THE UNIVERSITY OF MELBOURNE

Melbourne Energy Institute

Meridian Energy Mount Mercer wind power forecasting

Extended forecasting project report

Lead authors:

Dr. Claire Vincent, Prof. Michael Brear, Kevin Sing Hoi Yang (Research associate) and Erdi Gao (Research Associate)

Project team:

Dr. Claire Vincent, Prof. Michael Brear, Prof. Chris Manzie, Prof. James Bailey, Prof. Jason Monty, Dr. Grant Skidmore, Dr. Rachael Quill, Dr. Mathieu Pichault

16 September 2022

Table of contents

Table of acronyms

AEMO	Australian Market Energy Operator
XGBoost	Extreme Gradient Boosting
LightGBM	Light Gradient Boosting Machine
SCADA	Supervisory Control and Data Aquisition
PMU	Phasor Measurement Unit
NMI	Normalised Mutual Information
ML	Machine Learning
RMSE	Root Mean Square Error
MAE	Mean Absolute Error

Acknowledgement and disclaimer

Meridian Energy Australia has received support from ARENA for the Wind Forecasting Demonstration Project, as part of ARENA's Advancing Renewables Program.

1. Project summary

The purpose of this project was to perform forecasting of wind power generation on Meridian Energy's Mount Mercer wind farm. Previous projects (Naemi 2020, Pichault 2021, Pichault 2021) had not incorporated data from the individual wind turbines for prediction of power on the wind farm scale. In this project we utilise the individual wind turbine data and aim to forecast wind power generation on the wind farm level using statistical and physical approaches. We were interested in producing the best forecasts for the power output six minutes and ten seconds ahead in the future, reflecting the time frame required to submit forecasts to the Australian Energy Market Operator (AEMO).

The aim of the physical approaches to this forecasting problem was to identify relationships between different groups of wind turbines and then utilise these relationships to predict power generation. Initially, we performed clustering on the instantaneous power separated by wind direction and found that the clusters generated do not differ greatly depending on wind direction. This clustering was performed using hierarchical and k-means

algorithms with cluster sizes of 2, 4, 6, 8, 16, and 32 wind turbines. Images of some of these clusters are available in Appendix II. Clustering on instantaneous power did not allow us to capture the propagation of wind throughout the wind farm as there were no time dependencies. To determine what time scales were relevant, we calculated the cross-correlation of power between different pairings of wind turbines. By separating the data on wind direction and choosing groups of turbines that were aligned in the same direction, we observed local peaks in the cross-correlation for specific time lags. This provided some evidence for the propagation of wind through the wind farm that is visible in the data. For groups of turbines that were closer in distance, the cross-correlation peaks were higher compared with turbines further away from each other. Because the forecast required is for six minutes and ten seconds ahead in time, this period of time is much longer than the short time scales between close turbines where the relationship between power output is similar. We utilised the k-shape clustering algorithm which accounts for the time dependencies in a data stream when clustering. The k-shape clustering algorithm is able to do this by using a normalized version of the cross-correlation so that shapes of the time series can be considered during the comparison. This algorithm captured the dependence on wind direction throughout time. Future work on this project could generate features that reflect the variance of power in these clusters, and use these as features in a modelling approach.

We initially explored three different machine learning approaches using basic wind turbine features such as lagged wind power, speed and speed cubed (Betz's law[1]). The first approach split the turbines per cluster, and built a separate model for each turbine in that cluster. The second approach focussed on using all wind turbine features to predict the total wind farm power output. The third approach attempted to predict the cluster average power output for each cluster. With these basic features, the second and third approaches produced the best performance but with only a 2% improvement in root-mean-squared-error (RMSE) compared with the persistence model. We sought to improve this model by performing feature generation and also creating a balanced training and testing dataset. We found that features that captured the variability over the wind farm were useful as it may provide some information about changing wind conditions throughout the wind farm. This was implemented through features that capture the dispersion of power within and between different clusters of the wind farm. The initial train test split was temporal, with earlier time periods used as the training data set. As this project only utilised one year's worth of data, this produced unbalanced datasets because of wind condition dependencies on the particular season. We created models that utilised a temporal train test split as well as one that attempted to sample data from each different time period. These models were implemented using different algorithms such as linear regression, robust linear regression, random forest tree, XGBoost and lightGBM. These more advanced models improved performance by 5-6% compared with the persistence model. Future work could improve the balance between train and test data sets, incorporate more wind farm variability features, and add weightings to the loss function to correct for unbalanced datasets.

2. Data

2.1. Site configuration

The Mount Mercer wind farm consists of 64 Senvion MM92 doubly fed induction generator wind turbines, which have a 2.05 MW capacity each. The maximum capacity of the wind farm is 131.2 MW. There are also two meteorological (met) masts 7.5 km apart in the North-West and South-East corners of the farm that collect

[1] Here we cite to Betz's law as a reference to the power output scaling with the cube of velocity, which is obtained as part of Betz's non-confined actuator disk derivation

various meteorological data. The turbines are ordered in rows (see Figure 1) where each subsequent turbine is 400 m apart, and each row is approximately 700 m apart.

