VIET NAM GENERAL CONFEDERATION OF LABOUR

TON DUC THANG UNIVERSITY

ELECTRICAL – ELECTRONIC DEPARTMENT

HA MAI PHUNG

SHORT-TERM FORECASTING OF WIND POWER

GRADUATION PROJECT

ELECTRICAL ENGINEERING

HO CHI MINH CITY. 2022

VIET NAM GENERAL CONFEDERATION OF LABOUR

TON DUC THANG UNIVERSITY

ELECTRICAL – ELECTRONIC DEPARTMENT

HA MAI PHUNG – 418H0382

SHORT-TERM FORECASTING OF WIND POWER

GRADUATION PROJECT

ELECTRICAL ENGINEERING

Instructor

Prof. HongTzer - Yang

HO CHI MINH CITY. 2022

ACKNOWLEDGEMENTS

First and foremost a big thank you is dedicated to my supervisor Prof. HongTzer – Yang, as NCKU students, especially are Beam and Andi. Without their help and guidance this thesis would hardly be completed. Thanks also to PhD. Huynh Van Van for providing me connected to the NCKU students.

Sincere gratitude and appreciation is directed to my girlfriend, Nhu Quynh. Without her spiritual encouragement, I would hardly done the thesis on time. She always beside me whenever I need, or felt pressure. A special thanks is directed to my classmates through four years, Vinh Cuong. He helped me when I had problem in coding, also taught me some tips or encourage me for the final phase of the thesis.

Lastly, I want to thank my family for their invaluable support through my studies.

I sincerely thank you!

Ho Chi Minh City, July, 15th, 2022

Author

Phung

Hà Mai Phụng

Công trình được hoàn thành tại Trường Đại học Tôn Đức Thắng

Cán bộ hướng dẫn khoa học: GS. HongTzer – Yang

Đồ án tốt nghiệp được bảo vệ tại **Hội đồng đánh giá Đồ án tốt nghiệp của Trường Đại học Tôn Đức Thắng** vào ngày … /…/……

Xác nhận của Chủ tịch Hội đồng đánh giá Đồ án tốt nghiệp và Trưởng khoa quản lý chuyên ngành sau khi nhận Đồ án tốt nghiệp đã được sửa chữa (nếu có).

CHỦ TỊCH HỘI ĐỒNG **TRƯỞNG KHOA**

……

WORK IS COMPLETED AT
TON DUC THONG UNIVERSITY

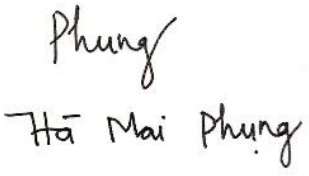
I hereby declare that this is my own research work and is under the scientific guidance of Prof. HongTzer Yang. The research contents and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments and evaluation are collected by the author himself from different sources, clearly stated in the reference section.

In addition, the Project also uses a number of comments, assessments as well as data of other authors, other agencies and organizations, with citations and source annotations.

If there is any fraud, I would like to take full responsibility for the content of my Project. Ton Duc Thang University is not related to copyright and copyright violations caused by me during the implementation process (if any).

Ho Chi Minh City, 16th July 2022

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

Họ tên sinh viên: Hà Mai Phụng

Ngành: Kỹ thuật điện Lớp: 18H40101 MSSV: 418H0382

1. Tên đề tài: 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 01 mô hình phân nhóm Fuzzy C – Means Clustering (FCM), 01 mô hình phân loại K – Nearest Neighbors, 01 mô hình dự báo Feed – Forward Neural Network (FFN).
- Có kết quả dự báo của các mô hình.
- Đánh giá các mô hình.
- Có sự so sánh kết quả của các mô hình, từ đó rút ra kết luận.

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. HongTzer Yang - Phần hướng dẫn: 100%

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

Ngày 28 tháng 3 năm 2022

CHỦ NHIỆM BỘ MÔN	GIẢNG VIÊN HƯỚNG DẪN
TS. Đinh Hoàng Bách	Prof. HongTzer Yang
TRƯỞNG KHOA	
TS. Đồng Sĩ Thiên Châu	

TON DUC THANG UNIVERSITY
ELECTRICAL ENGINEERING

SOCIAL REPUBLIC OF VIETNAM
Independence – Freedom – Happiness

GRADUATION PROJECT SCHEDULE

Student's name: Ha Mai Phung
Class: 18H40101 Student ID: 418H0382
Subject name: Short – term forecasting of Wind Power

Week/Day	**Amount of work**		**Instructor's signature**
	Done	**Not done yet**	
1 (from 28/03 to 05/04)	Overview about the thesis, works and methods for each steps	Have a reliable data, can not connect with Prof in Taiwan	
2 (from 05/04 to 11/04)	Purpose of the topic, ideas, method and implementation tools	Theoretical basis of the Machine Learning, have not find out the method to forecasting	
3 (from 12/04 to 18/04)	Research the theoretical background of the thesis, the wind speed and power energy	Research in details the Supervised and Unsupervised Learning	
4 (from 19/04 to 25/04)	Self – taught the programming language, and acquainted with the Visual Studio Code	Distinguish the Clustering and the Classification, methods of them	

5 (from 26/04 to 02/05)	Preprocessing Data with Remove outliers, Interpolation and Normalization	Data after preprocessing still not optimal	
6 (from 03/05 to 09/05)	Try to update the Preprocessing	Research the Clustering method	
7 (from 10/05 to 16/05)	Clustering by Fuzzy C – Means, Understand the FCM algorithm	Final result for Clustering	
Mid-term test	Evaluation of completed volume **50 %** **be continued.**		
8 (from 17/05 to 23/05)	Updated the FCM and learned about the methods uses for Classification	Output for the Classification	
9 (from 24/05 to 30/05)	Done with the Classification	Research the Artificial Neural Network	
10 (from 31/05 to 06/06)	Regression Method to forecasting by FNN	Regression Method to forecasting by FNN	
11 (from 07/06 to 13/06)	Regression Method to forecasting by FNN	Regression Method to forecasting by FNN	
12 (from 14/06 to 20/06)	Regression Method to forecasting by FNN	Write the fully of the report	

13 (from 21/06 to 27/06)	Updating the final report	Updating the final report	
14 (from 28/06 to 04/07)	Updating the final report	Updating the final report	
15 (from 05/07 to 11/07)	Updating the final report	Finalize the report and submit	
Submit the Individual Report	Accomplished 100% Individual Report **protected project**		

DỰ BÁO NGẮN HẠN NHÀ MÁY ĐIỆN GIÓ

TÓM TẮT

Năng lượng rất cần thiết cho đời sống con người và mọi hoạt động sản xuất kinh doanh. Để đáp ứng các yêu cầu trên, con người đã khai thác, sản xuất và tiêu thụ năng lượng từ hóa thạch như than đá, dầu mỏ,… Việc khai thác trên là nguyên nhân làm cho hiệu ứng nhà kính ngày càng gia tăng, dẫn đến biến đổi khí hậu và môi trường xung quanh. Do đó, ở các nước phát triển, họ đang dần sử dụng các năng lượng sạch và tái tạo như năng lượng mặt trời, năng lượng gió.

Năng lượng gió là một trong những năng lượng tái tạo được sử dụng rộng rãi nhất trên toàn cầu. Việc đầu tư vào năng lượng gió đang là xu hướng được các nước phát triển áp dụng để hướng tới sự phát triển bền vững. Bên cạnh đó, sản xuất điện gió dẫn đến biến động điện áp và tần số lưới điện do tính ngẫu nhiên và biến động của năng lượng gió, ảnh hưởng trực tiếp đến sự ổn định chất lượng điện lưới. Dự báo chính xác điện gió là một trong những chiến lược hiệu quả nhất để giảm thiểu các vấn đề nêu trên. Mặc dù dự báo điện gió có độ chính xác của dự báo thấp hơn dự báo phụ tải, nhưng dự báo điện gió vẫn đóng một vai trò thiết yếu trong việc giải quyết các vấn đề vận hành trong cung cấp năng lượng.

Đồ án này nghiên cứu về việc dự báo công suất phát của nhà máy điện gió, từ đó góp phần vào nâng cao hiệu quả vận hành lưới điện tại khu vực sử dụng năng lượng điện gió bằng công cụ chủ yếu là Machine Learning – kỹ thuật phân tích dữ liệu dạy máy tính làm những công việc như con người. Nghiên cứu trước tiên sẽ tập trung trước tiên vào việc phân cụm dữ liệu, mục đích của việc này là để xem xét mối tương quan giữa các điều kiện tốc độ và công suất gió. Điều này giúp các trang trại đưa ra các quyết định vận hành tốt hơn. Song song với việc phân cụm dữ liệu, đồ án cũng sử dụng thuật toán phân loại dữ liệu, mục đích ở đây là để biết được ứng với từng khung giờ nào thì dữ liệu đó sẽ thuộc vào nhóm nào. Sau cùng, sẽ sử dụng mạng nơ – ron để tạo ra một dự đoán công suất. Việc dự đoán công suất theo mạng nơ – ron mang lại hiệu quả cao là bởi nó thiết lập mối quan hệ giữa sản xuất điện gió và các yếu tố liên quan, khả năng lập bản đồ kết nối và học hỏi từ các mẫu dữ liệu; mặt khác,

phương pháp này không cần mô hình toán học để xây dựng mối liên hệ giữa đầu vào và đầu ra để ước tính sản lượng điện gió.

Các chương trình thuật toán và mạng nơ – ron thần kinh nhân tạo đều được viết bằng ngôn ngữ lập trình Python, trên nền tảng phần mềm Visual Studio Code. Đồ án sử dụng ngôn ngữ Python bởi tính dễ sử dụng và có đầy đủ các thư viện bổ trợ trong quá trình thực hiện.

SHORT – TERM FORECASTING OF WIND POWER ABSTRACT

Energy is essential for human life and all business activities. To meet the above requirements, humans have exploited, produced, and consumed fossil energy such as coal, oil, etc. The above exploitation is the cause of the increasing greenhouse effect, leading to climate change and the surrounding environment. Therefore, developed countries are gradually using clean and renewable energies such as solar and wind energy.

Wind energy is one of the most widely used renewable energies globally, and investing in wind energy is a trend applied by developed countries toward sustainable development. Besides, wind power production leads to fluctuations in grid voltage and frequency due to the randomness and changes of wind energy, which directly affects the stability of grid power quality. Accurate wind power forecasting is one of the most effective strategies for mitigating the abovementioned problems. Although wind power forecasting has lower accuracy than load forecasting, wind power forecasting still plays an essential role in solving operational issues in energy supply.

This project studies forecasting the generating capacity of wind power plants, thereby improving the efficiency of electricity grid operation in wind power use areas by the primary tool of Machine Learning - technology. Data analysis techniques teach computers to do things like humans. The study will focus first on data clustering, the purpose of which is to examine the correlation between wind speed and power conditions. This helps farms make better operational decisions. In parallel with data clustering, the project also uses a data classification algorithm; the purpose here is to know which time frame the data will belong. Finally, a neural network will be used to generate a power prediction. Neural network power prediction is highly effective because it establishes the relationship between wind power production and related factors, the ability to map connections, and learn from data patterns. ; on the other

hand, this method does not need a mathematical model to build the relationship between inputs and outputs to estimate wind power output.

Algorithmic programs and artificial neural networks are written in Python programming language based on Visual Studio Code software. The project uses Python language because of its ease of use and a full complement of libraries during implementation.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

LIST OF ABBREVIATIONS

Artificial Intelligent - AI

Artificial Neural Network – ANN

Feed Forward Neural Network – FNN

K Nearest Neighbors – KNN, k-NN, kNN

Fuzzy C Means - FCM

CHAPTER 1. OVERVIEW OF SHORT – TERM FORECASTING OF WIND POWER

1.1 The purpose of implementing the topic

Electric energy is essential for human life and all production and business activities. To meet the above requirements, people have exploited, produced, and consumed fossil-based energy such as coal, oil, etc. The above exploitation is the cause of the increasing greenhouse effect leading to change. Climate and environmental pollution, but the energy demand is increasing rapidly. Meanwhile, fossil energy sources are gradually depleted and substantially impact the environment; hydroelectricity sources affect the ecosystem and flood disasters... Nuclear energy has many risks of loss. Safe and non-disposal of long-term atomic waste.

Therefore, developed countries like Europe, the US, China, and Taiwan, are gradually turning to clean and renewable energy sources such as solar, wind, river, and tidal energy.

Wind energy is one of the most widely used renewable energies globally. Investment in renewable energy such as wind power is a trend adopted by developed countries toward sustainable development.

Grid-connected wind power generation results in fluctuations in grid voltage and frequency due to the randomness and volatility of wind energy, which directly affect the stability of the grid's power quality and the operation of the power system and introduce many uncertainties into power grid dispatching. An accurate wind power forecast is one of the most effective strategies to mitigate the issues mentioned above.

Even though wind power forecasting has a lower prediction accuracy than load forecasting, wind power projections continue to play an essential role in addressing operational issues in energy delivery. Several approaches have recently been used for wind power forecasting. Many publications have been devoted to advancing wind

power forecasting techniques by researchers with extensive field testing expertise. Several wind power forecasting systems have been developed and deployed on wind farms. Wind power forecasting methods are classified into six categories: persistence method, physical method, statistical method, spatial correlation method, artificial intelligence method, and hybrid approach.

Wind power forecasting systems can be categorized according to time scales or approaches. The time-scale categorization of wind power forecasting systems varies between literature sources. Wind power forecasting systems may be classified into four types based on time frames:

- Very – short – term forecasting (Ultra – short – term): From few minutes to 1 hour ahead.
- Short – term forecasting: From 1 hour to several hours ahead.
- Medium – term forecasting: From several hours to 1 week ahead.
- Long – term forecasting: From 1 week to 1 year or more ahead.

