Machine Learning-based Prediction of Integrated Water Vapor using Meteorological Data

LIM BING SHENG

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Machine Learning-based Prediction of Integrated Water Vapor using Meteorological Data

LIM BING SHENG

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

2025

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Machine Learning-based Prediction of Integrated Water Vapor using Meteorological Data

Sesi Pengajian : 2024/2025

Saya LIM BING SHENG 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**	

Disahkan oleh:

(TANDATANGAN PENULIS)

Alamat Tetap: .....

.....

.....

.....

Tarikh : 23 Januari 2025

(COP DAN TANDATANGAN PENYELIA)

TS. DR. HO YIH HWA
Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer
Universiti Teknikal Malaysia Melaka (UTeM)
Hang Tuah Jaya
76100, Durian Tunggal, Melaka

Tarikh : 23 Januari 2025

*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 Prediction of Integrated Water Vapor using Meteorological Data" is the result of my own work except for quotes as cited in the references.

Signature : ……

Author : Lim Bing Sheng

Date : 23 January 2025

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 : .....

Supervisor Name : Ts. Dr. Ho Yih Hwa

Date : 23 January 2025

DEDICATION

This thesis is devoted to all the experts and investigators who are working to comprehend and lessen the effects of global warming. To our loved ones, whose steadfast encouragement and support have been priceless along this journey. To our lecturers and mentors, whose insight and experience have led and encouraged us. And lastly, to everyone who thinks it's critical to use cutting-edge technology to advance meteorological research in order to improve our knowledge of the planet's weather systems and promote a more sustainable future.

ABSTRACT

This study investigates the use of machine learning models, including Feed-Forward Neural Networks (FFNN), Bagged Trees, and Fine Gaussian Support Vector Machines (SVM), to predict Integrated Water Vapor (IWV) from meteorological data. The models were trained and tested using data from the UTeM FTKEK weather station during Malaysia's northern monsoon season (23 Oct 2019 to 9 Mar 2020). RTKLIB was used to obtain Zenith Total Delay (ZTD) and derive IWV for output of the machine learning models. While FFNN and Bagged Trees excelled in training, Fine Gaussian SVM showed better generalization. Applying moving average filters improved model accuracy by 13%. Additionally, feature selection and time-lag analysis optimized predictions. The study suggests that predicting IWV at weekly intervals yields better results than minute-by-minute forecasts. Future work should focus on expanding the dataset and refining the models for real-world applications.

ABSTRAK

Kajian ini menyiasat penggunaan model pembelajaran mesin, termasuk Feed-Forward Neural Networks (FFNN), Bagged Trees, dan Fine Gaussian Support Vector Machines (SVM), untuk meramalkan Integrated Water Vapor (IWV) daripada data meteorologi. Model-model ini dilatih dan diuji menggunakan data dari stesen cuaca UTeM FTKEK semasa musim monsun utara Malaysia (23 Okt 2019 hingga 9 Mac 2020). RTKLIB digunakan untuk memperoleh Zenith Total Delay (ZTD) dan mengira IWV sebagai output model pembelajaran mesin. Walaupun FFNN dan Bagged Trees menunjukkan prestasi yang baik semasa latihan, Fine Gaussian SVM menunjukkan generalisasi yang lebih baik. Penggunaan penapis purata bergerak meningkatkan ketepatan model sebanyak 13%. Selain itu, pemilihan ciri dan analisis masa-lambat mengoptimumkan ramalan. Kajian ini mencadangkan bahawa meramalkan IWV pada selang masa mingguan menghasilkan keputusan yang lebih baik berbanding ramalan minit demi minit. Kerja masa depan harus memberi tumpuan kepada peluasan set data dan penambahbaikan model untuk aplikasi dunia sebenar.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to all who contributed to the success of this project, "Machine Learning-based Prediction of Integrated Water Vapor using Meteorological Data." I extend my sincerest thanks to my supervisor, Ts. Dr. Ho Yih Hwa, for his invaluable guidance, encouragement, and support throughout this journey. I am also grateful to Universiti Teknikal Malaysia Melaka (UTeM) and the GNSS and weather stations at FTKEK for providing the essential data that made this project possible. A special note of appreciation goes to Profesor Madya Dr. Ir. Syafeeza Binti Ahmad Radzi, Dr. Norhidayah Binti Mohamad Yatim, and Profesor Madya Dr. Soo Yew Guan for their insightful advice, which greatly improved the quality of my project. I would also like to thank my classmates, especially Yeap Sheng Zhuang, and my friends in the same field for their collaborative spirit and support. Last but not least, I am profoundly grateful to my family, especially my parents, for their unconditional love and encouragement, without which none of this would have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

IWV	:	Integrated Water Vapor
GNSS	:	Global Navigation Satellite Systems
GPS	:	Global Positioning System
GLONASS	:	GLObalnaya NAvigatsionnaya Sputnikovaya Sistema
QZSS	:	Quasi-Zenith Satellite System
NavIC	:	Navigation with Indian Constellation
SBAS	:	Satellite Based Augmentation System
MWR	:	Microwave Radiometers
FTIR	:	Fourier-Transform Infrared
AMSU	:	Advanced Microwave Sounding Unit
PBO	:	Plate Boundary Observatory
VBIT	:	Vignan Bharathi Institute of Technology
NASA	:	National Aeronautics and Space Administration
MODIS	:	Moderate Resolution Imaging Spectroradiometer
AOD	:	Aerosol Optical Depth
MAIAC	:	Multi-Angle Implementation of Atmospheric Correction
GSR	:	Global Solar Radiation
IMD	:	India Meteorological Department
RINEX	:	Receiver Independent Exchange Format

RTKLIB	:	Real-Time Kinematic Library
RTK	:	Real-Time Kinematic
PPK	:	Post-Processing Kinematic
PPP	:	Precise Point Positioning
RTCM	:	Radio Technical Commission for Maritime Services
MATLAB	:	Matrix Laboratory
TOW	:	Time of Week
MAF	:	Moving Average Filter
VIF	:	Variance Inflation Factor
FTKEK	:	Fakulti Teknologi Kejuruteraan Elektronik dan Komputer
ZTD	:	Zenith Total Delay
ZHD	:	Zenith Hydrostatic Delay
ZWD	:	Zenith Wet Delay
PWV	:	Precipitable Water Vapor
CWV	:	column water vapor
AI	:	Artificial Intelligence
ML	:	Machine Learning
FFNN	:	Feed-Forward Neural Network
BT	:	Bagged Tree
FGSVM	:	Fine Gaussian SVM
ANN	:	Artificial Neural Networks
NN	:	Neural Network
SVM	:	Support Vector Machines
RF	:	Random Forests
GBDT	:	Gradient Boosting Decision Tree

XGBoost	:	Extreme Gradient Boosting
LightGBM	:	Light Gradient Boosting Machine
RFR	:	Random Forest Regression
XGBR	:	Extreme Gradient Boosting Regression
OE	:	Optimized Ensemble Model
RQ-GPR	:	Rational Quadratic Gaussian Process Regression
CSVM	:	Cubic Support Vector Machine
QSVM	:	Quadratic Support Vector Machine
MAPE	:	Mean Absolute Percentage Error
MAE	:	Mean Absolute Error
RMSE	:	Root Mean Squared Error
MSE	:	Mean Squared Error

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

This chapter will give an introduction of the project. The background, problem statement, objectives and scope of the project are clearly stated in this chapter.

1.1 Background

Weather and Climatological studies are very important in assessing atmospheric conditions like storms and cyclones. Integrated water vapor (IWV) is an important greenhouse gas in the atmosphere responsible for the Earth's radiative balance [1]. Global Navigation Satellite System (GNSS) observations and meteorological data like pressure, temperature, relative humidity and wind speed have been used for monitoring the IWV variability. However, estimating IWV for forecasting applications is complex and costly with a GNSS system. So, this project introduces a methodology to predict IWV using machine learning (ML) techniques. If this project is successful, the measurement of IWV by using satellite can be replaced by machine learning which leads to a cost saving. Meteorological surface data like pressure, temperature, relative humidity and wind speed are given as input to the machine learning models. The IWV values computed from GNSS will be the output of the machine learning models.

1.1.1 GNSS-derived Integrated Water Vapor

GNSS (Global Navigation Satellite System), which includes systems like GPS (United States), Galileo (Europe), and GLONASS (Russia), BeiDou (China), QZSS (Japan), NavIC (India) and SBAS (from many countries) provides valuable data for deriving IWV. These systems transmit signals that are delayed by atmospheric water vapor. By measuring these delays, GNSS can accurately determine IWV along the signal path, offering continuous and high-resolution data [2][3][4].

Integrated Water Vapor (IWV) plays a crucial role in atmospheric studies, climate monitoring and weather forecasting. As a significant greenhouse gas, IWV is essential for understanding precipitation patterns, storms, and cyclones. Accurate measurement

and prediction of IWV are vital for enhancing weather forecasts and climate models [5]. IWV refers to the total amount of water vapor in a vertical column (90°) of the atmosphere, expressed in kilograms per square meter (kg/m²). Figure 1.1 simply shows the concept of IWV.

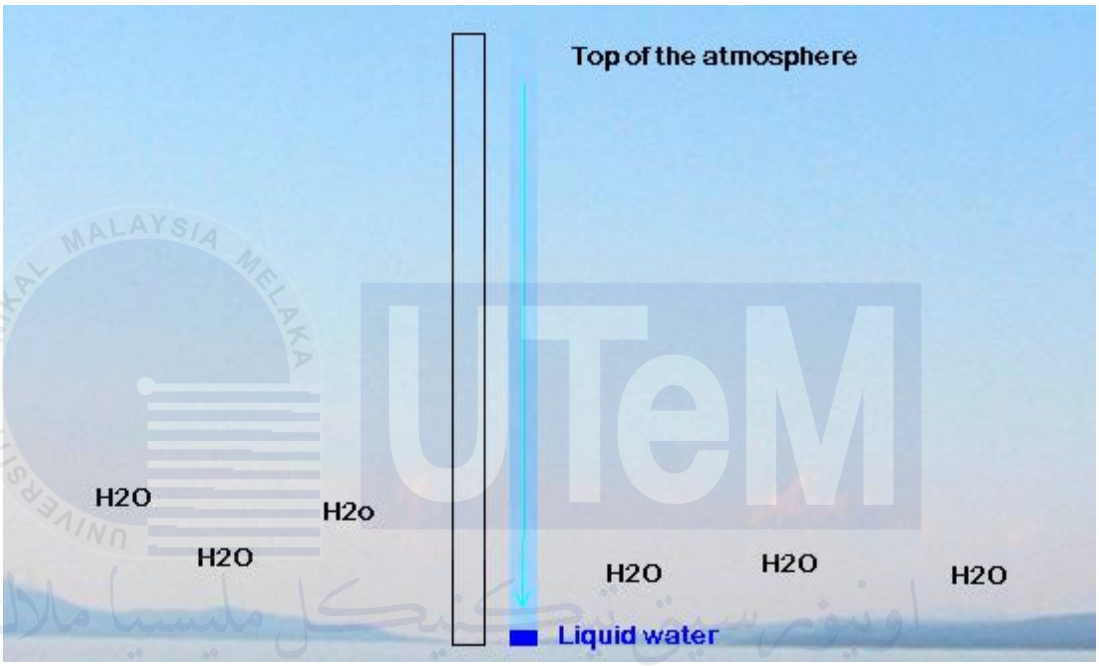


Figure 1.1: Integrated Water Vapor [6]

GNSS-derived IWV data improves understanding and prediction of weather events and has applications in hydrology, agriculture, and environmental monitoring. While traditional methods for measuring IWV face limitations. Some traditional methods are like radiosonde, Ground-Based Microwave Radiometers (MWR), Ground-Based Fourier-Transform Infrared (FTIR) Spectrometers Radiosondes and Satellite-Based Microwave Radiometers. For radiosonde, these are balloon-borne instruments that measure atmospheric parameters, including water vapor. They provide vertical profiles of the atmosphere but are limited by their temporal and spatial coverage, as

they are typically launched only twice a day from specific locations. For Ground-Based Microwave Radiometers (MWR), These instruments measure the natural microwave emissions from atmospheric water vapor. They provide continuous and high-resolution data, but their accuracy can be affected by cloud cover and precipitation. For Ground-Based Fourier-Transform Infrared (FTIR) Spectrometers, these instruments measure the absorption of infrared radiation by water vapor in the atmosphere. They are highly accurate but require clear skies for optimal performance For Satellite-Based Microwave Radiometers, these instruments, such as the Advanced Microwave Sounding Unit (AMSU), measure IWV from space [2]. They provide global coverage but have lower spatial resolution compared to ground-based methods. While the advantages of GNSS-derived IWV is it can retrieve IWV by measuring the absorption of specific wavelengths of light by water vapor without delay and without clear skies requirement [7].

1.1.2 Machine Learning in Meteorological Predictions

Machine learning (ML) has emerged as a powerful tool in various scientific and engineering fields, including meteorology. ML techniques can process large datasets, identify patterns, and make predictions with high accuracy [8]. In meteorology, ML algorithms such as Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Random Forests (RF) have been applied to predict various atmospheric phenomena, including IWV [9][10].

The use of ML in predicting IWV involves training models on historical meteorological data, such as temperature, pressure, and relative humidity. These models can then predict IWV based on new input data, providing a faster and

potentially more accurate alternative to traditional methods. The implementation of ML techniques for IWV prediction is particularly beneficial in regions where real-time and precise weather data is critical for disaster management, agriculture, and resource management [11][12].

Research has demonstrated the effectiveness of ML in enhancing the accuracy of meteorological predictions. For instance, studies have successfully used ML algorithms to predict IWV with high precision, significantly improving the reliability of weather forecasts. This approach not only reduces the dependency on traditional methods but also enables continuous monitoring and real-time data processing, which are crucial for timely decision-making.

1.2 Problem Statement

Global navigation satellite system GNSS, like GPS is not just for navigation. It's also used in climate studies to estimate water vapor in the atmosphere. This helps understand climate patterns, improve weather forecasts, and monitor long-term atmospheric changes. For places like Malaysia, alternative techniques like machine learning can complement GNSS for better accuracy in estimating water vapor. Since the demand for accurate and real-time weather services has increased, traditional methods such as radiosondes, water vapor radiometers, and solar photometers cannot continuously estimate water vapor with high accuracy and time resolution [13]. While Machine learning techniques who solve the demand by effectively implemented to develop models catering to the needs of predicting various meteorological phenomena that occur in nonlinear combinations of several processes in the Earth's atmosphere [1]. Nevertheless, these techniques warrant the knowledge of accurate IWV with the

shortest possible latency [11]. The better result comes with the ensemble model adopts NN as the meta-learning algorithm to optimally combine the predictions generated by the base models of RF, GBDT, XGBoost, LightGBM, and NN [14]. Addressing that, the main aim of this project is to create a model that critically predicts IWV using meteorological data, pressure, temperature and humidity as accurate as possible in term of more constrain area and longer prediction time.

1.3 Objectives

The primary objective of this research is to develop several machine learning models by supervised learning using meteorological data as input and IWV as output. Specifically, the research aims to achieve the following objectives:

i) To derive IWV from GPS data.

ii) To develop various machine learning models capable of predicting IWV from meteorological data.

iii) To validate and evaluate the developed machine learning models.

1.4 Scope of the Project

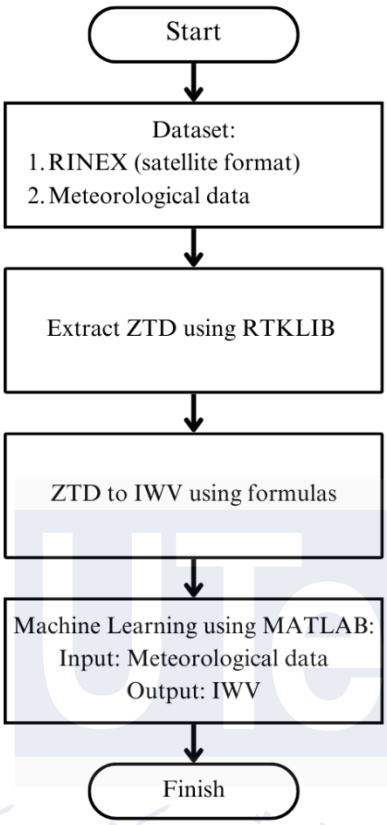


Figure 1.2: Flowchart of The Project

Figure 1.2 is the flowchart of the project. The dataset obtained is a raw satellite data in RINEX format that consist of .O (observation file from ground station), .N (navigation file from GPS satellite) and .G (navigation file from GLONASS) and a meteorological data in Excel such as temperature, pressure, relative humidity, wind speed, solar irradiance and rain rate. Both dataset comprising half year of GNSS data from 23/10/2019 to 09/03/2020 obtained from Fakulti Teknologi Kejuruteraan Elektronik dan Komputer, Faculty of Electronics and Computer Technology and Engineering (FTKEK) weather station (Melaka, Malaysia), location (2.31403,102.31851) FTKEK. The two datasets are obtained at the same location. The

RINEX files contain information with 30 seconds interval. The meteorological data contains information with 1 minute interval. This project necessitates the utilization of software for implementation. Specifically, RTKLIB will be employed to extract relevant data from raw satellite data. RTKLIB is an open-source program package tailored for GNSS positioning. Subsequently, the software MATLAB will be utilized for training and analyzing the predictive models. These models are supervised learning models and so their input will be meteorological data, and output will be the derived IWV. The flow until IWV is obtained fulfills objective 1. The development, validation and evaluation of various machine learning fulfills objectives 2 and 3.

CHAPTER 2

BACKGROUND STUDY

This chapter will cover the theory of GPS broadcast signals to integrated water vapor, machine learning models, background research and literature evaluations of earlier study papers about the project.

2.1 Theory

2.1.1 GPS Broadcast Signals to Integrated Water Vapor (IWV)

GPS receivers calculate atmospheric delay, including Zenith Total Delay (ZTD), by analyzing the signals transmitted by GPS satellites. Each satellite broadcasts signals on two primary frequencies, L1 (1575.42 MHz) and L2 (1227.60 MHz). These signals carry unique codes and timestamps generated by highly accurate atomic clocks onboard the satellites. The receiver detects these signals and records their arrival times, enabling the calculation of the signal's travel time from satellite to receiver [15][16][17].

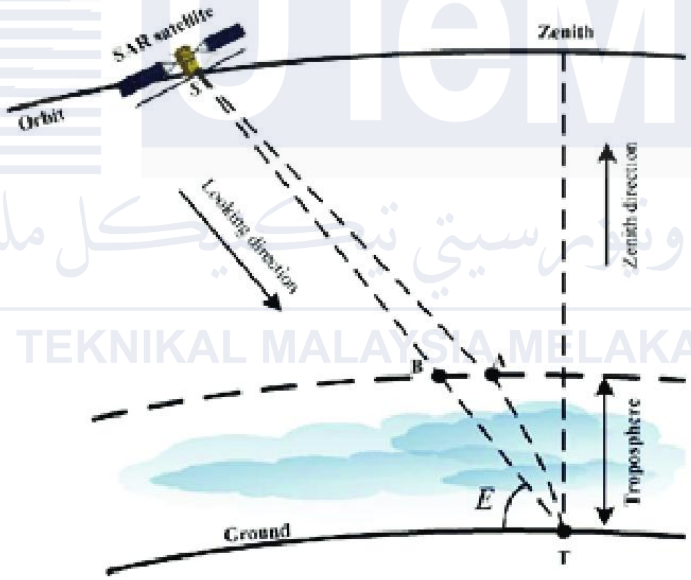


Figure 2.1: GPS Broadcast Signals [18]

The receiver determines the pseudo-range by multiplying the signal travel time by the speed of light. However, this measurement is not the true geometric distance because it includes delays caused by the atmosphere, as well as clock biases in both the satellite and receiver. The ionosphere introduces frequency-dependent delays,

which dual-frequency receivers correct by comparing the L1 and L2 signals. Single-frequency receivers, on the other hand, rely on ionospheric correction models provided in the satellite's navigation message [19][20][21][22].

After correcting for ionospheric delays, the remaining delay is due to the troposphere, which consists of the Zenith Hydrostatic Delay (ZHD) and Zenith Wet Delay (ZWD). Equation (2.1) shows the relationship between ZTD, ZHD and ZWD. The ZHD, caused by dry gases in the atmosphere, is predictable from surface pressure measurements. The ZWD, caused by atmospheric water vapor, is more variable and is obtained by subtracting the ZHD from the total delay [23].

$$ZTD = ZHD + ZWD \tag{2.1}$$

The signal path from a satellite to the receiver is slanted due to the satellite's position in the sky. To convert these slant delays (not 90°) into a Zenith Total Delay (ZTD) (90°), the receiver applies a mapping function that accounts for the satellite's elevation angle. This process ensures that the derived ZTD accurately represents the atmospheric delay directly above the receiver [24][25][26]. Equation (2.2) is about the relationship between slanted and zenith delay [27].

$$T_{Slant} = T_{hydro} \cdot M_{hydro}(\theta) + T_{Wet} \cdot M_{Wet}(\theta) \tag{2.2}$$

- T_{Slant}: total delay dependent on elevation
- T_{hydro}: hydrostatic delay in zenith direction
- T_{Wet}: wet delay in zenith direction
- M(θ): mapping function ($M_{hydro}(\theta) > M_{Wet}(\theta)$)
- θ: Satellite elevation angle

Finally, satellite clock corrections, which are included in the signal's navigation message, are applied to account for minor inaccuracies in the satellite's atomic clock. Receiver clock bias is estimated and corrected as part of the positioning calculation, which requires signals from at least four satellites. With these corrections, the receiver isolates the ZTD, enabling precise atmospheric delay measurements that are essential for applications like weather monitoring and Integrated Water Vapor (IWV) estimation [28][29].

