

$$G_i \quad (free) \qquad (9)$$

$$\forall i = 0, 1, ..., N$$

Equation (4) states that the dispatch revenue is found by multiplying the discharge rate with the price of energy at time i. Similarly in equation (5), the dispatch cost is calculated by multiplying the charging rate with the price at time i. Noting, that the charging rate will take a negative value resulting in the expression being multiplied by -1. Equation (6) sets the boundaries for the discharge rate, where, the lower bound is 0 and the upper bound is the maximum discharge rate. For the problem presented in this report, the maximum discharge rate and the maximum charging rate is 1 (MWh). Equation (7) sets the boundaries for the charging rate of the battery, where the upper bound is 0 and the lower bound is the maximum charging rate. The maximum charging rate will be non positive and will take the value of -1 (MWh). Equation (8) specifies that the energy available in storage at time i is non negative and the upper bound will be the maximum storage capacity S and is equal to 2(MW). Finally, equation (9) states that the dispatch energy generated by the battery is unbounded.

The linear optimisation problem is also subject to the following constraints:

$$E_i = E_{i-1} - (TCH_{i-1} + D_{i-1}) \qquad (10)$$

$$G_i = D_i + CH_i \qquad (11)$$

$$E_1 = B \qquad (12)$$

$$0 \leq MD \leq 2 \qquad (13)$$

$$-2 \leq MC \leq 0 \qquad (14)$$

$$S \geq 0, B \geq 0 \qquad (15)$$

$$P_i, C_i, R_i \quad (free) \qquad (16)$$

$$0 \leq T \leq 1 \qquad (17)$$

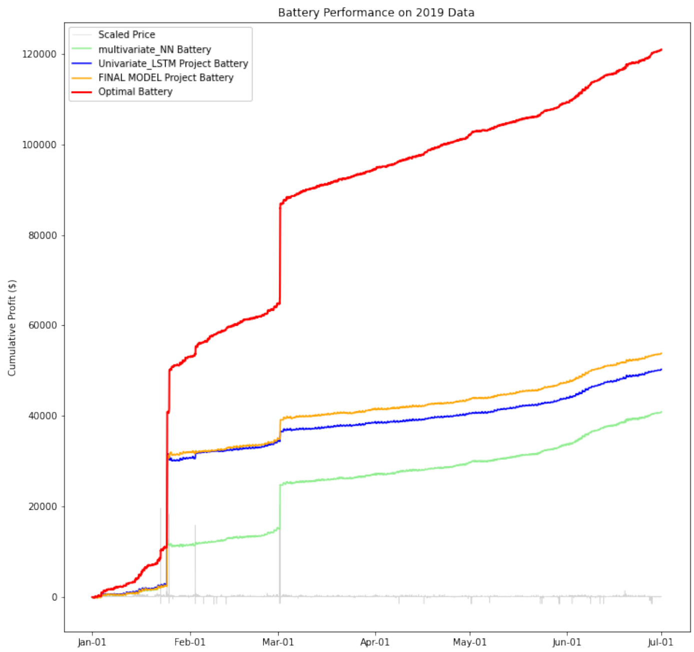
$$\forall i = 0, 1, ..., N$$

Constraint (10) expresses that the amount of energy in storage at time i will be equal to the amount of energy in the storage in the previous time period minus the round trip efficiency times the charge rate in the previous time period and the discharging rate in the previous time period. The round trip efficiency T is the ratio of the energy recovered from the battery storage and the energy supplied to the battery. This ratio will never reach 100% due to the losses in energy accrued from the storage process. Realistically, the ratio usually lies between 75% and 90% when dealing with batteries. The model in this report performs under the assumption that the round trip efficiency is about 81%. Constraint (11) shows that the total energy generated at time i is equal to the the sum of the amount of energy charged and discharged at time i. Constraint (12) sets the energy in storage at the first time period i equal to the storage capacity of the battery at time 0 B where B is set to 0(MW). Constraints (13) and (14) sets the boundaries for the maximum discharge rate and the maximum charge rate respectively. Constraint (15) expresses that the storage capacity of the battery S and the storage capacity of the battery at time 0 B are non negative. Constraint (16) establishes that the price, dispatch cost and dispatch revenue are unbounded. The final constraint (17), sets the boundaries for the round trip efficiency T. Since it is a percentage, T must be between 0 and 1.