Extract from Wen – Yean Chang (2014), we have:

Table 1-1 Time – scale classification for wind forecasting

Time – scale	Range	Applications
Ultra – short – term	Few minutes to 1 hour ahead	• Electricity market clearing. • Real – time grid operations. • Regulation actions.
Short – term	1 hour to several hours ahead	• Economic load dispatch planning. • Load reasonable decisions. • Operation security in electricity market.
Medium – term	Several hours to 1 week ahead	• Unit commitment decisions. • Reserve requirement decisions. • Generator online/offline decisions.
Long – term	1 week to 1 year or more ahead	• Maintenance planning.

		• Operation management. • Optimal operating cost. • Feasibility study for design of the wind farm.

Identifying the importance of forecasting wind power generation, researchers have applied numerous statistical, data mining, and machine learning techniques for developing prediction models.

ANN models used in wind energy have been significantly developed because of their capacity to 'learn' the link between inputs (independent variables) and outputs (dependent variables) by analyzing historical data, a process akin to non-linear regression. Many ANN models and architectures have been suggested and implemented in recent decades, depending on the aims of each study. Some of this research seeks to determine the geographic connection of wind speed and its estimation at a specific location (target point), taking into account wind speed records from nearby areas and their spatial correlations.

I present a revolutionary wind power forecasting approach for wind power plant requirements in this study. The goal is to estimate wind power at a particular place (location) 72 hours in advance using data from the same spot in the past. The primary motivation for this effort is to construct a model for enhanced short-term wind power forecasting by using wind speed data obtained in-situ on the power plant's wind turbines. An additional focus of this work is the application of the suggested approach to comparable locations.

I utilize a one-year time series of 10 - minute average wind speed and wind power data collected at the sites of the wind turbines whose performance I wish to anticipate. The primary forecasting tool in the model I propose is a feed-forward neural network, paired with the Fuzzy C - Means Clustering and the K - Neighbors Nearest Classification.

1.2 Requirements of the topic

To achieve the results mentioned in Section 1.1, the study has the following requirements:

- Preprocessing Data (Convert 10 Minutes to 1 hour resolution, Remove outliers, Interpolation, Visualization, Normalization by Min – Max Scaler).
- Clustering Method (Fuzzy C – Means).
- Classification Method (K – Nearest Neighbors).
- Regression Method to Forecasting (Feed – Forward Neural Network).

1.3 Ideas and method implementation

The primary tool in this topic is machine learning, which is a field of artificial intelligence that helps computers learn from data to solve problems without being programmed in great detail. It is a vast field and has many facets to learn, so it's hard to say which programming language is the most suitable fit for it. And there is no definitive answer to the question of which programming language you should use for your AI project. However, here we will use the primary programming language, Python, to run on a code editor called Visual Studio Code. Problems related to Machine learning can mention as classification and prediction of loads, forecasting PV generation, weather forecasting, stocks, etc.

The data for grouping is a set of power consumption data collected by each 10 minutes resolution of the day. Then it will be converted to 1 hour resolution for short – term forecasting. But data often contain erroneous data (outliers), and data is lost (missing data) due to some reasons such as system failure, repair maintenance, breakdown, etc. There are many methods to handle the error data. This report will generate regression lines and then trim the skewed data based on that regression line. Besides, the re-interpolation of lost data also provides a sufficiently large data set for the algorithms, increasing the ability to calculate more accurately for later steps.

After preprocessing, objects will be clustered. The process of grouping objects in data according to the principle: objects in the same group are more similar than

objects in different groups. In machine learning, data clustering is considered an unsupervised learning problem, because it has to solve the problem of finding a structure in a data set of unknown class or training information. training. During data training, data clustering will initialize classes for classification by defining labels for data groups. Currently, there are many clustering algorithms such as K-means, Mean-shift, DBSCAN, and Spectral clustering, ... among those methods, the famous and easy-to-use method is the Fuzzy clustering algorithm. C – Means (Khanh Dinh Tran, Nguyen Duc Vuong, Manh Kien Tran, Michael Fowler, 2020). This method helps to group the primary data into subgroups based on the characteristic relevance of the data samples to each other. It also allows the user to find the centers of groups, which represent the distinct properties of each group.

After clustering, we will have a labeled wind speed and power data set; I proceed to classify the data using the K - Nearest Neighbors algorithm, which assumes that similar data will exist in proximity in space, from which our job will be to find the k point closest to the data to be tested. Finding the distance between two points has many formulas that can be used, such as Euclidean, Manhattan, and Minkowski, depending on the case I choose appropriately. I chose the Euclidean method in this report because it is the most basic and easy-to-use method.

Wind power generation forecasting approaches include physical models, statistical models, artificial neural network (ANN) models, and hybrid intelligence methods. The physical method, such as numerical weather prediction that takes into account initial values and boundary conditions, employs hydro- and thermo-dynamic models of the physics and atmosphere, resulting in poor predicting performance owing to the creation of accurate mathematical models. To forecast power generation, a statistical technique, such as probabilistic auto-regression and probability mass bias, establishes a link between wind power generation and explanatory factors. Because of its capacity to map nonlinear connections and learn from data samples, ANN is commonly used to estimate wind power output. The fundamental benefit of this

method is that no mathematical model is required to construct a link between input and output data in order to estimate wind power generation.

In this report, the FFN model (also known as Multilayer Perceptrons) will be used to predict the capacity of wind power plants. The popularity of ANN stems from its ability to be applied in various areas such as classification, clustering, pattern recognition, and prediction. ANNs, which have mostly superseded traditional regression and statistical models, are model types created by replicating the framework of biological neurons and how neurons interact with one another. The neural network-based system is trained using a set of input data known as the training data set and then evaluated using test data, with findings confirmed before data sets are forecasted. Weighted activation connects the neurons, and the network is represented by an input layer connected to information sources, a hidden layer made up of multiple neurons, and an output layer made up of information passed from the network to the signal output. A multi – layer neural network formed by four layers, a passive input layer with 72 nodes (24 nodes/day), two hidden layers (nodes can be adjusted) and an active output layer with 72 nodes (24 nodes/day). All input characteristics are gathered for wind power estimate and examined using the correlation approach to characterize the link between the input and output.

CHAPTER 2. THEORETICAL BACKGROUND

2.1 Developments about the wind power

Wind power is one of the most rapidly expanding renewable energy technologies. Global usage is increasing, owing in part to lower costs. According to the most recent IRENA statistics, global installed wind-generation capacity onshore and offshore has expanded by over 75 in the last two decades, rising from 7.5 gigatonnes (GW) in 1997 to 564 GW in 2018. Wind energy production more than quadrupled between 2009 and 2013, accounting for 16 percent of renewable energy generation in 2016. Wind speeds are high in many places, yet the ideal spots for generating wind power are occasionally isolated. Offshore wind generation has enormous potential.

The wind is used to generate electricity by using the kinetic energy provided by moving air. Wind turbines or wind energy conversion systems convert this into electrical energy. Wind initially strikes a turbine's blades, forcing them to rotate and thereby turning the turbine to which they are linked. This converts kinetic energy to rotational energy by rotating a shaft attached to a generator and produces electrical power via electromagnetism.

2.2 Wind potential and reserves

Wind energy is a significant sector in Taiwan. Taiwan has enormous wind resources, but due to a shortage of land, most large initiatives are offshore. There were 361 installed onshore turbines and 22 offshore turbines in service as of February 2020, with a total installed capacity of 845.2 MW. Taiwan has an abundance of wind energy resources. Taiwan's onshore wind farm capacity factor was 28-29 percent in 2013, with a total built onshore wind capacity of 530 MW. Its prospective offshore wind farm capacity factor is 33-38 percent.

Formosa 1 Offshore Wind Farm, Taiwan's first offshore wind farm, began commercial operation in April 2017 off the coast of Miaoli County. Swancor Renewable is in charge of the development project. The first stage of development

included the installation of two 4 MW wind turbines in November 2016 for a total generation capacity of 8 MW. The second stage of the construction, which included 20 Siemens Gamesa SWT-6.0-154 turbines with 120 MW, went online in December 2019. In late 2021, the Yunlin project produced its first power.

Diagram with installed wind power capacity and generation in Taiwan in recent years:

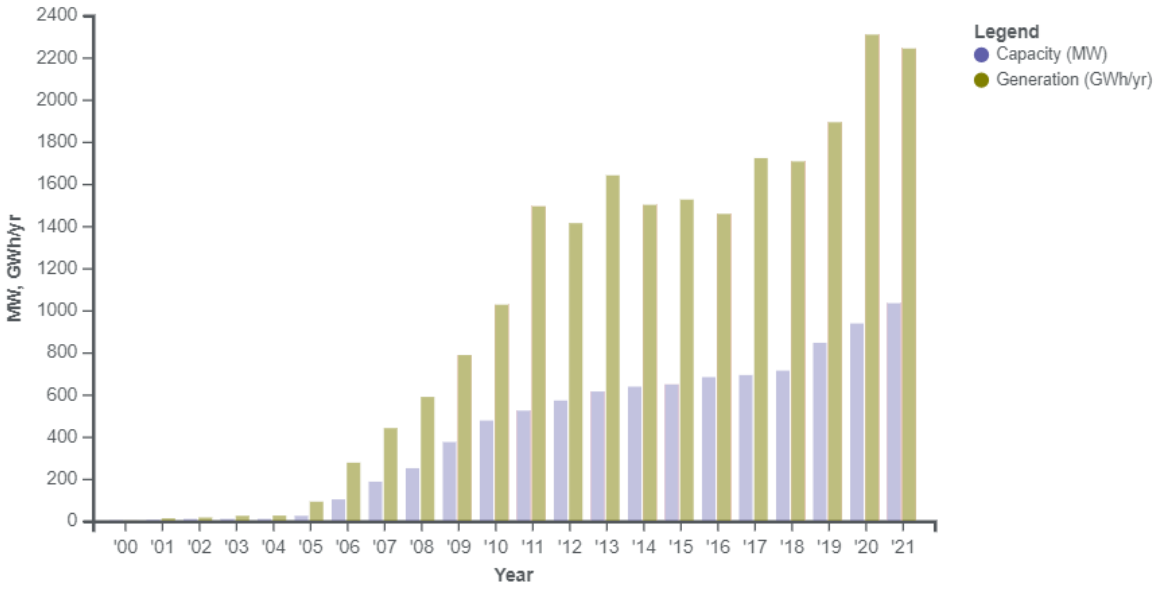


Figure 2-1 Installed wind power capacity and generation in Taiwan in recent years

2.3 Input parameters for forecasting model

Wind power forecasting is accomplished through the use of forecasting models. In order to anticipate wind power generated by turbines, these forecasting algorithms require input data. The input parameters that have been shown to be an effective predictor of wind power by the literature are taken into account. The input parameters were chosen based on correlation research published in the literature.

2.3.1 Wind speed

Wind speed is the most often utilized input metric for predicting wind power generated by turbines. Wind speed averages, such as hourly or monthly averages, have been used in the literature. Previous values of wind power and wind speed were

utilized as input factors to forecast wind power at a particular time. Senjyu, for example, has used wind speed forecasts several - hours ahead, such as data of every six hours interval wind speed has been used for the prediction of one day ahead. The amount of lag hours or days necessary to correctly estimate wind power has been established by completing an autocorrelation and cross-correlation study between several variables.

The equation below shows a well-known formula for converting wind speed to wind power. The swee[area of the blades is denoted by A(m^2). The p represents air density (kg/m^3), and the V represents wind speed (m/s). Air density may be computed as a function of temperature and pressure:

$$P = \frac{1}{2} A \rho V^3 \tag{2.1}$$

We can see from this equation that wind power production is proportional to the cube of wind speed. As a result, a method is necessary to estimate wind speed as accurately as possible, because the difference between anticipated and real wind speed is significant.

There are various plausible reasons why the literature prefers to anticipate wind power rather than wind speed: The equation may be used to anticipate wind power for a single turbine. However, the calculation fails to account for a wind farm's whole wind power production. A wind farm is a collection of turbines that generate wind electricity in the exact location. It is possible to calculate the wind power generated by one turbine using the predicted wind speed and multiplying it by the number of turbines in the wind farm, but this would result in a more significant forecasting error because essential details, such as shadowing effects or wake effects caused by other turbines, would not be taken into account. Second, because this equation takes a wind speed number obtained at one height, it ignores the wind speed profile, which is the relationship between wind speed values measured at different sizes.

The wind speed profiles were utilized to calculate an equivalent wind speed, which resulted in less dispersion in the power curve and hence improved power performance assessment.

2.3.2 Power energy

Wind energy is the energy content of airflow caused by its motion. This is known as kinetic energy, and it is a function of the fluid's mass m and velocity U, as given by:

$$E = \frac{1}{2}mU^2 \tag{2.2}$$

The power coefficient Cp quantifies the effectiveness of wind power extraction by the turbine. The power coefficient $C_p = P_T/P_{wind}$ reflects the ratio of power recovered by the turbine to the total capacity of the wind resource. As a result, the turbine power capture can never be more than the wind power, P_{wind}. The precise mechanism by which a turbine takes energy from the wind is a complicated area within meteorology and fluid dynamics that is outside the scope of this thesis and will not be addressed in full.