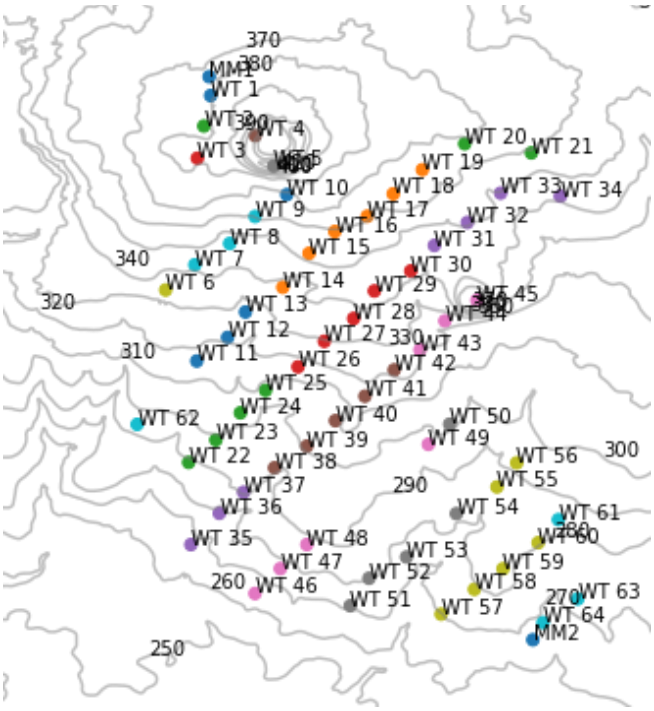


Figure 2.1: *Mount Mercer Wind Farm. Turbine codes utilised are reflective of Meridian Energy codes.*

2.2. Data dictionary

The data set utilised in this project ranges from the beginning of July 2019 until the end of June 2020. There are three distinct groups of data: met-mast data, wind turbine data, and total measured output data. The raw data can be found on the University of Melbourne Australian Renewable Energy Agency (ARENA) share drive 'uom_arena' in the 'DANIEL_DATA' folder. Refer to file *Data dictionary.docx* in the project folder *Internship* for more information.

2.3. Data pre-processing

2.3.1. Data logging process

The exact mechanisms of the SCADA data logging process is poorly understood. Based on the frequency of data logging, the upgrade of the SCADA system occurs at 7:20am on the 28th of November 2019. The raw data contains duplicate readings of the data stream. The previous value is logged again 1s before the value changes, these were removed from the data stream as a first stage of pre-processing. After removing duplicate values, we found that around 95% of the wind power and speed turbine data were logged at a frequency of 20 - 40 seconds (mostly at 30 second frequency) before the SCADA upgrade. Wind power and speed turbine data were logged at a frequency of 5 - 10 seconds after the SCADA upgrade. For the total power output data (PMU dataset), the frequency was 20 - 40 seconds before the SCADA upgrade, and at a 1 - 2 second frequency after the SCADA upgrade. Data logged in these consistent time frequencies consisted of about 95% of the total data set.

We expected a data threshold to be met for each subsequent value change (such as a 1% minimum relative change in value), but we could not identify any threshold or pattern when a new data value was recorded. The time difference between subsequent wind turbine readings was higher when the wind turbine power generation was negative (consumption of power), and this was the only noticeable pattern where the logging frequency was changed.

2.3.2. Pre-processing steps

Figure 2.3.2 demonstrates the pre-processing steps of the turbine data. A duplicate data value is defined as one that occurs one second before a new data value is recorded, and has the same value as the previous reading. The datastream is resampled at an interval of 10 seconds, when there are multiple readings in a 10 second interval these are averaged. There are quality flags associated with each data value, these can be due to the SCADA system ('Bad NonSpecific' labels), semi-dispatch capped generation values (where AEMO requires the wind farm to generate at lower than their maximum capacity), negative values as a result of no power generation, and not a number (NaN) values associated with resampling the data stream and the data frequency. The missing values in the data stream are linearly interpolated over a maximum time threshold of one minute.

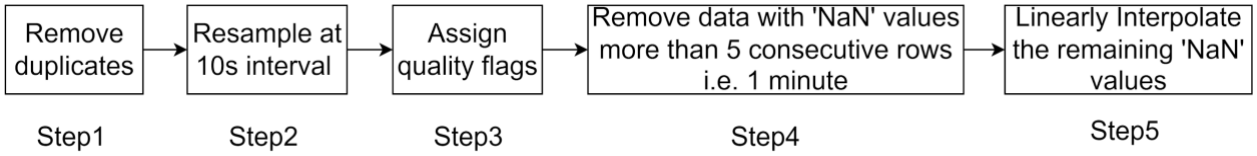


Figure 2.3.2: *Pre-processing Steps. The quality flags are used to indicate: 1. 'Bad NonSpecific' 2. 'Semi-dispatch' 3. 'Negative values' 4. 'NaN' values due to resampling/change of frequency*

The pre-processing of met mast wind speed and wind direction data follows the same process as demonstrated in Figure 2.3.2, with exception applied to long term data including temperature, humidity and air pressure. Since this data does not change as frequently as wind speed and direction, after assigning quality flags, all missing values were imputed by forward filling.

