Predicting Wind Power Generation using Time Series Modelling

Wanqing An

Supervisor: Professor Gillian Dobbie

Auckland University
June 6, 2022

Abstract

As pollution problems with traditional energy resources are getting much more serious than before, renewable energy resources are playing a more and more important role. As one of the main clean energy resources , the efficiency of utilizing wind power highly depends on the accuracy of its prediction. Many studies have been done in this area, with people using various models to predict the wind power, such as time series modelling using ARMA, but very few people have tried to use tree-based models like Random Forest to predict the wind power. In this paper, both tree-based models and time series models will be applied to predict wind power generation. The result shows that tree-based models have significant advantages over traditional time series models. What's more, Gradient Boost modelling performs better when predicting extreme values. After comparison between several different models, the application of these tree-based models should be considered in industry in the future.

1 Introduction

As many other traditional energy resources have caused many environmental issues globally, the renewable energy alternatives have gained much more attention than before. Due to the instability of wind power generation, to better manage the wind power generating system, the prediction of wind power both in the short-term and long-term is playing a more and more important role nowadays.

A lot of research has been done to summarize existing methods, which could help to increase the performance in predicting wind power. Generally, they can be divided into three categories (Lei et al., 2009); The first category is physical methods, which utilize atmospheric behaviors as predictors using mathematical equations, such as temperature, pressure and wind farm layout. The limitation of it is that it is only suitable for long term and large areas, meaning that it cannot predict short term wind power with high accuracy. The second category is statistical methods, which focus on analysing the relationship between power data at different time points. One popular statistical method is ARIMA, which establishes some regression relationships between input and output and is suitable for short-term prediction. One of its advantages is that it only needs historical data to predict, so the cost is more affordable (Milligan et al., 2003). Meanwhile the limitation of this method is obvious, the prediction ability of this method is not that good, as it cannot change its prediction strategy over time. The third category is machine learning methods, which could solve most of the problems mentioned above. The most commonly used method is artificial neural network, which has shown advantages in the prediction of wind power (Velázquez et al., 2011). Another commonly used method is Support Vector Machine, which is usually used for classification and has been expanded to regression (SVR). In some circumstances it outperforms the Multilayer Perceptron (MLP) neural networks (Mohandes et al., 2004). Though these methods have so many advantages, the problem is that all of they involve many parameters, and the tuning of parameters have significant impact on the accuracy of these methods.

Random Forests (Breiman, 2001) and Gradient Boosting (Friedman, 2001) are non-parametric methods suitable for short-term prediction. Some work has been done using bagged trees model to predict wind power (Harrou et al., 2019), which compared seven statistical and ensemble learning models. The bagged tree model outperforms the others in predicting wind power. However, the shortcoming of this paper is that it could only predict wind power based on current data. It has no capacity to predict wind power in the future. To increase application value in industry, being able to predict window power in the future is important, and that's where time series-based method comes into play. Embedding time series information into a tree-based model is a solution which combines the advantages of several methods. It takes into account the relationship between historical data and current wind power. What's more, in the ARIMA model, the purpose is to find a linear equation within wind power data and use it to predict. It tries to balance the prediction errors, which means ARIMA model is not that sensitive to the current data. In tree-based models, the story is very different. The training data in the past will not decrease the sensitivity of prediction, so it will have higher prediction accuracy in theory. A study proposed a model based on Random Forest could forecast an hour-ahead wind power showing another advantage of tree-based models. Unlike other machine learning methods, it does not need optimization and tuning (Lahouar & Slama, 2017).

Although they have so many advantages, very few people have tried to use them to predict wind power generation, so a framework for fitting these models on wind power data needs to be built to see if it has potential to increase the efficiency of predicting wind power. What's more, how much they are better than traditional ARIMA model is not known yet. Finally, which one of these two tree-based models will have better performance in dealing with wind power prediction problems remain unstudied. These three problems play an important role in wind power prediction.

Each model has its own advantage and disadvantage (Lei et al., 2009). The aim of this study is to find which model performs in particular circumstances. In this paper, I will first build three different models, including the ARIMA model, Random Forest model and Gradient Descent Boosting model, after fitting them on the same wind power data set. The results will be compared, so whether Random Forest model and Gradient Descent Boosting model are better than ARIMA model will be clear. Further more, the difference between the Random Forest and Gradient Descent Boosting model will tell which of them is a better choice for predicting wind power generation.

The rest of this paper is structured as follows: Section 2 describes the algorithms used to build model. Section 3 discusses the details of data preprocessing, the way to build model, the evaluation criteria and how to do forecasting. In Section 4, the results of comparison are displayed. Section 5 draws the conclusion of this research.

