MACHINE LEARNING-BASED SOLAR IRRADIANCE FORECASTING MODEL USING GPS

TANG SIN YEE

UTeM
اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

MACHINE LEARNING-BASED SOLAR IRRADIANCE FORECASTING MODEL USING GPS

TANG SIN YEE

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronic Engineering with Honours

Faculty of Electronics and Computer Technology and Engineering
Universiti Teknikal Malaysia Melaka

UTeM
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Machine Learning-based solar irradiance forecasting model using GPS

Sesi Pengajian : 2023/2024

Saya TANG SIN YEE mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐ **SULIT*** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☐ **TERHAD*** (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan.

☑ **TIDAK TERHAD**

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Disahkan oleh:

(TANG SIN YEE)

(TS. DR. HO YIH HWA)

Alamat Tetap:
Kampung Ketek
Buluh Semerak,
16700 Pasir Puteh,
Kelantan.

TS. DR. HO YIH HWA
Pensyarah Kanan
Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK)
Universiti Teknikal Malaysia Melaka (UTeM)

Tarikh : 12 Januari 2024

Tarikh : 12 Januari 2024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this report entitled “Machine Learning-based solar irradiance forecasting model using GPS” is the result of my own work except for quotes as cited in the references.

UTeM
اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Signature : ……………………………………

Author : TANG SIN YEE

Date : 12 JANUARY 2024

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

Signature : …………………………………

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Supervisor Name : TS. DR. HO YIH HWA

Date : 12 JANUARY 2024

DEDICATION

I dedicate this project report to my loving parents, whose unwavering support and encouragement have been instrumental in my academic journey. Your belief in my abilities and sacrifices you made have inspired me to strive for excellence. I would also like to express my deepest appreciation to my supervisor, Dr. Ho Yih Hwa, for his invaluable guidance, expertise, and continuous encouragement throughout this PSM project. Your insights and feedback have shaped my understanding and improved the quality of this work. Finally, I would like to thank my senior and friends for their constant encouragement and support. This project report is a tribute to everyone who has contributed, in various capacities, to my academic and personal development. Your support, whether significant or modest, has greatly influenced my journey. Thank you.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ABSTRACT

Integration of large-scale solar energy into an existing or future energy supply framework is becoming essential for the near future global energy supplies. This integration requires accurate forecasting of solar system output power, which is essential for the efficient functioning of the power grid and optimal utilization of energy fluxes within the solar system. In fact, this output power forecasting relies on solar irradiance forecasting. Accurate solar irradiance forecasting is essential for efficient integration of solar energy into the grid, as it enables grid operators and solar power plant operators to plan and manage energy production and consumption. Thus, this project proposed a Machine Learning-based solar irradiance forecasting model using GPS. Specifically, the model processed two types of input data, namely water vapor data (IWV) and total electron content (TEC) data obtained from RINEX GPS data. An Artificial Neural Network (ANN) machine learning algorithm has been used to process the input data and predict solar irradiance. Subsequently, the predicted results were displayed and validated using MATLAB GUI software. At the end of this project, the Bayesian Regularization algorithm with 10-training-layer size was identified as the best model among several algorithms that were tested, with a mean square error (MSE) of 20882.4233 and a correlation coefficient (R) of 0.86138.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ABSTRAK

Integrasi tenaga solar berskala besar ke dalam rangkaian bekalan tenaga sedia ada atau masa depan menjadi penting bagi bekalan tenaga global pada masa hadapan. Integrasi ini memerlukan ramalan tenaga sistem solar yang tepat, dimana ia penting bagi fungsi berkesan grid tenaga dan penggunaan optimum aliran tenaga dalam sistem solar. Ramalan radiasi solar yang tepat adalah penting bagi integrasi tenaga solar ke dalam grid, kerana ia membolehkan pengendali grid dan pengendali loji tenaga solar mengurus pengeluaran dan penggunaan tenaga. Oleh itu, projek ini mencadangkan model ramalan radiasi solar berdasarkan Pembelajaran Mesin menggunakan GPS. Secara khususnya, model ini akan memproses dua jenis data input, iaitu data wap air (IWV) dan data jumlah kandungan elektron (TEC) yang diperolehi daripada data GPS RINEX. Algoritma Pembelajaran Mesin Rangkaian Neural Tiruan (ANN) telah digunakan untuk memproses data input dan meramalkan radiasi solar. Kemudian, hasil ramalan dipaparkan dan disahkan menggunakan perisian MATLAB GUI. Pada akhir projek ini, algoritma "Bayesian Regularization" dengan saiz lapisan latihan 10 adalah model terbaik di antara beberapa algoritma yang telah diuji dengan ralat min kuasa dua, MSE sebanyak 20882.4233 dan pekali korelasi, R sebanyak 0.86138.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to many individuals who have helped me most throughout my final year project. First, I would like to thank my supervisor, Dr. Ho Yih Hwa, for his enthusiasm, patience, insightful comments, and advice that have helped me all the time throughout finishing my project. Without his guidance and help, this final year project would not have been possible.

I also wish to express my sincere thanks to all lecturers of Faculty of Electronic and Computer for their knowledge and expertise that they shared during my academic journey. Their teachings and mentorship have been instrumental in shaping my understanding of the subject matter. I am grateful to my friends for their collaboration and support during various stages of this project. Their input and discussions were invaluable in shaping my ideas and refining my work.

Lastly, I wish to thank my parents for their personal support. Their encouragement is my source of strength. This project would not have been possible without the collective support and contributions of all those mentioned above. I am truly grateful for their involvement, and I am indebted to their guidance and assistance.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

TABLE OF CONTENTS

Declaration

Approval

Dedication

Abstract i

Abstrak ii

Acknowledgements iii

Table of Contents iv

List of Figures vii

List of Tables x

List of Symbols and Abbreviations xi

List of Appendices xii

CHAPTER 1 INTRODUCTION 1

1.1 Background of project 1

1.2 Problem Statement 3

1.3 Objectives 3

1.4 Scope of Work 4

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

1.5 Importance of Study 4

1.6 Chapter Outline 5

CHAPTER 2 BACKGROUND STUDY 7

2.1 History of Solar Irradiance Forecasting 8

2.2 Machine Learning Algorithms 8

2.3 Evaluation of model accuracy 16

2.4 Single machine learning method 21

2.5 Total Electron Content 23

2.6 Integrated Water Vapor 26

2.7 Backpropagation Algorithm 28

CHAPTER 3 METHODOLOGY 30

3.1 Data Collection 32

3.2 Data Preprocessing 33

3.2.1 Total Electron Content (TEC) data preprocessing 33

3.2.2 Integrated Water Vapor (IWV) data preprocessing 40

3.2.3 Data Repository 49

3.3 Training and Testing suitable Machine Learning Model using Preprocessed Dataset 50

3.3.1 Input Preprocessed Dataset 51

3.3.2 Training the Preprocessed Dataset 52

3.3.3 Testing the Model 54

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

3.3.4 Validating the Model 55

CHAPTER 4 RESULTS AND DISCUSSION **56**

4.1 Training Model 57

4.2 Testing Model Plot 61

4.3 Comparison between Measurement and Predicted Solar Irradiance 63

4.4 Discussion on problems and solutions 65

CHAPTER 5 CONCLUSION AND FUTURE WORKS **66**

5.1 Conclusion 66

5.2 Future Work 68

REFERENCES **70**

APPENDICES **79**

UTeM
اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF FIGURES

Figure 2.1: SVM parameters. 10

Figure 2.2: Artificial Neural Networks (ANNs). 11

Figure 2.3: Random Forest algorithm. 12

Figure 2.4: Time series comparison. 16

Figure 2.5: Scatter plot. 17

Figure 2.6: ROC curve. 17

Figure 2.7: ANN, machine learning and SVM method in 5 main research of solar energy prediction. 22

Figure 2.8: Ionospheric Single layer Model [36]. 26

Figure 3.1: Project methodology flowchart. 31

Figure 3.2: Files in GPS_Gopi_v3.03 application folder [38]. 34

Figure 3.3: GPS_TEC.exe user interface. 34

Figure 3.4: RINEX 2.1 file format. 35

Figure 3.5: The batch processing and file selection options for the program. 36

Figure 3.6: Setting of batch processing and file selection options for the program. .37

Figure 3.7: Task processing completed. 37

Figure 3.8: Mean TEC output written to “TEC_output” folder. 38

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 3.9: Running checkstdfile.py in command prompt. 39

Figure 3.10: Run mean_TEC.py in CMD to merge all TEC into “ouput.csv” file....39

Figure 3.11: RTKLAUNCH and launcher icons in RTKLIB. 40

Figure 3.12: Main window of RTKPOST...41

Figure 3.13: RTKPOST Setting 1. 42

Figure 3.14: RTKPOST Setting 2. 42

Figure 3.15: RTKPOST Output setting...43

Figure 3.16: RTKPOST Stats setting. 43

Figure 3.17: RTKPOST Positions setting. 44

Figure 3.18: RTKPOST Files setting. 44

Figure 3.19: RTKPOST Misc setting..45

Figure 3.20: The completed pre-processing process in RTKPOST...........................45

Figure 3.21: Example of output from RTKPOST pre-processing. 46

Figure 3.22: Format of solution status file output of RTKPOST. 46

Figure 3.23: *checkztd.py* is executed in command prompt. 47

Figure 3.24: extract_ztd_sec.py is executed in command prompt. 48

Figure 3.25: extract_ztd_min.py is executed in command prompt. 48

Figure 3.26: Dataset folder in excel file...49

Figure 3.27: Input_new dataset in MATLAB workspace. 51

Figure 3.28: Target_new dataset in MATLAB workspace. 51

Figure 3.29: Neural Network start (nnstart) interface. 52

Figure 3.30: Three Neural Network Fitting training algorithm. 53

Figure 3.31: Training process is completed. 53

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 3.32: Testing solar irradiance model. 54

Figure 4.1: Training State Plot. 59

Figure 4.2: Mean Square Error (MSE) Plot. 59

Figure 4.3: Error Histogram Plot. 60

Figure 4.4: Regression Plot. 60

Figure 4.5: The model summary. 61

Figure 4.6: Additional testing Error Histogram Plot. 62

Figure 4.7: Regression Plot for additional test 1 month data (February). 62

Figure 4.8: Comparison between measurement and predicted solar irradiance plot for 1 day data. 64

Figure 4.9: Comparison between measurement and predicted solar irradiance plot for one month data (February). 64

اونيۏرسيتي تيکنيکل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF TABLES

Table 2.1: The use of machine learning algorithm method for solar irradiance forecasting in the past study. 14

Table 3.1: Dataset folder information. 50

Table 4.1: MSE and R results when training on different Algorithm and Layer size. 57

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF SYMBOLS AND ABBREVIATIONS

ANN	:	Artificial Neural Network
TEC	:	Total Electron Content
IWV	:	Integrated Water Vapor
GPS	:	Global Positioning System
AI	:	Artificial Intelligence
GBM	:	Gradient Boosting Machines
PCA	:	Principal Component Analysis
SVM	:	Support Vector Machines
SRM	:	Structural Risk Minimization
ROC	:	Receiver Operating Characteristic
MAPE	:	Mean Absolute Percentage Error
RMSE	:	Root-Mean-Square Error
MSE	:	Mean Square Error
GNSS	:	Global Navigation Satellite System
ZTD	:	Zenith Total Delay
ZHD	:	Zenith Hydrostatic Delay
RINEX	:	Receiver Independent Exchange Format

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF APPENDICES

Appendix A: rename_obs 79

Appendix B: rename_nav 81

Appendix C: checkstdfile.py 82

Appendix D: mean_TEC.py 83

Appendix E: checkztd.py 84

Appendix F: extract_ztd_sec.py 85

Appendix G: extract_ztd_min.py 86

Appendix H: mergedztd.py 87

Appendix I: BR10model.m 88

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 1

INTRODUCTION

UTeM
اونيۏرسيتي تيکنيکل مليسيا ملاک
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

1.1 Background of project

The demand for renewable energy sources, particularly solar power, has been steadily increasing as the world strives to reduce reliance on fossil fuels and decrease dependency on fossil resources. Solar irradiance, the amount of solar radiation reaching the Earth's surface, plays a crucial role in determining the potential energy generation from solar panels. Accurate forecasting of solar irradiance is vital for optimizing large-scale solar energy production and integration into the electrical grid. Traditional solar irradiance forecasting methods rely on weather data and physical models, which have limitations in accurately capturing the complex dynamics of solar radiation. In recent years, machine learning approaches have emerged as promising tools to improve the precision of solar irradiance predictions. By analyzing historical weather patterns and other relevant data, machine learning algorithms can identify complex patterns and relationships that traditional models might miss.

One key aspect that can significantly enhance the accuracy of solar irradiance forecasting is the integration of Global Positioning System (GPS) data. GPS provides precise information about the position and movement of cloud cover, which directly affects solar irradiance levels. By incorporating GPS data into the machine learning-based forecasting model, we can capture the spatial and temporal variability of clouds more effectively. The primary objective of this research project was to develop a machine learning-based solar irradiance forecasting model that leverages GPS data. By training the model on historical solar irradiance and GPS datasets, this study aimed to improve reliability of solar irradiance predictions. This would enable solar energy operators to make informed decisions regarding energy production, storage, and grid integration.

In this project, there is no hardware used. Firstly, an integrated water vapor (IWV) and Total Electron Content (TEC) served as input to the system. Six months of GPS data from the FTKEK station were utilized for IWV and TEC derivation, while six months of solar irradiance data from the FTKEK weather station were employed for the modeling. Then, Artificial Neural Networks (ANN) machine learning algorithm was trained to process the input data. The ANN is considered the most used method in global radiation forecasting [1]. Then, the solar irradiance model was evaluated. Finally, the measurement and predicted solar irradiance values were visualized in MATLAB GUI software.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

1.2 Problem Statement

Solar irradiance forecasting plays a critical role in the planning and integration of renewable energy sources into the electricity grid. It also helps energy managers and utilities forecast and plan their energy supply and demand more accurately. However, these lead to the development of high-accuracy solar irradiance forecasting model [2]. This is more challenging as solar activity is unpredictable and not steady. Therefore, it is crucial to be able to accurately estimate solar radiation, especially in cases of high energy integration [3]. The selection of appropriate input data from GPS data, developing, and evaluating suitable machine learning algorithms is very important to obtain the desired results for solar irradiance forecasting. This is because choosing the right data and preparing them properly enables the trained model to predict the results accurately. Solar irradiance forecasting using GPS data is more accurate than traditional forecasting without using GPS. Thus, this project aims to develop a machine learning-based solar irradiance forecasting model that has GPS data as input.

اونيۆرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

1.3 Objectives

1. To analyze atmospheric integrated water vapor (IWV) and ionospheric Total Electron Content (TEC) from GPS data.
2. To train machine learning model using preprocessed dataset.
3. To forecast solar irradiance with IWV and TEC calculated from GPS using ANN.

1.4 Scope of Work

1. Six months of GPS data from FTKEK station will be used for IWV and TEC derivation.
2. Six months of solar irradiance data from FTKEK weather station will be used for the modeling.
3. ANN will be used to predict solar irradiance by using water vapor and TEC as inputs.

1.5 Importance of Study

One of the importance of study is to enhance solar energy forecasting. By accurately predicting solar irradiance levels, solar power plants can adjust their operations, such as adjusting the tilt angles and orientations of solar panels, managing energy storage systems, and scheduling maintenance activities. This can result in higher solar energy yields and reduced dependence on other energy sources, contributing to sustainability by increasing the utilization of renewable energy and reducing greenhouse gas emissions.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Furthermore, solar power plants with accurate solar irradiance forecasts can reduce energy waste by optimizing energy production, avoiding overproduction in low irradiance periods, and reducing production in high irradiance periods. This minimizes energy waste and maximizes solar energy utilization, aligning with environmental friendliness by reducing unnecessary energy consumption.

Lastly, this project can reduce environmental impact. Solar energy is considered environmentally friendly as it is a clean and renewable source of energy. By using machine learning-based solar irradiance forecasting models, solar power plants can optimize their operations and reduce the need for backup power from fossil fuel-based sources, resulting in lower carbon emissions and reducing environmental impact. This contributes to sustainability by mitigating climate change and promoting environmentally friendly practices.

1.6 Chapter Outline

The problems and importance of Machine Learning-based Solar Irradiance Forecasting model using GPS have been described. All the details regarding this project have been outlined within each chapter of this report, as depicted below.

Chapter 1: In this section, a brief introduction to the project was provided. Additionally, explanations about the development of the machine learning-based solar irradiance system were presented to provide insight into the system. The chapter encompasses a thorough explanation of the problem statement, objectives, scopes, the significance of the study, and the project outline for the entire project.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Chapter 2: In this chapter, the previous work related to this project is discussed. A lot of research has been done to study the fundamental knowledge that is related to this project. The research is about previous work related to suitable machine learning algorithms, atmospheric integrated water vapor (IWV) calculation from GPS data, ionospheric Total Electron Content (TEC) calculation from GPS data etc.

Chapter 3: This chapter discusses the steps involved in completing the project. Several steps were applied in designing the machine learning-based solar irradiance forecasting model. This part provided a project flowchart, detailed methodology about how the project was done, training and testing the model, generating model coding, and plotting model precision.

Chapter 4: The results obtained from the final project are presented in this section. Following that, the section discusses the problems encountered during the completion of this project and outlines the solutions employed. The outcomes of the project are also addressed in this section, relying on the completed solar irradiance model.

Chapter 5: This chapter described the conclusion and recommendations for the machine learning-based solar irradiance forecasting model. The section included a project summary, project findings, and further recommendations to improve the project.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 2

BACKGROUND STUDY

This chapter will discuss research and articles related to the project. Numerous sources and studies have been conducted previously, providing details about this project that can be understood briefly. The theoretical background, literature review of previous work, and summaries of the preceding research will be covered in this chapter.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.1 History of Solar Irradiance Forecasting

Solar irradiance forecasting refers to the prediction of solar irradiance levels at a specific future location and time. Solar radiation is the radiant energy emitted from the sun. Solar irradiance is the amount of solar radiation received on a surface per unit area, typically measured in watts per square meter (W/m^2). Solar irradiance forecasting plays a crucial role in various applications related to solar energy, such as solar power plant operations, grid integration, energy management, and solar resource assessment. There are various methods for solar irradiance forecasting, which are physical, statistical, and machine learning models. Physical models are based on physical principles, such as the laws of thermodynamics and radiation transfer, and require input data such as humidity, cloud cover and temperature. Statistical models use historical data to identify patterns and make predictions, while machine learning models use algorithms to learn from historical data and make predictions based on that learning. In this project machine learning methods are used for solar irradiance forecasting.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.2 Machine Learning Algorithms

Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn and make predictions or forecasts without being explicitly programmed [4]. It involves the construction and study of systems that can learn from and adapt to data, improving their performance over time [5]. At its core, machine learning algorithms analyze and interpret patterns and relationships within datasets to generate insights, predictions, or decisions. These algorithms are designed to learn from experience and data by automatically identifying patterns, making statistical inferences, and adjusting their behavior accordingly.

Furthermore, one notable characteristic of machine learning models is their ability to find relationships between inputs and outputs, even in cases where the representation may seem impossible. This makes them useful in a wide range of applications such as pattern recognition, classification, spam filtering, data mining, and forecasting. In the domain of global horizontal irradiance forecasting, machine learning models can be employed in three different ways. Additionally, in this domain, classification and data mining tasks are particularly interesting as machine learning models can handle large datasets and assist with preprocessing and data preparation, followed by forecasting using the trained models which includes discriminant analysis and principal component analysis (PCA), Naive Bayes classification and Bayesian networks, and data mining approach [1]. For the training model, there are many machines learning algorithms that can be used such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), Random Forests, and Gradient Boosting Machines (GBM).

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

- **Support Vector Machines (SVM)**

Support Vector Machines (SVM) is a machine learning algorithm that follows the theory of inductive structural risk minimization (SRM). It aims to minimize both training errors and the trust level by considering a broader range of generalization errors compared to conventional neural networks [6]. SVM's solution is often efficient and avoids the problem of getting stuck in local minima. It relies on supportive vectors, a subset of training points, to find solutions. In addition, SVM are supervised learning models equipped with learning algorithms. It analyzes data and is widely used for non-linear problem resolution, regression, and classification in various fields [4]. The parameters of SVM are depicted in Figure 2.1 [6].

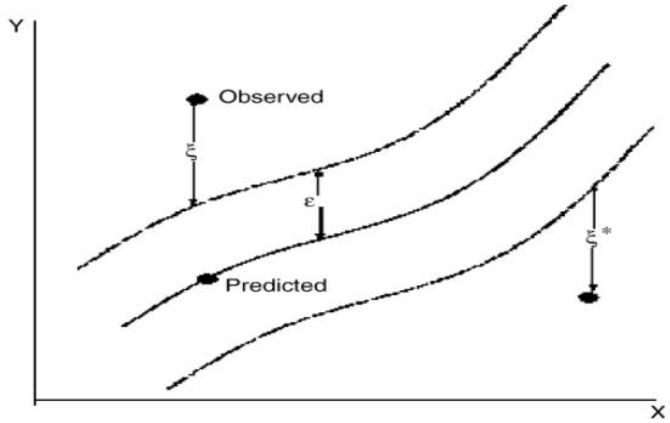


Figure 2.1: SVM parameters.

- Artificial Neural Networks (ANN)

Artificial Neural Networks (ANNs) are computational models inspired by the structure and functioning of the human brain [7]. They are a subset of machine learning algorithms that are designed to recognize complex patterns and relationships in data. ANNs consist of interconnected nodes, called artificial neurons or "nodes," that mimic the behavior of biological neurons. The basic building block of an artificial neuron is the perceptron, which takes multiple inputs, applies weights to each input, sums them up, and applies an activation function to produce an output. The activation function determines whether the neuron should "fire" or not, based on the weighted sum of inputs. Artificial Neural Networks have two learning types as unsupervised and supervised. In supervised learning, the Neural Network undergoes training based on patterns observed between input and output data. Throughout this process, the Network adjusts its weights iteratively to generate accurate results, halting when it

UTeM
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
اونيۏرسيتي تيكنيكل مليسيا ملاك

achieves the optimal output. Conversely, in unsupervised learning, only raw data is available without accompanying information or labels.

ANNs are typically organized in layers, including an input layer, hidden layers, and an output layer. Figure 2.2 below shows the integration between layers in the Artificial Neural Networks model. The input layer will receive the input data, and then process through the hidden layers. Each hidden layer consists of multiple interconnected neurons that perform computations and pass the results to the next layer. The output layer produces the result of the network's computation. During the training phase, ANNs learn to perform specific tasks by changing the weights and biases associated with the relationship between neurons. This adjustment is achieved through a process called backpropagation, whereby the network gradually enhances performance by comparing its output to the expected output and updating the weights accordingly.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

In recent years, ANNs have been effectively implemented to various fields, such as image and speech recognition, natural language processing, financial forecasting, solar irradiance forecasting and many others. They can learn and generalize from large amounts of data, enabling them to make predictions and forecasts based on patterns and relationships in the input data [7].

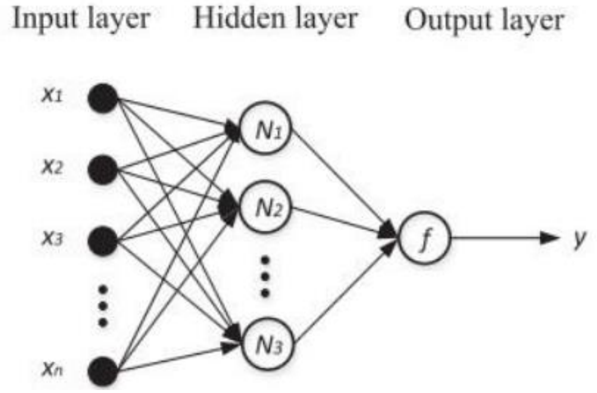


Figure 2.2: Artificial Neural Networks (ANNs).