2.4. Exploratory data analysis

Data visualisation was performed throughout the internship. Here we provide a high level summary of our key findings, refer to the weekly report and previously published paper (Naemi 2020, Pichault 2021, Pichault 2021) for more information.

2.4.1. Wind speed distribution and persistence model errors

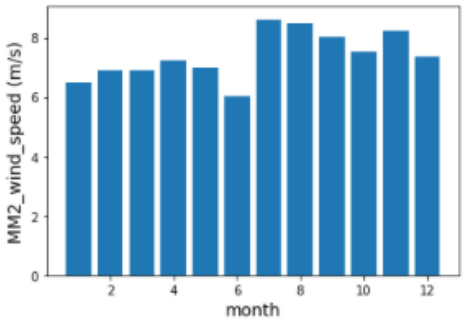
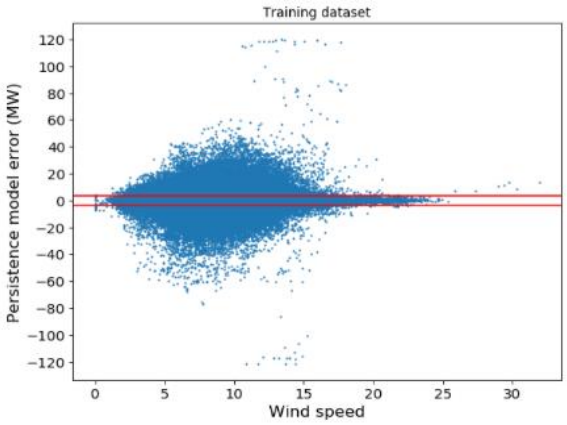


Figure 2.4.1(a). *Bar plot showing the average wind speed throughout the assessment year.*

It can be seen from the above bar graph that the average wind speed for the period July/2019 to December are higher than Jan/2019 to June/2020.

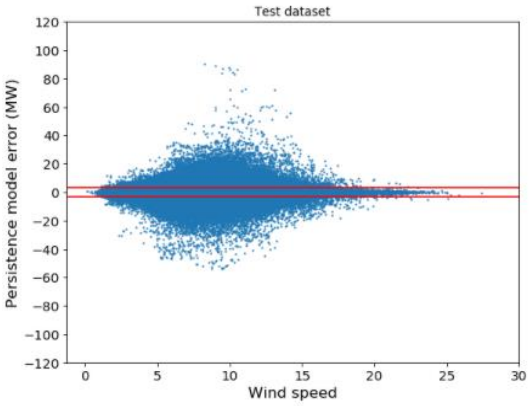



```
errorTrain['abs_diff'].describe()
```

```
count    484282.000000
mean          3.510618
std           4.531732
min           0.000000
25%           0.800181
50%           2.103749
75%           4.506086
max         121.691016
```

Figure 2.4.1 (b)

Scatter plot showing persistence model vs wind speed for the period July/2019 - Feb/2020, the red solid line indicating the persistence model MAE (±3.51 MW)




```
errorTest['abs_diff'].describe()
```

```
count    414099.000000
mean          2.949980
std           3.783595
min           0.000006
25%           0.668535
50%           1.726585
75%           3.829753
max          98.276865
```

Figure 2.4.1 (c)

Scatter plot showing persistence model vs wind speed for the period Mar/2020 - Jun/2020, the red solid line indicating the persistence model MAE (±2.95 MW)

2.4.2. Wind power, speed, and direction

Figure 2.4.2 is a polar scatterplot of wind farm power generation, speed, and direction over July 2019 - June 2020. We observe that when the farm operates at full capacity that the wind is coming from the Northerly or Westerly directions. The three most frequent wind directions are North, West, and South-East and average wind speeds seem to be at 6 - 7m/s. More detailed summaries of wind power, speed, and direction can be found in the weekly updates. A right skewed power distribution can be observed due to the higher frequency of purple scatters compared with the lower frequency of yellow scatters.

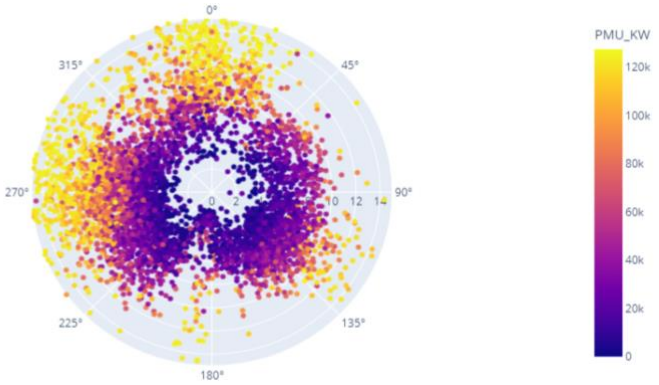


Figure 2.4.2: *Polar Scatterplot of Wind Power, Speed, and Direction, July 2019 - June 2020. Radius shows the wind speed, and the colour shows wind power generation.*

2.4.3. Outliers

Turbines with bad data: turbines with bad quality data are summarised in the table below.