As radio waves traverse the Earth's atmosphere, they experience delay due to refraction, primarily caused by water vapor, a critical component of the troposphere. Analyzing water vapor in detail yields more accurate precipitation estimates, thereby enhancing climate change analysis, given water vapor's significant influence on the Earth's radiation budget and temperature [1]. The conventional approach to extracting integrated water vapor (IWV) from Zenith Total Delay (ZTD) relies on fundamental physical principles. Typically, zenith wet delays (ZWD) are initially separated from ZTD by subtracting zenith hydrostatic delays (ZHD), then converted into IWV using a dimensionless constant of proportionality as shown in equation 2.3 [14]. In case you get confused in some research papers, Integrated Water Vapor (IWV) and Precipitable Water Vapor (PWV) shared identical information. Their sole distinction lies in PWV's adjustment for water density, showing it expressed in millimeters as a unit [30]. Equation 2.3 and equation 2.4 shows the difference between IWV and PWV. IWV differs PWV by dividing by density of water, $\rho = 1000\, kgm^{-3}$ [31][32]. In conclusion, it has no difference except for the definition and units.

$$IWV(kgm^{-2}) = \frac{ZWD \cdot 10^6}{\left(\left(\frac{k_3}{T_m}\right) + k_2'\right) R_V} \tag{2.3}$$

$$PWV(mm) = \frac{ZWD \cdot 10^6}{\rho\left(\left(\frac{k_3}{T_m}\right) + k_2'\right) R_V} \tag{2.4}$$

- Constant, $k_3 = 3.776 \times 10^5 K^2/mbar$
- Constant, $k_2' = 17\ K/mbar$
- Specific gas constant of water vapor, $R_V = 461.5\ J/(kg \cdot K)$
- Water density, $\rho = 1000\ kgm^{-3}$

2.1.2 Machine Learning Models

Several studies have leveraged artificial intelligence (AI) techniques such as feed-forward neural network, bagged trees and fine Gaussian support vector machine (SVM) to predict Integrated Water Vapor (IWV). These AI models utilize ground parameters such as temperature, pressure, and humidity as inputs to train the predictive models [30].

A feed-forward neural network (FFNN) is a type of artificial neural network where connections between nodes do not form cycles. It consists of an input layer, one or more hidden layers, and an output layer. Data flows in one direction, from inputs to outputs, without looping back. Each layer has neurons that apply a weighted sum and an activation function (e.g., ReLU, sigmoid) to model complex, non-linear relationships. Feed-forward networks are widely used for regression and classification tasks and are trained using algorithms like backpropagation. The Levenberg-Marquardt Algorithm (trainlm) is a powerful optimization method used in MATLAB

to train feed-forward neural networks, particularly for regression tasks. It combines the speed of the Gauss-Newton method with the stability of gradient descent, providing efficient error minimization. By iteratively updating weights and biases to reduce the mean squared error, trainlm adapts its approach based on the proximity to the optimal solution, ensuring faster convergence. While it is highly accurate and suitable for medium-sized datasets, its memory requirements can be high, making it less ideal for very large datasets [33].

Bagged trees, also known as bootstrap-aggregated decision trees, are an ensemble learning method. The algorithm creates multiple decision trees by training each one on a random bootstrap sample of the training data. Predictions are made by averaging the outputs of the individual trees for regression or using a majority vote for classification. While in this project it is uses for regression. Bagging reduces overfitting and improves model stability, making it effective for handling noisy datasets and complex relationships [34].

The fine Gaussian Support Vector Machine (SVM) is a variant of SVM that uses the Gaussian or Radial Basis Function kernel. It maps data into a higher-dimensional space where it can separate classes or fit a regression model with a fine-tuned decision boundary. The "fine" aspect refers to the small kernel scale used, which creates a more localized model with high sensitivity to data variations. SVMs work by finding a hyperplane that maximizes the margin between classes or minimizes the regression loss [35].

2.2 Literature Review

Several research is done in different areas contributes to a more completeness of studies. A research related to GNSS-derived precipitable water vapor is conduct at the Plate Boundary Observatory (PBO) network stations that were launched in 2008 for 3D strain monitoring in North America and Alaska [13]. Another study about prediction of integrated water vapor using a machine learning technique uses two and half years (June 2018 - December 2020) of GNSS receiver measurements made at Gadanki (13.4593 N, 79.1684 E) [11]. While another similar study also used six-year rainfall data in South Tengerang, Indonesia, Suparta, and Samah (2020) predicted rainfall using the ANFIS time-series technique with 80% data validity [1].

While in machine learning, in an article of Izanlou et al. (2023) state that two machine learning methods, Random Forest Regression (RFR) and Extreme Gradient Boosting Regression (XGBR), are used to estimate PWV in the American region. Random forest is an ensemble learning method that can be used for regression or classification. The XGBR model was developed by Chen and Guestrin and is an advanced and popular algorithm used in ML [13]. Each ML algorithm possesses strengths and weaknesses. For example, the Neural Network (NN) is adapted in the nonlinear mapping of high-dimensional data but with poor interpretability, while the tree ensemble methods like GBDT and RF shows good interpretability and provide good prediction with smaller data set. Fortunately, the stacked ensemble algorithm can combine information from multiple heterogeneous models to acquire better predictive performance and robustness than any constituent base models [14].

There is a comparison or reference in terms of location and study duration from others. For example, in this study, Nirmala Bai Jadala el at.,2023 employs IWV time

series data from weather stations located at two distinct latitudes: VBIT, Hyderabad (India), and PWVUO station, Oregon (US). The datasets consist of GPS-derived IWV and meteorological data such as pressure (P), temperature (T), and relative humidity (RH) from the year 2014 for VBIT station and from 2020 for PWVUO station. Five machine learning algorithms were used: Optimized Ensemble (OE) model, Rational Quadratic Gaussian Process Regression (RQ-GPR), Neural Networks (NN), Cubic Support Vector Machine (CSVM), and Quadratic Support Vector Machine (QSVM). The analysis revealed that the OE model had the highest correlation coefficient of 99% at VBIT and 88% at PWVUO. Additionally, the performance metrics for the OE model at VBIT included a mean absolute error (MAE) of 0.64 kg/m², mean absolute percentage error (MAPE) of 3.80%, and root mean squared error (RMSE) of 0.94 kg/m². At PWVUO, the OE model achieved an MAE of 1.91 kg/m², MAPE of 11.76%, and RMSE of 1.97 kg/m². These results indicate that the OE model outperformed the other models in terms of accuracy and reliability [12].

As stated in the paper of Just, A. C. el at., 2020, the Multi-Angle Implementation of Atmospheric Correction (MAIAC) algorithm provides column water vapor (CWV) data at a 1 km resolution from NASA's MODIS instruments aboard the Aqua and Terra satellites. This study demonstrates the use of the extreme gradient boosting (XGBoost) algorithm to enhance the estimation of MAIAC aerosol optical depth (AOD) and CWV. The XGBoost model, which incorporates nine features derived from land use terms, date, and ancillary variables, was validated using a spatiotemporal cross-validation approach to avoid overfitting. The model significantly reduced the RMSE by 26.9% for the Terra dataset and 16.5% for the Aqua dataset in the Northeastern USA from 2000 to 2015. Further validation at independent SuomiNet GPS network stations showed RMSE reductions of 19.7% for Terra and 9.5% for Aqua. These

findings suggest that machine learning algorithms like XGBoost can effectively correct measurement errors and improve satellite-derived CWV data [36].

This research from R. Meenal el at., 2021 focuses on predicting global solar radiation (GSR) and wind speed for Tamil Nadu, India, using a random forest ML model. The model was validated with measured wind and solar radiation data collected from the India Meteorological Department (IMD) in Pune. The random forest model's performance was compared to statistical regression models and support vector machine (SVM) models. The random forest model exhibited superior accuracy, with a minimum mean squared error (MSE) of 0.750 and an R^2 score of 0.97. These results indicate that the random forest model provides more accurate weather predictions compared to regression and SVM models, highlighting its potential to reduce the dependency on expensive measuring instruments for acquiring solar radiation and wind speed data [37].

In conclusion, some machine learning models have been considered for this project after this literature review. This project is to investigate the modelling method to improve IWV forecasting models and contribute to better integrated water vapor prediction by using meteorological data in tropical regions from FTKEK satellite (Melaka, Malaysia), location (2.31403,102.31851) with half year dataset from 23/10/2019 to 09/03/2020.

CHAPTER 3

METHODOLOGY

This chapter will discuss on steps involve for completing the project. The software, datasets, methods and formulas used are introduced in this chapter. There are several steps from obtaining ZTD, deriving IWV, Pearson correlation and VIF data analysis, machine learning data split and evaluation metrics until moving average filter.

3.1 Project Flowchart

Figure 3.1 shows the flowchart of the project.

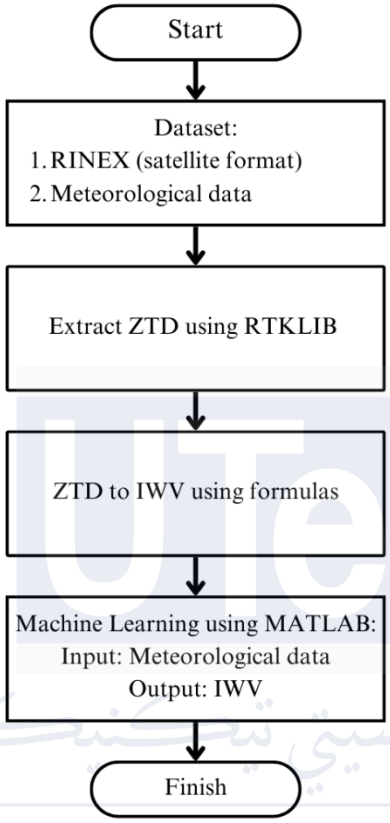


Figure 3.1: Flowchart of The Project

The workflow for deriving and predicting Integrated Water Vapor (IWV) begins with the collection of two datasets, raw GNSS data in RINEX format and meteorological data. The RINEX data consists of observation files (.O), GPS navigation files (.N), and GLONASS navigation files (.G), while the meteorological data includes parameters such as temperature, pressure, relative humidity, wind speed, solar irradiance, and rain rate. These datasets are collected for the period from October 23, 2019, to March 9, 2020, at the FTKEK weather station in Melaka, Malaysia.

The next step involves using RTKLIB, an open-source GNSS data processing software. It is used to extract the Zenith Total Delay (ZTD) from the raw satellite data. Once the ZTD is obtained, it is converted into IWV using established formulas. This conversion utilizes some meteorological data (as it is in the same location) to separate ZTD into Zenith Hydrostatic Delay (ZHD) and Zenith Wet Delay (ZWD) and subsequently calculates IWV from ZWD.

Finally, MATLAB is employed to develop various supervised machine learning models for IWV prediction. In this step, the meteorological data serves as the input to the models, while the IWV derived from ZTD acts as the output. The models are trained and analyzed to improve the prediction accuracy of IWV, ultimately providing valuable insights for atmospheric studies and weather forecasting.

3.2 Dataset

The dataset obtained is a raw satellite data in RINEX format that consist of .O (observation file contain information of base station), .N (navigation file contain information of GPS satellites) and .G (navigation file from GLONASS) and a meteorological data in Excel such as temperature, pressure, relative humidity, wind speed, solar irradiance and rain rate. Both dataset comprising half year of GNSS data from 23/10/2019 to 09/03/2020 obtained from FTKEK weather station (Melaka, Malaysia), location (2.31403,102.31851) FTKEK. The two datasets are obtained at the same location. The RINEX files contain information with 30 seconds interval. The meteorological data contains information with 1 minute interval.

3.3 Software

3.3.1 RTKLIB

Figure 3.2: RTKLIB

RTKLIB (Real-Time Kinematic Library) is a versatile open-source software library designed for processing Global Navigation Satellite System (GNSS) data, created by Tomoji Takasu. It supports various positioning techniques, including Real-Time Kinematic (RTK), Post-Processing Kinematic (PPK), Precise Point Positioning (PPP), and Static GNSS positioning. Its adaptability, open-source accessibility, and compatibility with multiple GNSS systems—such as GPS, GLONASS, Galileo, and BeiDou—make it a valuable tool for researchers and engineers in satellite navigation and geospatial science.

The RTKLIB toolbox includes several utilities tailored for GNSS data handling and analysis. Some key components include [38]:

- RTKPOST: a post-processing tool for computing precise positions.
- RTKNAVI: a real-time navigation application.
- RTKCONV: a conversion tool for pre-processing raw GNSS data into RINEX and other formats.

- RTKPLOT: provides visual analysis of positioning accuracy, satellite visibility, and other metrics.
- SBAS: precise ephemeris data corrections.
- RTCM protocols: further enhancing positioning accuracy.

A notable feature of RTKLIB is its compatibility with Python, which enhances the analysis and visualization of tropospheric delay data. This integration allows researchers to automate processing works and suitable for larger datasets which then leading to more convenient customizations. Moreover, RTKLIB's ability to compute Zenith Total Delay (ZTD) values makes it a critical tool for atmospheric studies, such as Integrated Water Vapor (IWV) estimation. However, the RTKPOST is widely used in this project.

3.3.2 MATLAB

Figure 3.3: MATLAB

MATLAB is a high-level programming language and environment designed for numerical computing and data analysis. It provides a comprehensive platform for

performing complex mathematical operations, visualizing data, and developing algorithms. For machine learning, MATLAB includes specialized toolboxes such as the Deep Learning Toolbox and Regression Learner app, which simplify the process of building and training models. These tools allow you to input data, preprocess it, and apply machine learning algorithms with minimal coding. MATLAB offers functionalities for training models like feed-forward neural networks, decision trees, and support vector machines, making it a powerful tool for tasks such as predicting IWV from meteorological data. The environment also supports model evaluation, hyperparameter tuning, and performance visualization, ensuring a streamlined workflow for machine learning projects.

3.3.3 Python

Figure 3.4: Python

Python, with tools like Pandas, Matplotlib, and Seaborn, is great for working with large datasets. Pandas makes it easy to arrange and clean dates, fix missing values by filling them with previous data, and change timestamps into a standard format. For plotting and visualizing data, library such as Matplotlib helps create graphs that make it easier to spot issues. Python also helps clear invalid dates and ensures all expected

dates are included, even if some data is missing. These tools make Python an excellent choice for managing and analyzing time-based data.

3.4 Obtaining ZTD

3.4.1 Combination of All Dates of RINEX file

RINEX (Receiver Independent Exchange Format) is a standardized data format used in the field of satellite navigation and positioning. It was developed to enable the sharing and processing of raw data from GNSS (Global Navigation Satellite Systems) receivers, regardless of the manufacturer or model. It supports data from multiple GNSS constellations, such as GPS, GLONASS, Galileo, and BeiDou, enabling precise and versatile positioning solutions. RINEX files are widely used in applications such as geodesy, surveying, and precise positioning. Some files is in the raw format such as .o files (observation files), .n files (GPS navigation files), .g files (GLONASS navigation files), .e files (Galileo navigation files), .q files (QZSS navigation files), .i files (NAVIC navigation files), .s files (SBAS navigation files), .p or .nav files (combined or multi-GNSS navigation files) and .leo files (LEO satellites navigation files). Other files like .sp3 (precise ephemeris) and .clk (precise clock corrections) are auxiliary products generated by external analysis centers and are not part of raw GNSS data collected from the receiver. They are used for assisting the raw data files in RTKLIB to obtain more accurate results [39]. However, the .O file and .N file (GPS) .G file (GLONASS) is provided and only .O file and .N file are used in this project because we use only GPS to convert ZTD.

The .O file in RINEX format contains the observations made by the GNSS receiver. This file records various types of measurements, including pseudorange, carrier phase,

doppler shift, and signal-to-noise ratio (SNR), collected from the satellites being tracked. These observations are made at specific timestamps and are associated with each tracked satellite in the GNSS constellation. Each entry in the .O file typically includes information about the satellite's PRN (Pseudo-Random Noise) code, signal type, and the measurement values, which are crucial for precise positioning tasks like Real-Time Kinematic (RTK) or Post-Processed Kinematic (PPK) solutions [39].

The .N file in RINEX format contains the navigation data provided by the GNSS satellites, which includes the satellite ephemerides and other related information required to compute the satellite's position at any given time. This file provides data such as satellite clock corrections, orbital parameters, almanac, and health status for each satellite. The .N file is essential for performing precise GNSS positioning, as it helps interpret the satellite signals received in the .O file. It allows for the calculation of the satellite's location and time adjustments, facilitating accurate positioning calculations by the receiver [39].

The original files contain both 3 .O, .N and .G in one zip file for 1 day. There is a total of 139 days, which contributes to 139 zip files in total. To avoid discontinuity of ZTD from day to day, the .O files of each day need to be combined by removing the header of the consecutive days, leaving the header of the first day. The same goes to .N files. While not doing for .G files as we don’t needed for it. All the operation is done by using python. Figure 3.5 and Figure 3.6 shows the example of the header and information contained of .O and .N files respectively.

```
     2.11           OBSERVATION DATA    M (MIXED)           RINEX VERSION / TYPE
NetR9 5.15          Receiver Operator   20191023 000000 UTC PGM / RUN BY / DATE
UTEM                                                        MARKER NAME
0002                                                        MARKER NUMBER
Micheal W. Walsh    SERIS                                   OBSERVER / AGENCY
5702R51131          Trimble NetR9       5.15                REC # / TYPE / VERS
                    TRM57971.00     NONE                    ANT # / TYPE
 -1359662.8397  6226312.4223   255808.1633                 APPROX POSITION XYZ
        0.0000        0.0000        0.0000                 ANTENNA: DELTA H/E/N
     1     1                                                WAVELENGTH FACT L1/2
     8    C1    L1    S1    P1    C2    L2    S2    P2     # / TYPES OF OBSERV
    30.000                                                  INTERVAL
  2019    10    23     0     0    0.0000000     GPS         TIME OF FIRST OBS
GLONASS C/A & P PHASE MATCH: PHASE SHIFTS REMOVED           COMMENT
                                                            END OF HEADER
 19 10 23  0  0  0.0000000  0 18G 2G28G19R23G 5G 9R13G 6G13G 4R14R 2 0.000000000
                               G30R24R 3R 1G12G17
  23427518.336 7 123112423.292 7        43.300
  95931752.04048        31.600    23427516.47748
  21298565.578 8 111924526.502 8        45.100
  87214063.45249        34.100    21298565.45749
  22285445.086 7 117110818.983 7        43.500
  91255152.62149        34.000    22285444.02749
  23686497.195 7 126706769.565 7        42.300    23686496.465 7  23686495.746 3
  98549723.913 3        33.400
  21148587.398 9 111136612.807 9        48.800
```

Figure 3.5: Observation File

```
     2.11           N: GPS NAV DATA                         RINEX VERSION / TYPE
NetR9 5.15          Receiver Operator   23-OCT-19 00:00:00  PGM / RUN BY / DATE
     .1211D-07   .0000D+00  -.1192D-06   .0000D+00          ION ALPHA
     .9421D+05   .0000D+00  -.1966D+06   .0000D+00          ION BETA
     .186264514923D-08  .444089209850D-14   503808     2076 DELTA-UTC: A0,A1,T,W
    18                                                      LEAP SECONDS
                                                            END OF HEADER
 2 19 10 23  0  0  0.0 -.331458635628D-03 -.784439180279D-11  .000000000000D+00
     .690000000000D+02 -.217187500000D+02  .461304929488D-08  .256332083358D+01
    -.117532908916D-05  .192644156050D-01  .242143869400D-05  .515356111908D+04
     .259200000000D+06  .165775418282D-06  .601432090964D+00 -.912696123123D-07
     .956560435942D+00  .325468750000D+03 -.170629506307D+01 -.813033866124D-08
     .635740766869D-10  .100000000000D+01  .207600000000D+04  .000000000000D+00
     .240000000000D+01  .000000000000D+00 -.176951289177D-07  .690000000000D+02
     .252018000000D+06  .400000000000D+01
28 19 10 22 23 59 44.0  .758070964366D-03 -.147792889038D-11  .000000000000D+00
     .600000000000D+01  .386250000000D+02  .407338395851D-08 -.104052254176D+01
     .201351940632D-05  .191135313362D-01  .993534922600D-05  .515371279907D+04
     .259184000000D+06 -.352039933205D-06 -.133547150450D+01  .931322574615D-07
     .980076206234D+00  .202968750000D+03 -.144933575826D+01 -.810676625078D-08
    -.444661379074D-09  .100000000000D+01  .207600000000D+04  .000000000000D+00
     .240000000000D+01  .000000000000D+00 -.111758708954D-07  .600000000000D+01
```

Figure 3.6: Navigation File

After combining the output will be a single .O file and a single .N file which contain their information from 23/10/2019 to 09/03/2020 respectively. The next step is to input them into RTKLIB to obtain a .pos (positioning solution file) and .pos.stat (its solution status file).