2 Methods

This section describes the algorithms used to build the models in this paper.

2.1 ARIMA

The ARIMA model comes from the ARMA model, which is a combination of AR and MA time series model. AR is a regression model which uses lagged variable of interest as predictors (Kotu & Deshpande, 2019). An autoregressive model of order p, could be represented:

$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \phi_3 y_{t-3} + ... + \phi_p y_{t-p} + \epsilon_t$, where y_i is the observed value in time i, ϕ_i is the corresponding coefficient which determines how much the past values impact the current value and ϵ_t is white noise.

While the MA model is a regression model which uses past prediction errors as predictors. Moving average model of order q, could be represented:

$y_t = c + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + ... + \theta_n\epsilon_{t-q}$, where y_i is the predicted value in time t, ϵ_i is white noise and θ_i is the corresponding coefficient which determines how much the past errors impact the current value.

Another thing needs to be considered in ARIMA model is : time series has to be stationary before fitting AR and MA models. Difference is wildly used to turn the non-stationary time series data to be be stationary, it is an operation on time-series data which computes the difference between two consecutive data points and rearrange them to be a new data set. The parameter d represents the order of difference which is needed to transform the original time series to stationary. Overall, there are 3 parameters in the ARIMA model that need to be consideration:

1. The autoregressive order p.
2. The moving average order q.
3. The order of difference applied in the original time series d.

2.2 RandomForest

Random Forest is a supervised machine learning algorithm made up of decision trees (Breiman, 2001). It is an improved algorithm using bagging. Bagging generates B different bootstrapped training data sets, then trains the model on the bth bootstrapped training data set in order to get $\hat{f}^b(x)$, and finally average all the predictions, to obtain (James et al., 2021):

$$\hat{f}_{bag}(x) = \tfrac{1}{B}\textstyle\sum_{b=1}^{B} \hat{f}^b(x)$$

As with bagging, Random Forest build a number of decision trees on bootstrapped training samples. When building these decision trees, instead of using whole predictors, it will only use part of them. The number of predictors will be used is a tuning parameter. By forcing each split to consider only a subset of the predictors, Random Forest guarantees the bagged trees have low correlation. The logic behind the Random Forest model is that individual decision trees perform much better as a group than they do alone. Random Forest has two tuning parameters:

1. The number of predictors will be used when building each tree m. Typically m = $\sqrt{p}$ is chosen, where p is the number of whole predictors.

2. B is the number of trees that will be built. As with bagging, Random Forest will not overfit if B is increased. In practice, B will be sufficiently large for the error rate to flatten.

2.3 Boosting

Like Random Forest building a number of decision trees, boosting works in a similar way, except that the trees are built sequentially: the establishment of each tree uses the information of the previous tree. Boosting is described in Algorithm 1 (James et al., 2021).

Algorithm 1 Boosting for Regression Tree

```
1. Set $\hat{f} = 0$ and $r_i = y_i$ for all i in the training set.

2. For b = 1,2,.....B, repeat:
    (a) Fit a tree $\hat{f}^b$ with d splits(d + 1 terminal nodes) to the training
        data(X,r).
    (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:
        $\hat{f} \to \hat{f}(x) + \lambda \hat{f}^b(x)$
    (c) Update the residuals,
        $r_i \to r_i - \lambda \hat{f}^b(x_i)$
3. Output the boosted model,
        $\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$
```

Boosting has three tuning parameters:

1. The number of trees, B. If B is too large, boosting tends to overfit. B could be selected by cross-validation.

2. The shrinkage parameter, λ, a small positive number, usually between 0.01 and 0.001. This is the learning rate of boosting. This parameter interact with the number of trees. When tuning them, they need to be considered together.

3. The number of splits in each tree, d, which controls the complexity of the boosted ensemble. Often when d is a small value works well, in which case each tree is a weak classifier.

3 Experiments

In this section, we describe the data used, preprocessing that was carried out, building model and tuning parameters, evaluation criteria and how the forecasting is done.

3.1 Data and Feature selection

The data contains various weather, and turbine and rotor features related to a certain windmill. It is recorded between January 2018 and March 2020 at 10-minute intervals (Bhaskarpandit, n.d.).

The original data has 20 columns, after dropping duplicated columns and missing columns, there are 16 left. Figure 1 shows the correlation between each pair of columns, which indicates any linear relationships between variables. The variables with highest correlation with ActivePower is WindSpeed. Meanwhile, other variables that are highly related to ActivePower is significantly related to WindSpeed as well, so WindSpeed is chosen to be a part of inputs of models.