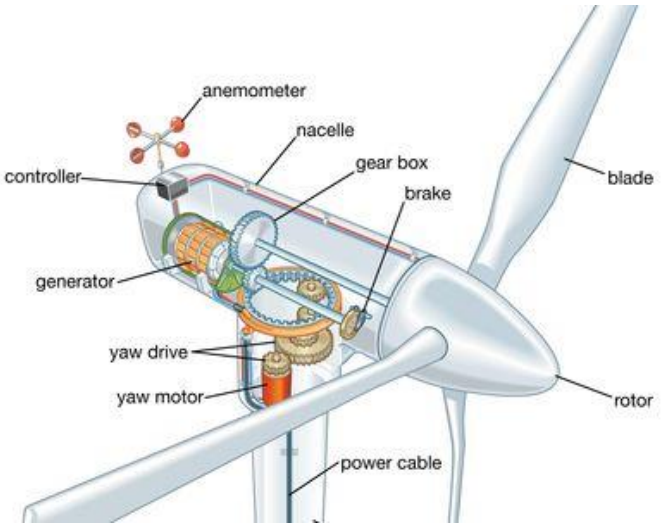


Figure 2-2 The component of wind turbines

The power curve of a wind turbine may be used to study the relationship between wind speed and power generation. The lowest speed necessary for the motor to start spinning and producing power may be found on the power curve. This wind speed is known as the wind turbine's cut-in speed. The power curve also indicates when the wind speed becomes too high for the turbine to produce energy, known as the cut-out speed. Because there is a risk of damage to the wind turbine at cut-out speed, the brakes will bring the rotor to a halt at this wind speed. Wind speed data is thus crucial in wind power forecasting with exogenous factors.

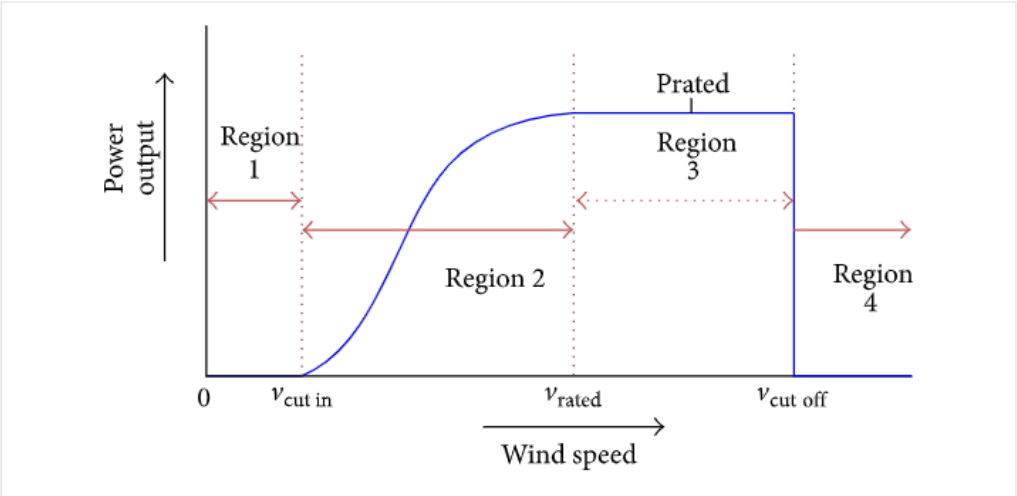


Figure 2-3 Typical power curve of a pitch regulated wind turbine

A WT's performance is indicated by its power curve. Accurate power curve models are critical power forecasts and online turbine monitoring tools. Several approaches for modeling the wind turbine power curve have been developed in various papers. Many academics have used these approaches, which employ data from manufacturer specs and actual data from wind farms, in various wind power applications. According to the reviewed literature, using the correct power curve models can enhance wind energy-based systems' performance. This study discusses the current state of research and the future directions of several wind turbine power curve modeling methodologies. The necessity for modeling, modeling technique, model categorization, and assessment methodologies have all been examined. The effect of various parameters on these curves, the conventional process for measuring

wind turbine power performance, and the requirement for designing site-specific curves are also highlighted. Different models suggested and employed in various investigations have been rigorously examined, and eventually, conclusions have been formed.

2.4 About Data

Through discussions with Taiwanese professors, with Beam's permission, I was provided with a data set with accurate and reliable data. The wind power plant of investigation is located in Taiwan. In this data set, the 2020 dataset with a resolution of 10 min is shown. Changhua Taiwan Power is an offshore wind power plant with a capacity of 109.2MW. The project is located in Taiwan. The project is presently in progress. It was created in a single period. The project construction commenced in 2020 and subsequently entered into commercial operation in September 2021. Taiwan Power created and owned the project. The project's wind turbines are erected on permanent foundations, and the concept includes a variety of jacket foundations. The project produces 360,000MWh of electricity and provides enough renewable energy to power 90,000 homes. The project will cost $800 million. The tubular steel towers built on the project site stand 90 meters tall.

The data set has a total of 4 components, the date column is evenly split from January 1, 2020 at 00:00 to December 31, 2020 at 23:50. Turbine power in kW. Wind speed in m/s and wind direction. The capacity of the wind power is 2000kWp.

CHAPTER 3. IMPLEMENTATION TOOLS

In this chapter, we focus into Unsupervised Learning with Fuzzy C – Means (FCM), Supervised Learning with K – Nearest Neighbors (KNN) and Artificial Neural Network (ANN) with Feed – Forward Neural Network (FNN).

3.1 Machine learning

3.1.1 Definition

"*Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed*" – Arthur Samuel, 1959.

As a definition of Wikipedia: Machine learning (ML) is the study of computer algorithms that can automatically improve through experience and by using data. It is seen as a component of artificial intelligence. Machine learning algorithms make a model based on sample data, known as "training data," to make forecasts or determinations without being explicitly programmed to do so. Machine learning algorithms are used in various applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is challenging or unworkable to generate conventional algorithms to perform the needed tasks.

AI is tied in with extricating information from information. It is an exploration field at the convergence of insights, man-made reasoning, and software engineering and is otherwise called prescient examination or factual learning. The use of AI techniques has lately become pervasive in regular day to day existence. From programmed suggestions of which motion pictures to watch, to what food to request or which items to purchase, to customized online radio and perceiving your companions in your photographs, numerous cutting edge sites and gadgets have AI calculations at their center.

Outside of business applications, AI has affected the way information driven exploration is done today. The devices presented in this book have been applied to assorted logical issues like agreement stars, tracking down far off planets, finding

new particles, investigating DNA groupings, and giving customized disease medicines.

However, your application shouldn't be as enormous scope or world-changing as these models to profit from AI. In this part, we will clarify why AI has become so well-known and talk about what sorts of issues can be tackled utilizing AI. Then, we will display you how to make your first machine learning model, introducing essential concepts along the way.

3.1.2 Supervised Learning

When we have a set of input variables $X = \{x_1, x_2, \dots, x_N\}$ and a corresponding set of labels $Y = \{y_1, y_2, \dots, y_N\}$, which x_i and y_i are vectors. Known data pairs $(x_i, y_i) \in X \times Y$ called as training data set. We need to create a function that maps each element from set X to a corresponding (approximate) element of the set Y from this training data set.

$$y_i \approx f(x_i), \forall i = 1, 2, \dots, N \tag{2.3}$$

Thanks to the iterative optimization of an objective role, supervised learning algorithms learn a role can forecast the output associated with some new inputs. An optimal role will permit the algorithm to determine the output for inputs correctly, which did not relative to a part of the training data. An algorithm that develops the accuracy of its outputs or predictions over time is said to have learned to perform that task.

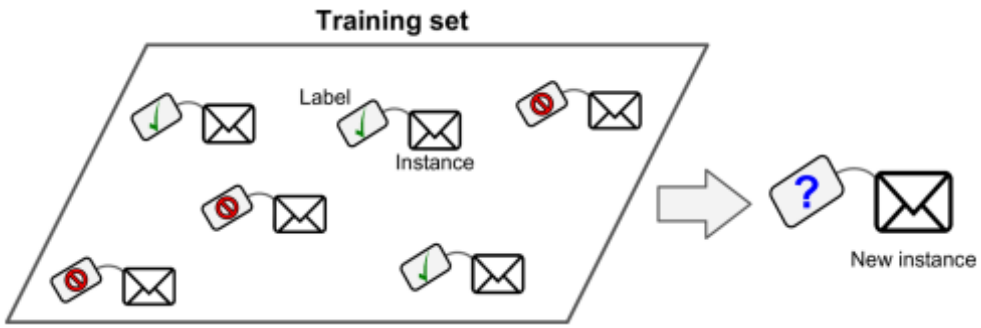


Figure 3-1 A labeled training set for spam classification (An example of supervised learning)

3.1.3 Unsupervised Learning

When we only have input X without knowing the corresponding label Y, we don't know the exact answer for each input data. The algorithms, learn from test data that has not been labeled, or categorized. Preference to responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of commonalities in each new piece of data.

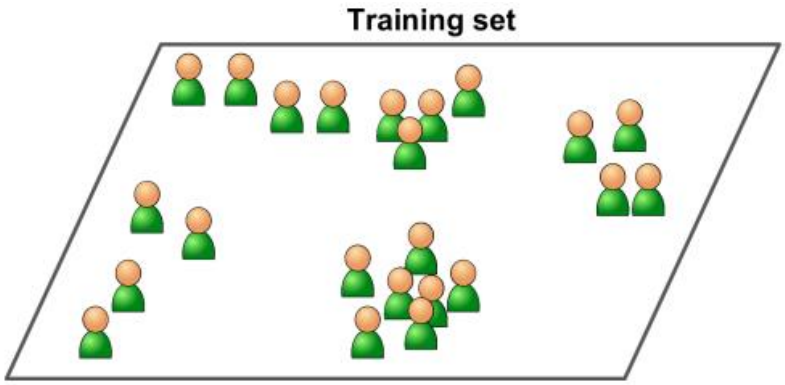


Figure 3-2 An unlabeled training set for unsupervised learning

3.1.4 Clustering

Clustering algorithms, on the other hand, divide data into discrete groups of comparable elements. Consider the uploading of images to a social networking platform. To help you manage your photos, the site may group photographs that depict the same individual. However, the site has no idea which photographs depict whom, or how many distinct persons feature in your photo collection. A logical strategy would be to take all of the faces and organize them into groups of faces that seem alike. Hopefully, these photographs belong to the same individual, and they may be grouped for you.

Visualization algorithms are also ideal examples of unsupervised learning algorithms since they are fed a large amount of complicated and unlabeled data. They generate a 2D or 3D representation of your data that may be plotted. These algorithms attempt to keep as much structure as possible so that you may understand how the

data is arranged and possibly uncover unexpected patterns.

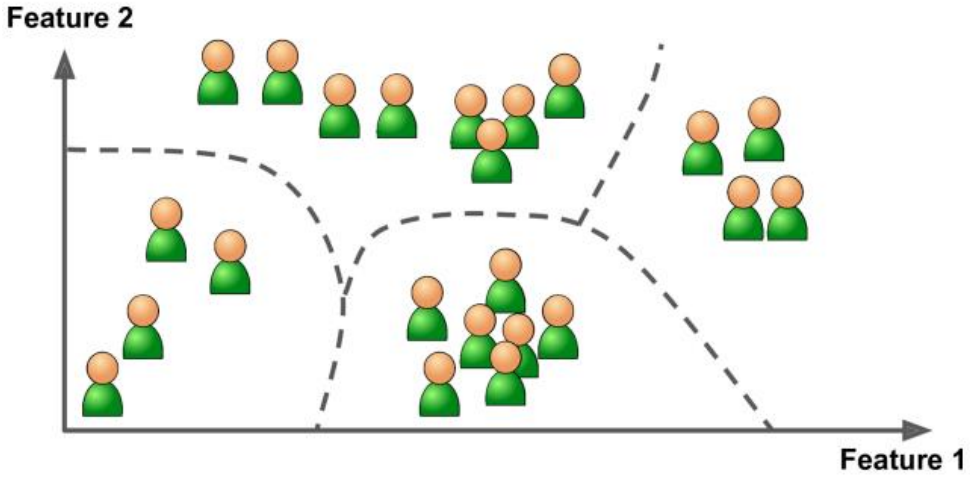


Figure 3-3 Clustering

3.1.5 Classification

Classification is a job that needs machine learning algorithms to learn how to assign a class label to problem domain instances. Classifying emails as "spam" or "not spam" is a simple example. There are many distinct sorts of categorization problems that you may meet in machine learning, as well as specific modeling techniques for each.

There are many types of Classification task in Machine Learning, they are:

- Classification Predicted Modeling.
- Binary Classification.
- Multi – Class Classification.
- Multi – Label Classification.
- Imbalanced Classification.

Classification is the process of classifying a set of data, which may be done on both structured and unstructured data. Predicting the class of provided data points is

the first step in the procedure. The classes are also known as the goal, label, or categories.

Approximating the mapping function from discrete input to discrete output variables is the problem of classification predictive modeling. The essential purpose is to determine which class/category the new data belongs to.

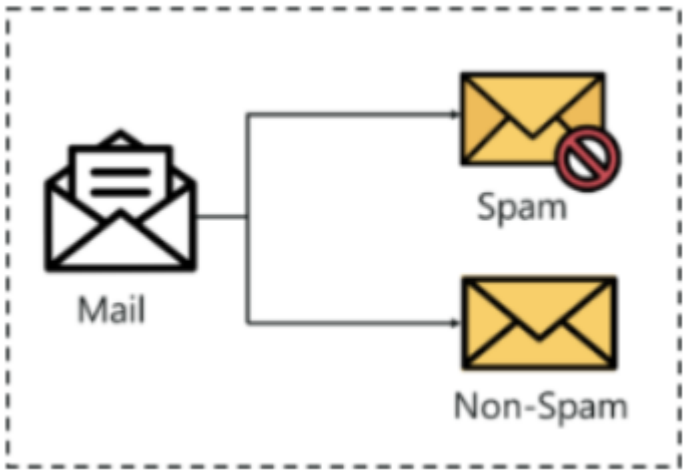


Figure 3-4 Classification

3.2 Programming language

Information technology is increasingly developed, so many programming languages support Machine Learning algorithms. Some popular programming languages like MATLAB, Python, R, Java,…

The programming language tool used in this project is Python. Python is a high-level computer programming language created by Guido van Rossum (1991).

