Advisiandigital

Lesson Learnt Report #2

Date: 30 April 2020

Project: Advisian Wind and Solar Forecasting for the NEM Project

This Project received funding from ARENA as part of ARENA's Advancing Renewables Program – Short Term Forecasting funding round.

Key Learnings

SCADA Integration

- Integration with Site SCADA systems requires specialist support from SCADA engineers to connect data into the Edge Gateway software. In some instances, depending on the site configurations it may require upgrade of the site hardware/software the project cost for the asset owner and the deployment lead time.

Wind Farm Forecasting Algorithm

- The power curve is non-linear with respect to wind speed so using a non-linear model is the obvious choice. However, when modelling the relationship of past/present wind power and future wind power, the choice is not so obvious and in forecasting often models that are linear in the lags and/or the past errors show good performance. We tested both linear and non-linear models and concluded that a non-linear model had superior accuracy.

- During periods where the wind farm is operating under constraint, the wind power cannot be used as an input to the model. As such we have developed two models, one using Wind Power and during non-constrained periods and the other which relies on external factors such as wind speed and direction. These two models run in parallel and the algorithm switches between models depending on whether a constraint is active. This is essential for delivering an accurate forecast for when constraints are lifted.

- Extra care is needed to ensure that the algorithm normalises time zones. AEMO uses a specific time zone for forecasting which was different from the SCADA system time zone and the Edge Gateway time zones. Trending of our early models found an hour offset between the AEMO power values and SCADA wind power values which took time to debug.

- Misalignment in Units of Measure used for input variables in the model vs those received from SCADA caused erroneous forecasts which needed to be debugged. (Eg kW vs MW). These problems were solved after monitoring the input and outputs of the model over different time periods (approximately 3 hours) and comparing them with recent SCADA data.

Wind Direction

- We've seen in our exploratory data analysis that the turbines don't shadow each other and are quite independent with wind direction having a negligible impact, though in literature wind direction is a factor. We believe this is due to the unique arrangement of the Waterloo wind

turbines which are aligned along a ridge line North to South with the predominating wind direction being West/East.

- In other wind farms where the turbines are clustered we would expect wind direction to be a more important feature to include within a model, ie the turbines are less independent.

Solar Camera

- Two scientific grade sky cameras have been sourced for the solar farms. The cameras were ordered with filters for use in daylight hours and this was tested in the UK prior to delivery. During commissioning we found that the images captured were over exposed due to the much higher brightness of the sky in Australia. As such we have had to order new lens filters which were significantly delayed due to COVID-19.

- A Raspberry Pi computer controls the camera and is housed in an enclosure beneath the camera. During commissioning it was observed that these were experiencing very high temperatures which would affect their long-term reliability and we have had to redesign the enclosure by adding vents and a new fan cooled power supply.

AEMO API

- We are using event driven process via the available Azure Event Hub to connect with the AEMO API successfully. The Edge Gateway is an Event Producer that publishes data events with "Tag" and "Value" elements. All "prediction" tag values are model output values and these are sent to the AEMO API. When using an Azure Event Hub, you must checkpoint the data during processing to remove the need to reprocess previous events. This is because the time taken to reprocess from the start of unexpired Events on the Event Hub exceeds the requirement of being able to send any new "prediction" event values to the AEMO API within the applicable 5 minute window. Events can have an expiry of up to 7 days and can number 100 events per minute, resulting in a worst case of needing to reprocess 1 million + events before receiving new (current) data.

- To achieve the required resilience and forecast submission frequency at least 2 processes (where a process is defined as receiving Event Hub messages and pushing these to the AEMO API) must be running concurrently with some form of locking process required to ensure only one instance is actively submitting data at any point in time.

- We found that the AEMO API documentation contained some technical errors and is also lacking business process information. The example below indicates that "BasicAuth" should be used as a custom header value in the POST, although later there is mention of "BASIC Authentication details" which would indicate the usual Authorization header with BASIC should be used.

  Likewise "Content-type" should be "Content-Type" (they specifically indicate that these are case sensitive).

2.5.1 Header

Table 1 Header attributes

HTTP header attributes are case sensitive.

Parameter	Description	Use
Description	API description	
Content-type	application/json The format of the attached request file (only required for POST methods).	
BasicAuth	Your AEMO username and password provided by your Participant Administrator (PA) encoded into a Base64 authorisation token. For help, see Authentication and authorisation on page 15.	Mandatory
Participant ID	The initiating Participant ID	Mandatory
Market	For example: NEM, GasBB, GSH.	Mandatory

The HTTP Basic authentication header takes the following format:

Authorization: Basic {Base64 hash of user:password}, for example:

```
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

- As requests to the AEMO API need to be signed, the certificate to be used must contain both the public and private key. We recommend AEMO provide an example of key contents using an example key tool to highlight the structure.
- AEMO API requires time zone to be specified, however we found that any time zone other than UTC +10 (AEST) was invalid.
- Forecast data submissions need to be timestamped with the time at the end of the forecast window. The windows are in 5 minutes increments (starting from the 0 minute / 0 second of the hour) and data must be submitted at least 5 minutes before the end of the window.
- When submitting a forecast to the AEMO API, it must be between 0 and the capacity of the entity; however, the maximum capacity is stored as an integer and always rounded down. For example, Waterloo Wind Farm has a rated capacity of 130.8 MW and therefore a max capacity forecast should not exceed 130 MW else if will be rejected by AEMO.

Model Performance Monitoring

- Self-forecasting is not a "set and forget" process - a project owner is required by both the site asset manager and forecast provider to ensure MSATS access remains valid, SCADA signals are correct, and to analyse performance for ongoing optimisation and improvement of the forecast.
- To support real time monitoring of the model performance a user interface is recommended to compare live AEMO SCADA MW data (MMS 5 minute data) from AEMO against the delivered forecasts and to calculate errors. If the self-forecasts fail to out-perform AWEFS they run the risk

of being suspended, so continuous Realtime performance monitoring is preferred. Advisian will be deploying a user interface based on the Arundo Marathon platform.

Implications for future projects

- Extra care is needed to ensure alignment of units of measure and time zone information used in forecasting models can self-monitor model performance.
- Deployment of the forecasting algorithm into operation for multiple end users will require development of a user interface which would allow end users to self-monitor the model performance.
- Depending on performance, the forecasting models may need to be periodically enhanced either through retraining or adjustments in the model code especially during the early phase post model deployment. This will require ad hoc technical support from the model developer over the first year of operation.
- Customers should be made aware of the effort needed for SCADA integration and plan for this as part of the implementation process.