5.5 Forecasting & Linear Optimization Results

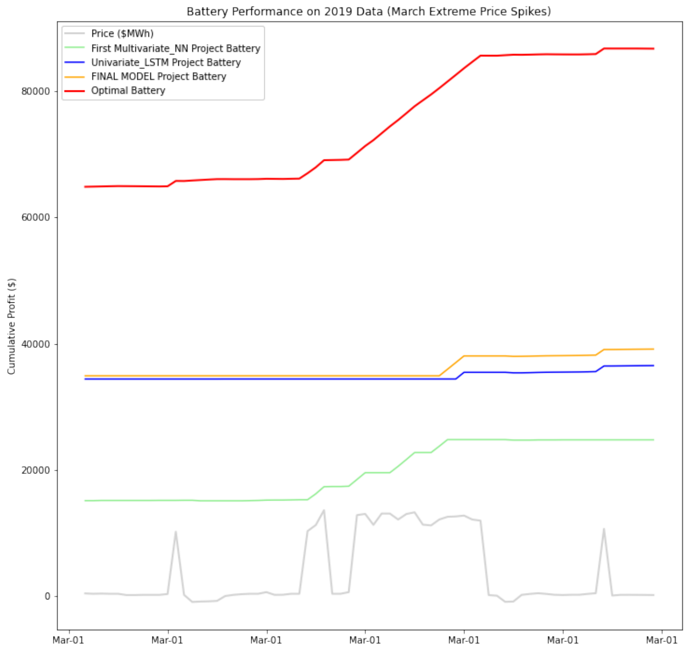
The price predictions obtained from the final model are then used in our linear optimisation model to maximise the profit of the battery subject to the battery constraints. The final model predicts the prices for a time horizon of 6 months (Jan 2019 -Jun 2019) and using these prices the optimisation model then calculates the decisions which are to be performed by the battery in order to maximise profit. The linear optimisation model outputs the optimal predicted profit, the amount of energy (MW) to be charged and discharged as well as the amount of energy lost due to battery storage. The linear optimisation model is then run again with the actual prices keeping the decision variables chosen by the model that ran with the forecasted prices. The results are shown in figure 5.6 where the the optimal profit over the 6 month period is graphed with the actual profits made by 3 models explored in our analysis.

Figure 5.6 battery performance under the different neural network models



It is interesting to note that the first multivariate NN model capitalised on the prices and spikes in March better than the other two models as shown in figure(5.7) but performed worse than the rest of the models overall.

Figure 5.7 battery performance under the different neural network models in June



The optimal objective value of the linear optimisation model was $120921.00 and this will be used as the benchmark for all the models performance. below is a selection of the Gurobi output when running the linear optimisation model with the actual prices only. The predicted profit percentage is calculated as (total revenue - total costs)/(total cost) and the actual profit percentage is calculated as (Actual profit)/(Total optimal profit).

Optimal Battery use:
Model objective value: 120921.00
Actual profit = $120921.00 - 169.78$\%Profit$
Charged 1060.64 MW over 13017 charge intervals (Lost 201.52 MW due to battery inefficiency)
Discharged 858.71 MW over 10529 discharge intervals
Did nothing during 28577 time intervals
Min price during time period: $-1000.00
Max price during time period: $14500.00

The First multivariate_NN model fails to predict the high price spikes in January which causes it to not perform as well as the other models.

First multivariate_NN model:

Model objective value: 22920.11
Predicted profit = $22920.11- 39.04%

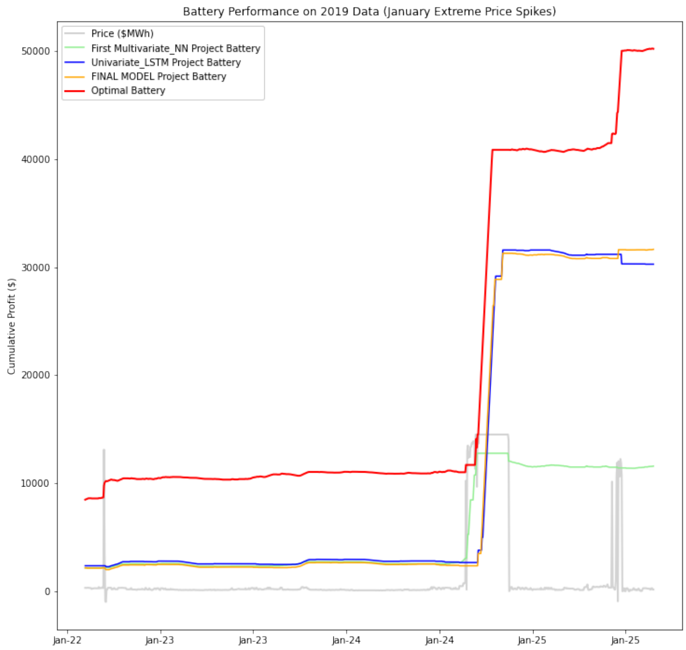
Charged 863.33 MW over 10530 charge intervals (Lost 164.03 MW due to battery inefficiency)
Discharged 699.30 MW over 8473 discharge intervals
Did nothing during 33120 time intervals
Min price predicted during time period: \$21.94
Max price predicted during time period: \$255.82
Actual profit: \$40908.04 - 33.83$\%ofoptimalprofit$