Turbine id	Details
WT01	No data was recorded from 10th April 2020 to 30th April 2020
WT02	No data was recorded from 10th April 2020 to 30th April 2020
WT13	This turbine only had positive power output was only observed in Jul-2019, Aug-2019 and Nov-2019, it has negative power output in all the other months.
WT39	High percentage of 'Bad NonSpecific' data was recorded for the period from Mar-2020 to Jun-2020
WT45	High percentage of negative power output was recorded for Jan-2020 and Feb-2020
WT52	High percentage of negative power output in Apr-2020
WT64	High percentage of 'Bad NonSpecific' data was recorded for Sep-2019 and Oct-2019

Table 2.4.3: *summary of turbines with bad quality data. Note: the above summary of turbines with bad data might not be exhaustive.*

3. Physical analysis

3.1. Clustering power per wind direction

We applied K-means and Hierarchical clustering algorithms on the instantaneous power of each turbine with the power data stratified on wind direction. We expected that some physical clusters would emerge based on the wind propagation throughout the wind farm. The steps followed through this analysis were:

1. Split data based on wind direction using met mast 1 or met mast 2 wind direction data (e.g. four 90 degree bins or three different bins based on the modes of wind direction).
2. Utilise K-means algorithm or hierarchical clustering algorithm on the power of each turbine.
3. Use normalised mutual information (NMI) score to evaluate similarities between cluster labels.

We found that the cluster labels were highly similar to each other based on average NMI scores of 0.5 and above. Inspecting the clustering visually, we did not find strong evidence for a relationship between instantaneous power and wind direction. As this analysis did not account for time lags and therefore wind propagation through the form, we investigated lagged power values in 3.2 onwards.

3.2. Power cross-correlation analysis per direction

In this subsection we analyse the power cross correlation values for different groupings of turbines based on the direction of the wind. We are primarily interested in where the local maximum of the cross correlation occurs as this may indicate the time lag of wind propagation between two wind turbines. We investigate the cross correlation values using time lags, a dynamic measurement of distance, and the Strouhal number. The steps followed through this analysis were:

1. Create groupings of three turbines based on bearing between each other. As an example, if we were interested in groupings in the south-east direction, we can calculate the bearings between pairs of turbines and choose those that are 135 degrees (with some tolerance) from each other. A suitable grouping in this example would be wind turbine 12, 40, and 59.
2. Extract periods of time with continuous data streams for each turbine grouping based on the direction.
3. Compute power cross correlations for a number of different lags, distances, or Strouhal numbers.
4. Plot these cross correlations and identify if local peaks exist.

Ideally, identifying strong evidence of cross-correlation peaks will allow us to incorporate lagged power variables as features into statistical or machine learning models.

3.2.1. Time lags

From this analysis we observe that cross correlation peaks are higher for wind turbines that are closer to each other than those further from each other. The high peak in figure 3.2.1 (a) demonstrates the cross-correlations between wind turbines 1, 2, and 3 which are spaces at around 400m apart from each other. The cross-correlation between wind turbine pairs (1, 2) and (2, 3) also peak higher than (1, 3), which might be due to the larger distance between wind turbine 1 and 3. There is a smaller peak in cross correlations exhibited in figure 3.2.1 (b) where the turbines are spaced further apart. These plots demonstrate that using the lagged values from an upstream turbine may be a helpful predictor for a downstream turbine's power. Although closer turbines are more cross-correlated,

this might not be helpful for predicting 6 minutes and ten seconds into the future. In figure 3.2.1 (a), the optimal time lags occur at 70 seconds and 160 seconds which is much earlier than prediction 370 seconds into the future. The peaks for wind turbines further apart are smaller, but they are of similar time scales to the prediction timeframe for this task.

