VIET NAM GENERAL CONFEDERATION OF LABOUR

TON DUC THANG UNIVERSITY

FALCUTY OF ELECTRICAL – ELECTRONIC

NINH THE VINH CUONG

VERY SHORT-TERM FORECASTING OF WIND POWER

UNDERGRADUATE THESIS OF ELECTRICAL ENGINEERING

HO CHI MINH CITY, YEAR 2022

VIET NAM GENERAL CONFEDERATION OF LABOUR
TON DUC THANG UNIVERSITY
FALCUTY OF ELECTRICAL – ELECTRONIC

NINH THE VINH CUONG – 418H0363

VERY SHORT-TERM FORECASTING OF WIND POWER

UNDERGRADUATE THESIS OF ELECTRICAL ENGINEERING

Advised by
Prof. Hong-Tzer Yang
Dr. Huynh Van Van

HO CHI MINH CITY, YEAR 2022

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my Instructor Prof. Hong-Tzer Yang from National Cheng Kung University, without his guidance, this project can not be complete. I also could not have undertaken this journey without the help of Prof. Huynh Van Van from Ton Duc Thang University, Prof. Chao-Ming Huang from National Cheng Kung University for showing me the right way. Additionally, I am extremely grateful to Nuttapat "Beam" Jittratorn and Andi Abd. Halik Lateko from NCKU, their comments and questions are excessively helpful for my project.

Thanks should also go to Prof. Dinh Hoang Bach, my homeroom teacher for 4 years in Ton Duc Thang University. He led me through hardness and obstacles in my adventure. I am also thankful to Prof. Vu Tri Vien, who gives inspiration to my learning process.

I would be remiss in not mentioning family and friends, especially Ha Mai Phung, for encourage me in my days in university.

Ho Chi Minh City,

Author

(Signature and full name)

This thesis was carried out at Ton Duc Thang University.

Adviser: ....

....

(Title, full name and signature)

This thesis is defended at the Undergraduate Thesis Examination Committee was hold at Ton Duc Thang University on … /…/……

Confirmation of the Chairman of the Undergraduate Thesis Examination Committee and the Dean of the faculty after receiving the modified thesis (if any).

CHAIRMAN	**DEAN OF FACULTY**
…………………………….	………………………………

DECLARATION OF AUTHORSHIP

I hereby declare that this thesis was carried out by myself under the guidance and supervision of Dr. Huynh Van Van and Prof. Hong-Tzer Yang and that the work and the results contained in it are original and have not been submitted anywhere for any previous purposes. The data and figures presented in this thesis are for analysis, comments, and evaluations from various resources by my own work and have been duly acknowledged in the reference part.

In addition, other comments, reviews and data used by other authors, and organizations have been acknowledged, and explicitly cited.

I will take full responsibility for any fraud detected in my thesis. Ton Duc Thang University is unrelated to any copyright infringement caused on my work (if any).

Ho Chi Minh City,

Author

ĐẠI HỌC TÔN ĐỨC THẮNG **KHOA ĐIỆN –ĐIỆN TỬ**	**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM** **Độc lập – Tự do – Hạnh phúc**

Tp.HCM, ngày 28 tháng 3 năm 2022

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

(Bản nhiệm vụ này được đóng vào trang thứ nhất của đồ án tốt nghiệp)

Họ tên sinh viên: Ninh Thế Vĩnh Cường

Ngành: Kỹ thuật điện   Lớp: 18H40101   MSSV: 418H0363

1. Tên đề tài: Very short-term forecasting of Wind power.

2. Nhiệm vụ (yêu cầu về nội dung và số liệu ban đầu):

- Có được dữ liệu tin cậy
- Thực hiện tiền xử lý dữ liệu
- Thực hiện ít nhất 02 mô hình dự báo
- Có kết quả dự báo của các mô hình
- Đánh giá các mô hình
- Có sự so sánh hoặc kết hợp kết quả của các mô hình
- Nêu kết luận và dự định

3. Ngày giao nhiệm vụ đồ án tốt nghiệp: 28/03/2022

4. Ngày bảo vệ 50% đồ án tốt nghiệp: 20/05/2022

5. Ngày hoàn thành và nộp về khoa: 15/7/2022

6. Giáo viên hướng dẫn: TS. Huỳnh Văn Vạn

GS. HongTzer Yang

Nội dung và yêu cầu ĐATN đã được thông qua Khoa và Bộ môn.

Ngày 28 tháng 3 năm 2022 CHỦ NHIỆM BỘ MÔN TS. Đinh Hoàng Bách	GIẢNG VIÊN HƯỚNG DẪN TS. Huỳnh Văn Vạn

TRƯỞNG KHOA

TS. Đồng Sĩ Thiên Châu

TON DUC THANG UNIVERSITY
ELECTRICAL – ELECTRONIC

SOCIALIST REPUBLIC OF VIETNAM
Independence - Freedom – Happiness

UNDERGRADUATE THESIS SCHEDULE

Student: Ninh The Vinh Cuong
Class: 18H40101 ID: 418H0363
Project: Very short-term forecasting of Wind Power

Week/Day	**Work**		**Instructor**
	Done	**Continue**	
1-19/04	Introduction, Objectives	Theoretical background	[signature]
2-26/04	Theoretical background	Data Preprocessing	[signature]
3-03/05	Data Preprocessing	Data Preprocessing	[signature]
4-11/05	Data Preprocessing	Statistical methods	[signature]
5-18/05	ARIMA model	Statistical methods	[signature]
6-25/05	ARIMAX model	Statistical methods	
Mid-term check	Evaluation of completed volume ..50...% be allowed to continue/discontinue Undergraduate thesis.		
7-01/06	Statistical methods	Machine Learning methods	[signature]
8-08/06	Linear Regression	SVR, Nu-SVR	[signature]
9-15/06	SVR, Nu-SVR	Ensemble	[signature]
10-22/06	Bagging models	Ensemble	[signature]
11-29/06	Bagging models	Ensemble	[signature]
12-06/07	Stacking model	Ensemble	[signature]
13-13/07	Ensemble	Finish project	[signature]
Submit	Completed.../00% Undergraduate thesis protected/unprotected Undergraduate thesis.		

VERY SHORT-TERM FORECASTING OF WIND POWER

ABSTRACT

Due to the fluctuation of the wind, the power produced by it is also not flat enough consequently. Therefore, the possibility of system reliability issues increases rapidly when a large amount of wind power is installed on the grid. Because wind power cannot be timetabled beforehand like traditional generation units, forecasts of wind power production in the next hours in advance can schedule the number of wind power available. Forecasting in wind power has the potential to reduce the amount of reserves necessary and lower the cost of electricity in such systems eventually. In this work, from historical data, the author examines the ability of statistical time series analysis and machine learning tools and forecast future output of wind power. The ability of models has been examined at several different error input data. Therefore, a case study involving wind farm data from Changhua, Taiwan is used to show how these tools may be beneficial for planning unit commitment, scheduling, and dispatch by system operators, and maximizing profit by electricity traders.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ARIMA	Autoregressive integrated moving average
ARIMAX	Autoregressive integrated moving average with exogenous variables
MA	Moving Average
MRE	Mean Relative Error
SVR	Support Vector Regression
WT	Wind Turbine

CHAPTER 1. INTRODUCTION

1.1 Introduction

Wind power is the cleanest prevalent renewable energy due to the least greenhouse gas emissions (Bruckner T., 2014). It has the largest resource available thus, the installation to the electricity system is essential. However, the introduction of large amounts of wind power capacity into the electricity system has the possibility of disrupting normal operations as a consequence of the intermittent and uncertain in nature of wind power output. The best solution for this situation is to be able to forecast the wind output as it is impossible to control the wind. The uncertainty involved might not be eliminated by this solution, but the variability can decrease. When operating system, the amounts of variances, amounts of spinning reserves, and overall system cost can be reduced by improvements in wind forecasting technology. Autoregressive integrated moving average (ARIMA) and its variant, Linear Regressions, and Support Vector Regressions (SVR) are models that will be highlighted in this work. These models come from statistical time series analysis and machine learning methods.

The models in this work will forecast future wind power values in an attempt to outperform persistence for 10-min and one-hour forecasts. the utility of the ARIMA forecasting method can be affected by a number of different factors, including training period length and model orders. The author has attempted to apply as systematic of an approach toward the selection of these values as possible. The Regression models is a classical method, and it is suitable for me, who have no background in machine learning. The author have chosen to focus on forecasting the power of a single turbine since this is my first attempt to analyze data as well as apply machine learning models.

The data utilized in this study comes from data supplied by the National Cheng Kung University (NCKU) which are real wind data from a wind farm in Changhua, Taiwan

(2020). These case studies are used to showcase the advantages that these statistical techniques and machine learning models may offer independent system operators, or large balancing authorities.

1.2 Objectives

This project aims to apply the statistical tools as well as machine learning tools to forecast very short-term future output of a Wind Turbine, that are one-step ahead and one-hour ahead from the current time. Models applied in this project should be able to handle up to 50% of error in input values, the have reasonable errors for the outcomes (will be clarify in next chapter).

CHAPTER 2. THEORETICAL BACKGROUND

2.1 Wind Energy

2.1.1 Wind turbine

A wind turbine (WT) is a machine which converts the power in the wind into electricity. This is in contrast to a 'windmill', which is a machine which converts the wind's power into mechanical power. As electricity generators, WTs are connected to some electrical network. These networks include battery-charging circuits, residential scale power systems, isolated or island networks, and large utility grids (J.F. Manwell, 2010).

A WT is a structure included:

- Tower and foundation: place for putting all other parts.
- Generator: generate electricity from wind power.
- Gearbox: control rotor's speed.
- Blades: convert wind energy into mechanical power.

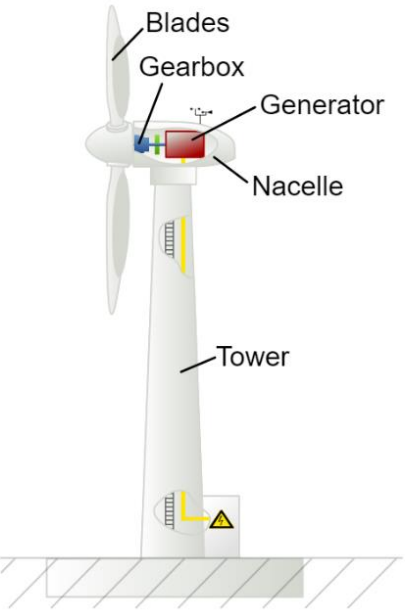


Figure 2-1: Structure of a WT (Source: Internet)

When wind flow across the blades, on one side of the blade the air pressure decreases. Lift and drag force create as a consequence of difference in air pressure across two sides of the blade. rotor to rotate as a result of the force of the lift and drag, by the reason of the first force is stronger than the second. There are two types of rotors – generator connection: directly (if it's a direct drive turbine) or through a shaft and a series of gears (a gearbox) that allow for a physically smaller generator by speed up the rotation. The generator produces electricity because of the rotation which come from the force of aerodynamic. Notwithstanding, in real life, WT cannot capture more than 59.3% of energy from the wind, which is known as Betz limit. It is tried to design WT to operate at efficiencies close to the Betz limit, but this is also limited by the losses from mechanical assembly. Hence, the cost and the efficiency must be balanced, one way selecting the number of blades.

The speed and efficiency of WT are affected by the number and configuration of the blades. The slipstream effect is covariate with the number of the blades. Inadequate performance and poor efficiency are the results if there are too few blades. In contrast, too large number of blades causes weight to increase as well as production cost. It is essential to select the correct number of blades to fit the generator performance curve to optimize overall WT performance and efficiency.

Two-blade wind turbines are slightly less efficient than three-blade wind turbines and must rotate faster for maximum efficiency. Similarly, two blades will produce more electricity than three blades, but have their own problems. Two-blade turbines are sensitive to gyroscopic precision, which results in wobbling. Naturally, this wobbling will cause stability problems for the entire turbine. This will also put pressure on the turbine's components, reducing its efficiency and lifetime (Kehinde A. A., 2021).

Having an array of WTs in the same location makes wind power plants for producing electricity. Wind conditions, access to electric transmission, the surrounding terrain, and other siting considerations are several factors that was impacted by the location of a wind power.

The performance of a given WT generator can be related to three key points on the velocity scale:

- **Cut-in speed**: the minimum wind speed at which the machine will deliver useful power.

- **Rated wind speed**: the wind speed at which the rated power (generally the maximum power output of the electrical generator) is reached.
- **Cut-out speed**: the maximum wind speed at which the turbine is allowed to deliver power (usually limited by engineering design and safety constraints).

Power curves for existing machines can normally be obtained from the manufacturer. The curves are derived from field tests, using standardized testing methods.

Figure 2-2: Phuong Mai wind power in Binh Dinh Province – Viet Nam (Source: Internet)

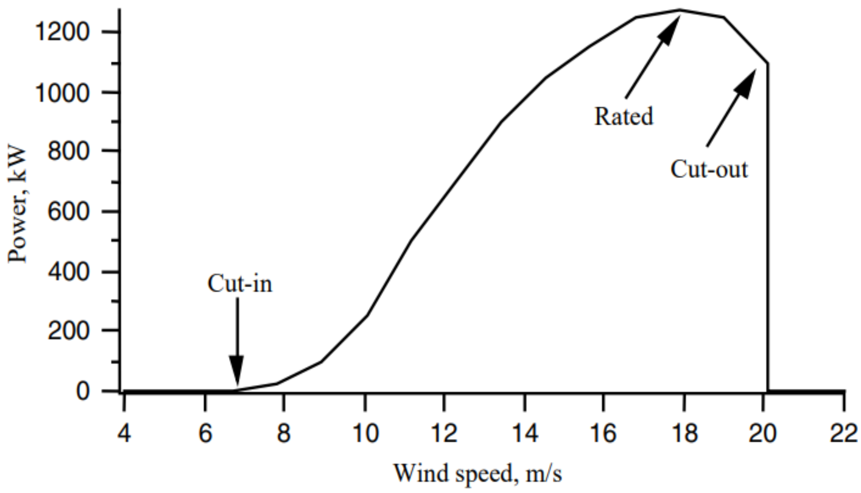


Figure 2-3: Typical wind turbine power curve (J.F. Manwell, 2010)

2.1.2 Status of wind power and potential in Viet Nam

Because of the increasing of greenhouse gas emission over the years (Figure 2-4) the world is changing to use renewable energy. Companies selling electrical vehicles are growing their market share (Figure 2-5). The International Energy Agency released a new report looking at car sales around the world and confirmed that EV market share jumped from 4.11% in 2020 to 8.57% in 2021 (Leonardo Paoli, 2022).

The agency wrote about the impressive electric vehicle growth rate over the last 3 years:

> *Growth has been particularly impressive over the last three years, even as the global pandemic shrank the market for conventional cars and as manufacturers started grappling with supply chain bottlenecks. In 2019, 2.2 million electric cars 1 were sold, representing just 2.5% of global car sales. In 2020, the overall car market contracted but electric car sales bucked the trend, rising to 3 million and representing 4.1% of total car sales. In 2021, electric car sales more than doubled to 6.6 million, representing close to 9% of the global car market and more than tripling their market share from two years earlier. All the net growth in global car sales in 2021 came from electric cars.*

This led to the requirement of electrical supply and wind is the choice since it is the cleanest prevalent renewable energy due to the least greenhouse gas emissions (Bruckner T., 2014).

China, a major energy-consuming carbon emission country, is one of many countries that have installed WTs (Figure 2-6). In 2021, China, single-handedly accounted for 40% of wind capacity of the entire World as in Figure 2-7 (IRENA, 2022). Viet Nam, even though, is not have large scale in wind capacity, but it is rapidly growing (Figure 2-8 and Figure 2-9), especially in the period of 2020 – 2021. With a coastline of more than 3000 km and its location in the monsoonal climate zone, Vietnam is expected to have good potential for wind energy development (Duc Luong, 2015).

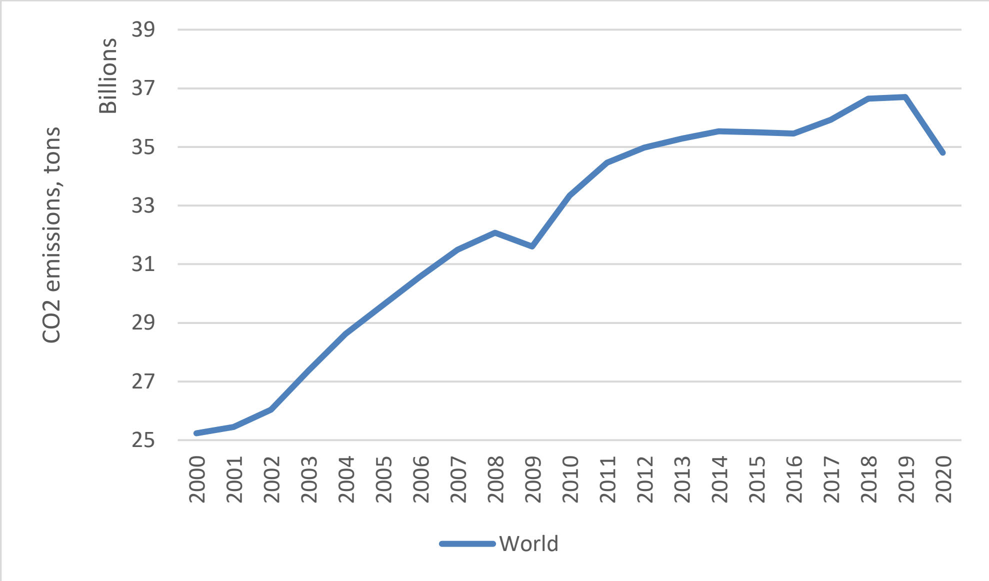


Figure 2-4: Annual greenhouse gas emission by fuel fossil (Ritchie, 2020)

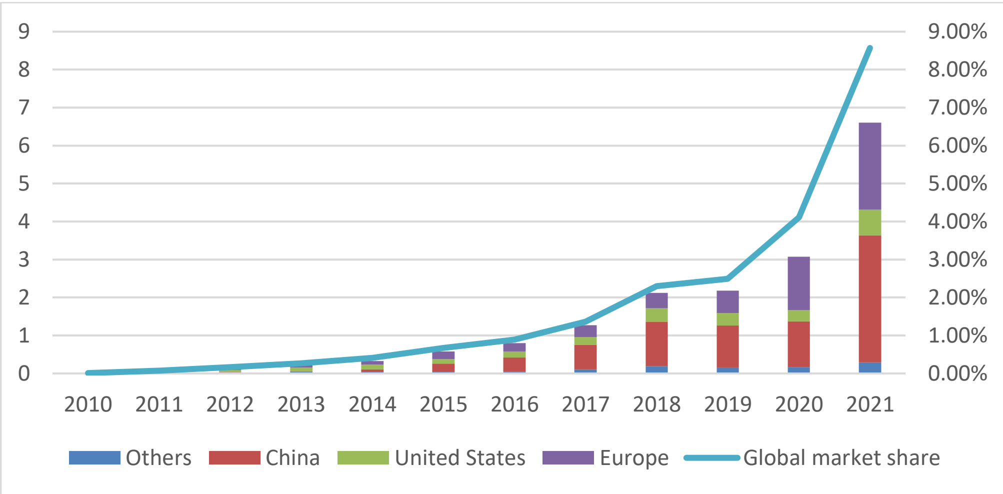


Figure 2-5: Electric vehicles global sales and sales market share (Leonardo Paoli, 2022)

In order to meet the rapidly increasing energy demand, the Government of Vietnam has decided to increase its reliance on renewable energy sources. Following the Prime Minister's Decision No. 1855/QD-TTg dated 27 Dec 2007 approving the "National energy development strategy up to 2020, with a vision to 2050", the specific targets of increasing the renewable energy proportion have been set at 5% and 11% of the total *primary energy consumption by 2020 and 2050, respectively* (Quyết định số 1855/QĐ-TTg của Thủ tướng Chính phủ ngày 27/12/2007 về phê duyệt Chiến lược phát triển năng lượng quốc gia của Việt Nam đến năm 2020, tầm nhìn đến năm 2050).