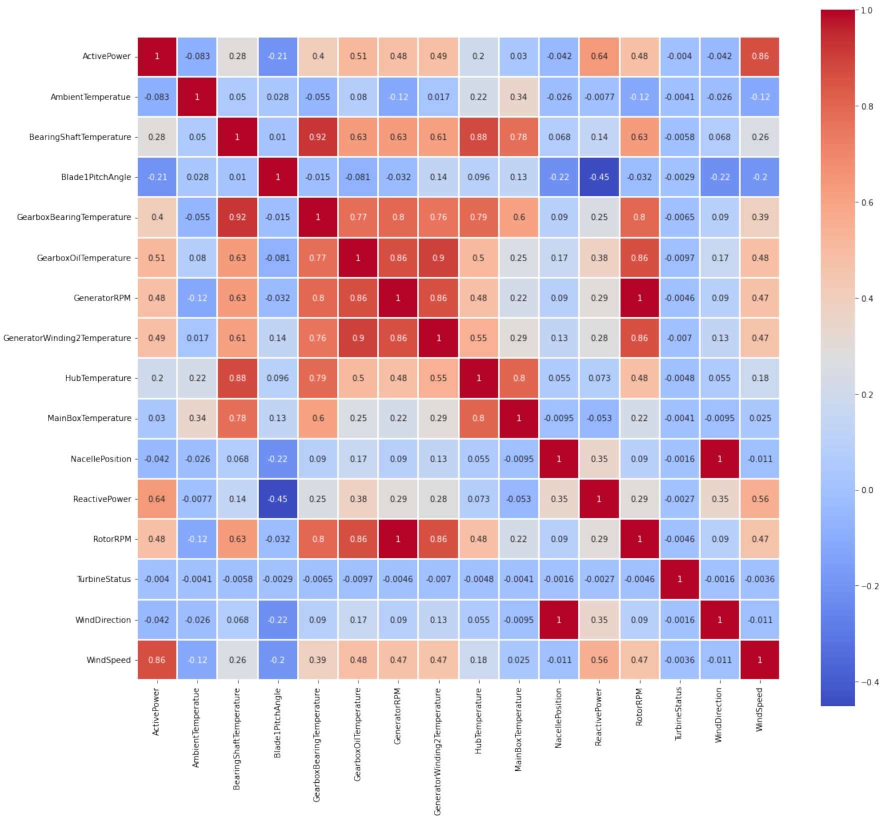


Figure 1: Correlation value between features

Apart from the linear relationship, some variables have non-linear relationships with ActivePower. Figure 2 shows the relationship between wind power and wind direction. Figure 2(a) displays that most recorded wind power data points appear around the 180 degree and 360 degree. On the other hand, Figure 2 (b) indicates that wind in 180 degree and 360 degree contribute most to the wind power generated over the whole year. As can be seen, there are some visible peaks and valleys, which asserts using wind direction to predict wind power is necessary.

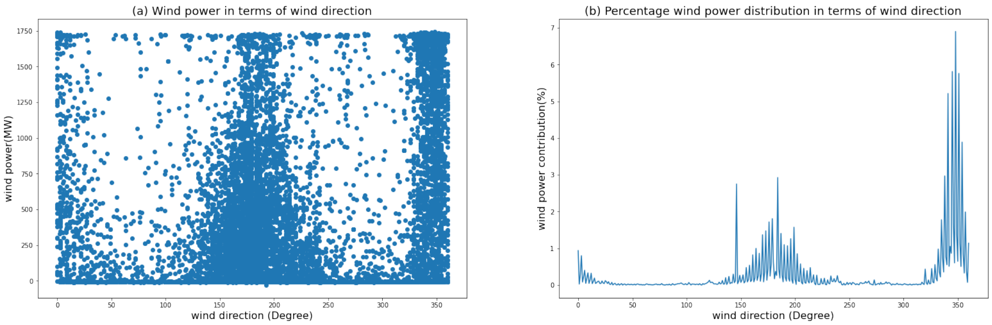


Figure 2: Relationship between wind power and wind direction

3.2 Data Preprocessing

As mentioned in Section 3.1, two features are chosen to predict wind power: wind speed and wind direction. This section describes how to preprocess the data in different ways so it can match different models.