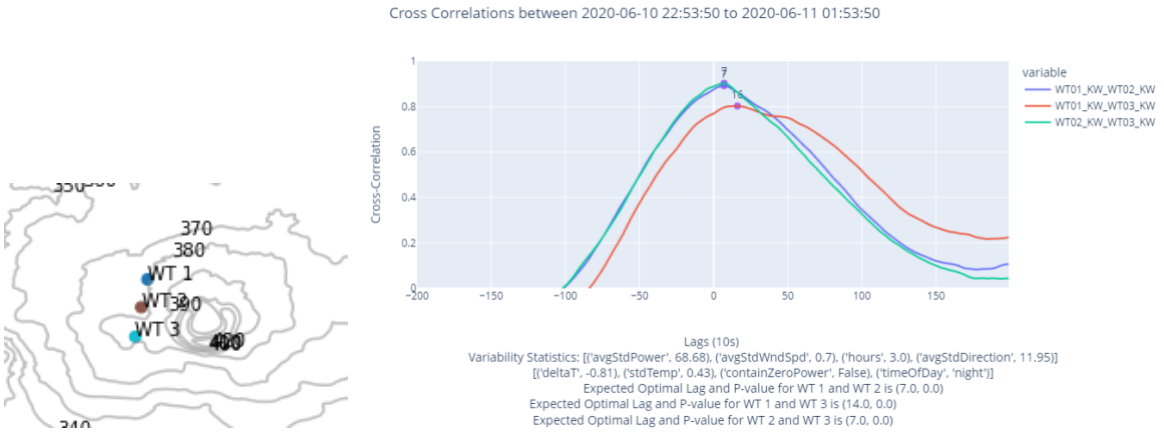


Figure 3.2.1 (a): *Power Cross-Correlations for Wind Turbines 1, 2, 3.*

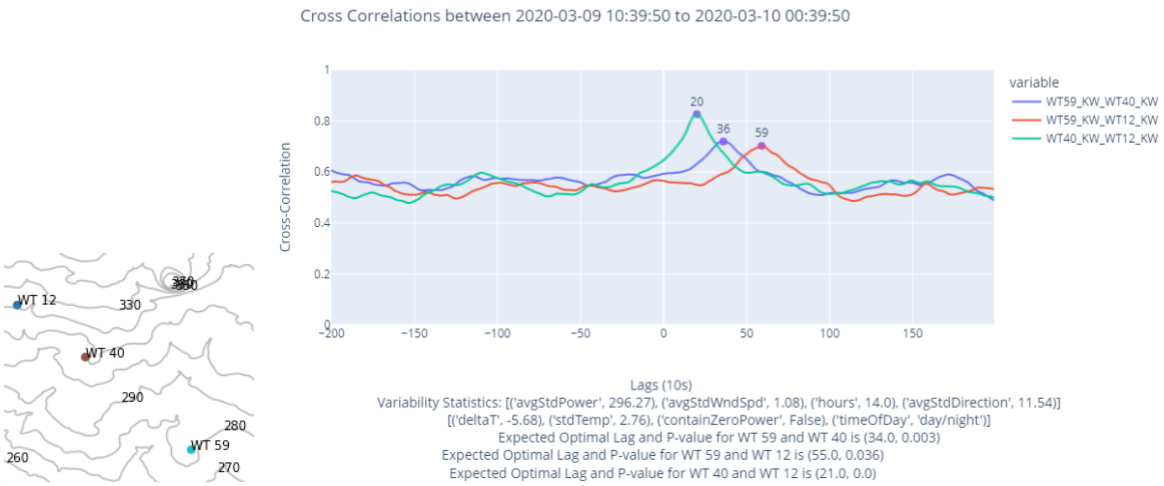


Figure 3.2.1 (b): *Power Cross-Correlations for Wind Turbines 59, 40, 12*

3.2.2. Distance

In the analysis in 3.2.1, the time lag values for the cross-correlations were static, and the data streams of different turbines were shifted by a fixed time period. In this analysis, we perform a dynamic matching of two turbine time series based on the instantaneous wind speed and the distance between the groups of turbines. This matches the future power reading value from a reference turbine based on the speed of the wind at that particular point in time, reflecting a wind vector propagating in a specific direction at that instantaneous speed. Instead of varying the time lags in 3.2.1, we vary the distance. We expect there to be a peak in the cross-correlation values when the

distance used to match the two time series is the actual distance between two turbines. The method for computing these cross-correlations includes the steps in 3.2, with these additional steps:

1. Utilise the wind speed time series to match future power values on the downstream turbine. Create a new time series for matched power values of the downstream turbine where the distance varies between -8000 and 8000. Given a specific distance, we calculate the time taken for the wind to travel using the instantaneous speed, this constitutes the lag for one power reading. Using this time, we match each upstream turbine power value to a downstream turbine power value.

Figure 3.2.2 shows the same time period and cross correlations as Figure 3.2.1 (b) but with distance now on the x-axis. The distance between wind turbine 40 and 59 is 2400m, and the highest cross-correlation occurs at 2500m.

3.2.3. Strouhal number

The Strouhal number is a dimensionless value and can also be used to investigate the cross correlations. Using consistent time lags Δt, the average wind speed u, and the length between two turbines L, we can compute the inverse Strouhal number $L/(u\,\Delta t)$ associated with each cross-correlation. If the wind is propagating between two turbines at the average wind speed, we shall observe a peak in the cross correlation at the inverse Strouhal number of 1. This is a pattern we observed when turbine groupings were consistent with the wind direction, and this did not appear when the wind came from other directions.

3.3. k-shape clustering

The k-shape clustering algorithm (tslearn.clustering.KShape) is a clustering algorithm applicable to time series data. It is based on matching time series based on the optimal lag that maximises the cross correlation, and minimising the shape-based distance (Paparrizos and Gravano 2017). As opposed to creating clusters based on instantaneous power as in 3.1, the k-Shape algorithm accounts for relationships between lagged time series. For this forecasting application, the algorithm runs in $O(m^3)$ where m is the length of the time series. This provides a constraint as running this algorithm for a long time series is computationally expensive. We computed different k-shape clusters for different time periods at different wind directions. The results from running this clustering algorithm can be found in the results folder. For each of these time periods, various statistics describing the environment were captured such as average power, wind speed, wind direction, and temperature. Comparing different pairs of cluster labels, we found that pairs that shared the same wind direction had significantly larger normalised mutual information (NMI) scores compared to the average NMI over all pairs of cluster labels.

3.4. Future works

1. Incorporate cluster information into modelling as features, create features from clusters based on variability between clusters. Based on findings in section 4, we expect the variability statistics across clusters to be an informative feature.
2. Perform individual wind turbine power prediction using multiple wind turbine's lagged information.
3. Dynamic feature engineering using different wind turbines based on wind direction at a particular time.

4. Machine learning approaches

4.1 Clustering and preliminary modelling using basic features

As advised during the weekly meeting, modelling was carried out based on the idea of clustering turbines into different groups. Different clustering methods were explored and it was found that clustering based on instantaneous power output resulted in turbines that were located closely together being allocated to the same group. Having obtained the clustering results, 3 different modelling scenarios were then developed and evaluated. In all 3 scenarios, a single wind farm total power output prediction was obtained and was used for model evaluation:

- Scenario 1: For each cluster, a model was built to predict the single turbine power in that cluster. The wind farm total prediction is then obtained by summing up individual turbine power output prediction.
- Scenario 2: One single model was built to predict the wind farm level total power output.
- Scenario 3: For each cluster, a model was built to predict the cluster total, i.e. n clusters would result in n models to predict n cluster power outputs. The wind farm total prediction is then obtained by summing up these n cluster power predictions.

Basic features i.e. power, wind speed and wind cubic were used to compare the performance of different modelling scenarios. Cluster average wind speed and cluster average wind speed cubic were used for scenario 2 and scenario 3.

Key findings:

- While scenario 1 seems to perform the worst (the 'average' features used for scenario 2 and scenario 3 may have helped to reduce individual turbine data fluctuations), no obvious differences were observed between scenario 2 and 3, and both models were only able to improve the RMSE from persistence model by 2%. Hence it was decided to use the simpler model (scenario 3) for further model development. It was also noted that adding the current met mast variables was not able to further improve the prediction.

4.2 Research question: How to beat the persistence model?

Having discovered that the above approaches could only beat the persistence model by a small amount, a natural question to ask was: how to outperform the persistence model?

Since the persistence model assumes that power output will remain constant, to beat the persistence model, a machine learning (ML) model would need to make accurate predictions during times when the power output is not constant. To facilitate this, it is important that the ML model has:

1. **Features that can detect change:** For the ML model to perform well during scenarios where power is changing, we need to find a set of features that are correlated with a change of power output in the forecasting horizon. One way to find such a set of features is to find features that could indicate 'change'. An example of such features could include measures of the level of difference in power output between different clusters which could indicate changing wind conditions across the wind farm.
2. **A balanced training set and test set:** The distribution of samples in the training data set should include a sufficient number of data points where power is changing significantly to enable the ML model to learn how to predict these events. Additionally, the distribution of such events in the test data set, should be similar to the overall data set, otherwise the metrics may suggest that the persistence model is performing better than is actually the case (e.g. if the test data set contains a smaller proportion of events where power is changing

significantly). Data exploration indicated that this may not be the case, although further work is needed to explore this, as noted in Section 4.5.

4.3 Methodology and results

4.3.1. Feature generation

The list of features generated are summarised in Table 4.3.1. In particular, features under categories 'inter-cluster variabilities' and 'Lag terms' are generated to address item 1 of Section 4.2 above.

Cluster based features	Met mast	Inter-cluster variabilities	Lag terms	Persistence
For each cluster 3 features are calculated: 1. Average power 2. Average wind 3. Average wind cubic	1. Met mast 1 (and 2) air pressure with 3 hour lead time 2. M1 direction cos() and sin() 3. M1 vertical temperature difference 4. M1 humidity, temperature, pressure, square of pressure difference between M1 and M2, square of humidity difference between M1 and M2	1. Variance 2. Difference between min and max 3. Sum of absolute difference between clusters (as a percentage of the total wind farm level power output) 4. Entropy 5. Min 6. Max 7. Median 8. Variation	1. Lag1_avg (total power) 2. Lag2_avg 3. Lag3_avg 4. Lag4_avg 5. Difference between min and max 6. Difference between max{lag1_avg, lag2_avg, lag3_avg, lag4_avg} and P_t 7. Difference between P_t and min{lag1_avg, lag2_avg, lag3_avg, lag4_avg}	Current power wind farm level

***Table 4.3.1** Feature list*

4.3.2 Training and test dataset split

Two different methods for splitting the training and test dataset were explored:

- Method 1: following previous work, the train / test split was done chronologically to avoid overfitting i.e. July-2019 to Feb-2020 was used for training and March-2020 to June-2020 was used for testing.
- Method 2: a preliminary attempt was made to address item 2 of Section 4.2 above, although further work is needed. The split method is illustrated in Figure 1 in Appendix C of Weekly_Progress_Report_21_02_2022.pdf

4.3.3 Machine learning algorithms

- Different machine learning algorithms were explored for the given dataset and features described above including linear regression, robust linear regression, random forest tree, XGBoost and lightGBM. From both training speed and prediction accuracy perspectives, lightGBM seems to have the best performance. It was found that an ensemble model could help further improve the prediction.

