Team 8 DataX Final Report: Energy Optimization using ML

Team Members:Eric Lo, Lihui Ji, Zihui Xu, Dan Zhao, Jingtong Zhao

Mentor: Ashkan Yousefi

Background

Our project aims to support the Hornsdale Power Reserve to optimize their revenue in **power storage** and **power charging system**. The Hornsdale Power Reserve is a facility comprising of a Tesla 100 MW/129 MWh Powerpack system in South Australia [1]. Their main task is to provide large amount of dispatchable renewable energy and to stabilize the energy price in the electricity market.

Our project focuses on different aspects of helping the company in optimizing the revenue. Just as how constant grid frequency is fundamental to the functionality of a grid system, frequency regulation capacity is essential for energy storage monetization. In Hornsdale, energy regulation is accomplished by Tesla Powerpacks. These giant batteries are charged by wind farms and are able to keep grid frequency within nominal range by discharging energy to increase frequency and consuming energy to lower frequency in a faster and more economical way. The value of the throughput ratio implies the energy intensity of providing regulation: the higher the ratio, the higher the intensity. The energy intensity decides the amount of energy charge and discharge. Therefore, one of the aspects of our project is to predict throughput ratio to optimize the state of charge (SOC) of storage resources using machine learning methods.

The next aspect of our work focuses on energy price prediction. The Hornsdale Power Reserve has its business in the energy exchanging market, which requires charging and discharging electricity with the current price. Accurate prediction of the future electricity price is essential for making the best decision on when to charge and discharge. Therefore, the second goal of our team is to make accurate predictions on price and make energy transaction decisions accordingly.

Part One

In order to predict regulation throughput ratio, grid frequency and power reading from 15 wind farms in five-minute time intervals are prepared as potential features. Linear regression, LSTM and decision tree (CART) are tested.

Regression

Linear regression represents the linear relationship between independent features (power reading from 15 wind farms and grid frequency) and the dependent variable (regulation throughput ratio). However, the linear regression trained model has really low r^2 score, approximately 0.104, which is much lower than our expectation. The MSE for training dataset is 0.0449, and for testing dataset is 0.0665. Even though our MSE value seems really low, it is due to the fact the input values in the model was small after the standardization. As shown in the Fig 1, this model is able to capture the basic data patterns for both the training set and the test set, but the range of the predicted data is much lower than the actual one. Based on these results, we don't think the linear regression is a good fit for this dataset. The suspected reason is that the dataset we obtained has 19 features and using the simplest linear model is highly underfitting.

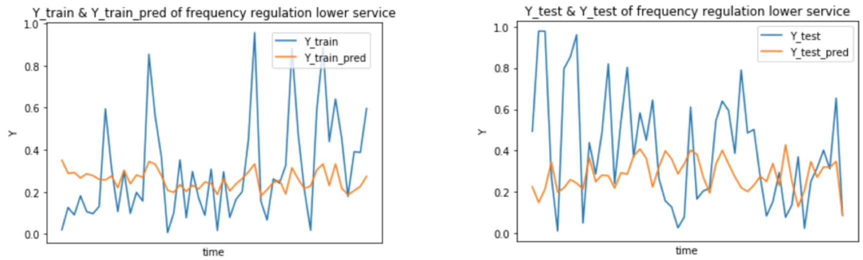


Fig. 1. Predicted value compared with the actual values for x_train (left) and x_test (right) in linear regression model

LSTM

LSTM with 4 units and sigmoid activation function is built with Keras (Tensorflow backend). The model is trained over 20 epochs with Adam optimizer. Figure 2 shows the predicted and real throughput ratio of lower and higher cases. Although the model successfully captures the trends of throughputs ratio, the magnitude of predicted and real values are different.

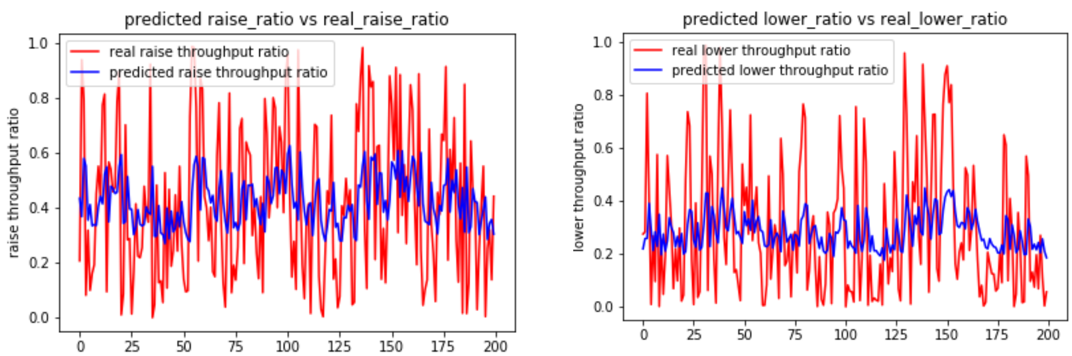


Fig. 2. Predicted value compared with the actual values of test lower/raise data with LSTM