3.4.2 RTKPOST Settings

The setting is crucial in obtaining the accurate ZTD. Figure 3.7, Figure 3.8 and Figure 3.9 show the settings in rtkpost.exe for this project. The important parameters will be mentioned in the following paragraphs.

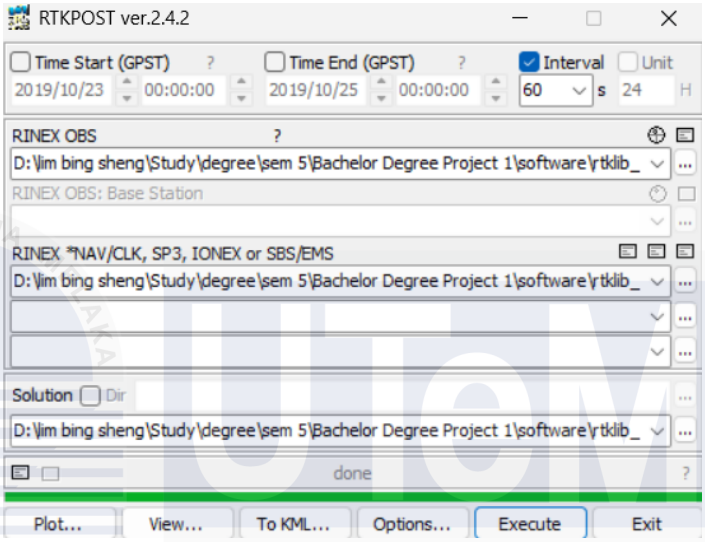


Figure 3.7: RTKPOST Interface

The interval is ticked and set to 60 seconds. Choose the path for .O, .N and output files.

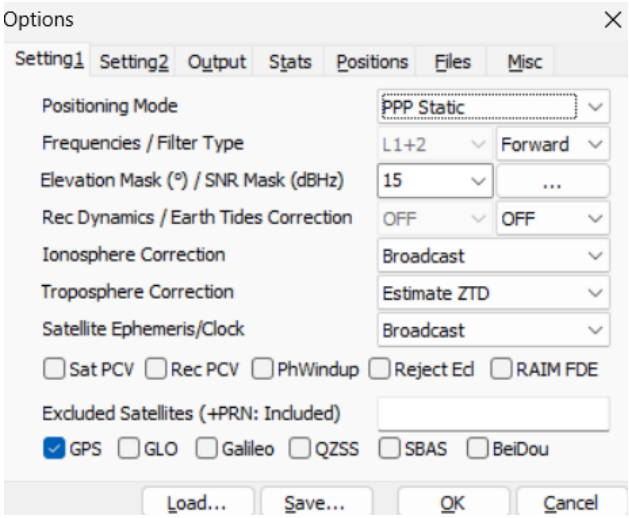
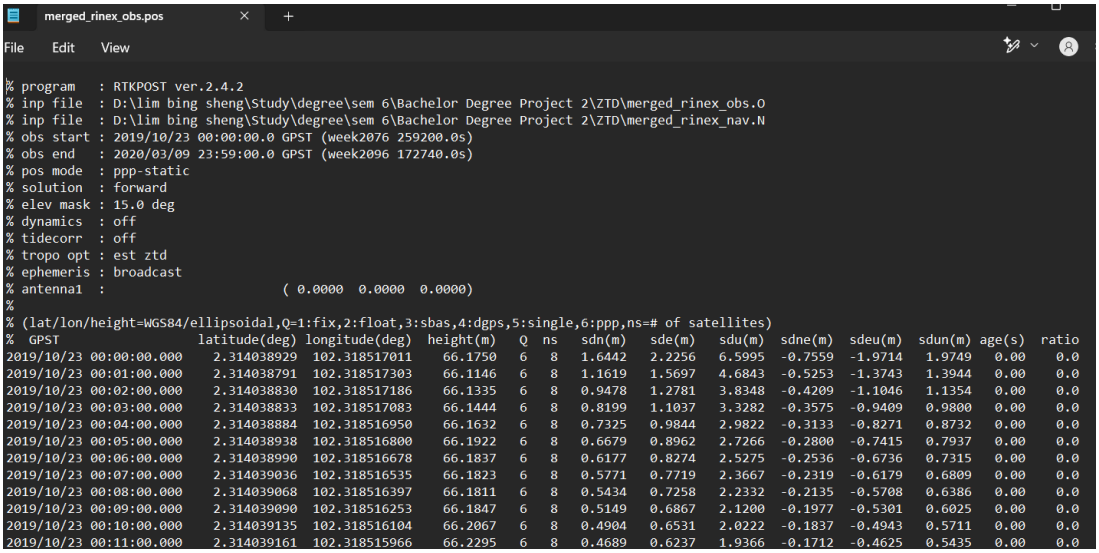


Figure 3.8: RTKPOST Setting 1

The positioning mode is set to PPP static which means for a static receiver at the FTKEK weather station at location (2.31403,102.31851) which achieves a high-quality positioning. Precise Point Position (PPP) is crucial for obtaining ZTD. To obtain ZTD, it needs a precise point position and Q=6.



merged_rinex_obs.pos

File Edit View

```
% program   : RTKPOST ver.2.4.2
% inp file  : D:\lim bing sheng\Study\degree\sem 6\Bachelor Degree Project 2\ZTD\merged_rinex_obs.O
% inp file  : D:\lim bing sheng\Study\degree\sem 6\Bachelor Degree Project 2\ZTD\merged_rinex_nav.N
% obs start : 2019/10/23 00:00:00.0 GPST (week2076 259200.0s)
% obs end   : 2020/03/09 23:59:00.0 GPST (week2096 172740.0s)
% pos mode  : ppp-static
% solution  : forward
% elev mask : 15.0 deg
% dynamics  : off
% tidecorr  : off
% tropo opt : est ztd
% ephemeris : broadcast
% antenna1  :                    ( 0.0000  0.0000  0.0000)
%
% (lat/lon/height=WGS84/ellipsoidal,Q=1:fix,2:float,3:sbas,4:dgps,5:single,6:ppp,ns=# of satellites)
%  GPST                  latitude(deg) longitude(deg)  height(m)   Q  ns   sdn(m)   sde(m)   sdu(m)  sdne(m)  sdeu(m)  sdun(m) age(s)  ratio
2019/10/23 00:00:00.000   2.314038929  102.318517011    66.1750   6   8   1.6442   2.2256   6.5995  -0.7559  -1.9714   1.9749   0.00    0.0
2019/10/23 00:01:00.000   2.314038791  102.318517303    66.1146   6   8   1.1619   1.5697   4.6843  -0.5253  -1.3743   1.3944   0.00    0.0
2019/10/23 00:02:00.000   2.314038830  102.318517186    66.1335   6   8   0.9478   1.2781   3.8348  -0.4209  -1.1046   1.1354   0.00    0.0
2019/10/23 00:03:00.000   2.314038833  102.318517083    66.1444   6   8   0.8199   1.1037   3.3282  -0.3575  -0.9409   0.9800   0.00    0.0
2019/10/23 00:04:00.000   2.314038884  102.318516950    66.1632   6   8   0.7325   0.9844   2.9822  -0.3133  -0.8271   0.8732   0.00    0.0
2019/10/23 00:05:00.000   2.314038938  102.318516800    66.1922   6   8   0.6679   0.8962   2.7266  -0.2800  -0.7415   0.7937   0.00    0.0
2019/10/23 00:06:00.000   2.314038990  102.318516678    66.1837   6   8   0.6177   0.8274   2.5275  -0.2536  -0.6736   0.7315   0.00    0.0
2019/10/23 00:07:00.000   2.314039036  102.318516535    66.1823   6   8   0.5771   0.7719   2.3667  -0.2319  -0.6179   0.6809   0.00    0.0
2019/10/23 00:08:00.000   2.314039068  102.318516397    66.1811   6   8   0.5434   0.7258   2.2332  -0.2135  -0.5708   0.6386   0.00    0.0
2019/10/23 00:09:00.000   2.314039090  102.318516253    66.1847   6   8   0.5149   0.6867   2.1200  -0.1977  -0.5301   0.6025   0.00    0.0
2019/10/23 00:10:00.000   2.314039135  102.318516104    66.2067   6   8   0.4904   0.6531   2.0222  -0.1837  -0.4943   0.5711   0.00    0.0
2019/10/23 00:11:00.000   2.314039161  102.318515966    66.2295   6   8   0.4689   0.6237   1.9366  -0.1712  -0.4625   0.5435   0.00    0.0
```

Figure 3.9: Positioning Solution File

In RTKLIB, the Q flag in the positioning solution file (.pos file) indicates the quality flag or type of the GNSS solution [38].

- Q = 0: No solution
- Q = 1: Fixed, solution by carrier-based relative positioning and the integer ambiguity is properly resolved.
- Q = 2: Float, solution by carrier-based relative positioning but the integer ambiguity is not resolved.
- Q = 3: SBAS Solution
- Q = 4: DGPS, solution by code-based DGPS solutions or single point positioning with SBAS corrections
- Q = 5: Single, solution by single point positioning
- Q = 6: PPP Solution

The frequencies used to analyze are L1 (1575.42 MHz) and L2 (1227.60 MHz). Filter type is forward. The elevation mask is at 15° which means satellite that lower than 15° from the horizon of the earth is excluded in analyzation. The signal from GPS we use is broadcast signal, so the ionosphere correction and satellite ephemeris/clock is set to broadcast. For the troposphere correction, it is set to estimate ZTD in order to calculate and display the ZTD in the .pos.stat file. Tick for GPS only.

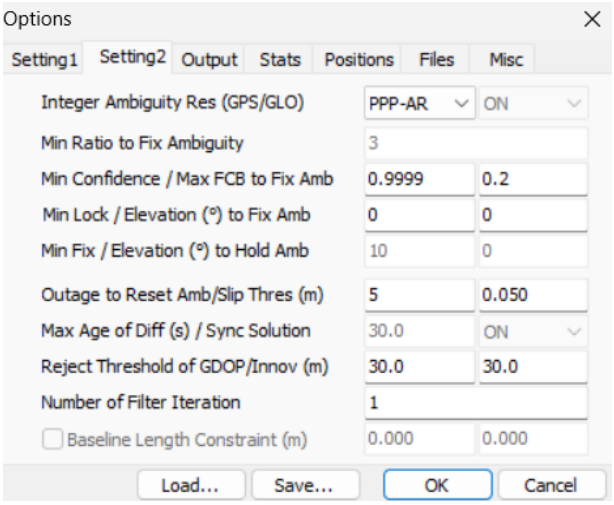


Figure 3.10: RTKPOST Setting 2

PPP-AR the most accurate setting in Integer Ambiguity Resolution. The number of filter iteration is left default as 1. Because over-filter will cause inaccuracy in the ZTD. The unmentioned parameters are left default.

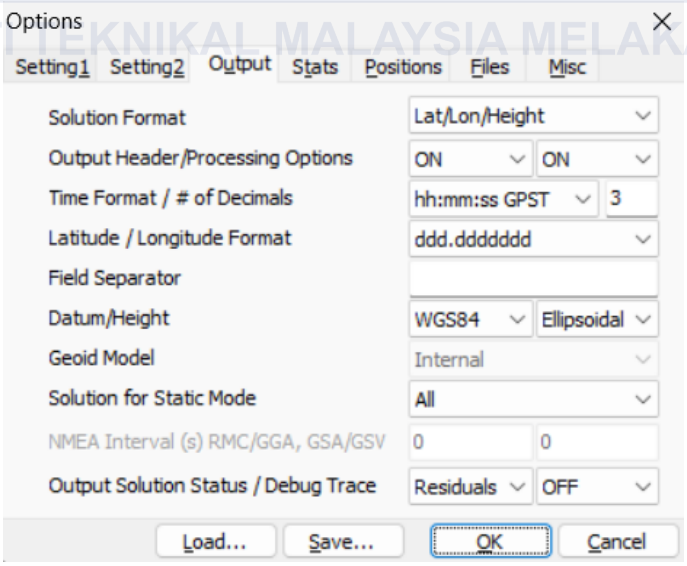


Figure 3.11: RTKPOST Output Setting

Solution format is set to Lat/Lon/Height to obtain latitude and height for the usage of what called surface pressure correction factor or gravity correction factor as shown in equation (3.1) [31]. This factor is then used for ZHD formula (equation (3.2)) [31].

$$f(\theta, h) = 1 - 0.00266\cos(2\theta) - 0.00028h \tag{3.1}$$

- Latitude, θ
- Height (km), h

$$ZHD = 2.2779 \cdot \frac{\rho_S}{f(\theta, h)} \tag{3.2}$$

- Surface Pressure (bar), ρ_S
- Gravity Correction Factor, $f(\theta, h)$

The height is based on the ellipsoidal model, meaning the Earth is assumed to have an elliptical shape instead of its actual uneven shape. This assumption is made to apply the gravity correction factor formula. The output solution status is set to residual, allowing it to generate a solution status file (.pos.stat file), from which the ZTD is obtained. Figure 3.12 shows the solution status file.

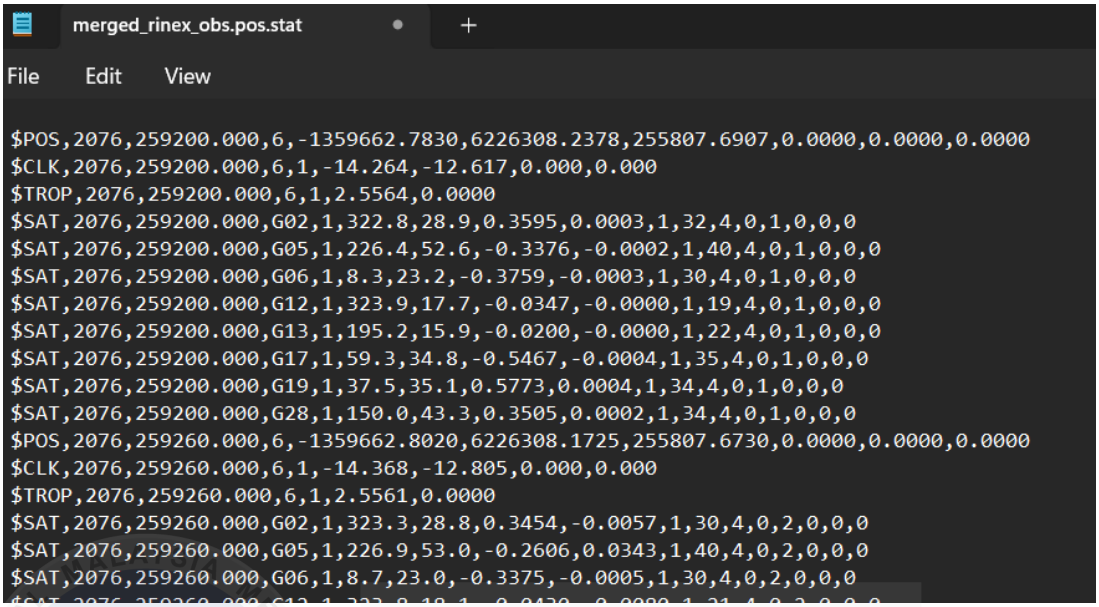


Figure 3.12: Solution Status File

The $TROP line in the solution status file is then extracted out. The $TROP line means Troposphere Parameter States. It is the estimated troposphere parameter (vertical troposphere delay residual). The format of a record is as follows as shown in the RTKLIB version 2.4.2 manual. $TROP,week,tow,stat,rcv,ztd,ztdf [38].

- week/tow: GPS week no./time of week (seconds)
- stat: solution status (Q)
- rcv: receiver (1: rover, 2: base station)
- ztd: zenith total delay (m) float
- ztdf: zenith total delay (m) fixed

As we can see, ztdf which is a more precise solution for ZTD is unsolvable. It is because if the processing didn't successfully resolve the phase ambiguities to integer values, that is the solution is not fixed, ztd may be available, but ztdf (Zenith Total

Delay Fixed) may remain 0. The "fixed" solution relies on resolving the ambiguities to integers, and if this resolution isn't achieved, the ztdf will not be calculated. If the RINEX file is from a single base station and there are issues with the data (outdated or missing), it could affect the ambiguity resolution. So, it's essential to verify the quality and accuracy of the base station data as well. If the solution is using a single station (like rover-only data), ambiguity resolution is often harder to achieve. Base station data is needed to help resolve ambiguities, so if you're relying solely on rover data, achieving a fixed solution might be more difficult [40].

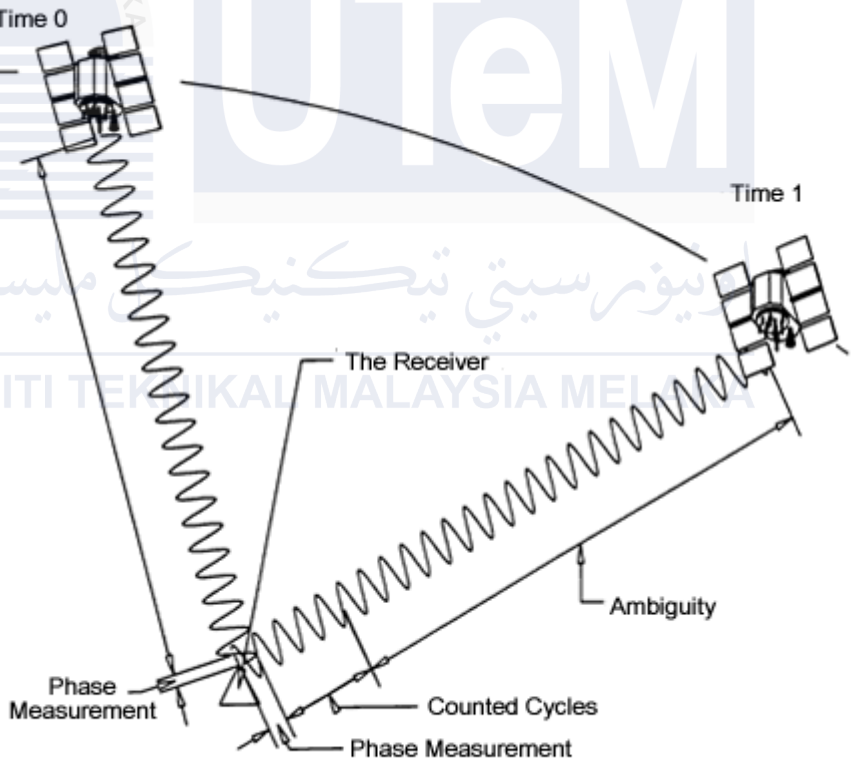


Figure 3.13: Phase Ambiguity [41]

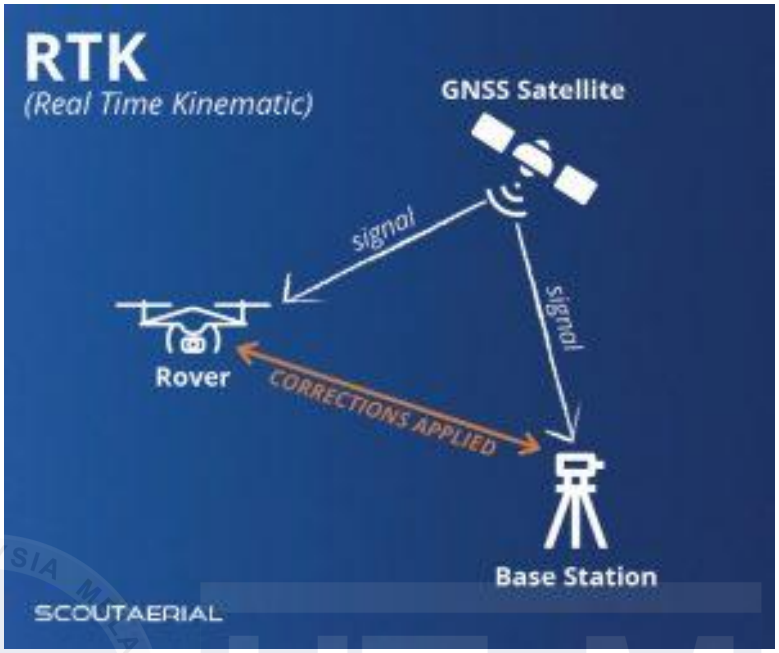


Figure 3.14: Base Station, Rover and Satellite

The GPS week and time of week (TOW) are converted into standard date and time format and paired with their corresponding ZTD values.

3.4.3 Cleaning ZTD data

Figure 3.15 and Figure 3.16 illustrates the uncleaned ZTD data, which includes missing timestamps, invalid date such as the date 31/11/2019, and the presence of outliers.

