Solar World Congress 2025

LONG SHORT TERM MEMORY RECURRENT NEURAL NETWORK AND STATISTICAL TIME SERIES ANALYSIS FORECAST MODELS FOR WIND FARM POWER OUTPUT AND ELECTRICITY PRICE

Summary

There is a plethora of modelling methods for forecasting weather and renewable energy generation utilising both statistical time series analysis and artificial neural networks. Autoregressive Moving Average (ARMA) and Long Short-Term Memory (LSTM) neural networks are both versatile and easy to train. In this paper we construct one step ahead forecast models for wind farm output and regional electricity price data using both traditional time series methods and hyper parameter tuned LSTM models to compare and contrast both the model performance and the modelling effort.

Keywords: Wind farm, electricity price, forecasting, ARMA, LSTM

1. Introduction

We use data publicly available data from the Australian Energy Market Operator (AEMO) for the National Electricity Market (NEM) spanning 7 years for the 30 minute average power output of a single wind farm situated at Cathedral Rock, South Australia for the years 2011-2018, starting at the time where the wind farm was initially commissioned. A separate list of 30 minute closing prices for the region of South Australia over the six year period starting in 2015 and ending in 2020 was used.

The focus is on wind farm output and price data series as wind farm power output is typically easier to forecast than solar farm power output. Solar farm power output is more complicated as it requires separate modelling of seasonality effects, whereas wind farm power output usually does not. This complexity allows a modelling effort comparison where only the autoregressive effects need to be considered. Due to the range of closing prices from -$1,000 to $14,700 per MWh, forecasting these price signals is more complicated and somewhat dependent on the demand or lack of renewable generation. In this analysis though, we simply focused on the price series itself. Most studies in the literature concentrate on trying to forecast the price since the other variables are often not easily available.

2. Methods

The data was split into training and testing sets for both variables, with the testing set being the last two years in each case. The modelling methods involve the use of statistical times series modelling methods as compared to using both Tensor Flow (Abadi et al. 2016) and PyTorch (Paszke et al. 2019) python based libraries to develop and test LSTM models. Power spectrum analysis showed no significant seasonality in the wind farm output data. The forecast models are constructed for the wind farm output itself. Given the wide range of possible prices, and the fact that the measure of central tendency, the median, is two orders of magnitude lower than the maximum, the forecast model is determined for the logarithm of price (suitably translated to avoid negatives).

The most parsimonious ARMA model was an AR(3), three lag autoregressive model, as in Equation 1.

$$x_t = \alpha_0 + \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \alpha_3 x_{t-3} \qquad (eq. 1)$$

For the transformed price data, the most appropriate model was determined to be an ARMA(1,1) as in Equation 2.

$$p_t = \beta_0 + \beta_1 p_{t-1} - \gamma_1 z_{t-1} \qquad (eq.2)$$

With the Tensor Flow libraries, the set of hyper-parameters was tuned manually through an exhaustive search of fixed sets of hyper parameter values to identify a combination that minimises the Mean Bias Error (MBE) in the first instance. The PyTorch library was investigated as it was the basis for the Dar(ts) (Herzen 40 et al., 2022) time series modelling framework allowing for the use of the Optuna (Akiba et al., 2019) automated hyper-parameter tuning Application Programming Interface (API).

To evaluate the skill of forecasting, the error metrics mean bias error (MBE), mean absolute error (MAE) and root mean square error (RMSE) and their normalized versions were used. The normalizing factor in the case of the wind farm is the mean output. Some use the capacity as the factor, but the mean is probably more realistically indicative of the performance of the farm. In the case of electricity price, the difference between the mean and the maximum that it is more sensible to evaluate the metrics on the performance forecasting the logarithm of the price.

3. Results

Model	NMBE	NMAE	NRMSE
Wind Farm ARMA	-0.003	0.149	0.229
Wind Farm LSTM	0.0003	0.143	0.225
Price ARMA	0.002	0.025	0.220
Price LSTM	-0.0002	0.027	0.233

Tab. 1: Error metrics for comparing the two approaches to forecasting

Note that there were three variations tested for the LSTM forecasts, for each of the wind farm and price forecasts. For the wind farm, we tested Tensor Flow, Pytorch and Optima Tuning. For the Price data we used Tensor Flow, but with three different batch sizes. The results in Table 1 for LSTM are the best of the three options in each case. The results for the two methods are very similar, with LSTM marginally better for the wind farm and the reverse true for the electricity prices. The ARMA approach is much more parsimonious so can be considered superior from that perspective. More importantly, it has a very simple structure and its physical nature can be easily understood, addressing the need for model explainability.

4. Conclusions and Future Work

As stated in Section 3, LSTM and ARMA produce similar results for short term forecasting of both wind farm output and South Australian electricity prices. ARMA is more parsimonious and explainable. Future work will include examining other machine learning tools to see if there is an improvement for the short-term forecasting. There will also be work on forecasting output from solar farms, which may require a hybrid technique, capturing the seasonality with a structural model, and then using machine learning tools for the residual series, after removal of the seasonal component. One aspect of the electricity prices is that they appear in very specific bins, relatively low prices, medium ones and extreme ones. This may lend them to be modelled with Hidden Markov models and this will be investigated.

5. References

TensorFlow: A System for Large-Scale Machine Learning, 2016, Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, Google Brain, Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), November 2–4, 2016 • Savannah, GA, USA. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." arXiv. https://doi.org/10.48550/arXiv.1912.01703.

Herzen, Julien, Francesco Lässig, Samuele Giuliano Piazzetta, Thomas Neuer, Léo Tafti, Guillaume Raille, Tomas Van Pottelbergh, et al. 2022. "Darts: User-Friendly Modern Machine Learning for Time Series." Journal of Machine Learning Research 23 (124): 1–6. http://jmlr.org/papers/v23/21-1177.html

Akiba, Takuya, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. "Optuna: A Next-Generation Hyperparameter Optimization Framework." In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2623–31. KDD '19. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3292500.3330701