Compared to the other Southeast Asian countries, Vietnam has a huge basin of renewable energy sources, including hydropower, wind, solar, biomass, geothermal, and wave and tide—all capable of supplying a large part of the country's energy requirements.

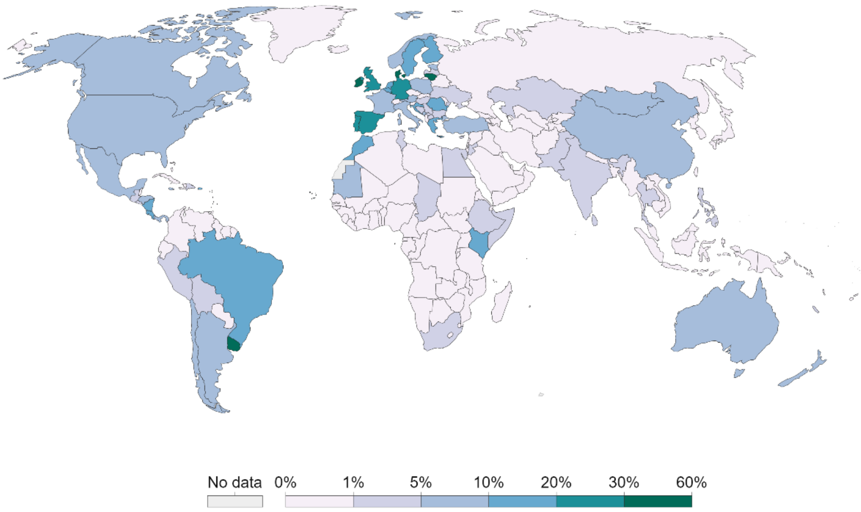


Figure 2-6: Share of electricity production from wind of the world, 2021 (Share of electricity production from wind, 2022)

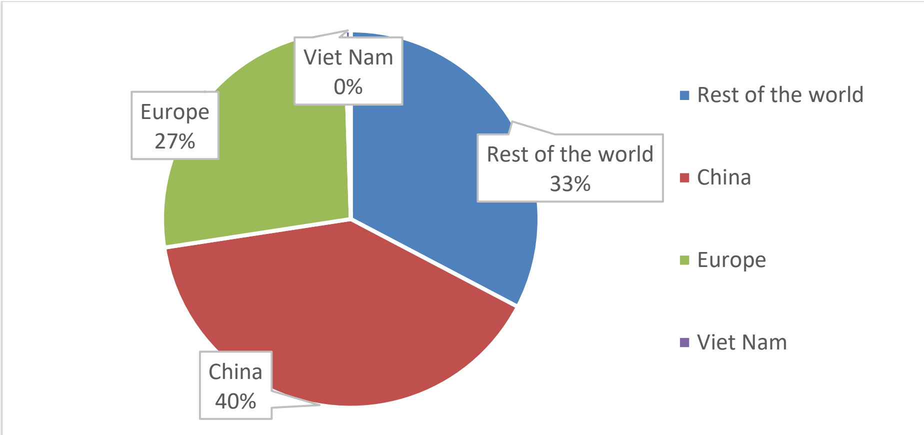


Figure 2-7: Cumulative wind capacity of China, Europe, Viet Nam, and the rest of the World (IRENA, 2022)

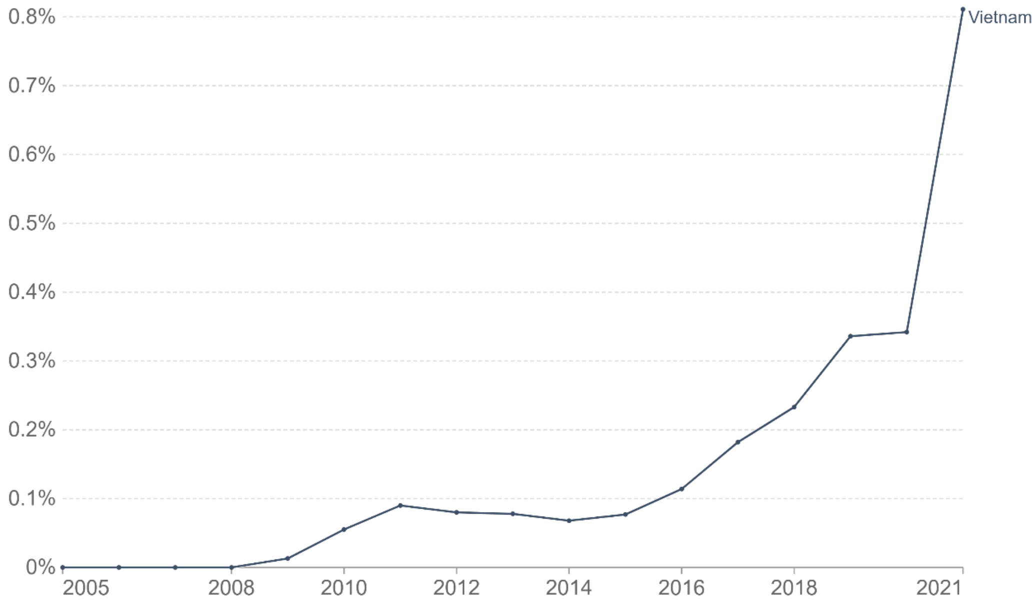


Figure 2-8: Share of electricity production from wind of Viet Nam (Share of electricity production from wind, 2022)

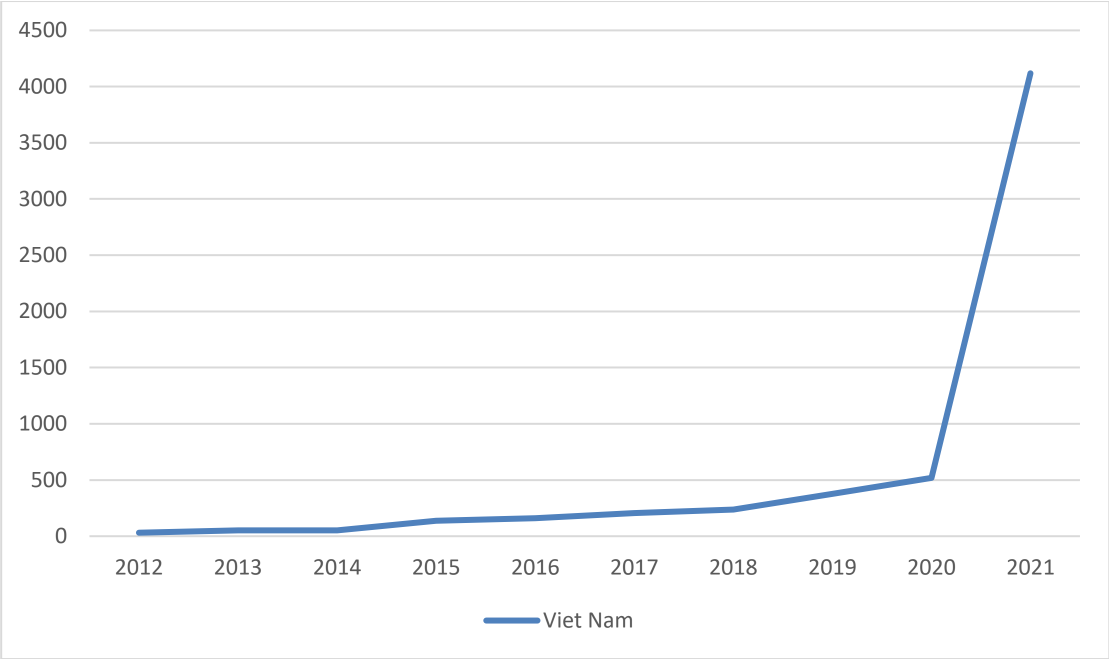


Figure 2-9: Cumulative wind capacity Viet Nam over 10 years (IRENA, 2022)

2.2 Time Series Forecasting

A time series is a series of data indexed in time order. Commonly, a time series is a discrete-time data, which is a sequence taken at successive equally spaced points in time. Time series analysis analyzing time series data in order to get statistics, which are meaningful, and other characteristics of the data too. The use of a model to predict future values based on previously observed values is called time series forecasting.

In this project, a statistical method and three machine learning methods will be applied. However, a pre-processing method will be used to make sure the data is reliable.

2.2.1 Preprocessing methods

Transforming raw data into understandable and useful per se, is called data preprocessing. Conflicting formatting, human errors, and incomplete is what real-world or raw data usually has. Data preprocessing tackles these problems and makes datasets completer and more efficient to perform data analysis.

It’s a vital step that can affect the success machine learning projects. Knowledge from datasets can be discovered faster and ultimately affect the performance of machine learning models.

There are 4 vital parameters:

- **Accuracy**: the information must be correct. A dataset’s accuracy can be affected by typos, redundancies, and old information.
- **Consistency**: contrary data may give you different answers to the same question. Thus, the data should have no conflicts
- **Completeness**: The dataset shouldn’t have blank fields. The access to a complete picture of the situation the data describes allows data scientists to perform accurate forecasting.
- **Validity**: Invalid datasets are hard to analyze or organize. if the data samples are of the right type, in the correct format, and are within a specified range, then it is considered as a valid dataset.

The are 4 steps in a pre-processing process:

- **Data cleaning**

Data cleaning, also known as cleansing, is the process of removing anomalies from datasets, accounting for missing values, resolving inconsistencies in the data, and smoothing noisy data. Data cleaning's main goal is to provide complete and accurate samples for machine learning models.

Missing data values are a rather prevalent issue. It could occur during data collection or as a result of a particular data validation rule. In such circumstances, you must gather more data samples or search for more datasets.

In this project, the missing values are refilled with linear interpolate formula, which is given as below:

$$y = y_a + (y_b - y_a) \times \frac{x - x_a}{x_b - x_a} \tag{2.1}$$

Noise is a big amount of useless data. More specifically, it's the random volatility in a measured variable or inaccurate attribute values in data. Data points that are duplicated or semi-duplicated, data segments that are useless for a certain research method, or undesirable information fields are all examples of noise.

- **Data integration**

Data integration is an essential component of data preparation because data may be gathered from many sources. Integration may result in numerous redundant and inconsistent data points, which would ultimately produce less accurate models.

There are several ways to integrating data:

  - Data consolidation: Information is physically brought together and put away in a single put. Having all information in one put increments productivity and efficiency. This step ordinarily includes utilizing information distribution center program.

  - Data virtualization: In this approach, an interface gives a bound together and real-time view of information from numerous sources. In other words, information can be seen from a single perspective.

  - Data propagation: Copying data from one location to another with the assistance of specific applications. This process is usually event-driven and can be synchronous or asynchronous.

- **Data reduction**

It provides a condensed version of the dataset. Although this step reduces the volume, it keeps the original data intact. This data preprocessing step is especially important when working with big data because the amount of data involved is massive.

The process of selecting a subset of features or attributes that contribute the most or are the most important is known as feature subset selection.

Dimensionality reduction, also referred to as dimension reduction, is the process of reducing the number of features or input variables in a dataset.

The dimensionality of a dataset is defined as the number of features or input variables. The more features there are, the more difficult it is to visualize the training dataset and create a predictive model.

Some ways to perform dimensionally reduction:

- Random forest: This technique is used to evaluate the significance of each feature in a dataset, allowing us to retain only the most important features.
- PCA: A statistical technique for extracting a new set of variables from a large set of variables. Principal components are the newly extracted variables. This method is only applicable to features with numerical values.
- Low variance filter: Removes normalized attributes with variances less than a certain threshold because minor changes in data translate to less information.
- Missing values ratio: This method removes attributes that have more missing values than a certain threshold.
- High correlation filter: A technique for detecting and removing highly correlated features; otherwise, a pair of highly correlated variables can increase the dataset's multicollinearity.

- **Data transformation**

Data transformation is the process of converting data from one format to another. In essence, it entails methods for converting data into appropriate formats from which the computer can learn efficiently.

Some data transformation strategies are as follows:

- Smoothing is a statistical approach that uses algorithms to remove noise from data. It can predict patterns and highlight the most valuable features in a dataset. It also entails removing outliers from the dataset to highlight the patterns.
- Gathering data from various sources and presenting it in a standardized format for data mining or analysis called aggregation. Aggregating data from multiple sources to increase the number of data points is critical because only then will the ML model have enough examples from which to learn.
- Discretization is the process of converting continuous data into smaller intervals. For example, it is more efficient to categorize people as "teen," "young adult," "middle age," or "senior" rather than using continuous age values.
- Converting low-level data features to high-level data features is what generalization entails. Categorical attributes, such as home address, can, for example, be generalized to higher-level definitions such as city or state.
- The process of converting all data variables into a specific range is referred to as normalization. In other words, it is used to scale an attribute's values so that they fall within a smaller range, such as 0 to 1. Data normalization methods include decimal scaling, min-max normalization, and z-score normalization.
- Feature construction is the process of creating new features from a given set of features. This method simplifies the original dataset and makes data analysis, mining, and visualization easier.

- Although it is not specified, concept hierarchy generation allows you to create a hierarchy between features. For example, if you have a house address dataset that includes information about the street, city, state, and country, you can use this method to organize the data in hierarchical forms.

2.2.2 Statistical method

Statistic is the science of using information discovered from collecting, organizing, and studying numbers (CambridgeDictionary, 22). In raw research data statistical analysis, Statistical methods are mathematical formulas, models, and techniques used.

Statistical methods extract information from research data and provide various ways to assess the robustness of research outputs. Two main statistical methods are used in data analysis: descriptive statistics and inferential statistics. These two methods using indexes like the mean or standard deviation to summarize data from sample and state assumptions from data which are subject from random variation (i.e., observational errors, sampling variation) respectively.

In this project, a time series analysis model will be use is ARIMA and its variant, ARIMAX (ARIMA with exogenous variables).

An ARIMA model can be considered as a special type of regression model--in which the dependent variable has been stationed and the independent variables are all lags of the dependent variable and/or lags of the errors--so it is straightforward in principle to extend an ARIMA model to incorporate information provided by leading indicators and other exogenous variables: you simply add one or more regressors to the forecasting equation.

ARIMA is very versatile and makes few presumptions. It is theoretically and statistically sound, and no a priori model postulation is required when analyzing failure data (Ho, 1998). The ARIMA model is a combination of AR (p) and MA (q) models with integrate parameter(d). In short, an ARIMA model has 3 parameters which are (p, d, q).

This is the Moving Average (q) model:

$$Y_t = \alpha + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q} \quad (2.2)$$

$$Y_t = \alpha + \sum_{i=1}^{q} \phi_i \epsilon_{t-i} \quad (2.3)$$

Whereas:

- Y_t is the value at time step t
- ϕ_1 is the coefficient of error 1
- ϵ_{t-1} is a white noise error term lag 1
- α is a constant

This is Auto Regression (p) model:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} + \epsilon_t \quad (2.4)$$

$$Y_t = \alpha + \epsilon_t + \sum_{i=1}^{p} \beta_i Y_{t-i} \quad (2.5)$$

Whereas:

- Y_t is the value at time step t
- Y_{t-1} is the lag 1 of the series
- β_1 is the coefficient of lag 1 that the model estimates
- α is a constant, also estimated by the model
- ϵ_t is a white noise error term

After differencing a time series, the next step in fitting an ARIMA model is determining whether AR or MA terms are required to correct any autocorrelation that remains in the differenced series. The autocorrelation function (ACF) and partial autocorrelation function (PACF) plots of the differenced series can be used to estimate the number of AR and/or MA terms required. If you've seen the ACF plot before it's simply a bar chart of the coefficients of correlation between a time series and its own lags. The PACF plot depicts the partial correlation coefficients between the series and its own lags.

Other alternative methods include AIC, BIC, etc. Akaike information criterion (AIC) is a useful criterion to determine the order of a non-seasonal ARIMA model. It can be written as

$$AIC = -2\log(L) + 2(p + q + k) \qquad (2.6)$$

Whereas

- L is the log-likelihood of the data,
- k represents the intercept of the ARIMA model

The corrected AIC for ARIMA model

$$AICc = AIC + \frac{2(p + q + k + 1) \times (3 + 5 + k + 2)}{N - p - q - k - 2} \qquad (2.7)$$

The Bayesian Information Criterion (BIC) can be written as

$$BIC = AIC + ((logT) - 2)(p + q + k) \qquad (2.8)$$

Minimize the AIC and BIC values for a good model is the aim when using these criterions. The AIC and the BIC are served two purposes which are not related. The AIC tries to estimate model toward the reality and the BIC wants to have the perfect fit. Real-life data is very complex; therefore, BIC usually being criticized for finding the perfection.

In ARIMA model, the integrate parameter (d) must be chosen first.

Table 2-1: Differencing orders

d = 0	$y_t = Y_T$
d = 1	$y_t = Y_t - Y_{t-1}$
d = 2	$y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2})$

The meaning of parameter d is that is consider if the series is stationary or not. One method is plotting the series after differencing, the other is using Augmented Dickey-Fuller (ADF) Test.

A unit root is present in a time series sample is the null hypothesis this test tests. The alternative hypothesis is usually stationarity of the series.