- **Random Forests**

The Random Forest algorithm, a well-recognized method in machine learning, belongs to the ensemble learning category. Leo Breiman introduced Random Forests, drawing inspiration from earlier research conducted by Amit and Geman. [8] [9]. It is designed to tackle both classification and regression tasks [8] by combining the predictions of multiple decision trees to obtain a single output. In classification tasks, the random forest outputs the class that is most frequently chosen by the individual trees. In regression tasks, the collective average prediction made by the individual trees is provided as the output. Figure 2.3 below explains the working of the Random Forest algorithm.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

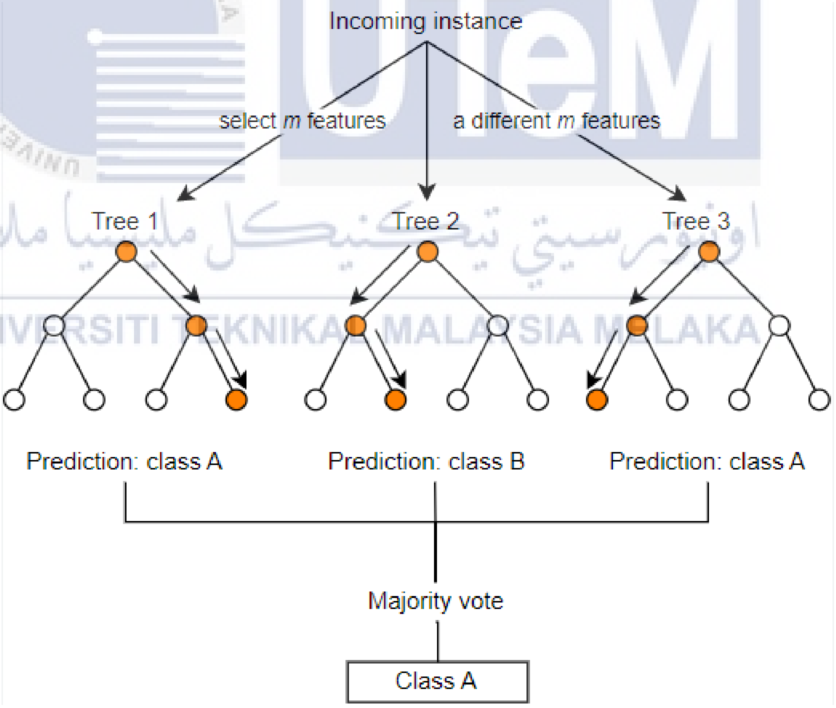


Figure 2.3: Random Forest algorithm.

The Random Forest algorithm works as follows:

Data sampling: Given a training dataset, Random Forest performs random sampling with replacement (known as bootstrapping) to create multiple subsets of the original data [6].

Building decision trees: For every subset of data, a decision tree is constructed. Decision trees consist of binary tree structures that iteratively divide the dataset according to various features and their respective values. Each decision tree is built independently, considering a random subset of features at each split, which helps introduce randomness and reduce overfitting.

Voting and prediction: After the decision trees are built, predictions are made by combining the results of each individual tree. In the case of classification tasks, the most common predicted class across all trees is chosen as the final prediction [10]. In regression tasks, the predicted values from all trees are averaged to derive the final output.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The advantage of Random Forest is its ability to reduce variance and handle high-dimensional datasets. By using multiple decision trees, it can capture complex relationships and provide robust predictions. Additionally, Random Forest can estimate the importance of each input feature, which can be useful for feature selection. Random Forest also possesses some other advantages, including reducing the overfitting of datasets. The combination of random sampling and feature selection in each decision tree helps mitigate overfitting, making Random Forest less prone to errors caused by noise or outliers in the data. Random Forest also can handle missing values in the dataset by utilizing the available features to make predictions. Lastly, this algorithm can efficiently handle large datasets and manage both categorical and numerical data without the need for substantial preprocessing.

- **Gradient Boosting Machines (GBM)**

Gradient Boosting Machine (GBM) is a powerful machine learning algorithm that combines weak models in a sequential manner to create a strong predictive model [11]. It uses gradient descent to minimize errors and iteratively improves the ensemble by correcting the mistakes made by previous models. GBM is widely used for regression and classification tasks and is known for its ability to handle complex relationships and noisy data. There are many machine learning algorithms used to forecast solar irradiance and assess the accuracy of the model, as shown in Table 2.1 below.

Table 2.1: The use of machine learning algorithm method for solar irradiance forecasting in the past study.

References /Year	Evaluation criteria	Input parameters	Output parameters	Data Scale	Accuracy/ Results
[12] 2018	SVM, ANN,	Minimum and maximum temperature, altitude, longitude, and latitude	Daily global solar irradiance	1966–2015	MAE, RMSE, R^2, and MSE
[13] 2018	SVM-R	Sunshine ratio	Daily global solar radiation	2005–2007	RMSE, rRMSE, and R^2
[14] 2019	WNN and ANN	cloud-cover(cc), relative humidity (H), temperature (T), day (D) and hour (h)	hourly Global Solar Radiation	2007-2010	nRMSE, R^2
[15] 2019	SP, ANN, and RF	Historical dataset	Solar radiation (Diffuse horizontal, beam normal, and global horizontal)	3 years of hourly data	RMSE, MAE,nRMSE, and nMAE
[16] 2019	ANN	ASHRAE Clear-Sky model and the local weather information	Daily solar irradiance	2019	MAE
[17] 2019	SVR, ANN, and DT	Hourly solar radiation	Hourly solar radiation	2012 to 2016	R^2 and RMSE
[18] 2020	SVR, ANN, random forest	Temperature, Humidity, sunshine duration and measured solar irradiance	Hourly basis solar irradiance	2017-2018	nRMSE, MAE

UTeM
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[19] 2020	ANN, and RNN	Soil, and air temperature sunshine duration, relative humidity, cloudiness, and extraterrestrial solar radiation	Daily global solar radiation	January 14, 2019, to January 21, 2019	RMSE, NMBE CV(RMSE), and R^2
[20] 2020	ANN	Relative humidity, precipitation, minimum and maximum temperature, altitude, longitude, months, latitude, sunshine duration, and wind speed	Global, direct, and diffuse solar radiation	2011–2016	R^2, MAPE, and RMSE
[21] 2020	ANN	air temperature, wind speed, precipitation, humidity, surface pressure, insolation clearness index, and earth skin temperature	Daily solar irradiance	2000-2015	MSE & R
[22] 2021	SVM, ANN, KNN, DL	daily minimum and maximum ambient temperature, cloud cover, daily extraterrestrial solar radiation, day length and solar radiation	daily global solar radiation	2018-2019	R^2, RMSE, rRMSE, MBE, MABE, t-stat & MAPE
[23] 2022	Multiple regression & ANN	Temp, RH, WS, Pressure, Time	June 2020	March to May 2020	R, RMSE
[24] 2022	ANFIS, ANN	time, temperature, relative humidity, wind speed, and wind direction	Hourly solar irradiance	2019 and 2020	mean absolute square error (MAPE), root mean square error (RMSE)
[25] 2022	ANN, CNN, RNN, SVR, PR RF	Wind speed, sun height, and ambient temperature	Global and diffuse solar radiation	2005–2016	R, MAE, RMSE, and NMBE
[26] 2022	Gradient Boost & bootstrap aggregation	hourly cloud cover, solar zenith angle, surface albedo and the global horizontal irradiance	Day ahead irradiance	2014	MSE, RMSE, MAPE, MAE
[27] 2022	ANN	daily temperature, humidity, pressure, wind speed, sunset time, and sunrise time	Daily solar irradiance	2017	R
[28] 2023	SVM, ANN, KNN, LSTM, Random Forest, gradiend boosting	DNI, Temperature, Wind speed, relative humidity, surface albedo, solar zenith angle	Daily solar irradiance	2017-2019	R^2

UTeM
اونيورسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.3 Evaluation of model accuracy

Evaluation is an important process that assesses the quality of the machine learning model, and it plays a significant role in various stages of model development. For example, it is important to evaluate the forecasting model during training, to measure the model's improvements after model modifications, and to compare different models. However, comparing model performance can be challenging due to factors like forecasted time horizons, time scale variations in predicted data, and variations in meteorological conditions across different sites. There are several graphical methods available to estimate how well the performance of the model, including:

1. Time series comparison: This tool allows for a visual assessment of the forecast quality by comparing predicted irradiance with measured irradiance over time. For example, Figure 2.4 demonstrates high accuracy in clear-sky situations and lower accuracy in partly cloudy situations.

اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

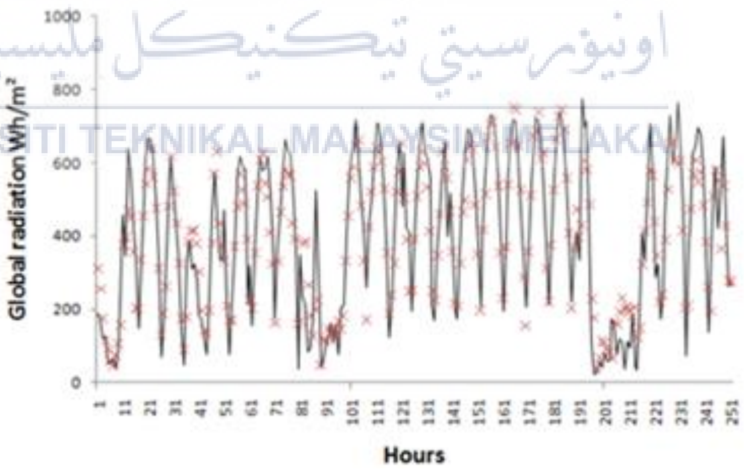


Figure 2.4: Time series comparison.

2. Scatter plots: These plots depict the relationship between predicted and measured irradiance, revealing systematic biases and deviations based on

irradiance conditions. They provide insights into the range of deviations associated with the forecasts. An example of a scatter plot is depicted as in Figure 2.5.

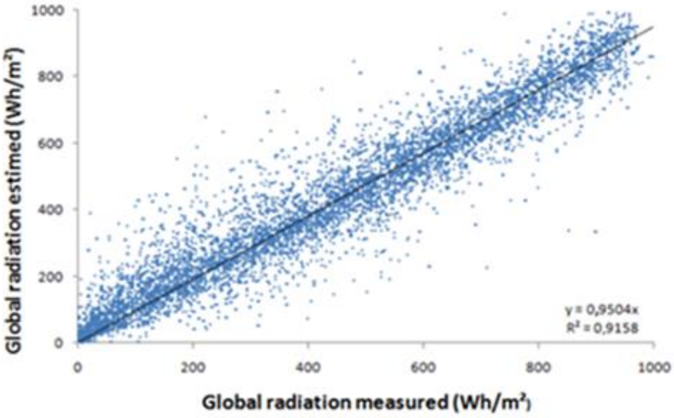


Figure 2.5: Scatter plot.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
UTeM
اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

3. Receiver Operating Characteristic (ROC) curves: ROC curves assess the percentages between true positives and false positives. Figure 2.6 shows an example of the ROC curve.

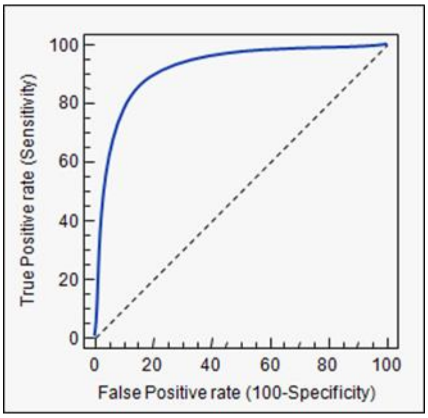


Figure 2.6: ROC curve.

Comparing forecasting methods becomes challenging due to the lack of accepted standard evaluation measures. However, Sperati et al. [29] conducted a benchmarking exercise within the framework of the European Actions Weather Intelligence for Renewable Energies (WIRE) to evaluate the performance of advanced models for short-term renewable energy forecasting. This study exemplifies the utilization of reliability parameters to assess forecasting accuracy. The authors emphasized the need for further research involving a broader range of test cases, data, and models to obtain a comprehensive understanding of different scenarios. They proposed considering test cases across various locations in Europe, the US, and other relevant countries to encompass diverse meteorological conditions. This paper effectively highlights the complexities involved in comparing the performance of forecasting methods.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

There were different performance matrices used to evaluate the effectiveness of the prediction models, including mean absolute percentage error (MAPE), root-mean-square error (RMSE), R-squared (R^2), mean absolute error (MAE), and mean square error (MSE). Mean Absolute Percentage Error (MAPE) measures the proportion of the mean absolute value of prediction errors to the mean absolute value of the actual data. A lower MAPE value indicates better performance of the model.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{p_i - q_i}{p_i}\right| \times 100 \qquad (2.1)$$

where y_i is the prediction, x_i is the actual value, N is the number of samples.

RMSE (Root Mean Square Error) is a widely used metric that quantifies the differences between predicted and observed values in a model. It is computed by taking the square root of the sum of squared differences between predicted and observed values within a specific sample. The root mean square error (RMSE) is particularly sensitive to significant forecast errors, making it suitable for situations where small errors are acceptable while larger errors incur significantly higher costs. This characteristic is especially valuable in utility applications. Therefore, RMSE is often considered the most important and commonly used reliability factor.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x}_i)^2}{N}} \qquad (2.2)$$

Where x_i is the actual value, $\bar{x}_i$is the predicted values, and N is the number of data points.

The **MAE (Mean Absolute Error)** measures the average magnitude of forecasting errors without considering their direction. The mean absolute error (MAE) is suitable for situations characterized by linear cost functions. In other words, it is applicable when the costs associated with a poor forecast are directly proportional to the magnitude of the forecast error.

$$MAE = \frac{\sum_{i=1}^{N}|y_i - x_i|}{N} \qquad (2.3)$$

اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The **Mean Squared Error (MSE)** stands as one of the fundamental and widely used loss functions. Computing the MSE involves subtracting your model's predictions from the ground truth, squaring it, and averaging it out across the whole dataset.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad (2.4)$$

Where N is the number of samples being tested against.

R^2 (Coefficient of Determination) is a statistical metric that indicates the percentages of variance for a dependent variable within a regression model that can be explained by one or more independent variables. This relationship is expressed through the following equation.

$$R^2 = 1 - \frac{\sum\left(y_{actual} - y_{predicted}\right)^2}{\sum(y_{actual} - \bar{y})^2} \qquad (2.5)$$

Where:

y_{actual}= actual value of dependent variable

$y_{predicted}$= predicted value of the dependent variable from the model

$\bar{y}$= mean of the actual value of the dependent variable

$\sum$ = summation (sum of all values)

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

R (Correlation Coefficient) measures the strength and direction of the linear relationship between two variables. It ranges from -1 to 1, where 1 signifies a perfect positive linear relationship, -1 indicates a perfect negative linear relationship, and 0 represents no linear relationship between the variables.

$$R = \frac{n\sum(xy) - \sum x \sum y}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y)^2)}} \qquad (2.6)$$

Where:

n = number of data points

x = values of the independent variable

y = values of the dependent variable

$\sum$ = summation (sum of all values)

xy = product of the independent variable and dependent variable values

2.4 Single machine learning method

Over the past few years, several researchers have conducted a comparative analysis of different machine learning algorithms. [30,31,32]. However, all these studies reveal that the ANN algorithm did not achieve most accurate prediction outcomes, yet it offered valuable insights for enhancing algorithm performance. The application of machine learning has gained prominence in developing solar radiation models and has become a popular research area. In the paper by [33], machine learning proved to be an effective investigative tool in several fields, including imagine recognition and natural language processing.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The paper [1] presents a graph that shows the number of ANN, machine learning and SVM term that have been used in the 5 main research of solar energy prediction. The graph is shown in Figure 2.7 below. As shown, the ANN is the method that was most frequently used to forecast radiation globally.

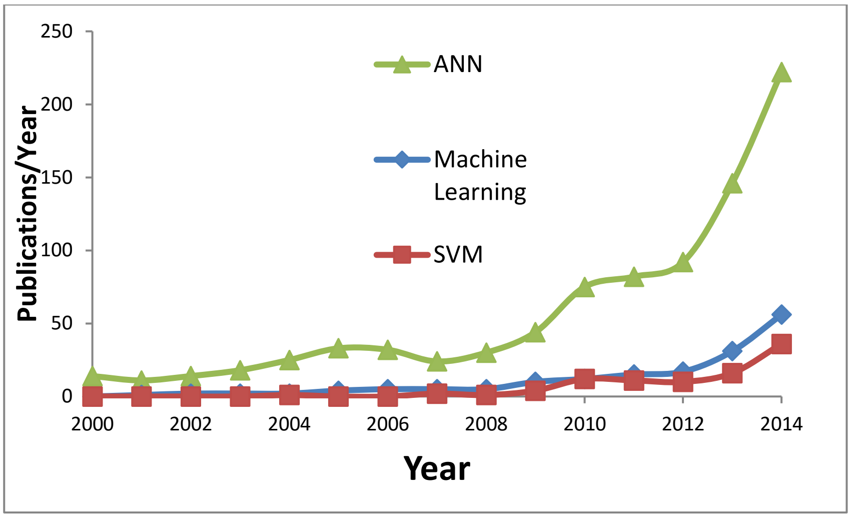


Figure 2.7: ANN, machine learning and SVM method in 5 main research of solar energy prediction.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Artificial Neural Networks (ANNs) are often used for solar irradiance prediction with GPS data due to several advantages they offer over other machine learning algorithms. One of the reasons why ANNs are commonly chosen is due to their non-linearity. Solar irradiance prediction involves complex relationships between various input variables, such as geographical coordinates, time of day, and weather conditions. ANNs excel at capturing non-linear patterns and relationships, allowing them to model the intricate interactions between these factors accurately. After that, ANNs are highly flexible and can handle different types of input data. In the case of solar irradiance forecasting, ANNs could accommodate the multidimensional nature of GPS data, incorporating both spatial and temporal information. ANNs could be scaled up to handle large and complex datasets. Solar irradiance prediction often involves dealing with a significant volume of data due to the temporal and spatial dimensions involved.

Furthermore, ANNs could be trained on large datasets efficiently using parallel processing or distributed computing, allowing them to handle the computational demands of solar irradiance prediction. While other machine learning algorithms, such as decision trees, support vector machines, or random forests, may also be applicable to solar irradiance prediction, ANNs are favored due to their ability to handle the complexity and non-linear nature of the problem, as well as their flexibility and scalability.

2.5 Total Electron Content

Total Electron Content (TEC) refers to the total number of free electrons present in a column of the Earth's ionosphere, typically measured along the path of radio signals transmitted from satellites to receivers on the ground [34]. The satellites of the GPS system, which orbit the Earth at approximately 20,200 km above its surface, emit signals that travel through the ionosphere spanning a range of around 60 to 1500 km above the Earth [35]. The oblique total electron content measurements can be obtained by observing the advance or delay of GPS signals on channels L1 (1575.42 MHz) and L2 (1227.6 MHz) [36]. The existence of electrons influences radio waves. The more electrons in the path of the radio wave, the more the radio signal will be affected. TEC quantifies the density of electrons in the ionosphere, representing the cumulative effect of the electron concentrations across the ionospheric layer. It is commonly expressed in electrons per square meter, 1 TEC unit (1 TECU = 10^16 electrons/m^2) [37]. Vertical TEC values in Earth’s ionosphere can range from a few to several hundred TECU.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

However, the TEC was influenced by factors such as local time, latitude, longitude, season, solar cycle and activity, geomagnetic conditions, and troposphere conditions. The ionosphere influences radio wave propagation. As a radio wave travels through the ionosphere's electrons, its velocity varies. The radio wave's frequency and the TEC between the transmitter and the receiver determine how much delay a radio wave experiences when travelling through the ionosphere. The ionosphere allows radio waves to travel through it at certain frequencies. At other frequencies, the waves are reflected by the ionosphere. In addition, the precision of satellite navigation systems, like GPS/GNSS, is greatly impacted by variations in the course and velocity of radio waves in the ionosphere. TEC data is crucial in various fields, particularly in satellite communications and global navigation systems (such as GPS), as variations in TEC can affect the propagation of radio signals, leading to inaccuracies in signal reception and navigation. Studying TEC variations aids in understanding ionospheric disturbances and space weather phenomena, contributing to improved modeling and prediction of ionospheric behavior [34].

TEC can be divided into two parts. Which is **Slant TEC (sTEC)** and **Vertical TEC (vTEC)**. As previously stated, Total Electron Content (TEC) can be derived by calculating the ionospheric delay between the L1 and L2 signals, as represented by Equation 2.7 [36]:

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

$$TEC = [9.483 \times (PR_{L2} - PR_{L1} - \Delta)] + CAL \tag{2.7}$$

Where:

PR_{L2}= L2 pseudo-range in meters

PR_{L1}=L1 pseudo-range in meters

Δ = Input bias between the C/A and P code chip transitions in meters

CAL = TEC result due to internal receiver L1/L2 delay and the offset

TEC stated in equation 2.7 (Slant TEC) is a measure of the total electron content of the ionosphere along the ray path from the satellite to the receiver and is measured at different elevation angles, represented in figure 2.8 [36]. Although sTEC is measured at differing elevation angles, usually the vTEC is modeled. The representation of Vertical Total Electron Content (vTEC) at elevation angle of 90° is depicted in Figure 2.8 and as per equation 2.8 below.

$$vTEC = sTEC(cos\chi') \qquad (2.8)$$

With

$$cos\chi' = \sqrt{1 - sin^2\chi'} \qquad (2.9)$$

$$sin\chi' = \frac{R_E}{R_E + h_m} sin\chi \qquad (2.10)$$

Where:

اونيۆرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

χ and χ' = Zenith angles at the receiver site and at the ionospheric pierce point, IPP

R_E = Mean earth radius

h_m = Height of maximum electron density (450km)

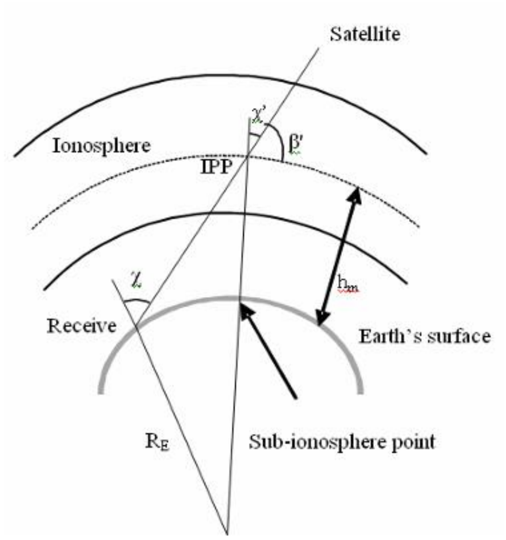


Figure 2.8: Ionospheric Single layer Model [36].

Based on the conducted research, Dr. Gopi Krishna Seemala developed a software application utilizing the formula to compute sTEC, vTEC and mean TEC [38]. Mean TEC or average Total Electron Content, often derived from multiple vertical Total Electron Content (vTEC).

2.6 Integrated Water Vapor

Integrated water vapor (IWV) is the total amount of precipitable water in an atmospheric column between the Earth's surface and the top of the atmosphere [39]. It is in units of kilogram per square meter (kg/m^2). The zenith total delay (ZTD) can be derived from GPS data processing and can be converted into integrated water vapor (IWV) [40].

$$IWV = \frac{10^6}{R_v \cdot [k'_2 + k_3/T_m]} \cdot (ZTD - ZHD) \tag{2.11}$$

Where:

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

R_v = gas constant for water vapor

k'_2 and k_3 = atmospheric refractivity constants [40]

T_m= weighted mean temperature

$$T_m = \frac{\int_{H_s}^{H_{top}} \frac{e}{T} d\rho}{\int_{H_s}^{top} \frac{e}{T^2 d\rho}} \qquad (2.12)$$

Where e and T are water vapor pressure and temperature profiles of the geopotential heights of the GPS station (H_s) to the top level of reanalysis (H_{top}), respectively.

As the GPS signals travel from the GPS satellites to the ground receivers, they experience a delay caused by the atmosphere. The total delay referred to as Zenith Total Delay (ZTD), is presented as an apparent additional distance rather than a time-based delay. Within this delay, a portion known as Zenith Hydrostatic Delay (ZHD) is attributed to dry gases like oxygen and nitrogen. The remaining part of the delay, known as the Zenith Wet Delay (ZWD), is due to water vapor. Equation 2.13 illustrates how the ZWD can be computed using the difference between ZTD and ZHD.

$$ZWD = ZTD - ZHD \qquad (2.13)$$

Where zenith hydrostatic delay (ZHD) can be calculated from the local pressure, with the ZHD formula represented by equation 2.14 below [41]:

$$ZHD = 2.2768 \frac{\rho_s}{1-2.66\times10^{-3}\cdot\cos(2\varphi_s)-2.8\times10^{-7}H_s} \qquad (2.14)$$

Where ρ_s is the pressure at the GPS station with a latitude of φ_s and a height of H_s.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

This delay can be mathematically represented to acquire zenith total delay (ZTD), which demonstrates a close proportionality to integrated water vapor (IWV). Therefore, the ZTD can be computed using relationships of temperature (Temp_out) and Bar (Pressure) in artificial neural network (ANN) processing in MATLAB. This is because ZTD is influenced by variations in temperature and pressure, and changes in these atmospheric parameters affect the refractive index of air. As temperature and pressure change, the density and refractivity of the atmosphere change accordingly. This alteration in refractivity leads to variations in the speed of propagation of electromagnetic waves, such as those used in GPS signals, causing delays in their travel through the atmosphere, thus affecting ZTD. A unique feature of ANN is that it can establish empirical relationships between independent and dependent variables and extract subtle information and complex knowledge from representative data sets. ANN networks can establish relationships between independent and dependent variables without presumptions regarding any specific mathematical depiction of the underlying phenomena. This is the reason ZTD is used in determining IWV by using ANN.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.7 Backpropagation Algorithm

Artificial Neural Networks (ANNs) employ multiple algorithms extensively utilized across various domains to tackle different problems, including classification, prediction, and machine learning tasks. One of these algorithms is backpropagation. Backpropagation keeps adjusting the weight values that are calculated from input-output mappings and reduces the error between the correct output value and the target value. It iteratively computes the values of weight using the gradient descent algorithm [42]. There are three backpropagation algorithms in MATLAB neural network fitting which:

1. Levenberg-Marquardt Algorithm

The Levenberg-Marquardt Algorithm is a widely used computational method for solving nonlinear optimization problems. It operates with loss functions represented as a sum of squared errors and computes the Jacobian matrix and gradient vector [43]. Utilizing these, it approximates the Hessian matrix and adjusts weights and biases iteratively to minimize errors. It exhibits characteristics of gradient descent and the Newton method based on a damping scalar parameter [44].