Today, there are many Integrated Development Environments (IDE) - an integrated development environment for programming Python language and Visual Studio Code is the choice to implement the project because of the convenience of coding and running tests. easy algorithms for novices to use and code software.

CHAPTER 4. FORECASTING OF THE WIND POWER

4.1 Data preparation

Before the forecasting models are built, the dataset has to be prepared. Missing values are imputed, and feature vectors are processed in order to make it easier for the model to interpret them. This part also describes the splitting of data into training and test sets.

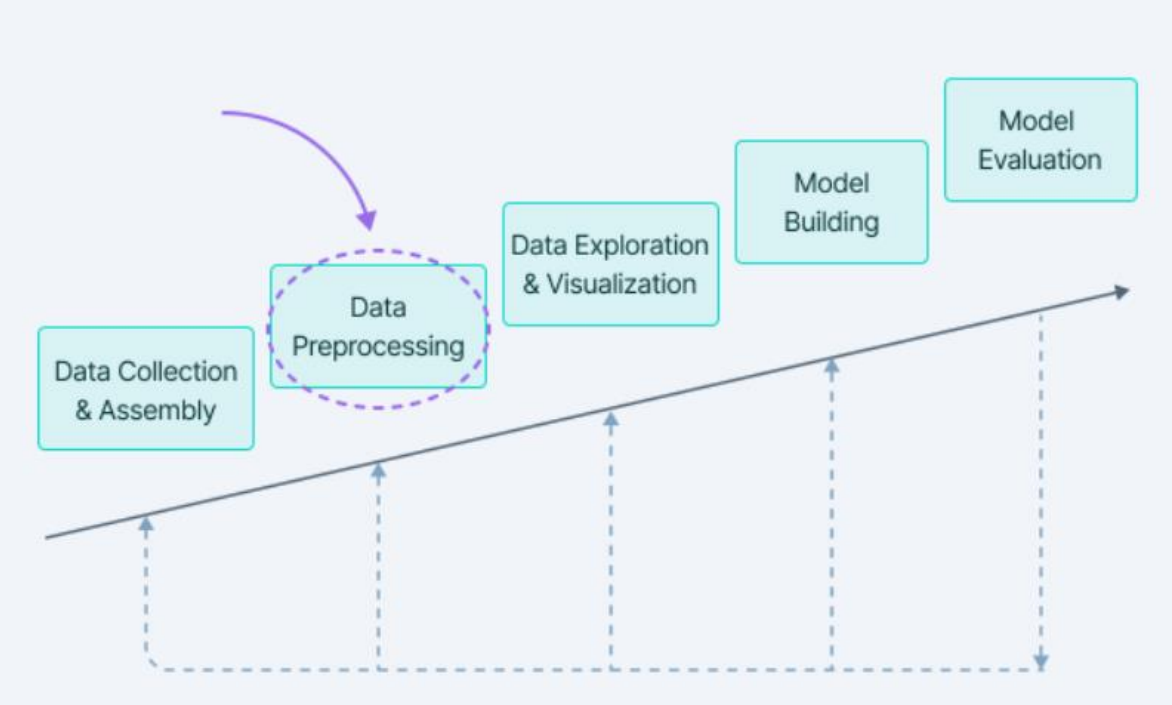


Figure 4-1 Data preprocessing is important

4.1.1 Convert data to 1 hour resolution

The original data was the 10 min resolution, so I have to convert the dataset to 1 hour resolution to use in the forecasting of the wind power. After doing this, the new dataset with 8704 data (original 52704 data) will be used.

4.1.2 Missing value and Interpolation

There are some missing forecasts in the dataset, which are regarded as missing values. To prevent losing information or training data, missing values are imputed

using the slope of the time series and the preceding time step's 72-hour prediction. The forecast's uncertainty will be higher, but it will be more accurate, and the data may be utilized for training.

I have used the interpolation to solve the missing values. Interpolation is a method for estimating unknown data points between two known data points. It frequently fills missing values in a table or dataset with previously known values. When the average does not match the missing numbers well, we must use a different strategy, which most people find to be an interpolation. Interpolation is commonly employed when working with time-series data since we prefer to fill missing values with the initial one or two values. Consider temperature; we would always choose to serve today's temperature with the past two days' mean rather than the month's. Interpolation may also be used to calculate moving averages.

$$y = y_1 + (x - x_1) \times \frac{(y_2 - y_1)}{(x_2 - x_1)} \quad (4.1)$$

In which:

- y: linear interpolation value.
- x: independent variable.
- x_1, x_2: values of the function at one point.
- x_2, y_2: values of the function at another point.

4.1.3 Remove outliers

Outliers are unusual values in your dataset that might skew statistical analysis and cause them to fail. Unfortunately, all analysts will be confronted with outliers and must determine what to do with them. Given the issues they might bring, you may believe it is preferable to eliminate them from your data. However, this is not always the case. Only for certain reasons should outliers be removed.

Outliers can provide valuable insight into the topic matter and data gathering procedure. It is critical to understand how outliers develop and whether they are likely to reoccur as a typical component of the process or research area. Unfortunately, it might be difficult to resist the desire to delete outliers improperly. Outliers increase data variability, which reduces statistical power. As a result of removing outliers, your results may become statistically significant.

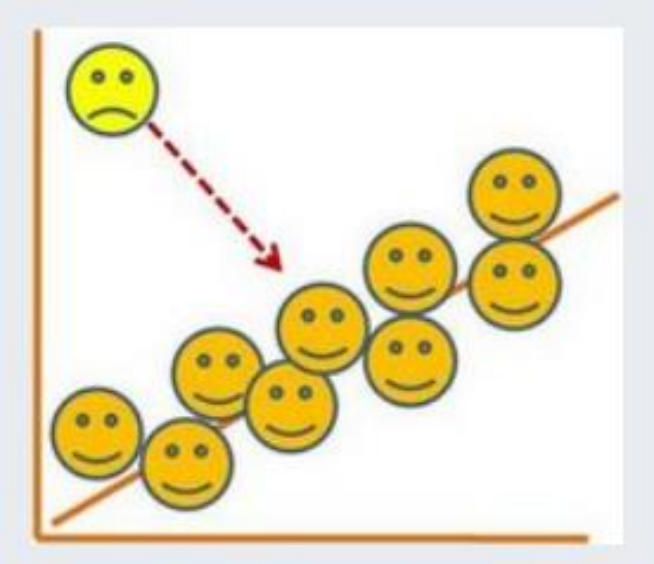


Figure 4-2 Outliers

Data visualize before remove outliers:

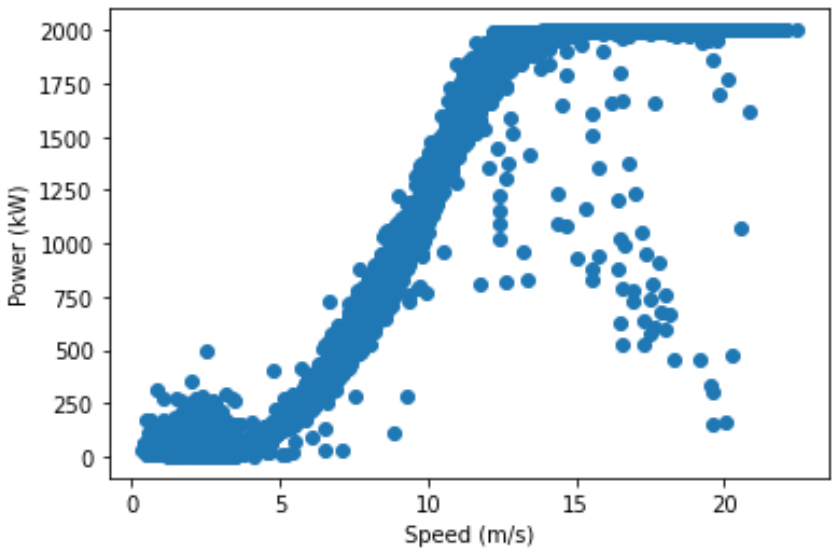


Figure 4-3 Data before cut outliers

So, let's come to some way to treat outliers:

- Trimming: It removes outlier values from our analysis. When we use this strategy, our data grows thinner since there are more outliers in the sample. Its primary benefit is its speed.
- Capping: In this strategy, we cap our outliers data and establish a limit, i.e., if the value is more than or less than a certain threshold, all values are deemed outliers, and the number of outliers in the dataset determines the capping number.
- Treat outliers as a missing value: By presuming outliers to be missing observations, handle them similarly to missing values.

In this project, I will treat the outlier by trimming and creating three Regression lines to cut one side of it, but you have to pay attention to limiting your Regression line by limit the upper of the cutting.

Create three Regression Line to cut outliers:

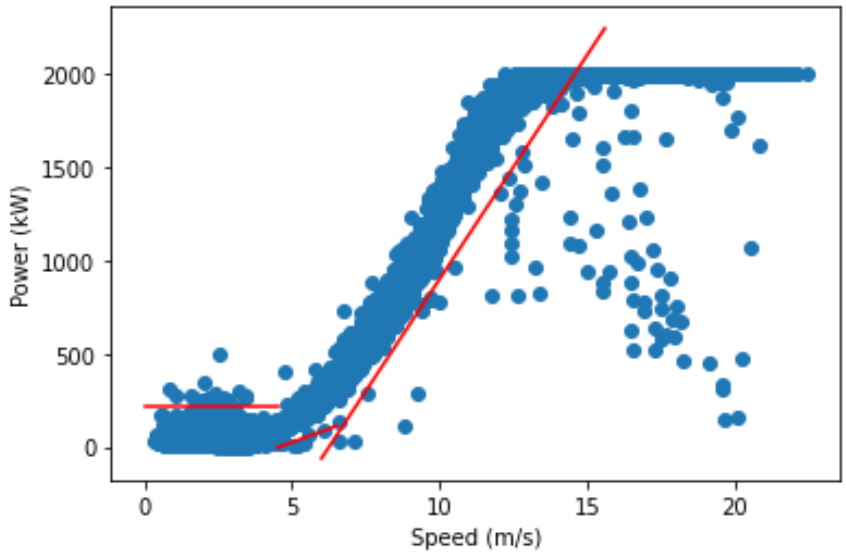


Figure 4-4 Three regression Line to cut outliers

Data after remove outliers:

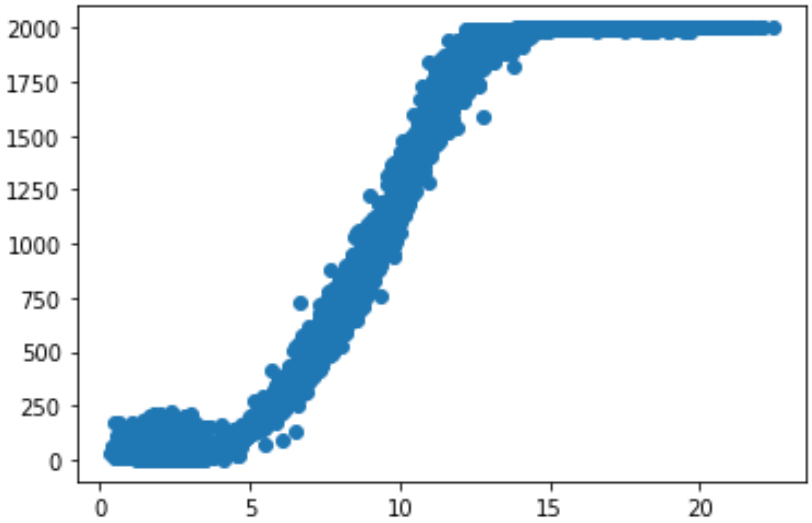


Figure 4-5 Data after remove outliers

4.1.4 Training, Test Data

Datasets are divided into two groups in machine learning. The training data is a subset of our actual dataset that is fed into the machine learning model to uncover and learn patterns. It trains our model in this way. The testing data is the other subset.

Typically, training data is more significant than testing data, and this is because we want to give the model as much data as possible to uncover and learn basic patterns. When we feed data from our datasets to a machine learning algorithm, it learns patterns and makes choices.

Testing data has two main criteria:

- Represent the actual dataset.
- Be large enough to generate meaningful predictions.

Test data is a last, real-world verification of an unknown dataset to ensure that the machine learning algorithm was trained correctly. In data science, it's common to see your data split 80 percent for training and 20% for testing.

Table 4-1 Split data for train and test

MODEL	TRAINING	TESTING
KNN	11 months, from 1st February to 31th December	3 days, from 7th to 9th in January
FNN	11 months, from 1st February to 31th December	3 days, from 7th to 9th in January

4.1.5 Normalization

The act of collecting each sample in the dataset and changing the values from their natural range to the model's operational range is referred to as data normalization. Normalizing the dataset aids in lowering neural network training time and minimizing the problem of exploding gradients. If the forecasting model employs the Euclidean distance, it may produce better results if all of the input attributes are scaled rather than supplying the data with raw values. If the dataset's characteristics comprise attributes on multiple number ranges, normalizing of the dataset aids in preventing qualities in larger numeric ranges from dominating attributes in smaller numeric ranges. Follow Vũ (2019), the normalization used for transforming the dataset in this thesis is calculated as:

$$x_i' = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \quad (4.2)$$

Where X' represents the normalized dataset, X represents the original dataset, and X_{min} and X_{max} represent the lowest and highest values in the dataset, respectively. Only the statistics from the training data are utilized for normalizing the dataset. Thus the model has no access to the values in the validation or test sets. The validation and test data are then normalized using the training data statistics.