As this test is asymmetrical, we are only concerned with negative values of our test statistic. If the calculated test statistic more negative than the critical value, then the null hypothesis is rejected, and no unit root is present.

$$y_t = c + \beta t + \alpha y_{t-1} + \phi_1 \Delta Y_{t-1} + \phi_2 \Delta Y_{t-2} .. + \phi_p \Delta \mathrm{Y_{t-p}} + \mathrm{e_t} \qquad (2.9)$$

- ΔY_t is the 1st difference of the series at time (t-1)
- y_{t-1} is the lag 1 of time series
- α is the coefficient of the 1st lag on Y

After decided the integrate parameter (*d*), ARIMA (*p*, *d*, *q*) can present as

$$Y_t = \alpha + \epsilon_t + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q} \quad (2.10)$$

Which can also be written in short as

$$Y_t = \alpha + \epsilon_t + \sum_{i=1}^{p} \beta_i Y_{t-i} + \sum_{j=1}^{q} \phi_i \epsilon_{t-j} \quad (2.11)$$

This can state as predicted value Y at time step t equal to Constant plus Noise plus Linear combination Lags of Y (up to p lags) plus Linear combination of Lagged forecast errors (up to q lags).

The ARIMAX model is the ARIMA model with Exogenous Covariates

$$Y_t = \alpha + \epsilon_t + \sum_{i=1}^{p} \beta_i Y_{t-i} + \sum_{j=1}^{q} \phi_i \epsilon_{t-j} + \sum_{k=1}^{r} \theta_k x_{tk} \quad (2.12)$$

There are also other variants of ARIMA model such as Seasonal ARIMA (SARIMA) and SARIMAX. The wind is a seasonal component. However, this project will not use SARIMA models because the data is only one-year length. It is not enough “season” to be computed.

2.2.3 Machine learning method

Machine learning (ML) is a field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks (Mitchell, 1997). It is regarded as a component of artificial intelligence. Without being explicitly programmed to do make a decisions or predictions, Machine Learning Algorithms construct a model from sample data, referred as training data.

Machine learning divides into several groups:

- Supervised learning

- Unsupervised learning
- Reinforcement learning

The author mainly uses regression models, which belong to supervised learning group, in this project, and then ensemble the predictions.

2.2.3.1 Linear Regression

A Linear Regression line can represent the trending of the whole dataset as a linear line. The general form of Linear Regression can be written as

$$Y_i = f(X_i, \beta) + \epsilon_i \tag{2.13}$$

Whereas:

- Y_i is the dependent variable
- f is the function
- X_i is the independent variable
- ϵ_i are the error terms

The formula above can also be written as

$$Y_i = a + bX_i + \epsilon_i \tag{2.14}$$

With

$$b = \frac{N \sum XY - \sum(X) \times \sum(Y)}{N \sum X^2 - (\sum X)^2} \tag{2.14.1 }$$

$$a = \frac{\sum Y - b \sum X}{N} \tag{2.14.2}$$

Whereas:

- Y is the dependent variable
- a is population Y intercept
- b is population slope coefficient
- X is the independent variable
- ϵ_i are the error terms

In short, Linear Regression is a line that carries the trend of the data at large.

2.2.3.2 Support vector regression

Support Vector Regression (SVR, $\epsilon - SVR$ or epsilon – SVR) is a supervised learning algorithm for predicting discrete values. SVR's basic concept is to find the best fit line. The best fit line in SVR is the hyperplane with the greatest number of points.

Linear Regression and Support Vector Regression are similar in that the equation of the line is used. This straight line is known as a hyperplane in SVR. The closest data points on either side of the hyperplane are called Support Vectors, and they are used to plot the boundary line.

Unlike other Regression models, which seek to minimize the difference between the true and predicted values, the SVR seeks to fit the best line within a given range. The distance between the hyperplane and the boundary line is the threshold value. The fit time complexity of SVR is more than quadratic with the number of samples, making it difficult to scale to datasets with more than a few tens of thousands of samples.

The SVR model in this project used Radial basis function as the kernel. This one was chosen after several try and errors with other kernels (such as linear, sigmoid, …). The parameter C in this model is the regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared L2 penalty. The epsilon specifies the boundaries where points predicted inside these lines do not get penalty in the training loss function.

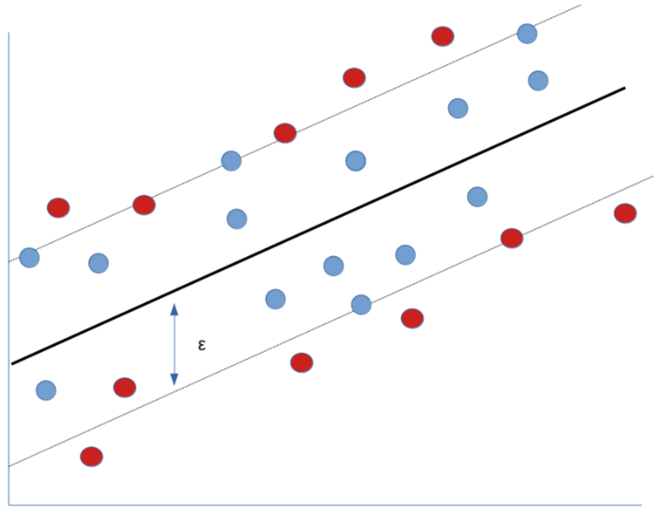


Figure 2-10: SVR model visualization (Source: Internet)

Nu-SVR uses a parameter nu (ν) to control the number of support vectors. Nu replaces the parameter epsilon of epsilon-SVR. To be more precise, epsilon is an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors. In addition, with probability 1, asymptotically, ν equals both fractions. The implementation is based on Libsvm (Chih-Chung Chang, 2002).

2.2.3.3 Ensemble

Ensemble learning is a general meta-machine learning approach that seeks to improve predictive performance by combining predictions from multiple models.
Although you can create an apparently infinite number of ensembles for your predictive modeling problem, three methods dominate the field of ensemble learning. So much so that, rather than being algorithms in and of themselves, each is a field of study that has spawned a plethora of more specialized methods.
Bagging, stacking, and boosting are the three main classes of ensemble learning methods. However, this project can only use bagging and stacking as the time coming to an end.

Bootstrap aggregation, or **bagging** for short, by varying the training data, it seeks a diverse group of ensemble members.

Bagging use a single machine learning algorithm and training each model on a different sample of the same training dataset. The algorithm used almost always a decision tree but can be change to others. Such as, in this project, the algorithm used is Nu-SVR, not decision tree. After training, voting or averaging are methods that use for combining results. The way each sample of the dataset is prepared to train ensemble members is critical to the method. Each model receives a distinct sample of the dataset.

The diversity in the ensemble is ensured by the variations within the bootstrapped replicas on which each classifier is trained, as well as by using a relatively weak classifier whose decision boundaries measurably vary with respect to relatively small perturbations in the training data (Cha Zhang, 2012).

This is known as a bootstrap sample. It is a technique used in statistics to estimate the statistical value of a data sample with small datasets. A better overall estimate of the desired quantity can be obtained by preparing multiple different bootstrap samples and estimating a statistical quantity and calculating the mean of the estimates rather than simply estimating from the dataset directly.

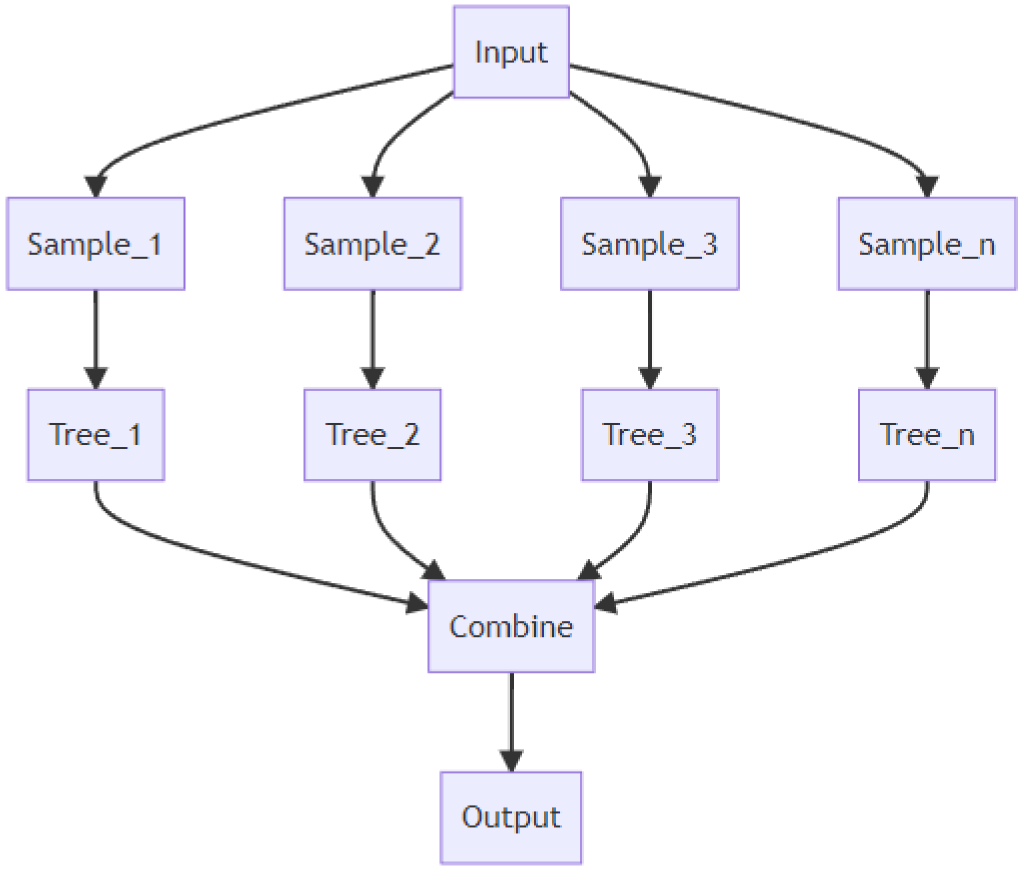


Figure 2-11: Visualized bagging ensemble model

The Bagging method will be used in this project are:

- **Bagging Regression** will fit base regressors each on random subsets of the original dataset and then aggregate their individual predictions to form a final prediction. Permuted congruential generator (PCG) is used to create random subsets.
- **Voting Regression** will fit several base regressors, each on the whole dataset. Then it averages the individual predictions to form a final prediction.
- **Random Forest Regression** will fit a number of classifying decision trees on various sub-samples of the dataset. To control over-fitting as well as increase the accuracy, the predictions will be averaged. The whole dataset is used to build each tree. Permuted congruential generator (PCG) is used to create random subsets.

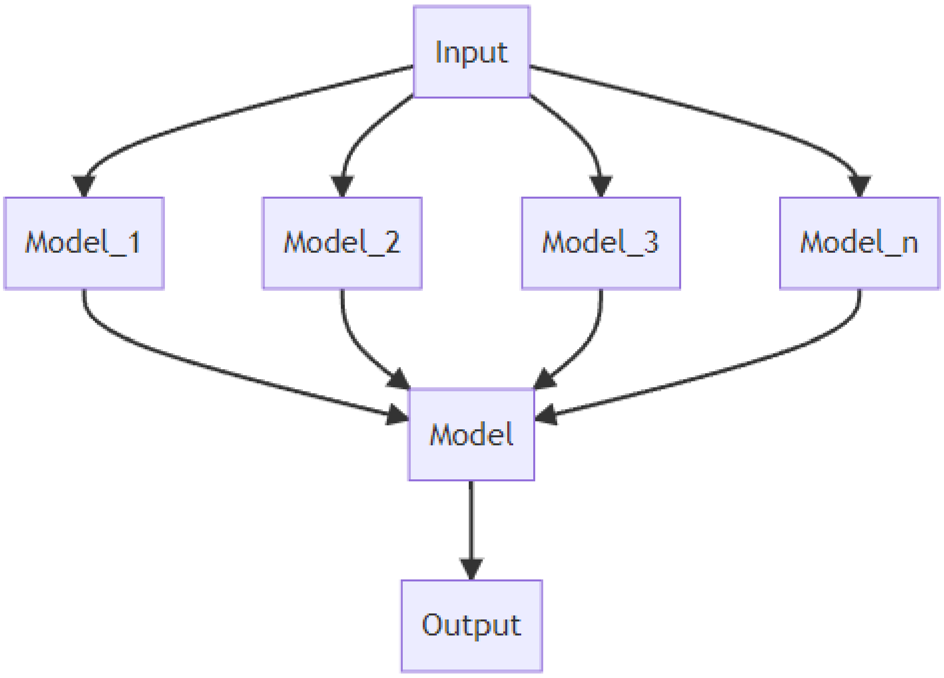


Figure 2-12: Visualized stacking ensemble model

Stacked Generalization, or stacking for short, is an ensemble method that seeks a diverse group of members by varying the model types fit on the training data and using a model to combine predictions.

Stacking has its own terminology, with ensemble members referred to as level-0 models and the model used to combine the predictions referred to as a level-1 model. The most common approach is a two-level model hierarchy, though more layers of models can be used. Instead of a single level-1 model, we could have three or five level-1 models and a single level-2 model that combines the predictions of the level-1 models to make a prediction.

Any machine learning model can be used to aggregate the predictions, but linear models, such as linear regression for regression and logistic regression for binary classification, are commonly used. This encourages the model's complexity to reside at the lower-level ensemble member models and simple models in order to learn how to harness the variety of predictions made.

Using trainable combiners, it is possible to determine which classifiers are likely to be successful in which part of the feature space and combine them accordingly. (Cha Zhang, 2012). The only stacking model will be used in this project is the Stacking Regressor.

2.2.4 Forecast evaluation

2.2.4.1 Predict data

Since the data only have real value, which means using real data to forecast is unrealistic. Hence, the solution for this situation is give the input predict data some errors (H. Bludszuweit, 2008). The errors will be distributed in normal distribution with maximum values are 10, 20, ...50. Each group of error (5 sets) will be given to the test data and keeping the same throughout the whole project.

2.2.4.2 Evaluation

Model evaluation is the process of analyzing a machine learning model's performance, as well as its strengths and weaknesses, using various evaluation metrics. Model evaluation is necessary to assess the efficacy of a model during the early stages of research, and it also plays a role in model monitoring.

Standard metrics root means square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), normalized RMSE (NRMSE), and normalized mean absolute error or the author would like to call it mean relative error (MRE), are adopted to evaluate the performance of deterministic forecasts. They are defined by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(P_{fore} - P_{true}\right)^2}{n}} \tag{2.15}$$

$$NRMSE = \frac{1}{P_{cap}} \times \sqrt{\frac{\sum_{i=1}^{N}\left(P_{fore} - P_{true}\right)^2}{N}} \tag{2.16}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|P_{fore} - P_{true}| \tag{2.17}$$

$$MRE = \frac{1}{N}\sum_{i=1}^{N}\frac{|P_{fore} - P_{true}|}{P_{cap}} \tag{2.18}$$

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{P_{true} - P_{fore}}{P_{true}}\right| \tag{2.19}$$

Whereas:

- P_{fore} is estimated value
- P_{true} is the actual value
- P_{cap} is the capacity for wind power generation

For these metrics, a smaller value indicates better performance. Usually, it is seen that the 1-h-ahead NMAE, MRE, and MAPE are in the range of 3–5%, 4–7%, and 18–40%, respectively (Sun, 2019).

CHAPTER 3. IMPLEMENTATION

3.1 Program

In this project, the program that the author am using for basically every task is Python. Python is a general programming language that is easy to learn and have huge community to support me. Statmodels and Scikit-learn is the two libraries which the author uses for all models in this project. The visualization used Matplotlib library and Seaborn library for wonderful graphs.

3.2 Data

- Source: Changhua, Taiwan (given by NCKU)
- Time: 01-Jan-20 to 31-Dec-20
- Resolution: 10 minutes
- Data's capacity: 52704
- WT's capacity: 2000 kW
- WT type: 3 blades
- WT location: offshore

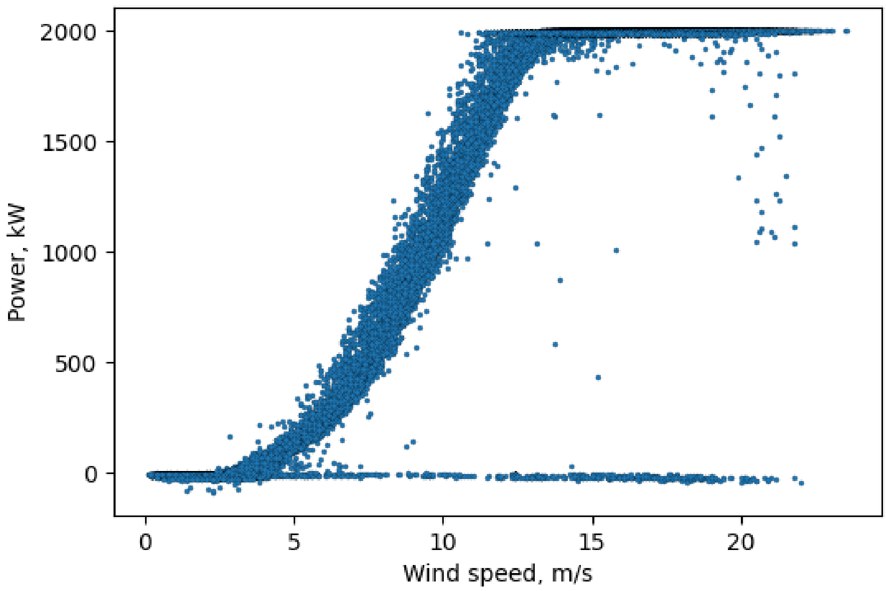


Figure 3-1: Raw data

3.3 Data Preparation

- Missing values

In this project, there are lots of bad data, which are removed in the pre-processing process

- Label as bad data (from the data set) and missing values. These data contain string "Not good for calculation", thus when load into the program, they are not considered as number types (integer, float, …) but string type. Hence, it can easily be selected and convert to NaN (not a number), which will be clean up after other steps.

- Negative power data, which is considered as bad data. The author set the condition that if power is less than zero, then it will be converted to NaN.

