**Autonomous Battery Operation for Frequency Regulation**

**Using machine learning techniques**

Lihui Ji1, Eric Lo1, Zihui Xu1, Dan Zhao1, Jingtong Zhao1, Ashkan Yousefi2

1 University of California, Berkeley

2 Victoria University

**Abstract.** Tesla’s giant batteries Powerpacks become the spotlight in Australia’s electricity market with its ability to keep constant grid frequency in a quicker and more cost-effective way than traditional generators. In this paper, we study three models, LSTM, ARIMA and decision tree (CART) and test their performance on prediction of regulation throughput ratio, a key component in grid frequency regulation. Grid frequency and power reading from 15 wind farms in five-minute time intervals are prepared as potential features, and Mean Square Error on historical throughput ratios are compared analyzed.

**1 Introduction**

Just as how constant grid frequency is fundamental to the functionality of a grid system, frequency regulation capacity is essential for energy storage monetization. Grid frequency maintenance is achieved through two types of services: frequency regulation raise and frequency regulation lower. In Australia, these two services are accomplished by Tesla Powerpacks. These giant batteries are charged by wind farms and are able to keep grid frequency within nominal ranges by discharging energy to increase frequency and consuming energy to lower frequency in a faster and more economical way [1].

A critical economic component of frequency regulation is regulation throughput ratio. Taking both the amount of energy conversion and power market award into consideration, regulation throughput ratio is obtained through division of absolute value of average power by regulation capacity award. The value of the ratio implies the energy intensity of providing regulation: the higher the ratio, the higher the intensity. Therefore, optimization of the state of charge (SOC) of storage resources depends on the accurate prediction of regulation throughput ratio.

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. LSTM, ARIMA and decision tree (CART) are tested. By comparing and analyzing the accuracy of these three models on historical regulation throughput ratios, the most weighted features and the best performing model are selected to predict the future regulation throughput ratio.

**2 Methodology**

**2.1 Data Preparation**

The project utilizes pandas to read and clean data. The original data is mainly in three excel files, “frequency”, “wind\_generation” and “throughput ratio”. In wind farm dataset, the date is documented in time stamp, which is seconds elapse from January 1st, 1970. The date is converted to Australian Eastern Standard Time (AEST) to comply with other datasets. The frequency data is processed to generate average frequency, median frequency, cumulative sum of frequency and difference between the initial and last frequency in every five-minute interval. Then, the three treated datasets, with elimination of empty values, are combined to generate a comprehensive excel file for the project.

* 1. **Models**

**Long short-term memory (LSTM)  (In progress of working)**

Long short-term memory is an artificial recurrent neural network (RNN) architecture in deep learning. It can not only process single data points but also entire sequences of data, which means all the previous data can have reflection on the model. 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 exploding [2], a problem encountered in other neural network architectures. Thanks to this feature, LSTM is capable to remember data long time ago. LSTM is mainly used in language and image processing. The project tries to evaluate LSTM performance in predicting Tesla throughput ratio.

TensorFlow is utilized to build the LSTM architecture [3]. The project uses the power generation data from all 15 wind farms and four types of frequency (i.e. mean frequency, median frequency, cumulative frequency, frequency difference) as features to predict throughput ratio. The data is spited to training set (70%), validation set (15%) and testing set (15%). Linear map is used to prepare data to make it suitable for existing code. As the services between raise throughput ratio and lower throughput ratio are different from each other, the machine learning of throughput ratio is processed separately. 2 Layer LSTM is built with hidden layer sizes of 128 and 64, respectively. The model is trained over 20 epochs with embedding size of 256. The initial learning rate is set to be 0.1, though the Optimizer in the model will adapt it over time. Mean square error (MSE) is calculated to evaluate model’s performance.

**ARIMA**

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.