For the ARIMA model, standardize the wind power **P** so it falls between 0 and 1. Then only take data every six data points, which means take the data of the last 10 minute as the active wind power of that hour. Its equation form is **P(h:m)**, where h is hour and m is minutes, and $m \in [00, 10, 20, 30, 40, 50]$. This wind power data vector will be used as input of the ARIMA model,this structure is depicted by these equations:

Power input (n depends on ARIMA model fitting):
X = [P(h:00),P(h-1:00),...,P(h-n:00)]

Measured power output:
Y = P(h+1:00)

Predicted power output:
$(\hat{Y}) = \hat{P}(h + 1 : 00)$

For Random Forest and Gradient Boosting, first standardize the wind speed **S** and wind direction **D** vectors so they fall between 0 and 1, then take six consecutive data points of wind speed and wind direction as input vectors to predict the wind power of the next time point. This structure is depicted by these equations:

Speed input:
S(h) = [S(h:00),S(h:10),...,S(h:50)]

Direction input:
D(h) = [D(h:00),D(h:10),...,D(h:50)]

Input vector:
X = [S(h),D(h)]^T

Measured power output:
Y = P(h+1:00)

Predicted power output:
$\hat{(Y)} = \hat{P}(h+1:00)$

3.3 Model building

For the ARIMA model, to make it efficient and useful, its input has to be stationary. If it is not, the difference is taken to make it stationary. In this project, the ADF test is used to test if its stationary and the original data passed the ADF test, so it can be used to build the model directly, then AIC and BIC can be used to decide the order of AR and MA. In this project, it turns out that both AIC and BIC model chooses 3 as the order of AR and 4 as the order of MA, so the ARIMA model for Active wind power prediction can be written as **ARIMA(3,0,4)**.

Random Forest and Gradient Boosting models use online learning. The principle for online learning is set the time of one year as a fixed window, use the data within the window as the training set, and use the data of one day after this as the test set. Then roll the window, rolling back the length of one day each time, and finally get the prediction result of n days. These n results will be used as the final test result. More details will discuss later in section 3.5.

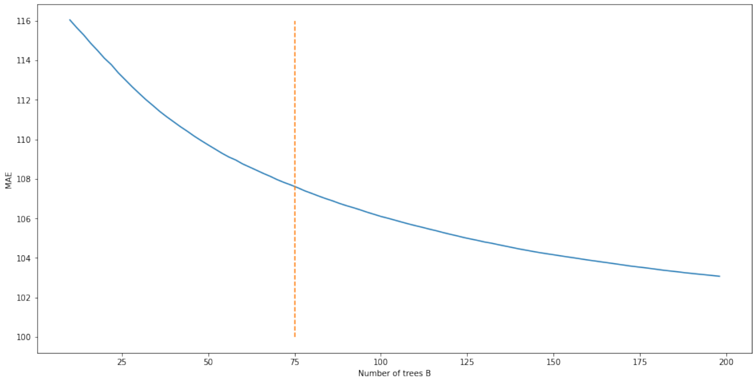


Figure 3: MAE with number of trees, B

As for tuning parameters, grid search is used to determine the optimal parameters. For Random Forest, two parameters need to be considered. In our experiments m will use the most commonly used value, which is $\sqrt{p}$. The value of B is set in the following way, first the value of B is set from 10 to 200, and a model is built for every second. As the online learning principle, there is no need to consider overfitting, MAE is recorded for those models in order to select optimal B. From Figure 1, the decreasing rate of MAE tends to be slow at around 75, so B will be set as 75. For Gradient Boosting, three parameters need to be considered. Firstly, set d=1 which means each tree is a stump. Then finding the optimal λ and corresponding B, set paired values $\lambda \in$ (0.01, 0.05, 0.1, 0.2), B $\in$ (100,200,300,400,500,600,700) to build a model respectively. The result of model performance with different parameter combinations is shown in Figure 4.

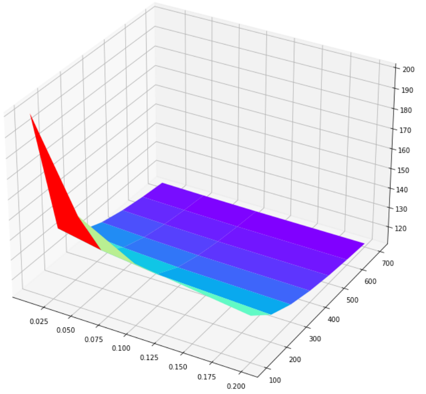


Figure 4: MAE with different combinations of λ and B

Figure 4 indicates that when B increases up to 400, the decreasing rate of MAE is tending to slow down. Considering the balance between efficiency and accuracy, set B = 400. Then use a narrow range of combinations to determine the λ. Setting $\lambda \in$ (0.02, 0.05, 0.1, 0.15, 0.2) and B $\in$ (300, 400 ,500) to fit models, the result is shown in Figure 5.