4.3.4 Results

	Train / test split based on method 1				Train / test split based on method 2			
No. of clusters	**Ensemble RMSE**	**LightGBM RMSE**	**Ensemble MAE**	**LightGBM MAE**	**Ensemble RMSE**	**LightGBM RMSE**	**Ensemble MAE**	**LightGBM MAE**
1	4.61 (3.87%)	4.61 (3.91%)	2.92 (0.91%)	2.92 (0.85%)	5.28(4.28%)	5.28 (4.34%)	3.14 (2.98%)	3.16 (2.59%)
2	4.59 (4.29%)	4.59 (4.27%)	2.92 (1.10%)	2.92 (0.92%)	5.24(5.03%)	5.24 (4.60%)	3.13 (3.51%)	3.15 (2.64%)
3	4.59 (4.37%)	4.59 (4.29%)	2.91 (1.24%)	2.93 (0.78%)	5.23 (5.18%)	5.23 (4.91%)	3.13 (3.45%)	3.14 (2.95%)
4	4.59 (4.36%)	4.59 (4.10%)	2.91 (1.49%)	2.93 (0.82%)	5.25 (4.89%)	5.25 (4.88%)	3.13 (3.37%)	3.14 (2.98%)
5	4.57 (4.73%)	4.57 (4.87%)	2.91 (1.24%)	2.92 (0.95%)	5.20(5.74%)	5.20 (5.45%)	3.11 (3.93%)	3.13 (3.42%)
6	4.57 (4.83%)	4.57 (4.77%)	2.91 (1.33%)	2.92 (0.96%)	5.21(5.60%)	5.21 (5.43%)	3.11 (3.99%)	3.13 (3.46%)
7	4.56 (5.02%)	4.56 (5.21%)	2.91 (1.33%)	2.92 (1.17%)	5.18 (6.11%)	5.18 (5.91%)	3.10 (4.35%)	3.12 (3.80%)
8	4.55 (5.20%)	4.55 (4.48%)	**2.90 (1.82%)**	2.93 (0.61%)	5.18 (6.21%)	5.18 (6.08%)	3.10 (4.38%)	3.11 (3.94%)
…	…				…			
18	4.53 (5.57%)	4.53 (4.81%)	2.90 (1.71%)	2.93 (0.59%)	5.14 (6.85%)	5.14 (6.54%)	3.08 (5.02%)	3.10 (4.30%)
…	…				…			

Table 4.3.4 *Results based on different train/test dataset split methods. Note turbines with bad quality data i.e. WT13, WT39, WT45, WT52 and WT64 were excluded from this analysis.*

Key findings:

- In the weekly progress report dated 14th Feb 2022, the ML model was able to achieve an RMSE 2% better than the persistence model. The above results (Method 1) show that the addition of features indicating change have further improved the RMSE, resulting in a 5.6% improvement when compared to the persistence model.
- Using Method 2, we see that the RMSE has also reduced significantly, with a 5.6% improvement when compared to the persistence model. The MAE achieved with Method 2 of 5.0% was also found to be better than what was achieved using Method 1 (1.8%).

4.5 Conclusions and future work

For forecast horizon considered herein, there were improvements that could be made from the persistence forecast; however, care had to be taken with the approach. For a naïve data split, only marginal improvements could be obtained from persistence. However, when we focused the machine learning data split such that it focused on learning when the power time series was changing, we could obtain significantly more accurate forecasts. We also found that 8 clusters was the modelling "sweet point" offering enough averaging to attenuate some forecasting error, whilst being precise enough to capture small fluctuations that might have been averaged over with fewer clusters.

Future work should consider:

1. Investigation of additional features that are correlated with significant power change in the forecasting horizon.
2. Alternative means of splitting the test and training data set to ensure the data sets are more similar and to reduce imbalance. While further analysis is required, our exploratory data analysis indicated that the training and test data sets contain significant differences which could be adversely impacting model training and testing - refer above to Section 2.4.1 for details.
3. Consideration of adding weights to the loss function as a means of correcting issues with the imbalanced training data set.

Appendix: clustering images

1. k-means based clustering

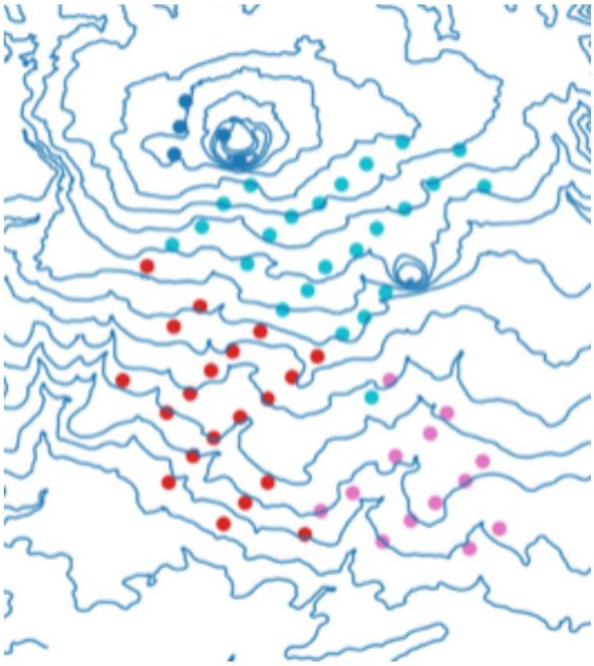


Figure II.1.1 *k-means clustering with 4 groups*

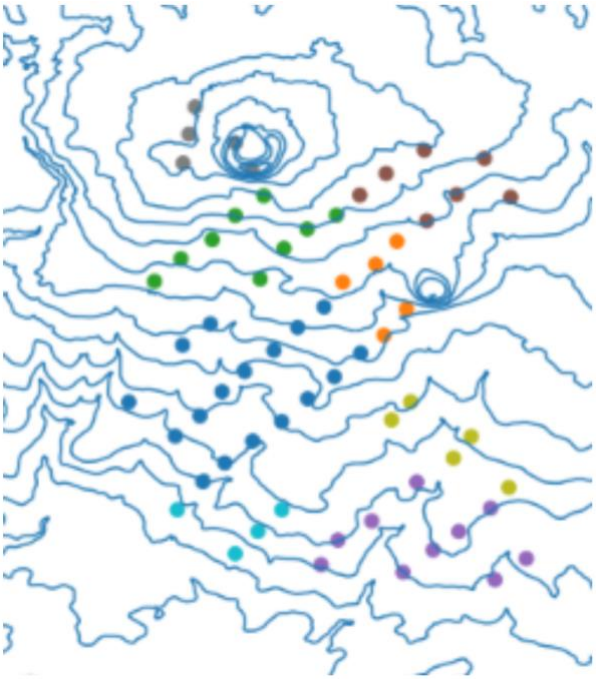


Figure II.1.2 *k-means clustering with 8 groups*

2. Hierarchical based clustering

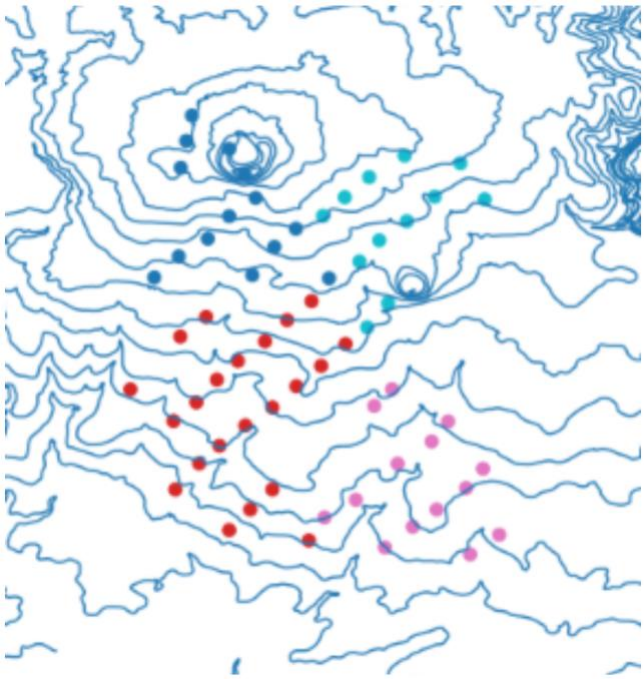


Figure II.2.1 *hierarchical clustering with 4 groups*

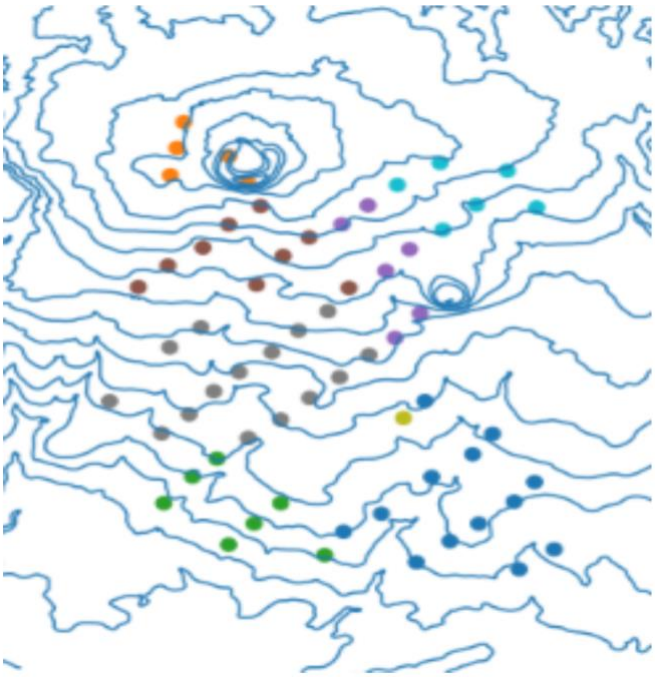


Figure II.2.2 *hierarchical clustering with 8 groups*

Melbourne
Energy
Institute

Melbourne Energy Institute

Level 1, Melbourne Connect

700 Swanston Street, Carlton, VIC 3053

mei-info@unimelb.edu.au

energy.unimelb.edu.au