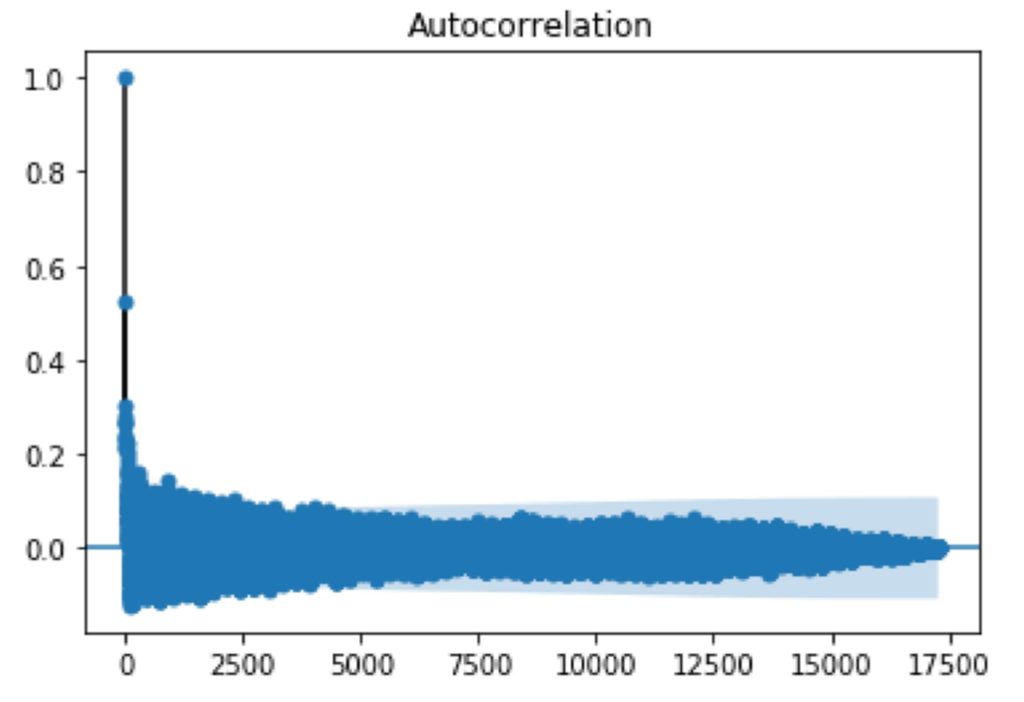
The ARIMA model is the combination of the AR model and the MA model, considering the difference, which can be seen in the following formula:

. (1)

Where represents the degree of difference, being the parameter of the MA model, and being the parameter of the AR model. and are the coefficient of the basic time series model, and is the observation at time [4].

ARIMA takes the input of only the y, and forms a model with the parameters (p, d, q) that is stated above, and adjusted via cross validation further in the analysis. Since the ARIMA model takes only the y as the input, the input data for this analysis is the throughput ratio for every five minutes from the dataset. The time format is initially “1/1/18” with a string type, which stands for 2018/1/1. The study thus changes it into “2018/1/1” into a date time format. The ratio for each time period is alternated in a logic that if it’s a “raise” the value would be a positive, if it’s a “lower”, it would be a negative. It both “raise” and “lower” are both “nan”, the ratio value would be 0.

A second step for the ARIMA model is to check whether the data is stationary. By importing a python library “statsmodels.graphics.tsaplots” import “plot\_acf”, a plot can be drawn to show whether the data is stationary or not[5].



**Fig. 1.** Autocorrelation plot for data shows that the data is stationary and any additional trick is not needed to make the data stationary, the study would go straight to the part to fit the model and identify the parameters.

**CART**

CART stands for Classification and Regression Trees, a decision tree algorithm that can be used for classification or regression predictive modeling problems. The models are obtained by recursive binary partitioning which divides the data space into two subsets so that every non-leaf node generated by CART algorithm is a simple-structured binary tree [6]. There are mainly two steps in the CART algorithm: the binary tree construction by binary recursive partitioning and pruning with verified data.

The entire dataset, including throughput ratio and 19 features, 15 from wind generation and 4 from frequency, is split to training set (70%) and testing set (30%). Since CART is not significantly impacted by outliers in the input features, the outliers are not removed when the models are constructed. Since frequency regulation lower and raise are two independent services, two models are constructed separately.

Creating a CART model involves selecting input features and split points on those feature until a suitable tree is constructed. This process is achieved by using the greedy algorithm to minimize a cost function. For classification trees, the Gini index function is used as the cost function to evaluate division points of features. For regression trees, the cost function is the mean squared error(MSE). In this project, since all the features are continuous values, MSE is chosen as the cost function. In other words, as can be seen from the formula below, by minimizing MSE or sum squared error[7], the CART algorithm will go through all the input features to find the optimal partitioning feature j and the optimal partitioning point s.