CART

Two binary decision trees are constructed for frequency regulation lower and raise service[2][3]. The depth is 5 for both trees. The most important feature is 5-minute mean frequency for both models. The number of features which are used as the criterions of nodes is much less than the total number of features. So correlation between the unused features and throughput ratio is relatively weak. The training MSE and testing MSE of frequency regulation lower service are both in the range of 0.048-0.051 while those of frequency regulation raise service are both in the range of 0.070-0.071. As can be seen from Fig. 3, the predicted values tend to fluctuate much less than the actual value especially when extreme values appear.

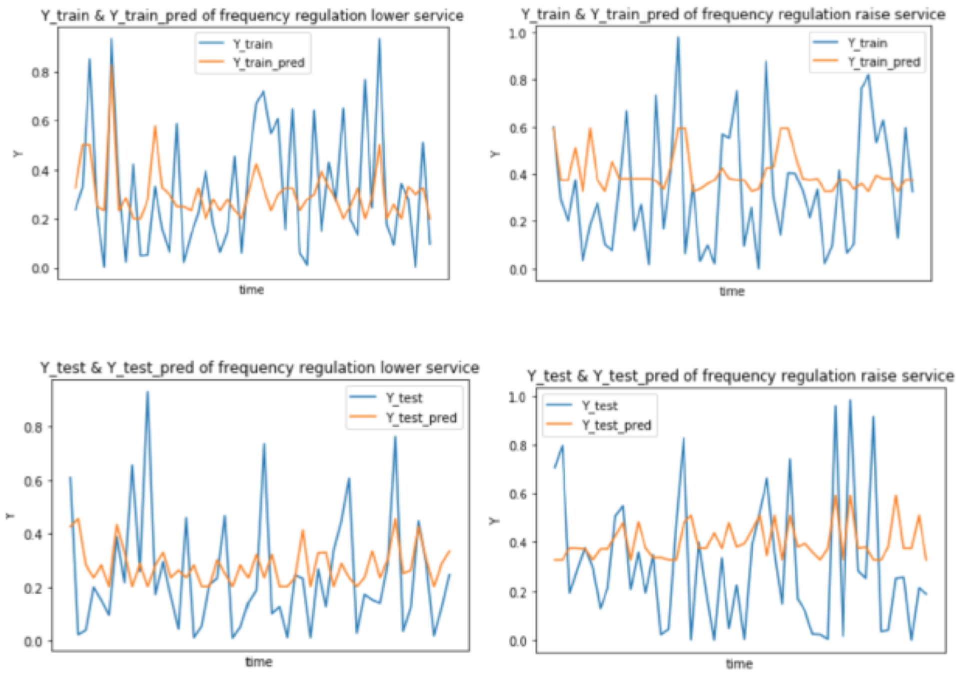


Fig. 3. Predicted value compared with the actual values for x_train (top) and x_test (bottom) in CART

To predict regulation throughput ratio for grid frequency, we built three models, linear regression, LSTM, and decision tree(CART). From the comparison of the predicted values and actual values for both the training and test data, we realize that all these models fail to capture the magnitude of the extreme values. After careful discussion with our mentor about these results, we concluded that there's no space to improve the models on the available dataset for future ratio prediction. Therefore, we decided to take another approach, which is to predict the electricity price and choose the charge and discharge volume accordingly.