Result of Min – Max scaler:

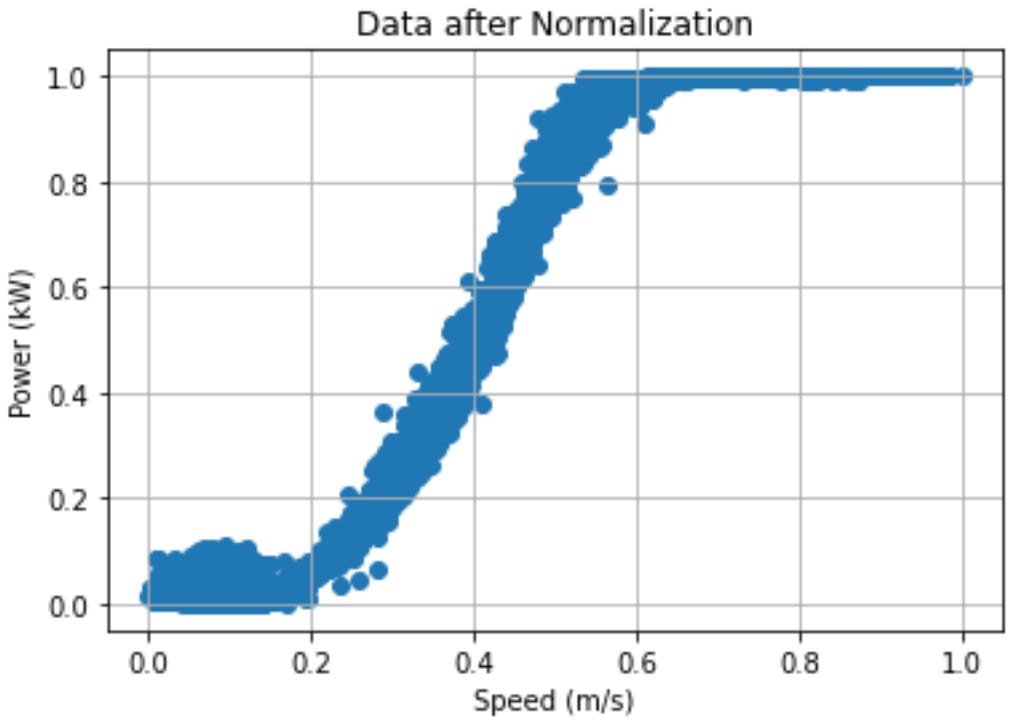


Figure 4-6 Data after Normalization

4.2 Clustering method (Unsupervised Learning) by Fuzzy C – means

4.2.1 Introduction

Fuzzy logic principles may be used to cluster multidimensional data, with each point assigned membership in each cluster center ranging from 0% to 100%. Compared to standard hard-threshold clustering, where each point is allocated a precise label, this can be pretty strong. This approach assigns membership to each data point associated with each cluster center based on the distance between the cluster center and the data point. The closer the data is to the cluster center, the more likely it is to belong to that cluster center. The total of each data point's membership should equal one.

An unsupervised clustering approach allows us to construct a fuzzy division from data. The system is based on a parameter m representing the degree of fuzziness of

the answer. When m is large, the classes get muddled, and all items tend to belong to all clusters. The optimization problem's solutions are determined by the parameter m. That is, various values of m will usually result in distinct partitions.

K – Means versus Fuzzy C – Means:

Let’s compare these two powerful algorithms to get a clear idea of where the fuzzy c-means algorithm fits in.

1. **Cluster attribution:** In fuzzy clustering, each point has a chance of belonging to each group, rather than just one, as in traditional k-means. In Fuzzy-C Means clustering, each point has a weighting associated with a particular cluster. As a result, a point does not sit "in a cluster" so much as it has a weak or considerable link to the cluster's center, determined by the inverse distance to the cluster's center.

2. **Speed:** Fuzzy-C means are slower than K because they perform more work. Each cluster evaluates each point, and each evaluation necessitates additional processes. K-Means necessitates a distance calculation, whereas fuzzy c means necessitates a complete inverse-distance weighting.

3. **Personal Opinion:** FCM/Soft-K-Means is "less silly" than Hard-K-Means when it comes to prolonged clusters (when points otherwise consistent in other dimensions tend to scatter along a particular size or two).

Pros and Cons:

Pros:

- Produces the best results for overlapping data sets and outperforms the k-means method.

- Unlike k-means, where data points must only belong to one cluster center, here data points are awarded membership to each cluster center, allowing them to belong to more than one cluster center.

Cons:

- Specifying the number of clusters in advance.
- We achieve a better outcome with a lower value of but at the price of more iterations.
- Underlying causes might be unequally weighted by Euclidean distance metrics.

The FCM algorithm's performance is determined by the initial cluster center and/or the starting membership value.

4.2.2 Standard Fuzzy C – Means (FCM) algorithm

The algorithm partitions a finite collection of N elements $X = \{X_1, X_2, \ldots X_n\}$ into a collection of C clusters with respect to some given criteria. Each element $X_i \in X$, i = 1, 2, …, n is a vector with d dimensions. We define a way to divide X into C clusters with cluster centers $V_1, V_2, \ldots, V_C$ in the centroid set V.

In the FCM algorithm, U is representative matrix for the membership of each element in each cluster. The matrix U has some characteristics as below:

- U(i,k) is the membership value of the element X_i in a cluster with center V_k, $1 \leq i \leq N; 1 \leq k \leq C$
- $0 \leq U(i,k) \leq 1, 1 \leq i \leq N; 1 \leq k \leq C$ and $\sum_{k=1}^{C} U(i,k) = 1$ for each X_i.
- The larger U(i,k) is, the higher the degree of confidence that the element X_i belongs to the cluster k.

If data are distributed in well separated groups, then crip clustering is used. But if the data are not partitioned into well separated clusters then fuzzy clustering is used. Mathematically, w_{ij} is the weight with which data x_i belong to cluster c_j.

- Data X = { $x_1, x_2, \dots, x_n$} – n is number of data.
- Cluster c = {$c_1, c_2, \dots, c_k$) – k is number of cluster.
- X_1 = {$x_{11}, x_{12}, \dots, x_{1d}$} – d is dimension of the data.

Condition:

1. All the weight for the a given data x_i, add up to 1 $\sum_{j=1}^{k} w_{ij} = 1$.
2. Each cluster c_j, contains, with non-zero weight, at least one data, but does not contain, with a weight of one, all of the data: $0 < \sum_{i=1}^{n} w\_ij < n$.

4.2.3 Evaluation metrics for Clusters

Most of the metrics used to evaluate clusters in other clustering methods are also applicable to Fuzzy C-Means. Despite the fact that these approaches are dependent on the expert's subject area, we will mention some typical measurements used to analyze the clusters generated below:

- A homogeneity study of the produced clusters.
- The clusters created using Fuzzy C-Means must be homogenous and distinct from other clusters.
- Each cluster's coefficient of variation analysis
- Pearson Correlation may be used to validate cluster quality.
- If we know ground truth cluster values, we may additionally examine accuracy, recall, and f-score.

4.2.4 The steps for the FCM algorithm

Input: the dataset, k numbers

Output: $w_{ij},\ c_j,\ 1 \le j \le k,\ 1 \le i \le n$, the partition of X into C clusters.

Steps:

1. Randomly assign values to all $w_{ij},\ 1 \le j \le k,\ 1 \le i \le n$
2. (re) calculate centroid of each cluster using the fuzzy p-seudo paritition.
3. (re) calculate the fuzzy p – seudo partition, w_{ij} assignment step.
4. Repeat step 2 and 3 if centroids don't change.

In more detail:

1. Initialization $\sum_{j=1}^{k} w_{ij} = 1$
2. Compute centroids: $c_j = \frac{\sum_{i=1}^{n} w_{ij}^2 x_i}{\sum_{i=1}^{n} w_{ij}^2}$
3. Updating the fuzzy p-seudo partition:

$$w_{ij} = \frac{\frac{1}{dist(x_i, c_j)}}{\sum_{n=1}^{k} \frac{1}{dist(x_i, c_n)}} \tag{4.3}$$

4. Visualization the clusters.

The result of Fuzzy C – Means algorithm:

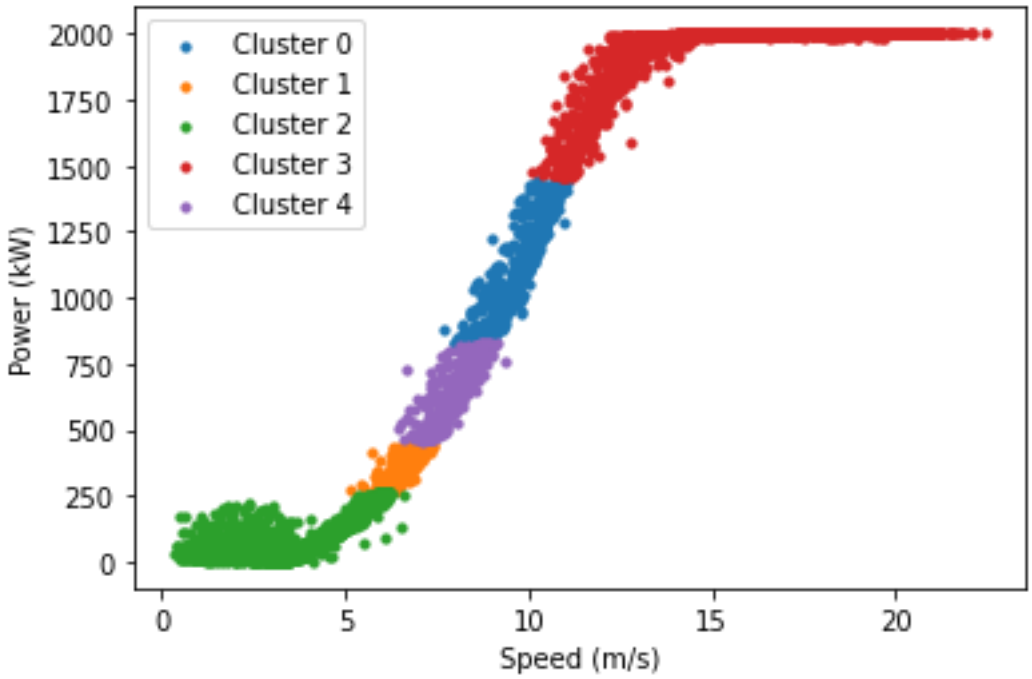


Figure 4-7 Result of clustering by FCM

Table 4-2 Number of Data and Centroid each Cluster

Cluster	Number of Data	Centroid
1	660	[1076.4665 9.7765]
2	748	[342.4413 5.8203]
3	3619	[187.1981 4.7498]
4	2023	[1822.9568 13.902]
5	697	[575.1565 7.2454]

4.3 Classification Method (Supervised Learning) by K – Nearest Neighbors

The k-nearest neighbors algorithm, often known as KNN or k-NN, is a non-parametric, supervised learning classifier that employs proximity to classify or predict the grouping of a single data point. While it may be used for either regression or classification issues, it is most commonly utilized as a classification technique, assuming that comparable points can be identified near one another.

A majority vote is used to apply a class label to a classification problem—that is, the label that is more commonly expressed around a specific data point is utilized. While officially this is referred to as "plurality voting," the term "majority vote" is more generally used in literature. The difference between these terms is that "majority voting" officially requires a majority of more than 50%, which only works when there are only two options. When there are numerous classes—say, four categories—you don't always need 50% of the vote to make a decision about a class; you might give a class label with a vote of more than 25%.

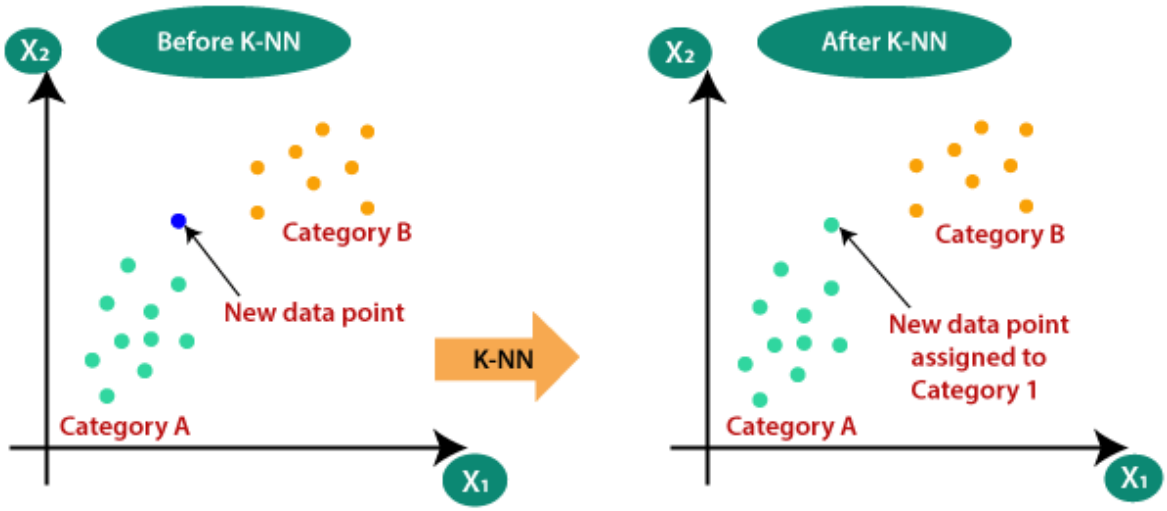


Figure 4-8 kNN for classification the new data point

4.3.1 Compute the distance metrics

The k-nearest neighbor algorithm's purpose is to find the nearest neighbors of a given query point so that we may apply a class label to it. KNN requires a few needs to do so: decide on your distance measurements.