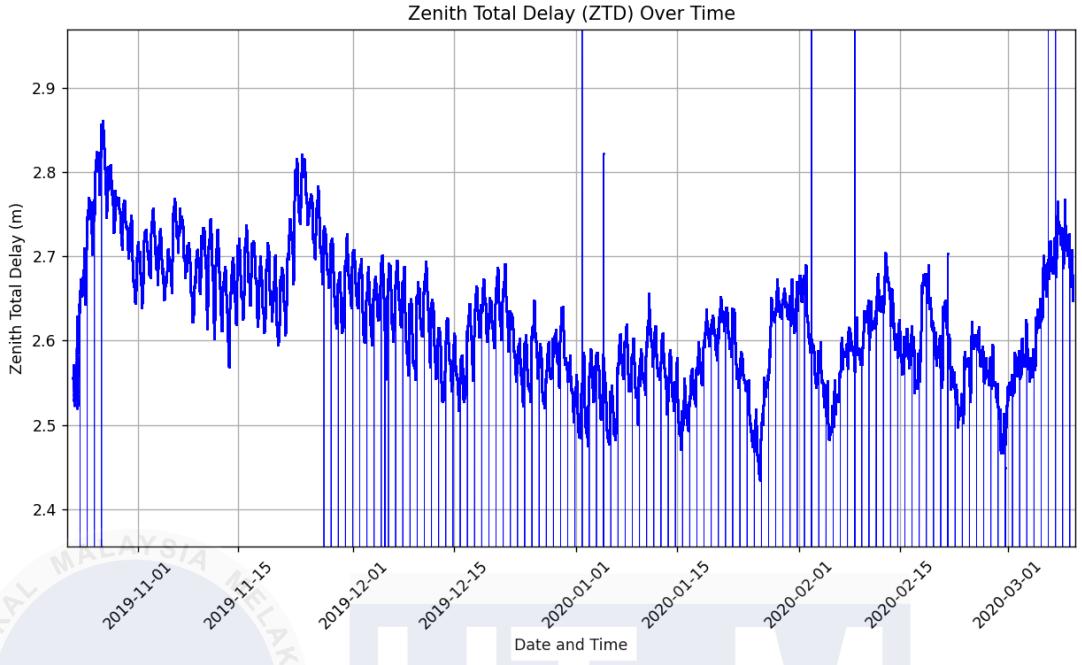


Figure 3.15: Uncleaned ZTD

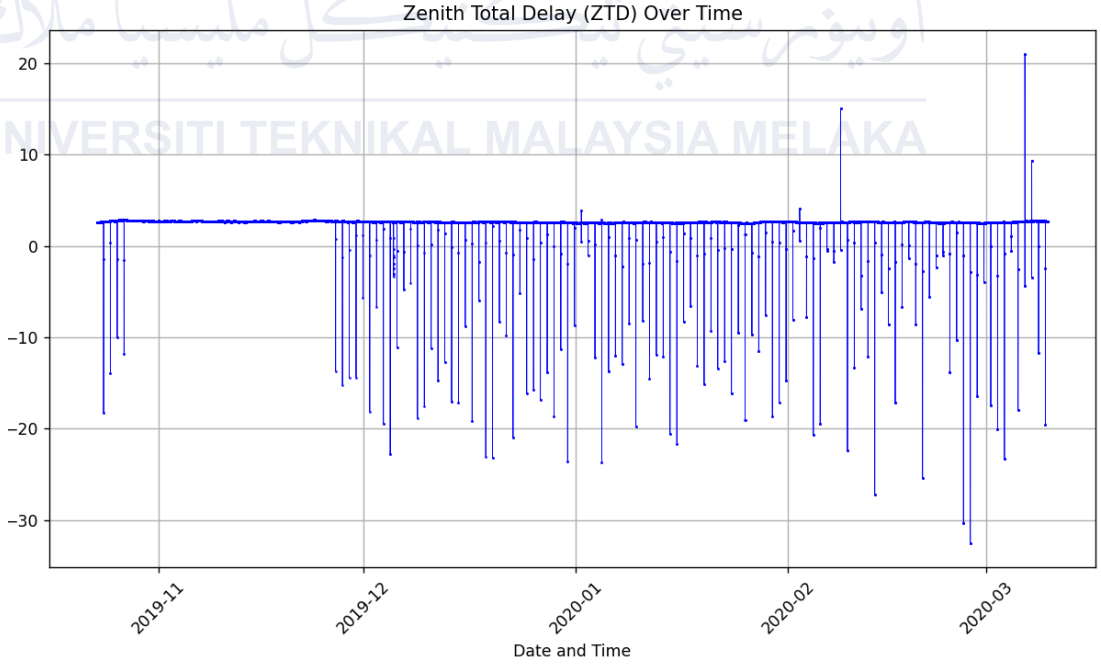


Figure 3.16: ZTD Outliers

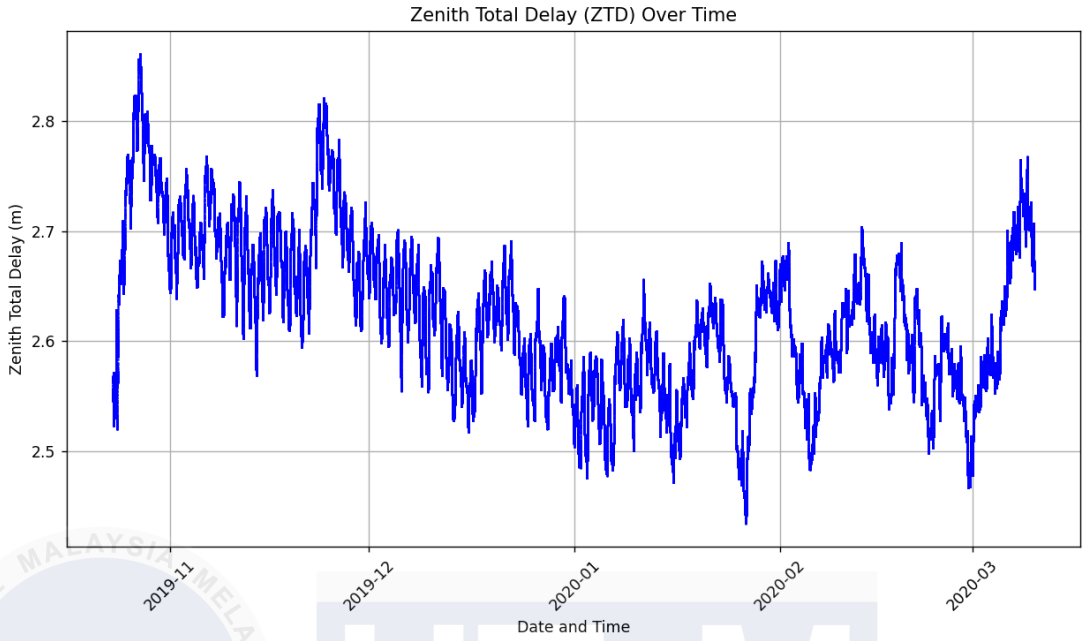


Figure 3.17: Cleaned ZTD

Figure 3.17 displays the cleaned ZTD data, where missing timestamps and outlier values have been filled with the previous data points. This method is considered valid as it involves only a small portion of the dataset. While if it involves a larger portion, interpolation will be considered. The operation was performed using Python code and with manual data adjustments. The timestamp of ZTD is then +8 hours since the time in RINEX is using UTC 0. In Malaysia, the local time is UTC +8. The meteorological data is also recorded in UTC +8. This adjustment ensures that the timestamps in both files are aligned.

3.5 Deriving IWV

3.5.1 Cleaning Latitude and Height

Latitude and height are cleaned by replacing missing values and outliers with interpolations. The cleaned latitude and height are shown in Figure 3.18.

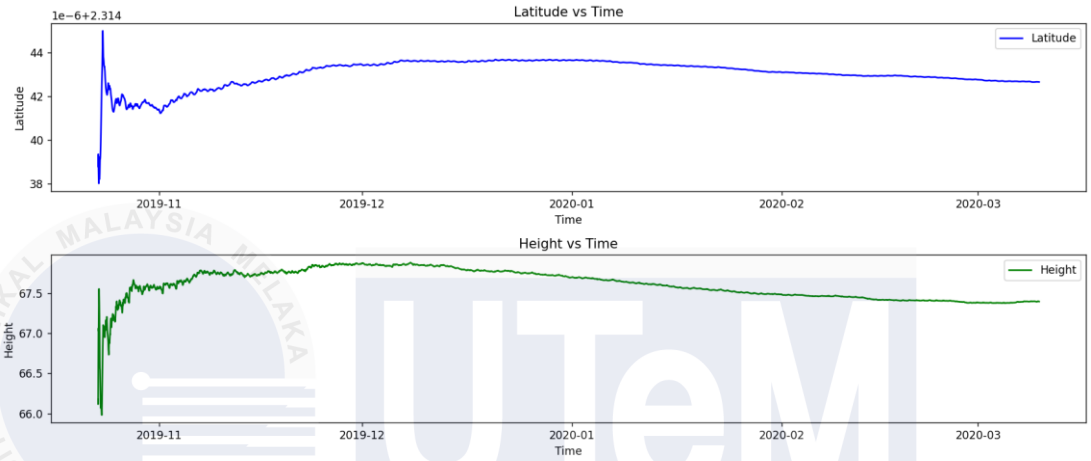


Figure 3.18: Latitude and Height

The initial part of the latitude and height data is inconsistent because the RTKLIB analyzer filters in the forward direction, and it does not yet have enough data to produce a stable solution initially. However, the average latitude and height are calculated for subsequent use because the station is at a static point, rather than a floating one.

3.5.2 Cleaning Meteorological Data

The timestamp of the meteorological data such as temperature, pressure, relative humidity, wind speed, rain rate and solar irradiance is cleaned by replacing missing

values and outliers with the previous data point for small portions of data and using interpolation for larger gaps. The cleaned meteorological data is shown in Figure 3.19.

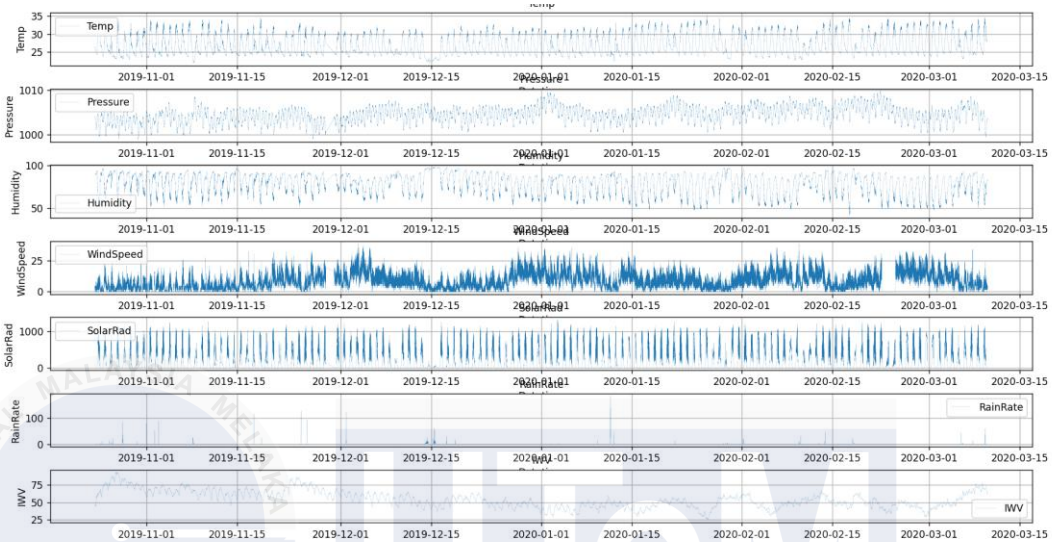


Figure 3.19: Cleaned Meteorological Data

3.5.3 Finding for IWV

Gravity correction factor as shown in equation (3.1) is applied in calculating the Zenith Hydrostatic Delay (ZHD) of Saastamoinen model as shown in equation (3.2) [31]. The average latitude and height are then substituted into Equation (3.1).

$$f(\theta, h) = 1 - 0.00266\cos(2\theta) - 0.00028h \tag{3.1}$$

- Latitude, θ
- Height (km), h

$$ZHD = 2.2779 \cdot \frac{\rho_S}{f(\theta, h)} \tag{3.2}$$

- Surface Pressure (bar), ρ_S
- Gravity Correction Factor, $f(\theta, h)$

After obtaining the ZHD, it can be subtracted from the ZTD, as shown in Equation (3.3).

$$ZWD = ZTD - ZHD \tag{3.3}$$

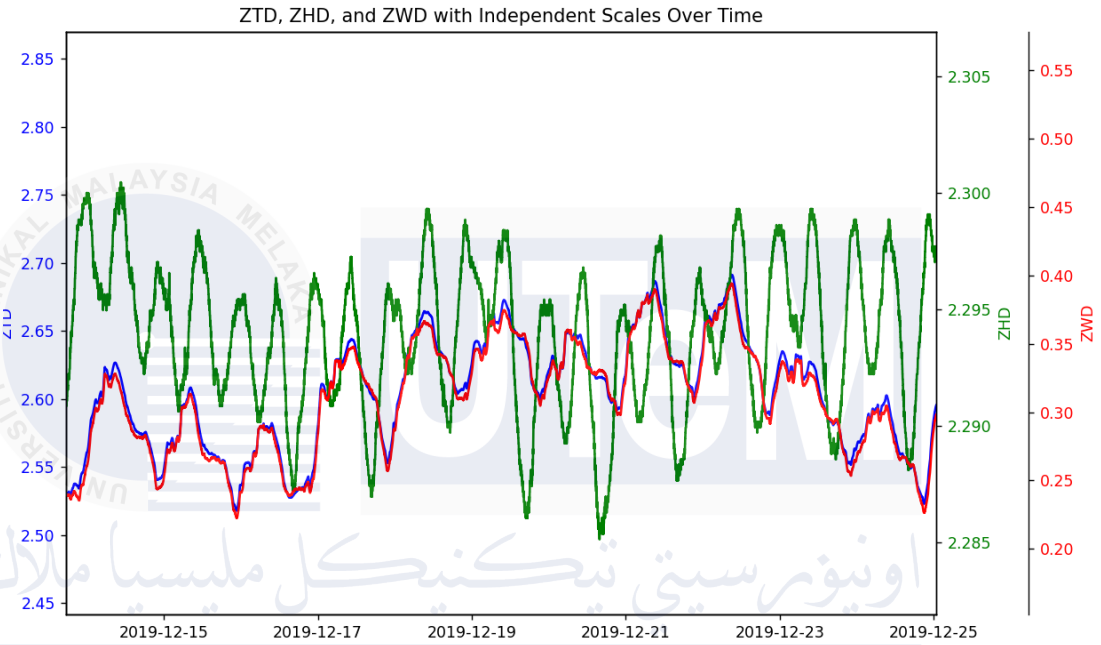


Figure 3.20: ZTD, ZHD and ZWD

As shown in Figure 3.20, the composition percentages can be analyzed as follows:

- Zenith Total Delay (ZTD) ~ 100%
- Zenith Hydrostatic Delays (ZHD) ~ 90%
- Zenith Wet Delays (ZWD) ~ 10%

The ZHD (Zenith Hydrostatic Delay) is primarily influenced by surface pressure, which changes slowly over time and remains relatively stable. In contrast, the ZWD

(Zenith Wet Delay) is affected by atmospheric water vapor, which can vary significantly due to weather conditions, leading to greater fluctuations [42][43][44].

Equation 3.3 represents IWV, equation 3.4 shows the conversion constant, $\bar{K}$ [31], equation 3.5 represents the global weighted mean temperature of the atmosphere and equation 3.6 presents the weighted mean temperature of the atmosphere in the peninsular Malaysia region [45].

$$IWV(kgm^{-2}) = \bar{K} \cdot ZWD \tag{3.3}$$

$$\bar{K} = \frac{10^6}{\left(\left(\frac{k_3}{T_m}\right) + k_2'\right) R_V} \tag{3.4}$$

- Constant, $k_3 = 3.776 \times 10^5 K^2/mbar$
- Constant, $k_2' = 17\ K/mbar$
- Specific gas constant of water vapor, $R_V = 461.5\ J/(kg \cdot K)$
- Water density, $\rho = 1000\ kgm^{-3}$

$$T_M = 0.72T_S + 70.2 \tag{3.5}$$

$$T_M = 0.36T_S + 182.4 \tag{3.6}$$

- Atmosphere mean temperature (unit in K), T_M
- Surface temperature (unit in K), T_S

Equation (3.6) weighted mean temperature of the atmosphere in the Peninsular Malaysia region is used to obtain more accurate results compared to using a global model (equation (3.5)).

The conversion constant, $\bar{K}$, should be around 160. However, the calculated value was incorrect. After confirming the units, it was found that the formula needed to be

multiplied by 100, likely due to a mistake in the unit used. The draft of proof is shown in appendix A.

After obtaining the IWV by substituting in these formulas. The output of the supervised machine learning is prepared.

3.6 Pearson Correlation and VIF Data Analysis

Pearson correlation is a statistical measure that quantifies the strength and direction of a linear relationship between two continuous variables. It is one of the most commonly used correlation coefficients in statistics.

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \quad (3.7)$$

- x_i and y_i are individual data points.
- $\bar{x}$ and $\bar{y}$ are the means of x and y.

Table 3.1: Relationship in Pearson Correlation

Range	Relationship
r = 1	A perfect positive linear relationship.
r = −1	A perfect negative linear relationship.
r = 0	no linear relationship.

The value of r ranges between -1 and 1. But variables may still have a non-linear relationship especially non-linear machine learning are used.

Variance Inflation Factor (VIF) is a statistical metric used to detect multicollinearity in a dataset. Multicollinearity occurs when two or more independent variables in a regression model are highly correlated, which can distort the estimated coefficients and reduce the model's interpretability. VIF measures how much the variance of a regression coefficient is inflated due to multicollinearity with other variables in the model.

$$VIF(X_i) = \frac{1}{1 - R_i^2} \tag{3.8}$$

Where R_i^2 is the coefficient of determination (R-squared) obtained by regressing X_i on all other independent variables. While R-squared is the squared of Pearson correlation. A linear regression model is fit with X_i as shown in equation (3.9).

$$X_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{k-1} X_{k-1} + \epsilon \tag{3.9}$$

Table 3.2: Interpretation of VIF

Condition	Interpretation
VIF = 1	No multicollinearity. The variable is completely independent of other variables.
1< VIF ≤ 5	Low to moderate multicollinearity (acceptable).
VIF > 5	High multicollinearity (problematic and may need further investigation).
VIF > 10	Severe multicollinearity. This variable may need to be removed or the model re-evaluated.

3.7 Machine Learning Data Split and Evaluation Metrics

The investigated inputs or features (meteorological data) and output (IWV) is fed into various machine learning model to evaluate its performance. The best 3 machine learning models will be presented in the result. In machine learning, it is practical to split dataset to a common 80% and 20% or 70% and 30% for train and validate and test respectively. Cross validation is employed to address limitations caused by insufficient data for the shortest training time model (bagged tree). Table 3.3 shows the data splitting of 3 machine learning models.

Table 3.3: Data Splitting

Models	Feedforward Neural Network	Bagged Tree	Fine Gaussian SVM
Train phase	80% train 10% validation 10% test	80% train 20% validation	85% train 15% validation
Cross validation	-	With cross-validation k=5	-
Test with unseen data	Last 1 week for test	Last 1 week for test	Last 1 week for test

In regression tasks, the goal is to predict continuous values based on input features. Evaluation metrics for regression models help measure how well the predicted values match the actual target values. Common regression evaluation metrics include:

1. **Mean Absolute Error (MAE)**:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.9}$$

2. Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.10}$$

3. Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.11}$$

4. R-Squared (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{3.12}$$

5. Adjusted R-Squared:

$$Adj\ R^2 = 1 - \frac{(1 - R^2)(n-1)}{n - p - 1} \tag{3.13}$$

- predicted value, y_i
- actual value, $\hat{y}_i$
- mean of actual value, $\bar{y}_i$
- number of samples, n
- number of features/inputs in the model, p

3.8 Moving Average Filter (MAF)

A moving average filter (MAF) is a simple and widely used technique in signal processing and time series analysis for smoothing data. It works by calculating the average of a fixed number of consecutive data points within a sliding window that moves across the dataset. The resulting smoothed data reduces noise or fluctuations, making trends or patterns more apparent.

However, the moving average filter also has some side effects. It introduces a lag in the data because the filtered value at each point is influenced by neighboring points within the window. Additionally, while larger window sizes produce smoother results, they may distort or eliminate short-term variations, potentially masking important details in the data.

In this study, a MAF is applied to reduce noise and improve data quality before further analysis. For example, in Figure 3.21 shows the effect of applying a moving average filter. The blue line represents unfiltered data, while the red line shows the smoothed data after filtering.

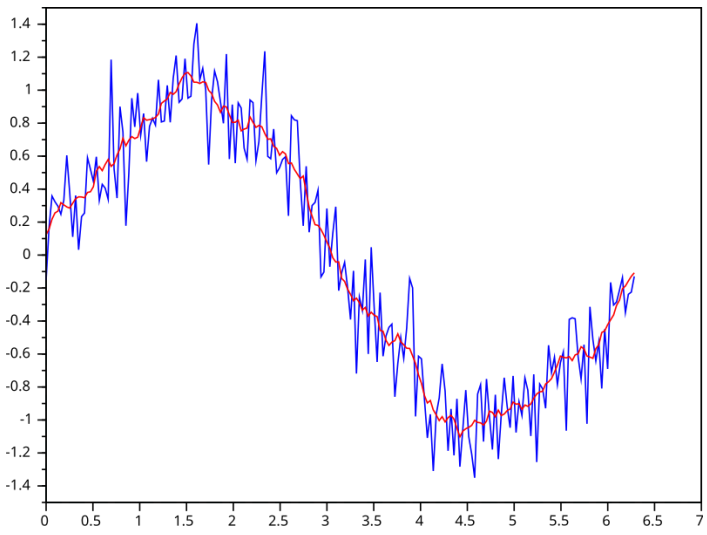


Figure 3.21: Moving Average Filter

CHAPTER 4

RESULTS AND DISCUSSION

This chapter explores into the results of the Pearson correlation, variance inflation factor, investigation on the relationship between meteorological data and IWV, moving average filter improvement, machine learning models and adding filter and summary of machine learning models.

4.1 Pearson Correlation

Since correlation and VIF only account for linear relationships, they serve as references to simplify the selection of features. However, testing with nonlinear models is still necessary to confirm their suitability. Figure 4.1 shows the correlation between all data including meteorological data and IWV.

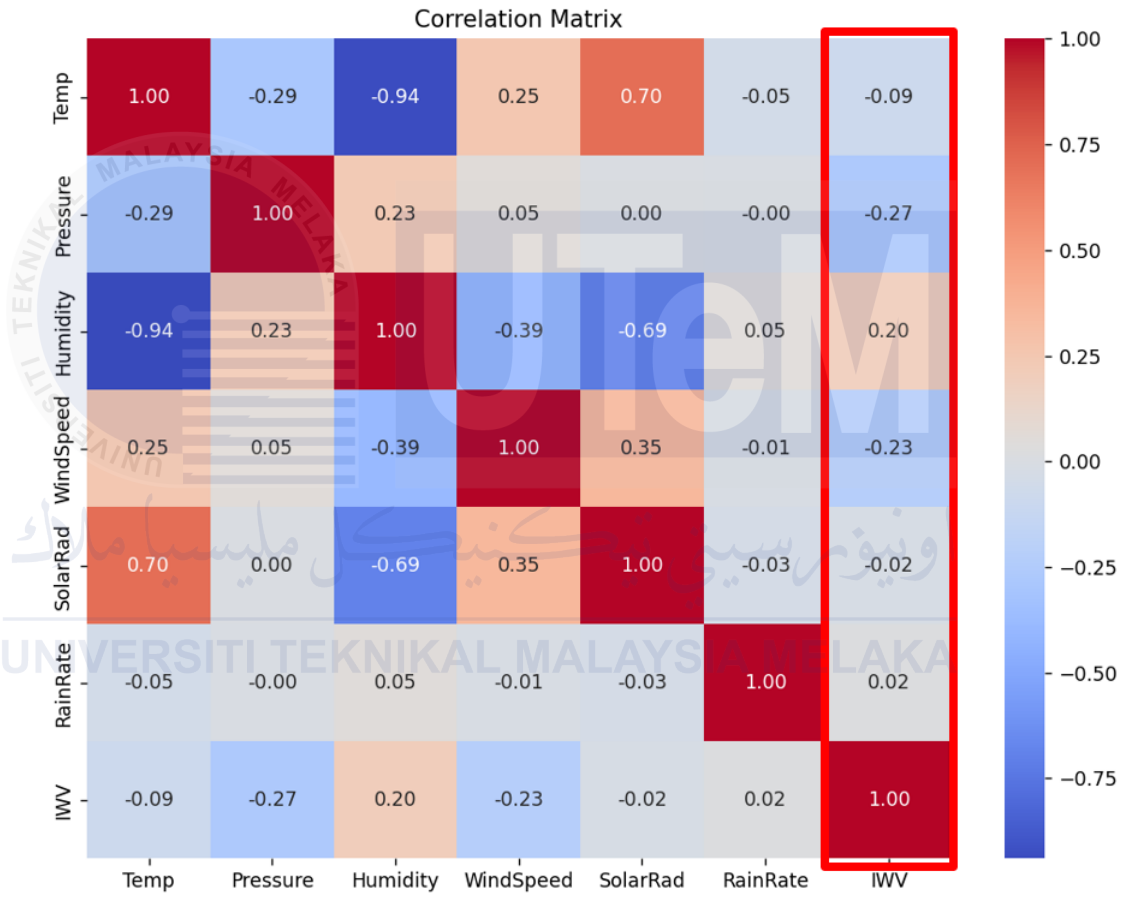


Figure 4.1: Pearson Correlation

The temperature, pressure, humidity, and wind speed were used as input features and tested using a neural network. This combination produced the best results compared to using other meteorological data.

A positive time-lag Pearson correlation of meteorological data vs IWV with a forwarded 2-hour interval is investigated as shown in Table 4.1. This means the meteorological is forecasting the IWV in a few hours later. The forwarded means the present meteorological data is forecasting or predicting the future IWV. This analysis is conducted to assess the validity of the applied filter, as the filter introduces a lag in the data. Because adding filter can improve the predicted result. In the table, the IWV is forwarded according to the forwarding hours, and the correlation is calculated. The investigation is done in the range of day and week respectively as shown in Table 4.1, Figure 4.2, Table 4.2 and Figure 4.3.

Table 4.1: Positive Time-Lag Pearson Correlation in a Day

Forecasted hours	Temp(%)	Pressure(%)	Humidity(%)	Wind Speed(%)	Solar Radiance(%)	Rain Rate(%)	Total (%)
0	9	27	20	23	2	2	83
2	13	24	25	25	6	2	95
4	17	20	28	27	10	1	103
6	19	18	30	28	14	1	110
8	17	21	29	29	17	1	114
10	12	27	25	28	17	1	110
12	5	33	18	26	11	1	94
14	2	33	11	23	3	2	74
16	7	29	7	20	5	1	69
18	7	25	7	19	7	1	66
20	3	24	10	20	6	1	64
22	3	26	15	22	2	1	69
24	8	26	20	24	3	1	82

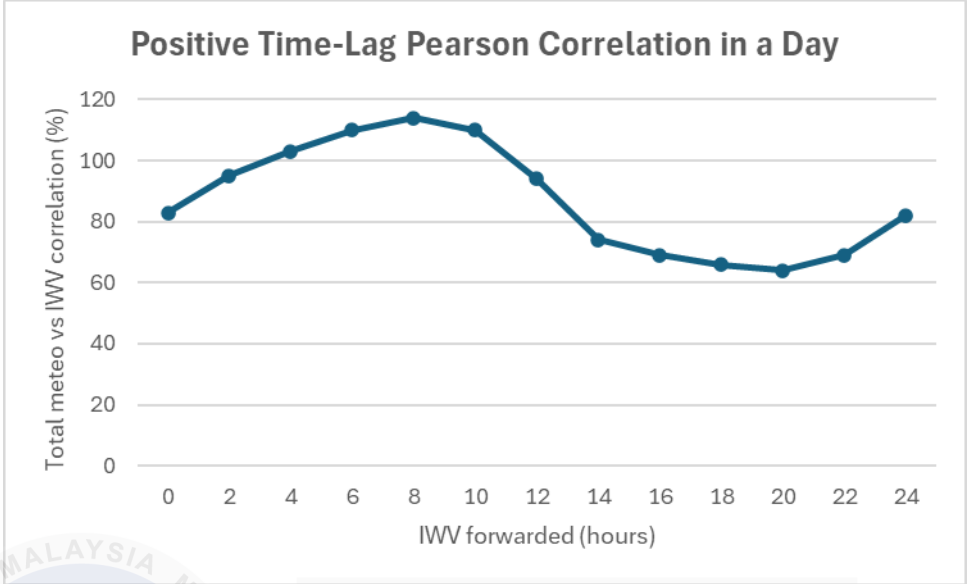


Figure 4.2: Total Positive Time-Lag Pearson Correlation in a Day

Table 4.2: Positive Time-Lag Pearson Correlation in a Week

Forecasted days	Temp(%)	Pressure(%)	Humidity(%)	Wind Speed(%)	Solar Radiance(%)	Rain Rate(%)	Total (%)
0	9	27	20	23	2	2	83
1	8	26	20	24	3	1	82
2	8	26	18	26	2	1	81
3	8	29	17	27	2	1	84
4	4	34	11	24	0	0	73
5	3	35	7	19	1	0	65
6	3	33	6	13	2	1	58
7	5	31	5	11	2	2	56

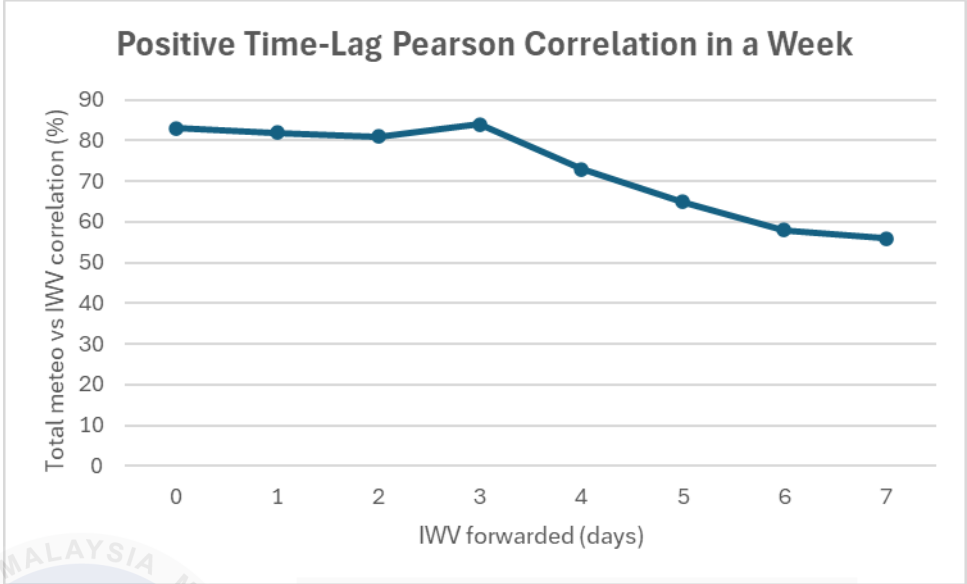


Figure 4.3: Total Positive Time-Lag Pearson Correlation in a Week

From the graphs, the conclusion can be made that the IWV can be forecasted up to a day ahead without exceeding 12 hours. However, over a week, the linear correlation decreases significantly.

4.2 Variance Inflation Factor (VIF)

The Variance Inflation Factor (VIF) is used for feature selection as shown in Table 4.3. While the time-lagged (forwarded) IWV is also examined to evaluate the impact of the applied filter as shown in Table 4.4 and Table 4.5. Figure 4.4 and Figure 4.5 show the VIF of IWV-only with consecutive forecasts in a day and in a week respectively. The word “forwarded” means the present meteorological data is forecasting or predicting the future IWV.

Table 4.3: Variance Inflation Factor (VIF)

	Feature	VIF
1.	Temp	11.512744
2.	Pressure	1.358619
3.	Humidity	11.751136
4.	Wind Speed	1.441075
5.	Solar Radiance	2.418669
6.	Rain Rate	1.003366
7.	IWV	1.312346

Table 4.4: VIF with Forecasted Hours

Forecasted hours	Temp(%)	Pressure(%)	Humidity(%)	Windspeed(%)	Solar Radiance(%)	Rain Rate(%)	IWV(%)
0	11.5127	1.3586	11.7511	1.4410	2.4186	1.0033	1.3123
2	11.5185	1.3413	11.7884	1.4426	2.4008	1.0033	1.3168
4	11.5605	1.3038	11.8211	1.4443	2.3685	1.0033	1.3019
6	11.6503	1.2801	11.8919	1.4428	2.3335	1.0033	1.2955
8	11.7955	1.2866	11.9846	1.4399	2.3094	1.0033	1.3054
10	11.9335	1.3198	12.0339	1.4375	2.3039	1.0034	1.3303
12	12.0062	1.3566	12.0456	1.4377	2.3064	1.0034	1.3478
14	12.0910	1.3510	12.0922	1.4376	2.3220	1.0035	1.3340
16	12.1761	1.3130	12.1630	1.4364	2.3478	1.0035	1.3044
18	12.1033	1.2887	12.1201	1.4365	2.3702	1.0034	1.2770
20	11.8893	1.2940	11.9734	1.4393	2.3882	1.0034	1.2681
22	11.6840	1.3186	11.8260	1.4434	2.3978	1.0034	1.2819
24	11.5749	1.3349	11.7562	1.4467	2.3974	1.0034	1.3011

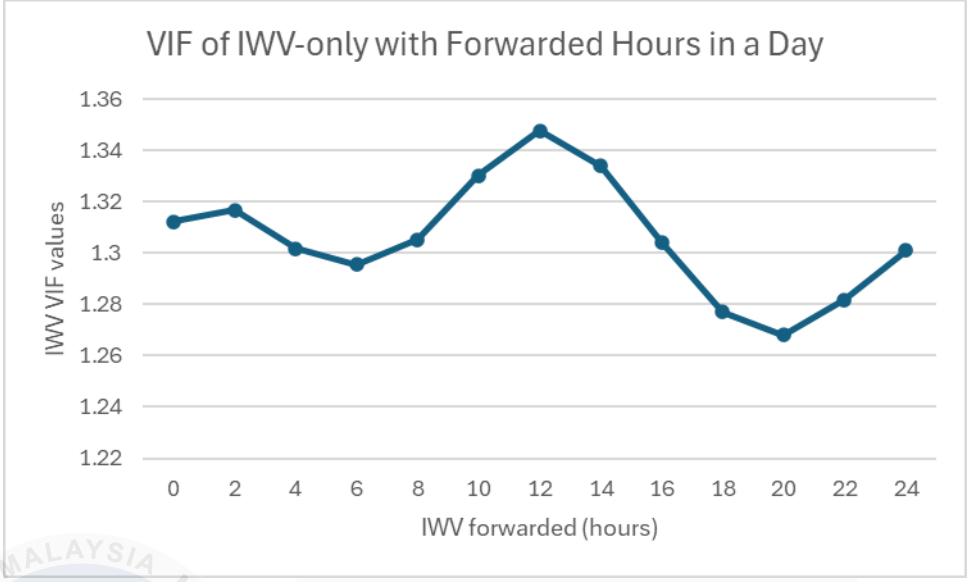


Figure 4.4: VIF of IWV-only with Forwarded Hours in a Day

Table 4.5: VIF with Forecasted Days

Forwarded Time(days)	Temp(%)	Pressure(%)	Humidity(%)	Windspeed(%)	Solar Radiance(%)	Rain Rate(%)	IWV(%)
0	11.5127	1.3586	11.7511	1.4410	2.4186	1.0033	1.3123
1	11.5749	1.3349	11.7562	1.4467	2.3974	1.0034	1.3011
2	11.3680	1.3341	11.3569	1.4641	2.3971	1.0034	1.2763
3	11.2150	1.3722	11.0060	1.4841	2.4109	1.0034	1.2849
4	11.2618	1.4208	10.7904	1.4919	2.4119	1.0034	1.2766
5	11.3737	1.4372	10.7627	1.4824	2.4115	1.0035	1.2504
6	11.4177	1.4198	10.7120	1.4659	2.4139	1.0035	1.2137
7	11.5009	1.3985	10.6798	1.4600	2.4101	1.0035	1.1935

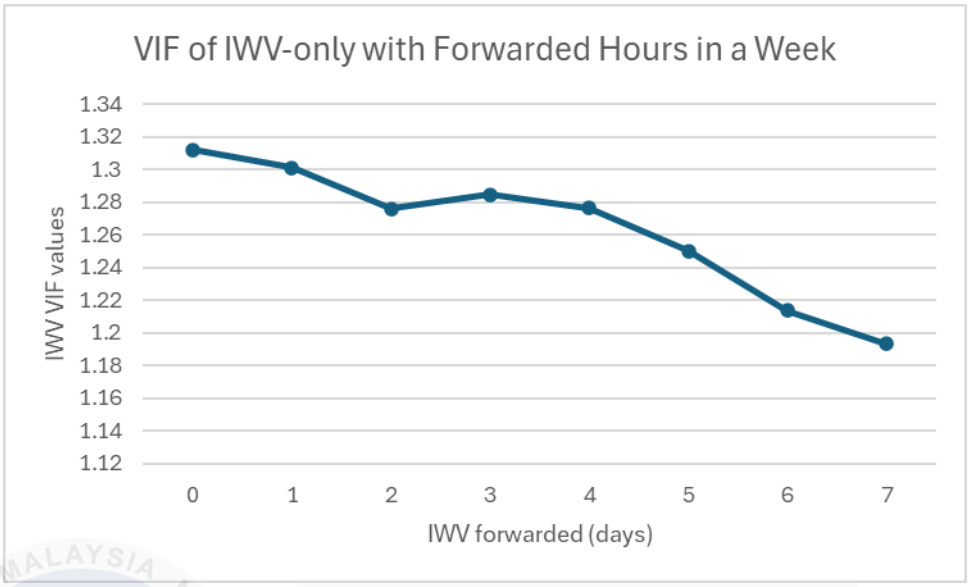


Figure 4.5: VIF of IWV-only with Forwarded Hours in a Week

From the graphs, the conclusion can be made that the IWV can be forecasted up to a day ahead without exceeding 16 hours. However, over a week, the linear correlation decreases significantly.

The final input features selected for the machine learning model include temperature, pressure, relative humidity and wind speed, with the output being IWV, as shown in Figure 4.6.

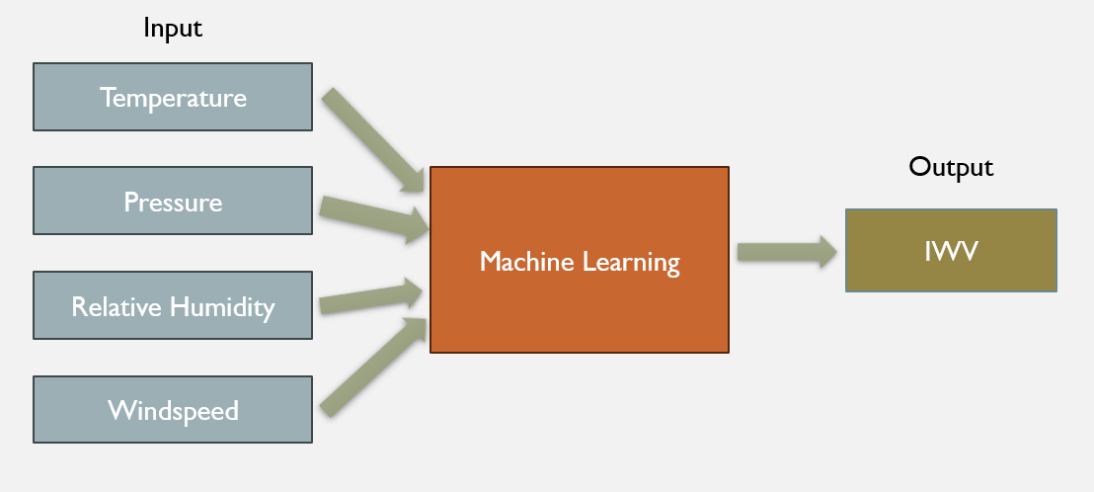


Figure 4.6: Input Features and Output of the Machine Learning

4.3 Investigation on the relationship between meteorological data and IWV

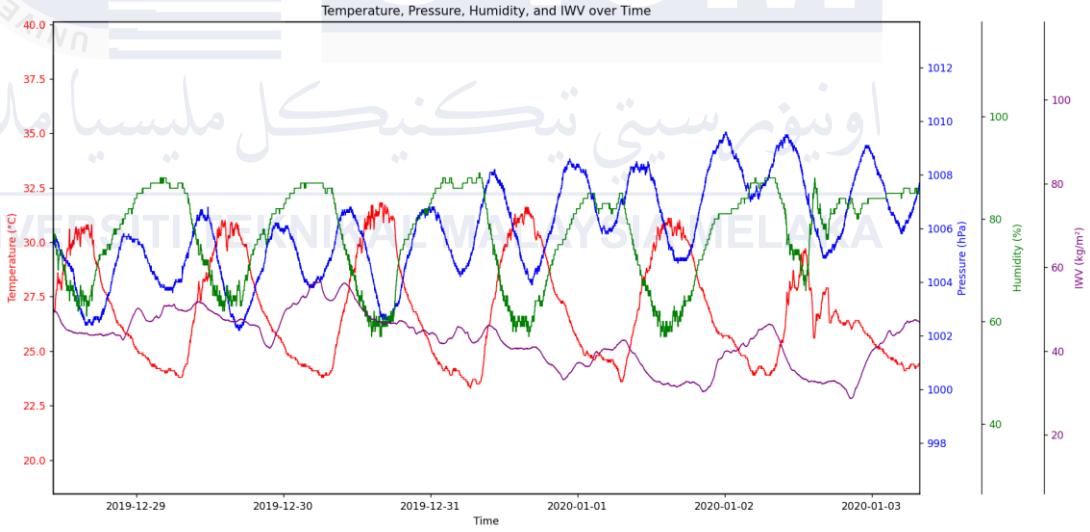


Figure 4.7: Relationship between 3 important meteorological data and IWV

As shown in Figure 4.7, the relationship between temperature (in red), pressure (in blue), relative humidity (in green), and IWV (in purple) can be analyzed. When the temperature rises, relative humidity tends to decrease. This is because relative

humidity is influenced by temperature. Relative humidity is defined as the air's capacity to hold moisture. As temperature increases, the air's capacity to hold moisture increases, causing a drop in relative humidity. Therefore, these two features temperature and relative humidity together can provide valuable information about the moisture content in the air.

Both temperature and relative humidity follow a daily cycle. In contrast, pressure follows a bi-daily cycle. When the temperature rises or falls, pressure tends to increase. However, when the temperature remains stable, pressure tends to decrease. This indicates an inverse relationship between pressure and temperature variation in the dataset.

However, for the wind speed, it will fluctuate and have higher intensity during daytime and when higher temperatures.

The IWV is typically the lowest at 6 to 8 am and achieve the highest at 4 to 6 pm. It goes through this single cycle in a day.

4.4 Moving Average Filter Improvement

Applying a longer filter can result in a much higher R, but it may eliminate the daily pattern in the data. To achieve an optimal balance, the goal is to retain the daily pattern while filtering out high-frequency noise. The best filter duration is determined by observing the time response plot and identifying the filter hour that provides the most suitable balance between preserving the daily pattern and reducing noise. An example from feedforward neural network as shown in Figure 4.8 demonstrates how the degree of filter affects the filter result.

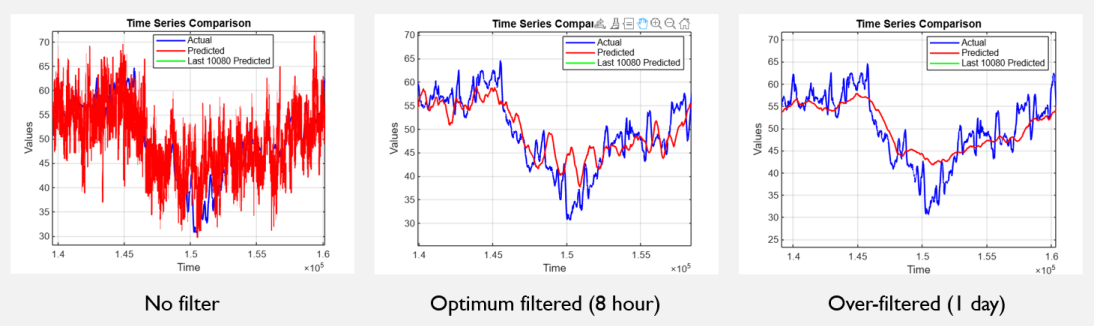


Figure 4.8: Filter Effect

The best filter window for feed-forward neural network is 8 hours, for bagged tree is 2 hours, and for fine gaussian SVM is 8 hours which will cause 4 hours, 1 hour and 4 hours lag for the respective models.

4.5 Machine Learning Models and Adding Filter

4.5.1 Feedforward Neural Network

The data split while train is Train 80%, validation 10% and test 10%. After training, the model is tested with one week of unseen data. The feed-forward neural network architecture is optimized by gradually increasing the number of neurons in the first layer until there is no significant improvement in the results. A second layer is then added, and the number of neurons in the second layer is increased until further improvements become negligible. Another factor to consider is the training time. If the training time becomes excessively long, the process is stopped at that point. The front layer primarily focuses on feature extraction, while the back layer is responsible for classification. The final architecture consists of 80 neurons in the first layer and 20 neurons in the second layer, as illustrated in Figure 4.9.

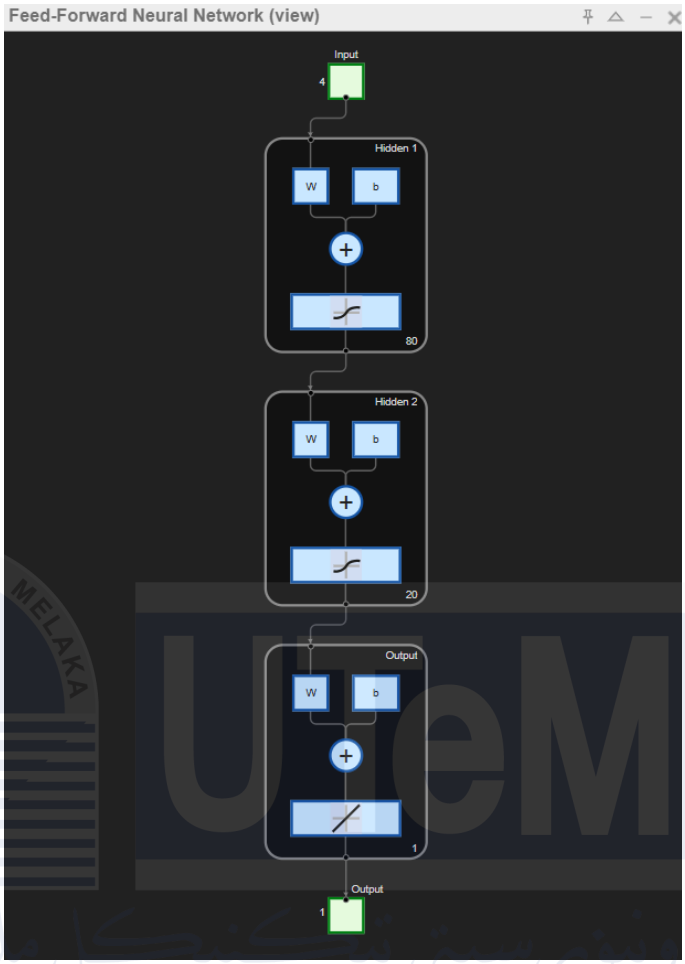


Figure 4.9: Feedforward Neural Network Architecture

The R value is treated as the evaluation metrics in producing the best model. The graph for R is shown in Figure 4.10. The test and validation data point are selected randomly from the whole data.

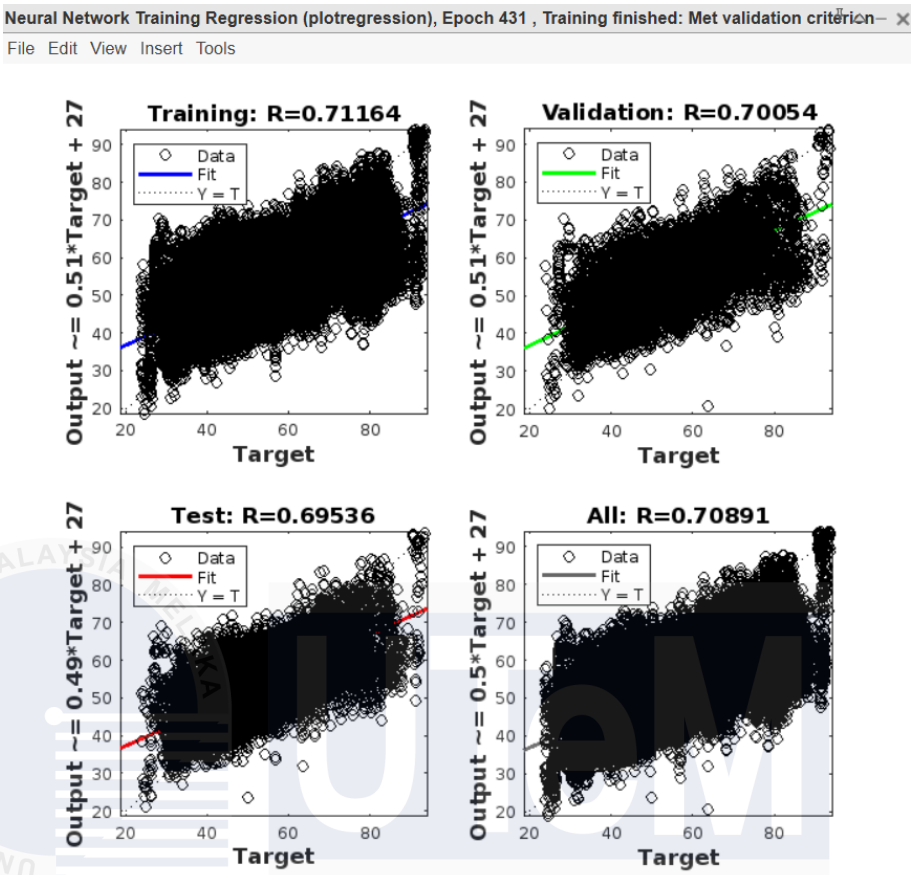


Figure 4.10: R Value for Training, Validation, Test and Total

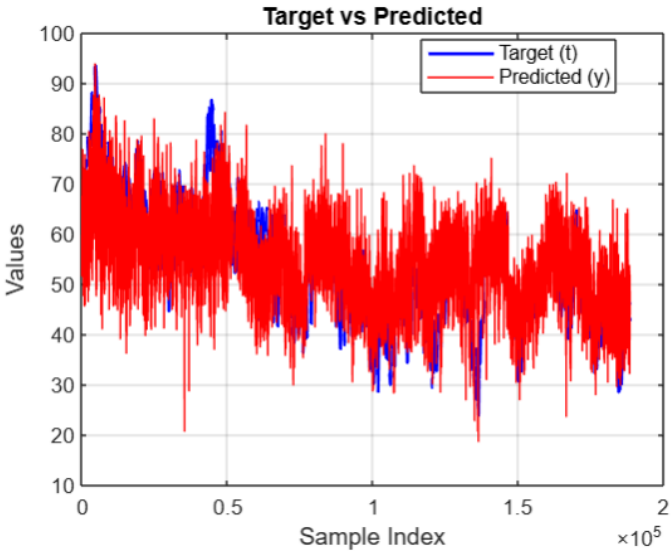


Figure 4.11: Feedforward Neural Network Time Series Graph

Figure 4.12 shows the comparison of test result between without filter and filtered of each hyperparameters. R is the evaluate metric. While the result of each hyperparameter and without filter, forwarded for purpose of adding filter and filtered is shown in appendix B.

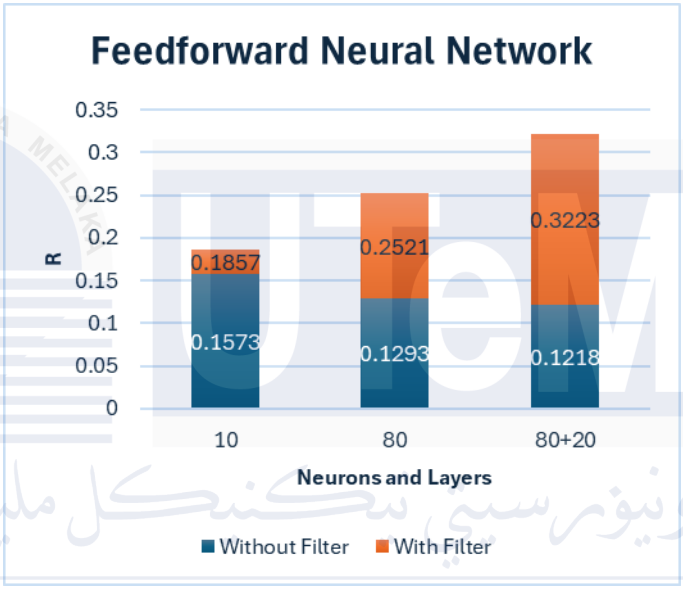


Figure 4.12: FFNN without Filter and Filtered of Each Hyperparameter

The best test result obtained is R = 0.3223.

4.5.2 Bagged Tree

The data split while train is Train 80% and validation 20%. After training, the model is tested with one week of unseen data. To optimize a bagged tree, the number of learners is increased, and the minimum number of leaves is decreased. The more learners, the better the performance, but it may lead to overfitting. The fewer leaves,

the better the performance, but it also risks overfitting. However, the final number of learners is set to 100, and the minimum number of leaves is set to 5. Figure 4.13, Figure 4.14 and Figure 4.15 shows the training results.

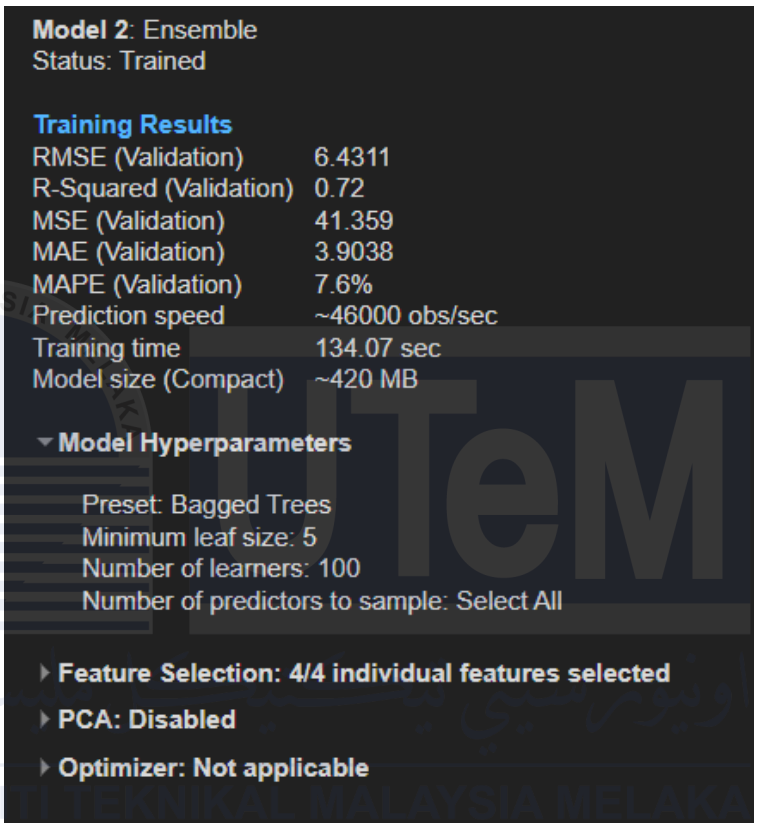


Figure 4.13: Bagged Tree Training Result

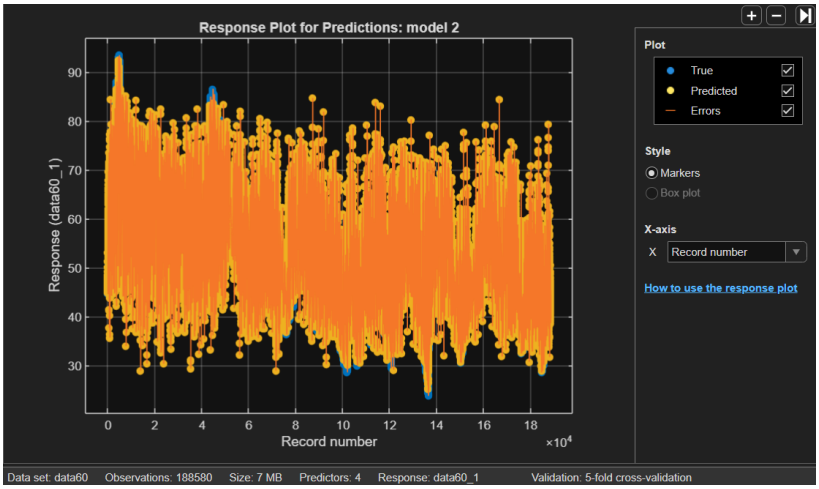


Figure 4.14: Bagged Tree Time Series Response Plot

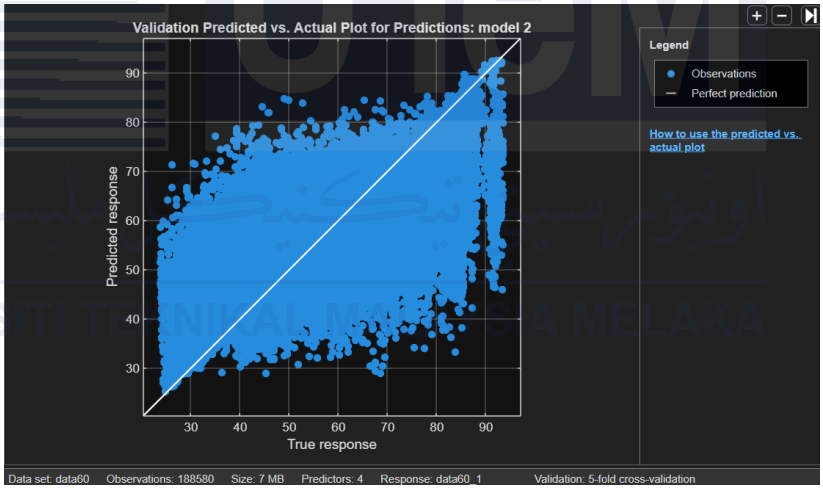


Figure 4.15: Bagged Tree R Plot

Figure 4.16 shows the comparison of test result between without filter and filtered of each hyperparameters. R is the evaluate metric. While the result of each hyperparameter and without filter, forwarded for purpose of adding filter and filtered is shown in appendix B.

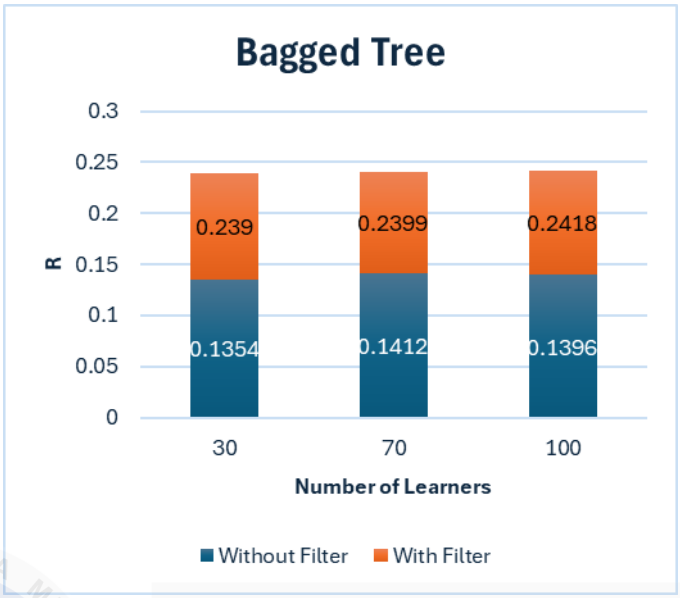


Figure 4.16: Bagged Tree without Filter and Filtered of Each Hyperparameter

The best test result obtained is R = 0.2418.

4.5.3 Fine Gaussian SVM

The data is split into 85% for training and 15% for validation. After training, the model is tested with one week of unseen data. The Fine Gaussian SVM model is set to automatic mode, allowing it to optimize and tune itself to achieve the best performance. Figures 4.17, 4.18, and 4.19 illustrate the training results.

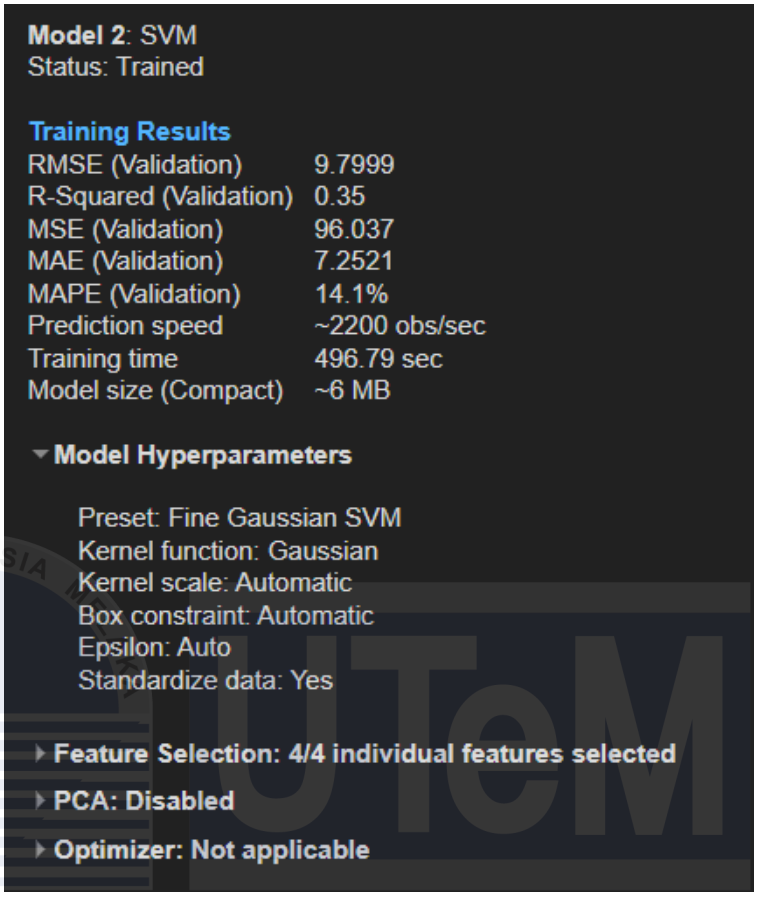


Figure 4.17: Fine Gaussian SVM Training Result

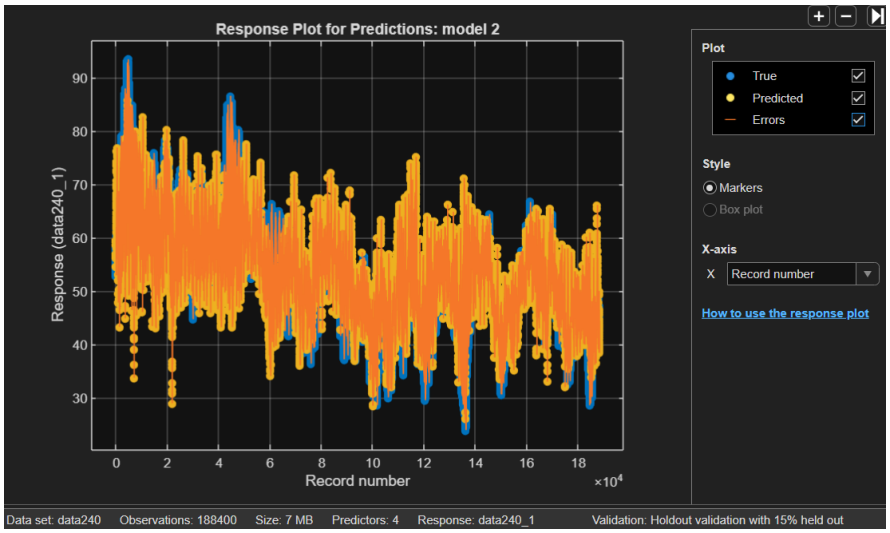


Figure 4.18: Fine Gaussian SVM Time Series Response Plot

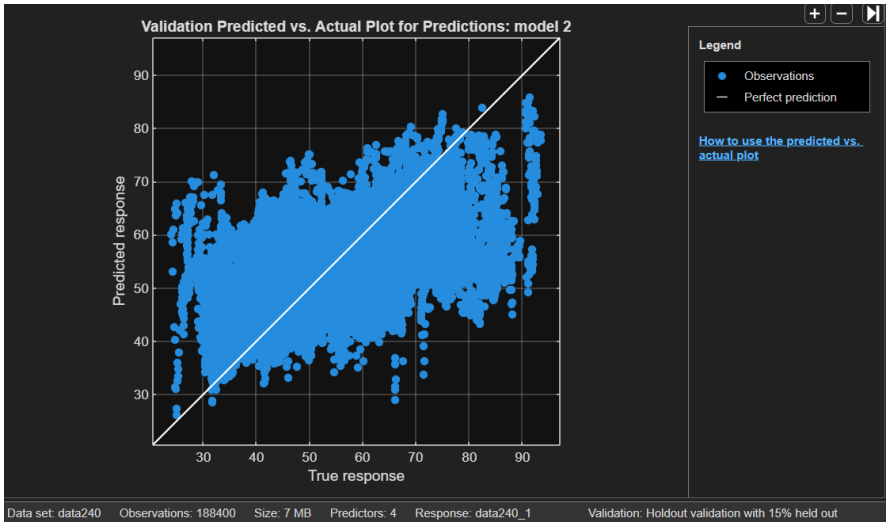


Figure 4.19: Fine Gaussian SVM R Plot

Figure 4.19 shows the comparison of test result between without filter and filtered. R is the evaluate metric. While the result of without filter, forwarded for purpose of adding filter and filtered is shown in appendix B.

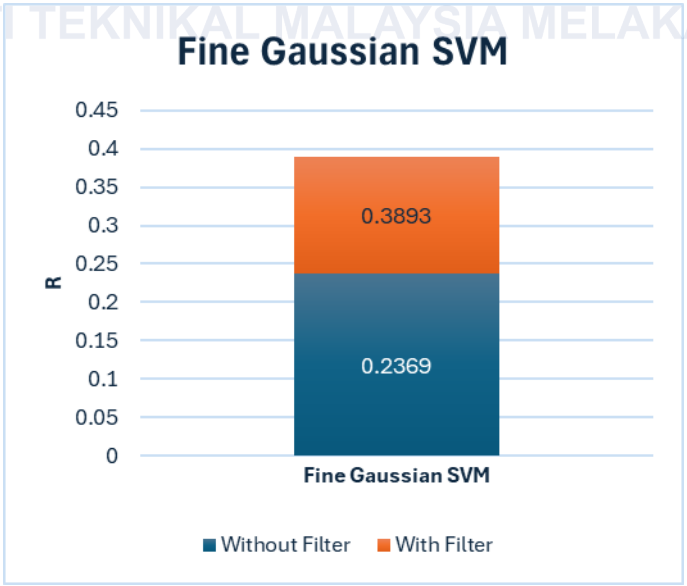


Figure 4.20: Fine Gaussian SVM without Filter and Filtered of Each Hyperparameter

The best test result obtained is R = 0.3893.

4.6 Summary of Machine Learning Models

Afterward, the filtered is considered for the best result of both 3 models. Filter has averagely increased the R value by 13% in total.

Figure 4.21 displays a comparison of the time series graphs for the three models with the actual data. Figure 4.22 provides a close-up view of this comparison. Figure 4.23 presents the time series graph averaged to weeks, and Figure 4.24 summarizes the best results for all three models.

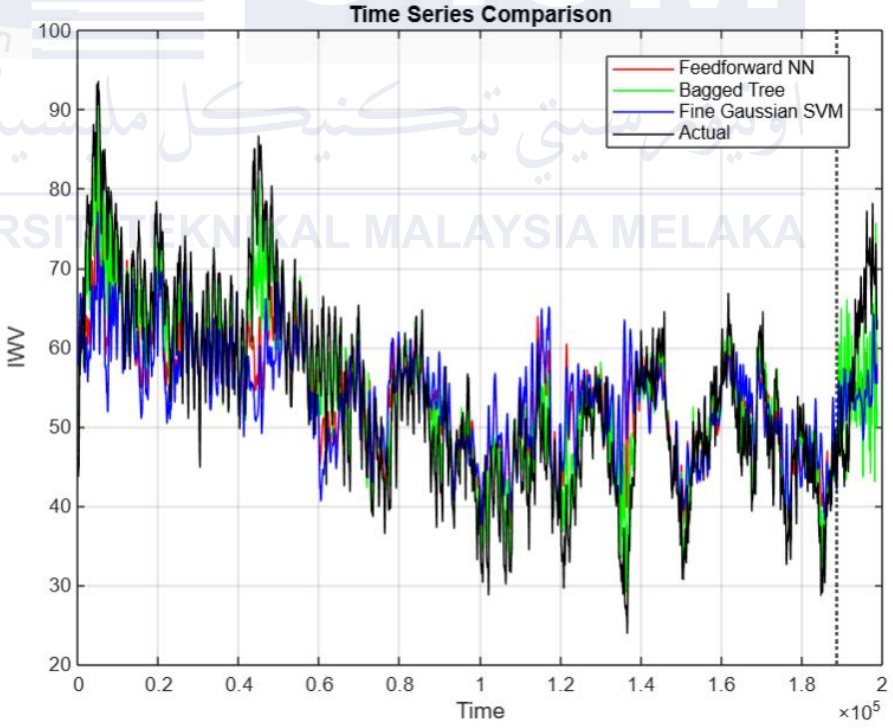


Figure 4.21: Time Series Graphs for 3 Models with Actual Data

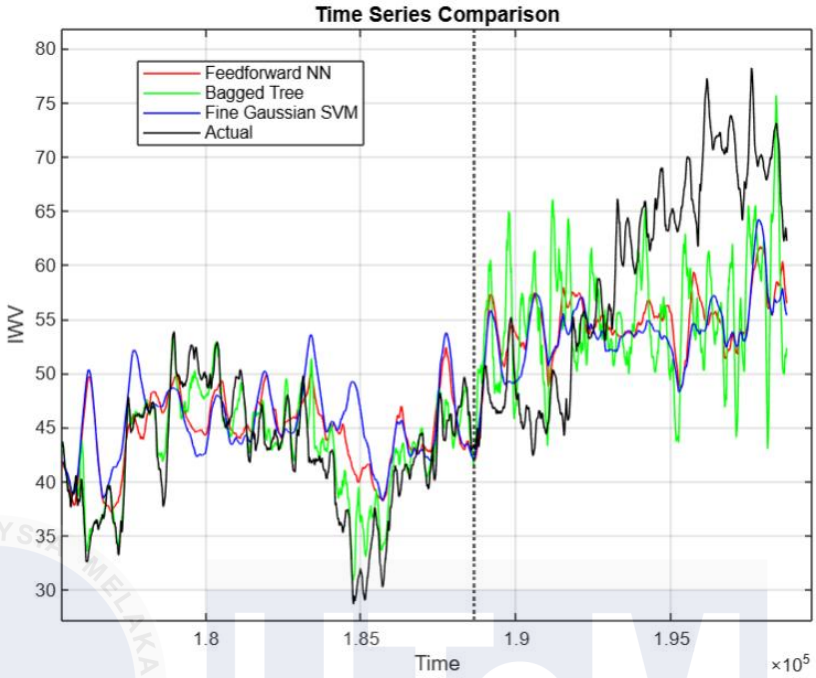


Figure 4.22: Close up of Time Series Graphs for 3 Models with Actual Data

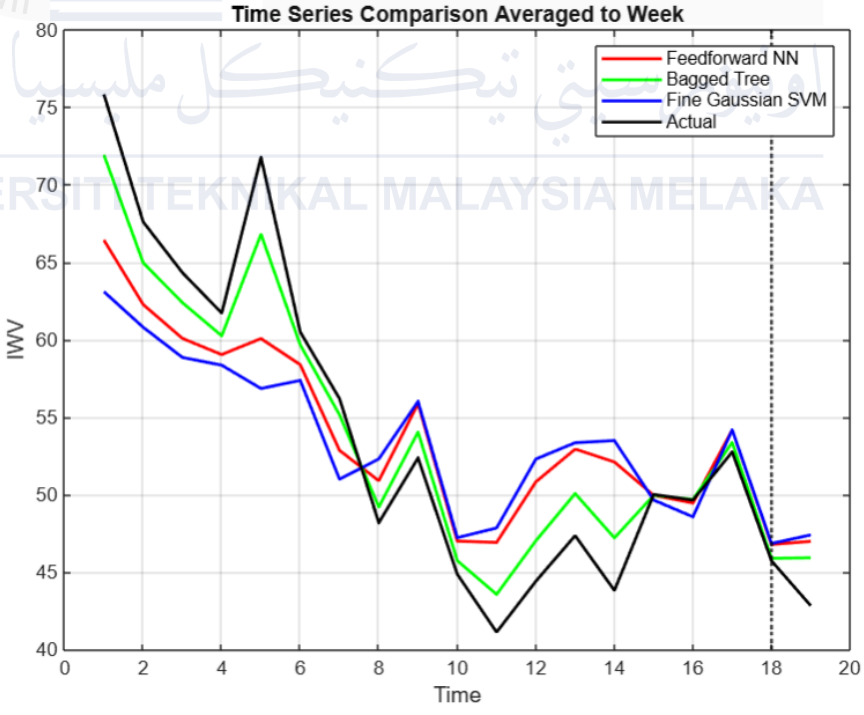


Figure 4.23: Time Series Graph Averaged to Weeks

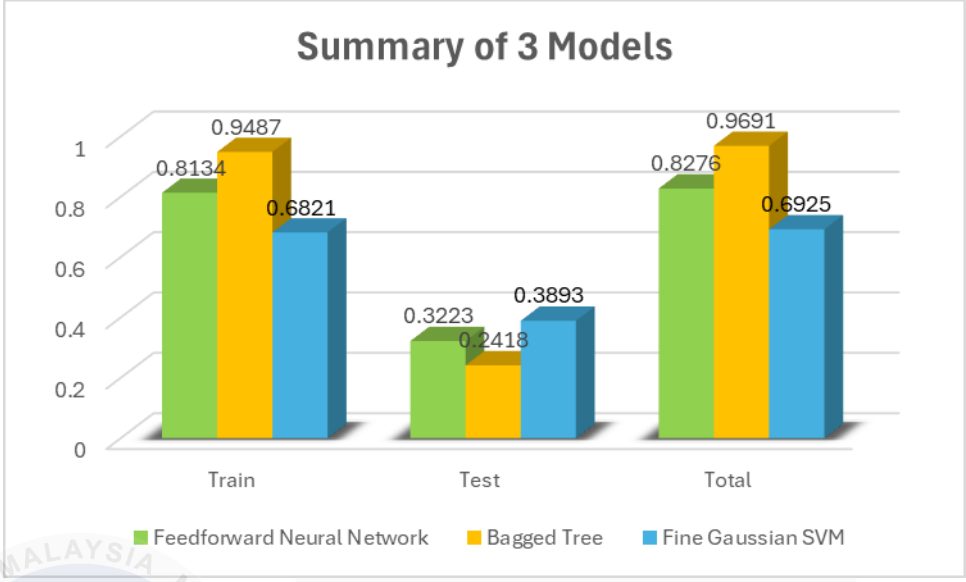


Figure 4.24: Best Results for 3 Models

In regression, a test R value between 0.7 and 0.9 is typically considered sufficient for real-world applications. As shown in the graph, the training data for FFNN (0.8134) and bagged tree (0.9487) models aligns closely with the actual data compared to the fine Gaussian SVM (0.6821). However, on test data, the fine Gaussian SVM generalizes better (0.3893), while FFNN (0.3223) and bagged tree (0.2418) struggle to perform as well. This is a common challenge in machine learning, where a model may fit the training data well but fail to make accurate predictions on unseen test data due to overfitting.

The test data results seem relatively low in this case, but it's important to understand the context. In Figure 4.10, Figure 4.13, and Figure 4.17, the validation R values are 0.70 for FFNN, 0.72 for the bagged tree, and 0.35 for fine Gaussian SVM. The validation data was randomly selected from the training data which leads to better results compared to the test data. The test data on the other hand, consists of the last

week of the entire dataset. As shown in Figure 4.3, the total positive time-lag Pearson correlation over a week, and in Figure 4.5, the VIF of IWV-only data with forwarded hours, the linear correlation decreases significantly after one week. This indicates that using test data from the last week could lead to poor prediction results.

Why don’t just pick randomly from the data for the test data? While in time series regression, it is generally better to test the model on the last part of the data, also called the "holdout" or "out-of-sample" set, rather than randomly picking from the entire dataset. This approach mimics how the model will perform in the real world, where future data is used to make predictions, and ensures the model is tested on unseen data that follows the temporal order. Randomly picking test data can lead to data leakage, where the model might have access to future information during training, which is not realistic for time series problems. By reserving the last portion of the data for testing, you preserve the chronological integrity of the dataset.

Given that the data only covers a six-month period from 23/10/2019 to 09/03/2020 and was collected during Malaysia's Northeast Monsoon season (November to March) which often brings unpredictable weather, the results are understandable even if they are not ideal. However, the model's performance could be improved by gathering more data, ideally over a year or more to capture more trends and better account for variations in the weather.

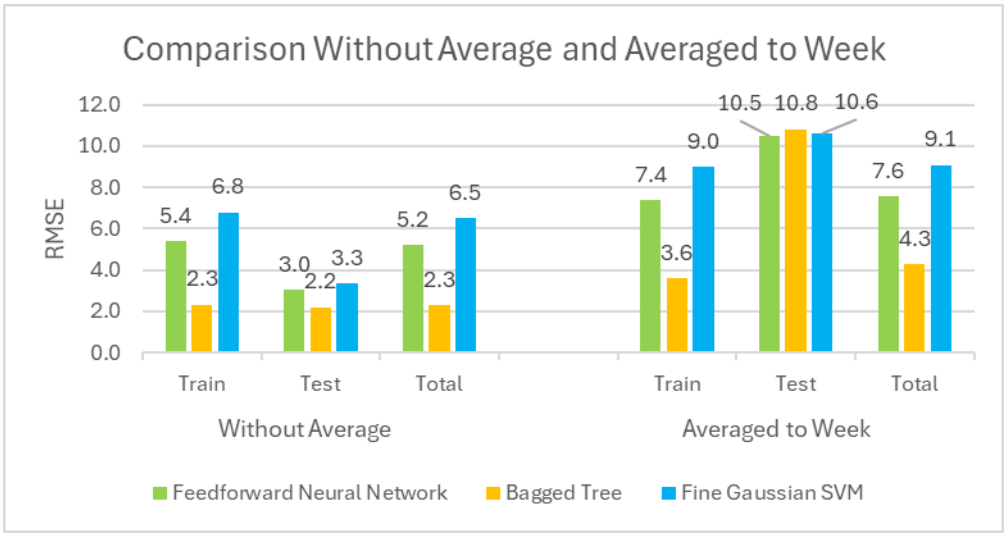


Figure 4.25: Comparison Without Average and Averaged to Week

Figure 4.25 suggests that averaging the output (IWV) over a longer period such as a week could lead to better results. It shows a potential that predicting IWV in a longer term such as weeks instead of minutes will produce a better result. However, it needs to retrain and redo the project for week intervals. The RMSE is used in this graph instead of R is because after averaged to weeks, it left too few data points to effectively assess R.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

This chapter discussed about conclusion and future work recommendation for the project Machine Learning-based Prediction of Integrated Water Vapor using Meteorological Data.

5.1 Conclusion

This study explored the prediction of Integrated Water Vapor (IWV) using machine learning models and derived IWV from Zenith Total Delay (ZTD) obtained via RTKLIB, applying the Saastamoinen model to separate Zenith Hydrostatic Delay (ZHD) and Zenith Wet Delay (ZWD). Models like Feed-Forward Neural Networks (FFNN), Bagged Trees, and Fine Gaussian SVM were used to predict IWV from meteorological inputs.

The FFNN and Bagged Trees achieved high R values during training, at 0.81 and 0.95, respectively, but struggled to generalize to test data. Fine Gaussian SVM, while achieving a lower training R value of 0.68, demonstrated better generalization with an R value of 0.39 on unseen test data. The application of moving average filters improved the overall model output by 13%, effectively smoothing high-frequency noise and enhancing prediction stability. Data preprocessing, feature optimization, and time-lag analysis contributed to improved performance. Weekly averaged IWV predictions demonstrated lower RMSE, suggesting that longer-term forecasting is more robust.

5.2 Future Work

For future work, extending the dataset used in this study is a key step to improving the model's accuracy. The dataset currently spans only a few months, and collecting data over a longer period, ideally covering at least one full year, would provide a more comprehensive understanding of seasonal variations and trends, which would enhance the model's generalization. In addition, while weekly intervals showed promising results for IWV prediction, further investigation into different time intervals such as

daily or monthly could help determine the most optimal time resolution for accurate forecasting.

Exploring alternative machine learning techniques, such as deep learning models or ensemble methods, could potentially lead to even better predictive performance. Expanding the set of features used in the models is also crucial. Including additional meteorological data, such as wind speed, solar radiation, or cloud cover, could provide more context for predicting IWV and help refine the model's accuracy.

REFERENCES

[1] Bai Jadala, N., Sridhar, M., Venkata Ratnam, D., & Dutta, G. (2022). Assessment of machine learning techniques for prediction of integrated water vapor using meteorological data. *Vietnam Journal of Earth Sciences*. https://doi.org/10.15625/2615-9783/17373

[2] Buehler, S. A., Östman, S., Melsheimer, C., Holl, G., Eliasson, S., John, V. O., Blumenstock, T., Hase, F., Elgered, G., Raffalski, U., Nasuno, T., Satoh, M., Milz, M., and Mendrok, J.(2012) A multi-instrument comparison of integrated water vapour measurements at a high latitude site, Atmos. Chem. Phys., 12, 10925–10943, https://doi.org/10.5194/acp-12-10925-2012

[3] Ning, T. *et al.* (2016) 'The uncertainty of the atmospheric integrated water vapour estimated from GNSS observations', *Atmospheric Measurement Techniques*, 9(1), pp. 79–92. doi:10.5194/amt-9-79-2016.

[4] Wang, D., Yuan, P., Kutterer, H. (2024). Real-Time GNSS Integrated Water Vapor Sensing Based on Time Series Correction Deep Learning Models. In: International Association of Geodesy Symposia. Springer, Berlin, Heidelberg. https://doi.org/10.1007/1345_2024_273

[5] Stephens, G. (2005) 'The role of water vapor and clouds in the climate system', *Encyclopedia of Hydrological Sciences* [Preprint]. doi:10.1002/0470848944.hsa206.

[6] Mária, P. (n.d.). Product Tutorial on TPW Content Products.

[7] Rannat, K., Keernik, H., & Madonna, F. (2023). *The Novel Copernicus Global Dataset of Atmospheric Total Water Vapour Content with Related Uncertainties from GNSS Observations*. Remote Sensing, 15(21), 5150. https://doi.org/10.3390/rs15215150

[8] Chen, L., Han, B., Wang, X., Zhao, J., & Yang, Z. (2023). *Machine Learning Methods in Weather and Climate Applications: A Survey*. Applied Sciences, 13(21), 12019. https://doi.org/10.3390/app132112019

[9] Bochenek, B., & Ustrnul, Z. (2022). *Machine Learning in Weather Prediction and Climate Analyses—Applications and Perspectives*. Atmosphere, 13(2), 180. https://doi.org/10.3390/atmos13020180

[10] Zhang, H., Liu, Y., Zhang, C., & Li, N. (2025). *Machine Learning Methods for Weather Forecasting: A Survey*. Atmosphere, 16(1), 82. https://doi.org/10.3390/atmos16010082

[11] Bisht, D. S., Rao, T. N., Rao, N. R., Chandrakanth, S. V., & Sharma, A. (2022). Prediction of integrated water vapor using a machine learning technique. *IEEE Geoscience and Remote Sensing Letters, 19,* 1–5. https://doi.org/10.1109/lgrs.2022.3217094

[12] Jadala, N. B., Sridhar, M., Ratnam, D. V., & Tummala, S. N. M. (2023).