Part Two

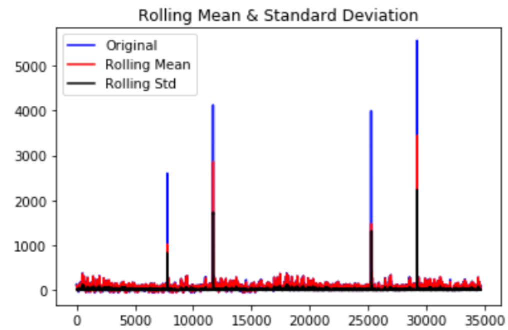
ARIMA

We used ARIMA to make the prediction about the price of the electricity. ARIMA stands for Auto Regressive Integrated Moving Average. ARIMA is a popular model in time series analysis, which has advantages in dealing with non-stationary data. ARIMA is basically as regression model, but with consideration of time series. It can be split into two parts, the AR part and the MA part. There are three parameters controlling the ARIMA model, that is *(p, d, q)*. The p parameter stands for the periods to lag, and d refers to the number of differencing transformations required by the time series to get stationary, where q denotes the lag of the error component.

The ARIMA model [4] is the combination of the AR model and the MA model, considering the difference, which can be seen in the following formula:

$$X_t(d) = \sum_{i=1}^{q} \beta_t \epsilon_{t-i} + \sum_{i=1}^{k} \alpha_t X_{t-i}(d) + \epsilon_i \, . \tag{1}$$

The ARIMA model only takes the price of the electricity in every half hour as the input. We did the data cleaning and fill all nan values with average of the price. For the first step, we conducted the Augmented Dickey-Fuller (ADF) statistical test to validate the stationarity of the dataset. ADF tests the null hypothesis that a unit root is present in time series sample. If accepted, it suggests the time series has a unit root, meaning it is non-stationary. The statistical P-value is 0.05. For our result, the P-value is 4.4689e-29, which is much lower than 0.05. So, we rejected the null hypothesis and accepted the alternative hypothesis that time series is stationary. The ADF statistic is a negative number (-16.1607) and more negative it is the stronger the rejection of the null hypothesis.




```
Results of Dickey-Fuller Test:
Test Statistic                 -1.616139e+01
p-value                         4.462000e-29
#Lags Used                      4.800000e+01
Number of Observations Used     3.455900e+04
Critical Value (1%)            -3.430539e+00
Critical Value (5%)            -2.861624e+00
Critical Value (10%)           -2.566815e+00
dtype: float64
```

Fig. 4. Results of Dickey-Fuller Test

Now that we know the time series is stationary,which means that differencing is not needed (d=0), the next step in fitting an ARIMA model is to determine whether AR or MA terms are needed to correct any autocorrelation that remains in the time series. We conducted a systematic way to achieve this goal is by looking at the autocorrelation function(ACF) and partial autocorrelation(PACF) plots of the time series. If the PACF of the series displays a sharp cutoff and/or the lag-1 autocorrelation is positive, an AR term should be added to the model and the lag at which the PACF cuts off is the indicated number of AR terms. If the ACF plot displays a sharp cutoff and/or the lag-1 autocorrelation is negative, then an MA term should be added into the model and the lag at which the ACF cuts off is the indicated number of MA terms. However, according to Fig. 5, the ACF and PACF both tail off as lag increases in this case, which indicates that we should include both AR and MA terms in our model. Since the plots of ACF and PACF are not typical plots and it is subjective to determine p and in this way, we use ACF and PACF as references and take another approach by applying BIC and AIC criteria to determine the best parameters of p and q.

