Abstract

Having the ability to predict future electricity price proposes an interesting strategy to electricity consumption. One can increase his usage during time of low prices and reduce the usage when prices are high to achieve the optimal cost efficiency. However, the lack of correlation of electricity prices in Singapore has made predicting it using other known factors a difficult problem. Singapore has only recently opened its electricity retail market to everyone in 2018 and most research done on this market has been using statistical methods. In this project, we will be utilising the Multilayer Perceptron to model the electricity price market and try to forecast the price of the next 10 days while comparing it to other statistical methods. Experiment was done to find the most optimised parameters in building the neural network using machine learning libraries in Python. Our neural network model was able to successfully predict the trend of the future price, but more experimentation must be done to detect outliers and predict a more accurate price value.

Acknowledgements

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to my supervising professor, Asst Prof Long Cheng, who had walked with me in this journey of discovery and providing advice in writing this report.

Contents

Abstract 1
Acknowledgements 2
Contents 3
List of Tables 4
List of Figures 5
1. Introduction 6
1.1. Background 6
1.2. Purpose and Scope 7
2. Literature Review 8
2.1. Factors Affecting Electric Prices and Trends 8
2.2. Methodology in Price Forecasting 9
2.2.1. Data Pre-processing and Analysis 9
2.2.2. Neural Network 9
2.2.3. ARIMA 10
3. Discussion 11
3.1. Data and Analysis 11
3.2. Feature Engineering 14
3.2.1. Creating the training set 14
3.3. Neural Network 16
3.3.1. Building the model 17
3.4. SARIMA 20
3.4.1. Building the model 20
3.5. Other Statistical Methods 20
3.6. Results 21
4. Conclusion 33
References 33
5. Appendix 35

List of Tables

Table 1. Factors influencing electric prices 8
Table 2. 2019 Electric data 11
Table 3. Correlation matrix of electric price data 14
Table 4. First 8 items of the WEP Time series 15
Table 5. Supervised data generated from 2019 data 15
Table 6. Testing et generated from 2020 data 16
Table 7. Software versions used for neural network 17
Table 8. Model parameters 18
Table 9. Model compilation and fit parameters 18
Table 10. Results of testing neurons number 18
Table 11. Epoch testing results 19
Table 12. Batch size testing results 19
Table 13. Software versions used for SARIMA 20
Table 14. 1-day prediction – Others 24
Table 15. 10 days prediction – Others 28
Table 16. 10th-day prediction – Others 29
Table 17. Error metrics for 1-day forecast on 1st Jan 2020 30
Table 18. Error metrics for 1-day forecast on 26th Feb 2020 30
Table 19. Error metrics for 10th-day forecast on 1st Jan 2020 30
Table 20. Error metrics for 10th-day forecast on 26th Feb 2020 31
Table 21. Runtime for each method 32

List of Figures

Figure 1. Cashflow of the electric market in SIngapore 7
Figure 2. Electric Prices in 2019 11
Figure 3. Outliers truncated 12
Figure 4. Daily mean price data plotted across the hour of the day over the past 4 years 13
Figure 5. Training loss over time 19
Figure 6. 1-day forecast results – Neural Network 22
Figure 7. 1-day forecast results – SARIMA 23
Figure 8. Sliding window for predicting 10 days ahead 25
Figure 9. 10-day forecast results – Neural Network 25
Figure 10. 10th-day prediction - Neural Network 26
Figure 11. SARIMA using 3 weeks of prices to forecast 10 days ahead 26
Figure 12. 10-day forecast results - SARIMA 27
Figure 13. 10th-day prediction – SARIMA 27
Figure 14. Error metrics comparison for 1-day forecast 31
Figure 15. Error metrics comparison for 10th-day forecast 32

1. Introduction

1.1. Background

The Energy Market Authority (EMA) is a government entity that was started to promote competition in Singapore's energy market and to ensure the energy supplied is reliable and secure.

Power generation companies generate electricity from natural gas and oil and sell them to the wholesale market. Electricity retailers then buy the electricity from this wholesale market and sell it to their customers.

Since 2001 [1], EMA has slowly opened the energy retail market to other competitors for business consumers. The high electrical consumption by these businesses means that the cost of electricity plays a big part in their running cost and profits. They will benefit from the increased flexibility and choices when choosing their own retailers who offer different pricing plans for different needs.

Then in May 2019, EMA fully opened the energy market to all households and smaller business accounts. Everyone can now benefit from the flexible plans offered by the numerous retailers.

As of March 2020, there were 3 ways of purchasing electricity for households: Fixed price plans, discounted off the regulated tariff plans and wholesale price.

Fixed-price plans are like traditional regulated tariffs from SP Group where a fixed price agreed beforehand and calculated per kWh is billed every month.

The discounted off the regulated tariff plans offers a fixed discount off the traditional regulated tariff from SP Group and calculated per kWh for billing every month.

Lastly, any consumers can buy from the energy supplier directly in the wholesale market where half-hourly prices are used to determine the cost at the time of usage. However, a 10-day lag is imposed in calculating and releasing of the wholesale price.

Having the ability to forecast future prices is important to both electrical suppliers and consumers during the bidding process. Suppliers can optimise their generation of electricity to prevent wastage and consumers can adjust their usage habits, less during high prices and more during low prices. Thus, successful predictions can lead to rewarding monetary returns.

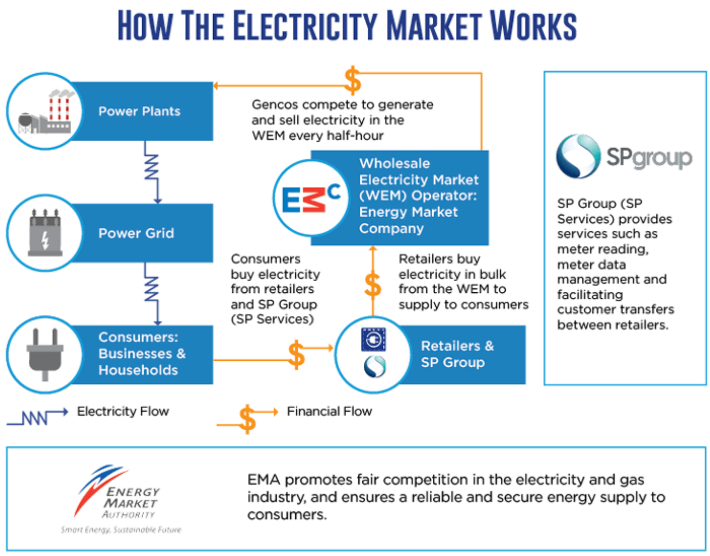


Figure 1. Cashflow of the electric market in SIngapore

1.2. Purpose and Scope

Many pieces of research had been done on predicting future electrical price. Both statistical methods and artificial neural networks have both achieved certain rates of success in predicting the future.

Statistical methods like Autoregressive Integrated Moving Average (ARIMA) usually involve solely the historical prices to perform the regression while the artificial neural network can involve both historical prices and other factors that can influence the electrical prices.

For this project, we will be focusing on the artificial neural network method to forecast the Wholesale Electricity Price (WEP) in Singapore. The WEP is the cost charged to the consumer upon the time of use, and ultimately the price that the consumer must pay.

Due to the lag time of 10 days in which the WEP is released to the public, we will need to be able to forecast 10 days in advance for the viability of this project. Furthermore, a real-time prediction requires fast computation and any delay is detrimental for the user experience.

2. Literature Review

2.1. Factors Affecting Electric Prices and Trends

Sanjeev et al. generalise electric price to contain the following attributes: high volume data with small timesteps, constantly changing mean and variance, highly volatile and outliers are common. They deduce that this is due to the non-tangible aspect of electricity where it cannot be easily stored and there must be an equilibrium between the electric load and generators. Also, demand rarely changes over the small timeframe and the electricity market is commonly oligopolistic. Lastly, both load and generation of electricity can be affected by very unpredictable events. Sudden rain and cloud covers can reduce electricity generated via solar while unforeseen dip in temperature can lead to higher load consumed by heaters. [2]

Sanjeev et al. then categorised the probable factors affecting the price of electricity into 5 classes: market characteristics, nonstrategic uncertainties, other stochastic uncertainties, behaviour indices, and temporal effects.

Class	Input variable
Market Characteristics	(1) Historical load f(load), (2) System load rate, (3) imports/exports, (4) capacity excess/shortfall (5) Historical reserves (6) Nuclear, (7) thermal, (8) hydro generation, (9) generation capacity, (10) net-tie flows, (11) MRR, (12) system's binding constraints, (13) line limits
Nonstrategic Uncertainties	(15) Forecast load, (16) Forecast reserves, (17) temperature, (18) dew point temperature, (19) weather, (20) oil price, (21) gas price, (22) fuel price
Other Stochastic Uncertainties	(23) Generation outages, (24) line status, (25) line contingency information, (26) congestion index
Behaviour Indices	(27) Historical prices, (28) Demand elasticity, (29) bidding strategies, (30) spike existence index, (31) ID flag
Temporal Effects	(32) Settlement period, (33) day type, (34) month, (35) holiday code, (36) Xmas code, (37) clock change, (38) season, (39) summer index, (40) winter index

Table 1. Factors influencing electric prices

Using these factors, they further classify them into different input variables used by different researchers with their own predictive models. Specifically, majority of the factors used are those of (1) Historical load, (15) Forecast load and (27) Historical prices in Table 1.