Ensemble based deep learning model for prediction of integrated water vapor (IWV) using GPS and meteorological observations. Journal of Applied Geodesy, 18(2), 253–265. https://doi.org/10.1515/jag-2023-0053

[13] Izanlou, S., Amerian, Y., & Seyed Mousavi, S. M. (2023). GNSS-derived precipitable water vapor modeling using machine learning methods. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, X-4/W1-2022*, 307–313. https://doi.org/10.5194/isprs-annals-x-4-w1-2022-307-2023

[14] Zheng, Y., Lu, C., Wu, Z., Liao, J., Zhang, Y., & Wang, Q. (2022). Machine learning-based model for real-time GNSS precipitable water vapor sensing. *Geophysical Research Letters*, *49*(3). https://doi.org/10.1029/2021gl096408

[15] Huang, Q., Wang, X., Li, H., Zhang, J., Liu, D., & Zhang, H. (2024). *The Zenith Total Delay Combination of International GNSS Service Repro3 and the Analysis of Its Precision*. Remote Sensing, 16(20), 3885. https://doi.org/10.3390/rs16203885

[16] Xia, S., Jin, S., & Jin, X. (2023). *Estimation and Evaluation of Zenith Tropospheric Delay from Single and Multiple GNSS Observations*. Remote Sensing, 15(23), 5457. https://doi.org/10.3390/rs15235457

[17] Bennitt, G. V., & Jupp, A. (2012). *Operational Assimilation of GPS Zenith Total Delay Observations into the Met Office Numerical Weather Prediction Models*. Monthly Weather Review, 140(8), 2706-2719. https://doi.org/10.1175/MWR-D-11-00156.1

[18] Andreas Vollrath (2016) *Towards the operational generation of Advanced-DInSAR time-series products suited for the integration into seismic hazard assessments* [Preprint].

[19] Bevis, M., S. Businger, T. Herring, C. Rocken, R. Anthes,
and R. Ware, 1992: GPS meteorology: Remote sensing of atmospheric water vapor using the Global Positioning System. *J. Geophys. Res.*, 97, 15 787–15 801.

[20] Poli, P., and Coauthors, 2007: Forecast impact studies of zenith total delay data from European near real-time GPS stations in Météo France 4DVAR. *J. Geophys. Res.*, 112, D06114, doi:10.1029/2006JD007430.

[21] Yan, X., V. Ducrocq, P. Poli, M. Hakam, G. Jaubert,
and A. Walpersdorf, 2009: Impact of GPS zenith delay assimilation on convective-scale prediction of Mediterranean heavy rainfall. *J. Geophys. Res.*, 114, D03104, doi:10.1029/2008JD011036.

[22] Boniface, K., and Coauthors, 2009: Impact of high-resolution data assimilation of GPS zenith delay on Mediterranean heavy rainfall forecasting. *Ann. Geophys.*, **27**, 2739–2753.

[23] Bevis, M. *et al.* (1994) 'GPS meteorology: Mapping Zenith wet delays onto precipitable water', *Journal of Applied Meteorology*, 33(3), pp. 379–386. doi:10.1175/1520-0450(1994)033<0379:gmmzwd>2.0.co;2.

[24] Yang, F., Meng, X., Guo, J., Yuan, D., & Chen, M. (2021). *Development and evaluation of the refined zenith tropospheric delay (ZTD) models*. Satellite

Navigation, 2(1), 21. https://doi.org/10.1186/s43020-021-00052-0

[25] Nzelibe, I. U., Tata, H., & Idowu, T. O. (2023). *Assessment of GNSS zenith tropospheric delay responses to atmospheric variables derived from ERA5 data over Nigeria*. Satellite Navigation, 4(1), 15. https://doi.org/10.1186/s43020-023-00104-7

[26] Kačmařík, M. *et al.* (2017) 'Inter-technique validation of tropospheric slant total delays', *Atmospheric Measurement Techniques*, 10(6), pp. 2183–2208. doi:10.5194/amt-10-2183-2017.

[27] Chen, G., & Herring, T. (1997). *Effects of atmospheric azimuthal asymmetry on the analysis of space geodetic data*. Journal of Geophysical Research, 102, 20489-20502. https://doi.org/10.1029/97JB01739

[28] Li, H., Li, X., & Xiao, J. (2023). Estimating GNSS Satellite Clock Error to Provide a New Final Product and Real-Time Services. *GPS Solutions*, 28(17). https://doi.org/10.1007/s10291-023-01558-7

[29] Maciuk, K., Varna, I., & Krzykowska-Piotrowska, K. (2024). A Study of Outliers in GNSS Clock Products. *Sensors*, 24(3), 799. https://doi.org/10.3390/s24030799

[30] Baldysz, Z., Nykiel, G., Figurski, M., & Araszkiewicz, A. (2018). Assessment of the impact of GNSS processing strategies on the long-term parameters of 20 years IWV time series. *Remote Sensing*, *10*(4), 496. https://doi.org/10.3390/rs10040496

[31] Musa, T. A., Amir, S., Othman, R., Ses, S., Omar, K., Abdullah, K., Lim, S., &

Rizos, C. (2011). *GPS meteorology in a low-latitude region: Remote sensing of atmospheric water vapor over the Malaysian Peninsula*. Journal of Atmospheric and Solar-Terrestrial Physics. Elsevier. https://doi.org/10.1016/j.jastp.2011.08.014

[32] Wang, J., Zhang, L., Dai, A., Van Hove, T., & Van Baelen, J. (2007). *A near-global, 2-hourly data set of atmospheric precipitable water from ground-based GPS measurements*. Journal of Geophysical Research: Atmospheres, 112(D11). https://doi.org/10.1029/2006JD007529

[33] Tawfiq, L. N. M. (2015). *Improve Levenberg-Marquardt Training Algorithm for Feed Forward Neural Networks*. International Journal of Modern Engineering Sciences, 4(1), 14-21. Modern Scientific Press. https://www.researchgate.net/publication/329164372_Improve_Levenberg-Marquardt_Training_Algorithm_for_Feed_Forward_Neural_Networks

[34] Candia Jr., J., Adonis, A. M., & Perlas, J. (2024). *Optimizing Bagged Trees in an Ensemble Classifier for Improved Prediction of Diabetes Prevalence in Women*. Pertanika Journal of Science & Technology, 32(4), 1753-1764. Retrieved from http://www.pertanika.upm.edu.my/resources/files/Pertanika%20PAPERS/JST%20Vol.%2032%20%284%29%20Jul.%202024/16%20JST-4343-2023.pdf

[35] Fischetti, M. (2015). *Fast training of Support Vector Machines with Gaussian kernel*. DEI, University of Padova. Retrieved from https://www.dei.unipd.it/~fisch/papers/fastSVMtraining.pdf

[36] Just, A. C., Liu, Y., Sorek-Hamer, M., Rush, J., Dorman, M., Chatfield, R.,

Wang, Y., Lyapustin, A., & Kloog, I. (2020). Gradient boosting machine learning to improve satellite-derived column water vapor measurement error. Atmospheric Measurement Techniques, 13(9), 4669–4681. https://doi.org/10.5194/amt-13-4669-2020

[37] Meenal, R., Michael, P. A., Pamela, D., & Rajasekaran, E. (2021). Weather prediction using random forest machine learning model. Indonesian Journal of Electrical Engineering and Computer Science, 22(2), 1208. https://doi.org/10.11591/ijeecs.v22.i2.pp1208-1215

[38] T. Takasu. (2013). *RTKLIB ver. 2.4.2 Manual*. Retrieved from https://www.rtklib.com/prog/manual_2.4.2.pdf

[39] Gurtner, W., & Estey, L. (2007). *RINEX: The Receiver Independent Exchange Format Version 3.00*. Retrieved from https://files.igs.org/pub/data/format/rinex300.pdf

[40] Boda, A. (2019, January 31). *What is ambiguity resolution?* Aaron Boda. Retrieved from https://aaronboda.wordpress.com/2019/01/31/what-is-ambiguity-resolution/

[41] Pérez-Ruiz, M., & Upadhyaya, S. K. (2012). GNSS in precision agricultural operation. In F. Boukour Elbahhar & A. Rivenq (Eds.), *New approach of indoor and outdoor localization systems* (pp. xx-xx). DOI: 10.13140/2.1.4162.3362

[42] Yao, Y., & Hu, Y. (2018). *An empirical zenith wet delay correction model using piecewise height functions*. Annals of Geophysics, 36(6), 1507-1519. doi:10.5194/angeo-36-1507-2018.

[43] Zhou, F., Li, L., Wang, Y., Dai, Z., Ding, C., & Li, H. (2024). *Analysis of Different Height Correction Models for Tropospheric Delay Grid Products over the Yunnan Mountains*. Atmosphere, 15(8), 872. doi: 10.3390/atmos15080872.

[44] Jiang, C., Gao, X., Zhu, H., Wang, S., Liu, S., Chen, S., & Liu, G. (2024). *An improved global pressure and zenith wet delay model with optimized vertical correction considering spatiotemporal variability*. Geoscientific Model Development, 17, 5939-5959. doi:10.5194/gmd-17-5939-2024.

[45] Musa, T.A., Mazlan, M.H., Opaluwa, Y.D., Musliman, I.A., & Radzi, Z.M. (2017). Water Vapour Weighted Mean Temperature Model for GPS-Derived Integrated Water Vapour in Peninsular Malaysia. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-4/W5*, GGT 2017, 4 October 2017, Kuala Lumpur, Malaysia.

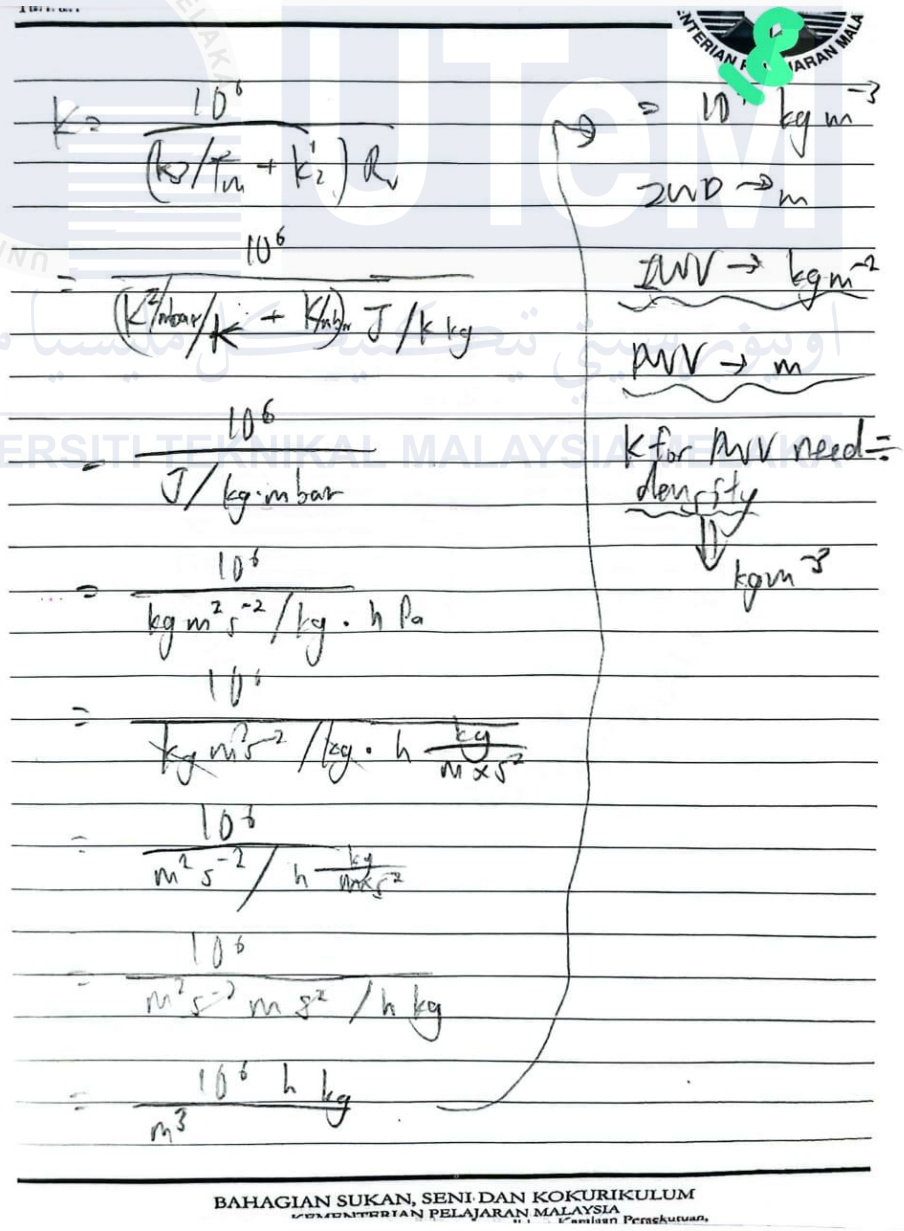
LIST OF PUBLICATIONS AND PAPERS PRESENTED

Published works as well as papers presented at conferences, seminars, symposiums etc pertaining to the research topic of the research report/ dissertation/ thesis are suggested be included in this section. The first page of the article may also be appended as reference.

APPENDICES

Appendix A

Draft of Unit Prove



Appendix B

Full Result of Each Parameters

Feedforward Neural Network

Table 1: Result Without Filter

Tuned Hyperparameter (neurons and layers)	Train/Validation/ Test	MSE	RMSE	MAE	R-squared	Adjusted R
10 Train time: 0:50 Model size: 5 KB	Train	100.5940	10.0297	7.6629	0.5690	0.3237
	Test	125.0350	11.1819	9.5863	0.1573	0.0245
	Total	101.8338	10.0913	7.7604	0.5571	0.3103
80 Train time: 10:17 Model size: 12 KB	Train	83.5697	9.1416	6.8281	0.6620	0.4382
	Test	143.3316	11.9721	10.2708	0.1293	0.0164
	Total	86.6011	9.3060	7.0027	0.6433	0.4138
80+20 Train time: 19:27 Model size: 36 KB	Train	76.9800	8.7738	6.4827	0.6946	0.4825
	Test	143.8969	11.9957	10.0765	0.1218	0.0145
	Total	80.3744	8.9652	6.6650	0.6753	0.4560

Table 2: Result With IWV Forwarded 4 Hours for Filter

Tuned Hyperparameter (neurons and layers)	Train/Validation/ Test	MSE	RMSE	MAE	R-squared	Adjusted R
10 Train time: 0:36 Model size: 5 KB	Train	97.2987	9.8640	7.5205	0.5886	0.3465
	Test	118.9508	10.9065	8.9631	0.2141	0.0455
	Total	98.3983	9.9196	7.5937	0.5780	0.3341
80 Train time: 10:10 Model size: 12 KB	Train	81.0764	9.0042	6.7638	0.6749	0.4554
	Test	145.8371	12.0763	9.6318	0.1224	0.0147
	Total	84.3653	9.1851	6.9094	0.6553	0.4294
80+20	Train	74.0609	8.6059	6.3700	0.7089	0.5025

Train time: 46:55	Test	144.8649	12.0360	9.7504	0.1551	0.0238
Model size: 36 KB	Total	77.6568	8.8123	6.5417	0.6891	0.4749

Table 3: Result After 8 Hours Filter

Tuned Hyperparameter (neurons and layers)	Train/Validation/ Test	MSE	RMSE	MAE	R-squared	Adjusted R
10	Train	86.8094	9.3172	7.1662	0.6666	0.4443
Train time: 0:36	Test	115.4286	10.7438	9.1397	0.1857	0.0342
Model size: 5 KB	Total	88.2611	9.3947	7.2663	0.6548	0.4287
80	Train	63.6928	7.9808	6.0098	0.7871	0.6196
Train time: 10:10	Test	113.6802	10.6621	8.8472	0.2521	0.0633
Model size: 12 KB	Total	66.2284	8.1381	6.1537	0.7727	0.5970
80+20	Train	54.7274	7.3978	5.5224	0.8276	0.6849
Train time: 46:55	Test	109.6800	10.4728	8.8256	0.3223	0.1036
Model size: 36 KB	Total	57.5148	7.5839	5.6899	0.8134	0.6616

Bagged Tree

Table 4: Result Without Filter

Tuned Hyperparameter (number of learners)	Train/Validation/ Test	MSE	RMSE	MAE	R-squared	Adjusted R
30	Train	23.7084	4.8691	2.8635	0.9193	0.8452
Train time: 0:40	Test	159.2879	12.6209	10.1058	0.1354	0.0180
Model size: 108.215 MB	Total	30.5856	5.5304	3.2309	0.8916	0.7949
70	Train	23.3786	4.8351	2.8478	0.9207	0.8477
Train time: 1:34	Test	158.0985	12.5737	10.0966	0.1412	0.0196
Model size: 243.209 MB	Total	30.2122	5.4966	3.2155	0.8931	0.7977

100 Train time: 2:11 Model size: 344.577 MB	Train	23.2921	4.8262	2.8442	0.9211	0.8484
	Test	158.3648	12.5843	10.1190	0.1396	0.0192
	Total	30.1437	5.4903	3.2132	0.8934	0.7982

Table 5: Result With IWV Forwarded 1 Hours for Filter

Tuned Hyperparameter (number of learners)	Train/Validation/ Test	MSE	RMSE	MAE	R-squared	Adjusted R
30 Train time: 1:00 Model size: 108.439 MB	Train	23.9912	4.8981	2.8806	0.9182	0.8431
	Test	158.1799	12.5770	10.1351	0.1558	0.0240
	Total	30.8000	5.5498	3.2487	0.8907	0.7934
70 Train time: 2:03 Model size: 243.804 MB	Train	23.5503	4.8529	2.8571	0.9201	0.8465
	Test	156.9300	12.5272	10.1148	0.1587	0.0249
	Total	30.3180	5.5062	3.2253	0.8928	0.7970
100 Train time: 2:14 Model size: 345.010 MB	Train	23.5332	4.8511	2.8524	0.9201	0.8466
	Test	156.8918	12.5256	10.0807	0.1597	0.0252
	Total	30.2998	5.5045	3.2191	0.8928	0.7971

Table 6: Result After 2 Hours Filter

Tuned Hyperparameter (number of learners)	Train/Validation/ Test	MSE	RMSE	MAE	R-squared	Adjusted R
30 Train time: 1:00 Model size: 108.439 MB	Train	13.0651	3.6146	2.4327	0.9690	0.9390
	Test	117.3263	10.8317	9.1820	0.2390	0.0568
	Total	18.3537	4.2841	2.7750	0.9485	0.8997
70 Train time: 2:03 Model size: 243.804 MB	Train	13.0400	3.6111	2.4290	0.9691	0.9391
	Test	117.4737	10.8385	9.1997	0.2399	0.0573
	Total	18.3373	4.2822	2.7724	0.9486	0.8998
100	Train	13.0268	3.6093	2.4268	0.9691	0.9392

Train time: 2:14 Model size: 345.010 MB	Test	117.2282	10.8272	9.1518	0.2418	0.0582
	Total	18.3123	4.2793	2.7679	0.9487	0.9000

Fine Gaussian SVM

Table 7: Result Without Filter, Forwarded, and Filtered

Model	Train/Validation/Test	MSE	RMSE	MAE	R-squared	Adjusted R
Normal Train time: 11:44 Model size: 12.063 MB	Train	96.7351	9.8354	7.2931	0.5932	0.3519
	Test	127.3082	11.2831	9.5791	0.2369	0.0558
	Total	98.2859	9.9139	7.4091	0.5819	0.3386
Forwarded 4hr Train time: 8:17 Model size: 12.143 MB	Train	94.9081	9.7421	7.1958	0.6037	0.3645
	Test	127.7591	11.3031	9.3988	0.2589	0.0668
	Total	96.5765	9.8273	7.3077	0.5918	0.3503
Filtered Train time: 8:17 Model size: 12.143 MB	Train	80.8273	8.9904	6.7375	0.6925	0.4795
	Test	113.2843	10.6435	9.0489	0.3893	0.1513
	Total	82.4737	9.0815	6.8547	0.6821	0.4652

Best Result of 3 Models

Table 8: Best Result of 3 Models

Models	Train/Validation/Test	MSE	RMSE	MAE	R-squared	Adjusted R
Feedforward NN Train time: 46:55 Model size: 36 KB	Train	54.7274	7.3978	5.5224	0.8276	0.6849
	Test	109.6800	10.4728	8.8256	0.3223	0.1036
	Total	57.5148	7.5839	5.6899	0.8134	0.6616
	Train	13.0268	3.6093	2.4268	0.9691	0.9392

Bagged Tree Train time: 2:14 Model size: 345.010 MB	Test	117.2282	10.8272	9.1518	0.2418	0.0582
	Total	18.3123	4.2793	2.7679	0.9487	0.9000
SVM 8:17 Model size: 12.143 MB	Train	80.8273	8.9904	6.7375	0.6925	0.4795
	Test	113.2843	10.6435	9.0489	0.3893	0.1513
	Total	82.4737	9.0815	6.8547	0.6821	0.4652

After Averaged to Week

Table 9: Result Averaged to Week

Models	Train/Validation/ Test	MSE	RMSE	MAE	R-squared	Adjusted R
Feedforward NN Train time: 46:55 Model size: 36 KB	Train	29.1252	5.3968	4.4048	0.9359	0.8473
	Test	9.2015	3.0334	2.6079	-1.0000	1.0000
	Total	27.0280	5.1988	4.2156	0.9378	0.8553
Bagged Tree Train time: 2:14 Model size: 345.010 MB	Train	5.3593	2.3150	1.8935	0.9950	0.9876
	Test	4.8044	2.1919	1.6299	-1.0000	1.0000
	Total	5.3009	2.3024	1.8657	0.9942	0.9862
Filtered Train time: 8:17 Model size: 12.143 MB	Train	46.1622	6.7943	5.5792	0.8454	0.6489
	Test	11.0794	3.3286	2.8473	-1.0000	1.0000
	Total	42.4693	6.5168	5.2917	0.8571	0.6816

END OF THESIS