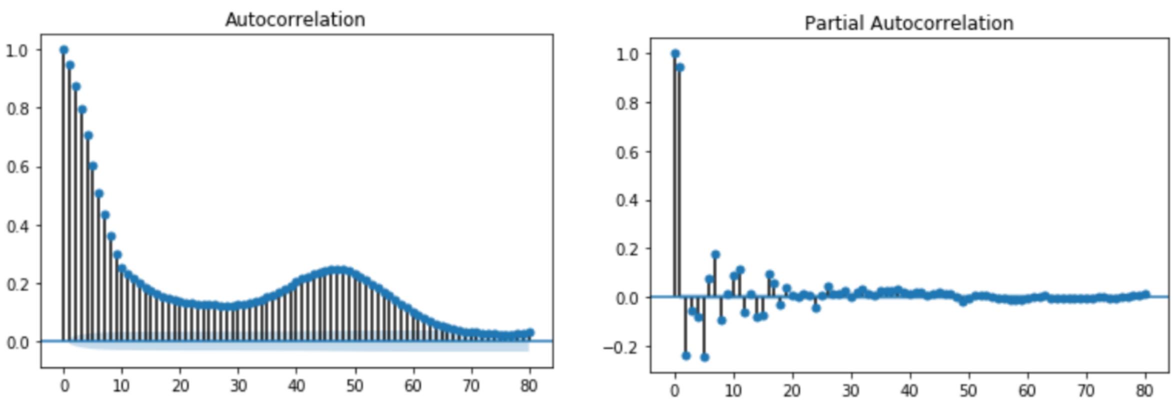


Fig.5. ACF and PACF of the time series

AIC and BIC are both penalized-likelihood criteria which are often used for comparing nonnested models. AIC is short for Akaike Information Criterion, which is an estimate of a constant plus the relative distance between the unknown true likelihood function of the data and the fitted likelihood function of the model, so that a lower AIC means a model is considered to be closer to the truth. BIC is an estimate of a function of the posterior probability of a model being true, under a certain Bayesian setup, so that a lower BIC means that a model is considered to be more likely to be the true model. By plot BIC and AIC with ranges of p and q, we are able to find that both BIC and AIC get the minimum values when p = 8, q = 4. As a result, we choose p = 8 and q = 4 as the parameters of our ARIMA model.

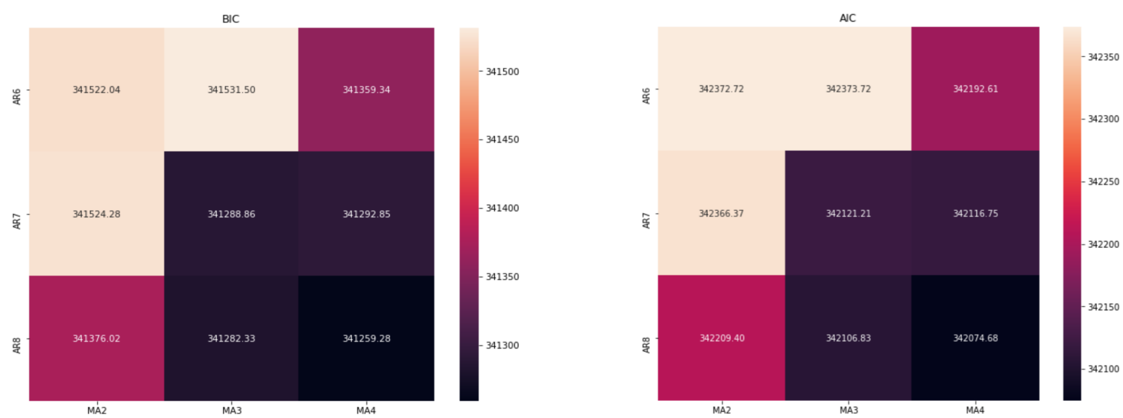


Fig.6. BIC and AIC

Result from ARIMA:

We trained model and make tested and predicted values on the same plot for comparison. The yellow line is the actual testing data, as for the blue line represents the prediction data. The model captures the general pattern of the data. The final results for the model has the mean absolute percentage error of 0.08. The forecast result highly aligned with the true result, but with little fluctuation.