- Data located in right hand side of the S curve (Figure 3-1). To deal with these "bad" data, the author set some conditions follow by the order below:

1. Power > 1800 (kW) and power < 1950 (kW), then:

$$Power > Speed \times 280 - 2520 \rightarrow \begin{cases} True \rightarrow Power = Power \\ False \rightarrow Power = NaN \end{cases}$$

2. Power < 1800 (kW), then:

$$Power > Speed \times 280 - 1960 \rightarrow \begin{cases} True \rightarrow Power = Power \\ False \rightarrow Power = NaN \end{cases}$$

3. Power > 240 (kW) and power < 800 (kW), then:

$$Power > Speed \times 280 - 840 \rightarrow \begin{cases} True \rightarrow Power = NaN \\ False \rightarrow Power = Power \end{cases}$$

4. Power > 0 (kW) and power < 400 (kW), then:

$$Power > Speed \times 130 - 780 \rightarrow \begin{cases} True \rightarrow Power = Power \\ False \rightarrow Power = NaN \end{cases}$$

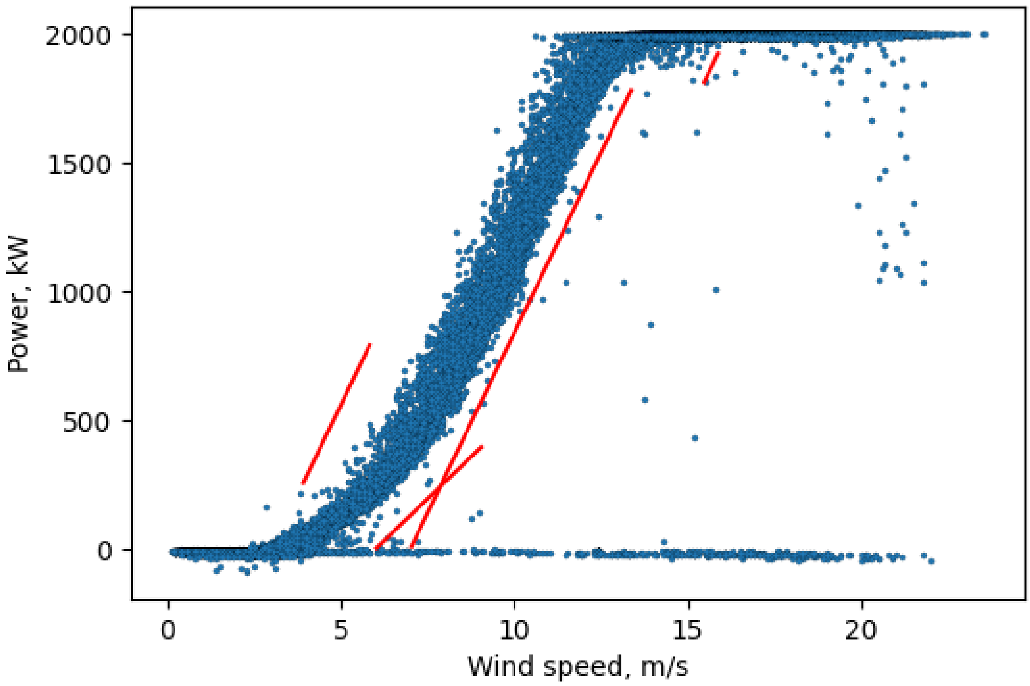


Figure 3-2: Raw dataset with condition lines

After removing, some missing part can be re-fill by the condition of the length of the continuous missing list are not greater than 5, this number being chosen because of it is smaller than 6, which is equal to an hour in the dataset. The author used the Linear Interpolation method, using the formula (2.1), i.e.

$$y = y_a + (y_b - y_a) \times \frac{x - x_a}{x_b - x_a} \tag{3.1}$$

After interpolating, the author run the command follow conditions above again to make sure there is no interpolated value that is out of those conditions. That explains the reason the plot has a line on the left of data. After interpolating, there are some values showed in that part.

To finish the preparation, the author now removes all NaN data (which are create by above steps), the result is illustrated in the figure below.

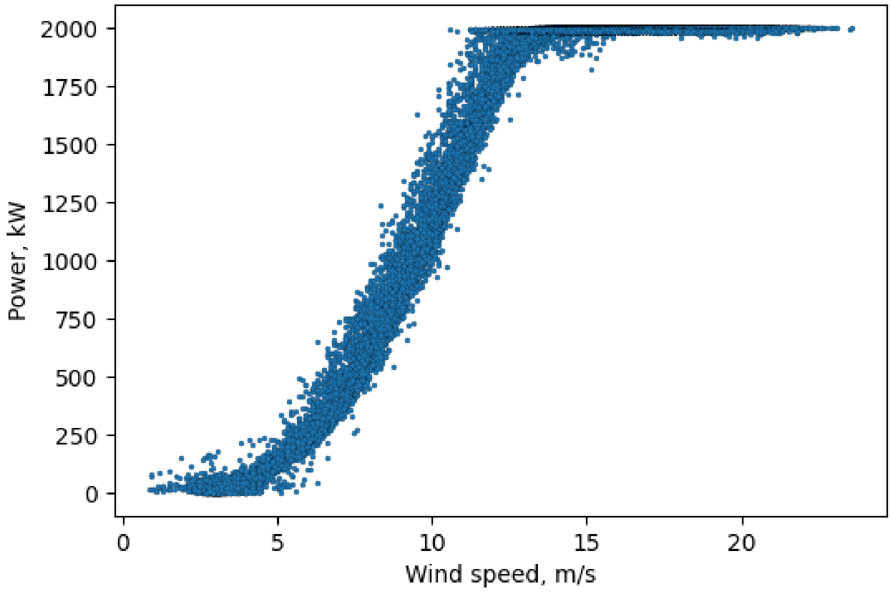


Figure 3-3: Data after cleaning

- Feature engineering

The given data have the information of Wind Power generated, Wind Speed and Wind Direction at a time which separate by 10 minutes. With the methods used in this project, excluding Wind Direction is enough.

- Normalization

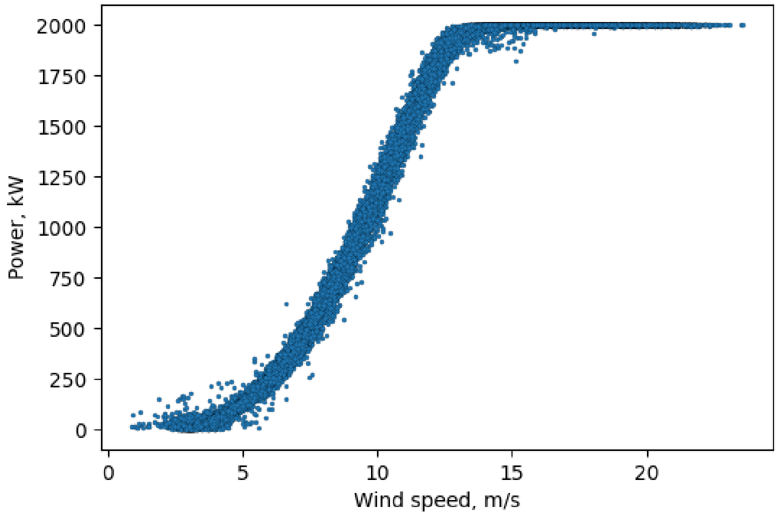
This project actually does not use normalization in working process because it is not needed. Normalization is needed when the input values is too different (e.g.: x_1 can have the range from 0 to 2000, and the other input x_2 have the range from 0 to 10). In this project, the input value is only wind speed, so the normalization is not needed.

- Training and Test Data

In this project, every model will use 32400 data to train, equivalent to 225 days of data. And use 432 data to test, equivalent to 3 days of data. However, the test data will be applied 10%, 20%, …, 50% of error to see how well the models handle error (Sun, 2019).

The training data is selected from January 1st, 2020, to August 13th, 2020. This dataset is showed in the figure 3-4 below.

And the testing data is 3 days after that, which are August 14th to August 17th.



\

Figure 3-4: Data which is used for training

The testing data is real data at the moment, that why it is not being used to test. But must be applied the error to the input. The reason is that when forecasting, the wind speed data is also being forecast by the supplier, and then given to the user. The error giving to the test input are generated by using permuted congruential generator (PCG), which is and pseudo random generator algorithm, and distributed by Gaussian distribution (probability density function) with the mean $\mu = 0$ and standard deviation $\sigma = percent \div 3.5$ with the percent equals to 10%, 20%…50%. It is divided by 3.5 so that the maximum values of error will not surpass the percent value given.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \times e^{\frac{-1}{2}\times\left(\frac{x-\mu}{\sigma}\right)^2} \tag{3.2}$$

With 10% error, the density function shows as

$$f(x) = \frac{1}{\frac{0.1}{3.5}\sqrt{2\pi}} \times e^{\frac{-1}{2}\times\left(\frac{x}{0.1/3.5}\right)^2} \tag{3.3}$$

And illustrated in the figure below

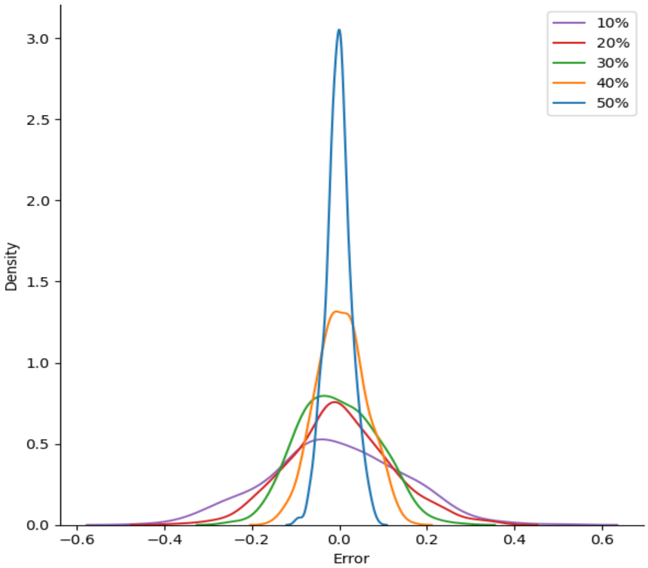


Figure 3-5: The densities of error sets that will be given to the test data

3.4 Implementation

3.4.1 Statistical method

With ARIMA method, first the author needs to find the *d* parameter, which can be decided by observations or tests.

The Figure 3-6 show that the first order differencing is kind of station. The first plot is the original power by indexing, the second plot is the first order differencing by indexing.

The first order differencing calculates by follow:

$$y_t = Y_t - Y_{t-1} \tag{3.4}$$

e.g., $y_2 = Y_2 - Y_1, \ldots$

Figure 3-6 only show from the second value since Y_0 is not exist, leads to y_1 is also not exist.

However, to be sure about this decision, the author decided to have an ADF test with the first order differencing series. The result shows in the Figure 3-7, the *ADF statistic* is way below the critical values and the *p-value* is smaller than 5% (follow by the definition of the ADF test, which is stated in chapter 2, equation 2.8). That is an acceptable result. Hence, the parameter *d* in ARIMA model is chosen as 1 (means differencing 1 time is using for further calculations).

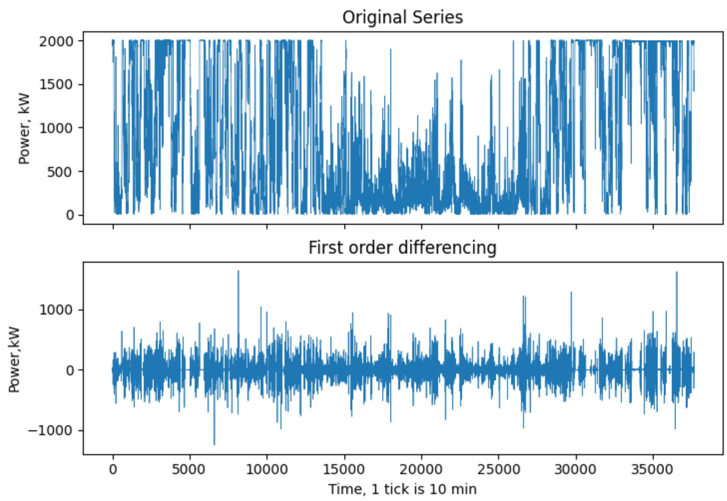


Figure 3-6: Original and First order differencing of the series

ADF Statistic: -31.577966

p-value: 0.000000

used lag: 53.000000

The number of observations: 37594.000000

Critical Values:

1%, -3.4305239571890094

Critical Values:

5%, -2.8616168849458075

Critical Values:

10%, -2.566810923411166

Figure 3-7: ADF test result

After having the *d* parameter, the author now finds the *q* parameter with ACF plot. To create an ACF plot, first, the author needs to find the mean of the series

$$\overline{P} = \frac{\sum_{i=1}^{N} P_i}{N} = 782.58\ (kW)\ with\ N = 32400 \qquad (3.5)$$

The autocovariance function at lag k with the average power that has just been calculated

$$s_k = \frac{1}{n}\sum_{i=k+1}^{N}(P_i - \overline{P})(P_{i-k} - \overline{P})$$
$$= \frac{1}{n}\sum_{i=k+1}^{N}(P_i - 782.58\ )(P_{i-k} - 782.58\ ) \tag{3.6}$$

With square deviations

$$\sigma^2 = \left\|P - \overline{P}\right\|^2 = \|P - 782.58\ \|^2 \tag{3.7}$$

Thus, variance is

$$s_0 = \frac{\sum_{i=1}^{N}\sigma^2}{N} = \frac{\sum_{i=1}^{N}\sigma^2}{32400} = 563991.88\ (kW) \tag{3.8}$$

The autocorrelation function (ACF) at lag k, for k ≥ 0, of the time series is defined by

$$r_k = \frac{s_k}{s_0} = \frac{s_k}{563991.88} \tag{3.9}$$

$$r_k = \frac{\frac{1}{N}\sum_{i=k+1}^{N}(P_i - 782.58\ )(P_{i-k} - 782.58\ )}{563991.88} \tag{3.10}$$

E.g., for k=3, equation 3.10 is written as

$$r_3 = \frac{1}{32400 \times 563991.88} \times ((P_4 - 782.58)(P_1 - 782.58)$$
$$+ (P_5 - 782.58)(P_1 - 782.58) + \cdots + (P_{32400} - 782.58)(P_1 - 782.58))$$

The calculations were done by computer, given in the table below, and illustrated in the figure.

Table 3-1: ACF table

STT	ACF	STT	ACF	STT	ACF	STT	ACF
1	1	6	-0.00197	11	-0.02517	16	-0.00815
2	0.026991	7	-0.00933	12	0.00558	17	-0.00326
3	-0.03875	8	-0.00962	13	-0.00437	18	0.004463

4	-0.00993	9	-0.02341	14	-0.01097	19	0.001756
5	0.007655	10	-0.01097	15	-0.00432	20	-0.0074

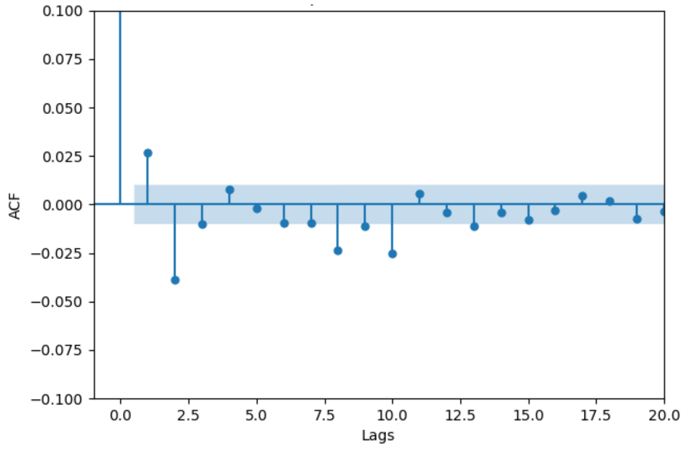


Figure 3-8: ACF plot of first order differencing series

The number of lags is chosen to minimize the corresponding information criterion, in this case it is minimize (AIC).

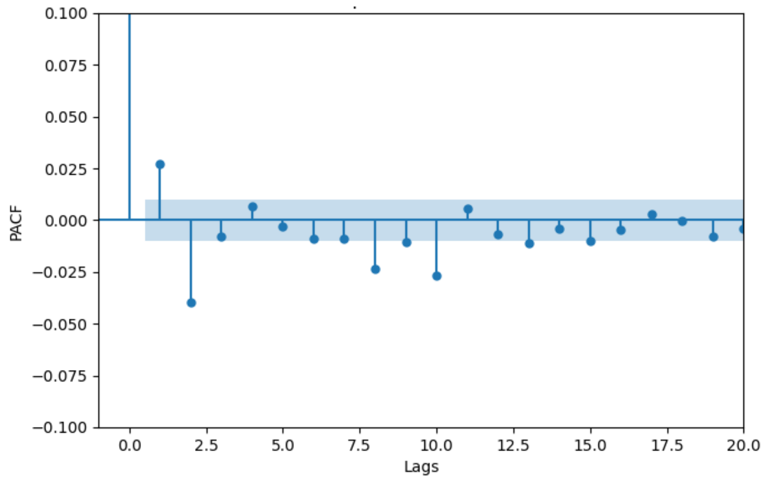


Figure 3-9: PACF plot of first order differencing series

PACF is the relationship between the values P_i and P_j, remove all P_k with k run form i to j. It can be written similarly to ACF, however the number using in the calculation below (782.58, 563991.88) is approximate because when remove some P from the calculating, these values will also change. Notwithstanding, the length of the series is so greater than 1, 2 digits number. Thus, remove some of these values does not affect too much to the mean or the variance. However, when the computer calculates, it will be doing more precisely than this equation.

$$r_k = \frac{\frac{1}{n}\sum_{i,j}^{N}(P_i - 782.58)(P_j - 782.58)}{563991.88} \tag{3.11}$$

E.g., with k=3, the PACF is written as

$$r_3 = \frac{1}{32400 \times 563991.88} \times ((P_3 - 782.58)(P_1 - 782.58) + (P_4 - 782.58)(P_1 - 782.58) + \cdots + (P_{32400} - 782.58)(P_1 - 782.58))$$

And since the mean and variance do not change too much, the different between ACF and PACF values in several first values are very look-a-like. That's explain why the ACF and PACF plots are so similar.