The distance between the query point and the other data points must be determined to discover which data points are closest to a particular query point. These distance measures aid in forming decision borders, which divide query points into distinct areas, and Voronoi diagrams are often used to depict decision boundaries. While there are several distance metrics available, this thesis will only discuss the following:

- **Euclidean distance:**

This is the most often used distance metric, and it is confined to vectors with real values. Using the formula below measures a straight line between the query location and the other measured point.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2} \tag{4.4}$$

- **Manhattan distance:**

Another standard distance metric quantifies the absolute value between two places. It is also known as taxicab or city block distance. It is usually represented by a grid, indicating how one may get from one location to another through city streets.

$$d(x,y) = (\sum_{i=1}^{m} |x_i - y_i|) \tag{4.5}$$

- **Minkowski distance:**

This distance metric is a synthesis of the Euclidean and Manhattan distance metrics. Other distance metrics can be calculated using the parameter p in the method below. This formula represents Euclidean distance when p equals two, and Manhattan distance when p equals one.

$$d(x,y) = (\sum_{i=1}^{n} |x_i - y_i|)^{\frac{1}{p}} \tag{4.6}$$

- **Hamming distance:**

This method is commonly used with Boolean or string vectors to detect the spots where the vectors do not match. As a result, it is also known as the overlap metric. The following formula expresses this:

$$d_H = (\sum_{i=1}^{k} |x_i - y_i|) \tag{4.7}$$

4.3.2 The steps of the kNN algorithm

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Classification report.

In the k-NN algorithm, the k parameter specifies how many neighbors will be searched to determine the classification of a single query point. If k=1, for example, the instance is allocated to the same class as its single nearest neighbor. Different values of k might lead to overfitting or underfitting; therefore, defining it can be a balancing act. Lower k values can have high variance but low bias, while higher k values can have a high tendency but a low variation. The value of k will be determined mainly by the input data since data with more outliers or noise would most likely perform better with larger values of k. Overall, an odd number for k is suggested to avoid ties in classification, and cross-validation techniques can assist you in determining the best k for your dataset.

In this thesis, I use the Euclidean distance because of the ease to use. Then the final value of k was chosen between the calculated of k plus one or minus 1. In this thesis, the **k was equal to 35.62**, so **the k – the number is 36**.

4.3.3 Confusion matrix and Classification report

- **Confusion Matrix**

A Confusion matrix is a N x N matrix used to assess the effectiveness of a classification model, where N represents the number of target classes. The matrix compares the actual goal values to the machine learning model's predictions. This

gives us a comprehensive picture of how well our classification model is working and the kind of errors it produces. For example, for a binary classification task, we would have a two × two matrix with four values, as illustrated below:

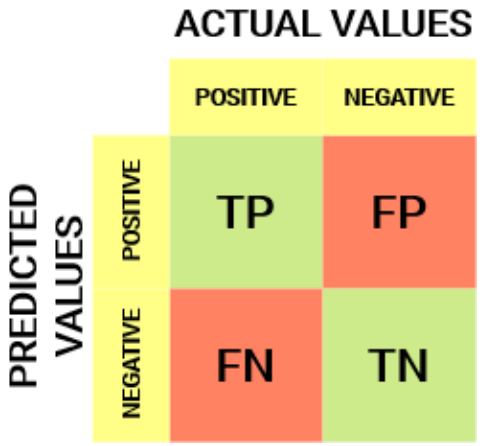


Figure 4-9 Binary Classification with 4 values

Let’s decipher the matrix:

- The target variable has two possible outcomes: positive or negative.
- The columns represent the target variable's actual values.
- The rows indicate the target variable's expected values.

Let’s understand each term of the example below:

- True Positive (TP):
  - The projected value corresponds to the actual value.
  - The actual result was positive, and the model predicted it would be positive.
- True negative (TN):
  - The projected value corresponds to the actual value.
  - The exact number was negative, and the model anticipated it would be negative.
- False positive (FP):
  - The anticipated value was incorrect.

- The model anticipated a positive result, but the actual value was negative.

- False negative (FP):
  - The anticipated value was incorrect.
  - The model projected a negative result, while the actual value was positive.

The result of the confusion matrix:

```
Confusion Matrix:
 [[ 8  0  0  0  2]
 [ 0  3  0  0  0]
 [ 0  0 16  0  0]
 [ 0  0  0 27  0]
 [ 1  3  0  0  7]]
```

Figure 4-10 Confusion matrix

The different values of the Confusion Matrix would be as follows:

- The number of class data points were correctly classified by the model are: 8 class data for model 1; 3 class data for model 2; 16 class data for model 3; 27 class data for model 4 and 7 class data for model 5.
- The number of class data points were not correctly classified by the model are: 2 class data for model 1; 4 class data for model 5.

The **accuracy score** function computes the accuracy as either a fraction or a count of correct predictions. The function returns the subset accuracy in multilabel classification. The subset accuracy is 1.0 if a sample's complete collection of predicted labels precisely matches the real set of labels; otherwise, it is 0.0. This is how I calculated the accuracy:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (4.8)$$

After using the kNN for the dataset, the **accuracy gets to 0.91**. This accuracy is reliable for using this classification for the Feed–Forward Neural Network.

- **Classification report**

The report, as the name implies, explains everything about the categorization. This summarizes the classification quality achieved by the built ML model. It consists mainly of 5 columns and (N+3) rows. The first column contains the name of the class label, followed by Precision, Recall, F1-score, and Support. N rows represent N class labels, whereas the remaining three represent the accuracy, macro average, and weighted average.

- **Precision** is calculated in relation to the expected values. The precision for class-A is defined as the proportion of total predictions that really correspond to class-A in the actual dataset. It is the product of the [i][i] cell of the confusion matrix with the sum of the I columns.

$$Precision = \frac{TP}{TP+FP} \quad (4.9)$$

- **Recall:** This is determined in relation to the actual values in the dataset. The recall for class-A is defined as the proportion of total items in the dataset that were categorized as class-A by the ML model. It is the product of the [i][i] cell of the confusion matrix with the sum of the I row.

$$Recall = \frac{TP}{TP+FN} \quad (4.10)$$

- **F_1-score:** It is the summing of accuracy and recall.

$$F_1 - score = \frac{2}{\left(\frac{1}{Recall}\right)+\left(\frac{1}{Precision}\right)} \quad (4.11)$$

- **Support:** It is the total number of items in each class in the real dataset. It's just the total of the rows for each class-i.
- **Macro average:** the average of precision/recall/f_1 – score.
- **Weighted average:** the weighted average of precision/recall/f_1 – score.

	precision	recall	f1-score	support
0.0	0.89	0.80	0.84	10
1.0	0.50	1.00	0.67	3
2.0	1.00	1.00	1.00	16
3.0	1.00	1.00	1.00	27
4.0	0.78	0.64	0.70	11
accuracy			0.91	67
macro avg	0.83	0.89	0.84	67
weighted avg	0.92	0.91	0.91	67

Figure 4-11 Classification report

4.4 Regression Method to Forecasting by Feed – Forward Neural Network

Feed - Forward Neural Networks are artificial neural networks in which the connections between nodes do not create a loop. They are biologically inspired algorithms with layers of neuron-like features. Neural networks are made up of linked components known as nodes. Data enters the network at the input point and travels through each layer before reaching the output. The connections, however, vary in strength or weight. The number of connections provides essential information about a network.

Feed - Forward Neural networks are sometimes called multi-layered neural networks (MLN). The neuron network is called feedforward because information only travels through the network via the input nodes. Because there is no feedback connection, network output is transmitted back into the network rather than flowing out.

Here are some of the reasons why feedforward networks outperform standard models:

- Traditional models, such as the Perceptron, take factual inputs and provide Boolean output only if the facts can be separated linearly. This means that the positive and negative points should be on different sides of the

boundary. It is also relatively simple to select the best alternative for separating the good and negative aspects.

- The output of the sigmoid neuron model is smoother than that of the perceptron type.
- Feedforward neural networks employ sigmoid neurons to overcome the limitations of older models such as perceptrons to process non-linear input successfully.
- With convolutional neural networks and recurrent neural networks providing cutting-edge performance in computer science, they are finding extensive use in various fields to solve complex decision-making difficulties.

4.4.1 Structure and terminology

The basic structure of a neural network is an ensemble of neurons coupled to levels known as layers. This structure is based on the human brain. In this project, we will describe the FNN approach used in this research.

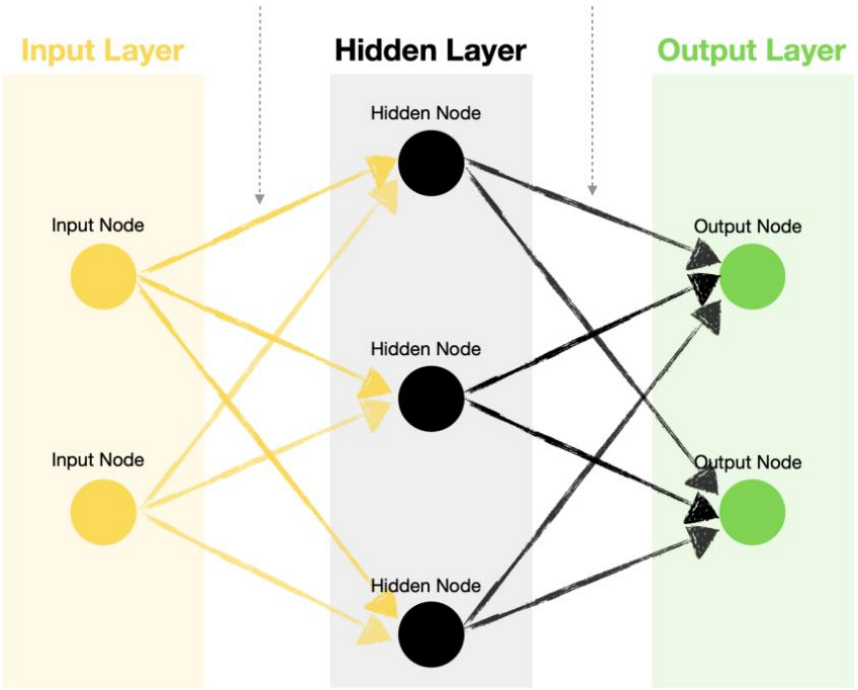


Figure 4-12 Basic structure of a Feed Forward Neural Network

The neural network has three levels: the input layer, the hidden layer(s), and the output layer. The neural network is entirely linked. Every node in one layer is connected to every node in the next layer. However, the nodes are not connected in the same layer. Each link between two nodes has a weight w_{ij}; i is the node, while j is the layer.

Input Layer — is made up of one or more input nodes. For instance, assume you want to forecast whether it will rain tomorrow based on two variables: humidity and wind speed. For example, your first input would be the humidity value, and your second input would be the wind speed value.

Hidden Layer — This layer contains hidden nodes, each with an activation function (more on these later). Deep Neural Networks are Neural Networks with numerous hidden layers.

Output Layer — is made up of one or more output nodes. Using the weather forecast example, you could have only one output node provide a rain probability (where >0.5 indicates rain tomorrow, and 0.5 indicates no rain tomorrow). You may also have two output nodes, one for rain and one for no rain. It should be noted that you can use a different activation function for output nodes than for concealed nodes.

Connections — Connections are lines that connect distinct nodes. These include kernels (weights) and biases, which are the parameters tuned during neural network training.

4.4.2 Parameters and activation functions

Follow John Pomerat, Aviv Segev, Rituparna Datta (2019) and Christopher Bourez's Blog (2016):

Kernels (weights) — Weights allow the artificial neural network to increase or decrease the number of connections between neurons. Assume you build an artificial neural network to discriminate between different dog breeds. Each neuron in one layer of the neural network may concentrate on a distinct trait, such as the snout, ears, eyes, tail, size, form, and color. With weighted inputs, the network may adjust the

strength of the link between each neuron in this layer and the neurons in the next layer, for example, to place less emphasis on the tail and more focus on the size and shape.

Biases — While weights allow an artificial neural network to change the strength of connections between neurons, bias may be used to change the strength of connections inside neurons. Bias can be positive or negative, raising or reducing the output of a neuron. The neuron collects and sums its inputs, then adds bias (positive or negative) and sends the total to the activation function.

Cost function – A cost function or loss function reflects the model's accuracy. Its output notifies the neural network whether the model's weights and biases need to be modified. Consider the cost function to be a way of rewarding or punishing the machine for success or failure. It allows the device to learn from its successes and failures. As a result, loss functions may be used to train a neural network. They compute the loss, which is the difference between the output and the goal variable, given an input and a target. Loss functions are classified into four types:

- Regressive loss functions: They are employed in regressive issues when the goal variable is continuous. Mean Square Error is the most commonly used regressive loss function. Absolute error — quantifies the mean absolute value of the element-wise difference between input; Smooth Absolute Error is a smooth variation of the Abs Criterion.

- Classification loss functions: In a classification problem, the output variable is often a probability value f(x), referred to as the score for the input x. In general, the size of the score signifies our prediction's confidence. The target variable y is a binary variable, with 1 indicating true and -1 indicating false. Binary Cross Entropy, Negative Log Likelihood, Margin Classifier, and Soft Margin Classifier are some classification methods.

- Embedding loss functions: This method is used to determine if two inputs are similar or dissimilar. Here are several examples: L1 Hinge Error- computes the L1 distance between two inputs; Cosine Error- computes the cosine distance between two inputs.