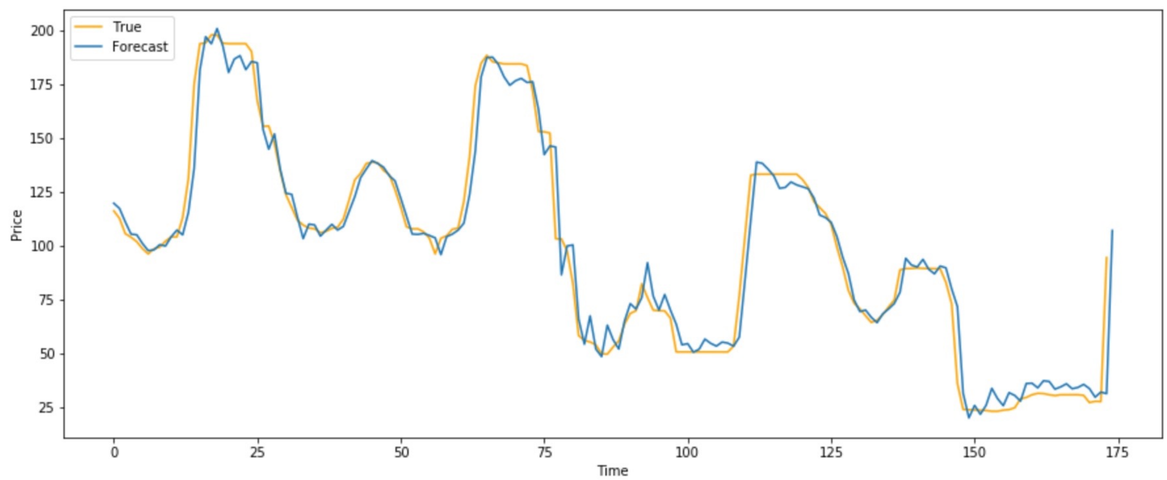


Figure 7: Result of rolling ARIMA performs a one-step forecast vs. true values

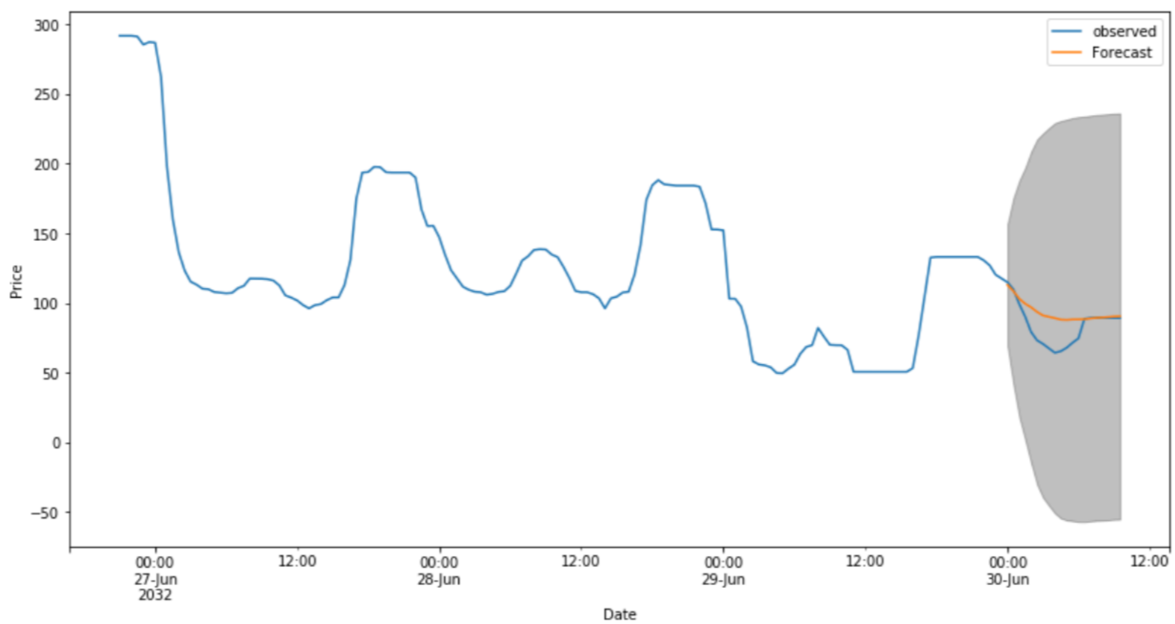


Figure 8: Predictions of 20 Timesteps by Using ARIMA with Confidence Interval of 80%

Based on the ARIMA model we trained, we predicted the price twenty time steps ahead. The blue line represents the actual price, and the yellow line represents the predicted price. There's 80% chance that the predicted price will be in the shading area. The predicted values fail to follow the pattern when there's a steep, but they've shown a great performance on the following timesteps when the price change is less rapid. The mean square error and mean absolute percentage error of twenty-timestep prediction is 188.253 and 0.139, respectively.

For better understand of the dataset, we introduced seasonal decomposition. The additive decomposition is the most appropriate for our case since the magnitude of the seasonal fluctuations, or the variation around the trend-cycle, does not vary with the level of the time series after plotted it. we can write

$$y_t = S_t + T_t + R_t$$

where yt is the data, St is the seasonal component, Tt is the trend-cycle component, and Rt is the remainder component, all at period t. The three components are shown separately in the bottom three panels of the figure 9.

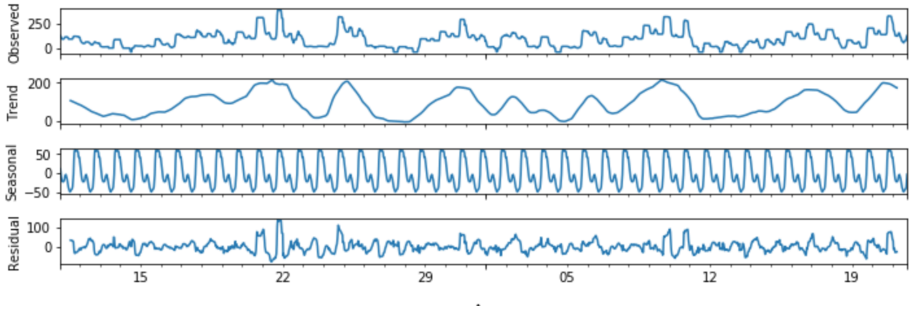


Figure 9: Seasonal decomposition of the data.