2.2. Methodology in Price Forecasting

2.2.1. Data Pre-processing and Analysis

According to 2 papers [3] [4], removal of price spikes and outliers gives better accuracy in their neural network models with H.Y. Yamin et al improving their models from 39.89% and 15.47% in mean absolute percentage error (MAPE) of their training and testing sets respectively to 7.98 and 13.7%. Furthermore, instead of simply removing the spikes, a price ceiling was implemented, allowing the model to be trained with these spikes still.

In the Singapore context, Shrestha and Qiao were able to determine that the available generation capacity has the greatest influence in determining the price of the electricity [5]. The correlation was only relevant when the mean price was calculated over some time but the spot price during the time of usage is more crucial for end-users to optimise their electricity usage and reduce cost.

2.2.2. Neural Network

Qi and Zhang [6] describe the difficulty faced when building an artificial neural network. To prevent overfitting, the dataset should be split into 3 parts: training, validating and testing. The training set should be used for modelling the network, tuning the parameters estimations with different network configurations. This model is then evaluated with the validating datasets to find the best specifications. Validity is then lastly checked with the testing dataset.

They also decided to use the past observations of their time series datasets as the inputs and the future values as the output.

Yilmaz and Kaynar [7] also state that neural networks can be a substitute for statistical methods, in solving autocorrelation and regression problems. The artificial network is great for extracting patterns and trends from complex datasets too difficult for humans or other computing methods to recognise. Multilayer perceptron (MLP) and radial basis function (RBF) are found to be widely used for regression and classification problems.

2.2.3. ARIMA

Autoregressive integrated moving average (ARIMA) is a general class of the autoregressive moving average model use for forecasting time series data. The 3 components making this model are the Autoregression (AR), Integrated (I) and Moving Average (MA). ARIMA is often denoted by *ARIMA(p,d,q)* where [8]:

- **p**: The number of lag observations included in the model, also called the lag order.
- **d**: The number of times that the raw observations are differenced, also called the degree of differencing.
- **q**: The size of the moving average window, also called the order of moving average.

Contreras et al. were able to model the Spain and Californian electricity markets using ARIMA [9]. For the Spanish market, 3 different weeks of prices were forecasted: May 25th to 31st, 2020, August 25th to 31st, 2020 and November 13th to 19th, 2000. They were modelled with prices from January 1st to May 24th, 2000, June 1st to August 24th, 2000 and September 1st to November 12th, 2000 respectively. For the Californian market, the week of April 3rd to 9th 2000 was forecasted with prices from January 1st to April 2nd, 2000. They achieved a daily mean error of 5%, 8%, 7% and 5% respectively for each week forecasted.

Seasonal ARIMA (SARIMA) was used by Ismail and Mahpol to model the electric demand in Malaysia [10]. The seasonal effect of the dataset was modelled together into the ARIMA model of the dataset with S*ARIMA(p,d,q)(P,D,Q)m* where [11]:

- **P**: Seasonal autoregressive order.
- **D**: Seasonal difference order.
- **Q**: Seasonal moving average order.
- **m**: The number of time steps for a single seasonal period.

Their model with parameters *SARIMA(1,1,0)(1,0,1)* performed better than their ARIMA model with the lowest mean square error of 0.00184.

3. Discussion

3.1. Data and Analysis

All data are downloaded and compiled from the Energy Market Company Pte Ltd (EMC) website. They are the middleman between electricity buyer and seller, regulating the market and providing the trading infrastructure for Singapore.

The first five rows of the 2019 data are shown in Figure 2 where the Wholesale Electricity Price (WEP), Uniform Singapore Energy Price (USEP), power demanded, gross power generated, and net power generated are compiled against the DateTime index.

DATE	PERIOD	WEP ($/MWh)	USEP ($/MWh)	DEMAND (MW)	GROSS INJECTION (MWh)	NET INJECTION (MWh)
1/1/2019 0:00	1	83.33	82.7	5201.89	2555.1	2362.296
1/1/2019 0:30	2	83.83	82.71	5150.461	2549.41	2357.615
1/1/2019 1:00	3	83.19	82.7	5106.794	2519.013	2327.042
1/1/2019 1:30	4	83.13	82.69	5075.841	2492.473	2300.457
1/1/2019 2:00	5	83.2	82.67	5044.147	2453.576	2261.511

Table 2. 2019 Electric data

The prices of 2019 in Figure 3 intuitively shows a large amount of random price spikes and any trends or seasonality are not immediately obvious within a year.

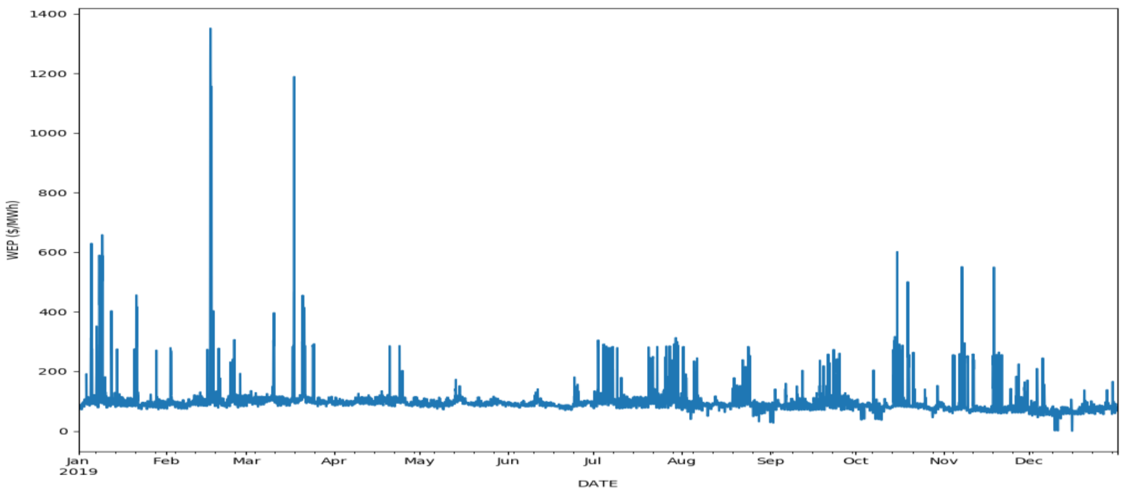


Figure 2. Electric Prices in 2019

In Figure 4, outliers were truncated to within 3 standard deviations from the mean to allow better visualisation of the monthly and daily trends if any.

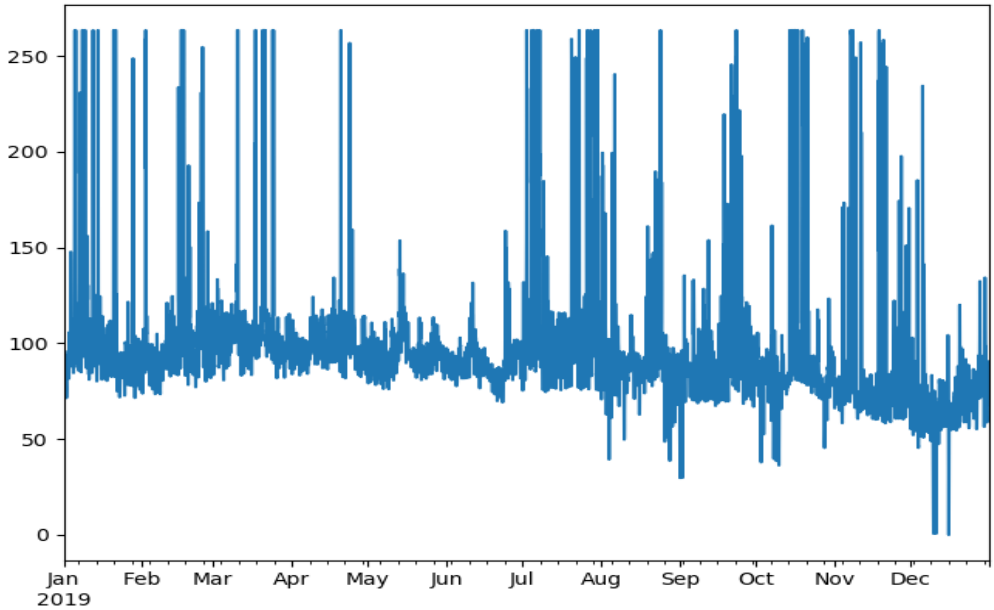


Figure 3. Outliers truncated

We further analyse the effect of date and time on the WEP in Figure 5. Observation on the daily mean price shows the same trend in the past 4 years of data. The duration between 8 to 10 am and 6 to 9 pm shows large spikes in prices across all the days of the week with Sunday being the lowest during the day but highest during the night.