Table 3-2: PACF table

STT	PACF	STT	PACF	STT	PACF	STT	PACF
1	1	6	-0.00307	11	-0.02655	16	-0.0098
2	0.026992	7	-0.00874	12	0.005817	17	-0.00449
3	-0.03951	8	-0.00921	13	-0.00674	18	0.003128
4	-0.0078	9	-0.02375	14	-0.0109	19	-0.00029
5	0.006635	10	-0.01056	15	-0.00432	20	-0.00775

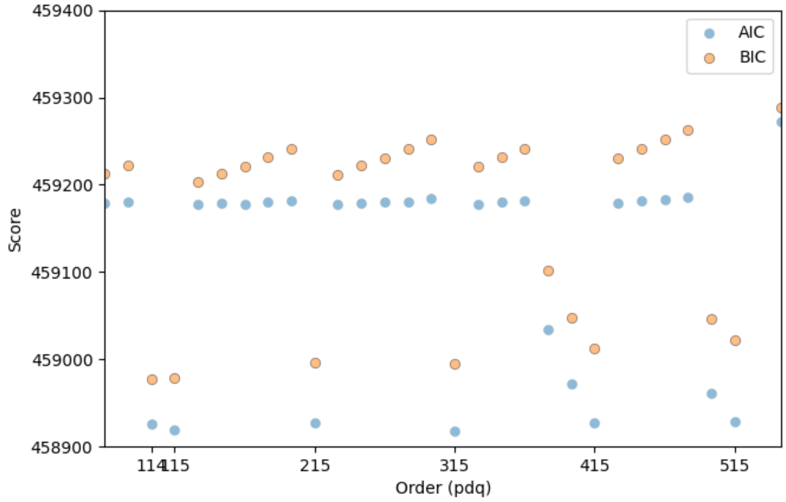


Figure 3-10: AIC and BIC scores of several ARIMA models

The author want to make sure that the order he chooses it good (or even the best), he calculates the AIC and BIC score for plenty of ARIMA order.

First, calculate maximum log-likelihood by formula

$$L = ln\left[\prod_{i=1}^{N} ARIMA(p,d,q)(Speed)\right] \tag{3.12}$$

Then, AIC, using L that has just been calculate

$$AIC = -2\log(L) + 2(3+5+k) \tag{3.13}$$

And AICc as

$$AICc = AIC + \frac{2(3+5+k+1)\times(3+5+k+2)}{32400-3-5-k-2} \tag{3.14}$$

Finally, BIC with

$$BIC = AIC + ((log32400) - 2)(3+5+k) \tag{3.15}$$

With *k* represents the intercept of the ARIMA model

These calculations are big and expensive (time), that why the author leave it to computer.

The result of these calculation given the smallest AIC value 458995, which belongs to order (3,1,5). This value different with other orders with same value *d* and *q* is in range of 1 to 2 digits, which are very small. However, the adviser said that the author should use this order (3,1,5). And showed in figure (the y axis is the score of AIC and BIC, the x axis is the order p, d, q but written as pdq e.g., 3, 1, 5 is written as 315 because it is easier to plot for the author)

In conclusions, this is the ARIMA model using table 2-1 and equation 2.12

$$Y_t = y_t - y_{t-1}$$

$$Y_t = \alpha + \epsilon_t + \sum_{i=1}^{3} \beta_i Y_{t-i} + \sum_{j=1}^{5} \phi_i \epsilon_{t-j}$$

With the estimates by computer (show in figure 3-11), the author has the ARIMA (3, 1, 5) equation:

$$\begin{aligned} Y_t = & -0.3987Y_{t-1} + 0.5998Y_{t-2} + 0.7487Y_{t-3} + 0.4484\epsilon_{t-1} \\ & - 0.6258\epsilon_{t-2} - 0.8224\epsilon_{t-3} - 0.027\epsilon_{t-4} + 0.0302\epsilon_{t-5} \end{aligned} \tag{3.16}$$

Dep. Variable:	y	No. Observations:	32400
Model:	ARIMA(3, 1, 5)	Log Likelihood	-196648.320
Date:	Mon, 08 Aug 2022	AIC	393314.641
Time:	08:43:54	BIC	393390.114
Sample:	0	HQIC	393338.769
	- 32400		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.3987	0.115	-3.473	0.001	-0.624	-0.174
ar.L2	0.5998	0.081	7.398	0.000	0.441	0.759
ar.L3	0.7487	0.091	8.191	0.000	0.570	0.928
ma.L1	0.4484	0.115	3.907	0.000	0.223	0.673
ma.L2	-0.6258	0.078	-8.017	0.000	-0.779	-0.473
ma.L3	-0.8224	0.093	-8.818	0.000	-1.005	-0.640
ma.L4	-0.0270	0.008	-3.359	0.001	-0.043	-0.011
ma.L5	0.0302	0.005	6.657	0.000	0.021	0.039
sigma2	1.102e+04	34.234	321.979	0.000	1.1e+04	1.11e+04

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	226427.05
Prob(Q):	0.99	Prob(JB):	0.00
Heteroskedasticity (H):	0.85	Skew:	0.36
Prob(H) (two-sided):	0.00	Kurtosis:	15.93

Figure 3-11: ARIMA (3, 1, 5) summary

And the ARIMAX (3, 1, 5) model

$$Y_t = \alpha + \epsilon_t + \sum_{i=1}^{3} \beta_i Y_{t-i} + \sum_{j=1}^{5} \phi_i \epsilon_{t-j} + \sum_{k=1}^{1} \theta_k x_{tk}$$

With the estimates by computer, the author has

$$\begin{aligned} Y_t = & -0.4251Y_{t-1} + 0.5505Y_{t-2} + 0.7704Y_{t-3} + 0.3104\epsilon_{t-1} \\ & - 0.6728\epsilon_{t-2} - 0.7717\epsilon_{t-3} + 0.0925\epsilon_{t-4} + 0.0463\epsilon_{t-5} \\ & + 126.9251x_t \end{aligned} \tag{3.17}$$

Dep. Variable:	y	No. Observations:	32400
Model:	ARIMA(3, 1, 5)	Log Likelihood	-181561.507
Date:	Mon, 08 Aug 2022	AIC	363143.014
Time:	08:44:58	BIC	363226.872
Sample:	0	HQIC	363169.823
	- 32400		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
x1	126.9251	0.261	485.826	0.000	126.413	127.437
ar.L1	-0.4251	0.105	-4.032	0.000	-0.632	-0.218
ar.L2	0.5505	0.074	7.464	0.000	0.406	0.695
ar.L3	0.7704	0.104	7.379	0.000	0.566	0.975
ma.L1	0.3104	0.105	2.943	0.003	0.104	0.517
ma.L2	-0.6728	0.078	-8.632	0.000	-0.826	-0.520
ma.L3	-0.7717	0.113	-6.855	0.000	-0.992	-0.551
ma.L4	0.0925	0.009	10.046	0.000	0.074	0.111
ma.L5	0.0463	0.005	8.996	0.000	0.036	0.056
sigma2	4275.9363	18.421	232.122	0.000	4239.832	4312.041

Ljung-Box (L1) (Q):	0.02	Jarque-Bera (JB):	46275.46
Prob(Q):	0.89	Prob(JB):	0.00
Heteroskedasticity (H):	0.74	Skew:	0.20
Prob(H) (two-sided):	0.00	Kurtosis:	8.84

Figure 3-12: ARIMAX (3, 1, 5) summary

3.4.2 Machine Learning method

The Machine Learning method are well written in the scikit-learn python library. The table below show the parameters the author used for each model.

For Linear Regression model, the author calculates the intercept and the slope as formula 2.13. This can also be calculated by the programs (which will have more precise result), and that is:

$$Power = 161.59 \times Speed - 598.88 \tag{3.18}$$

However, to make sure that

The author limits that linear function above with constrains

$$Power = 161.59 \times Speed - 598.88 \;\; when \; \begin{cases} Power \leq 2000 \\ Power \geq 0 \end{cases} \tag{3.19}$$

If these constrains is not pleased, then Power will be equal to it closest boundary (i.e., Power equals to 2000 kW if the original data greater than 2000 kW or equal to 0 if the original is less than 0 kW)

The regression line shows in the figure below

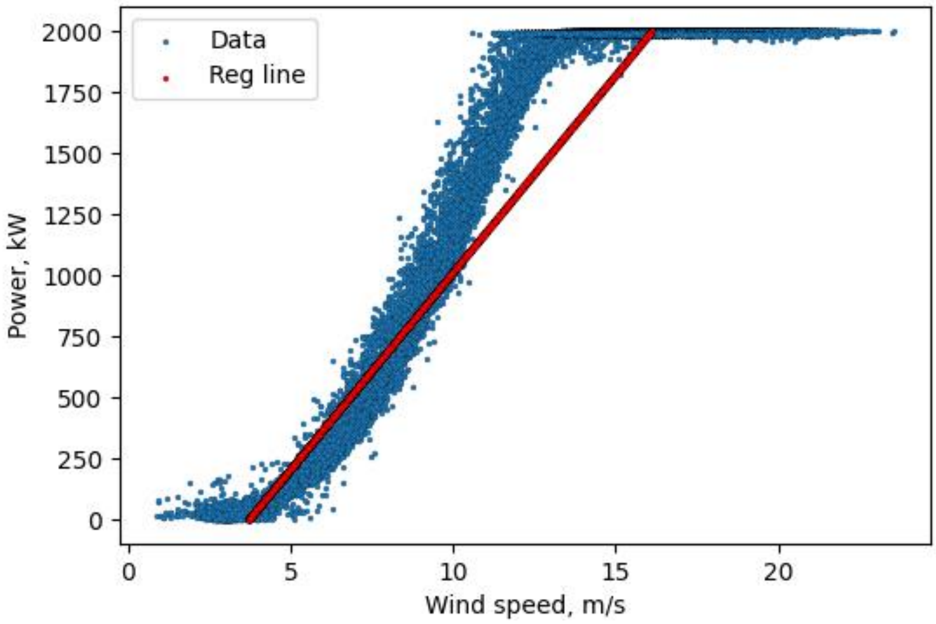


Figure 3-13: Regression line

When predict, the speed value is given, therefor the author only needs to put it in the equation above to calculate the power. The result of evaluating this model with test data showed in chapter 4.

The SVR model have kernel RBF, this is chosen among other kernels i.e., poly, sigmoid and linear. It is based on trial and error. The RBF simply given the best results with same parameter for input. The tolerance is given because….

To calculate with RBF kernel, first, calculate the variance, which is the average of the squared deviations from the mean.

Square deviations from mean calculate by the formula,

$$\sigma^2 = \left\| a - \frac{\sum_{i=1}^{N} a}{N} \right\|^2 \tag{3.20}$$

with N = 32400 (number of observations)

$$\sigma^2 = \|a - 8.55\|^2 \tag{3.21}$$

Thus, variance is

$$var(x) = \frac{\sum_{i=1}^{N} \sigma^2}{N} \tag{3.22}$$

with N = 32400, we have

$$var(x) = 19.825 \tag{3.23}$$

Therefore, gamma equal to

$$\gamma = \frac{1}{N \times \ var(x)} = \frac{1}{32400 \times 19.825} = 3.8 \times 10^{-7} \tag{3.24}$$

Radial Basic Function kernel

$$K(x, x') = \exp(-\gamma \times \|x - x'\|) = \exp\left(-3.8 \times 10^{-7} \times \|x - x'\|\right) \tag{3.25}$$

With $\|x - x'\|$ is the distance between $x\ and\ x'$. Therefore, with N values of x, we have Correlation Matrix

$$K_{i,j}\left(x_i, x_j\right) = \exp\left(-3.8 \times 10^{-7} \times \left\|x_i - x_j\right\|\right) with\ i,j\ run\ from\ 0\ to\ N \tag{3.26}$$

$$\vec{K} = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1j} \\ K_{21} & K_{22} & & K_{2j} \\ \vdots & & \ddots & \vdots \\ K_{i1} & K_{i2} & \cdots & K_{ij} \end{bmatrix}$$

Because $K_{12} = K_{21}$, etc. so it can be written as

$$\vec{K} = [K_1 \quad K_2 \quad K_3 \quad \cdots K_N]$$

Then, we have

$$\vec{Y} = \vec{\alpha} \times \vec{K} + b \tag{3.27}$$

With $\vec{Y}$ vector values that correspond to the training set, and $\vec{\alpha}$ is the unknown set that needs to be solved, and b is the intercept, which will also be calculated by the tool.

When calculate $\vec{\alpha}$ and b, the tool use $\epsilon = 0.1$ as the boundaries or the width of the tube around the line, meanwhile, regularization parameter C allows us to assign the weight to "slack," telling the algorithm how much we care about the error. In this model, the author chooses $C = 1$. This is also the default value; it means that the error is being care too much or too little.

The tool gives the author the intercept

$$b = 1195.88$$

And the support vectors $\vec{\alpha}$, some are showed in table below

Table 3-3: SVR support vectors

Index	**Support vector**
1	14.81
2	13.79
3	14.44
…	…
32246	11.40
32247	12.26
32248	12.05

In the table, we have 32248 support vectors, the reasons are

- The author set the boundaries too small, this happen because when applying this model, the deep understanding is not yet succeeded.
- There are only 152 data inside the "tube".

The values of the support vectors are for each of data outside the "tube", it will have a support vector with that length.

After the tool calculate $\vec{\alpha}$ is known, we can now estimate new power Y^* for the speed x^* as

$$Y^* = \vec{\alpha} \times \vec{k} + 1195.88 \quad (3.28)$$

With k

$$k_i(x_i, x^*) = \exp(-3.8 \times 10^{-7} \times \|x_i - x^*\|)\ with\ i\ run\ from\ 0\ to\ N \quad (3.29)$$

It means that when given a value of speed, it will find the nearest speed data and apply the same regression function that that neighbor used.

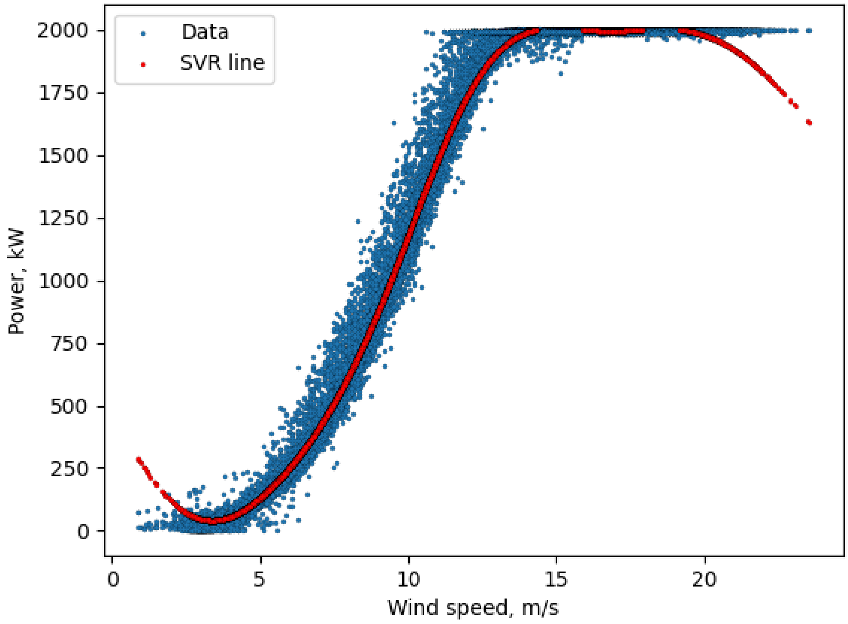


Figure 3-14: SVR line

As it is difficult to select an appropriate ϵ, introduced a new parameter ν that lets one control the number of support vectors and training errors. To be more precise, they proved that ν is an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors. In addition, with probability 1, asymptotically, ν equals both fractions.

In this model, the author chooses $\nu = 0.5$ because it will set half of the data inside the tube, the other half will not, and the author deems that are reasonable. However, $C = 10000$ is chosen because in this model

$$C_\epsilon = \frac{C_\nu}{N}$$

It is mean that if this is used in the SVR model above, it will be

$$C_\epsilon = \frac{10000}{32400} = 0.308$$

If you have a lot of noisy observations, you should decrease it: decreasing C corresponds to more regularization.

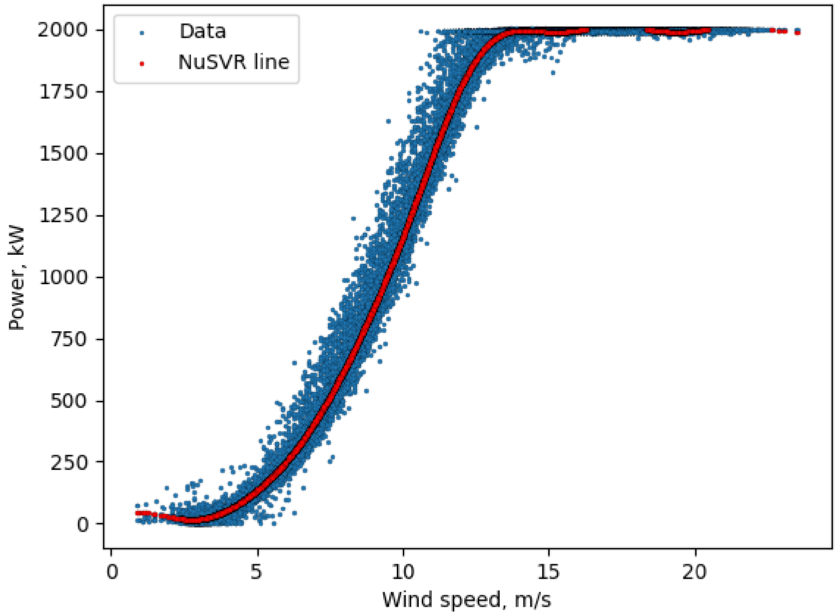


Figure 3-15: NuSVR line

The equation of NuSVR model

$$Y^* = \vec{\alpha} \times \vec{k} + 1087.35 \tag{3.30}$$

With $\vec{\alpha}$

Table 3-4: NuSVR support vectors

Index	Support vectors
1	13.87
2	13.65
3	13.40
…	…
16026	11.40
16027	12.26
16028	12.05

After having all above model, the author now doing the ensemble methods.
A random forest regressor is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. This model creates random subset by using PCG and apply, for each subset, model Decision Tree to estimate.