SARIMA

After applied seasonal decomposition, we did observe seasonal pattern involved in the dataset. So SARIMA is used to capture the seasonal trend. For the first step, we ran the AIC as before used in setting the arima parameters. We selected optimal parameters of p, d, q, and seasonal_p, seasonal_d, seasonal_q under the condition when the AIC value is lowest. Then, we did one step-ahead forecast. The line plot is showing the observed values compared to the rolling forecast predictions. Overall, our forecasts align with the true values very well.

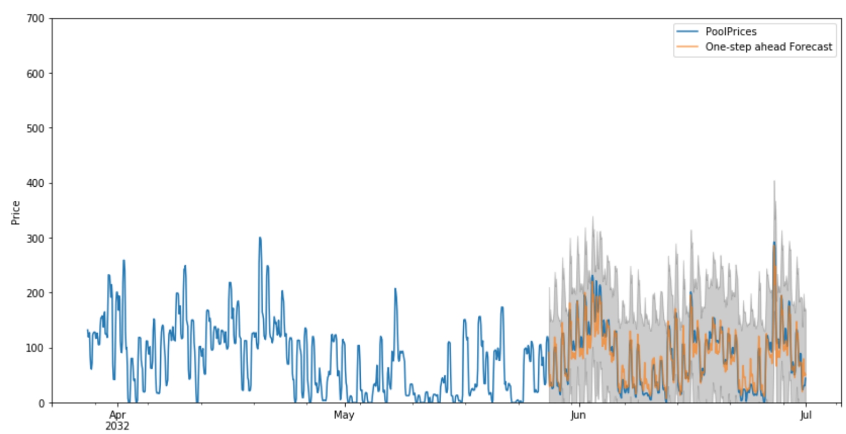


Figure 10: Sarima one step ahead forecast for the energy price

Since the model captured price seasonality, we used it to predict future unseen values in the 100 time steps, which is the forecasting two days' price in advance. From the figure 11, we find that the model forecast the seasonal pattern well. However, as we forecast further out into the future, it is natural for us to become less confident in our values. This is reflected by the confidence intervals generated by our model, which grow larger as we move further out into the future.

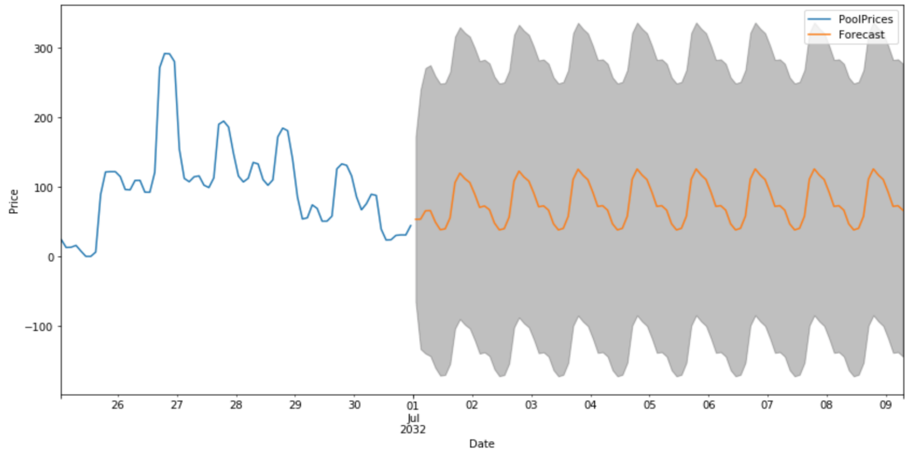


Figure 11: SARIMA forecasts the future price in 100 time steps

LSTM

Long short-term memory (LSTM) model is used to predict future electricity price.LSTM is a powerful artificial recurrent neural network (RNN) architecture in deep learning. As a major RNN, LSTM has "feedback" mechanism, which will improve model after each step. Moreover, LSTM has an outstanding performance in long time series prediction, which will avoid gradient explosion and vanish, a typical problem encountered in many other neural network architectures. Thanks to this feature, LSTM is capable to remember data long time ago. The project tries to evaluate LSTM performance in predicting future battery prices and charge/discharge actions[5].