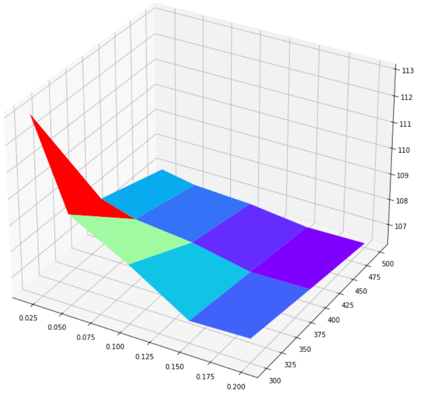


Figure 5: MAE with different combination of λ and B

From Figure 5, it can be seen that along with the learning rate increasing the MAE first increasing and then decreasing, the lowest point is at 0.15. So we set $\lambda = 0.15$.

After determining the value of B and λ, set d $\in$ (1,2,3,4,5,7,8,9,10) to build model respectively. As d value changes, the value of MAE changes slightly. Considering the balance between time efficiency and accuracy, set d=5.

3.4 Evaluation criteria

To compare the performance of different models, several criteria are used to evaluate the results. We choose the following: the mean absolute error (MAE), the mean squared error (MSE), the mean absolute percentage error (MAPE), and the maximum absolute error (MXE).

S stands for the total number of errors, $\hat{P}_i$ stands for the predicted wind power while P_i stands for true wind power.

$$MAE = \frac{1}{s}\sum_{i=1}^{s}|\hat{P}_i - P_i|$$

$$MXE = \max_{1\leq i\leq s}|\hat{P}_i - P_i|$$

$$MAPE = \frac{1}{s}\sum_{i=1}^{s}\frac{|\hat{P}_i - P_i|}{P_i} \times 100$$

$$MSE = \frac{1}{s}\sum_{i=1}^{s}(\hat{P}_i - P_i)^2$$

The MAE is one of the most commonly used criteria, avoiding error offset by using absolute distance between predicted value and true value. It simply shows the average of all found errors. The MSE is very similar to MAE, the only difference is that it will give more weight to more serious mistakes since it will square every error and sum them up before averaging. The MXE detects the maximum error, and is useful when analyzing some special situations which lead to the highest error. The last one, MAPE returns a percentage error of a model. Unlike MAE and MXE, the scale of data has no impact on it, but a problem might happen when the true value is close to zero. In that case, it will overestimate the error a lot and lead to some misunderstandings.

3.5 Forecasting

As mentioned in Section 3.2, we use an online learning method. When doing prediction we use the data of the past year as the training data to fit the model, and use the data of the next day as the test data to do predictions. There are 24 outcomes for each prediction. That is to say, if you want to predict the wind power of H days in the future and the current day is the ith day, you need data from (t-365)th day to (t+H-1)th day. Then build H models, one model can get the results of one day. Then combine these H days results as the overall prediction result. Figure 6 shows the principle of online learning and forecasting.