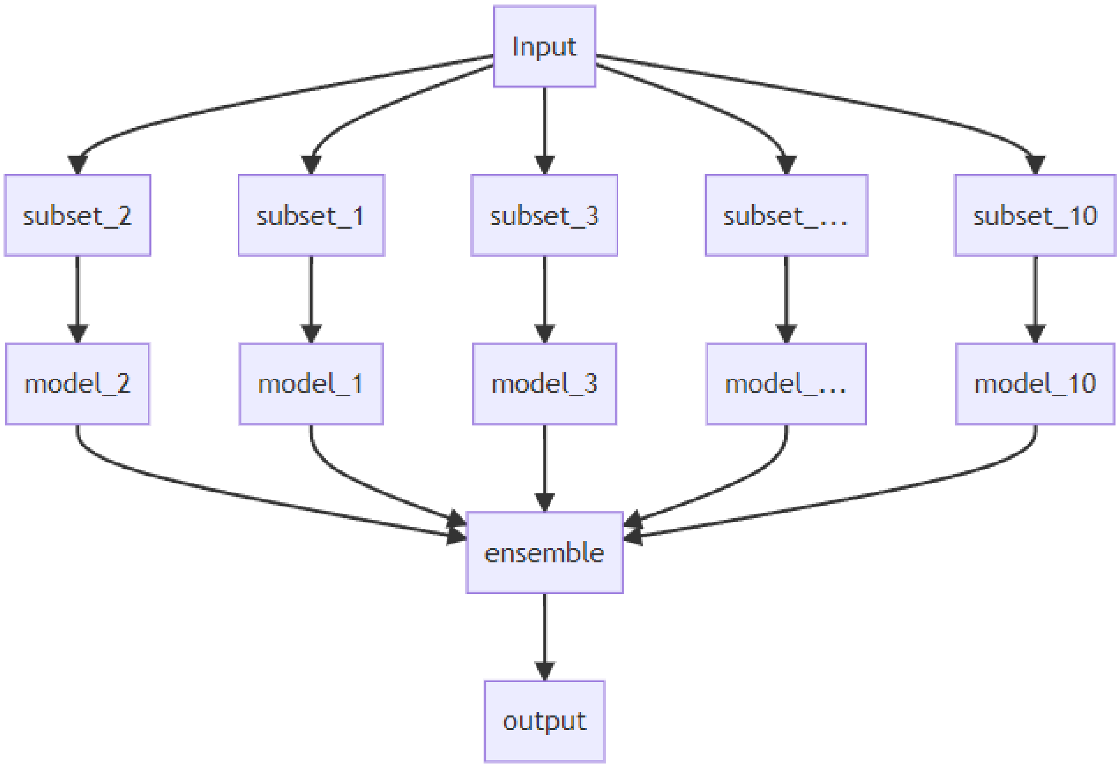


Figure 3-16: Bagging and Random Forest Regressions flowchart

The Bagging Regressor is similar to the Random Forest Regressor, the only different here is the model that it is used. While Random Forest Regressor use Decision Tree as base model, the Bagging Regressor use Nu-SVR, which are the model above that the author has calculated.

$$Bagging = \frac{NuSVR(subset\ 1) + \cdots + NuSVR(subset\ 10)}{10} \quad (3.31)$$

$$Random\ Forest = (DecisionTree(subset\ 1) + \ \ldots + DecisionTree(subset\ 10))/10 \quad (3.32)$$

The Voting Regressor, on the other hand, use whole data set for each base estimator. In this case, the author uses models which has calculated before, to average the output.

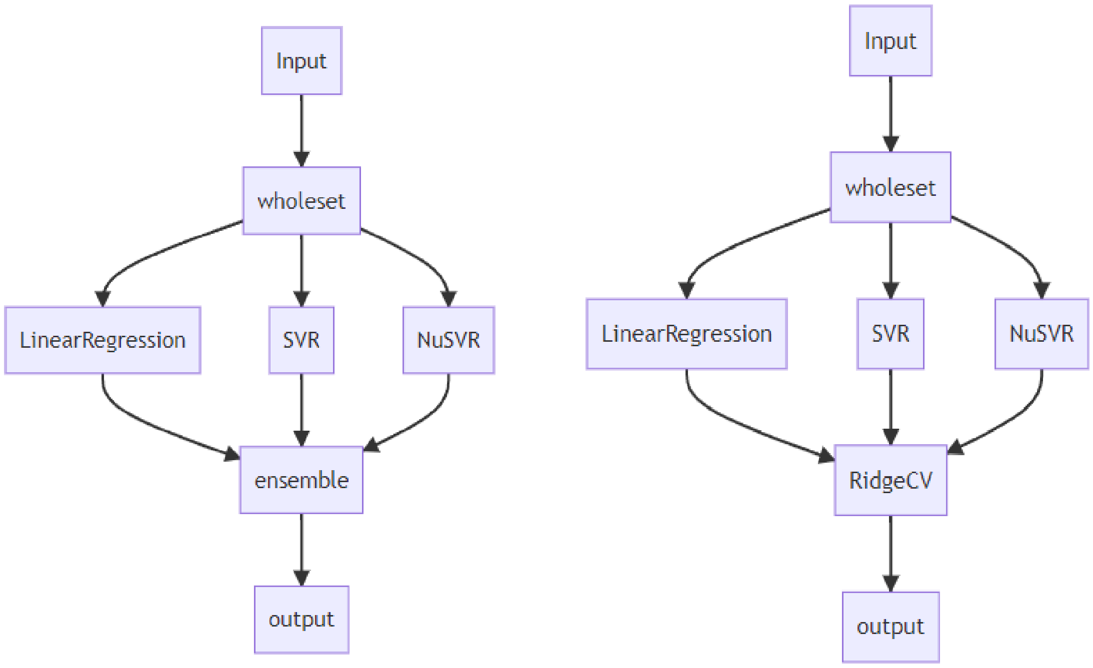


Figure 3-17: Voting Regressor (left) and Stacking Regressor (right) flow charts

$$Voting = \frac{LR(dataset) + SVR(dataset) + NuSVR(dataset)}{3} \tag{3.33}$$

$$\begin{aligned} Stacking = RidgeCV(LR(dataset) + SVR(dataset) \\ + NuSVR(dataset)) \end{aligned} \tag{3.34}$$

Stacking Regressor has a little different with Voting Regressor, it puts the output of individual estimators into final estimator, which is, in this case, RidgeCV. This is near the end working time for this project; therefore the author does not have enough time for deeper understanding about these models.

This cycle can be state as after each time new real data updated, the model will be train again, then predict the new output power in the next 10-min or 1-hour. That is the aim of this project. The difference between these two forecasts is the input and the output. In 10-min-ahead prediction, one wind speed data is the input as the output is one wind power value. In contrast, 6 input and 6 output is given and create by 1-h-ahead predictions.

CHAPTER 4. RESULTS AND DISCUSSION

4.1 Results

In this chapter, the results of using the above models to predict the test data with 432 values. The test data is defined as the forecast wind speeds that the user will buy from supplier. Hence, with wind speed data at 3:10pm, author's models will predict 3:10pm wind power. With one-step ahead or 10-min ahead, the author firstly gives one Speed data point from the test set to the estimators, it predicted a forecast Power. After 10 minutes, the author gains the real data given by supplier, this real data will be put into training set, and train the model once again, then continue to predict, …With one-hour ahead, it is similar. Notwithstanding, instead of givens one Speed at a time, the author now gives 6 data as an hour to the estimators, and the cycle continues.

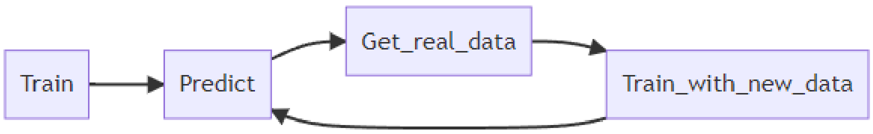


Figure 4-1: Train-Predict cycle

4.1.1 Statistical method

4.1.1.1 One step ahead

Apply formula 3.16 and 3.17 that the author has built for the test dataset, with 1 input at a time. The predictions when using ARIMA and ARIMAX models for 10-min ahead forecasts are illustrated in these figures below. The evaluation results are showed in Table 4-1.

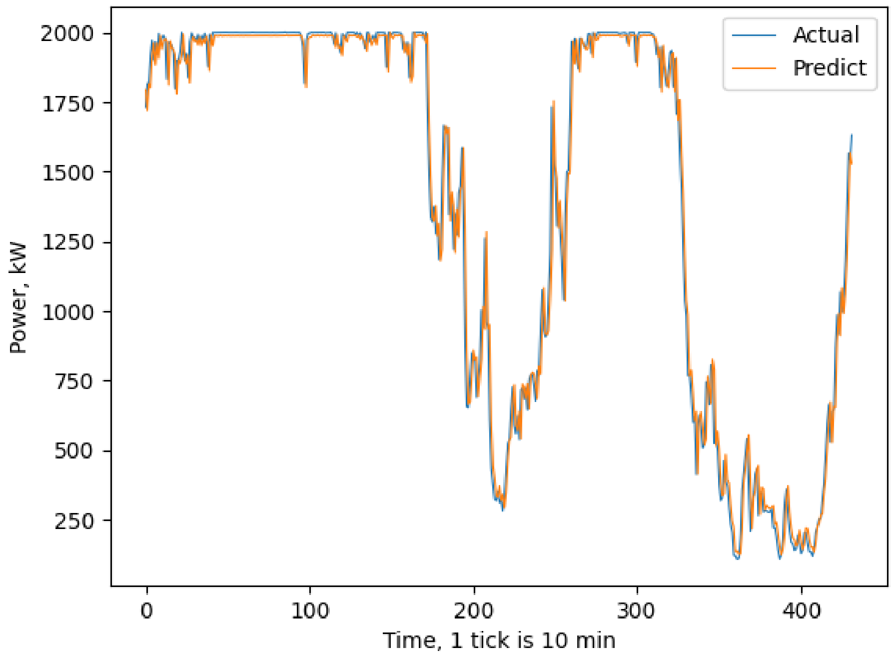


Figure 4-2: One-step ahead predictions using ARIMA model

The figure above only shows two lines, one is "Actual" – which is real data, and one is "Predict" – which is forecast value. As ARIMA formula, it only uses historical data of the data it will predict, hence, it does not use the wind speed. Therefore, it can only give one output.

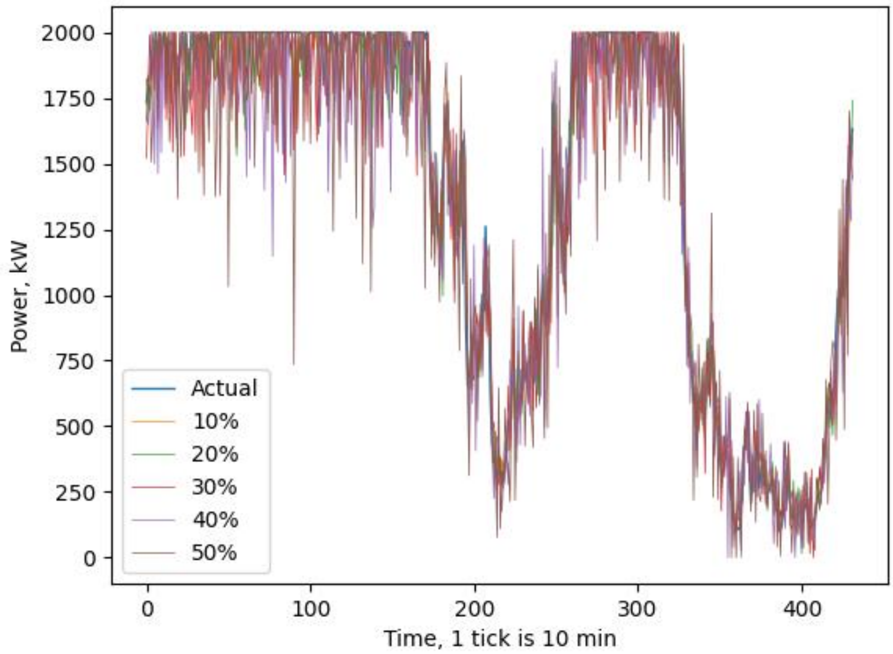


Figure 4-3: One-step ahead predictions using ARIMAX model

In contrast, the figure below has 6 lines in total. One "Actual" and 5 lines has named 10%, 20%…,50%. These 5 lines stand for the maximum error of the testing data set. This setting will be applied for all other figures which have error in testing data. ARIMAX has Exogenous Covariates in the formula, thus that variable will be the wind speed.

Table 4-1: Evaluations of 1-step ahead predictions using statistical method with input errors

	RMSE (kW)	NRMSE (%)	MAE (kW)	MRE (%)	MAPE (%)
ARIMA	91.44	4.57	58.00	2.90	7.97
ARIMAX					
10%	82.91	4.15	57.45	2.87	6.17
20%	112.87	5.64	75.87	3.79	8.74
30%	155.91	7.80	107.05	5.35	11.77
40%	199.16	9.96	134.06	6.70	15.64
50%	255.33	12.77	170.51	8.53	19.72

In the table above, it is showed that ARIMA has MRE is equal to 2.90%, which is a good value. And ARIMAX can handle up to 20% with input error, this have not met the requirement the author settle in the beginning. By "handle", the author means that the MRE value is less than 5% is acceptable. Other evaluations give same comment. However, the 10-min ahead forecasting is not as vital as 1-hour ahead forecasting. The 1-hour ahead give the reasonable time for the users (grid operators, plant operators, etc.) to tackle problems if there is any present.

4.1.1.2 One hour ahead

For 1-h-ahead forecasting, apply equation 3.16 and 3.17 again, the outcomes are worse than 10-min-ahead. Furthermore, discusses will be state in later of this project.

The 1-h ahead forecasting using ARIMA method show in the figure above clearly show that it has large errors as 2 lines can be seen separately. It is understandable since ARIMA follows the given trend of its historical data. And predict 6 values ahead of course will give more error than only predict 1 data.

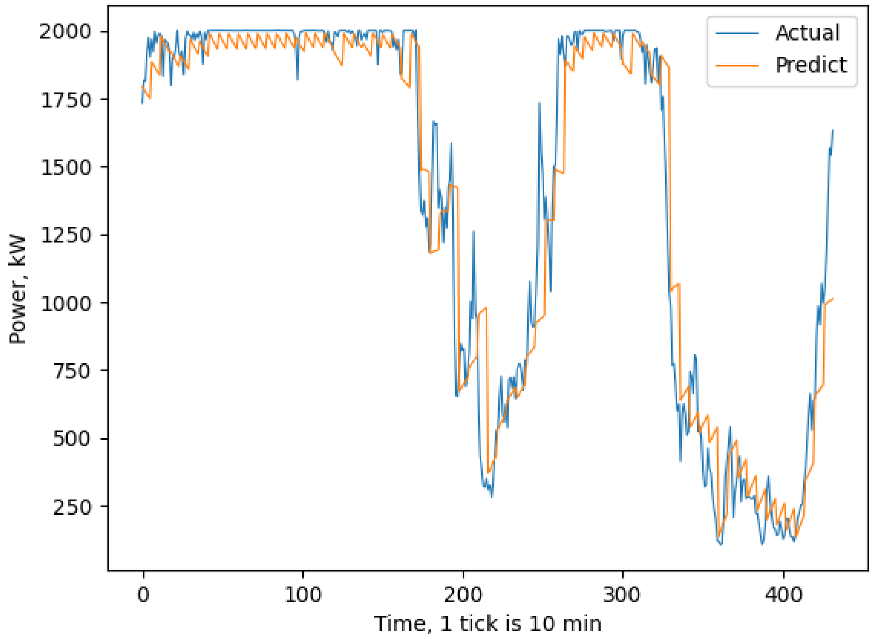


Figure 4-4: One hour ahead predictions using ARIMA model

The ARIMAX acts the same with ARIMA. Notwithstanding, it has Exogenous Covariates in the equation, hence it might give better results. But since it has to predict 6 values in a row, the MRE will be greater than 1-min ahead prediction.

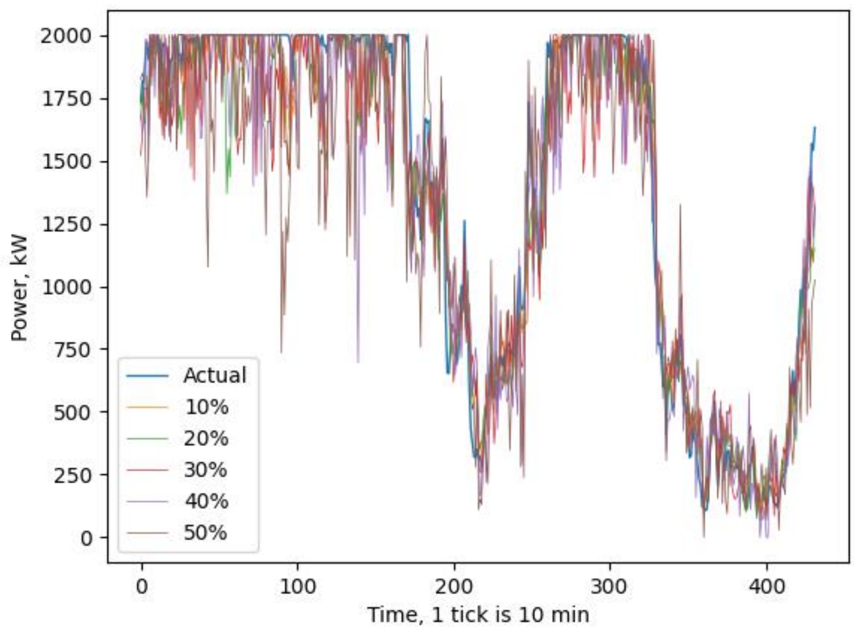


Figure 4-5: One hour ahead predictions using ARIMAX model

The table below showed that ARIMA method cannot handle the situation because its MRE is greater than 5%. The ARIMAX method now can only tackle the problem when the error is less than 20%, which is not practical.

Table 4-2: Evaluations of 1-hour ahead prediction using statistical method with input errors

	RMSE (kW)	NRMSE (%)	MAE (kW)	MRE (%)	MAPE (%)
ARIMA	183.39	9.17	114.72	5.74	17.36
ARIMAX					
10%	136.38	6.82	87.08	4.35	9.80
20%	166.63	8.33	114.38	5.72	12.22
30%	201.12	10.06	142.05	7.10	15.33
40%	212.52	10.63	144.49	7.22	18.76
50%	283.24	14.16	190.41	9.52	20.84

4.1.2 Machine Learning

4.1.2.1 One step ahead

One step ahead results is not as essential as one hour ahead since they do not give the reasonable time for the users (grid operators, plant operators, etc.) to tackle problems if there is any present. However, it is good to see how models work if the outcomes are asked. Applying equation 3.19 for the test data give the figure below.