I. Price Prediction

We took the advantage of Keras (Tensorflow as backend) to build neural network. The model is trained with Adam Optimizer and epoch of 50. The input data is an array of all prices 60 timesteps ago. The output is the following 10 timestep prices in the future. 10 timesteps are enough to take action based on the prediction (whether charge or discharge). Also, there will be suggestion on the actions by LSTM action prediction (next task).

The predicted and real prices (2000 data points plotted) are shown in Figure 12. The prediction complies well with the real prices, indicating satisfying performance of LSTM model. The error increases in further future. The model sometimes fails to predict long future's sudden price change. It is expectable because the model did not learn enough sudden change from training data. The model can be improved if we utilize longer-time data to train and the LSTM network. Overall, the model is satisfying and engineering-applicable.

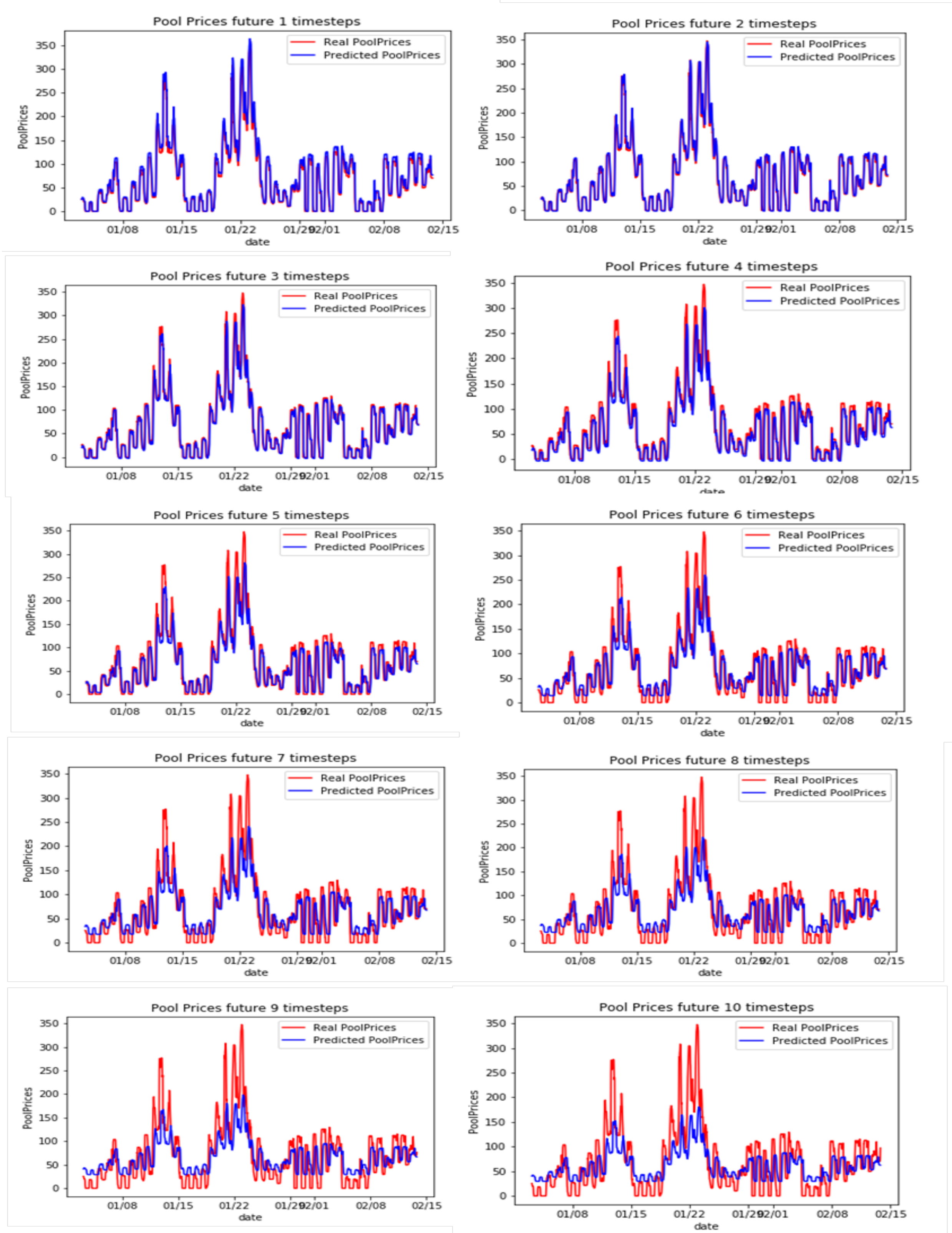


Figure 12: Ten-timestep-future predictions vs real prices

II. Action Prediction

The main purpose of predicting price is to guild company's action on whether to charge (purchase electricity) or discharge (sell electricity). LSTM is utilized here to predict actions (charge or discharge). We simplified the problem to a multi classification problem[J2]. There are three cases, charge, discharge and no action. Different layer numbers and different activation functions are tried. As a result, three layers are built in the model, with dense of 64, 32, 3 and activation function of relu, sigmoid and softmax, respectively. The input features are 20 timesteps' price, charge, discharge and action (i.e. 80 features total) because of limitation of computer performance. After training for 25 epoches, the model behaves well for the prediction as test accuracy is almost 1. The training and validation categorical accuracy is plotted below. Furthermore, we tried to predict the charge/discharge number, but the model did not perform well and probably due to lack of some important features (e.g. company economy status).