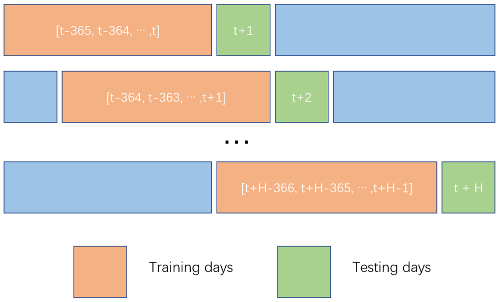


Figure 6: Online learning and forecasting principle

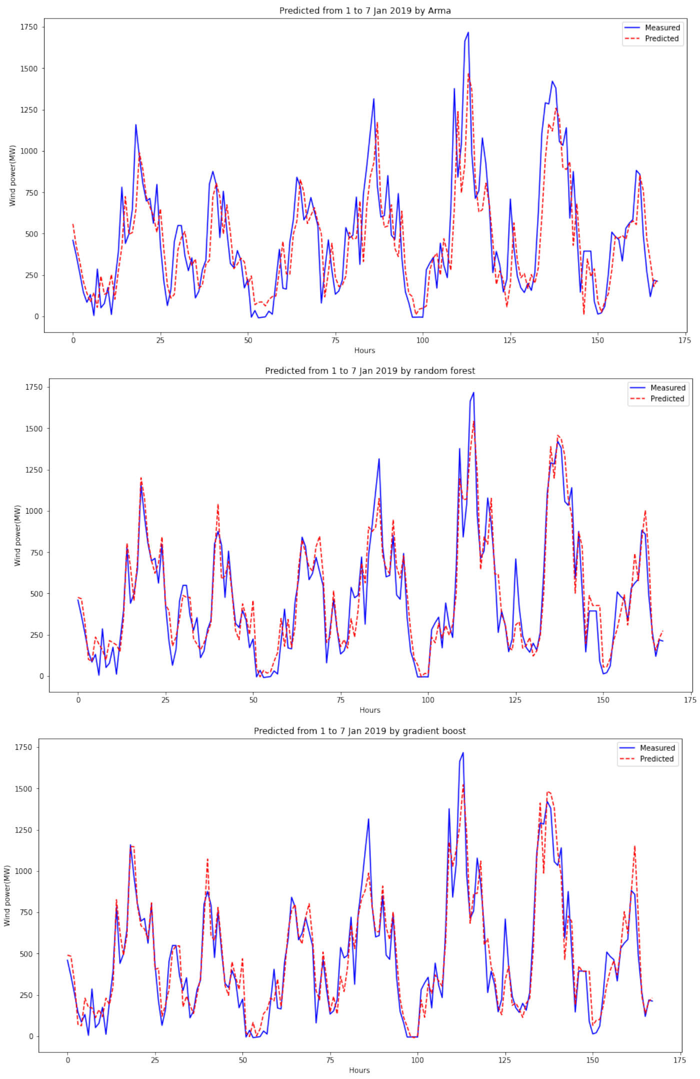
4 Result

As shown in Table 1, all evaluation criteria are considerably higher for the ARMA model than RF and GB. As for RF and GB, there is a slight difference in the evaluation criteria for the two models, thus we will further compare these two models use Wilcoxon's signed rank test.

Table 1: Forecast evaluation criteria of all methods throughout Jan 2019

criterion/method	Arma model	Random Forest(RF)	Gradient Boost(GB)
MAE	163.4955	90.4954	91.1679
MXE	745.2776	413.6846	393.5775
MAPE	1.7000	0.9686	1.0044
MSE	46240.8266	14158.9735	15374.5339

From Figure 7, there are some clues of the potential reasons for the poor performance of the ARIMA model. There is a lag in the predicted values of this model, and the predicted values are distributed in a more narrow range than the true values. For RF and GB, the forecasting result is accurate in most cases, except some abrupt points.


Predicted from 1 to 7 Jan 2019 by Arma

Predicted from 1 to 7 Jan 2019 by random forest

Predicted from 1 to 7 Jan 2019 by gradient boost

Figure 7: Experimental results of 3 models over the same testing periods

To further compare the performance of Random Forest and Gradient

Boosting models, each model is used to forecast 200 days through the method outlined in Section 3.4, and Wilcoxon's signed rank test is used to tell if the difference between the results of models is statistically significant.

The statistical analysis was conducted for 2 populations with 200 paired samples. The family-wise significance level of the tests is alpha=0.050. The null hypothesis is that the population is normal for the populations Random Forest (p=0.000) and Gradient Boosting (p=0.000). Therefore, assume that not all populations are normal. Because there are only two populations and both of them are not normal, Wilcoxon's signed rank test is used to determine the differences in the central tendency and report the median (MD) and the median absolute deviation (MAD) for each population.

Table 2 shows the test result. It failed to reject the null hypothesis at confidence interval of 1% (p=0.999) of Wilcoxon's signed rank test that population RandomForest (MD=115.446+-8.372, MAD=18.162) is not greater than population Gradient Boosting (MD=114.169+-9.097, MAD=17.187). Therefore, we cannot accept the alternative hypothesis that there is a statistically significant difference between the medians of the populations.

However, even though the difference in the long term is almost negligible, its possible that the difference of different periods just offsets each other, which means one model might have higher accuracy in a certain period while lower accuracy in others. Since as a natural phenomenon, wind has very strong seasonality, which leads to that one model might outperform the other in different seasons, so testing models in each season is necessary.

Table 2: Rank result for RF and GB

	meanrank	median	mad	ci_lower	ci_upper	magnitude
GradientBoost	1.59	115.4460	18.1620	106.5949	123.3383	negligible
RandomForest	1.41	114.1694	17.1872	106.6473	124.8420	negligible

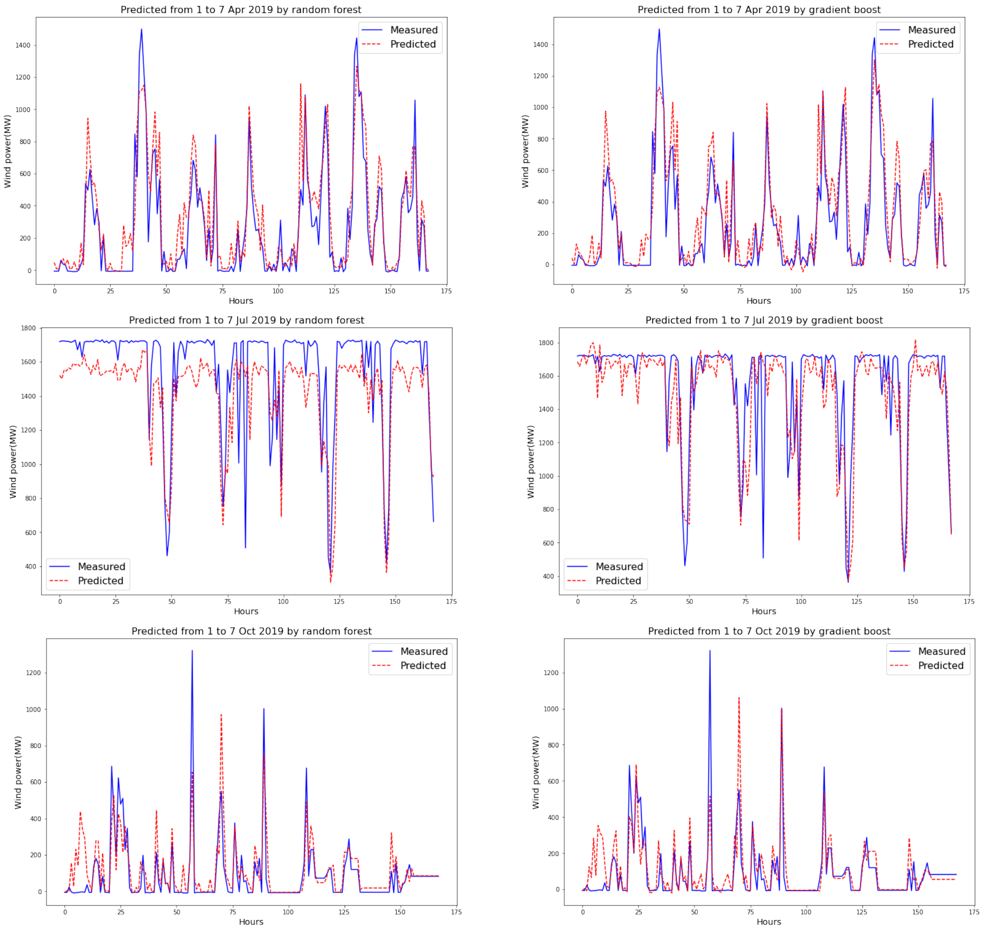


Figure 8: RF and GB experimental results over 3 testing periods

Figure 7 and Figure 8 indicate that Random Forest model and Gradient Boosting model have no obvious difference in prediction in January, April, and October. However, in July when wind is strong and wind power remains close to its maximum, the performance of Gradient Boosting model enormously outperforms Random Forest model.

Table 3: Forecast evaluation criteria of RF and GB over different seasons

criteria/period	Apr RF	Apr GB	Jul RF	Jul GB	Oct RF	Oct GB
MAE	107.97	108.96	206.50	132.38	57.95	60.78
MXE	668.11	513.46	1035.19	1170.78	649.54	805.74
MAPE	3.79	3.48	0.14	0.11	4.20	4.37
MSE	24196.90	23849.41	57841.29	42898.90	11483.28	136.02

The difference between Random Forest model and Gradient Boosting model is clear in Table 3, in July, the MAE and MSE of Random Forest model is significantly larger than Gradient Boosting model.

5 Discussion and Conclusion

The findings of this study suggest that, as expected, Random Forest model and Gradient Boosting model are significantly better than the ARIMA model in wind power prediction. The difference is obvious in all four evaluation criteria: MAE, MXE, MAPE and MSE.

For Random Forest model and Gradient Boosting model, in the long term, there is no evidence that the difference between the two is statistically significant. However, their predictive abilities differ in more subtle ways. The MAE, MAPE and MSE of Random Forest model are slightly lower than Gradient Boosting model, however the MXE of Random Forest model is larger than Gradient Boosting model. This might suggest Gradient Boosting is a better choice to predict wind power in extremely unexpected conditions. Meanwhile, Random Forest shows advantage in normal circumstance. While in certain periods when the wind is strong and the wind power remains close to its maximum, Gradient Boosting model has much higher accuracy.