The Univariate LSTM model predicts a lower profit than the First multivariate NN model however it ends up producing more profit when performing with the actual prices. The first model charged the battery more times than the Univariate LSTM which could be another reason as to why it did not perform as well in the actual scenario. The Univariate LSTM model also captures the high spike in prices during January as shown in figure 5.8. The predicted profit could also be lower than the First Multivariate NN Model due to the fact the battery was discharging and charging at a higher rate but since the price predictions weren't close to the real values the predicted profits were lower than the First model.

Univariate_LSTM model:

Model objective value: 15825.49
Predicted profit = \$15825.49 - 41.6%
Charged 597.96 MW over 7361 charge intervals (Lost 113.61 MW due to battery inefficiency)
Discharged 484.35 MW over 5964 discharge intervals
Did nothing during 38798 time intervals
Min price predicted during time period: \$58.78
Max price predicted during time period: \$130.76
Actual profit: \$50301.89 - 41.60$\%ofoptimalprofit$

5.8 battery performance under the different neural network models in January



The Final Multivariate NN model chosen also produces a low predicted profit. However, the actual profit was the highest out of all the models analysed. An observation made was that the battery which was more active achieved lower actual profit than those that did nothing for long intervals.

Final multivariate_NN model:

Model objective value: 18970.71 - 38.39%
Predicted profit = $18970.71
Charged 727.02 MW over 8921 charge intervals (Lost 138.13 MW due to battery inefficiency)
Discharged 588.89 MW over 7131 discharge intervals
Did nothing during 36071 time intervals
Min price predicted during time period: $18.15
Max price predicted during time period: $211.20
Actual profit: $53852.55 - 44.54%$of optimal profit$

Chapter 6
Conclusion

While our assumptions when approaching this task being that the price would follow a general pattern which would vary based on the seasonality of the year, it is important to note that as the years progress the factors that affect price change as well.

Taking into account the occurrences of 2020 and the global pandemic, this assumed predictability of demand could however be hugely undermined and a model based solely on seasonality can run into many issues. The desired model per contra is intended to predict the prices, which are subject to local and global economic forces in real time. A reinforced learning approach model developed in Chapter 4 clearly shows the progress of the most appropriate method of price forecasting taking lags of prices into a hyper-parameter neural network model. This multilayered neural network creates the most accurate model which can be compared to the "Long-Short term" recurrent artificial neural network model developed by LianLian Jiang PHD, and Guoqiang Hu PHD, of Singapore School of Electrical and Electronic Engineering and Nanyang Technological University, Singapore respectively.

Improvements for the future:

Our final model was able to capture some of the price peaks, due to the addition of seasonal data such as business hours, public holidays and seeing if it was a weekday or weekend. One way we can improve this model is including a validation set while training our model. This was hard to implement with regards to the multivariate models we used as we were combining two neural networks together. However we can see the benefits of having a validation set as which was used in model 1, when we were just using our LSTM recurrent neural network. Model 1 actually was more profitable than Model 2, however after adjusting the hyper-parameters which led us to our final model, it performed the best. Adding a validation set to train this model will substantially improve its predictive capacity.

Another area of improvement would be to keep retraining the model once a month. As seen from figure 5.6, we can see that our models captured the first spike in profit maximisation however it did not capture the second spike as well.

This can be mitigated by ensuring we are retraining the model frequently so it has access to the most recent data, so it can learn any new possible price trends.

Bibliography

[1] Kyle Bradbury and Maurice R. Greenberg. How data science can enable the evolution of energy systems. *Digital Decarbonization: Promoting Digital Innovations to Advance Clean Energy Systems*, pages 73–81, 2018.

[2] George Yarrow and Chris Decker. Bidding in energy-only wholesale electricity markets. *AEMC*, 2014.

[3] Rasoul Garmabdari, Mojtaba Moghimi, Fuwen Yang, Junwei Lu, Hui Li, and Zilong Yang. Optimisation of battery energy storage capacity for a grid-tied renewable microgrid. pages 1–6, 12 2017.

[4] Alexander Kies. Joint optimisation of arbitrage profits and battery life degradation for grid storage application of battery electric vehicles. *Journal of Physics: Conference Series*, 977:012005, feb 2018.

[5] Lian Jiang and Guoqiang Hu. Day-ahead price forecasting for electricity market using long-short term memory recurrent neural network. 11 2018.