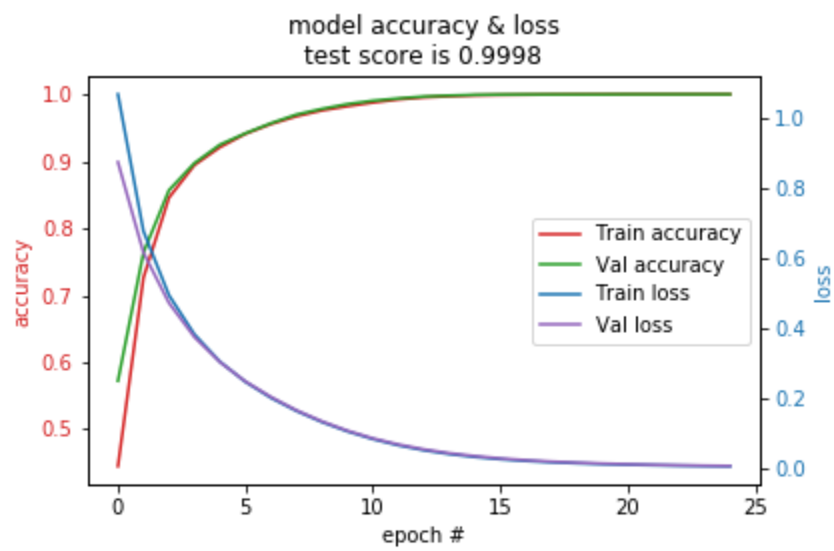


Figure 13: Model accuracy & loss along with epochs

Conclusion

In this project, we work on optimizing the energy revenue for Hornsdale Power Reserve from two aspects: prediction of throughput ratio and prediction of energy price. Among the three models we built for ratio prediction, LSTM achieves more convincing results than the other two regression models (linear and CART). LSTM has the lowest MSE for training and test data, and has the best performance on capturing the trend. To increase the prediction accuracy on throughput ratio, we'll keep improving the LSTM model with more input features, such as wind condition that directly effects on the electricity generation or apply reinforcement learning. We've also used ARIMA, SARIMA, and LSTM methods to predict energy price. Both Rolling ARIMA and SARIMA can predict well on the one step forecast but their performance is less promising when making predictions in the further future. The LSTM model is accurate on predicting price and charge actions. To further improve its performance, more features (e.g. economic status) and constraints (e.g. Battery capacity,charging efficiency) are needed. Overall, we think the prediction performance on both the throughput ratio and energy price can help the energy company make better decisions on state of charge and energy transactions.

Reference

[1] Hornsdale Power Reserve. (n.d.). Retrieved from https://hornsdalepowerreserve.com.au/

[2] Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and Regression Trees. CRC Press; 1984.

[3] Jason Brownlee. Classification And Regression Trees for Machine Learning. https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/

[4] Chenghao Liu. Online ARIMA Algorithms for Time Series Prediction. https://dl.acm.org/citation.cfm?id=3016160

[5]Morvan Zhou. What is LSTM in RNN ?. https://morvanzhou.github.io/tutorials/machine-learning/keras/2-4-B-LSTM/

[6]Eric Kim. Recurrent Neural Network to Predict Stock Prices

https://keras.io/getting-started/sequential-model-guide/