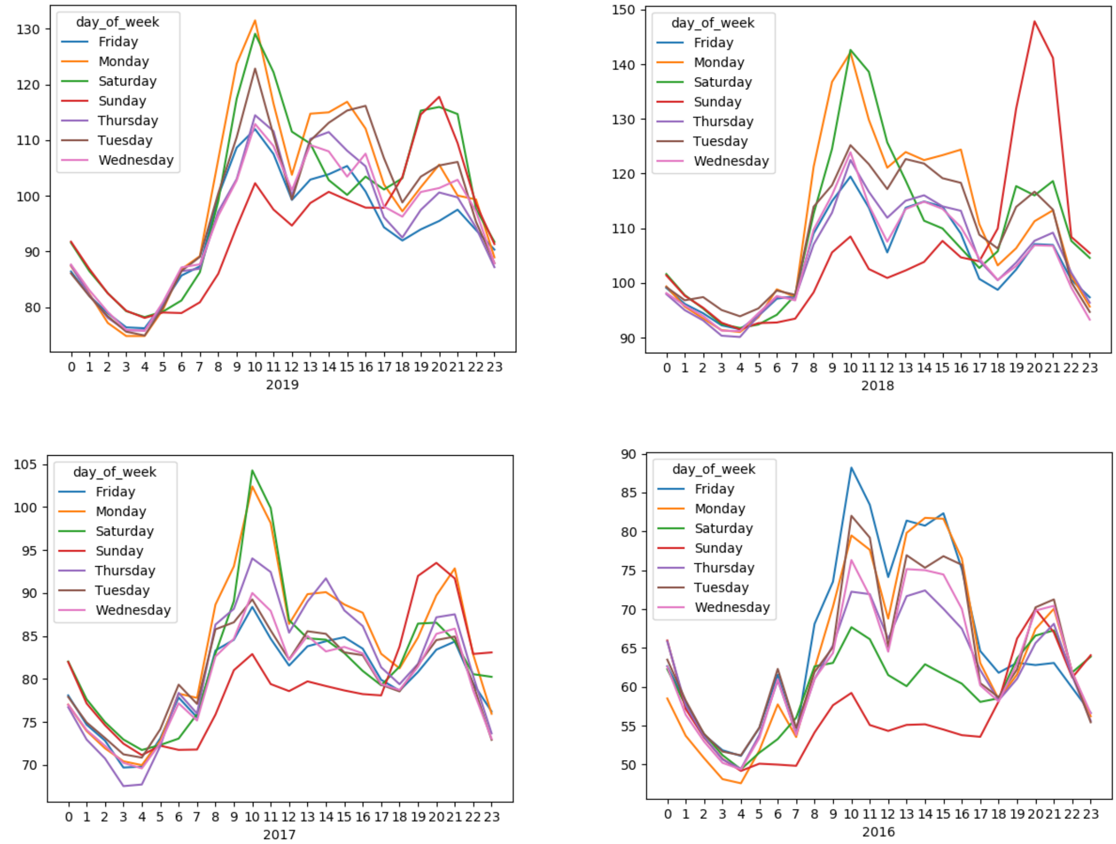


Figure 4. Daily mean price data plotted across the hour of the day over the past 4 years

Plotting the correlation matrix in Table 2 shows that neither demand nor supply is highly correlated to the WEP. The high correlation between WEP and USEP is due to WEP being derived from USEP and the different tariffs and administrative costs. From these observations, trying to predict the spot prices simply using demand and supply in Singapore may not be very effective. A time series approach will be taken for the prediction.

	PERIOD	WEP ($/MWh)	USEP ($/MWh)	DEMAND (MW)	GROSS INJECTION (MWh)	NET INJECTION (MWh)
PERIOD	1	0.141533322	0.1434684	0.5388172	0.5550295	0.5610973
WEP ($/MWh)	0.1415333	1	0.99990099	0.250603	0.2561924	0.2535366
USEP ($/MWh)	0.1434685	0.999900995	1	0.2544707	0.2601555	0.2574505
DEMAND (MW)	0.5388172	0.25060302	0.2544707	1	0.9954964	0.9925663
GROSS INJECTION (MWh)	0.5550295	0.256192447	0.2601555	0.9954964	1	0.9959568

Table 3. Correlation matrix of electric price data

3.2. Feature Engineering

Since the weekly data shows signs of seasonality and the characteristics of the prices, we propose a 7-day time lag to be used to train and predict the next day price. 7 days of data requires a lag time step of 336 and a day of data needs 48 future time steps due to the half-hour pricing in the dataset. Thus, we can create a supervised dataset where the input and output vector will be 336 and 48 respectively, totaling to a size of 384 columns in our training data.

The 2019 WEP price will be used for training and validation while the 2020 WEP price up to March will be used to testing.

3.2.1. Creating the training set

3.2.1.1. Transform data into time series

Generating the training data requires first converting the 2019 data into a time series data. The DateTime index will help to slice the data during the process of training and validating.

	WEP
2015-01-01 00:00:00	92.055
2015-01-01 01:00:00	90.660
2015-01-01 02:00:00	82.655
2015-01-01 03:00:00	76.625

2015-01-01 04:00:00	75.865
2015-01-01 05:00:00	75.325
2015-01-01 06:00:00	76.685
2015-01-01 07:00:00	77.510

Table 4. First 8 items of the WEP Time series

3.2.1.2. Building the lag dataset

The first 336 columns *(t-336, t-335, t-334, t-333, …, t-4, t-3, t-2, t-1)* are generated from the price before timestep *t*. The next 47 columns *(t+1, t+2, t+3, t+4, …, t+44, t+45, t+46, t+47)* are generated from the price after timestep *t*.

DATE	var1(t-336)	var1(t-335)	var1(t-334)	var1(t-333)	…	var1(t+44)	var1(t+45)	var1(t+46)	var1(t+47)
2019-01-08 00:30:00	83.83	83.19	83.13	83.20	…	115.01	103.04	93.63	105.15
2019-01-08 01:00:00	83.19	83.13	83.20	78.81	…	103.04	93.63	105.15	100.07
2019-01-08 01:30:00	83.13	83.20	78.81	78.30	…	93.63	105.15	100.07	97.01
2019-01-08 02:00:00	83.20	78.81	78.30	73.55	…	105.15	100.07	97.01	97.54
…	…	…	…	…	…	…	…	…	…
2019-12-30 22:00:00	80.92	75.99	71.83	67.34	…	79.40	79.94	78.58	75.21
2019-12-30 22:30:00	75.99	71.83	67.34	66.62	…	79.94	78.58	75.21	72.43
2019-12-30 23:00:00	71.83	67.34	66.62	66.32	…	78.58	75.21	72.43	69.58
2019-12-30 23:30:00	67.34	66.62	66.32	58.67	…	75.21	72.43	69.58	70.21

Table 5. Supervised data generated from 2019 data

3.2.1.3. *Building the test dataset*

Similarly, the 2020 data must be converted into a supervised dataset but without the need of the 47 future time steps. Only the input vectors will be fed into the model for prediction and the results will be compared to the existing data.

DATE	var1(t-336)	var1(t-335)	var1(t-334)	...	var1(t-2)	var1(t-1)	WEP
2020-01-08 00:30:00	75.50	72.69	70.36	...	65.77	53.85	69.74
2020-01-08 01:00:00	72.69	70.36	70.33	...	53.85	69.74	55.00
2020-01-08 01:30:00	70.36	70.33	69.55	...	69.74	55.00	53.31
2020-01-08 02:00:00	70.33	69.55	68.62	...	55.00	53.31	52.34
...	...	...	...	...	...	...	...
2020-03-01 21:30:00	79.59	76.70	73.22	...	79.78	85.19	79.49
2020-03-01 22:00:00	76.70	73.22	72.83	...	85.19	79.49	73.27
2020-03-01 22:30:00	73.22	72.83	70.71	...	79.49	73.27	71.14
2020-03-01 23:00:00	72.83	70.71	70.53	...	73.27	71.14	70.11

Table 6. Testing et generated from 2020 data

3.3. Neural Network

An artificial neural network is a system of interconnected nodes or neurons organized in layers, processing information between each neuron and layer. Typically, 3 layers are used to build the network: the input layer, the hidden layer and the output layer.

The input layer consists of neurons corresponding to the input of our data, the 336 historical timesteps from *t*. The hidden layer can consist of 1 or more layers with any number of neurons. The inputted information flowing through this layer is processed and characterized by weights, biases and a pre-set activation function to reach the output data. Lastly, the output layer will contain as many neurons as the number of output data needed, 48 in total for the timesteps we are predicting into the future.

A typical neural network that is built using these layers of neurons is the MLP. MLP requires a supervised learning dataset and backpropagation is used for training. In this project, we will be using the Model object in Keras [12] to sequentially build our network and fit it into a model for prediction.

Finding the optimal configuration for the network and model requires some experimentation. For this project, we will be settling with 4 layers, 1 input layer, 2 hidden layers and 1 output layer, using the mean absolute error (MAE) as the loss function and the mean square error (MSE) to validate.

3.3.1. Building the model

3.3.1.1. Software environment

Software	Version
Python	3.7.0
Keras	2.3.1
TensorFlow	2.1.0

Table 7. Software versions used for neural network

For this project, we are using Python3 as the scripting language for running and processing the tests used. Keras is a machine learning library written for Python and TensorFlow will be the backend engine for Keras.

3.3.1.2. Neural Network Layers

Building a model in Keras can be done in layers using their Sequential model API [13]. The default settings for this project are shown in Table 7 and 8.

1. The input shape, activation function and number of neurons must be declared in the first hidden layer
2. Subsequent hidden layers only need the activation function and number of neurons

3. The output layer needs the output function and the same number of neurons corresponding to the outputs and the
4. The model is compiled with the loss function, optimizer and validation metric.
5. The model is fitted with the dataset split into training and validation set. The epoch size and batch size will determine the training procedure.

The model will be trained 5 times per testing parameters and the lowest mean MSE will be chosen as the optimal option.

Layer	**Parameters**		
Hidden layer 1	Input shape = (336,)	Neurons = 240	Activation = ReLu
Hidden layer 2	Neurons = 240	Activation = ReLu	
Output layer	Neurons = 48	Activation = Linear	

Table 8. Model parameters

Model Compilation and Fit			
Compile	Loss = MAE	Optimizer = adam	Metrics = MSE
Fit	Epoch = 100	Batch = 48	Validation split = 0.2

Table 9. Model compilation and fit parameters

3.3.1.3. Finding the best neurons number for each layer

We have chosen 6 different iterations of the layers' neurons for testing and the results are shown in Table9. Our dataset seems to a favor high number of neurons during training as the fewer neurons we used, the higher the MSE during validation. Thus, we decided to use 240 neurons for each of our hidden layers.

Loss Function = MAE, Epoch = 200, Batch size = 48						
Neurons (1st layer, 2nd layer)	**(240,240)**	**(240,120)**	**(240,60)**	**(120,60)**	**(120,30)**	**(60,20)**
MSE	532.27	552.11	600.70	660.59	730.57	757.94

Table 10. Results of testing neurons number

3.3.1.4. Finding the epoch

5 epoch size was chosen for testing and the results are shown in Table 10. We observed that the longer we train, the lower the MSE. Therefore, we decided to go with the highest epoch we tested for our project. The training loss also sees diminishing after 200 epochs in Figure 6.

Loss Function = MAE, Layer 1 = 240 neurons, Layer 2 = 240 neurons, Batch size = 48					
Epoch	**20**	**50**	**100**	**200**	**300**
MSE	721.27	669.45	593.31	536.20	531.62

Table 11. Epoch testing results

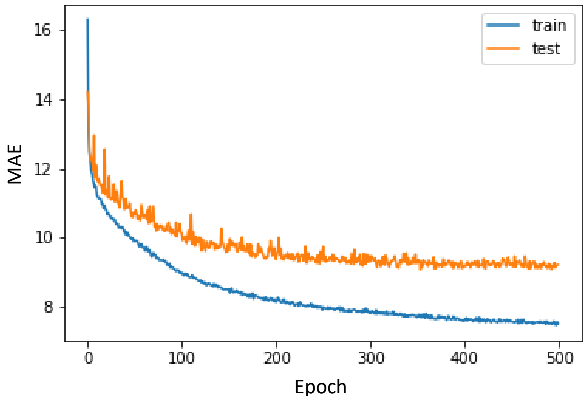


Figure 5. Training loss over time

3.3.1.5. Finding the optimal batch size

During the batch size testing, we observed that a batch size of 64 has the lowest MSE over an average of 5. This can be due to the network being able to see a larger amount of data and detecting the seasonal effect of WEP. Since a weekly season is present in our data, and a week is 48 timesteps, a batch size of 48 and higher should result in a faster learning rate and better prediction.

Loss Function = Mean Absolute Error, Epoch = 300, Batch size = 48						
Batch size	4	8	16	32	48	64
MSE	704.226965	649.406872	597.228112	512.462744	491.711132	485.659454

Table 12. Batch size testing results

3.3.1.6. Final model configuration

Using an epoch of 500, batch size of 64 and 240 neurons for each hidden layer, a trained model was generated. This model will be used to forecast the next day WEP.

3.4. SARIMA

3.4.1. Building the model

3.4.1.1. Software environment

Software	Version
Python	3.7.0
pmdarima	1.5.3

Table 13. Software versions used for SARIMA

3.4.1.2. Finding the SARIMA hyperparameter

The python library *pmdarima* [14] can iteratively find the best sets of parameters in the *SARIMA(p,d,q)(P,D,Q)m* model by finding the values with the lowest Akaike Information Criteria (AIC). We specified *m=48* due to the daily seasonal trend of our dataset and a day has 48 steps.

3.4.1.3. Modelling the sample

3 weeks of historical data will be used to model the next day prices. So, if we want to predict the prices of 14th January 2020, the historical prices from 24th December 2019 to 13rd January 2020 will be used.

3.5. Other Statistical Methods

The Simple Exponential Smooth (SES), Holt Winter's Exponential Smoothing (ES) additive method and Seasonal Naïve methods are used as a baseline to compare with the models we have built.

From the equation below, SES uses a single smooth factor or coefficient called *alpha (α)* to control how much the previous time step *(t)* has an influence on the current time step. Between 0 and 1, the larger the value, the more recent the history will be used for calculating the current time step.

$$St = \alpha yt - 1 + (1 - \alpha)St - 1$$

The Holt-Winters seasonal method shown below comprises the forecast equation and three smoothing equations — one for the level *ℓt*, one for the trend *bt*, and one for the seasonal component *st*, with corresponding smoothing parameters *α*, *β∗* and *γ*. We use *m* to denote the frequency of the seasonality, i.e., the number of seasons in a year. *k* is the integer part of *(h−1)/m*, which ensures that the estimates of the seasonal indices used for forecasting come from the final year of the sample. The level equation shows a weighted average between the seasonally adjusted observation *(yt−st−m)* and the non-seasonal forecast *(ℓt−1+bt−1)* for time t. The trend equation is identical to Holt's linear method. The seasonal equation shows a weighted average between the current seasonal index, *(yt−ℓt−1−bt−1)*, and the seasonal index of the same season last year (i.e., m time periods ago).

$$\begin{aligned} yt + h|t = \ell t + hbt + st + h - m(k+1)\ell t \\ = \alpha(yt - st - m) + (1-\alpha)(\ell t - 1 + bt - 1)bt \\ = \beta*(\ell t - \ell t - 1) + (1 - \beta*)bt - 1st \\ = \gamma(yt - \ell t - 1 - bt - 1) + (1-\gamma)st - m, \end{aligned}$$

Lastly, the Seasonal Naïve method simply use the last observed value from the previous week as the forecast due to the weekly seasonal effect of our dataset as shown in the equation below. The *m* = the seasonal period, and *k* is the integer part of *(h−1)/m* (i.e., the number weeks in the forecast period before time *T+h*).

$$yT + h|T = yT + h - m(k+1),$$

3.6. Results

The trained model is validated against the 2020 test set for the next day WEP. Similarly, when training the model, the outliers in the test set will be truncated to 3 standard deviations from the mean before being fed into the model. However, when comparing the predicted and actual, the non-truncated values will be used instead.

3.6.1.1. Forecasting 1-day ahead results – Neural Network

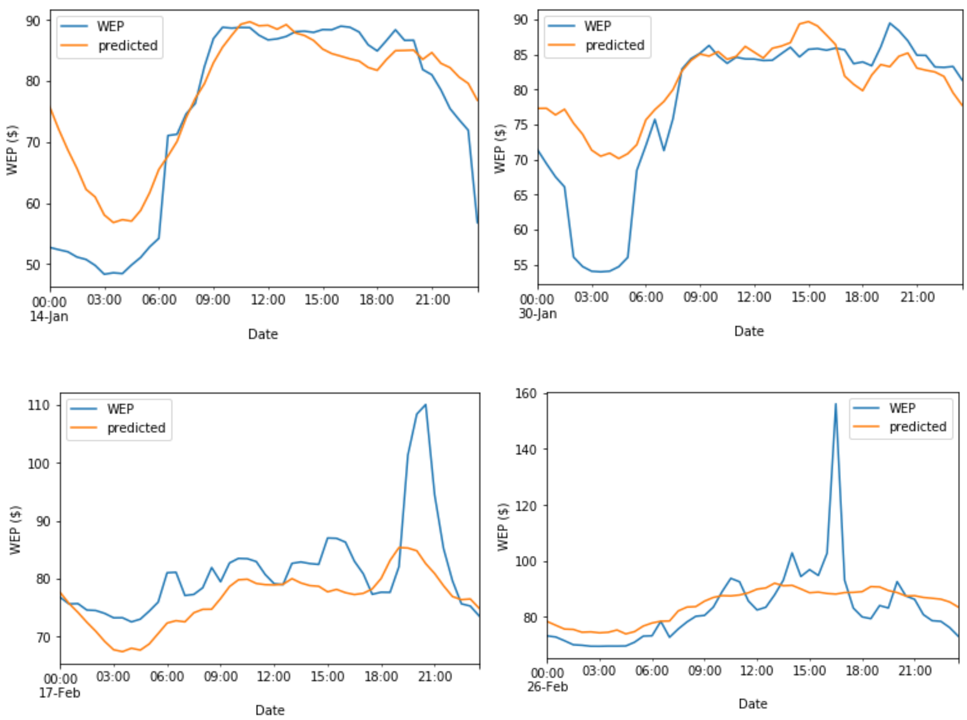


Figure 6. 1-day forecast results – Neural Network

From Figure 6, our model shows capability in predicting the daily trend shown in our test data. The trough between 2 to 4 am and the gradual increase after that were mostly predicted in the 4 cases. However, it is not able to detect outliers as shown in the peaks on 17 Feb 10pm and 26 Feb 4pm.

3.6.1.2. Forecasting 1-day ahead results – SARIMA

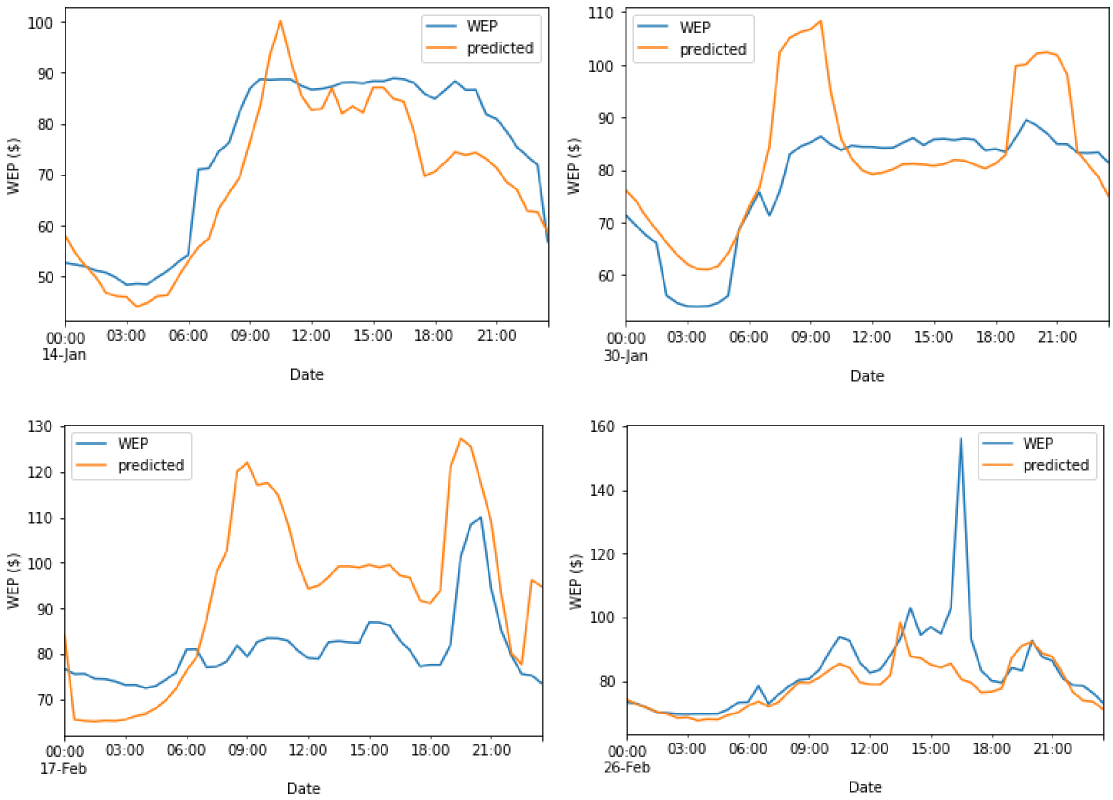
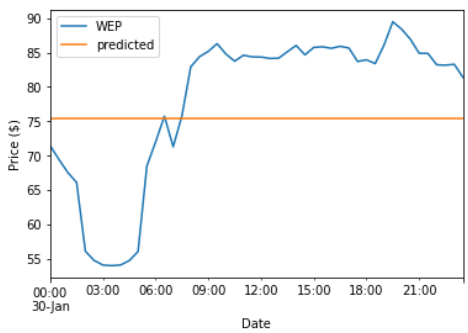
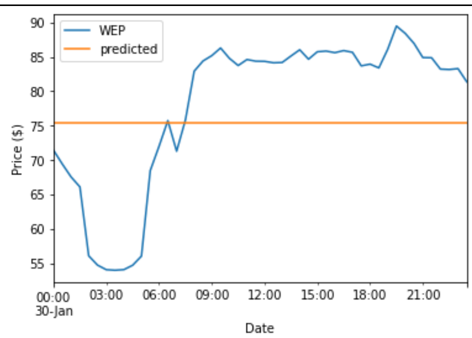


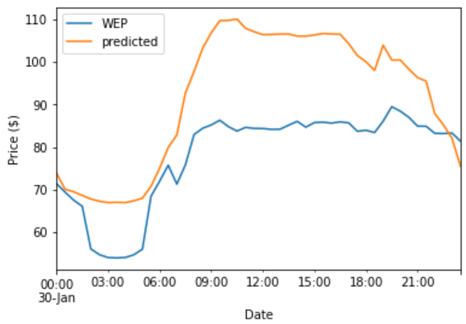
Figure 7. 1-day forecast results – SARIMA

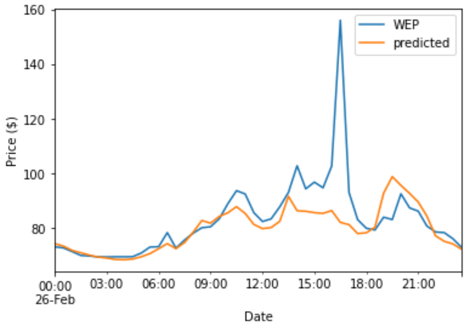
The SARIMA model is also able to capture the daily trend where 2 – 4 am will dip in price and gradually increase after that. Likewise, outliers cannot be modelled as shown in the 26 February 2020 results.

3.6.1.3. Forecasting results – 1-day ahead – Others

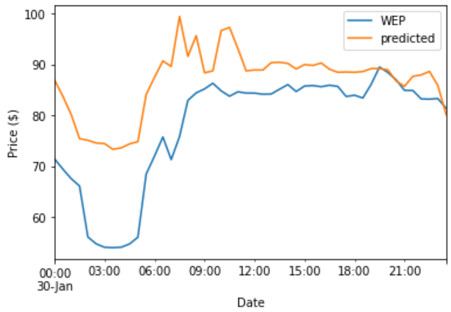
	30-01-2020	**26-02-2020**
Simple Exponential Smoothing		

Holt-Winters seasonal method (Additive)





Seasonal Naive



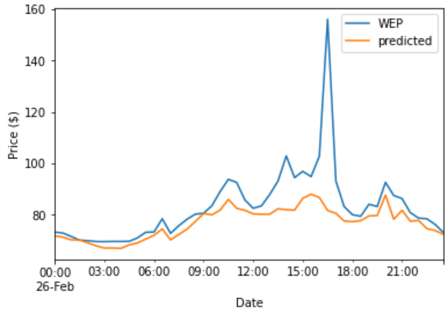


Table 14. 1-day prediction – Others

From Table 14, the SES method can only give us a basic peek into the future as it only calculates the mean from the historical prices. The model does not account for trends and seasonality and only project the calculated mean as a constant throughout the predicted timesteps.

For the ES method, the weekly seasonal effect of the WEP was modelled, however, the absolute values are inconsistent due to the natural fluctuation in the electric prices.

Similarly, for the naïve method, by simply using the previous week prices as the forecast, the seasonality can be modelled. However, if the previous week has many outliers, the forecast will fail staggeringly.

3.6.1.4. Forecasting results – 10th-day ahead – Neural Network

Forecasting 10 days ahead requires a sliding window for the inputted values as shown in Figure 7. The newly predicted values of the next day will be used as the latest input for the prediction of the following day.

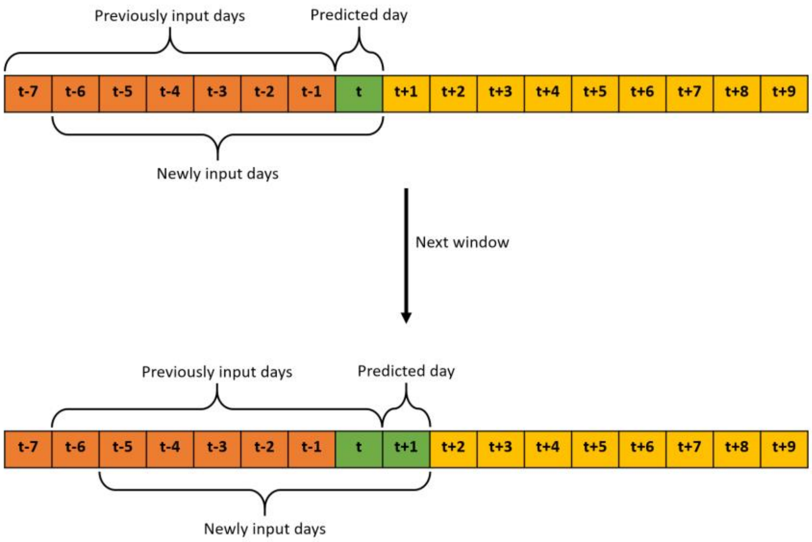


Figure 8. Sliding window for predicting 10 days ahead

Therefore, by sliding the window 10 days forward, our model was able to reproduce the daily trend in the forecast but similarly to the 1-day forecast, not able to predict the large spikes in prices as shown in Figure 8.

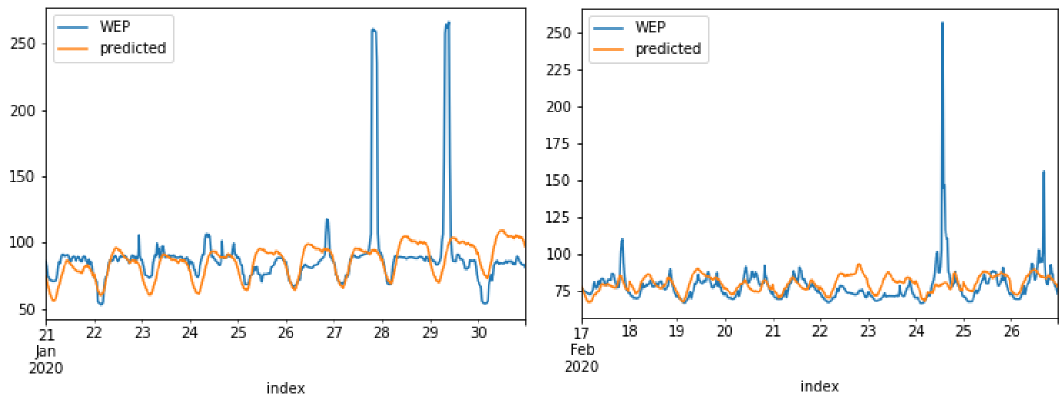


Figure 9. 10-day forecast results – Neural Network

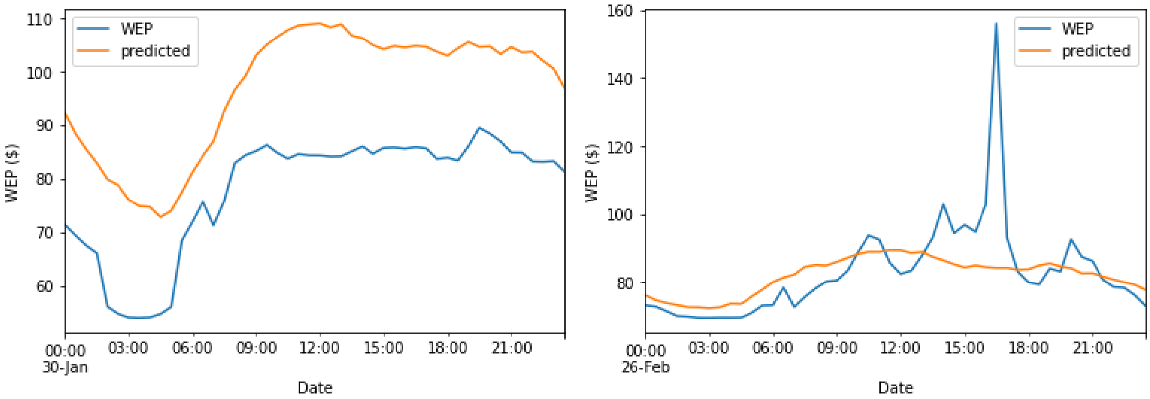


Figure 10. 10th-day prediction - Neural Network

The 10th-day forecast exhibits a similar trend to our 1-day forecast. Since the model has captured the daily seasonality of the prices, the results show good results when outliers are not present. For 30th January 2020, we got significantly worse results for all 3 metrics: 123%, 272% and 102% worse for the MAE MSE and MAPE respectively. For 26th February 2020, the increase in error rates were 17%, 21% and 17% for the MAE, MSE and MAPE respectively.

3.6.1.5. Forecasting results – 10th-day ahead – SARIMA

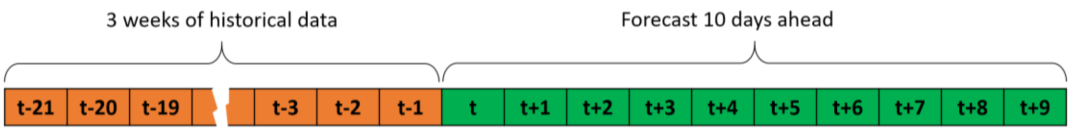


Figure 11. SARIMA using 3 weeks of prices to forecast 10 days ahead

From Figure 12, the SARIMA model will directly forecast the 10th-day price from the 3 weeks of data.

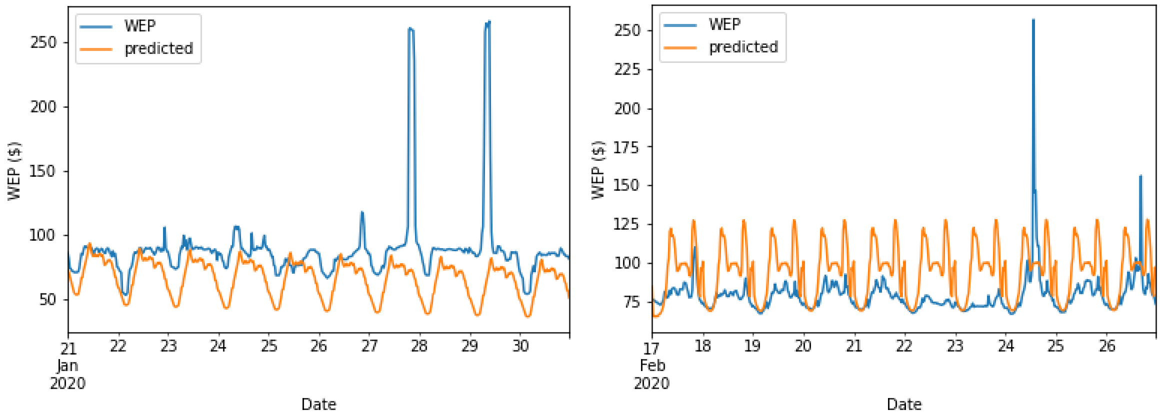


Figure 12. 10-day forecast results - SARIMA

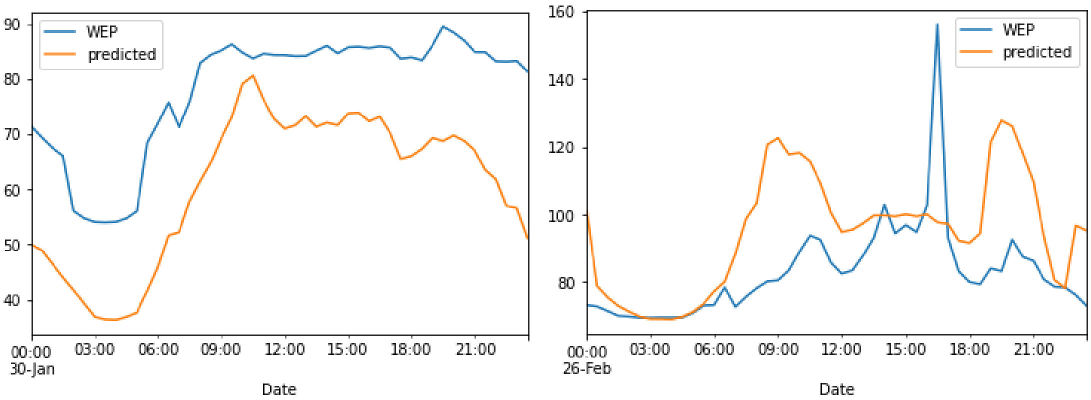


Figure 13. 10th-day prediction – SARIMA

The SARIMA model suffers the same problem as our neural network model. The daily trend was captured, and the outliers couldn't be detected. For 30th January 2020, we got significantly worse results for all 3 metrics: 122%, 219% and 127% worse for the MAE MSE and MAPE respectively. For 26th February 2020, the increase in error rates was 159%, 174% and 202% for the MAE, MSE and MAPE respectively.