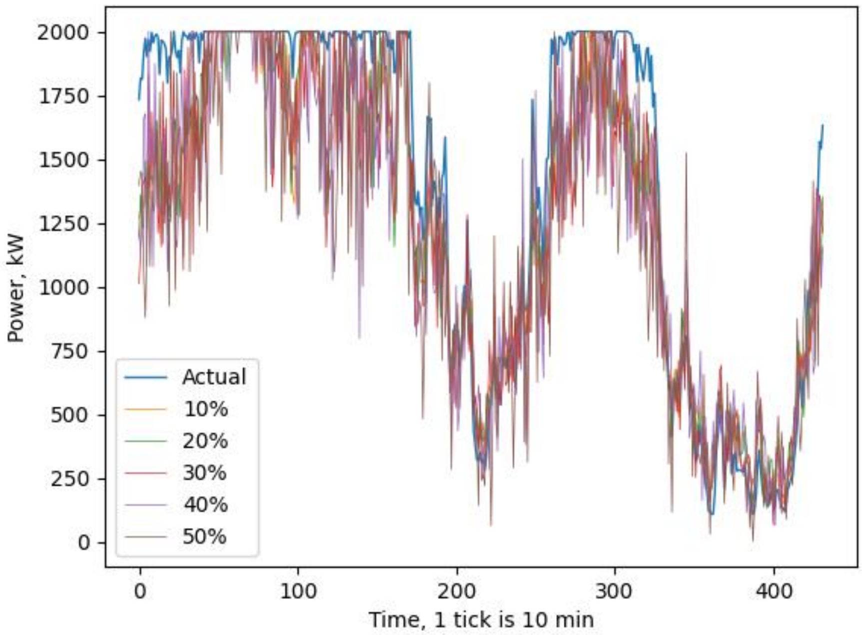


Figure 4-6: One-step ahead predictions using Linear Regression model

The figure above illustrated that the "Actual" line is kind of stand-alone because it is visible among others. That means the error in this model is quite large. That given the assumption that this model is not practical, and that has been proved in Table 4.3. The Linear Regression model is one of the easiest, classical models. Therefore, it is understandable the reason it gives bad results.

In the SVR model case, by applying equation 3.28, the "Actual" line is less visible than the Linear Regression model. However, the lines represent forecasting with errors still have large distance with the "Actual" one. It might be cause by the reason that the model that is being used is a "Regression" type. This type is one of the oldest, thus might give the not so good results.

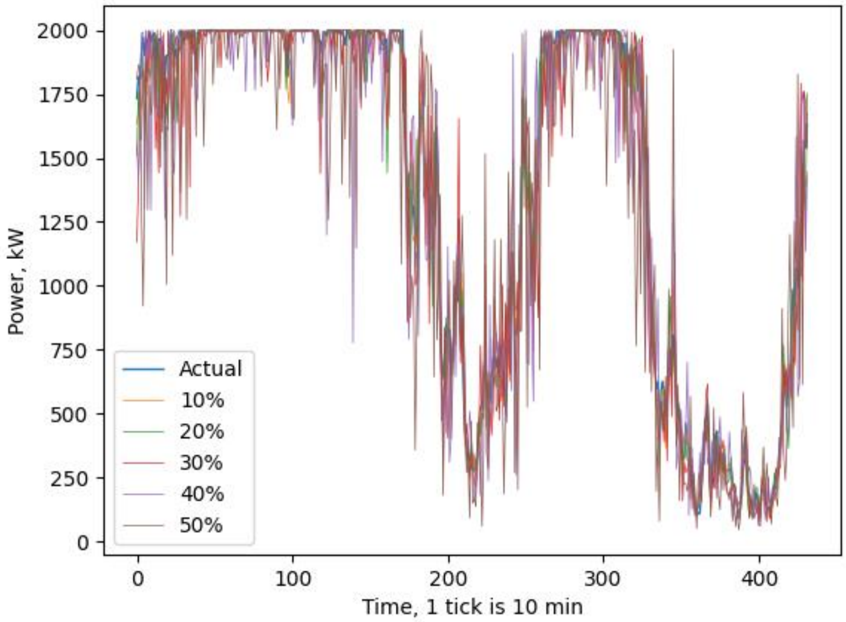


Figure 4-7: One-step ahead predictions using SVR model

The Nu-SVR model archive by equation 3.30 with test data as input, is a little better than SVR, the easiest to see evidence is the 50 to 100 in Time. The lines are kind of merged into one. The results showed in table 4-3 proved this assumption.

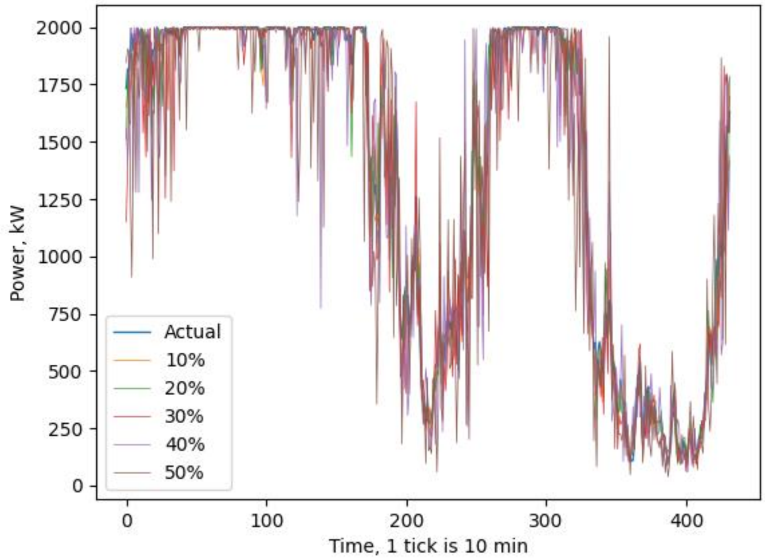


Figure 4-8: One-step ahead predictions using Nu-SVR model

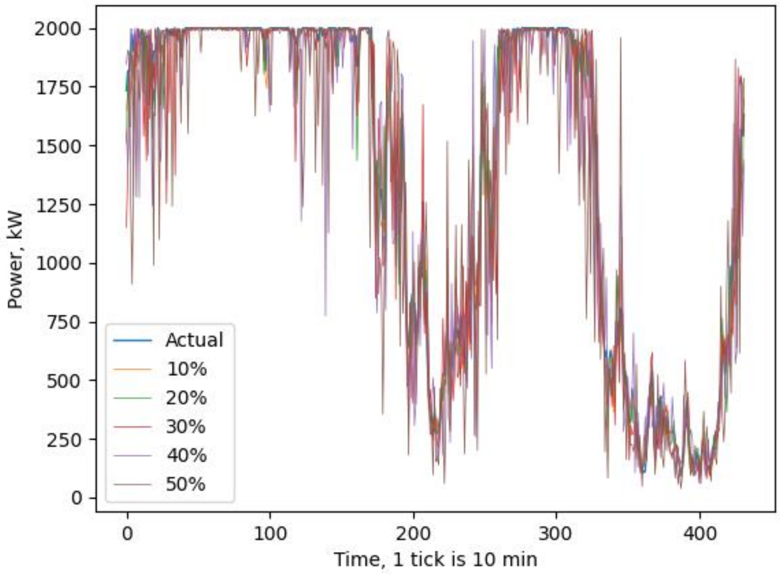


Figure 4-9: One-step ahead predictions using Bagging Regression model

The Ensemble models, in general, give good results. The Bagging Regression and Random Forest Regression, from equation 3.31 and 3.32, are similarly, it is only different in the base model which are Nu-SVR and Decision Tree respectively. Results illustrated in figures 4-8 and 4-9 are belong to these models. It can be seen that these figures are kind of the same to Nu-SVR one. And it is, the evaluations showed in table 4-3 is the proofs.

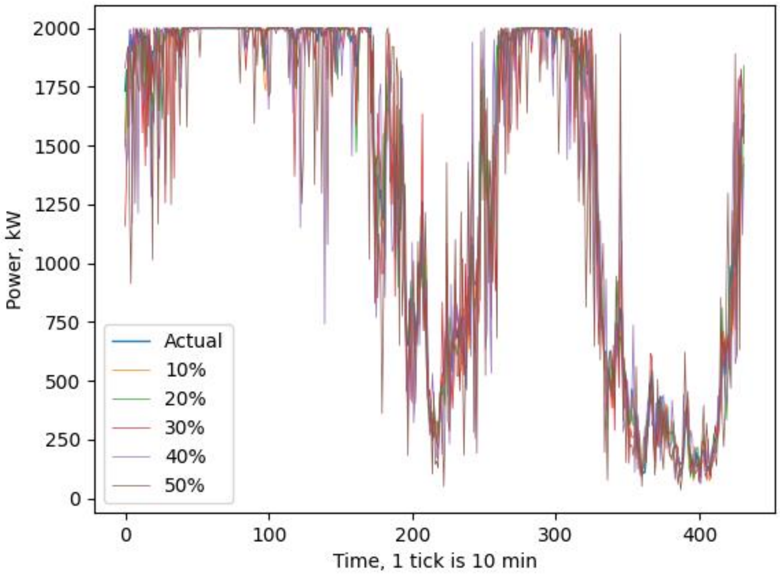


Figure 4-10: One-step ahead predictions using Random Forest Regression model

Stacking Regression give the best results out of them all when using 3.34 equation with test data. Its MRE is only 4.79% in average. The results of individual estimator are calculated and use a regressor is used to compute the final prediction. The RidgeCV is the final model which give the final forecasting. This gives the author a thought to use RidgeCV in the future works.

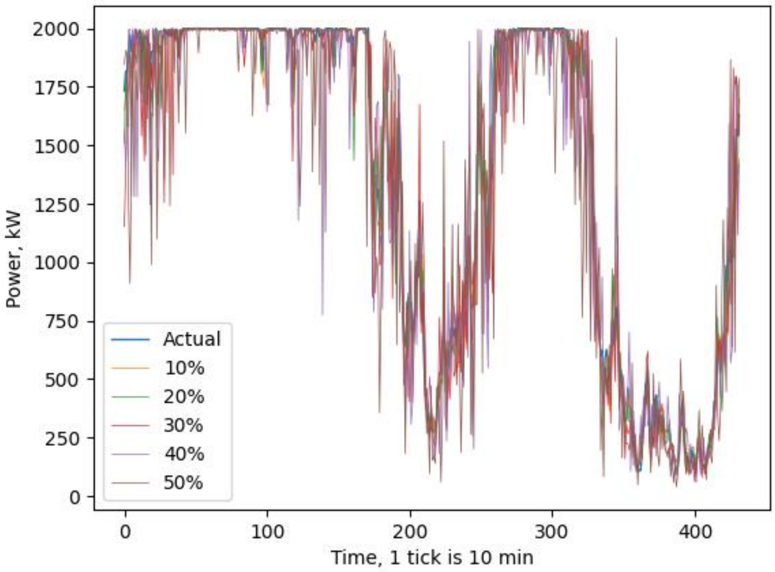


Figure 4-11: One-step ahead predictions using Stacking Regression model

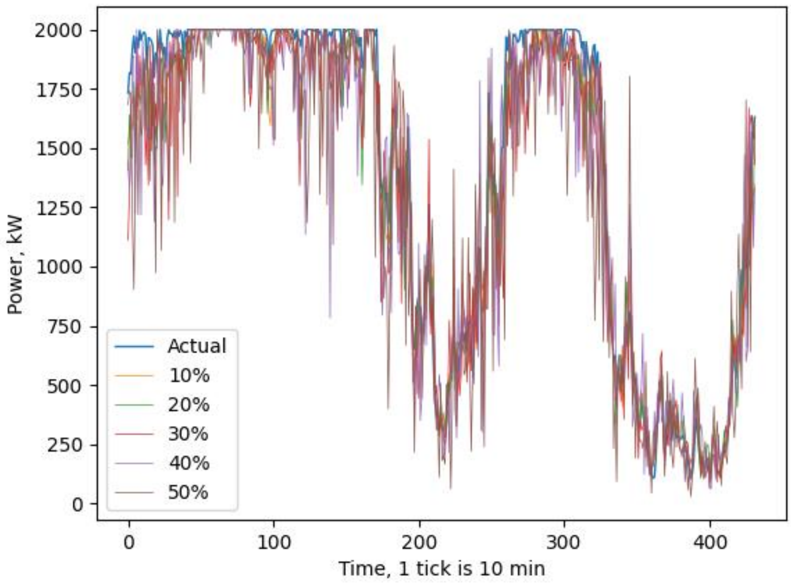


Figure 4-12: One-step ahead predictions using Voting Regression model

The Voting Regression, archived by formula 3.33, being used have the weights in None, which mean the function will give uniform weight. Results of this model is outstanding cause it's has quite big error compared to others. The "Actual" line can be seen in several parts of the figure showed that.

Table 4-3: Evaluations of one step ahead predictions using Machine Learning method

Models		RMSE (kW)	NRMSE (%)	MAE (kW)	MRE (%)	MAPE (%)
Linear Regression						
Error of test data input	10%	261.56	13.08	196.07	9.80	17.36
	20%	268.51	13.43	197.85	9.89	18.76
	30%	297.28	14.86	214.43	10.72	19.57
	40%	316.88	15.84	233.82	11.69	23.22
	50%	367.34	18.37	259.92	13.00	25.01
	Means	**302.31**	**15.12**	**220.42**	**11.02**	**20.78**
SVR						
Error of test data input	10%	62.94	3.15	40.28	2.01	4.87
	20%	103.34	5.17	63.51	3.18	7.45
	30%	158.40	7.92	98.65	4.93	11.17
	40%	205.13	10.26	121.48	6.07	14.25
	50%	279.83	13.99	173.13	8.66	19.54
	Means	**161.93**	**8.10**	**99.41**	**4.97**	**11.45**
Nu-SVR						
Error of test data input	10%	58.49	2.92	37.24	1.86	4.66
	20%	101.45	5.07	61.32	3.07	7.31

	30%	158.01	7.90	95.60	4.78	10.98
	40%	204.76	10.24	117.20	5.86	13.95
	50%	280.66	14.03	169.03	8.45	19.25
	Means	**160.68**	**8.03**	**96.08**	**4.80**	**11.23**
Bagging Regression						
Error of test data input	10%	58.50	2.92	37.25	1.86	4.66
	20%	101.45	5.07	61.32	3.07	7.31
	30%	158.02	7.90	95.61	4.78	10.98
	40%	204.78	10.24	117.20	5.86	13.95
	50%	280.68	14.03	169.03	8.45	19.25
	Means	**160.69**	**8.03**	**96.08**	**4.80**	**11.23**
Random Forest Regression						
Error of test data input	10%	61.12	3.06	38.71	1.94	5.00
	20%	106.77	5.34	64.44	3.22	7.86
	30%	161.85	8.09	97.70	4.89	11.27
	40%	209.55	10.48	118.61	5.93	13.94
	50%	281.27	14.06	169.43	8.47	19.52
	Means	**164.11**	**8.21**	**97.78**	**4.89**	**11.52**
Stacking Regression						
Error of test data input	10%	58.34	2.92	36.96	1.85	4.64
	20%	101.33	5.07	61.07	3.05	7.29
	30%	157.88	7.89	95.43	4.77	10.97
	40%	204.65	10.23	117.01	5.85	13.94
	50%	280.52	14.03	168.88	8.44	19.25
	Means	**160.55**	**8.03**	**95.87**	**4.79**	**11.22**
Voting Regression						
Error of test data input	10%	111.37	5.57	80.13	4.01	7.03
	20%	137.38	6.87	95.29	4.76	9.57
	30%	183.01	9.15	122.47	6.12	12.28
	40%	221.91	11.10	146.58	7.33	15.91
	50%	292.13	14.61	191.82	9.59	20.37
	Means	**189.16**	**9.46**	**127.26**	**6.36**	**13.03**

The table above showed that if only predict 10-min ahead, most Machine Learning models can handle to 30% maximum input error. The classical Linear Regression cannot be too old for the job. Notwithstanding, like it is mentioned before, the one-step head prediction results is not as vital as the one-hour ahead. But it is better to see how the models works in this condition, so it is prepared as one-hour ahead results is always worse than 10-min ahead.

4.1.2.2 One hour ahead

One-hour ahead forecasting give the users enough time to handle the problems if there is any present. It is far more essential than 10-min ahead and have less prediction time when testing. It is because with same testing data, one-hour ahead only need to predict the length of the testing data divided by six times. The results, without a doubt, have larger error than one-step ahead. However, this is what needed for users.
The Linear Regression has the worst outcome. It is foreseen, because the model is old. The evaluations in Table 4-4 showed that its MRE is nearly 10% with only 10% max input error, which is completely unusable and considered bad results.

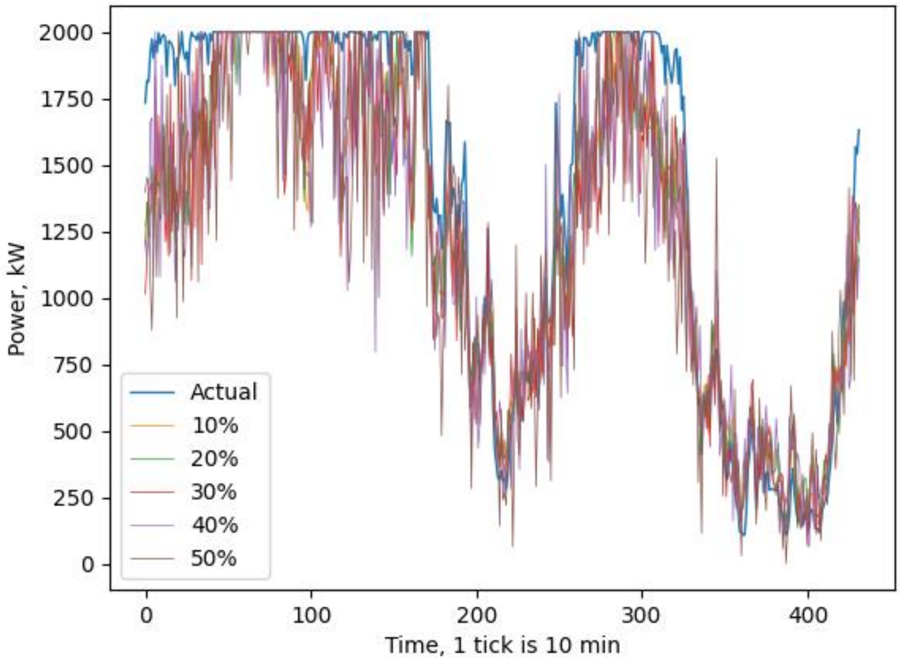


Figure 4-13: One hour ahead predictions using Linear Regression model

The SVR models in one-hour ahead forecasting give the similar wind power outcome with one-step ahead. The variance is a little bit higher than 10-min ahead. However, it still can handle up to 30% input error maximum, but it's near the margin.

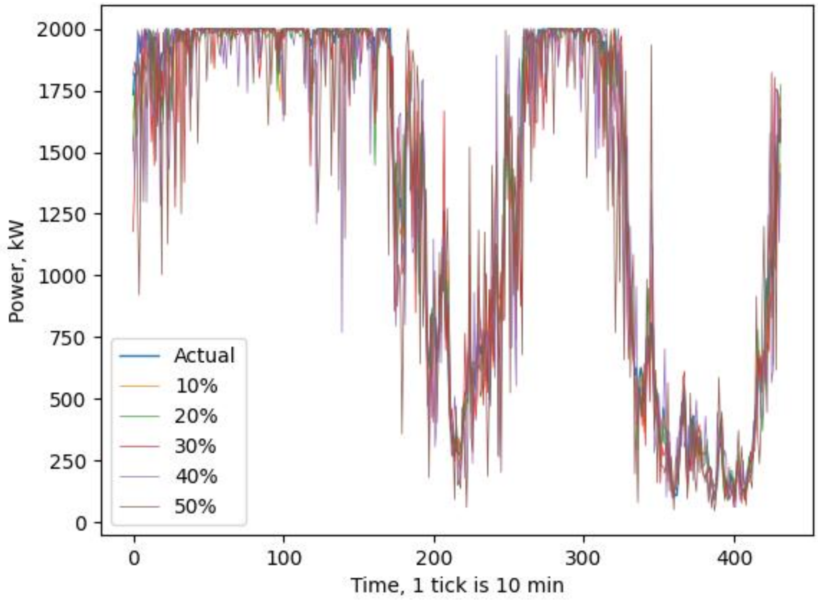


Figure 4-14: One hour ahead predictions using SVR model

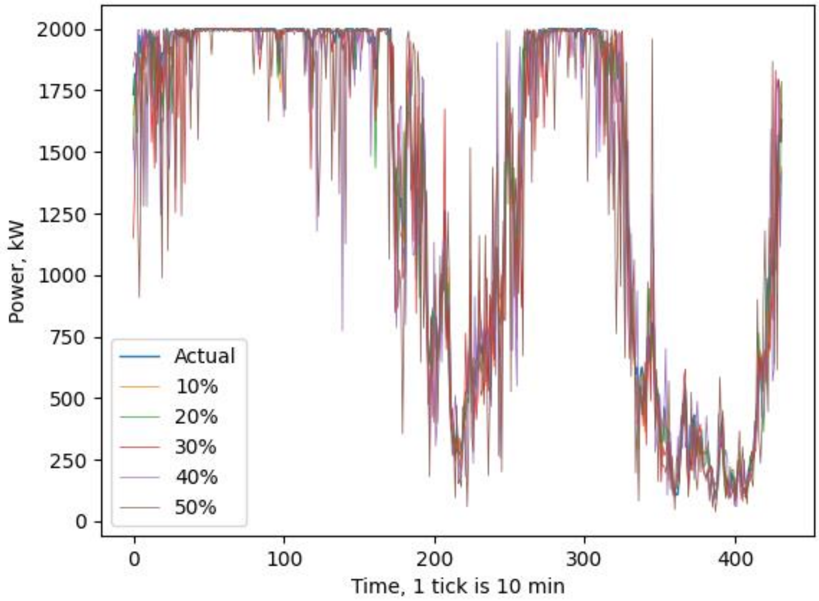


Figure 4-15: One hour ahead predictions using Nu-SVR model

The Nu-SVR is also like SVR, which is the results of one-hour and 10-min ahead are similar, only a little higher MRE in this condition. And nearly met the border line of acceptance. These results have not met the requirements of the author.

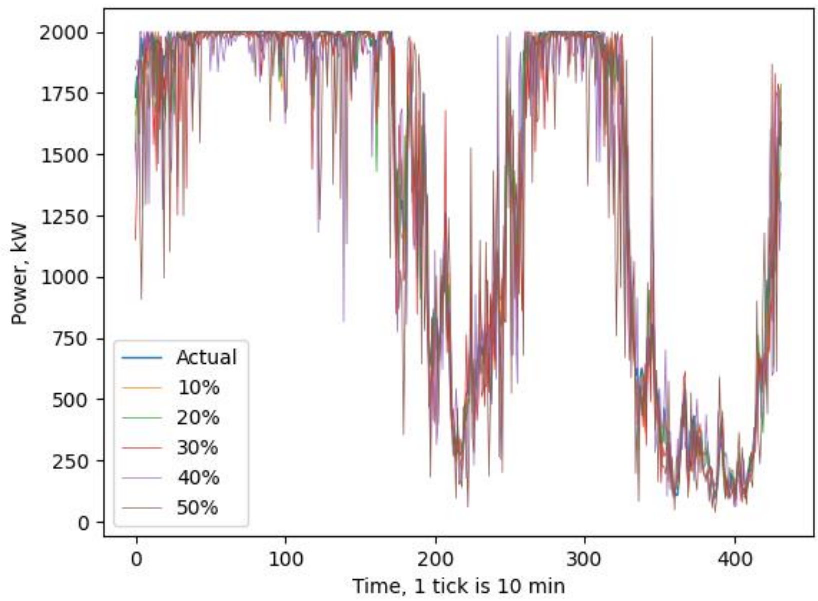


Figure 4-16: One hour ahead predictions using Bagging Regression model

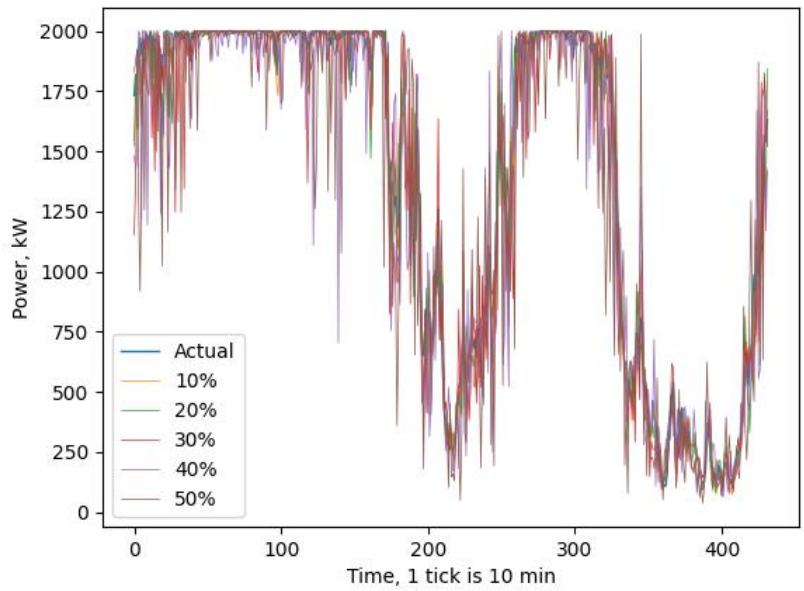


Figure 4-17: One hour ahead predictions using Random Forest Regression model

The Bagging Regression, surprisingly, have kind of not so good outcome. The 50 to 100 in Time axis showed that it not doing well. Table 4-4 show that its MRE is lower than above models, but still near the border line. Therefore, it is also not practical. This might be is a consequence of how the bagging model works. The similar results also present in Random Forest Regression model as it is only different in the base model.

The Stacking Regression is still working well like the 10-min ahead forecasting. It seems like that working the output if individual models with another model (formula) is better than just voting or averaging. Several parts of "Actual" line are still visible likes one-step ahead forecasting. Which means the results is not good. The results might change if the weight is not uniform. However, there is no way to determine the weight since the using dataset only have wind speed and wind power. This give the author an idea of using wind direction, which is neglect in this project, to determine the weight. But will be a work of future.

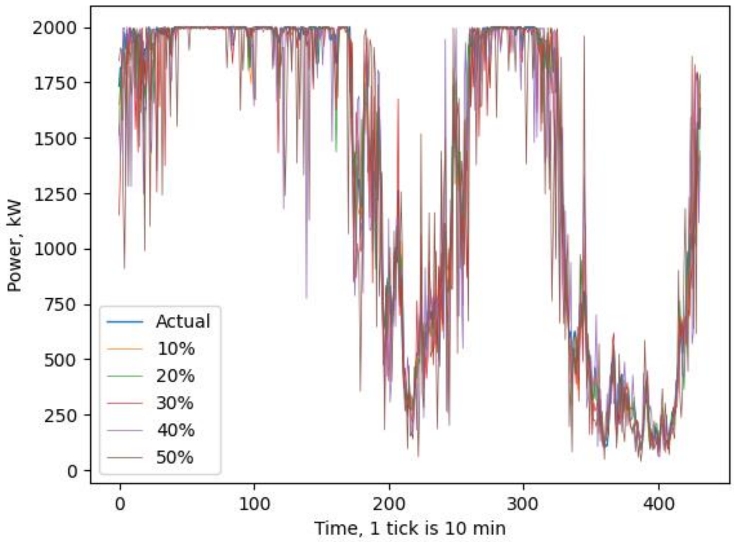


Figure 4-18: One hour ahead predictions using Stacking Regression model

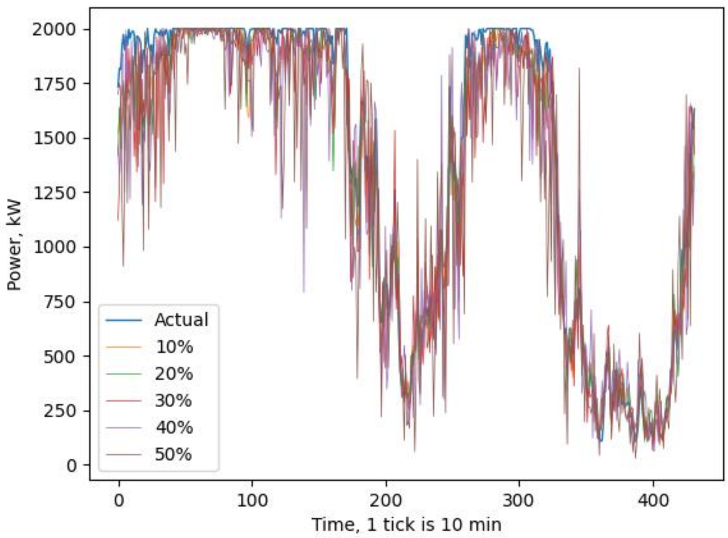


Figure 4-19: One hour ahead predictions using Voting Regression model

The table below showed that in predict 1-hour ahead, most Machine Learning models can handle to 30% maximum input error, but several are near the border line of acceptance. These results might be use in reality by users, but only for references. The author deem that these results are not good enough for using it directly.

Table 4-4: Evaluations of one hour ahead predictions using Machine Learning method

Models		RMSE (kW)	NRMSE (%)	MAE (kW)	MRE (%)	MAPE (%)
Linear Regression						
Error of test data input	10%	261.50	13.07	196.02	9.80	17.35
	20%	268.45	13.42	197.80	9.89	18.75
	30%	297.23	14.86	214.38	10.72	19.56
	40%	316.83	15.84	233.78	11.69	23.22
	50%	367.30	18.37	259.89	12.99	25.01
	Means	**302.26**	**15.11**	**220.37**	**11.02**	**20.78**
SVR						
Error of test data input	10%	63.08	3.15	41.42	2.07	4.91
	20%	104.57	5.23	66.74	3.34	7.62
	30%	158.48	7.92	99.16	4.96	11.19
	40%	204.86	10.24	122.23	6.11	14.24
	50%	279.77	13.99	173.20	8.66	19.54
	Means	**162.15**	**8.11**	**100.55**	**5.03**	**11.50**
Nu-SVR						
Error of test data input	10%	61.07	3.05	40.85	2.04	4.83
	20%	103.39	5.17	64.28	3.21	7.47
	30%	158.56	7.93	99.61	4.98	11.15
	40%	205.18	10.26	119.75	5.99	14.06
	50%	280.55	14.03	169.45	8.47	19.26
	Means	**161.75**	**8.09**	**98.79**	**4.94**	**11.35**
Bagging Regression						
Error of test data input	10%	58.77	2.94	38.35	1.92	4.71
	20%	101.13	5.06	61.61	3.08	7.31
	30%	158.12	7.91	96.70	4.83	11.00
	40%	205.29	10.26	125.53	6.28	14.38
	50%	280.69	14.03	169.62	8.48	19.29
	Means	**160.80**	**8.04**	**98.36**	**4.92**	**11.34**
Random Forest Regression						
Error of test data input	10%	62.05	3.10	40.03	2.00	5.08
	20%	106.87	5.34	66.75	3.34	7.99
	30%	161.65	8.08	98.23	4.91	11.28

	40%	213.37	10.67	127.74	6.39	14.49
	50%	280.92	14.05	170.26	8.51	19.55
	Means	**164.97**	**8.25**	**100.60**	**5.03**	**11.68**
Stacking Regression						
Error of test data input	10%	58.54	2.93	37.71	1.89	4.69
	20%	102.84	5.14	64.74	3.24	7.52
	30%	158.06	7.90	95.80	4.79	10.98
	40%	204.58	10.23	120.89	6.04	14.11
	50%	280.74	14.04	170.94	8.55	19.36
	Means	**160.95**	**8.05**	**98.02**	**4.90**	**11.33**
Voting Regression						
Error of test data input	10%	111.27	5.56	80.35	4.02	7.03
	20%	137.83	6.89	95.88	4.79	9.59
	30%	182.84	9.14	122.87	6.14	12.26
	40%	222.13	11.11	147.76	7.39	16.00
	50%	292.34	14.62	192.66	9.63	20.40
	Means	**189.28**	**9.46**	**127.90**	**6.40**	**13.05**

4.2 Discussion

In one step ahead forecasting, the Statistical method can handle to 20% maximum of error in wind speed given. The majority of Machine Learning models can deal with up to 30% of error of input. This clearly state that the Machine Learning method performs better in this field. However, with only 10 minutes ahead, it is hard to reserves any necessary power for the system operators. Therefore, the result in 1-h-ahead predictions has more value than this one.

In one hour ahead forecasting, the ARIMAX model is the only one can predict with a reasonable error, and it can only handle 10% of error in the input at max. Most of the Machine Learning models still be able to deal with 30% maximum error of the wind speed. The lowest MRE belongs to Stacking Regression model with 1.89%, 3.24%, 4.79% respectively to 10%, 20%, and 30% of error in wind speed input. The Ensemble models work better than solo models as they combine all results from each base model. The closest to actual Wind Power is predictions of Stacking Regression model, which have 4.90% MRE average in all test packs.

CHAPTER 5. CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

In conclusion, the author deems that:

- The variation in WT system causes instability of the grid system (overvoltage, power quality, ...). The weather (or seasons) and location of the WT is the main cause for the instability.
- The Statistical method is not practical for the very short-term forecasting of Wind Power.
- Regression models and Ensemble for previous models are better in forecasting Wind Power in very short-term of time. These models can deal with 30% error of input at maximum. For specific, the best result is given by Stacking Regression (Ensemble) with 4.90% MRE in average).
- The forecasting can be applied in reality, but only for reference since it is not reliable enough. In addition, improvement is also needed in future works.

5.2 Future works

The wind speed can be achieved by weather forecast has great error with real wind speed measured on the WT. It is because of the different in height between where the weather forecast tools locate and where the wind blows WT blades. If the model can handle 50% of error in input wind speed, the use of weather forecast in forecasting future wind power might be practical. Therefore, in the future, the author want to use more machine learning model (such as K-means clustering, K-nearest neighbors, etc.), involve with deep learning methods for creating better models (likes RNN, ANN, CNN, etc.). In addition, improvement for used models is also needed. Using wind direction to determine weight of each data is one method that will be applied.

REFERENCES

Vietnamese

Quyết định số 1855/QĐ-TTg của Thủ tướng Chính phủ ngày 27/12/2007 về phê duyệt Chiến lược phát triển năng lượng quốc gia của Việt Nam đến năm 2020, tầm nhìn đến năm 2050.

English

[1] Bruckner T., I. B. (2014). Chapter 7. Energy Systems. In O. R.-M. Edenhofer, *Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change* (p. 511). Cambridge, United Kingdom: Cambridge University Press. Retrieved from https://www.ipcc.ch/site/assets/uploads/2018/02/ipcc_wg3_ar5_chapter7.pdf

[2] Cha Zhang, Y. M. (2012). *Ensemble Machine Learning: Methods and Applications* (2012 ed.). Springer Publishing Company.

[3] Chih-Chung Chang, C.-J. L. (2002). Training v-support vector regression: theory and algorithms: Neural Computation. *Neural Computation, 14*. doi:https://dl.acm.org/doi/10.1162/089976602760128081

[4] Duc Luong, N. (2015). A critical review on potential and current status of wind energy in Vietnam. *enewable And Sustainable Energy Reviews*, 43. doi:10.1016/j.rser.2014.11.060

[5] H. Bludszuweit, J. A.-N. (2008). Statistical Analysis of Wind Power Forecast Error. *IEEE Transactions on Power Systems, 23, no.3*, pp. 983-991. doi:10.1109/TPWRS.2008.922526.

[6] Ho, S. &. (1998). The use of ARIMA models for reliability forecasting and analysis. *Computers & Industrial Engineering*, 35(1-2), 213-216.

[7] J.F. Manwell, J. M. (2010). *Wind energy explained: Theory, Design and Application 2nd Edition.* Massachusetts: Wiley.

[8] Kehinde A. A., N. I. (2021). The Effect of the Number of Blades on the Efficiency of a Wind Turbine. *IOP Conf. Ser.: Earth Environ. Sci.*, 801. doi:10.1088/1755-1315/801/1/012020

[9] Leonardo Paoli, T. G. (2022). Electric cars fend off supply challenges to more than double global sales – Analysis. *IEA*.

[10] Mitchell, T. (1997). *Machine Learning*. New York: McGraw Hill.

[11] Ritchie, H. R. (2020). CO_2 and Greenhouse Gas Emissions. *Our World In Data*.

[12] Soares-Ramos, E. d.-A.-M.-R. (2020). Current status and future trends of offshore wind power in Europe. *Energy*, 202. doi:10.1016/j.energy.2020.117787

[13] Sun, M. F.-M. (2019). A two-step short-term probabilistic wind forecasting methodology based on predictive distribution optimization. *Applied Energy*, 238, 1497-1505. doi:10.1016/j.apenergy.2019.01.182

[14] Zhang, S. W. (2020). China in global wind power development: Role, status and impact. . *Renewable And Sustainable Energy Reviews*, 127. doi:10.1016/j.rser.2020.109881

[15] CambridgeDictionary. (22, 07 16). *Statistics*. Retrieved from STATISTICS: https://dictionary.cambridge.org/dictionary/english/statistics

[16] IRENA. (2022, July 16). *Renewable Energy Capacity Statistics 2022*. Retrieved from International Renewable Energy Agency (IRENA): https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2022/Apr/IRENA_RE_Capacity_Statistics_2022.pdf