. (2)

. (3)

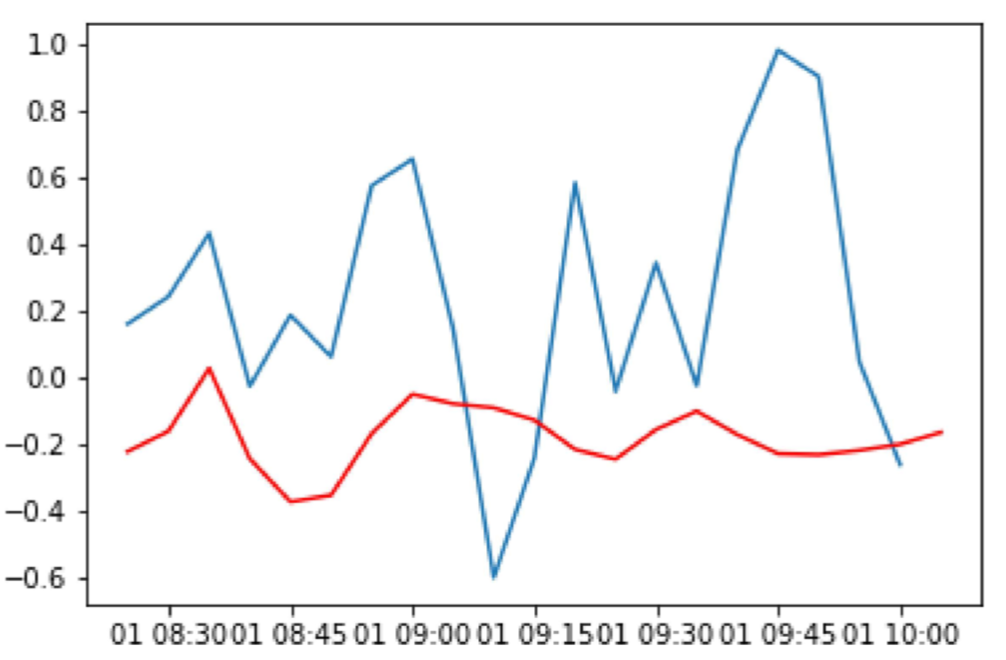
Stopping criterion is needed to stop recursive binary partitioning. Otherwise, the model will work its way down the tree with the training data and thus overfit the training data. Common stopping criterions include max\_depth, min\_samples\_split, min\_samples\_leaf. GridSearchCV is utilized to enumerate all the values in the list provided to build the model, compute the score of the training model multiple times and finally return the best parameters. The models for lower and raise frequency regulation services are constructed with the best parameters so that overfitting the training data is avoided and better performance is more likely to be obtained. For example, the max\_depth is 4 for both models.

**3 Discussion**

This sections compares and analyzes the performance of throughput ratio predicted on ARIMA and decision tree.

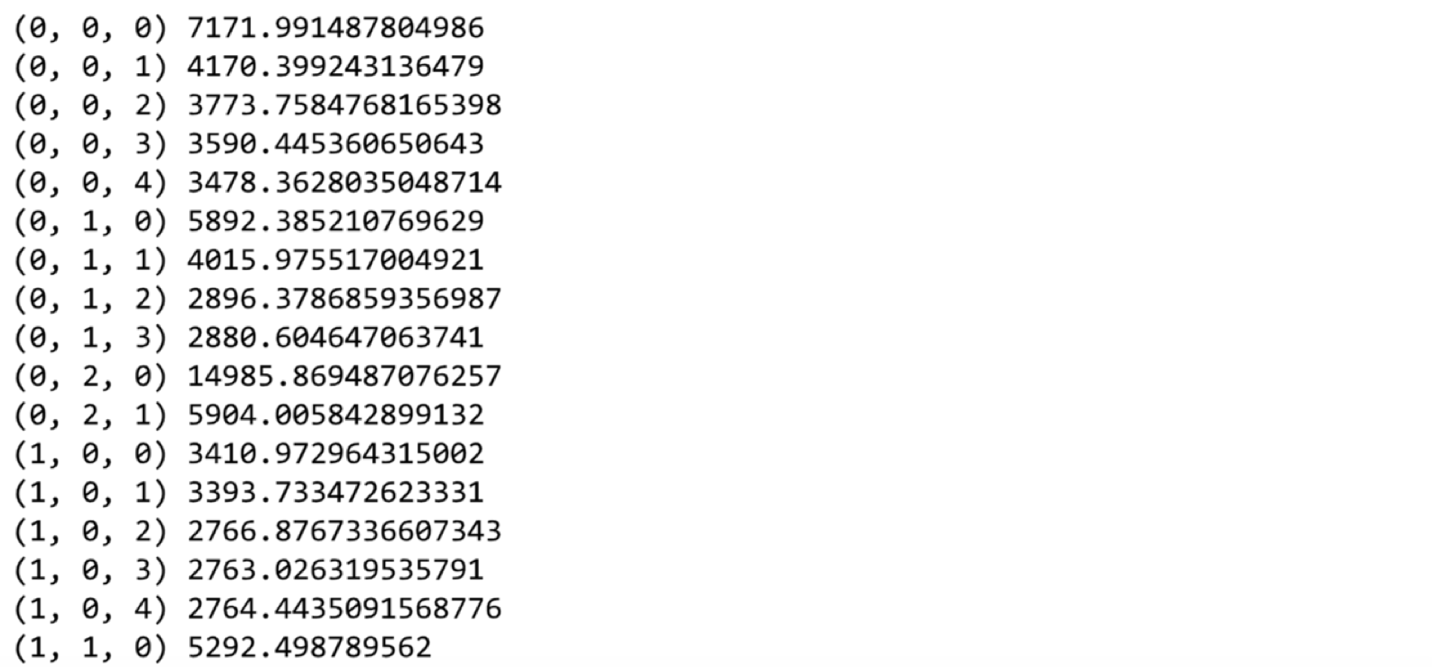
* 1. **ARIMA**

A small sample of 100 training data and 20 testing data was conducted to briefly illustrate how the model did.



**Fig. 3.** Small sample model testing

The blue line is the actual testing data, as for the red line represents the prediction data. Although the value doesn’t seem to be accurate, the shape does seem to be an okay fit. The next step is to decide the parameters in the model. In the ARIMA model introduced above, the three parameters that decides how the model performs are (p, q, d), by using cross validation testing on all (p, q, d) values from (0, 4), so there will be 5\*3\*5 combination of how the model perform.



**Fig. 4.** Cross validation of parameters

Fig. 4 is a simple demonstration of how the cross validating the parameters work, the measurement on how the model works with different parameters are by using the “model.aic” value, in general, the smaller the value is, the better the model performs. In the end, the study used the value (2, 0, 2) to train the model.

The final results for the model has the mean square error of 0.08, which will then be compared with other models. The code of the analysis is in the appendix as well so as the predictions.

* 1. **CART**



**Fig. 5.** Binary Decision Tree for Frequency Regulation Lower Service



**Fig. 6.** Binary Decision Tree for Frequency Regulation Raise Service

Two binary decision trees are constructed for frequency regulation lower and raise service. The nodes and structures of both trees are shown in Fig. 5 and Fig.6. The depth is 4 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.

**Table 1.** Train and Test MSEs of models

|  |  |  |
| --- | --- | --- |
|  | Frequency Regulation Lower Service | Frequency Regulation Raise Service |
| Train MSE | 0.049849143 | 0.070620093 |
| Test MSE | 0.049638048 | 0.072336448 |

The training MSE and testing MSE of models of frequency regulation lower and raise service are shown in Table. 1. With parameters selected with GridSearchCV, the training MSE and testing MSE is quite close in both models, which indicates that overfitting is avoided by choosing the appropriate stop criterion. The training MSE and testing MSE of frequency regulation lower service are both nearly 0.050 while those of frequency regulation raise service are 0.071 and 0.072.

References

[1] Megan Guess. Tesla’s new battery in Belgium shows value is in dispatch speed <https://arstechnica.com/information-technology/2018/05/teslas-new-battery-in-belgium-shows-value-is-in-dispatch-speed/>

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

[3]Garrett Hoffman. Introduction to LSTMs with TensorFlow. [*https://www.oreilly.com/ideas/introduction-to-lstms-with-tensorflow*](https://www.oreilly.com/ideas/introduction-to-lstms-with-tensorflow)

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

[5]  Brockwell, P. J.; Davis, R. A. (2009). Time Series: Theory and Methods (2nd ed.).  New York: Springer. p. 273. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [9781441903198](https://en.wikipedia.org/wiki/Special:BookSources/9781441903198).

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

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