3.6.1.6. Forecasting results – 10th-day ahead - Others

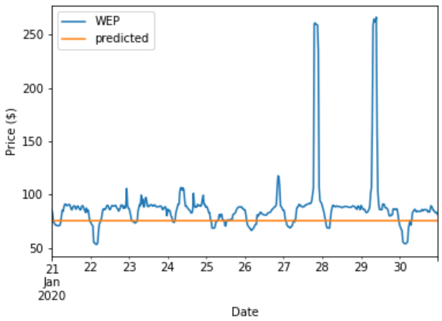
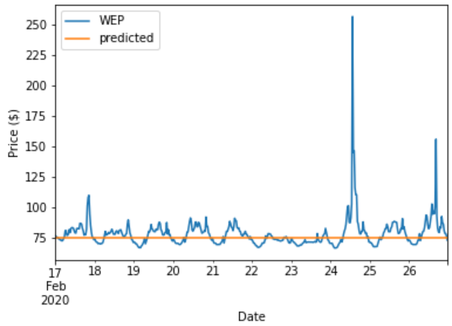
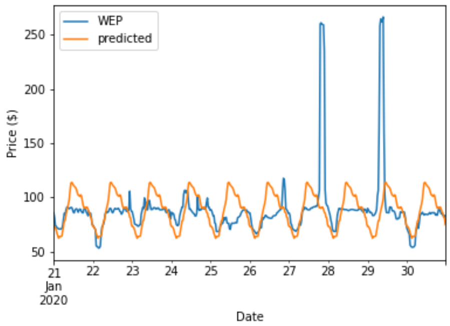
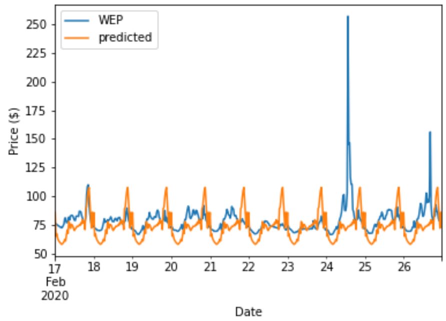
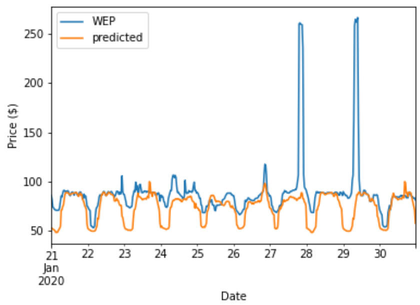
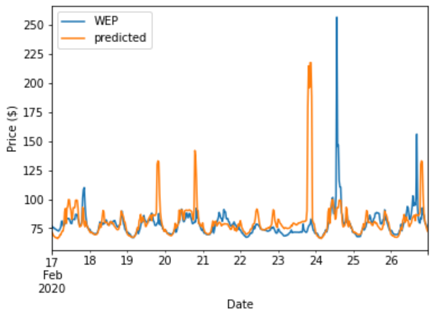
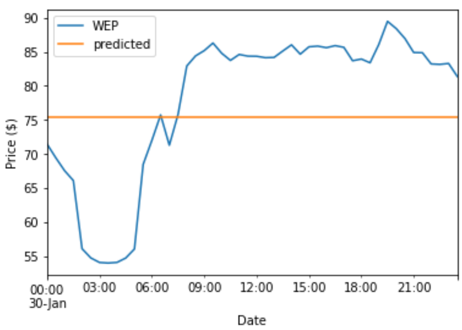
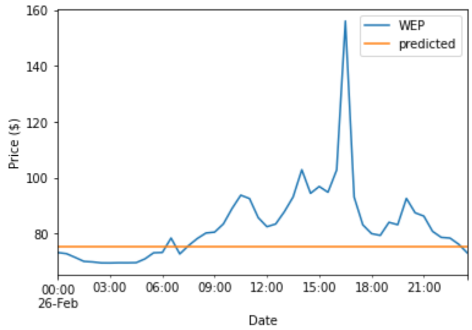
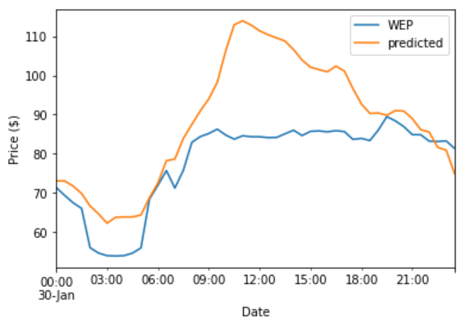
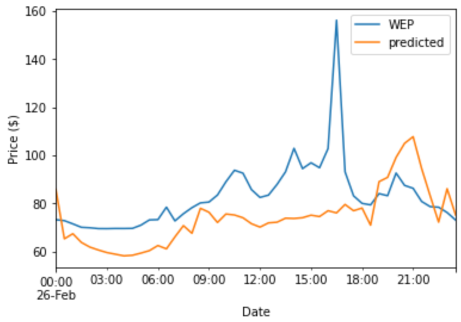
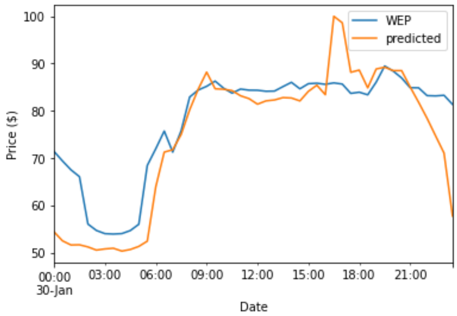
<table>
<tr><td>10 days</td><td>30-01-2020</td><td>26-02-2020</td></tr>
<tr><td>Simple Exponential Smoothing</td><td>
</td><td>
</td></tr>
<tr><td>Holt-Winters seasonal method (Additive)</td><td>
</td><td>
</td></tr>
<tr><td>Seasonal Naive</td><td>
</td><td>
</td></tr>
</table>

Table 15. 10 days prediction – Others

<table>
<tr><td>10th-day</td><td>30-01-2020</td><td>26-02-2020</td></tr>
<tr><td>Simple Exponential Smoothing</td><td>
</td><td>
</td></tr>
</table>

<table>
<tr>
<td>Holt-Winters seasonal method (Additive)</td>
<td>


</td>
</tr>
<tr>
<td>Seasonal Naive</td>
<td>

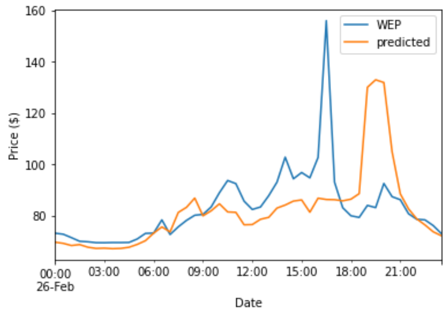

</td>
</tr>
</table>

Table 16. 10th-day prediction – Others

For the 10-th day prediction, the SES method performs similarly to the 1-day forecast with less than 1% difference for all 3 metrics for the 30th January 2020. However, due to the outliers on 26th February 2020, where the change in price did not follow the hourly trend, resulting in a seemingly improved prediction with the MAE, MSE and MAPE decreased by 9%, 15% and 8% respectively.

The ES method gained an improvement for about 23-24% for all 3 metrics for the 30th January 2020 forecast. It got worse for the 26th February 2020 forecast with the MAE, MSE and MAPE increased by 144%, 107% and 178% respectively.

Like the ES method, the naïve method shows improvement within the 10th-day forecast on the 30th January 2020 with the MAE, MSE and MAPE decreased by 41%, 53% and 46% respectively. The forecast on 26th February 2020 had the errors MAE, MSE and MAPE increased by 49 %, 77% and 58% respectively.

As both the ES and naïve methods follow the weekly and hourly seasonality of our historic prices, we will get good performance if the future prices can continue to follow the trend.

3.6.1.7. Comparing the error metrics

	30/1/2020		
	MAE	**MSE**	**MAPE**
MLP	8.489415862	100.2025	12.33354
SARIMA	7.755643225	102.5798	10.10524
SES	10.184583	129.0824	14.17657
ES	14.090496	253.8571	17.95805
Naive	8.841042	124.0843	12.93082

Table 17. Error metrics for 1-day forecast on 1st Jan 2020

	26/2/2020		
	MAE	**MSE**	**MAPE**
MLP	5.515156	120.1595	5.860455
SARIMA	5.601297	152.0624	5.590852
SES	11.38139	307.0058	12.03786
ES	5.343782	146.7667	5.283086
Naive	6.188958	155.7436	6.280658

Table 18. Error metrics for 1-day forecast on 26th Feb 2020

	30/1/2020 - 10th		
	MAE	**MSE**	**MAPE**
MLP	18.94647842	373.2159	25.00028
SARIMA	17.28722707	327.7626	22.97553
SES	10.222083	129.5238	14.20935
ES	10.78784	190.3937	13.76917
Naive	5.142708	58.07003	6.97943

Table 19. Error metrics for 10th-day forecast on 1st Jan 2020

	26/2/2020 - 10th		
	MAE	**MSE**	**MAPE**

MLP	6.449424	145.944	6.866359
SARIMA	14.53031	417.9716	16.92362
SES	10.28646	259.492	10.95702
ES	13.08831	304.9746	14.72382
Naive	9.233333	276.5361	9.935335