2. Bayesian Regularization

Bayesian Regularization transforms nonlinear regression into a statistical problem, enhancing neural network training's robustness. It updates weights and biases similarly to the Levenberg-Marquardt approach, aiming to minimize a combination of weights and squared errors. Training stops based on time, performance achievement, or reaching the maximum epochs [44].

اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

3. Scaled Conjugate Gradient

Scaled Conjugate Gradient (SCG) optimizes training by employing conjugate directions, avoiding time-consuming line searches [45]. It computes training directions without needing Hessian matrix inversion, accelerating convergence compared to gradient descent. SCG resets training directions to the negative gradient and updates neural network weights based on conjugate parameters. The method characteristics is in between Newton's method and gradient descent for efficient training [44].

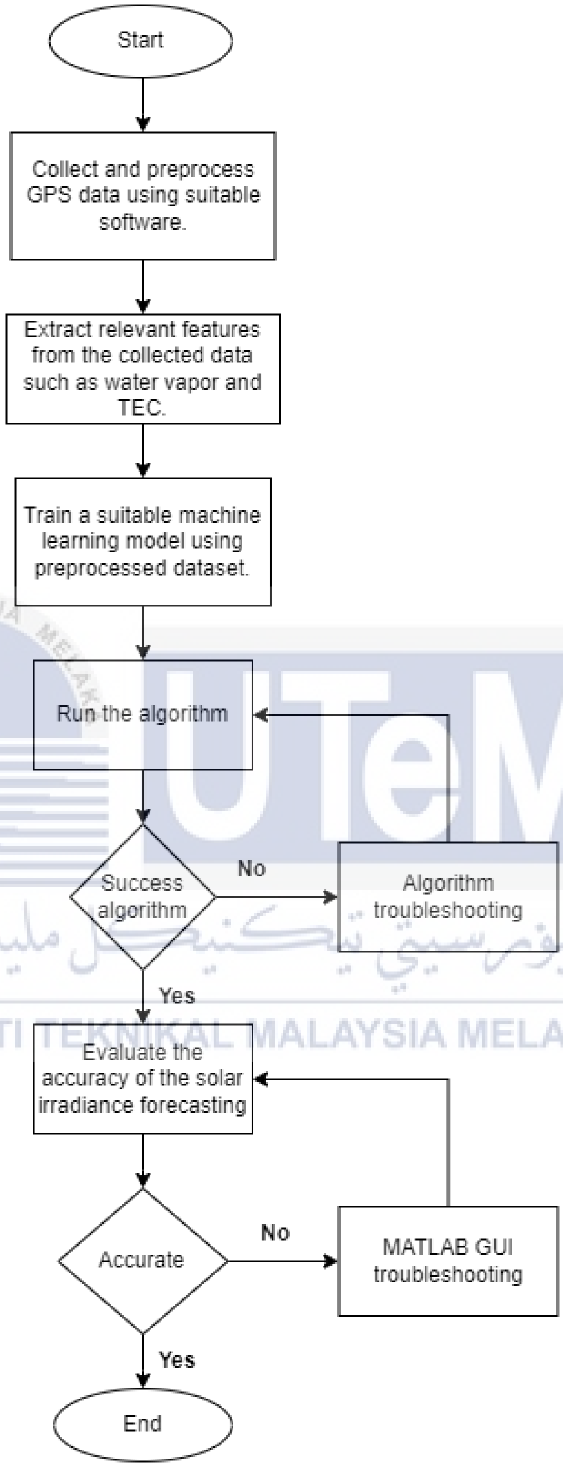
CHAPTER 3

METHODOLOGY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

In this chapter, we will discuss every process or step that was taken to control and deliver a project throughout the implementation process until the project was completed. From software installation to how to forecast solar irradiance. Besides, the function and meaning of the data pre-processing were also explained in detail in this part. This was explained part by part of the process for a better understanding. Furthermore, the machine learning model training was explained in more detail, such as steps, functions, advantages, and disadvantages of the process used to perform the model successfully. Lastly, the method in analyzing the model was also shown, including calculating model accuracy and calculating model error.

Flowchart


Start
Collect and preprocess GPS data using suitable software.
Extract relevant features from the collected data such as water vapor and TEC.
Train a suitable machine learning model using preprocessed dataset.
Run the algorithm
Success algorithm
No
Algorithm troubleshooting
Yes
Evaluate the accuracy of the solar irradiance forecasting
Accurate
No
MATLAB GUI troubleshooting
Yes
End

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 3.1: Project methodology flowchart.

The flowchart above shows steps required to complete the machine learning-based solar irradiance forecasting model using GPS project. The first step is to do research and study any information related to machine learning solar irradiance forecasting. The information obtained is gained from thesis, books, and websites. After conducting a background study, the project starts with software installation. The second step is collecting solar irradiance data and GPS data. After that, data pre-processing data was initiated to extract two relevant features from the collected data which are the integrated water vapor (IWV) and total electron content (TEC). If the data is not synchronous, data preprocessing needs to be redone for training and forecasting the output. For the training model, ANN algorithm is used to train solar irradiance forecasting. Then, by making IWV and TEC as the model input, the machine learning will produce an output which is a solar radiation forecast.

3.1 Data Collection

Two datasets needed for this project are GPS data and solar irradiance data. Existing GPS data sources are used by accessing publicly available GPS datasets from faculty. In this project, 12 months of historical GPS data from FTKEK station is collected and stored in a folder named "UTeM GPS Data 2022". The location is at (N 2.314100, E 102.318353) with the area of the GPS equal to 8860740.037km^2. The data taken is in 2022 and there are 362 days of data with 3 files in it. Those 3 data files are in Receiver Independent Exchange Format (RINEX) file format. Which is an open-source standard for raw satellite navigation system data. *n file is a navigation file, *g is a GLONASS navigation file and *m is a meteorological data file. RINEX files are ASCII based files that can be open with any text editor.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

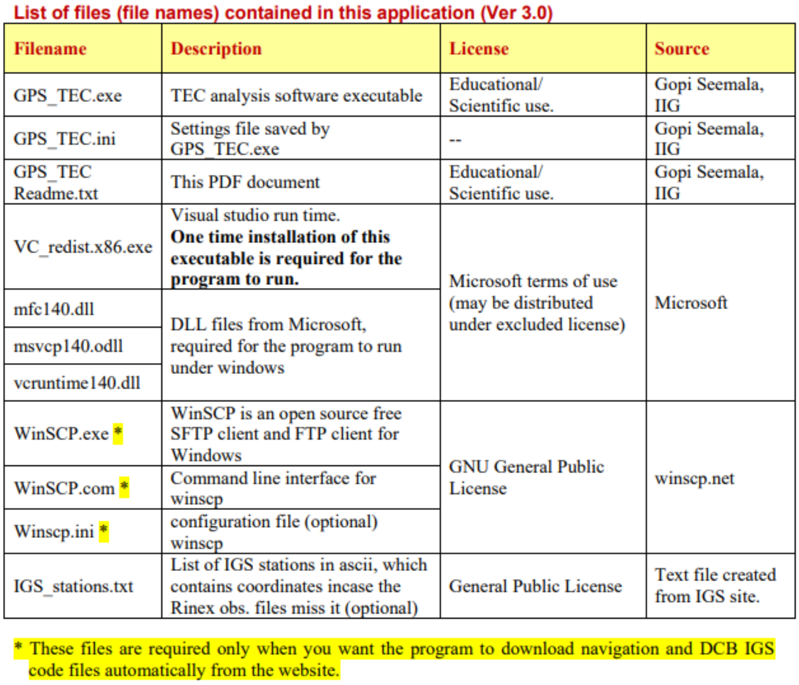
Besides, the solar irradiance data is retrieved from weather stations that record solar radiation data alongside other meteorological parameters. The weather data and solar irradiance are obtained from FTKEK weather station located in Melaka, Malaysia (N 2.314100, E 102.318353)). The file name for the dataset is “Year 2022”. There are 38 columns of data in the Year 2022 file. Which includes Date, Time, Temp Out, Hi Temp, Low Temp, Out Hum, Dew Pt., Wind speed, Wind Dir, Wind run, Hi Speed, Hi Dir, Wind Chill, Heat Index, THW Index, THSW Index, Bar, Rain, Rain Rate, Solar Rad., Solar Energy, Hi Solar Rad., UV Index, UV Dose, Hi UV, Heat D-D, Cool D-D, In Temp, In Hum, In Dew, In Heat, In EMC, In Air Density, ET, Wind Samp, Wind Tx, ISS Recept, and Arc. Int. But there are only 3 columns of data used in this project, which is Temp out, Bar/pressure, and Solar Rad.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

3.2 Data Preprocessing

3.2.1 Total Electron Content (TEC) data preprocessing

Total electron content is obtained from GPS-TEC application software developed by Dr Gopi Seemala. The application is a useful program that can be used to extract GPS-TEC data from the RINEX 2.1 and 3.02 observation files. GPS RINEX files used in this project is RINEX 2.11. Firstly, download the latest version of GPS TEC analysis (GPS_Gopi_v3.03 analysis application) from Dr Gopi Seemala blogspot (https://seemala.blogspot.com/)[38]. This application does not require any installation into program files or registry, just must unzip to use it. Figure 3.2 shows the list of files contained in the GPS_Gopi_v3.03 application folder.



List of files (file names) contained in this application (Ver 3.0)

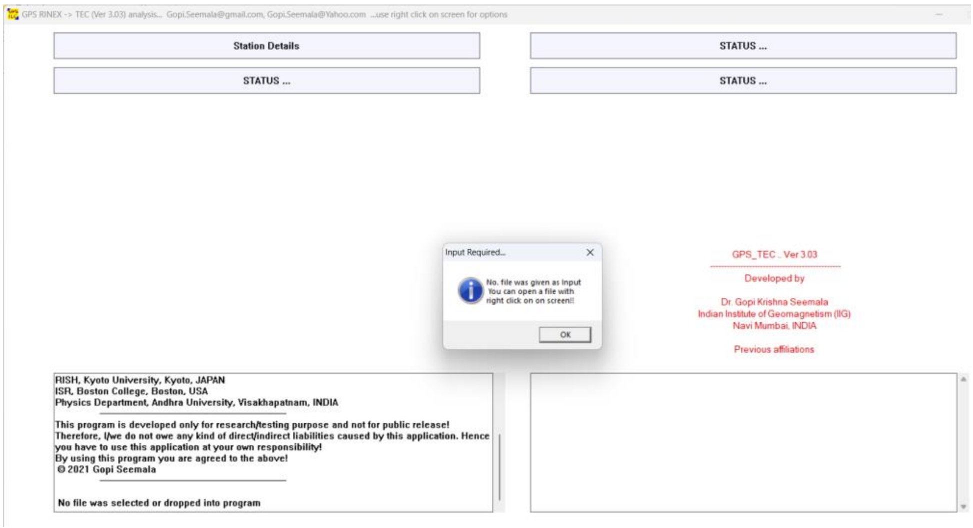
Filename	Description	License	Source
GPS_TEC.exe	TEC analysis software executable	Educational/ Scientific use.	Gopi Seemala, IIG
GPS_TEC.ini	Settings file saved by GPS_TEC.exe	--	Gopi Seemala, IIG
GPS_TEC Readme.txt	This PDF document	Educational/ Scientific use.	Gopi Seemala, IIG
VC_redist.x86.exe	Visual studio run time. **One time installation of this executable is required for the program to run.**	Microsoft terms of use (may be distributed under excluded license)	Microsoft
mfc140.dll	DLL files from Microsoft, required for the program to run under windows		
msvcp140.odll			
vcruntime140.dll			
WinSCP.exe *	WinSCP is an open source free SFTP client and FTP client for Windows	GNU General Public License	winscp.net
WinSCP.com *	Command line interface for winscp		
Winscp.ini *	configuration file (optional) winscp		
IGS_stations.txt	List of IGS stations in ascii, which contains coordinates incase the Rinex obs. files miss it (optional)	General Public License	Text file created from IGS site.

* These files are required only when you want the program to download navigation and DCB IGS code files automatically from the website.

Figure 3.2: Files in GPS_Gopi_v3.03 application folder [38].

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Next, install visual studio redistributable, VC_redist.x86 (32-bit version). This application file is already in the GPS_Gopi_v3.03 folder that was downloaded in the first step. After that, open the GPS_TEC.exe application. The user interface of the application is as shown in Figure 3.3 below.


GPS RINEX -> TEC (Ver 3.03) analysis... Gopi.Seemala@gmail.com, Gopi.Seemala@Yahoo.com ...use right click on screen for options
Station Details
STATUS ...
STATUS ...
STATUS ...
Input Required...
OK

No. file was given as Input
You can open a file with
right click on on screen!!

GPS_TEC .. Ver 3.03
Developed by
Dr. Gopi Krishna Seemala
Indian Institute of Geomagnetism (IIG)
Navi Mumbai, INDIA
Previous affiliations

RISH, Kyoto University, Kyoto, JAPAN
ISR, Boston College, Boston, USA
Physics Department, Andhra University, Visakhapatnam, INDIA

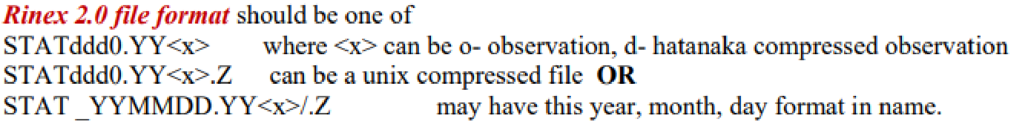
This program is developed only for research/testing purpose and not for public release!
Therefore, I/we do not owe any kind of direct/indirect liabilities caused by this application. Hence you have to use this application at your own responsibility!
By using this program you are agreed to the above!
© 2021 Gopi Seemala

No file was selected or dropped into program

Figure 3.3: GPS_TEC.exe user interface.

Then, the RINEX input file is entered into the GPS TEC analysis application. The RINEX file version for this project is RINEX 2.1. The input file or file required is RINEX navigation file and RINEX observation file for the observation date and differential code bias (DCBs) files. The RINEX navigation file from FTKEK station is to calculate the elevation and azimuth angles of the satellites for vertical TEC calculation. While differential code bias (DCBs) files are the systematic errors, inter-delay, or biases, between two GNSS code observations at the same or different frequencies. Nevertheless, for GNSS applications like determining total electron content (TEC) based on receiver observations [46,47,48], it is imperative to have knowledge of Differential Code Biases (DCBs). Neglecting DCBs can introduce errors of several meters in TEC estimations and may even lead to negative ionospheric delay values [49]. Furthermore, the RINEX 2.1 file format should be one of the three formats in Figure 3.4 below. Since the file in UTeM GPS dataset filename is not in these formats. Therefore, a python coding in Appendix A and Appendix B is used to change the file name according to the format STAT_YYMMDD.YY<x>. For example, **Trim202201010000C.22O** for 1 January 2022 observation file is changed to **Trim_220101.22O**.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

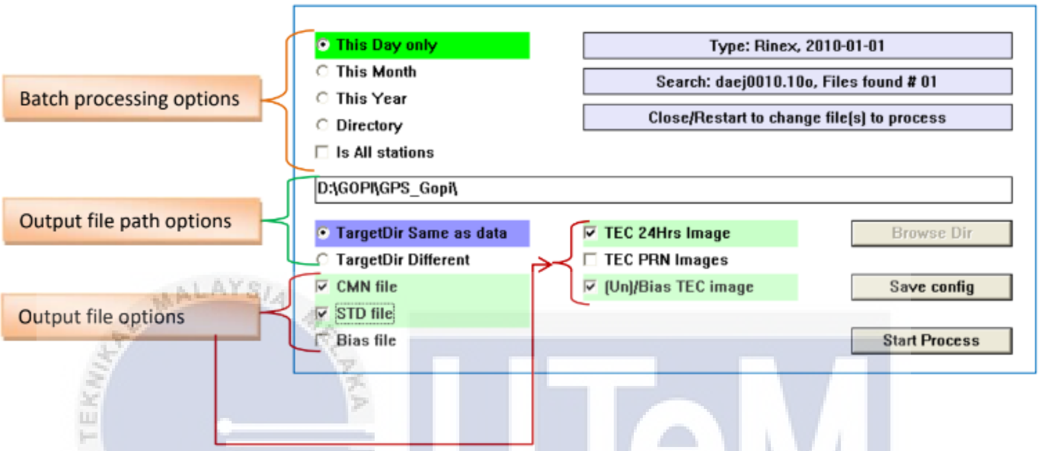


Rinex 2.0 file format should be one of
STATddd0.YY<x> where <x> can be o- observation, d- hatanaka compressed observation
STATddd0.YY<x>.Z can be a unix compressed file **OR**
STAT _YYMMDD.YY<x>/.Z may have this year, month, day format in name.

Figure 3.4: RINEX 2.1 file format.

While the differential code bias (DCB) files provided by the IGS code website (ftp://ftp.unibe.ch/aiub/CODE/) is for satellite biases. It contains two files which is

P1C1yymm.DCB.Z and P1P1yymm.DCB.Z files. Then place 12months of RINEX navigation, RINEX observation and DCB files in 1 folder named “TEC_input” and start batch processing option. To start the batch processing, right click on screen and add input file. Figure 3.5 show the screenshot that pop up after giving the input file of the program.



● This Day only
○ This Month
○ This Year
○ Directory
☐ Is All stations

Type: Rinex, 2010-01-01
Search: daej0010.10o, Files found # 01
Close/Restart to change file(s) to process

D:\GOPI\GPS_Gopi\

● TargetDir Same as data
○ TargetDir Different

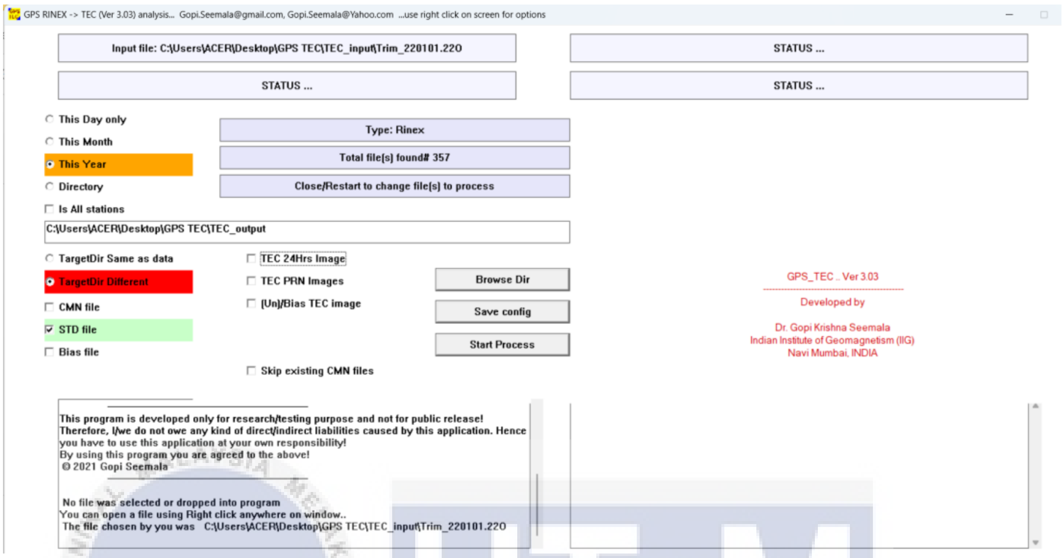
☑ CMN file
☑ STD file
☐ Bias file

☑ TEC 24Hrs Image
☐ TEC PRN Images
☑ (Un)/Bias TEC image

Figure 3.5: The batch processing and file selection options for the program.

اونيۆرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

From Figure 3.5 above, setting batch processing options to “This Year-this option will process the files of the entire year of the single station (of input file) or all the stations if “Is all stations” option is checked.”, set output file path options to “TargetDir Different” since want to set the target location to “TEC_output” folder. After that, set the output file options to “STD file” because this option will write the diurnal average TEC ascii file in the destination directory. The complete batch processing and file selection options for the program are as per Figure 3.6 below. Then, “Start Process” is clicked to start the batch processing. By pressing this button, the program will now start processing the given input(s) and the selected output files will be written to the output directory. The graph and task completion notes will be displayed according to Figure 3.7 once the processing is completed. The output of the dataset is in .Std file are written to the “TEC_output” directory as per Figure 3.8 below. For example, “Trim001-2022-01-01.Std” is for 1 January 2022 file name.


GPS RINEX -> TEC (Ver 3.03) analysis... Gopi.Seemala@gmail.com, Gopi.Seemala@Yahoo.com ...use right click on screen for options
Browse Dir
Save config
Start Process

Input file: C:\Users\ACER\Desktop\GPS TEC\TEC_input\Trim_220101.22O

STATUS ...

STATUS ...

STATUS ...

○ This Day only
○ This Month
● This Year
○ Directory
☐ Is All stations

Type: Rinex

Total file(s) found# 357

Close/Restart to change file(s) to process

C:\Users\ACER\Desktop\GPS TEC\TEC_output

○ TargetDir Same as data
● TargetDir Different
☐ CMN file
☑ STD file
☐ Bias file

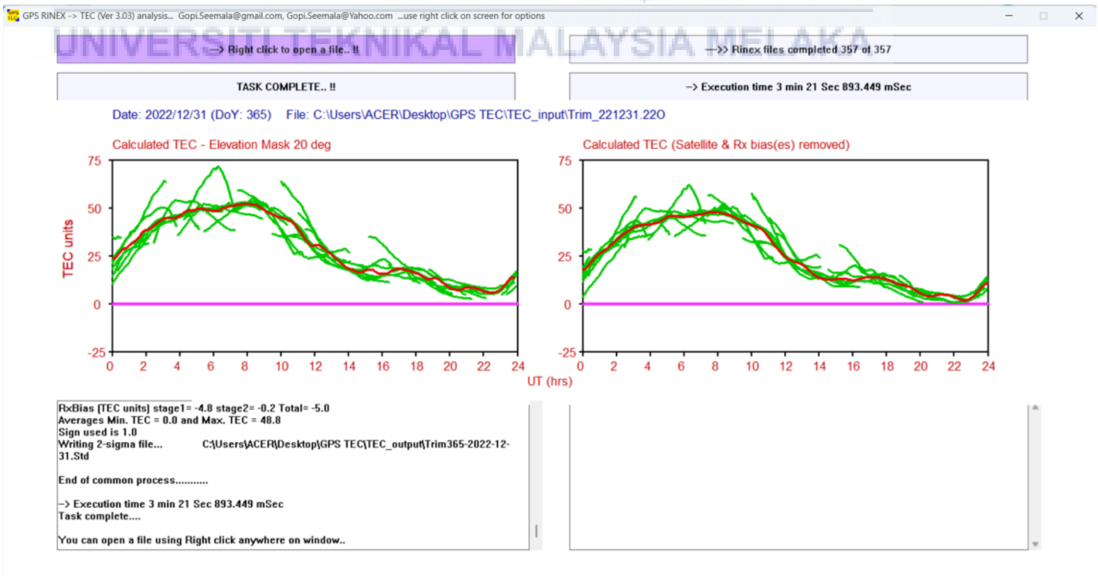
☐ TEC 24Hrs Image
☐ TEC PRN Images
☐ (Un)/Bias TEC image
☐ Skip existing CMN files

GPS_TEC .. Ver 3.03
Developed by
Dr. Gopi Krishna Seemala
Indian Institute of Geomagnetism (IIG)
Navi Mumbai, INDIA

```
This program is developed only for research/testing purpose and not for public release!
Therefore, I/we do not owe any kind of direct/indirect liabilities caused by this application. Hence
you have to use this application at your own responsibility!
By using this program you are agreed to the above!
 © 2021 Gopi Seemala

 No file was selected or dropped into program
You can open a file using Right click anywhere on window..
 The file chosen by you was   C:\Users\ACER\Desktop\GPS TEC\TEC_input\Trim_220101.22O
```

Figure 3.6: Setting of batch processing and file selection options for the program.

اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA


GPS RINEX -> TEC (Ver 3.03) analysis... Gopi.Seemala@gmail.com, Gopi.Seemala@Yahoo.com ...use right click on screen for options
Calculated TEC - Elevation Mask 20 deg
Calculated TEC (Satellite & Rx bias(es) removed)
TEC units
75
50
25
0
-25
0
2
4
6
8
10
12
14
16
18
20
22
24
UT (hrs)

—> Right click to open a file.. !!

—>> Rinex files completed 357 of 357

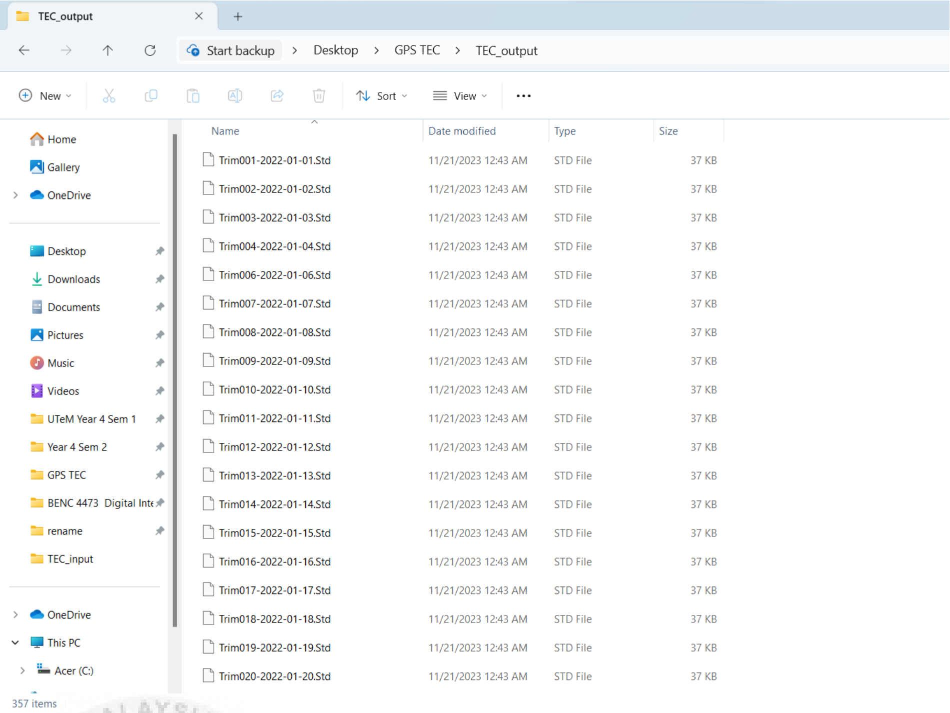
TASK COMPLETE.. !!

-> Execution time 3 min 21 Sec 893.449 mSec

Date: 2022/12/31 (DoY: 365)   File: C:\Users\ACER\Desktop\GPS TEC\TEC_input\Trim_221231.22O

```
RxBias (TEC units) stage1= -4.8 stage2= -0.2 Total= -5.0
Averages Min. TEC = 0.0 and Max. TEC = 48.8
Sign used is 1.0
Writing 2-sigma file...          C:\Users\ACER\Desktop\GPS TEC\TEC_output\Trim365-2022-12-
31.Std

End of common process...........

-> Execution time 3 min 21 Sec 893.449 mSec
Task complete....

You can open a file using Right click anywhere on window..
```

Figure 3.7: Task processing completed.



Name	Date modified	Type	Size
Trim001-2022-01-01.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim002-2022-01-02.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim003-2022-01-03.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim004-2022-01-04.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim006-2022-01-06.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim007-2022-01-07.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim008-2022-01-08.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim009-2022-01-09.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim010-2022-01-10.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim011-2022-01-11.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim012-2022-01-12.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim013-2022-01-13.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim014-2022-01-14.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim015-2022-01-15.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim016-2022-01-16.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim017-2022-01-17.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim018-2022-01-18.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim019-2022-01-19.Std	11/21/2023 12:43 AM	STD File	37 KB
Trim020-2022-01-20.Std	11/21/2023 12:43 AM	STD File	37 KB

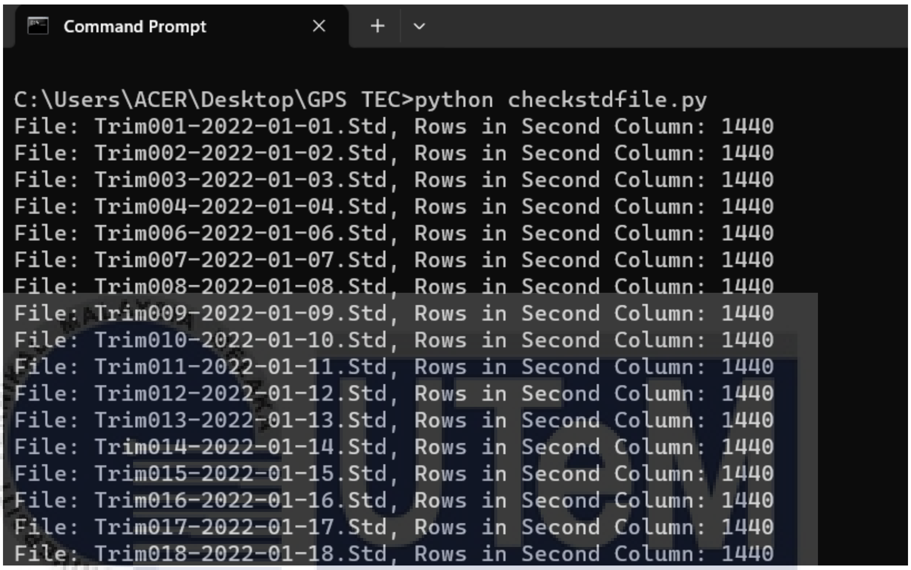
Figure 3.8: Mean TEC output written to "TEC_output" folder.

After complete preprocessing, the STD file that we obtained is in 4 columns. This ascii output file is in 4 columns separated by a tab, which is created in the same folder as data with same file name except the extension changes to ".std". The meaning of each column is:

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

- **Column 1:** Universal Time (time in UT (it is in decimals, means hrs + minutes/60 + Secs/3600, to convert back take integer as hours and multiply the fraction part with 60 & 3600 to get minutes & seconds respectively).
- **Column 2:** mean (2 sigma iterated) TEC, "-" (minus sign or hyphen) indicates no data.
- **Column 3:** standard deviation of TEC (at second iteration), "-" indicates no data.
- **Column 4:** Latitude of the station

Then, a python coding *checkstdfile.py* as in Appendix C is used to check there are how many rows of data in each .Std file. The code is run in command prompt as per Figure 3.9 below. The results show that there is total 357 file, 356 file got 1440 mean TEC data and 1 file (25 November 2022) got 911 mean TEC data. 1440 data means that there are a total of 1440 minutes data per day.

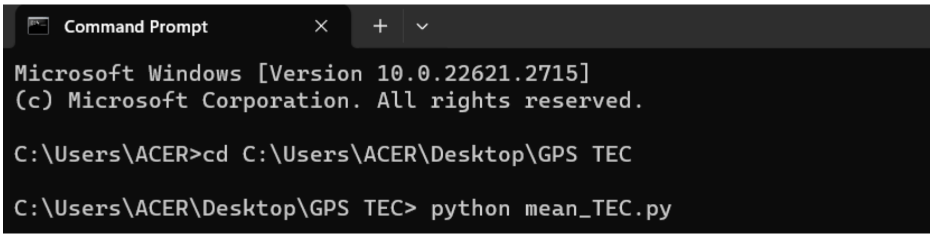


```
C:\Users\ACER\Desktop\GPS TEC>python checkstdfile.py
File: Trim001-2022-01-01.Std, Rows in Second Column: 1440
File: Trim002-2022-01-02.Std, Rows in Second Column: 1440
File: Trim003-2022-01-03.Std, Rows in Second Column: 1440
File: Trim004-2022-01-04.Std, Rows in Second Column: 1440
File: Trim006-2022-01-06.Std, Rows in Second Column: 1440
File: Trim007-2022-01-07.Std, Rows in Second Column: 1440
File: Trim008-2022-01-08.Std, Rows in Second Column: 1440
File: Trim009-2022-01-09.Std, Rows in Second Column: 1440
File: Trim010-2022-01-10.Std, Rows in Second Column: 1440
File: Trim011-2022-01-11.Std, Rows in Second Column: 1440
File: Trim012-2022-01-12.Std, Rows in Second Column: 1440
File: Trim013-2022-01-13.Std, Rows in Second Column: 1440
File: Trim014-2022-01-14.Std, Rows in Second Column: 1440
File: Trim015-2022-01-15.Std, Rows in Second Column: 1440
File: Trim016-2022-01-16.Std, Rows in Second Column: 1440
File: Trim017-2022-01-17.Std, Rows in Second Column: 1440
File: Trim018-2022-01-18.Std, Rows in Second Column: 1440
```

Figure 3.9: Running checkstdfile.py in command prompt.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

To make data processing easier, use python code *mean_TEC.py* as per Appendix D to take all second column data (mean TEC data) and put in 1 column data in excel file named “output.csv”. The coding is run in command prompt (CMD) as per Figure 3.10 below.



```
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ACER>cd C:\Users\ACER\Desktop\GPS TEC

C:\Users\ACER\Desktop\GPS TEC> python mean_TEC.py
```

Figure 3.10: Run mean_TEC.py in CMD to merge all TEC into “ouput.csv” file.

3.2.2 Integrated Water Vapor (IWV) data preprocessing

Based on the research conducted, it is not possible to directly derive Integrated Water Vapor (IWV) from preprocessed GPS data. However, through GPS data processing, it is feasible to determine zenith total delay (ZTD), which can then be used to calculate IWV. zenith total delay (ZTD) is obtained from the RTKLIB software. RTKLIB is an open-source program package for GNSS positioning. It enables standard and precise positioning algorithms accommodating GPS, GLONASS, Galileo, QZSS, BeiDou, and SBAS. Various positioning modes are supported, both for real-time and post-processing GNSS data: Single, DGPS/DGNSS, Kinematic, Static, Moving-Baseline, Fixed, PPP-Kinematic, PPP-Static, and PPP-Fixed. It also supports many standard formats and protocols for GNSS and provides an array of library functions and APIs dedicated to GNSS data processing. Firstly, download the latest version of the RTKLIB_2.4.2 package from (https://www.rtklib.com/rtklib.htm) and install it according to the RTKLIB manual in this link (http://www.rtklib.com/prog/manual_2.4.2.pdf). This software does not require any installation into program files or registry, just must unzip to use it. Figure 3.2 shows the list of files contained in the RTKLIB_2.4.2 software folder.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

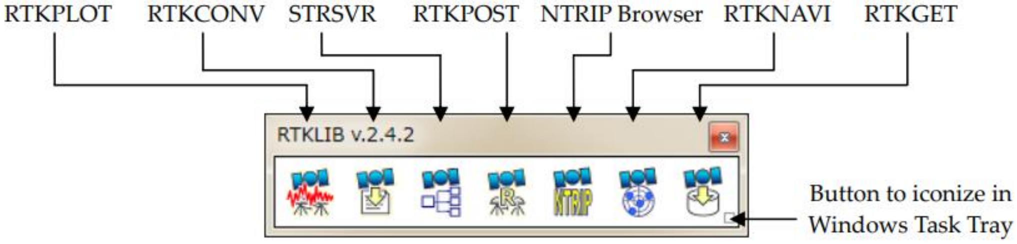
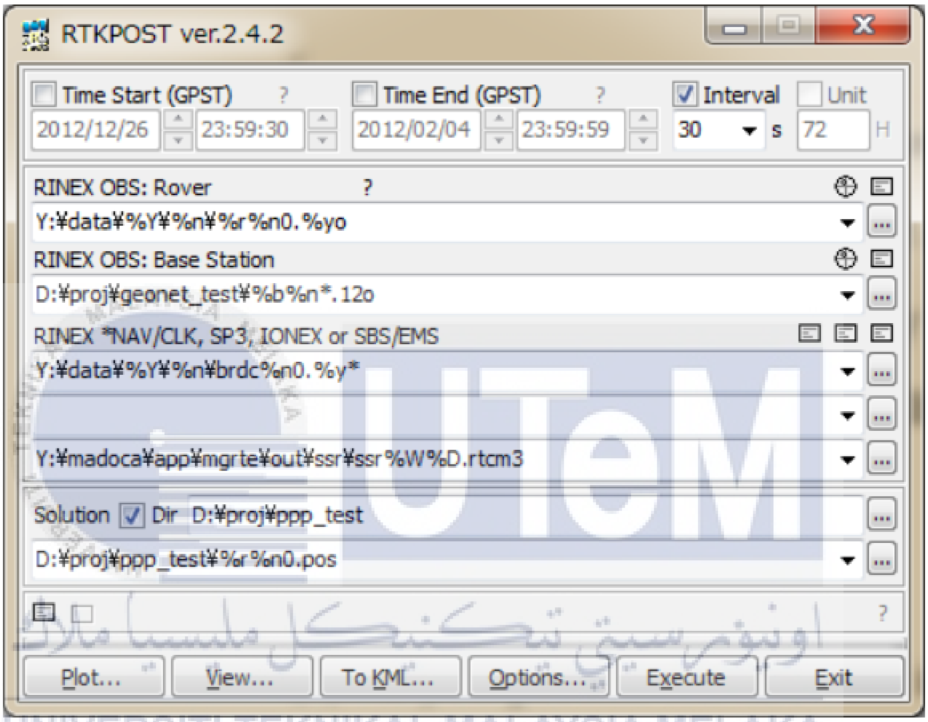


Figure 3.11: RTKLAUNCH and launcher icons in RTKLIB.

The application used to preprocess data related to Integrated Vapor (IWV) is a post-processing analysis tool called RTKPOST. RTKPOST processes standard RINEX observation data (versions 2.10, 2.11, 2.12, 3.00, 3.01, 3.02-draft) and navigation message files from GPS, GLONASS, Galileo, QZSS, BeiDou, and SBAS. The second step is to open the RTKPOST software. Figure 3.12 is the main window of RTKPOST.



☐ Time Start (GPST) ? 2012/12/26 23:59:30

☐ Time End (GPST) ? 2012/02/04 23:59:59

☑ Interval 30 s ☐ Unit 72 H

RINEX OBS: Rover ? Y:¥data¥%Y¥%n¥%r%n0.%yo

RINEX OBS: Base Station D:¥proj¥geonet_test¥%b%n*.12o

RINEX *NAV/CLK, SP3, IONEX or SBS/EMS Y:¥data¥%Y¥%n¥brdc%n0.%y*

Y:¥madoca¥app¥mgrte¥out¥ssr¥ssr%W%D.rtcm3

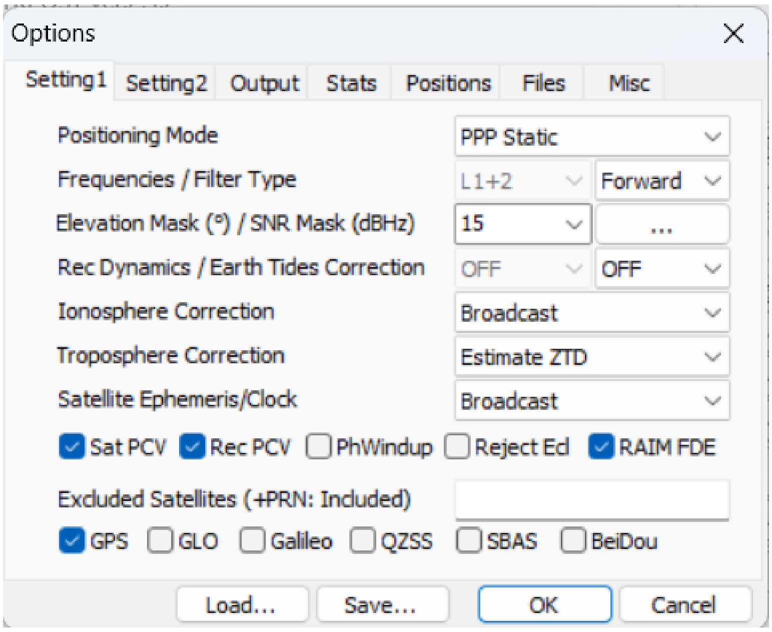
Solution ☑ Dir D:¥proj¥ppp_test

D:¥proj¥ppp_test¥%r%n0.pos

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 3.12: Main window of RTKPOST.

Click “options” button to set the processing options. For setting 1, choose Positioning Mode as PPP Static, Ionosphere correction as Broadcast, troposphere Correction as Estimate ZTD and Satellite Ephemeris/Clock as Broadcast. Setting 1 is set as per Figure 3.12 below.

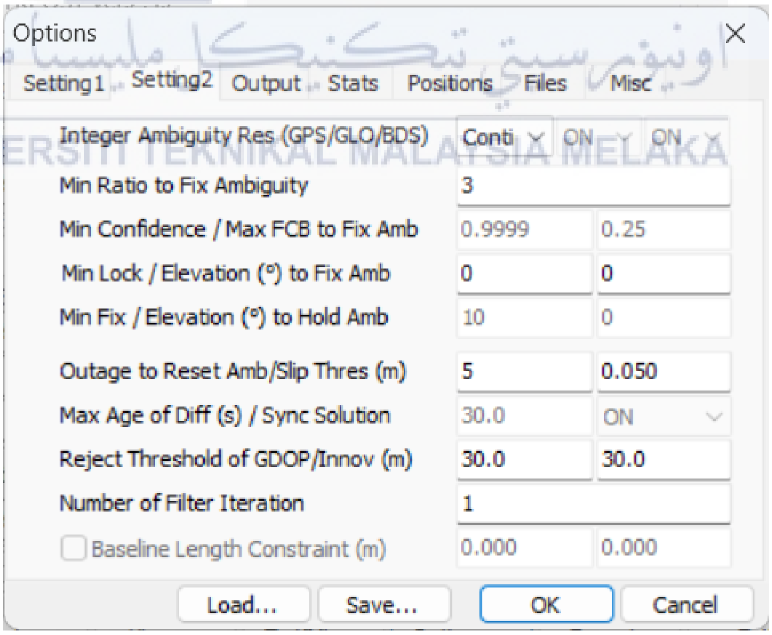


Positioning Mode: PPP Static
Frequencies / Filter Type: L1+2 / Forward
Elevation Mask (°) / SNR Mask (dBHz): 15 / ...
Rec Dynamics / Earth Tides Correction: OFF / OFF
Ionosphere Correction: Broadcast
Troposphere Correction: Estimate ZTD
Satellite Ephemeris/Clock: Broadcast
☑ Sat PCV ☑ Rec PCV ☐ PhWindup ☐ Reject Ed ☑ RAIM FDE
Excluded Satellites (+PRN: Included):
☑ GPS ☐ GLO ☐ Galileo ☐ QZSS ☐ SBAS ☐ BeiDou

Figure 3.13: RTKPOST Setting 1.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

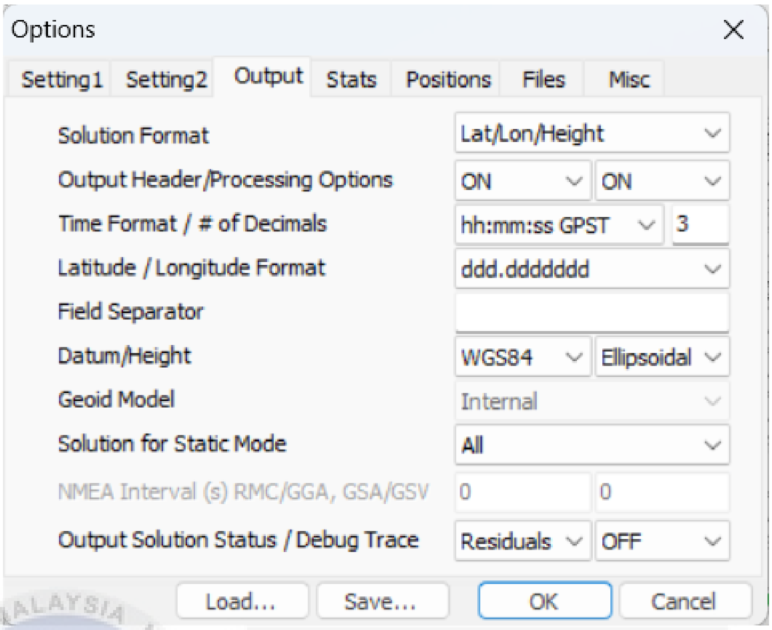
For Setting 2, set the processing as per Figure 3.14 below.



Integer Ambiguity Res (GPS/GLO/BDS): Conti / ON / ON
Min Ratio to Fix Ambiguity: 3
Min Confidence / Max FCB to Fix Amb: 0.9999 / 0.25
Min Lock / Elevation (°) to Fix Amb: 0 / 0
Min Fix / Elevation (°) to Hold Amb: 10 / 0
Outage to Reset Amb/Slip Thres (m): 5 / 0.050
Max Age of Diff (s) / Sync Solution: 30.0 / ON
Reject Threshold of GDOP/Innov (m): 30.0 / 30.0
Number of Filter Iteration: 1
☐ Baseline Length Constraint (m): 0.000 / 0.000

Figure 3.14: RTKPOST Setting 2.

For Output setting, set the processing as per Figure 3.15 below.

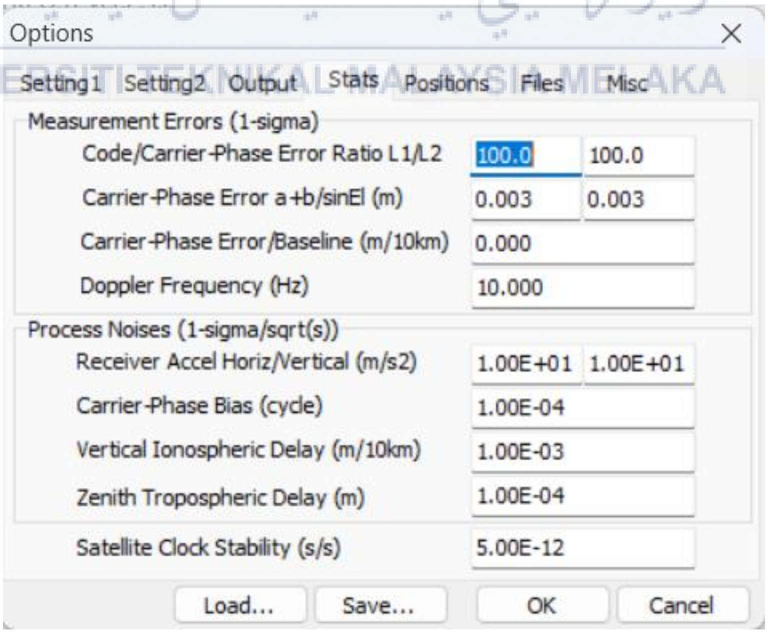


Solution Format	Lat/Lon/Height	
Output Header/Processing Options	ON	ON
Time Format / # of Decimals	hh:mm:ss GPST	3
Latitude / Longitude Format	ddd.ddddddd	
Field Separator		
Datum/Height	WGS84	Ellipsoidal
Geoid Model	Internal	
Solution for Static Mode	All	
NMEA Interval (s) RMC/GGA, GSA/GSV	0	0
Output Solution Status / Debug Trace	Residuals	OFF

Figure 3.15: RTKPOST Output setting.

For Stats setting, set the processing as per Figure 3.16 below.

اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA



Measurement Errors (1-sigma)

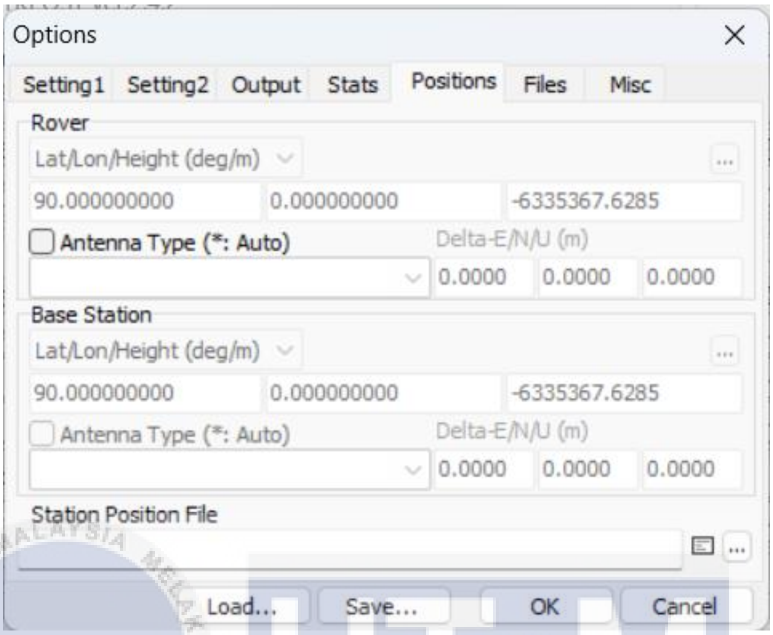
Code/Carrier-Phase Error Ratio L1/L2	100.0	100.0
Carrier-Phase Error a+b/sinEl (m)	0.003	0.003
Carrier-Phase Error/Baseline (m/10km)	0.000	
Doppler Frequency (Hz)	10.000	

Process Noises (1-sigma/sqrt(s))

Receiver Accel Horiz/Vertical (m/s2)	1.00E+01	1.00E+01
Carrier-Phase Bias (cycle)	1.00E-04	
Vertical Ionospheric Delay (m/10km)	1.00E-03	
Zenith Tropospheric Delay (m)	1.00E-04	
Satellite Clock Stability (s/s)	5.00E-12	

Figure 3.16: RTKPOST Stats setting.

For Positions, Files, and Misc setting, set the processing as default as per Figure 3.17, 3.18 and 3.19 below.



Rover
Lat/Lon/Height (deg/m)
90.000000000 0.000000000 -6335367.6285
☐ Antenna Type (*: Auto)
Delta-E/N/U (m)
0.0000 0.0000 0.0000

Base Station
Lat/Lon/Height (deg/m)
90.000000000 0.000000000 -6335367.6285
☐ Antenna Type (*: Auto)
Delta-E/N/U (m)
0.0000 0.0000 0.0000

Station Position File

Figure 3.17: RTKPOST Positions setting.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
اونيۏرسيتي تيکنيکل مليسيا ملاك
UTeM

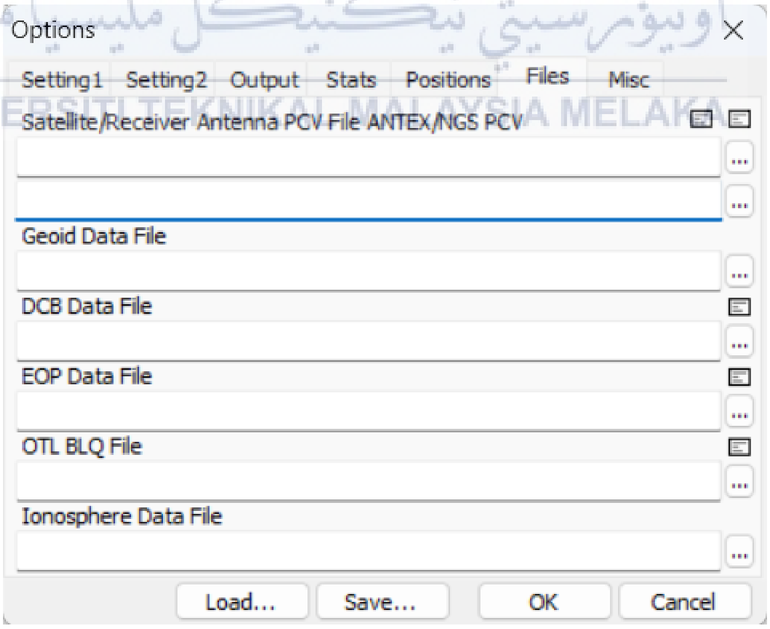
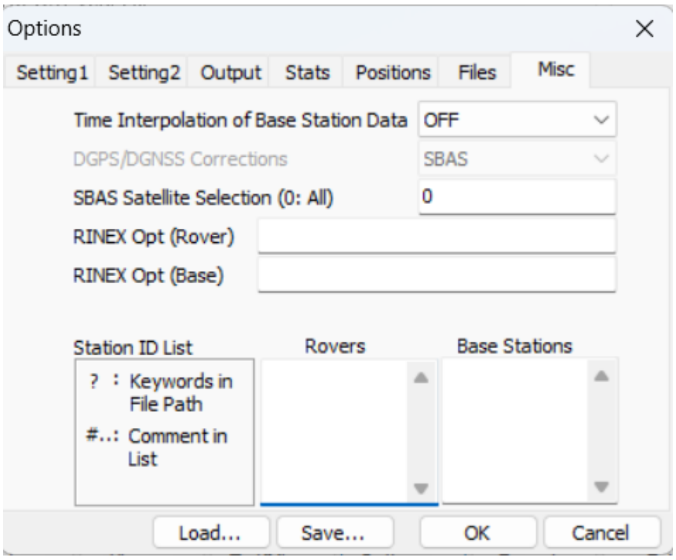


Figure 3.18: RTKPOST Files setting.



Options

Time Interpolation of Base Station Data: OFF

DGPS/DGNSS Corrections: SBAS

SBAS Satellite Selection (0: All): 0

RINEX Opt (Rover):

RINEX Opt (Base):

Station ID List

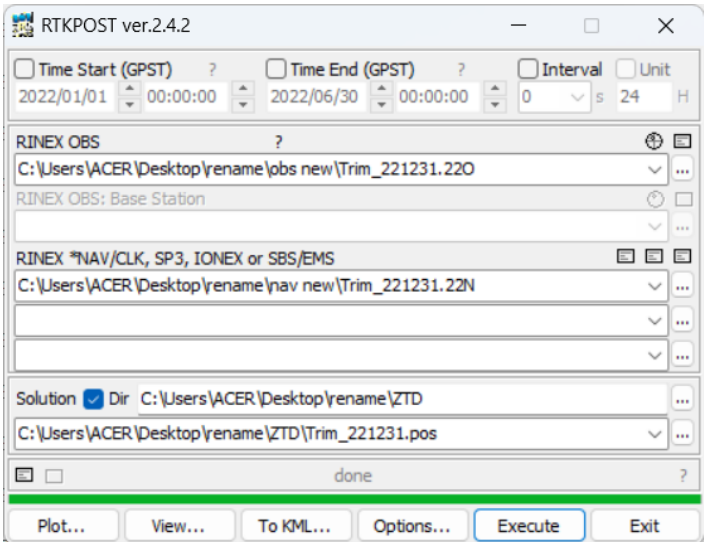
? : Keywords in File Path

#..: Comment in List

Figure 3.19: RTKPOST Misc setting.

After finishing setting the process, click OK to save the setting. The third step is input the RINEX observation data file in the text field (RINEX OBS: Rover) and the RINEX navigation data file in the text field (RINEX OBS: Base Station), then choose the desired output directory and click "Execute" button to start preprocess the data. The processing status is shown in the status message field lower center in the main window. When ″done″ message is shown, the analysis is completed as shown in Figure 3.20.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA



RTKPOST ver.2.4.2

☐ Time Start (GPST) ? 2022/01/01 00:00:00

☐ Time End (GPST) ? 2022/06/30 00:00:00

☐ Interval 0 s

☐ Unit 24 H

RINEX OBS ?

C:\Users\ACER\Desktop\rename\obs new\Trim_221231.22O

RINEX OBS: Base Station

RINEX *NAV/CLK, SP3, IONEX or SBS/EMS

C:\Users\ACER\Desktop\rename\nav new\Trim_221231.22N

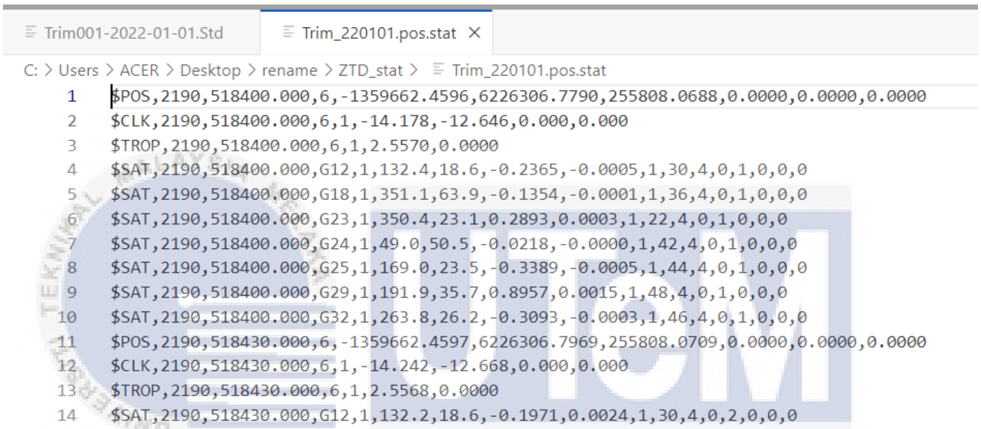
Solution ☑ Dir C:\Users\ACER\Desktop\rename\ZTD

C:\Users\ACER\Desktop\rename\ZTD\Trim_221231.pos

done

Figure 3.20: The completed pre-processing process in RTKPOST.

The dataset needs to be preprocessed day by day. If there is 1 year dataset, the preprocessing needs to be done 365 times. Although the manual suggests that data can be processed in batches, attempting to do so does not work as expected. Instead, the data can only be processed daily, not in the intended batches as described in the manual. The output of the zenith total delay (ZTD) for 1 January 2022 from RTKPOST is as per Figure 3.21 below. For example, "Trim_220101.pos.stat" is the file name for 1 January 2022 ZTD file.

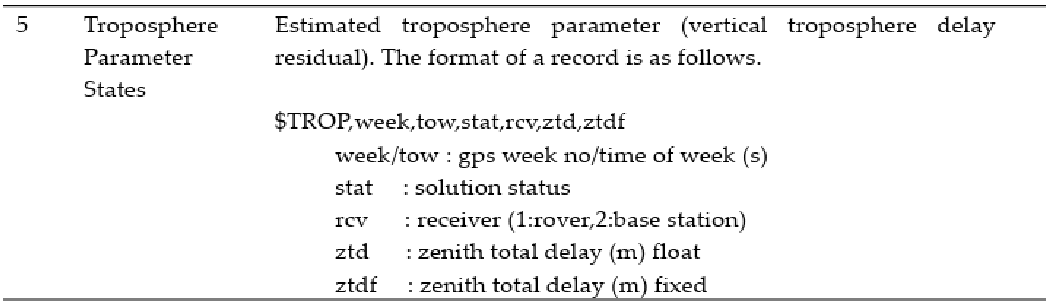

Trim001-2022-01-01.Std
Trim_220101.pos.stat ×
C: > Users > ACER > Desktop > rename > ZTD_stat > Trim_220101.pos.stat

```
$POS,2190,518400.000,6,-1359662.4596,6226306.7790,255808.0688,0.0000,0.0000,0.0000
$CLK,2190,518400.000,6,1,-14.178,-12.646,0.000,0.000
$TROP,2190,518400.000,6,1,2.5570,0.0000
$SAT,2190,518400.000,G12,1,132.4,18.6,-0.2365,-0.0005,1,30,4,0,1,0,0,0
$SAT,2190,518400.000,G18,1,351.1,63.9,-0.1354,-0.0001,1,36,4,0,1,0,0,0
$SAT,2190,518400.000,G23,1,350.4,23.1,0.2893,0.0003,1,22,4,0,1,0,0,0
$SAT,2190,518400.000,G24,1,49.0,50.5,-0.0218,-0.0000,1,42,4,0,1,0,0,0
$SAT,2190,518400.000,G25,1,169.0,23.5,-0.3389,-0.0005,1,44,4,0,1,0,0,0
$SAT,2190,518400.000,G29,1,191.9,35.7,0.8957,0.0015,1,48,4,0,1,0,0,0
$SAT,2190,518400.000,G32,1,263.8,26.2,-0.3093,-0.0003,1,46,4,0,1,0,0,0
$POS,2190,518430.000,6,-1359662.4597,6226306.7969,255808.0709,0.0000,0.0000,0.0000
$CLK,2190,518430.000,6,1,-14.242,-12.668,0.000,0.000
$TROP,2190,518430.000,6,1,2.5568,0.0000
$SAT,2190,518430.000,G12,1,132.2,18.6,-0.1971,0.0024,1,30,4,0,2,0,0,0
```

Figure 3.21: Example of output from RTKPOST pre-processing.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

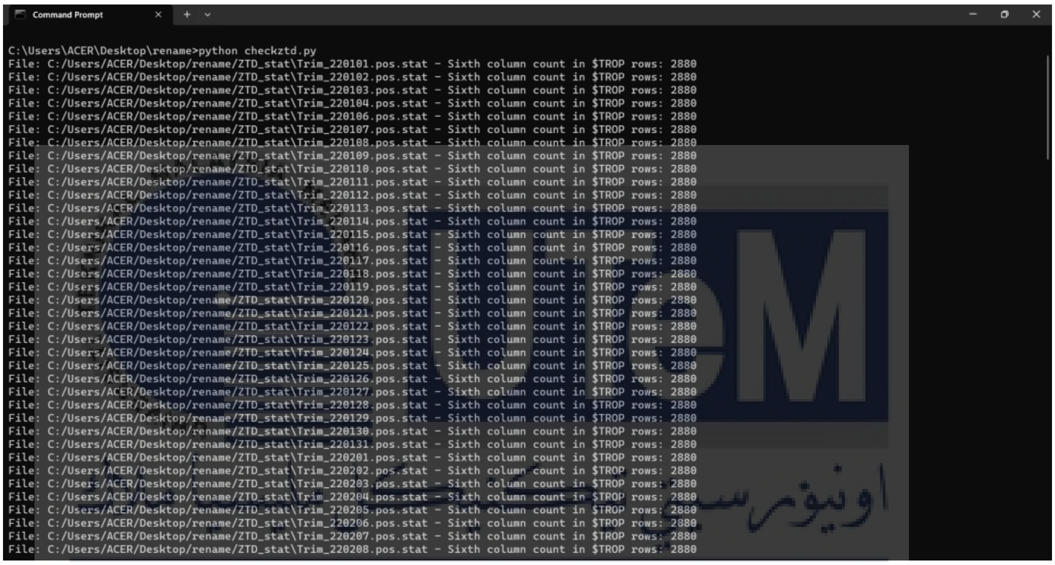
From the output obtained, manual book is referred to find the ZTD data. Page 107 of RTKLIB ver. 2.4.2 Manual Book shows that ZTD is in sixth column in $TROP, week,tow,stat,rcv,ztd,ztdf as per description in Figure 3.22 below.



5	Troposphere Parameter States	Estimated troposphere parameter (vertical troposphere delay residual). The format of a record is as follows. $TROP,week,tow,stat,rcv,ztd,ztdf week/tow : gps week no/time of week (s) stat : solution status rcv : receiver (1:rover,2:base station) ztd : zenith total delay (m) float ztdf : zenith total delay (m) fixed

Figure 3.22: Format of solution status file output of RTKPOST.

Before further processing the data, GPS week no. and time of week in second (week/tow) is checked to make sure that the output data obtained is correct. The next step is to check how many Zenith Total Delay (ZTD) values in each “. stat” file that obtained from RTKPOST. To perform this task, a Python code named *checkztd.py* is executed in the command prompt to assess the quantity of ZTD values within each .Std file as per Figure 3.23 below. The details of coding of *checkztd.py* coding are presented in Appendix E.

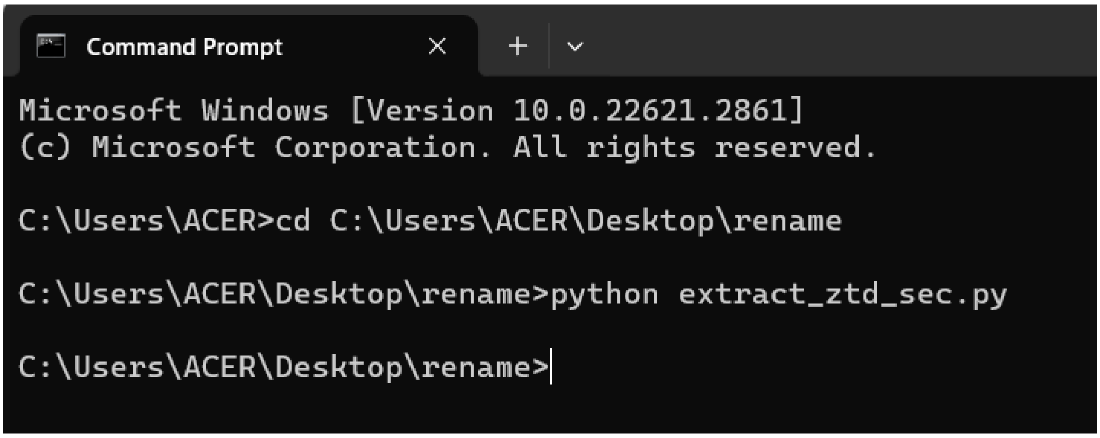


```
C:\Users\ACER\Desktop\rename>python checkztd.py
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220101.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220102.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220103.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220104.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220106.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220107.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220108.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220109.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220110.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220111.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220112.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220113.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220114.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220115.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220116.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220117.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220118.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220119.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220120.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220121.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220122.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220123.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220124.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220125.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220126.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220127.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220128.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220129.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220130.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220131.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220201.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220202.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220203.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220204.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220205.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220206.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220207.pos.stat - Sixth column count in $TROP rows: 2880
File: C:/Users/ACER/Desktop/rename/ZTD_stat\Trim_220208.pos.stat - Sixth column count in $TROP rows: 2880
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 3.23: *checkztd.py* is executed in command prompt.

According to the output of the checkztd.py code, there are 2880 records of zenith total delay (ZTD) data in each .stat file, except for specific dates: 25 November 2022 which has 1822 data, 29 November 2022 which has 2778 data, 5 December 2022 which has 2879 data and 16 December 2022 which has 2881 data. To make the process easier, the data is then extracted to a single .csv folder using *extract_ztd_sec.py* as per Appendix F. This extract_ztd_sec.py coding will take only zenith total delay value per day and save in excel .csv file with the same name as input file. The process is done in command prompt as presented in Figure 3.24.

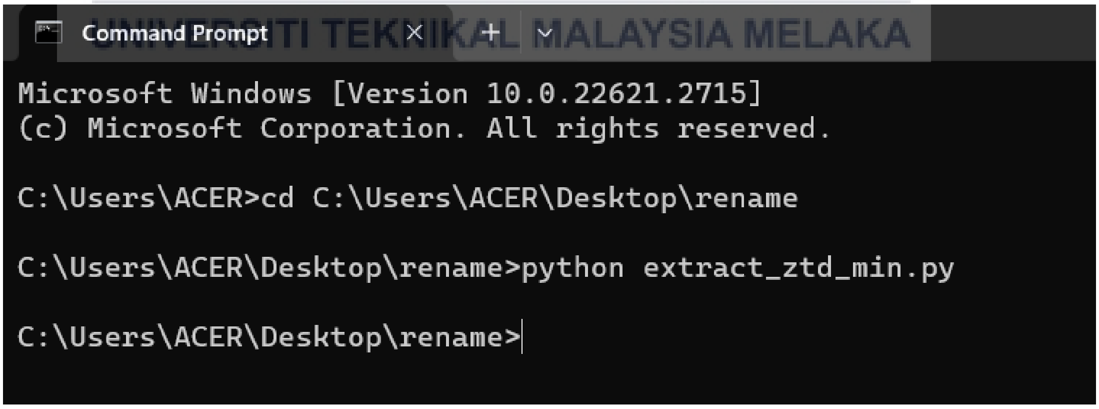


```
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ACER>cd C:\Users\ACER\Desktop\rename

C:\Users\ACER\Desktop\rename>python extract_ztd_sec.py

C:\Users\ACER\Desktop\rename>
```

Figure 3.24: extract_ztd_sec.py is executed in command prompt.

In ZTD result, 2880 units of data represent the quantity in seconds per day. To match the total electron content of 1440 units per day, measured in minutes per day, the 2880 units of data must undergo processing to synchronize and align with this measurement. Therefore, the next step is to change 2880 units of data (ZTD seconds) to 1440 units (ZTD minute) by using extract_ztd_min.py. The detail of the *extract_ztd_min.py* script can be found in Appendix G, and it is run using the command prompt as depicted in figure 3.25.

اونيورسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA



```
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ACER>cd C:\Users\ACER\Desktop\rename