Activation functions — Consider activation functions to be standard curves (building blocks) that the Neural Network may utilize to produce a bespoke curve that fits the training data. When various input values are sent through the network, different parts of the standard curve are selected, which are then integrated into a final custom-fit curve.

There are several activation functions to select from, the most common of which are Softplus, Softmax, ReLU, and Sigmoid. The forms and formulae of some commonly used activation functions in Neural Networks are shown here:

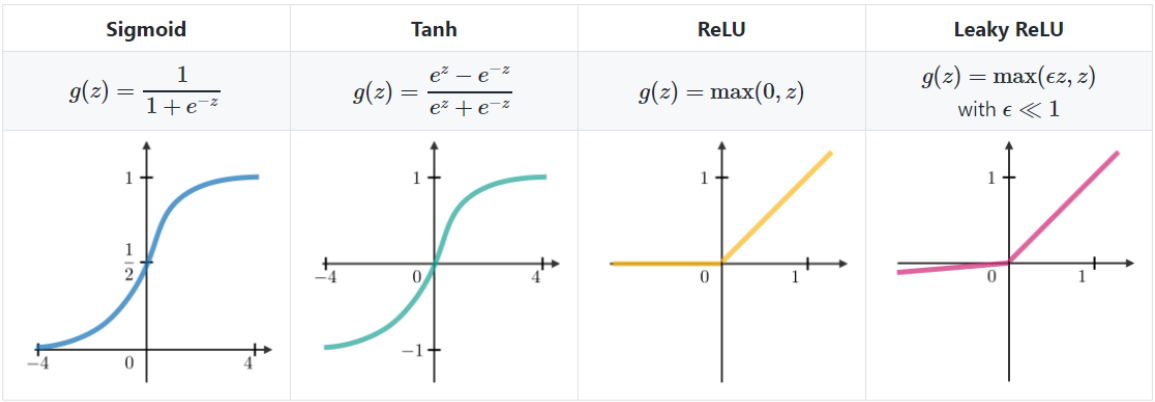


Figure 4-13 Sigmoid, Tanh, ReLU activation functions

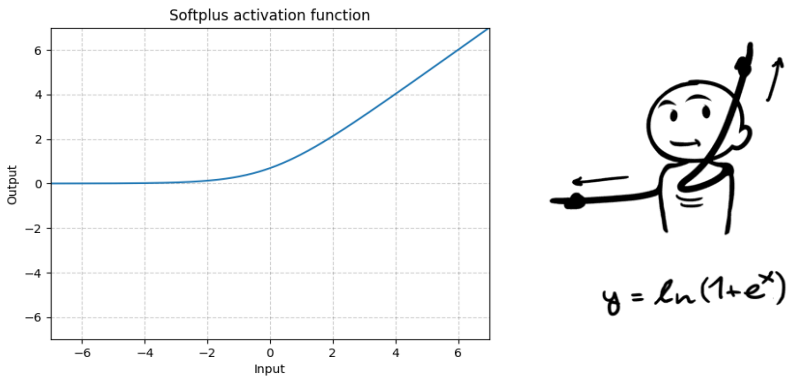


Figure 4-14 Softplus activation functions and formula

Many factors must be addressed for better performance and less erroneous results, including the number of hidden layers in a network, training techniques, hyperparameter tweaking, and so on, with the most significant activation function. Choosing the best activation function for a given activity can be time-consuming and require much research and study.

There are no hard and fast rules for picking any activation function. However, the choice of activation function is context-based, meaning the job at hand determines it. Different Activation Functions each have advantages and disadvantages, depending on the sort of system that we are developing.

Optimisation Functions – are the backpropagation algorithms. An optimizer's purpose is to determine the best collection of kernels (weights) and biases to minimize the loss. Optimizers often employ a gradient descent technique, which allows them to iteratively identify the "optimal" feasible weight and bias combination. The most commonly used ones are Stochastic Gradient Decent (SGD), Adagrad and Adam.

- Gradient Descent computes the gradient for the whole dataset and updates values in the opposite direction as the gradients until a local minimum is found. In contrast to standard Gradient Descent, which only conducts one update, stochastic Gradient Descent does a parameter update for each training sample. As a result, it is lot quicker. Gradient Decent techniques may be enhanced further by adjusting key factors such as momentum, learning rate, and so on.

- Adagrad is better suited for sparse data sets since it performs large updates for infrequent parameters and tiny updates for frequent parameters. It employs a variable learning Rate for each parameter at each time step, based on the previous gradients computed for that parameter. As a result, we do not need to manually adjust the learning rate.

- Adam is an abbreviation for Adaptive Moment Estimation. It also computes various learning rates. Adam practices well, is speedier, and excels other ways.

In my thesis, There is no rule of thumb to select the weights. You could start with random values of the same order, say - 1 to 1 or -0.5 to 0. 5, Same does for biases. The **Softplus activation function was used for the Hidden Layers**, while **Sigmoid was chosen for the Output Layer**; reasonable that the derivative of the softplus function was the logistic function, ReLU and Softplus were broadly similar, except near 0(zero) where the Softplus was enticingly smooth and differentiable, while the curve of sigmoid and output layer was similar. The output of a sigmoid function, when graphed, creates a "S" shape. Because they allow for modest modifications within a narrow range (0.0 to 1.0 on the vertical access) throughout the machine learning process, sigmoid functions are often utilized in neural networks. I used the **Mean Squared Error as the Cost Function** (Loss Function) because of the most commonly used regressive loss function. Adagrad and Adam generated better results than SGD, although they required more computing time. Adagrad was somewhat speedier than Adam. Thus, while utilizing a specific optimization function, one must choose between more processing power and optimal outcomes, so finally, I select **Adam for Optimisation Function**.

4.4.3 Normal distribution

According to Jim Frost (2020), for independent, random variables, the normal distribution, often known as the Gaussian distribution, is the most important statistical probability distribution. Most people are familiar with the bell-shaped curve found in statistics reports.

The normal distribution is a continuous probability distribution that is symmetrical around its mean, with most observations clustering around the center peak and probabilities tapering off equally in both directions. Extreme values in both

distribution tails are similarly uncommon. While the normal distribution is symmetrical, it is not the only balanced distribution. For example, the Student's t, Cauchy, and logistic distributions are symmetric. The normal distribution, like any other probability distribution, defines how the values of a variable are distributed. Because it adequately captures the distribution of values for many natural occurrences, it is the most important statistical probability distribution. Normal distributions are typically observed in characteristics that sum up numerous distinct processes. Height, blood pressure, measurement error, and IQ scores, for example, follow the normal distribution.

There are several other reasons why the normal distribution is crucial in statistics:

- Some statistical hypothesis tests are based on the assumption that the data follows a bell curve. However, as I explain in my essay about parametric and nonparametric tests, there is more to it than just whether or not the data is regularly distributed.
- Both linear and nonlinear regression assumes a Gaussian distribution for the residuals. Learn more about judging residual plots in my post.
- According to the central limit theorem, as sample size rises, the sampling distribution of the mean follows a normal distribution even if the underlying distribution of the original variable is non-normal.

The normal probability density function formula appears to be somewhat tricky. To utilize it, you need all the population mean and standard deviation. For every value of x, enter the mean and standard deviation into the formula to calculate the variable's probability density for that value of x.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{4.12}$$

Explanation:

- F(x): probability
- x = value of the variable
- μ = mean
- σ = standard deviation
- σ2 = variance

In this thesis, I will add the Normal Distribution consecutively from the 10% Error to 50% Error; this is because the original data of Wind Speed is actual values, but I just have the permission to predict the Wind Power. On the other hand, Wind Power prediction is often used to predict the height of the wind turbine shorter than 10m, but the reality wind turbine is higher.

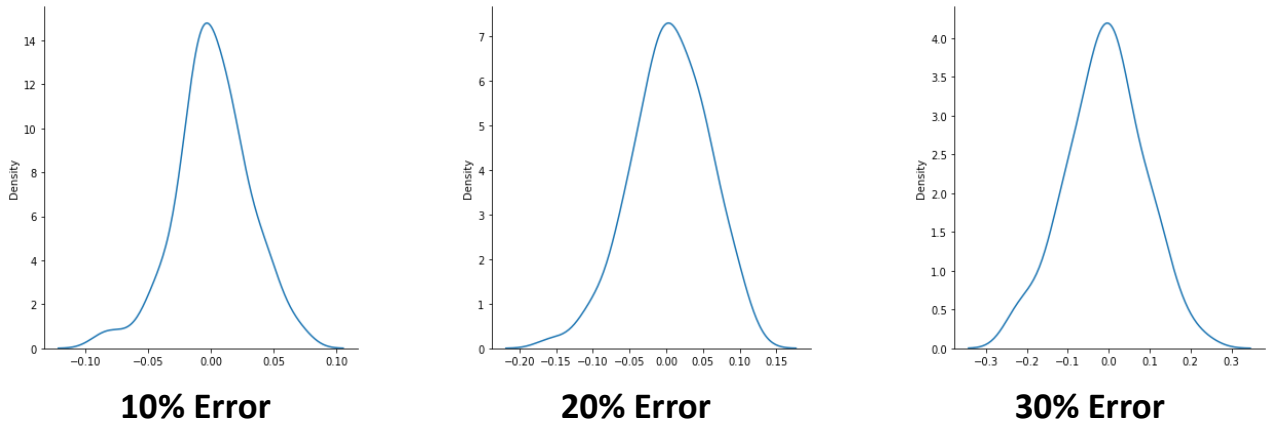


Figure 4-15 Normal Distribution at 10%, 20% and 30% Error

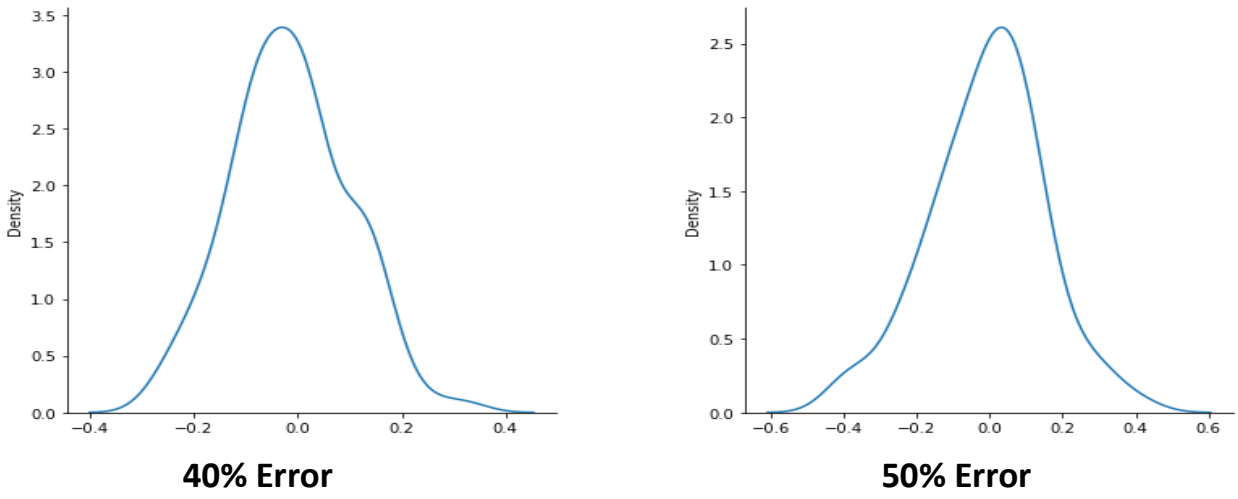


Figure 4-16 Normal Distribution at 40% and 50% Error

4.4.4 Learning the algorithm

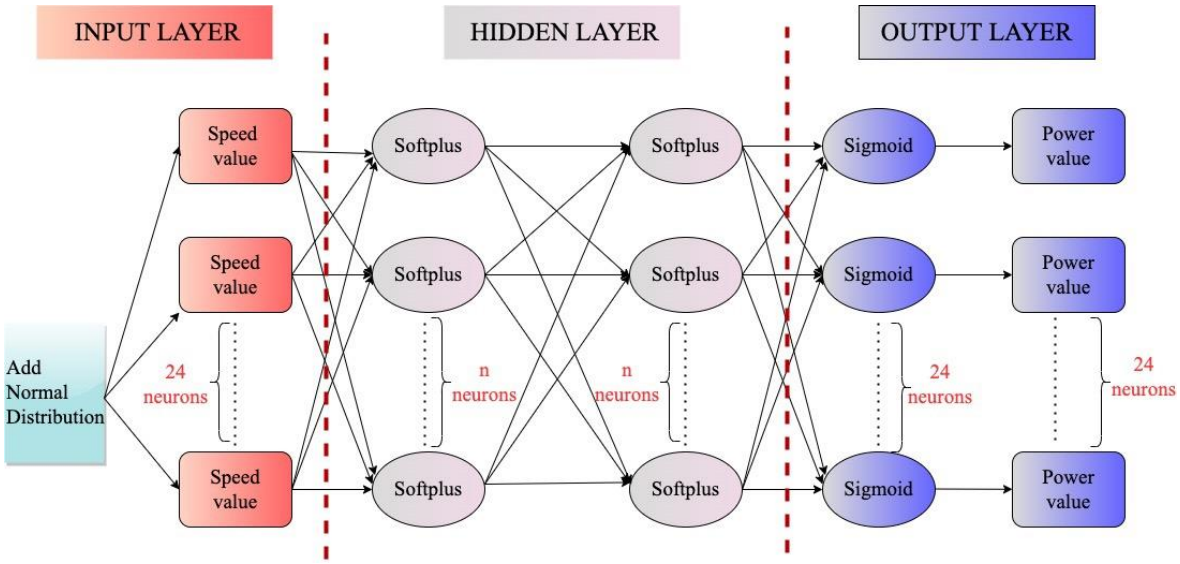


Figure 4-17 Structure of the FNN model

Each node in the hidden layers performs a mapping of the input vector x to a scalar value that will act as input to the next layer. The mapping performed by a neuron i in layer l can be described as (Gonzalez & Woods, 2018)

$$z_i(l) = \sum_{j=1}^{n_{l-1}} w_{ij}(l) v_j(l-1) + b_i(l) \qquad (4.13)$$

For i = 1, 2, …, n_l and l = 2, …, L; where L is the total number of layers, z_i is the total input to neuron i in layer l, w_{ij} is the weight that connects the output of neuron j to the input of neuron i, $v_j(l-1)$ is the output of neuron j in the previous layer ($l -$

1), and b_i is the bias value associated with the i^{th} neuron. This mapping of the input layer to the output layer is referred to as a forward pass through the network. The output of neuron i in layer l is given by (Gonzalez & Woods, 2018):

$$v_j(l) = h(z_i(l)) \tag{4.14}$$

For i = 1, 2, …, n_l; where h is an activation functions. The output function is an important feature in the network that enables the network to learn non – linear data patterns. Essentially, it converts the output signal of a previous node into information to be passed to the next node. Various activation functions can be utilized in the network.