Table 20. Error metrics for 10th-day forecast on 26th Feb 2020

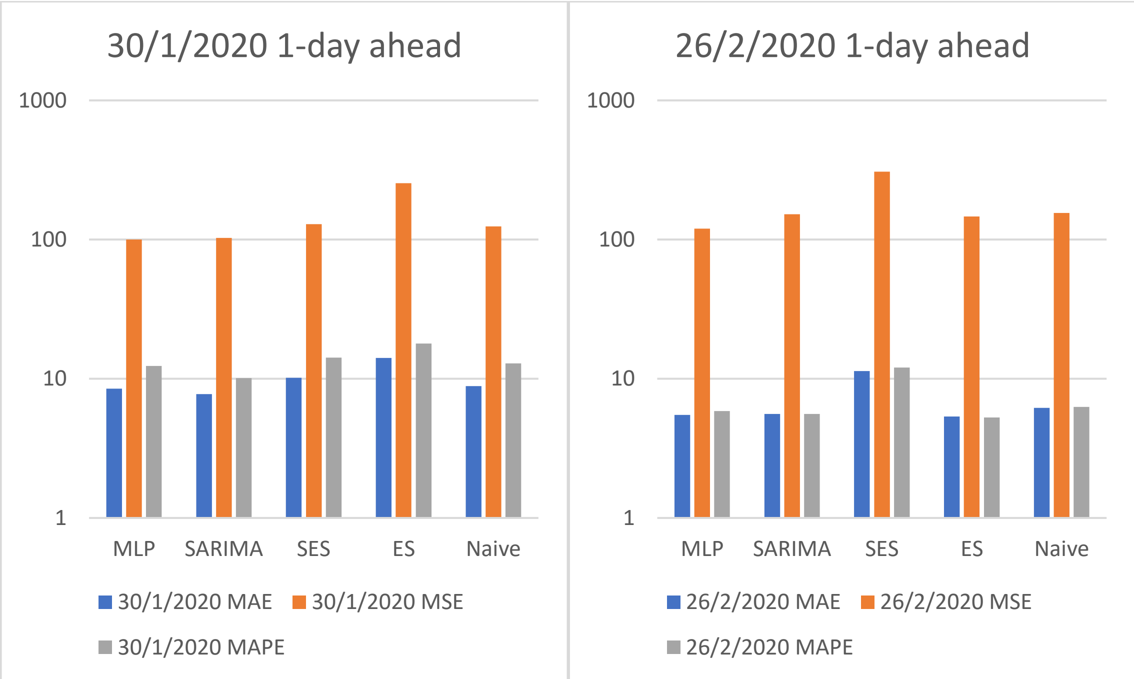


Figure 14. Error metrics comparison for 1-day forecast

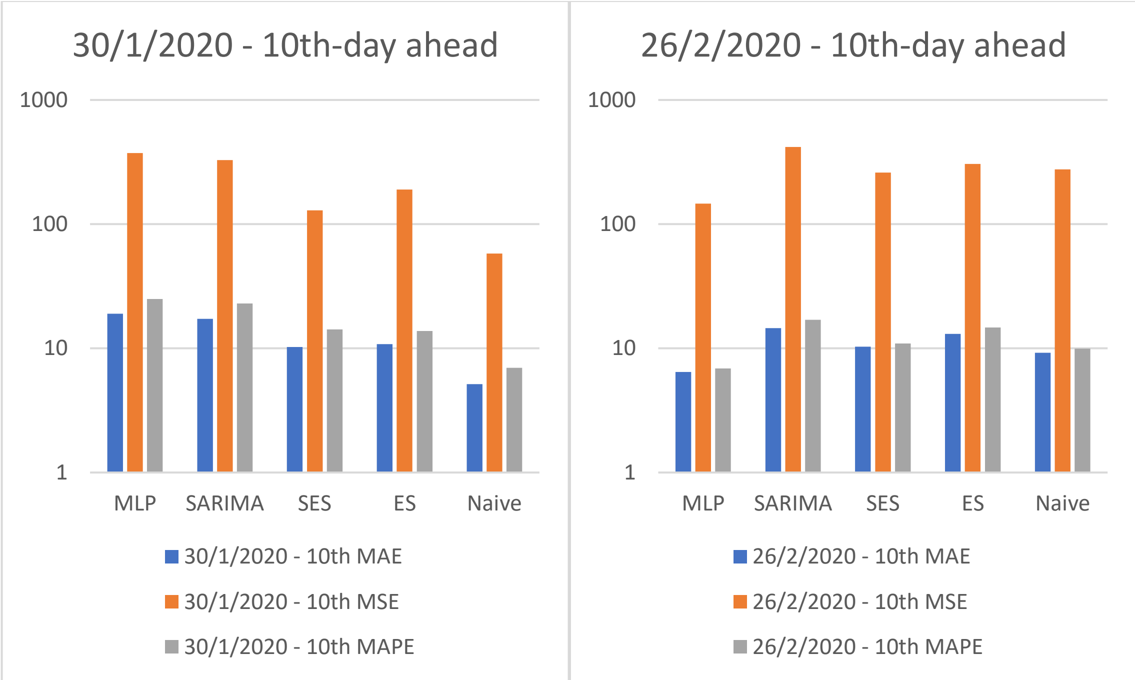


Figure 15. Error metrics comparison for 10th-day forecast

Comparing the error metrics across all the methods, the MLP and SARIMA models do well for the 1-day forecast as compared to the other statistical methods. However, for the 10-th day forecast, it might be due to the walk-forward prediction in those 2 models, where predications become worse and worse over more time steps, their error increased at a higher rate as compared to the other statistical methods.

3.6.1.8. Comparing runtime

	MLP	SARIMA	SES	ES	Naïve
Avg. Time (s) 1-day	0.257	98.101	0.0308	0.0620	0.00447
Avg. Time (s) 10th-day	1.879	94.427	0.0141	0.0633	0.0110

Table 21. Runtime for each method

The time take to gather the forecast is also an important factor in our research. For the simple statistics method, their runtime is generally very fast with all of them achieving their forecasts in less than 0.1s. The MLP model performs 2nd best with the 1-day forecast predicted in an average of 0.3s while the 10^{th}-day forecast took a little longer at 1.9s. lastly, the SARIMA suffered greatly in its runtime performance due to having to retrain its model every time new data is inserted.

4. Conclusion

The MLP proves to be effective in predicting future values with seasonal effects and slightly better than the SARIMA model. This was shown by the neural network ability to map the weekly, daily and hourly pattern of the historical WEP onto the prediction of the next day prices.

However, due to the unpredictable nature of supplying and distributing electricity in Singapore, where generator and transmission infrastructure might fail suddenly, our model cannot accurately forecast the large spikes in price we have seen in our data. We could only try to remove these outliers and predict for the best-case scenario where everything is working as per normal.

Furthermore, for real-time usage, a neural network approach might be better as the SARIMA requires the retraining of the model whenever there is new data. Computation time will be faster for the neural network and information can be displayed earlier.

References

[1] E. M. Authority, “Singapore's Electricity Market: Market Overview,” EMA, 2019. [Online]. Available: https://www.openelectricitymarket.sg/about/market-overview.

[2] S. K. Aggarwal, L. M. Saini and A. Kumar, “Electricity price forecasting in deregulated markets: A review and evaluation,” *International Journal of Electrical Power & Energy Systems,* vol. 31, no. 1, pp. 13-22, 2009.

[3] F. Gao, X. Guan, X.-R. Cao and A. Papalexopoulos, “Forecasting power market clearing price and quantity using a neural network method,” *2000 Power Engineering Society Summer Meeting (Cat. No.00CH37134),* 2000.

[4] S. S. Z. L. H.Y. Yamin, “Adaptive short-term electricity price forecasting using artificial neural networks in the restructured power markets,” *International Journal of Electrical Power & Energy Systems,* vol. 26, no. 8, pp. 571-581, 2004.

[5] Q. S. G. B. Shrestha, “Analysis of electricity price in competitive markets a study in Singapore market,” *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century,* 2008.

[6] G. P. Z. Min Qi, “An investigation of model selection criteria for neural network time series forecasting,” *European Journal of Operational Research,* vol. 132, no. 3, pp. 666-680, 2001.

[7] I. Yilmaz and O. Kaynar, “Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils,” *Expert Systems with Applications,* vol. 38, no. 5, pp. 5958-5966, 2011.

[8] J. Brownlee, “How to Create an ARIMA Model for Time Series Forecasting in Python,” Machine Learning Mastery, 18 September 2019. [Online]. Available: https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/. [Accessed 18 March 2020].

[9] J. Contreras, R. Espinola, F. Nogales and A. Conejo, “ARIMA models to predict next-day electricity prices,” *IEEE Transactions on Power Systems ,* vol. 18, no. 3, pp. 1014 - 1020, 2003 .

[10] Z. Ismail and K. A. Mahpol, “SARIMA Model for Forecasting Malaysian Electricity Generated,” *MATEMATIKA,* vol. 21, pp. 143-152, 2005.

[11] J. Brownlee, “A Gentle Introduction to SARIMA for Time Series Forecasting in Python,” Machine Learning Mastery, 21 August 2019. [Online]. Available: https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/. [Accessed 18 March 2020].

[12] F. Chollet, “Getting started with the Keras Sequential model,” [Online]. Available: https://keras.io/getting-started/sequential-model-guide/.

[13] F. Chollet, “Keras Documentation,” [Online]. Available: https://keras.io/models/sequential/.

[14] T. G. Smith, “pmdarima: ARIMA estimators for Python,” alkaline-ml, 2017. [Online]. Available: http://alkaline-ml.com/pmdarima/index.html. [Accessed 18 March 2020].

5. Appendix

Date	MAE	MSE	MAPE
14-01-2020	5.758729528	63.07466827	9.658947159
30-01-2020	8.4894158617	100.202528963	12.3335378776
17-02-2020	5.453149796	68.44849557	6.195421877
26-02-2020	5.515156008	120.1595212	5.860455309

Date	MAE	MSE	MAPE
14-01-2020	6.846655946	67.80496302	8.999685003
30-01-2020	7.755643225	102.5797854	10.10523702
17-02-2020	15.42296568	339.9769855	18.91910757
26-02-2020	5.601297253	152.0623992	5.590852117

Date 10^{th}-day ahead	MAE	MSE	MAPE
30-01-2020	18.94647842	373.2158843	25.00027688
26-02-2020	6.449423688	145.9440121	6.866359137

Date 10^{th}-day ahead	MAE	MSE	MAPE
30-01-2020	17.28722707	327.7625866	22.97552777
26-02-2020	14.53030881	417.9716263	16.92362128