C:\Users\ACER\Desktop\rename>python extract_ztd_min.py

C:\Users\ACER\Desktop\rename>
```

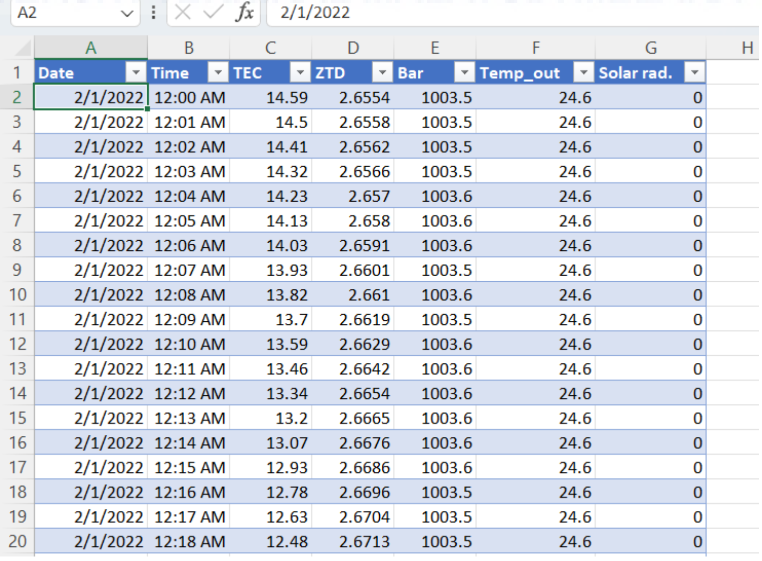
Figure 3.25: extract_ztd_min.py is executed in command prompt.

The last step for zenith total delay processing is to merge all data into one single .csv file. Python code *mergedztd.py* as in Appendix H is used to merge all data into one excel file named “*merged_ztd_data.csv*”.

3.2.3 Data Repository

To effectively train the appropriate dataset, it is necessary to consolidate all data into a designated folder, ensuring uniformity in the dataset size and synchronous timestamps across all included data. A dedicated data repository was established to centralize the datasets utilized in both the training and testing phases of the analysis. This repository served as a comprehensive collection point for all relevant datasets. In this project, the project data directory structure was organized into 7 columns and saved in an excel file. The 7 columns included Date, Time, TEC, ZTD, Bar, Temp_out, and Solar Rad. Figure 3.26 shows the dataset file in an excel format. Date and time are displayed to show the 6-month dataset used in this project. TEC, ZTD, Bar (Pressure), and Temp_out are dataset input while Solar Rad. is dataset output. Each column housed the specific datasets essential for different aspects of the research, ensuring a systematic arrangement and easy accessibility. The datasets within the repository were carefully curated and formatted to maintain consistency and facilitate seamless data handling during the experimentation phase.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA



Date	Time	TEC	ZTD	Bar	Temp_out	Solar rad.
2/1/2022	12:00 AM	14.59	2.6554	1003.5	24.6	0
2/1/2022	12:01 AM	14.5	2.6558	1003.5	24.6	0
2/1/2022	12:02 AM	14.41	2.6562	1003.5	24.6	0
2/1/2022	12:03 AM	14.32	2.6566	1003.5	24.6	0
2/1/2022	12:04 AM	14.23	2.657	1003.6	24.6	0
2/1/2022	12:05 AM	14.13	2.658	1003.6	24.6	0
2/1/2022	12:06 AM	14.03	2.6591	1003.6	24.6	0
2/1/2022	12:07 AM	13.93	2.6601	1003.5	24.6	0
2/1/2022	12:08 AM	13.82	2.661	1003.6	24.6	0
2/1/2022	12:09 AM	13.7	2.6619	1003.5	24.6	0
2/1/2022	12:10 AM	13.59	2.6629	1003.6	24.6	0
2/1/2022	12:11 AM	13.46	2.6642	1003.6	24.6	0
2/1/2022	12:12 AM	13.34	2.6654	1003.6	24.6	0
2/1/2022	12:13 AM	13.2	2.6665	1003.6	24.6	0
2/1/2022	12:14 AM	13.07	2.6676	1003.6	24.6	0
2/1/2022	12:15 AM	12.93	2.6686	1003.6	24.6	0
2/1/2022	12:16 AM	12.78	2.6696	1003.5	24.6	0
2/1/2022	12:17 AM	12.63	2.6704	1003.5	24.6	0
2/1/2022	12:18 AM	12.48	2.6713	1003.5	24.6	0

Figure 3.26: Dataset folder in excel file.

In addition, for the 6 months dataset, the optimal or most comprehensive six-month dataset is chosen from the processed one-year dataset. This comprehensive six-months dataset comprises data from February, March, April, August, September, and October. Each input and output dataset contains an equal amount of data, and the specifics of these six datasets are detailed in the table below.

Table 3.1: Dataset folder information.

No.	Month	Day in each month in 2022	Day in the dataset	Days/Dates that are not in the dataset	Total dataset for each month
1.	February	28 days	27 days	17 February 2022	38880
2.	March	31 days	31 days	-	44640
3.	April	30 days	29 days	4 April 2022	41760
4.	August	31 days	30 days	15 August 2022	43200
5.	September	30 days	30 days	-	43200
6.	October	31 days	31 days	-	44640
				Total:	256320

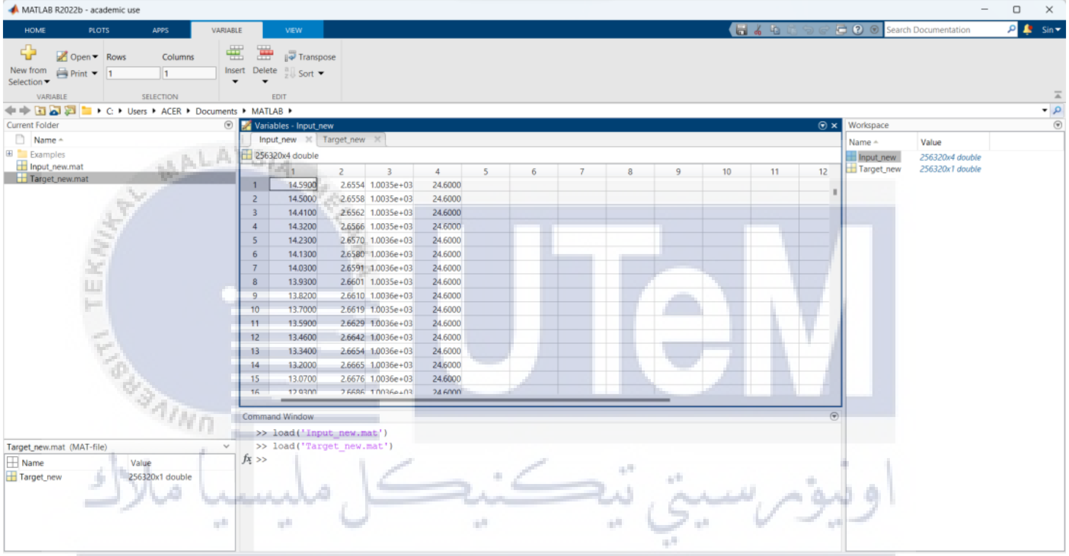
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

3.3 Training and Testing suitable Machine Learning Model using Preprocessed Dataset

This part will be discussing how to train and test the preprocessed dataset with suitable machine learning model. The process is done in MATLAB GUI toolbox or commonly called MATLAB deep learning toolbox. How to input the dataset into MATLAB deep learning toolbox, dataset partitioning, model selection and configuration, model training, model testing and validation will be discussed in detail in this part.

3.3.1 Input Preprocessed Dataset

Firstly, installing MATLAB software. The MATLAB software version used in this project is MATLAB R2022b. After that, two files are created in MATLAB workspace which is Input_new file and Target_new file. Paste the input dataset into Input_new file as per Figure 3.27 below, the data has 4 columns and 256320 rows. After that, paste the dataset output into Target_new file as per Figure 3.28 below. The output is the solar radiation dataset, only in one column.



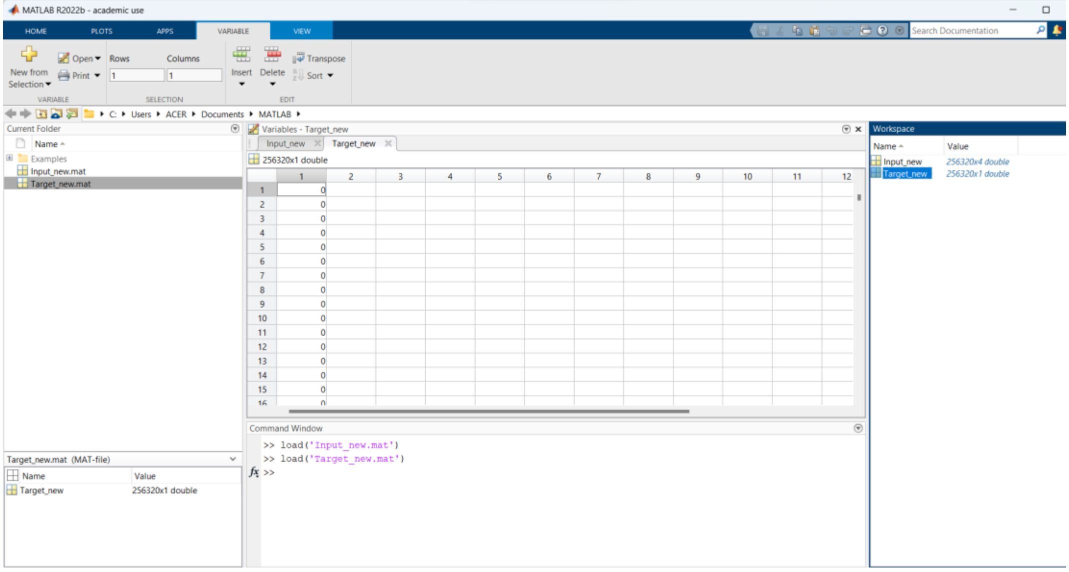
256320x4 double

	1	2	3	4
1	14.5900	2.6554	1.0035e+03	24.6000
2	14.5000	2.6558	1.0035e+03	24.6000
3	14.4100	2.6562	1.0035e+03	24.6000
4	14.3200	2.6566	1.0035e+03	24.6000
5	14.2300	2.6570	1.0036e+03	24.6000
6	14.1300	2.6580	1.0036e+03	24.6000
7	14.0300	2.6591	1.0036e+03	24.6000
8	13.9300	2.6601	1.0035e+03	24.6000
9	13.8200	2.6610	1.0036e+03	24.6000
10	13.7000	2.6619	1.0035e+03	24.6000
11	13.5900	2.6629	1.0036e+03	24.6000
12	13.4600	2.6642	1.0036e+03	24.6000
13	13.3400	2.6654	1.0036e+03	24.6000
14	13.2000	2.6665	1.0036e+03	24.6000
15	13.0700	2.6676	1.0036e+03	24.6000

```
>> load('Input_new.mat')
>> load('Target_new.mat')
fx >>
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 3.27: Input_new dataset in MATLAB workspace.



256320x1 double

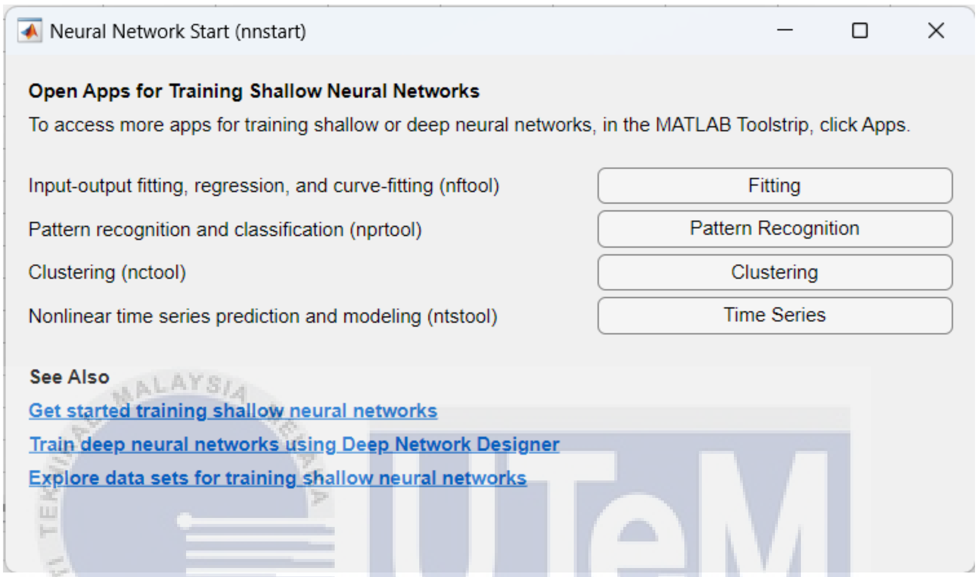
	1
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0

```
>> load('Input_new.mat')
>> load('Target_new.mat')
fx >>
```

Figure 3.28: Target_new dataset in MATLAB workspace.

3.3.2 Training the Preprocessed Dataset

The training process starts with opening the MATLAB deep learning toolbox by writing “nnstart” in MATLAB Command Window. Next, the interface as in Figure 3.29 will pop up and ‘fitting’ button is clicked.



Neural Network Start (nnstart)

Open Apps for Training Shallow Neural Networks

To access more apps for training shallow or deep neural networks, in the MATLAB Toolstrip, click Apps.

Input-output fitting, regression, and curve-fitting (nftool)	Fitting
Pattern recognition and classification (nprtool)	Pattern Recognition
Clustering (nctool)	Clustering
Nonlinear time series prediction and modeling (ntstool)	Time Series

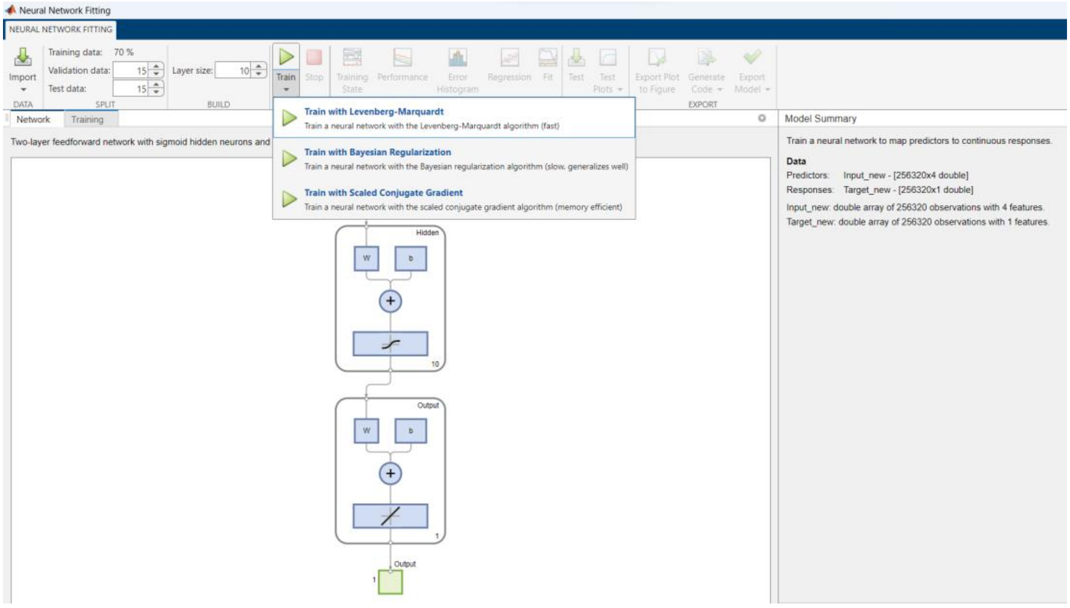
See Also

Get started training shallow neural networks
Train deep neural networks using Deep Network Designer
Explore data sets for training shallow neural networks

Figure 3.29: Neural Network start (nnstart) interface.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Furthermore, import datasets from workspace and select data for training the network. Predictor is the input data while responses is the target or output data. The next step is dataset partitioning. The preprocessed dataset was divided into three distinct subsets: a training set, a testing set, and a validation set. The dataset was divided such that 70% of the data formed the training set for model training. The remaining 30% was further split into a testing set of 15% and a validation set of 15% to evaluate the model's performance. The layer size can be modified to achieve optimal performance for the model. For model training, a backpropagation algorithm is used. There are three algorithms that can be chosen which are to train with Levenberg-Marquardt, Bayesian Regularization, and Scaled Conjugate Gradient as per Figure 3.30.



Training data: 70 %
Validation data: 15
Test data: 15
Layer size: 10

Train with Levenberg-Marquardt
Train a neural network with the Levenberg-Marquardt algorithm (fast)

Train with Bayesian Regularization
Train a neural network with the Bayesian regularization algorithm (slow, generalizes well)

Train with Scaled Conjugate Gradient
Train a neural network with the scaled conjugate gradient algorithm (memory efficient)

Two-layer feedforward network with sigmoid hidden neurons and

Model Summary

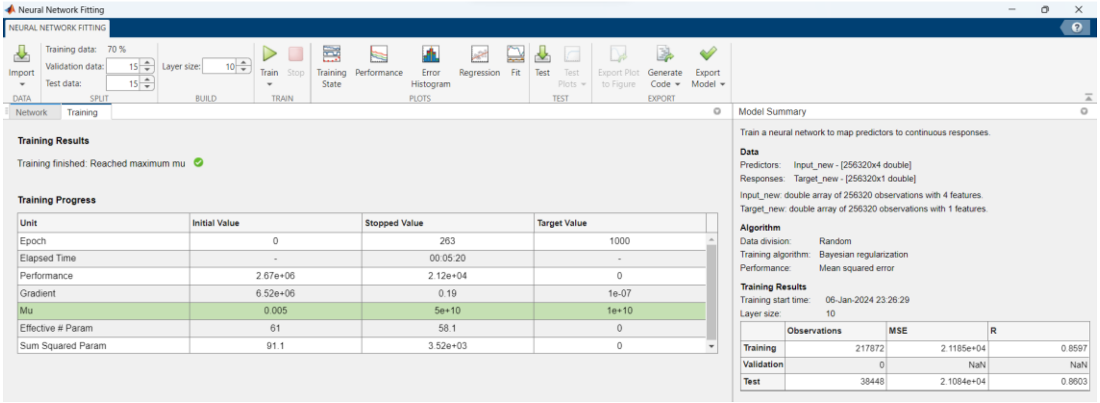
Train a neural network to map predictors to continuous responses.

Data
Predictors: Input_new - [256320x4 double]
Responses: Target_new - [256320x1 double]
Input_new: double array of 256320 observations with 4 features.
Target_new: double array of 256320 observations with 1 features.

Figure 3.30: Three Neural Network Fitting training algorithm.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

After choosing a suitable algorithm for training, click the train button to start training. Figure 3.31 shows the training process is completed. Upon completion of the training process, the training state, performance, error histogram and regression plot can be obtained from the PLOTS section of the neural network fitting.



Training data: 70 %
Validation data: 15
Test data: 15
Layer size: 10

Training Results

Training finished: Reached maximum mu ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	263	1000
Elapsed Time	-	00:05:20	-
Performance	2.67e+06	2.12e+04	0
Gradient	6.52e+06	0.19	1e-07
Mu	0.005	5e+10	1e+10
Effective # Param	61	58.1	0
Sum Squared Param	91.1	3.52e+03	0

Model Summary

Train a neural network to map predictors to continuous responses.

Data
Predictors: Input_new - [256320x4 double]
Responses: Target_new - [256320x1 double]
Input_new: double array of 256320 observations with 4 features.
Target_new: double array of 256320 observations with 1 features.

Algorithm
Data division: Random
Training algorithm: Bayesian regularization
Performance: Mean squared error

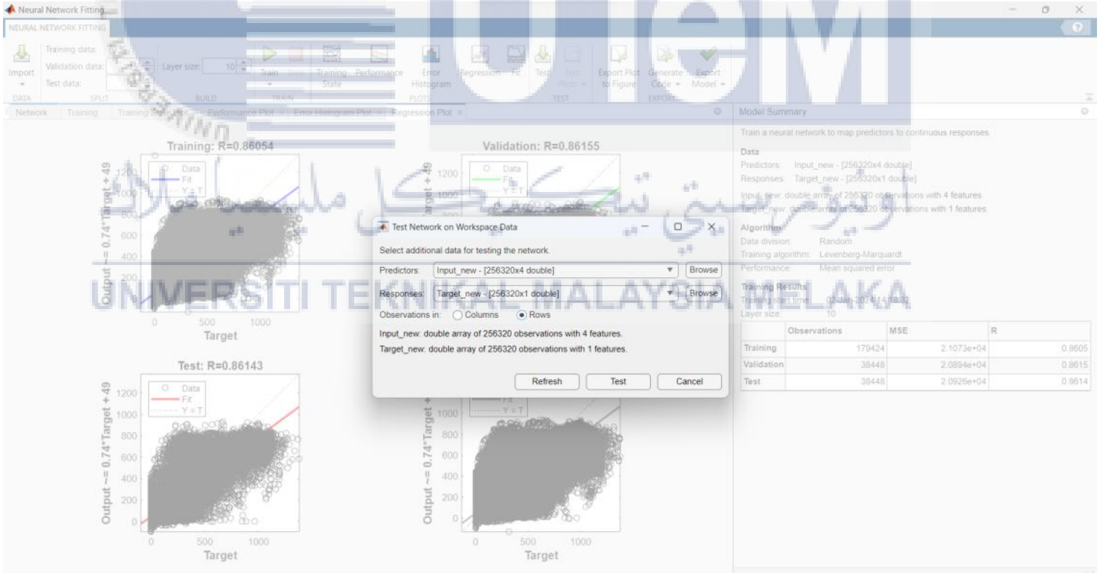
Training Results
Training start time: 06-Jan-2024 23:26:29
Layer size: 10

	Observations	MSE	R
Training	217872	2.1185e+04	0.8597
Validation	0	NaN	NaN
Test	38448	2.1084e+04	0.8603

Figure 3.31: Training process is completed.

3.3.3 Testing the Model

Next is to test the solar irradiance model that has been trained. “Test” button is clicked and interface “Test Network on Workspace Data” as in Figure 3.32 pops out. Daily, monthly, or yearly data can be input to the predictors and responses section and test the model. For this project, monthly data of February has been tested for solar irradiance model evaluation. First, insert a February data input into MATLAB with the name “Feb_input.mat” and insert February solar irradiance data as output with name “Feb_target.mat”. Click observations in columns or rows to make sure that the number of data observations and features is correct before conducting the testing process. The regression and error histogram of the tested dataset can be obtained from the test section.



Test Network on Workspace Data

Select additional data for testing the network.

Predictors: Input_new - [256320x4 double]

Responses: Target_new - [256320x1 double]

Observations in: ○ Columns ● Rows

Input_new: double array of 256320 observations with 4 features.

Target_new: double array of 256320 observations with 1 features.

Train a neural network to map predictors to continuous responses.

Data

Predictors: Input_new - [256320x4 double]

Responses: Target_new - [256320x1 double]

Input_new: double array of 256320 observations with 4 features.

Target_new: double array of 256320 observations with 1 features.

Algorithm

Data division: Random

Training algorithm: Levenberg-Marquardt

Performance: Mean squared error

Training Results

Training start time: 02-Jan-2024 14:18:32

Layer size: 10

	Observations	MSE	R
Training	179424	2.1073e+04	0.8605
Validation	38448	2.0894e+04	0.8615
Test	38448	2.0926e+04	0.8614

Figure 3.32: Testing solar irradiance model.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

3.3.4 Validating the Model

Then, click “Generate Code” in EXPORT section to produce the coding of the solar irradiance model. The coding for the model is saved as "BR10model.m" and is presented in detail within Appendix I. The code for the solar irradiance model is utilized for model validation. This validation involves utilizing the dataset obtained after running the code to generate a graph in MATLAB. The graph showcases a comparison between measured solar irradiance and predicted solar irradiance. Measurement solar irradiance is taken from FTKEK weather data solar radiation. While the predicted solar irradiance is obtained from the MATLAB workspace.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 4

RESULTS AND DISCUSSION

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

This chapter included the completed report on its theoretical scope and the best neural network model for this project. The analysis results, mean square error (MSE), and the Correlation Coefficient (R) were also recorded during the implementation of the project. The results for the training and testing of the machine learning model were shown in this part. Outputs of the model one by one were shown in this part, including details of each result.

4.1 Training Model

In this project, six months of dataset (February, March, April, August, September, October) were used for training the models to forecast solar irradiance for February 2022. Training a model with different algorithms and layer size resulted in different output, yielding different mean square error (MSE) and correlation coefficient (R). The preprocessed dataset underwent training employing three backpropagation algorithms, each utilizing different layer sizes: 5, 8, and 10, as illustrated in Table 4.1. During training, the MSE was used as a measure of how well the neural network learned the patterns within the training data. It means that the lower the mean square error of the model, the better the model training. While the correlation coefficient (R) could be used as a measure of how well the predicted values from the neural network model align with the actual target values. The correlation coefficient, ranging from -1 to 1, indicates the degree of closeness between variables. A higher value of R approaching 1 signifies a stronger association between the variables, implying a better model performance. Therefore, from Table 4.1 below, it is shown that Bayesian Regularization algorithm model with 10 hidden layer size is the best model among a few algorithms that were tested.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Table 4.1: MSE and R results when training on different Algorithm and Layer size.