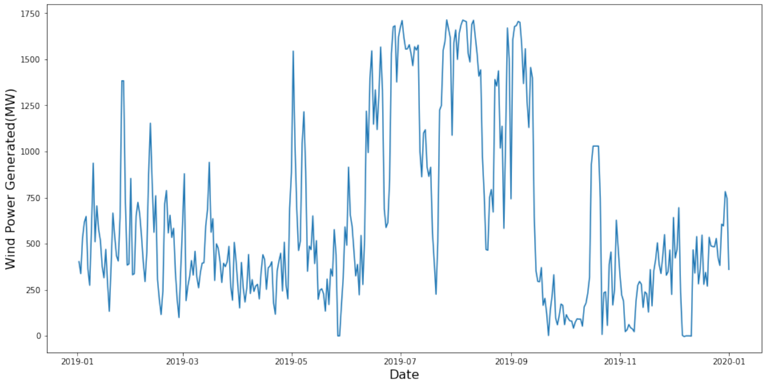


Figure 9: Wind power change in 2019

As Figure 9 shows, wind power usually has very strong seasonality, in most areas of this world, wind power will mainly distribute in several months, since Gradient Boosting model shows advantage in predicting extreme large values, Gradient Boosting model should be the first choice in most cases.

The reason behind the different forecasting performance of RF and GB over different periods may be relevant with the principle of these two methods. There is a possible explanation: as using bagging, the forecasting results of Random Forest tend to have lower variance. In contrast, Gradient Boosting assigns weight based on the residual, so the forecasting results tend to have lower bias. Gradient Boosting model could learn more information from the abrupt change and extreme points. How the principle behind these two methods impact the forecasting results is worth to explore in the future.

Another possible research is the comparison between the tree-based model and Neural Network model. Some work has been done (Lahouar & Slama, 2017), showing that when more irrelevant variables are added to the inputs, the results of Neutral Networks will be degraded massively, while the tree-based are still accurate in most cases. This means tree-based model are more robust in that it benefits from additional information without getting confused when these inputs are not related to each other. However, other research proposed a method for forecasting using LSTM which could prevent overfitting to some extent (Ookura & Mori, 2020), which shows that it has some advantages over other models. It looks like different models show advantages in different circumstances. Further comparisons between tree-based models and Neural Network models is needed to determine what kind of model should be applied in different conditions.

References

Bhaskarpandit, S. (n.d.). *Wind power forecasting. kaggle. retrieved june 4, 2022.* https://www.kaggle.com/datasets/theforcecoder/wind-power-forecasting./. (2020, July 24)

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Harrou, F., Saidi, A., & Sun, Y. (2019). Wind power prediction using bootstrap aggregating trees approach to enabling sustainable wind power integration in a smart grid. *Energy Conversion and Management*, *201*, 112077.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). Tree-based methods. In *An introduction to statistical learning* (pp. 327–365). Springer.

Kotu, V., & Deshpande, B. (2019). Time series forecasting. *Data Science; Elsevier: Amsterdam, The Netherlands*, 395–445.

Lahouar, A., & Slama, J. B. H. (2017). Hour-ahead wind power forecast based on random forests. *Renewable energy*, *109*, 529–541.

Lei, M., Shiyan, L., Chuanwen, J., Hongling, L., & Yan, Z. (2009). A review on the forecasting of wind speed and generated power. *Renewable and sustainable energy reviews*, *13*(4), 915–920.

Milligan, M., Schwartz, M., & Wan, Y.-h. (2003). *Statistical wind power forecasting models: Results for us wind farms* (Tech. Rep.). National Renewable Energy Lab.(NREL), Golden, CO (United States).

Mohandes, M. A., Halawani, T. O., Rehman, S., & Hussain, A. A. (2004). Support vector machines for wind speed prediction. *Renewable energy*, *29*(6), 939–947.

Ookura, S., & Mori, H. (2020). An efficient method for wind power generation forecasting by lstm in consideration of overfitting prevention. *IFAC-PapersOnLine*, *53*(2), 12169–12174.

Velázquez, S., Carta, J. A., & Matías, J. (2011). Influence of the input layer signals of anns on wind power estimation for a target site: A case study. *Renewable and Sustainable Energy Reviews*, *15*(3), 1556–1566.