The figure shows that the input layer corresponds with the input variables xi; in my research, these are the wind speed after adding the normal distribution. The algorithm runs five times; each time represents each value of the error of the normal distribution. The neurons from the input layer are connected with the hidden layer and are each affected by weights and biases. The input of the hidden layer is a summation of the inputs; then, it will transform based on a specific activation function called Softplus. The output of the first Hidden Layer is also affected by weight and passes to the input of the next hidden layer. The outputs of the second hidden layer are also affected by weights. The input of the outputs layer is a summation of the outputs of the second hidden layer; then it will plus the bias before transformed by the activation function called Sigmoid to create its final outputs.

In my thesis, there are 5 clusters corresponding to 5 models. Each model has the same structure but different nodes for each hidden layer. Here is the configuration of each model:

Table 4-3 Structure of each model

MODEL	Number of neurons in input layers	Number of hidden layers	Number of neurons for hidden layers	Number of neurons in output layers
Model 1	24 nodes	2 nodes	16 nodes	24 nodes
Model 2	24 nodes	2 nodes	16 nodes	24 nodes

Model 3	24 nodes	2 nodes	40 nodes	24 nodes
Model 4	24 nodes	2 nodes	36 nodes	24 nodes
Model 5	24 nodes	2 nodes	30 nodes	24 nodes

4.4.5 Mean Root Error

In another thesis, the error metrics used in this article to evaluate the performance of a forecasting model were Mean Squared Error—MSE and the Mean Absolute Error—MAE; these errors refer to the wind speed forecasts. The Normalized Mean Absolute Error—NMAE was used to illustrate the performance of the power generation. The equation used to calculate it is defined as follows:

$$NMAE = \left(\frac{100}{N \times P_{inst}}\right) \times \sum_{t=1}^{N} |e_P(t+k|t)| \quad (4.13)$$

Where k is the forecasting horizon (the number of time steps); and N is the number of data points utilized for model evaluation; P_{inst} – installed wind power capacity; $e_P(t+k|t)$ – error corresponding to time $t+k$ for the power generation projection provided at time t of origin.

The Mean Root Error—MRE was used to illustrate the performance of the power generation. The MRE that we are using now has the same formula with NMAE (Normalized Mean Absolute Error) in some papers.

$$MRE = \frac{1}{N} \sum_{i=1}^{N} \frac{|P_{fore} - P_{true}|}{P_{cap}} \times 100\% \quad (4.14)$$

The MRE results:

Table 4-4 NMAE (MRE) results

ADD NORMAL DISTRIBUTION TO THE WIND SPEED	MRE
10% ERROR	2.757

20% ERROR	3.396
30% ERROR	5.163
40% ERROR	5.177
50% ERROR	6.592

The result of the 3 – days prediction:

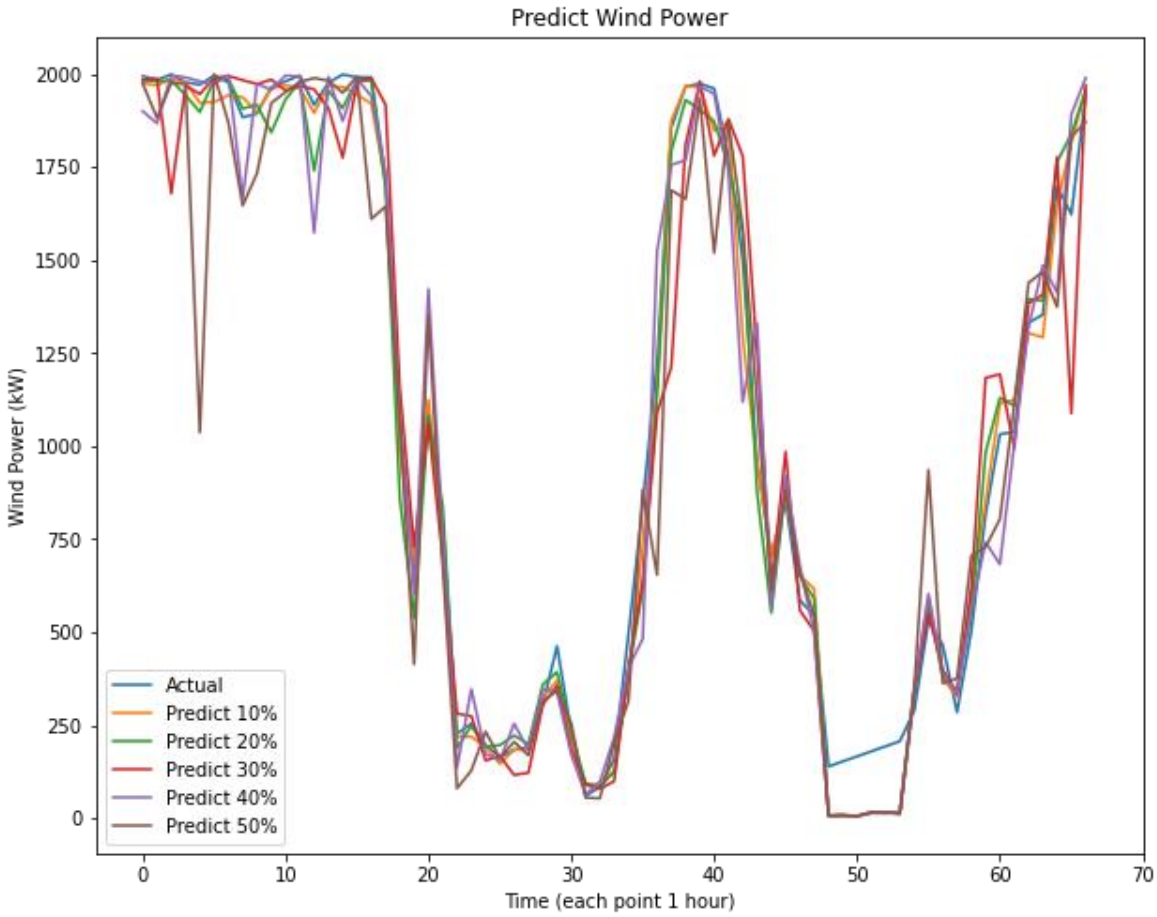


Figure 4-18 Final result for prediction

The graph shows the prediction for the next 3 – days prediction. The x - label presents the Time (1 tick is 1 hour) while the y - label gives the Wind Power Prediction (kW). The graph shows the difference between the actual value and the

predicted value of Wind Power is not huge. There are six lines indicates for six values of power predicted at each hour. The line with the legends “Predict” is the prediction value with added Normal distribution to the wind speed, respectively 10%, 20%, 30%, 40%, and 50%. The accuracy between the Real Value and the Predicted Value was calculated by the MRE as mentioned above.

CHAPTER 5. CONCLUSION

5.1 Conclusion

After running Machine Learning algorithms, learning Artificial Neural Network and evaluating the results after many test runs, this project has obtained favorable results in terms of clustering, classification, and forecasting models:

- The Fuzzy C - Means method uses the clustering method with the optimal number of groups of 5. Unlike k-means, where data points must only belong to one cluster center, here data points are awarded membership to each cluster center, allowing them to belong to more than one cluster center.This method is quite fast, and the number of groups is also quite reasonable in practice.
- Classification model using K - Nearest Neighbors algorithm achieves accuracy up to 91%, high efficiency, confirm the optimal K score (number of groups) by calculating the Euclidean value, get the K at 35.52 then choose k equal to 36. The KNN model is easy to understand and use as well as to classify which hours will belong to which group, contributing to the efficiency of the forecasting process.
- Adding Error using the Normal Distribution method is because the original data of Wind Speed is actual values, but I have the permission to predict the Wind Power. On the other hand, Wind Power prediction is often used to predict the height of the wind turbine shorter than 10m, but the reality wind turbine is higher. Therefore, the prediction problem will be more reliable because it is closer to reality.
- The FNN model is relatively easy to understand and use. Besides, the model brings high accuracy with the accuracy index between the forecast power, and the actual power that MRE brings is relatively small, 2.757% and 3.396%, with 10% and 20% Error. the results of the research open a new era for the wind power generations in Changhua (Taiwan) to plan its energy demand and supply in the future. Taiwan, as a developing country, can effectively use this

development to achieve its sustainable goals in renewable energy generation at a low cost. In addition, the results can effectively be used in similar wind speed areas in the country to showcase the feasibility of wind power potential and its economic worth.

5.2 Developments

Future work on this dataset that may contribute to better results for forecasting the wind power output of wind farms in Taiwan. Can be used the same data to predict others time – scales such as Ultra – short – term, Medium – term or Long – term. The project does not have enough time to use other forecasting methods, so there is no comparison between methods; besides using artificial neural networks, it is possible to consider using Some other forecasting methods such as Deep Neural networks, Long Short Term Memory, Convolutional Neural Network, etc.

The enormous body of literature has indicated that wind power forecasting research, development, and activity are highly active areas producing outcomes for producers, power system operators, and market operators. Wind power capacity has multiplied in the last 15 years, creating a requirement for breakthroughs in wind forecasting systems. Future developments in wind forecasting technology are likely to be driven by further increases in wind energy penetration of power systems, as well as the associated issues of managing wind variability, and current plans to massively increase offshore wind capacity will necessitate model improvements in this area.

REFERENCES

Wean – Yeau Chang. (2014). A Literature Review of Wind Forecasting Methods. *Journal of Power and Energy Engineering, 02,* 161 – 168. DOI: 10.4236/jpee.2014.24023

Khanh Dinh Tran, Nguyen Duc Vuong, Manh Kien Tran and Michael Fowler. (2020). "Fuzzy C – Means clustering Algorithm with Multiple Fuzzification Coefficients", ISSN: 1999 – 4893, *13*, 158, DOI: 10.3390/a13070158

Wikipedia (2022), *Wind power in Taiwan,* Retrieved June 07, 2022, from https://en.wikipedia.org/wiki/Wind_power_in_Taiwan

Vaishali Sohoni, S. C. Gupta, R. K. Nema. (2016). A Critical Review on Wind Turbine Power Curve Modelling Techniques and Their Applications in Wind Based Energy Systems, *Journal of Energy, 2016,* 18

Vũ, T. H. (2019). *Machine Learning cơ bản*. Việt Nam: Nhà xuất bản Khoa học và Kỹ thuật

Christopher Bourez's Blog (2016), *About loss functions, regularization and joint losses : multinomial logistic, cross entropy, square errors, euclidian, hinge, Crammer and Singer, one versus all, squared hinge, absolute value, infogain, L1 / L2 - Frobenius / L2,1 norms, connectionist temporal classification loss.* Retrieved June 12, 2022, from http://christopher5106.github.io/deep/learning/2016/09/16/about-loss-functions-multinomial-logistic-logarithm-cross-entropy-square-errors-euclidian-absolute-frobenius-hinge.html

Pomerat, John & Segev, Aviv & Datta, Rituparna. (2019). On Neural Network Activation Functions and Optimizers in Relation to Polynomial Regression. 6183-6185. DOI: 10.1109/BigData47090.2019.9005674.

Jim Frost. (2020), *Normal Distribution in Statistics.* Retrieved June 28, 2022, from https://statisticsbyjim.com/basics/normal-distribution/

Gonzalez, R. C., & Woods, R. E. (2018). Digital image processing. Pearson Global Edition

Altman, Naomi S. (1992). “An introduction to kernel and nearest – neighbors nonparametric regression”. *The American Statistician. 46 (3),* 175–185

Cunningham, Padraig & Delany, Sarah. (2007). K-Nearest Neighbors classifiers. Mult Classif Syst. *54*. DOI: 10.1145/3459665

Pritha Bhandari. (2020). Normal distribution | Examples, Formulas, & Uses. Retrieved July 02, from https://www.scribbr.com/statistics/normal-distribution/

Foley, Aoife & Leahy, Paul & Mckeogh, Eamon. (2010). “Wind power forecasting & prediction methods”. *2010 9th Conference on Environment and Electrical Engineering, EEEIC 2010*, 61 – 64, DOI: 10.1109/EEEIC.2010.5490016

APPENDIX PYTHON CODE

The code to execute the project can be downloaded at the link: https://drive.google.com/drive/u/2/folders/1PZMFdMpbudx9vzFe9JKIlKBIZVB_ZwYo , The files in this link need to be opened with software using the Python language.