Algorithm	Layer Size	Mean Square Error (MSE)	Correlation Coefficient (R)
Levenberg-Marquardt	5	22635.0758	0.85149
	8	21671.6315	0.85816
	10	20894.1821	0.86083
Bayesian Regularization	5	21740.4728	0.85549

	8	20899.5107	0.86136
	10	20882.4233	0.86138
Scaled Conjugate Gradient	5	22413.4795	0.84889
	8	25279.5041	0.83225
	10	24324.8056	0.83575

Figure 4.1 is the neural network training state, which refers to the condition or status of the neural network during the training process. Gradient represents slope of tangent of graph of function. It indicates the direction where the function experiences a notable increase in its rate. 'mu' serves as a governing factor in our modeled back-propagation neural network, directly influencing the convergence of errors. Validation checks are employed to halt the neural network's learning process. The count of validation checks corresponds to the count of consecutive iterations of the neural network. In Figure 4.2, a dotted line shows the best training performance plot (MSE plot) is 20882.4233 at epoch 144. It is the average of squares of errors or deviation. The histogram in Figure 4.3 illustrates the error plot of the model training. The distribution exhibits a symmetric shape with a significant concentration of data points centered around zero error. Both the mean and median values are close to zero, implying a central tendency of the data at this point. This suggests that a substantial proportion of the dataset records errors near zero, potentially indicating accurate predictions or measurements for a particular phenomenon. Figure 4.4 shows the regression plot of the training model. The plot displays the relationships between the measurement solar irradiance(T) and the predicted solar irradiance (Y). A strong positive correlation regression line is observed. However, as the plot extends to the right side, a deviation occurs where the regression line falls below the linear regression line Y=T.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

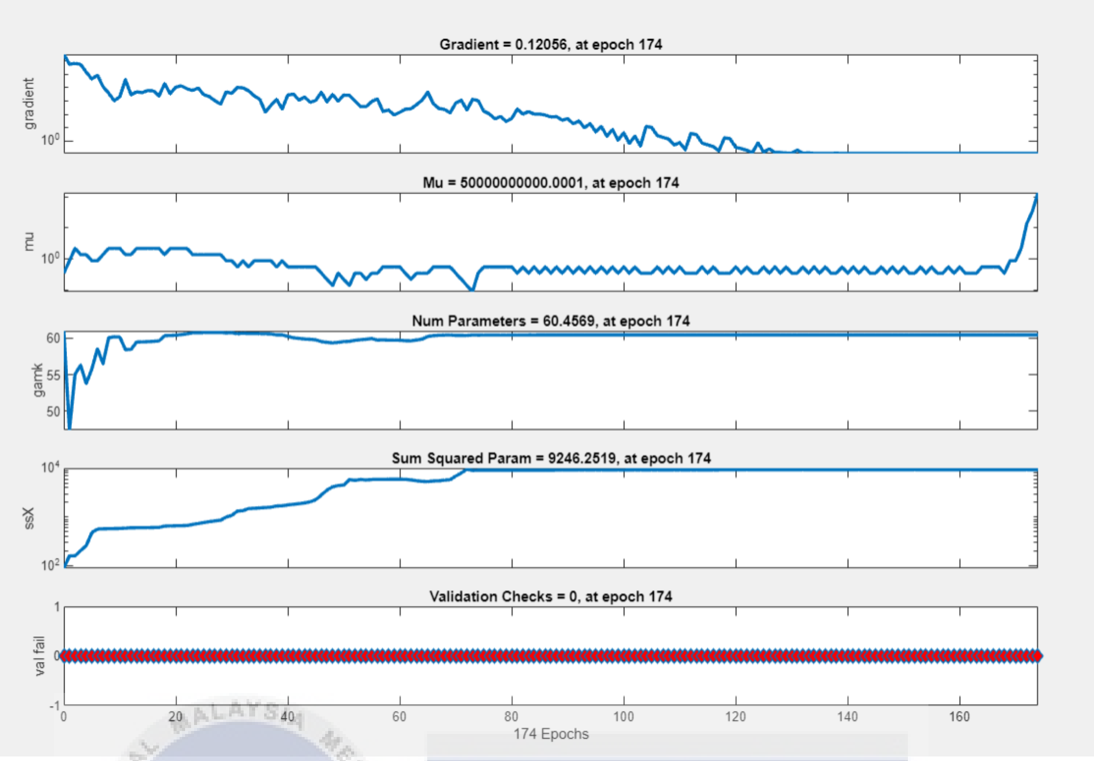


Figure 4.1: Training State Plot.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

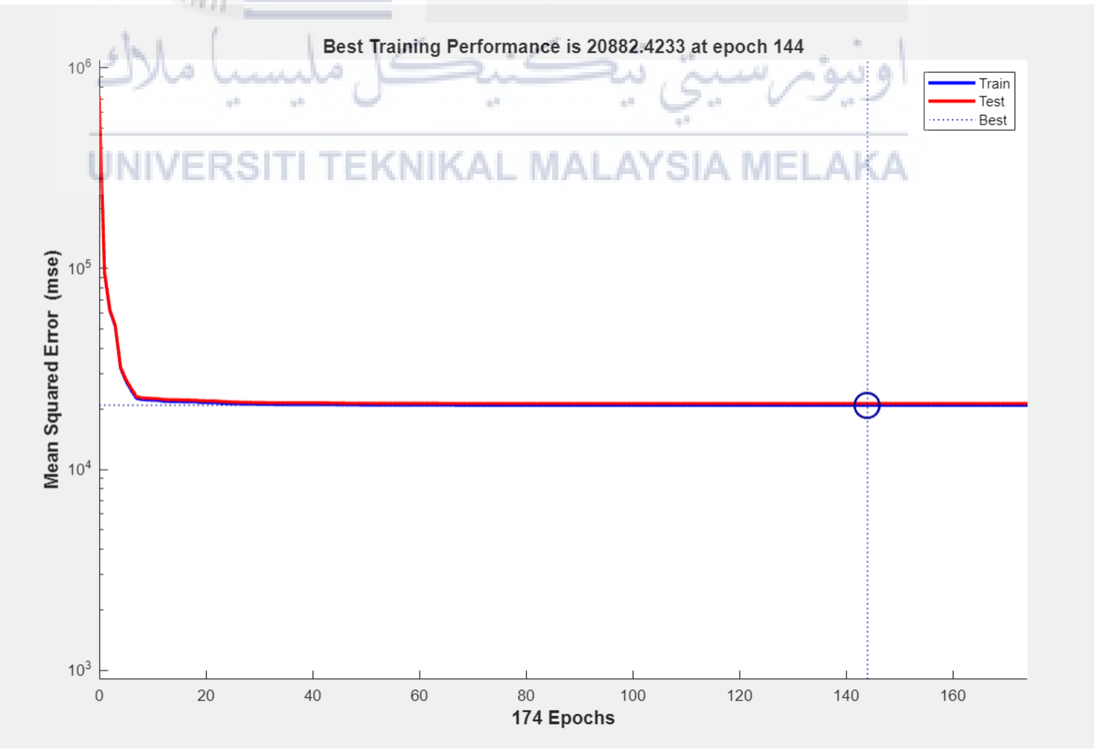


Figure 4.2: Mean Square Error (MSE) Plot.

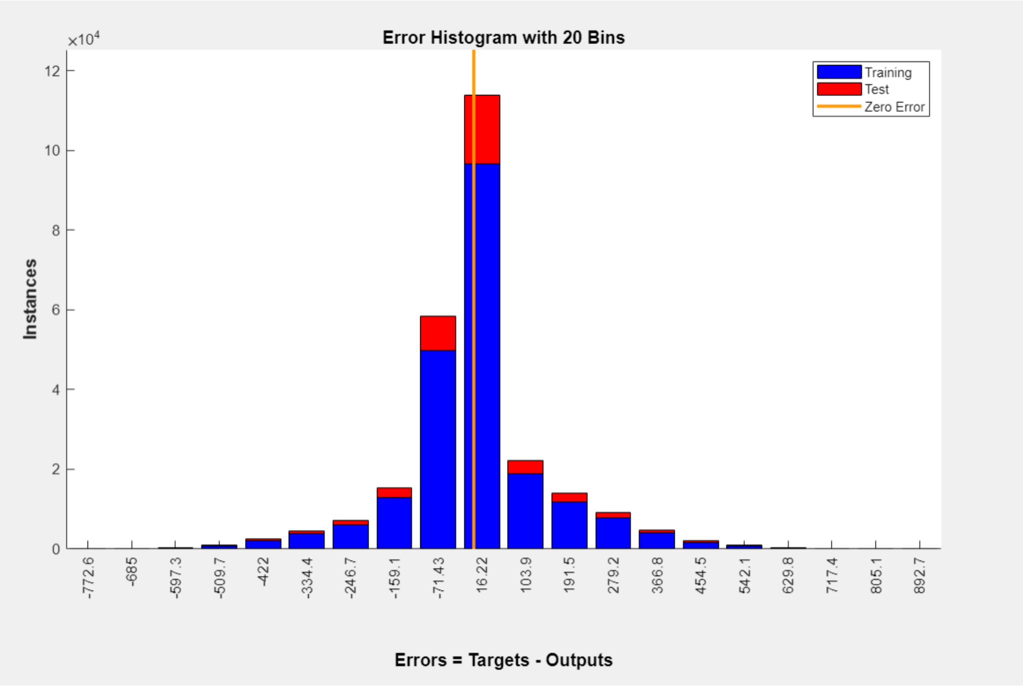


Figure 4.3: Error Histogram Plot.

UTeM
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

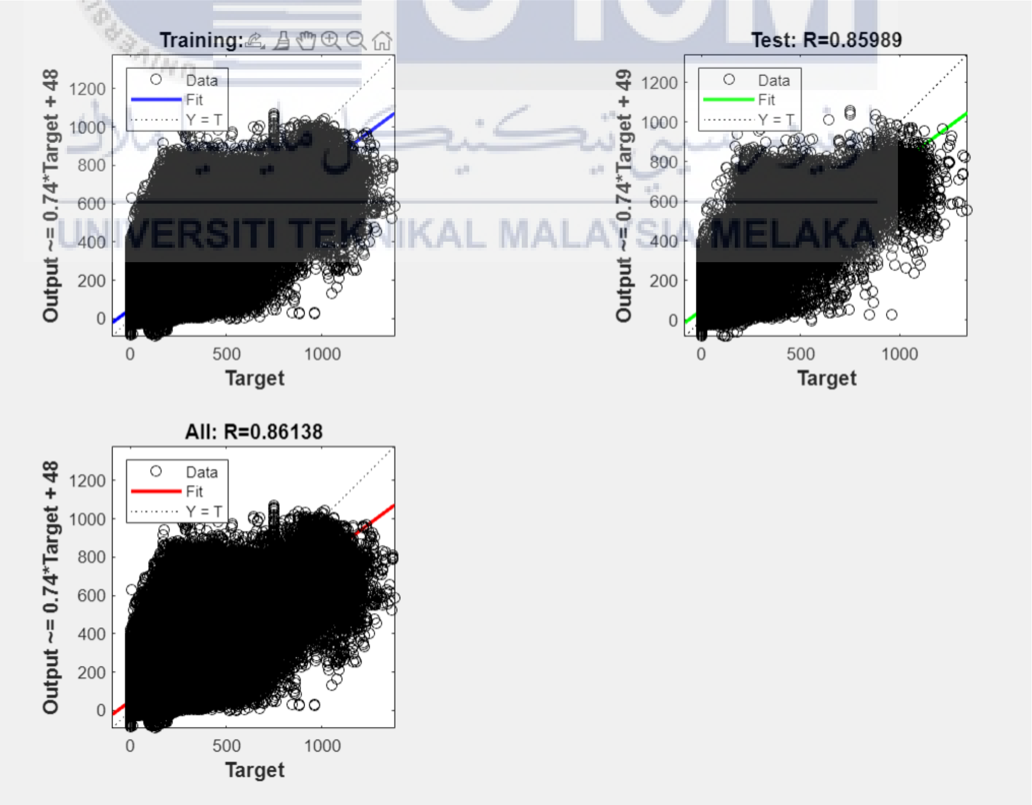
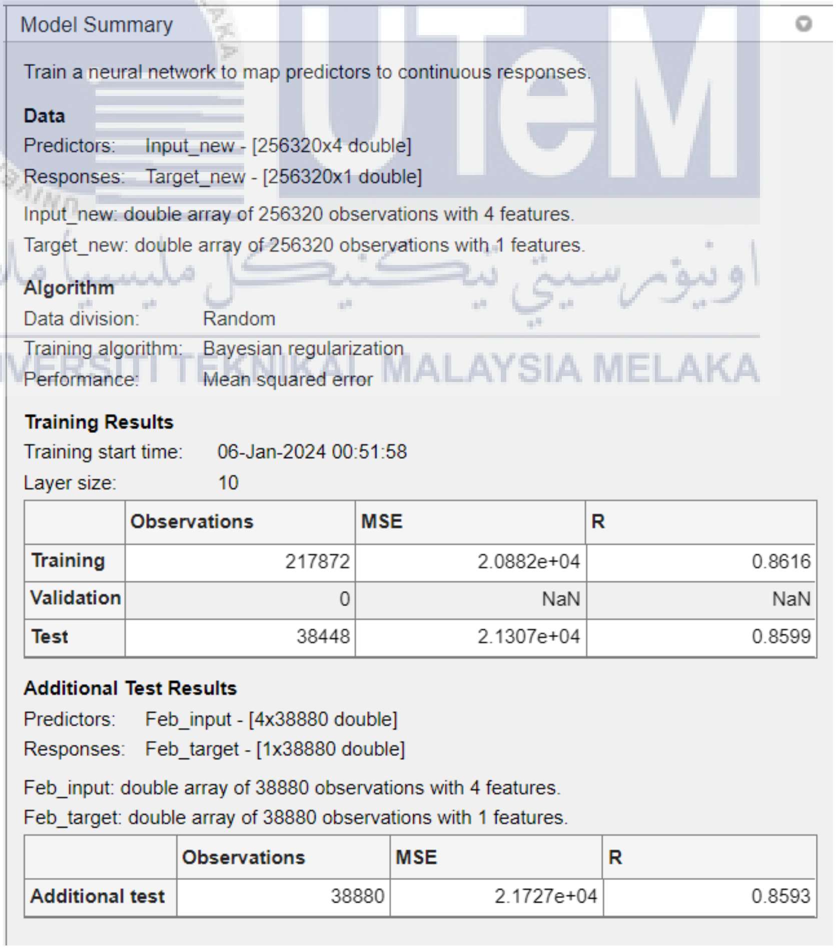


Figure 4.4: Regression Plot.

4.2 Testing Model Plot

After the training process, one month of data in February was tested. Figure 4.5 displays the model summary. It provides a comprehensive overview and essential information about the trained neural network model, including input data, training algorithm, training results, and additional test results. Predictors are the input data, responses are output/solar irradiance data. Figure 4.6 is the February error histogram plot symmetric shape with a significant concentration of data points centred around zero error. While Figure 4.7 shows the regression plot of the additional testing. A moderate to weak positive correlation regression line was observed. When the plot extends to the right side, the regression line falls below the linear regression line Y=T.



Model Summary

Train a neural network to map predictors to continuous responses.

Data
Predictors: Input_new - [256320x4 double]
Responses: Target_new - [256320x1 double]

Input_new: double array of 256320 observations with 4 features.
Target_new: double array of 256320 observations with 1 features.

Algorithm
Data division: Random
Training algorithm: Bayesian regularization
Performance: Mean squared error

Training Results
Training start time: 06-Jan-2024 00:51:58
Layer size: 10

	Observations	**MSE**	**R**
Training	217872	2.0882e+04	0.8616
Validation	0	NaN	NaN
Test	38448	2.1307e+04	0.8599

Additional Test Results
Predictors: Feb_input - [4x38880 double]
Responses: Feb_target - [1x38880 double]

Feb_input: double array of 38880 observations with 4 features.
Feb_target: double array of 38880 observations with 1 features.

	Observations	**MSE**	**R**
Additional test	38880	2.1727e+04	0.8593

UTeM
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 4.5: The model summary.

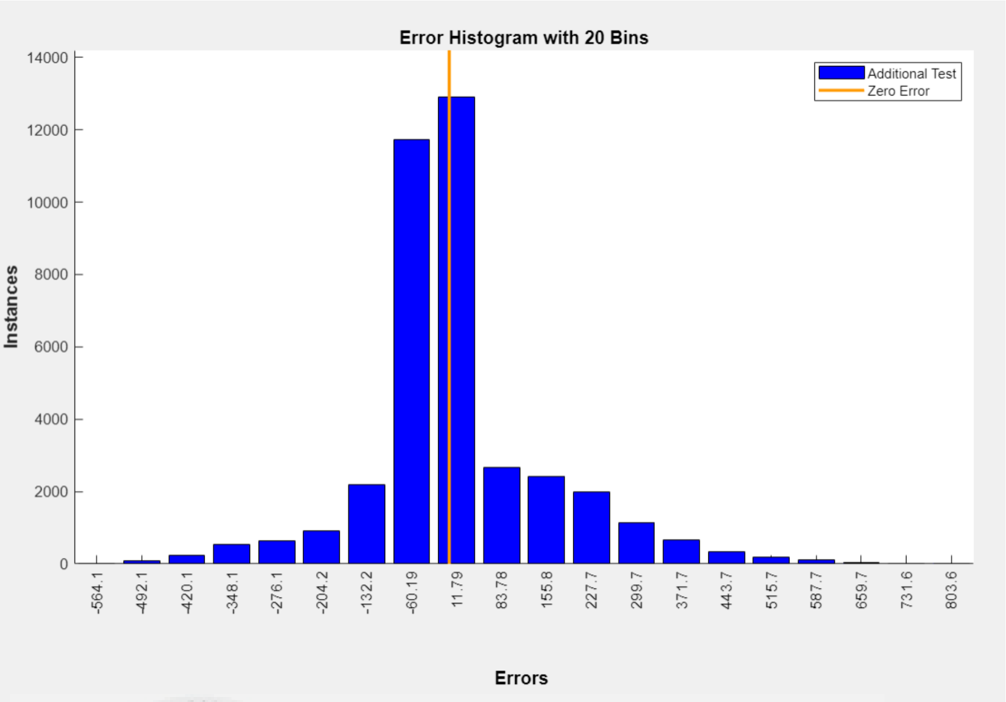


Figure 4.6: Additional testing Error Histogram Plot.

اونيورسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

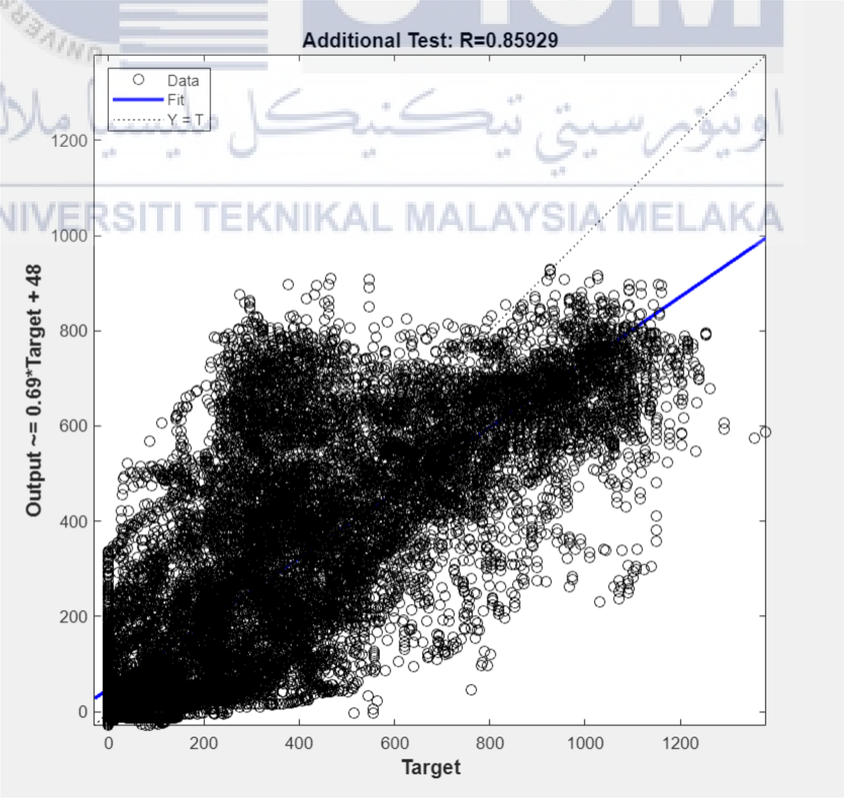


Figure 4.7: Regression Plot for additional test 1 month data (February).

4.3 Comparison between Measurement and Predicted Solar Irradiance

To assess the performance of the solar irradiance forecasting model, a dataset from a single day (February 1, 2022) and the entire month of February were employed to visualize both the measured and predicted solar irradiance. Figure 4.8 shows comparison between measurement and predicted solar irradiance plot for 1 day data. Figure 4.9 shows comparison between measurement and predicted solar irradiance plot for one month data (February). The measurement dataset comprises solar irradiance data sourced from FTKEK solar radiation records. Conversely, the predicted solar irradiance dataset originates from the output data generated by MATLAB software. This comparison shows that both the measurement and predicted data have almost the same graph shape, stating that the data output from the model is almost the same as the actual solar irradiance data. Moreover, error histogram below the waveform plot shows the data difference between measurement and predicted solar irradiance. The histogram illustrates a symmetric shape with a substantial concentration of data points centred around zero, indicating a favourable outcome, indicating a close alignment between the measured and predicted values for solar irradiance. Lastly, the graph plotted in MATLAB GUI shows satisfactory results for this project.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

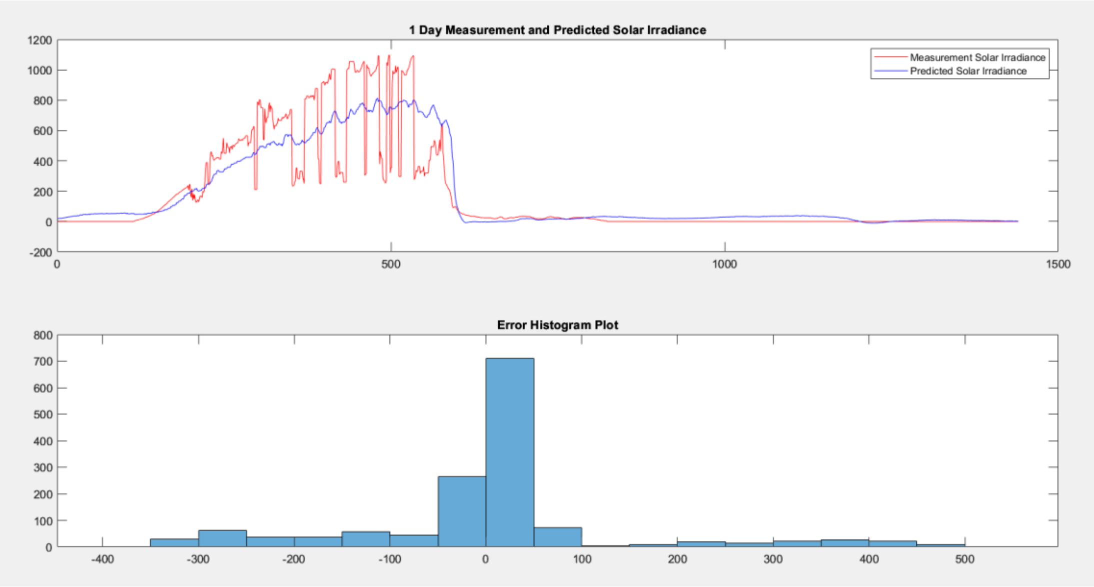


Figure 4.8: Comparison between measurement and predicted solar irradiance plot for 1 day data.

UTeM
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

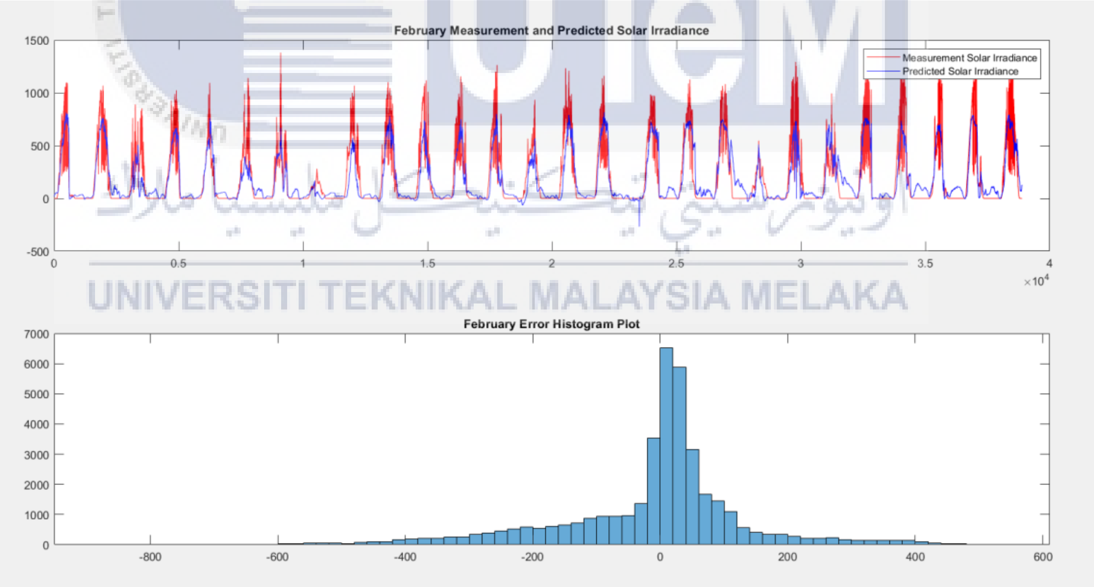


Figure 4.9: Comparison between measurement and predicted solar irradiance plot for one month data (February).

4.4 Discussion on problems and solutions

There were a few issues that arose in the process of completing this project. At the beginning of the project, there were errors in renaming the GPS files to find the TEC. To address this, various Python coding edits were made and tested numerous times in the command prompt until achieving success in renaming the files. The troubleshooting process consumed significant time due to the iterative nature of testing different approaches to identify the suitable solution. When extracting TEC from GPS Gopi Seemala software also needed to be tested many times until a correct result was obtained. The large amount of data led to data processing requiring coding and taking a longer time. In addition, while handling ZTD data, it was essential to note that the processing could not be executed in a batch mode; rather, it necessitated a day-by-day approach. For instance, in this study encompassing the 2022 dataset spanning 362 days, the processing protocol mandates the data to be individually processed 362 times to derive the specific ZTD value for each day. Moreover, another issue that occurred was the incomplete GPS data and FTKEK weather data. During the data processing phase, it was crucial to verify the sufficiency of data for each day and identify any dates where the data might be inadequate or insufficient. To overcome this problem, the data from Excel was inputted into a table, and a month-by-month assessment was conducted to identify dates with insufficient data. Hence, the appropriate six-month data could be determined.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 5

CONCLUSION AND FUTURE WORKS

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

5.1 Conclusion

In conclusion, a machine leaning-based solar irradiance forecasting model using GPS has been developed in this study. Throughout this study, the set objectives were effectively achieved. Firstly, the study successfully attained the goal of computing Atmospheric Integrated Water Vapor (IWV) and Ionospheric Total Electron Content (TEC) using GPS datasets. This analysis facilitated a comprehensive understanding of crucial meteorological and ionospheric parameters vital for solar irradiance prediction. Secondly, employing a meticulously preprocessed dataset, a machine learning model was aptly trained. This process involved careful data preparation and selection, ensuring the model's optimal learning from the input variables derived from GPS data. Finally, the ultimate objective of forecasting solar irradiance utilizing the derived IWV and TEC in conjunction with an Artificial Neural Network (ANN) was accomplished.

It is evident that the ANN machine learning model is able to forecast solar irradiance with an acceptable degree of accuracy and reliability. The ANN machine learning model with Bayesian Regularization algorithm and ten neurons provided the highest accuracy with an R of 0.86138 and MSE of 20882.4233.

The development of this study stands as a significant advancement in solar energy prediction. The significance of this model lies in its ability to aid solar energy planning, facilitate grid integration, and optimize energy resource allocation, thereby contributing to the creation of more efficient and sustainable energy systems. By leveraging machine learning techniques and GPS-derived data for solar irradiance forecasting, this model provides accurate predictions crucial for planning solar energy operations. These forecasts allow stakeholders to anticipate fluctuations in solar energy availability, enabling informed decisions on energy storage, distribution, and grid management. Additionally, the model aids in effectively integrating solar energy into existing power grids by providing reliable forecasts, ensuring a smooth transition between different energy sources, and enhancing grid stability. Moreover, its capacity to accurately allocate energy resources based on predicted solar irradiance levels results in optimized energy utilization, reducing costs and enhancing overall system efficiency. As a result, this model serves as a vital tool in fostering the development of sustainable energy systems that rely more effectively on renewable energy sources while minimizing environmental impact.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

However, the results of the study recorded in Chapter 4 could be improved. Within solar irradiance and weather station dataset, the presence of missing values denoted by '–' signifies instances where data was absent or unavailable at specific times or days. Moreover, notable gaps in the data exist, leading to the loss of 1-2 hours' worth of data on certain days. These inconsistencies and missing segments pose significant challenges in accurately capturing the complete temporal dynamics of solar irradiance and weather conditions. Lastly, there were many problems that occurred along with the process of developing this project. Thus, the problems encountered can ultimately be solved. The ways and steps in problem solving were described in chapter 4.

5.2 Future Work

Some suggestions for future research work and development to further improve the system are recommended in this section:

a) Addressing the critical aspect of data quality of input data and preprocessing in GPS-derived data

The accuracy of solar irradiance predictions hinges significantly upon the quality and preparation of input data. Enhancing data preprocessing techniques involves meticulous procedures to filter out noise, correct anomalies, and handle missing or incomplete data points within the GPS datasets. By refining data preprocessing methodologies and ensuring data accuracy, the reliability and precision of the solar irradiance forecasting model can be substantially improved.

b) Scaling and Geographic Expansion

This involves scaling and geographic expansion to enhance solar irradiance forecasting applicability on a global level. This encompasses adapting the model to various geographic regions and diverse environmental conditions, ensuring its effectiveness in predicting solar irradiance worldwide. By incorporating data from different locations and environmental variables, such as topography and weather patterns, the model can be optimized to provide accurate forecasts across a broader range of geographical areas. Expanding its scope globally would contribute significantly to the advancement of solar energy utilization by enabling more precise and reliable solar irradiance predictions across diverse regions.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

REFERENCES

[1] C. Voyant et al., “Machine learning methods for solar radiation forecasting: A review,” Renewable Energy, vol. 105, pp. 569–582, May 2017, doi: https://doi.org/10.1016/j.renene.2016.12.095.

[2] E. Lorenz, J. Remund, S.C. Müller, W. Traunmüller, G. Steinmaurer, D. Pozo, J.A. Ruiz-Arias, V.L. Fanego, L. Ramirez, M.G. Romeo, others, Benchmarking of different approaches to forecast solar irradiance, in: 24th Eur. Photovolt. Sol. Energy Conf. Hambg. Ger., 2009: p. 25. http://task3.ieashc.org/data/sites/1/publications/24th_EU_PVSEC_5BV.2.50_lorenz_final.pdf (accessed March 4, 2015).

[3] B. Espinar, J.-L. Aznarte, R. Girard, A.M. Moussa, G. Kariniotakis, Photovoltaic Forecasting: A state of the art, in: OTTI - Ostbayerisches Technologie-Transfer-Institut, 2010: p. Pages 250-255-ISBN 978-3-941785-15-1. https://hal-mines-paristech.archives-ouvertes.fr/hal-00771465/document (accessed March 4, 2015)

[4] Mahesh, B. (2020) Machine Learning Algorithms—A Review. International Journal of Science and Research, 9, 381-386. x1

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[5] T. Oladipupo, “Machine learning overview,” New Advances in Machine Learning, 2010. doi:10.5772/9374.

[6] U. Hemavathi, A. C. Medona, V. Dhilip Kumar, and R. Raja Sekar, “Review for the solar radiation forecasting methods based on machine learning approaches,” Journal of Physics: Conference Series, vol. 1964, no. 4, p. 042065, 2021. doi:10.1088/1742-6596/1964/4/042065.

[7] Shiruru, Kuldeep. (2016). An Introduction to Artificial Neural Network. International Journal of Advance Research and Innovative Ideas in Education. 1. 27-30.

[8] Breiman, L.: Random Forests. Machine Learning 45 (1) pp. 5–32 (2001)

[9] Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. NeuralComputation 9(7) pp. 1545-1588 (1997).

[10] A. Cutler, D. R. Cutler, and J. R. Stevens, “Random forests,” *Ensemble Machine Learning*, pp. 157–175, 2012. doi:10.1007/978-1-4419-9326-7_5.

[11] A. Natekin and A. Knoll, “Gradient Boosting Machines, a tutorial,” *Frontiers in Neurorobotics*, vol. 7, 2013. doi:10.3389/fnbot.2013.00021.

[12] J. Fan, X. Wang, L. Wu, H. Zhou, F. Zhang, X. Yu, et al., Comparison of Support Vector Machine and Extreme Gradient Boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: a case study in China, Energy Convers. Manag. 164 (2018) 102–111.

[13] M. Guermoui, A. Rabehi, K. Gairaa, S. Benkaciali, Support vector regression

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

methodology for estimating global solar radiation in Algeria, Eur. Phys. J. Plus 133(1) (2018) 1–9.

[14] B. M. Alluhaidah, H. H. Hamed Aly and M. E. El-Hawary, "Performance Testing of Different Configurations of Hybrid Proposed Models for Solar Radiation Prediction Using WNN and ANN," 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada, 2019, pp. 1-4, doi: 10.1109/CCECE.2019.8861522.

[15] L. Benali, G. Notton, A. Fouilloy, C. Voyant, R. Dizene, Solar radiation forecasting using artificial neural network and random forest methods: application to normal beam, horizontal diffuse and global components, Renew. Energy 132 (2019) 871–884.

[16] M. Fadhil, A. Prastiantono, A. Rahardjo, F. H. Jufri and F. Husnayain, "Solar Irradiance Estimation at Certain Location Using Artificial Neural Network and ASHRAE Clear-Sky Model," 2019 IEEE International Conference on Innovative Research and Development (ICIRD), Jakarta, Indonesia, 2019, pp. 1-5, doi: 10.1109/ICIRD47319.2019.9074735.

[17] K. Basaran, A. Ozçift, ¨ D. Kılınç, A new approach for prediction of solar radiation with using ensemble learning algorithm, Arabian J. Sci. Eng.(Springer Science & Business Media BV) 44 (8) (2019).

[18] C. N. Obiora, A. Ali and A. N. Hassan, "Predicting Hourly Solar Irradiance Using Machine Learning Methods," 2020 11th International Renewable Energy Congress (IREC), Hammamet, Tunisia, 2020, pp. 1-6, doi: 10.1109/IREC48820.2020.9310444.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[19] Z. Pang, F. Niu, Z. O’Neill, Solar radiation prediction using recurrent neural network and artificial neural network: a case study with comparisons, Renew. Energy 156 (2020) 279–289.

[20] A. Kurniawan, E. Shintaku, Estimation of the monthly global, direct, and diffuse solar radiation in Japan using artificial neural network, Int. J. Mach. Learn. Comput. 10 (1) (2020) 253–258.

[21] M. Burhan Uddin Shahin, A. Sarkar, T. Sabrina and S. Roy, "Forecasting Solar Irradiance Using Machine Learning," 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2020, pp. 1-6, doi: 10.1109/STI50764.2020.9350400.

[22] Ümit Ağbulut, Ali Etem Gürel, Yunus Biçen, Prediction of daily global solar radiation using different machine learning algorithms: Evaluation and comparison,Renewable and Sustainable Energy Reviews,Volume 135, 2021, 110114, ISSN 1364-0321, https://doi.org/10.1016/j.rser.2020.110114.

[23] Ho, Yih Hwa and Yew, Poh Leng (2022) Solar irradiance forecasting for Malaysia using multiple regression and artificial neural network. Defence S &T Technical Bulletin, 15 (1). pp. 83-90. ISSN 1985-6571.

[24] N. Mdluli, G. Sharma, K. Akindeji, K. Narayanan and S. Sharma, "Development of Short Term Solar Radiation Forecasting Using AI Techniques," 2022 30th Southern African Universities Power Engineering Conference (SAUPEC), Durban, South Africa, 2022, pp. 1-6, doi: 10.1109/SAUPEC55179.2022.9730779.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[25] O. Bamisile, A. Oluwasanmi, C. Ejiyi, N. Yimen, S. Obiora, Q. Huang, Comparison of machine learning and deep learning algorithms for hourly global/diffuse solar radiation predictions, Int. J. Energy Res. 46 (8) (2022) 10052–10073.

[26] S. Tiwari, R. Sabzehgar and M. Rasouli, "Short Term Solar Irradiance Forecast Using Numerical Weather Prediction (NWP) with Gradient Boost Regression," 2018 9th IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG), Charlotte, NC, USA, 2018, pp. 1-8, doi: 10.1109/PEDG.2018.8447751.

[27] P. K. Ray, A. Bharatee, P. S. Puhan and S. Sahoo, "Solar Irradiance Forecasting Using an Artificial Intelligence Model," 2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP), Hyderabad, India, 2022, pp. 1-5, doi: 10.1109/ICICCSP53532.2022.9862494.

[28] M. Karunanithi, A. A. Sajed Braitea, A. A. Rizvi and T. A. Khan, "Forecasting Solar Irradiance Using Machine Learning Methods," 2023 IEEE 64th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), Riga, Latvia, 2023, pp. 1-4, doi: 10.1109/ITMS59786.2023.10317687.

[29] S. Sperati, S. Alessandrini, P. Pinson, and G. Kariniotakis, “The ‘weather intelligence for renewable energies’ benchmarking exercise on short-term forecasting of wind and solar power generation,” Energies, vol. 8, no. 9, pp. 9594–9619, 2015. doi:10.3390/en8099594.

[30] Meenal, R., and Selvakumar, A. I. (2018). Assessment of SVM, empirical and

اونيورسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ANN based solar radiation prediction models with most influencing input parameters. *Renew. Energy* 121, 324–343. doi: 10.1016/j.renene.2017.12.005

[31] Pang, Z., Niu, F., and O'Neill, Z. (2020). Solar radiation prediction using recurrent neural network and artificial neural network: a case study with comparisons. *Renew. Energy* 156, 279–289. doi: 10.1016/j.renene.2020.04.042.

[32] Shamshirband, S., Mosavi, A., Rabczuk, T., Nabipour, N., and Chau, K. W. (2020). Prediction of significant wave height; comparison between nested grid numerical model, and machine learning models of artificial neural networks, extreme learning and support vector machines. *Eng. Appl. Comput. Fluid Mech.* 14, 805–817. doi: 10.1080/19942060.2020.1773932.

[33] Angra, S., and Ahuja, S. (2017). "Machine learning and its applications: a review," in *Proceedings of the 2017 International Conference On Big Data Analytics and Computational Intelligence, ICBDACI 2017*, (Piscataway, NJ: IEEE), 57–60. doi: 10.1109/ICBDACI.2017.8070809.

[34] E. D. Lopez Izurieta, E. Toapanta Guamanarca, and H. Barbier, "Ionospheric Total Electron Content (TEC) above Ecuador," Journal of Physics: Conference Series, vol. 2238, no. 1, p. 012010, 2022. doi:10.1088/1742-6596/2238/1/012010

[35] N. Yaacob, M. Abdullah, and M. Ismail, 'GPS Total Electron Content (TEC) Prediction at Ionosphere Layer over the Equatorial Region', Trends in Telecommunications Technologies. InTech, Mar. 01, 2010. doi: 10.5772/8474.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[36] Abdullah, “TEC and Scintillation Study of Equatorial Ionosphere: A month campaign over Sipitang and Parit Raja stations, Malaysia,” American Journal of Engineering and Applied Sciences, vol. 2, no. 1, pp. 44–49, 2009. doi:10.3844/ajeas.2009.44.49

[37] Bhattarai, Niraj & Chapagain, N. & Adhikari, Binod.,"Total Electron Content and Electron Density Profile Observations during Geomagnetic Storms using COSMIC Satellite Data",Study of Total Electron Content-TEC and electron density profile during geomagnetic storms,Discovery The International journal. 1979-1996(2018).

[38] G. Seemala, “GPS-Tec Analysis Software,” Academia.edu, https://www.academia.edu/23913567/GPS_TEC_analysis_software (accessed Jan. 9, 2024).

[39] E. K. Makama and H. S. Lim, “Variability and trend in integrated water vapour from era-interim and IGRA2 observations over Peninsular Malaysia,” Atmosphere, vol. 11, no. 9, p. 1012, 2020. doi:10.3390/atmos11091012

[40] P. Yuan et al., “Characterisations of Europe’s integrated water vapour and assessments of atmospheric reanalyses using more than 2 decades of ground-based GPS,” Atmospheric Chemistry and Physics, vol. 23, no. 6, pp. 3517–3541, 2023. doi:10.5194/acp-23-3517-2023

[41] J. Morland and C. Mätzler, “Spatial interpolation of GPS integrated water vapour measurements made in the Swiss alps,” Meteorological Applications, vol. 14, no. 1, pp. 15–26, 2007. doi:10.1002/met.2

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[42] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning representations by back-propagating errors, nature, vol. 323, no. 6088, p. 533, 1986.

[43] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).

[44] A. A. Bataineh and D. Kaur, "A Comparative Study of Different Curve Fitting Algorithms in Artificial Neural Network using Housing Dataset," NAECON 2018 - IEEE National Aerospace and Electronics Conference, Dayton, OH, USA, 2018, pp. 174-178, doi: 10.1109/NAECON.2018.8556738.

[45] M. F. Mller, A scaled conjugate gradient algorithm for fast supervised learning, Neural networks, vol. 6, no. 4, pp. 525533, 1993.

[46] Jin R., Jin S., Feng G. M_DCB: Matlab code for estimating GNSS satellite and receiver differential code biases. GPS Solut. 2012;16:541–548. doi: 10.1007/s10291-012-0279-3.

[47] Li H., Xiao J., Zhu W. Investigation and Validation of the Time-Varying Characteristic for the GPS Differential Code Bias. Remote Sens. 2019;11:428. doi: 10.3390/rs11040428.

[48] Conte J.F., Azpilicueta F., Brunini C. Accuracy assessment of the GPS-TEC calibration constants by means of a simulation technique. J. Geod. 2011;85:707. doi: 10.1007/s00190-011-0477-8.

[49] Sardon E., Rius A., Zarraoa N. Estimation of the transmitter and receiver differential biases and the ionospheric total electron content from Global Positioning System observations. Radio Sci. 1994;29:577–586. doi:

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

10.1029/94RS00449.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
UTeM
اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

APPENDICES

Appendix A

```
import os
import zipfile

# Directory where the zipped folders are located
zipped_folders_directory = 'C:/Users/ACER/Desktop/rename/UTeM GPS Data
2022'  # Replace with the actual path to your zipped folders

# Directory where you want to save the renamed files
output_directory = 'C:/Users/ACER/Desktop/rename/obs_edited'  #
Replace with the desired output directory

# Loop through each zipped folder
for root, _, files in os.walk(zipped_folders_directory):
    for filename in files:
        if filename.endswith(".zip"):
            zip_file_path = os.path.join(root, filename)
            try:
                with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
                    for inner_filename in zip_ref.namelist():
                        if inner_filename.endswith(".22O"):  # Change
".22O" to your actual file extension
                            # Extract the relevant parts of the
filename
                            year_part = inner_filename[6:8]  # Adjust
indices based on your actual filenames
                            extension = inner_filename[-3:]   # Adjust
indices based on your actual filenames

                            # Construct the new filename
                            new_filename =
f"Trim_{year_part}{inner_filename[8:12]}.{extension}"

                            # Extract the file to a temporary
directory
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

```
                            with zip_ref.open(inner_filename) as
source_file:
                                temp_path =
os.path.join(output_directory, new_filename)
                                with open(temp_path, 'wb') as
destination_file:
                                    destination_file.write(source_file
.read())

                                print(f"Renamed file: {inner_filename}
to {new_filename}")
            except zipfile.BadZipFile:
                print(f"Skipping invalid zip file: {zip_file_path}")

print("Files with '.220' extension within valid zip archives renamed
successfully.")
```

اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Appendix B

```
import os
import zipfile
# Directory where the zipped folders are located
zipped_folders_directory = 'C:/Users/ACER/Desktop/rename/UTeM GPS Data
2022'  # Replace with the actual path to your zipped folders
# Directory where you want to save the renamed files
output_directory = 'C:/Users/ACER/Desktop/rename/nav_edited'  #
Replace with the desired output directory

# Loop through each zipped folder
for root, _, files in os.walk(zipped_folders_directory):
    for filename in files:
        if filename.endswith(".zip"):
            zip_file_path = os.path.join(root, filename)
            try:
                with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
                    for inner_filename in zip_ref.namelist():
                        if inner_filename.endswith(".22N"):  # Change
".22N" to your actual file extension
                            # Extract the relevant parts of the
filename
                            year_part = inner_filename[6:8]  # Adjust
indices based on your actual filenames
                            extension = inner_filename[-3:]   # Adjust
indices based on your actual filenames
                            # Construct the new filename
                            new_filename =
f"Trim_{year_part}{inner_filename[8:12]}.{extension}"
                            # Extract the file to a temporary
directory
                            with zip_ref.open(inner_filename) as
source_file:
                                temp_path =
os.path.join(output_directory, new_filename)
                                with open(temp_path, 'wb') as
destination_file:
                                    destination_file.write(source_file
.read())

                                print(f"Renamed file: {inner_filename}
to {new_filename}")
            except zipfile.BadZipFile:
                print(f"Skipping invalid zip file: {zip_file_path}")

print("Files with '.22N' extension within valid zip archives renamed
successfully.")
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Appendix C

```python
import pandas as pd
import os

# Function to extract second column data from .Std files and count rows
def extract_and_count_rows(file_path):
    data = pd.read_csv(file_path, delim_whitespace=True, header=None)
    second_column = data.iloc[:, 1]  # Extracting the second column
    row_count = len(second_column)  # Counting the number of rows
    return row_count

# Path to the folder containing .Std files
folder_path = 'C:/Users/ACER/Desktop/GPS TEC/TEC_output'

# List all files in the folder
file_list = os.listdir(folder_path)

# Filter only .Std files
std_files = [file for file in file_list if file.endswith('.Std')]

# Loop through each .Std file, extract second column data, and count rows
for file in std_files:
    file_path = os.path.join(folder_path, file)
    row_count = extract_and_count_rows(file_path)
    print(f"File: {file}, Rows in Second Column: {row_count}")
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Appendix D

```
import pandas as pd
import os

# Function to extract second column data from .Std files
def extract_second_column(file_path):
    data = pd.read_csv(file_path, delim_whitespace=True, header=None)
    second_column = data.iloc[:, 1]  # Extracting the second column
    return second_column

# Path to the folder containing .Std files
folder_path = 'C:/Users/ACER/Desktop/GPS TEC/TEC_output'

# List all files in the folder
file_list = os.listdir(folder_path)

# Filter only .Std files
std_files = [file for file in file_list if file.endswith('.Std')]

# Create an empty DataFrame to store the second column data
all_data = pd.DataFrame()

# Loop through each .Std file, extract second column data, and append
to the DataFrame
for file in std_files:
    file_path = os.path.join(folder_path, file)
    second_column_data = extract_second_column(file_path)
    all_data = pd.concat([all_data, second_column_data], axis=0,
ignore_index=True)

# Specify the directory where you want to save the output .csv file
output_directory = 'C:/Users/ACER/Desktop/GPS TEC'

# Create the directory if it doesn't exist
os.makedirs(output_directory, exist_ok=True)

# Define the path to save the output .csv file
output_csv_path = os.path.join(output_directory, 'output.csv')

# Write the extracted data into a single .csv file in the specified
directory
all_data.to_csv(output_csv_path, index=False,
header=['Second_Column_Data'])
```

UTeM
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Appendix E

```
import os
import csv
import glob

def count_sixth_column(file_path):
    with open(file_path, 'r') as file:
        lines = file.readlines()
        count = 0
        for line in lines:
            if line.startswith("$TROP"):
                columns = line.strip().split(',')
                if len(columns) >= 6:
                    count += 1
    return count

if __name__ == "__main__":
    folder_path = 'C:/Users/ACER/Desktop/rename/ZTD_stat/*.stat'  # 
Change this to your folder path
    stat_files = glob.glob(folder_path)

    for file in stat_files:
        count = count_sixth_column(file)
        print(f"File: {file} - Sixth column count in $TROP rows: 
{count}")
```

UTeM
اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Appendix F

```
import os
import csv
import glob

def extract_sixth_column(file_path):
    with open(file_path, 'r') as file:
        lines = file.readlines()
        extracted_data = []
        for line in lines:
            if line.startswith("$TROP"):
                columns = line.strip().split(',')
                if len(columns) >= 6:
                    sixth_column = columns[5]
                    extracted_data.append([sixth_column])
    return extracted_data

def save_to_csv(data, file_path, output_folder):
    file_name = os.path.basename(file_path)
    csv_file_path = os.path.join(output_folder,
file_name.replace('.stat', '_extracted.csv'))
    with open(csv_file_path, 'w', newline='') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerows(data)

if __name__ == "__main__":
    input_folder_path =
'C:/Users/ACER/Desktop/rename/in_ztdsec/*.stat'  # Change this to your
input folder path
    output_folder_path = 'C:/Users/ACER/Desktop/rename/out_ztdsec'  #
Change this to your output folder path

    if not os.path.exists(output_folder_path):
        os.makedirs(output_folder_path)

    stat_files = glob.glob(input_folder_path)

    for file in stat_files:
        extracted_data = extract_sixth_column(file)
        if extracted_data:
            save_to_csv(extracted_data, file, output_folder_path)
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Appendix G

```
import os
import pandas as pd

# Define the folder path where your CSV files are located
input_folder_path = 'C:/Users/ACER/Desktop/rename/out_ztdsec'

# Define the folder path where you want to save the new CSV files
output_folder_path = 'C:/Users/ACER/Desktop/rename/out_ztdmin'

# Create the output folder if it doesn't exist
os.makedirs(output_folder_path, exist_ok=True)

# List all files in the input folder
files = os.listdir(input_folder_path)

# Filter only CSV files
csv_files = [file for file in files if file.endswith('.csv')]

# Process each CSV file
for file in csv_files:
    file_path = os.path.join(input_folder_path, file)

    # Read the CSV file into a pandas DataFrame, skip the first row as it doesn't contain headers
    df = pd.read_csv(file_path, header=None)

    # Extract only the data from even rows (2, 4, 6, etc.)
    even_rows_data = df.iloc[1::2]  # Extracts rows starting from index 1, skipping one row

    # Get the file name (without extension)
    file_name, file_extension = os.path.splitext(file)

    # Define the output file path in the output folder
    output_file_path = os.path.join(output_folder_path, f"{file_name}_even_rows.csv")

    # Write the even rows' data to a new CSV file in the output folder
    even_rows_data.to_csv(output_file_path, index=False, header=False)
```

UTeM
اونيۆرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Appendix H

```
import os
import pandas as pd

# Directory containing your CSV files
directory = 'C:/Users/ACER/Desktop/rename/out_ztdmin'

# List all CSV files in the directory
csv_files = [file for file in os.listdir(directory) if
file.endswith('.csv')]

# Initialize an empty list to store data
data = []

# Read each CSV file and append its content to the 'data' list
for file in csv_files:
    file_path = os.path.join(directory, file)
    df = pd.read_csv(file_path, header=None)  # Assuming no header in
each CSV
    data.append(df)

# Concatenate all data into a single DataFrame
merged_data = pd.concat(data, axis=0, ignore_index=True)

# Write the merged data to a new CSV file
merged_data.to_csv('merged_ztd_data.csv', index=False,
header=False)  # Change 'merged_data.csv' to your desired output file
name
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Appendix I

```
% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
% Created 06-Jan-2024 01:01:49
%
% This script assumes these variables are defined:
%
%   Input_new - input data.
%   Target_new - target data.

x = Input_new';
t = Target_new';

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainbr';  % Bayesian Regularization backpropagation.

% Create a Fitting Network
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivision
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse';  % Mean Squared Error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotregression', 'plotfit'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

```
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)

% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA