MACHINE LEARNING-BASED GPS TEC FORECASTING

LEOW HOCK YEW

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

MACHINE LEARNING-BASED GPS TEC FORECASTING

LEOW HOCK YEW

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

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

2024

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI KEJUTERAAN ELEKTRONIK DAN KEJURUTERAAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Machine Learning Based GPS TEC Forecasting
Sesi Pengajian : 2023/2024

Saya Leow Hock Yew 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) (COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: Q-2-2, Desa permai Indah , Jalan Helang, Sungai Dua, 11700, Gelugor, Pulau Pinang

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

Tarikh : 12/1/24 Tarikh : 12/1/24

*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 GPS TEC Forecasting” is based off the result of my own work except for quotes and references that are as cited in the references section.

Signature :

Author : Leow Hock Yew

Date : 12/1/24

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.

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

Signature : ……………………………………………

Supervisor's Name : Dr Ho Yih Hwa

Date : 12 Jan 2024

DEDICATION

This is to honour my beloved father and mother

ABSTRACT

Precise ionosphere total electron content (TEC) is crucial for various applications such as space weather prediction, satellite communication and navigation, ionosphere scintillation monitoring, and magnetic storm monitoring and so on. Forecasting ionosphere TEC is crucial because by predicting the short-term value of the TEC, it is capable in filling the empty gap between the TEC product latency and increase the precision of the TEC values. In this project, a single type of machine learning model is focused on which is the feed-forward backprop artificial neural network (ANN) and the Levenberg-Marquardt algorithm is used as the training technique. The machine is trained using 12 months of GPS data from FKEKK station together with 12 months of sun activity and magnetic activity from NASA which are the sunspot no, proton ratio, F10.7 index, Kp index, Dst Index, and Ap Index. The MATLAB software is used to train the machine learning algorithm. Net1 which is the neural network model which comprises of all the mentioned solar and geomagnetic input shows the best accuracy with Mean Absolute Error (MAE) of 2.35 TECU, Root Mean Square Error (RMSE) of 3 and R-square of 0.966 which is more accurate than the rest of the neural network models (Net2, Net3, and Net2016). This project applies that Net1 is capable in accurately forecasting the TEC in the southeast Asia region.

ABSTRAK

Ketepatan jumlah elektron keseluruhan (TEC) di ionosfera adalah penting untuk pelbagai aplikasi seperti ramalan cuaca angkasa, pemantauan kerlipan ionosfera, dan pemantauan ribut magnetik dan sebagainya. Meramal TEC ionosfera penting kerana ia berupaya untuk mengisi kesenjangan kosong di antara kelewatan produk TEC dan meningkatkan ketepatan nilai TEC. Dalam artikel ini, tumpuan akan diberikan kepada satu jenis model pembelajaran mesin iaitu rangkaian neural tiruan (ANN) aliran ke depan dengan menggunakan algoritma Levenberg-Marquardt sebagai teknik latihan. Mesin ini dilatih menggunakan data GPS selama 12 bulan dari stesen FKEKK dengan 12 bulan aktiviti matahari dan aktiviti magnetik dari NASA yang merangkumi nombor sunspot, nisbah proton, indeks F10.7, indeks Kp, Indeks Dst, dan Indeks Ap. Algoritma pembelajaran mesin akan dilatih menggunakan perisian MATLAB. Net1, yang merupakan model rangkaian neural yang mengandungi semua input solar dan geomagnetik yang disebutkan, menunjukkan ketepatan terbaik dengan Ralat Mutlak Minima (MAE) sebanyak 2.35 TECU, Ralat Minima Kuasa Dua (RMSE) sebanyak 3, dan R-kuasa dua sebanyak 0.966, yang lebih tepat berbanding model rangkaian neural lain (Net2, Net3, dan Net2016). Kajian ini menyatakan bahawa Net1 berupaya meramalkan TEC dengan tepat di rantau Asia Tenggara.

ACKNOWLEDGEMENTS

This report is dedicated to everyone that has been involved and provides me with aid to finish and make this project a success. Here from the bottom of my heart, I would like to sincerely thank everyone that has lent me a helping.

Firstl of all, I would like to provide my appreciation to my project supervisor Dr. Ho Yih Hwa. He had provided me full guidance towards finishing and completing this final project. Advises and guidance provided from Dr. Ho has become a key component in allowing me to complete this report. The guidance on knowledge and know how on artificial intelligence which provides me severe help on finishing the project.

Next, I would like to thank both my FYP panels, Dr. Azmi bin Awang Md isa and Dr Mawarni Binti Mohamed Yunus for providing me with useful informations and suggestions on changing and making a better presentation. Their advice and suggestions given after my presentation helped me to improve my overall report writing.

Besides that, I would also like to thank my parents, Mr. Leow Kean Tong, and Mrs. Lau Yoke San for always supporting my journey as a student. I would also like to thank my friends and seniors for providing me with aid when I needed it during this project.

Lastly, I would appreciate the time spent on the readers for reading my thesis report.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

ANN	:	Artificial Neural Network
TEC	:	Total Electron Content
GPS	:	Global Positioning System
LM	:	Levenberg Marquardt
ARMA	:	Auto-regressive moving average
ARIMA	:	Auto-regressive integrated moving average
LSTM	:	Long short-term memory
ANFIS	:	Adaptive neuro fuzzy inference system
ML	:	Machine learning
GUI	:	General User
VHF	:	Very high frequency
UHF	:	Ultra-high frequency
HF	:	High frequency
CNN	:	Convolutional Neural Network
RMSE	:	Root Mean Square Error
MAE	:	Mean Average Error
VBA	:	Visual Basic for Application

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Introduced in the early 1990s, the Global Positioning System (GPS) is a navigation system based on satellites orbiting the Earth, is a fully operational system that offers users around the globe accurate three-dimensional positioning and precise time synchronized with global time standards. [1] In order to obtain an accurate GPS reading, there are multiple barriers to go through in order to obtain a high accuracy reading. One of them being the ionosphere.

Ionosphere is a highly dynamic region and changes extremely fast and complex over different times and positions, which causes the inconsistency in acquiring precise ionosphere capacity. [2] Total Electron Content (TEC) is a key parameter that affects the propagation of radio signals, which includes the Global Positioning System signals. Since 1998, the ionosphere working group of the International GNSS Service

(IGS) has been consistently delivering dependable Global Ionosphere Maps (GIMs). However, the prediction value is still highly based off of the high solar activity period as the ionosphere values are highly affected by it as well as the magnetic activity.

Obtaining real-time ionosphere Total Electron Content (TEC) through ionosphere prediction is a crucial approach. Several studies have explored this topic using classical time-series analysis approaches. The auto-regressive moving average (ARMA) model and auto-regressive integrated moving average (ARIMA) model have been extensively applied for short-term TEC forecasting in various latitudes.[3]-[5]

In this project, machine learning algorithms are implied due to its ability to automatically learn the underlying nonlinear relationship between TEC values and external indicators. This capability enhances prediction performance, particularly in extreme environments. Various machine learning methods are introduced for the ionosphere predictions, namely the long short-term memory (LSTM), adaptive neuro-fuzzy inference system (ANFIS), and the gradient boosting decision tree (GBDT). This article will be focusing on the Artificial Neural Network (ANN) machine learning algorithm that mimics the human-brain process.

1.2 Motivation

The motivation on using machine learning for GPS TEC forecasting lies within the potential of machine learning language being capable of addressing the limitations of traditional forecasting methods. One of the first few important point being capable on addressing nonlinear relationships, where machine learning algorithms can capture complex and nonlinear relationships between TEC values and various external factors, such as solar activity, magnetic variations. Unlike traditional models that rely on simplified assumptions, machine learning can uncover intricate patterns that lead to more accurate predictions. The machine learning (ML) approach is also as data driven as it does its predictions based on historical TEC data and parameters. By training this data, ML models can extract useful information and then generalize the pattern, and by doing so, the forecasting accuracy is improved.

Machine learning also has a high adaptability rate as it is able to adjust its predictions accordingly to the dynamic and non-stationary TEC behavior. This is true due to the fact that machine learning is capable of capturing these interactions and considering multiple input features simultaneously, leading to more robust and accurate predictions.

1.3 Problem statement

The ionosphere is a highly dynamic region, and TEC is a key parameter that affects the propagation of radio signals. As the radio signal passes through the ionosphere, they are affected by the presence of electrons which causes signals to refract, scatter or attenuate. As a result, this causes errors in GPS positioning and communication, especially in regions where ionosphere is highly varying. Real-time precise ionosphere TEC monitoring is still underdeveloped, and classical prediction methods based on time series analysis cannot meet the requirements. [2] Addressing that, the main aim of the project would be to provide a high accuracy GPS signal through the trained model. By having a high accuracy GPC TEC forecasting system, fields involving usage of GPS system is supplemented such as improving satellite navigation, enhancing communication systems, monitoring space weather, optimizing aviation operations, enabling precision agriculture, and advancing scientific research related to the ionosphere and space weather.

1.4 Objective

- To investigate how the number of neurons and number of hidden layers affects the output TEC when using artificial neural network machine learning.
- To forecast TEC with solar and magnetic parameters using machine learning.

1.5 Scope of work

- 12 months of GPS data from FKEKK station will be used to estimate TEC.
- 12 months of solar (Sunspot Number, solar flux 10.7, proton ratio) and magnetic (Kp, Dst, and Ap) data from NASA is used as input.
- Pre – process the GPS data obtained from FKEKK station and from NASA website.
- ANN will be used to model the relationship among TEC, solar activities, and magnetic data.
- Neural Network architecture design in MATLAB.
- Model training and validation

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In order to develop and train a machine learning algorithm for GPS TEC forecasting, MATLAB needs to be installed together with an extension to the software which is the MATLAB GUI to access the machine learning section. The literature review covers a wide variety of theories related to the artificial neural network (ANN), machine learning, ionosphere physics, and GPS signal propagation.

2.2 Total Electron Content

TEC, or total electron content, is a crucial parameter used in studying the plasma environment near the Earth in the ionosphere. The ionosphere is a layer of the Earth's atmosphere that extends from about 60 km to 1000 km above the Earth's surface, and it contains a high concentration of ionized gases, or plasma, that can affect the propagation of radio waves and satellite signals. The scientific use dates back to the early days of artificial satellite exploration, and it has provided significant insights into the ionosphere's response to magnetic storms. While numerous reviews exist on solar-terrestrial disturbances, there is a lack of focus on the TEC parameter itself. With the increasing availability of TEC data from the Global Positioning System (GPS), there is a need for a comprehensive summary of pre-GPS studies to establish a foundation for progress in the GPS era. [6]

The ionosphere, from a scientific perspective, is a complex system consisting of neutral plasma that exhibits strong altitude-dependent effects on photochemistry and plasma dynamics. Total electron content (TEC), obtained by integrating the entire electron density profile of the ionosphere, provides a convenient way to represent, assess, and comprehend the overall behavior of the thermal plasma near Earth. The majority of TEC contributions come from the F2 layer, with approximately two-thirds originating from regions above the altitude (H_{max}) where the electron density reaches its peak (N_{max}). If significant changes occur in TEC, simple vertical redistributions of plasma within the F layer cannot be solely responsible for these variations. [7]

From an observational standpoint, measuring TEC is a relatively straightforward process that utilizes radio diagnostics with a trans ionosphere nature. Typically, radio frequencies in the VHF (Very High Frequency) and UHF (Ultra High Frequency) range are employed, which are less susceptible to severe degradation during storms compared to HF (High Frequency) ionosonde. As a result, the crucial time periods of interest for solar-terrestrial physics research are not compromised by the very effects being investigated. The given advantageous characteristic allows for continuous TEC studies using both regional and global networks. These networks provide a wealth of data, enabling researchers to conduct TEC investigations on an essentially uninterrupted basis. [8]

The utilization of TEC derived from the Global Positioning System (GPS) has brought about a revolution in ionosphere physics, similar to the impact of ionosonde in the 1930s and incoherent scatter radars in the 1960s. Through global and regional networks of GPS receivers, TEC data can be obtained over extensive geographic areas, with the exception of large oceanic regions where ground-based GPS receivers face deployment challenges. However, TEC measurements can now be acquired from virtually anywhere on the planet, often in near-real time. [9]

The impact of GPS-based research on ionosphere science is still in its early stages, and the use of the TEC parameter is expected to advance three primary focal points of contemporary ionosphere science: understanding ionosphere structure, developing forecasting capabilities, and employing data-assimilative modelling. From an applications perspective, TEC and its spatial and temporal variabilities can significantly influence radio communications and navigation systems, which rely on both "now-casting" and forecasting capabilities. To ensure that new GPS studies make substantial contributions, it is crucial to build upon existing research rather than duplicating past work. This approach ensures the efficient use of valuable time and expertise. By doing so, the potential benefits of GPS-based studies in ionosphere research and applications can be maximized. [9]

2.3 TEC extraction from GPS data

TEC data is collected for a total of 1 year time span on the year 2022 using the GPS station in UTeM (2.3138° N, 102.3211° E). Due to the ionosphere being 70-1000km above the earth's surface, a time delay is bound to occur, where GPS at L1 on frequency of 1575.42MHz and L2 on frequency of 1227.60 MHz will both undergo difference delays. The GPS signal of frequency f, experiences delay in time t, and is shown in the below equation 1.

$$t = 40.3 \times \frac{TEC}{cf^2}$$

In the equation above, c represents speed of light in space. Changes in Total Electron Content (TEC) introduce alterations to both the phase and group velocities of signals transmitted from the satellite. The cumulative quantity of free electrons within a 1 m² cross-sectional area along the path of the electromagnetic wave connecting the satellite to the receiver is termed Slant Total Electron Content (STEC). STEC is measured in TEC units (TECU), with 1 TECU equivalent to 10^{16} electrons per square meter. The formula expressing STEC is provided by the below equation 2.

$$STEC = \frac{1}{40.3}\left(\frac{f_1^2 f_1^2}{f_1^2 - f_1^2}\right)(P_1 - P_2)$$

In this context, P1 and P2 represent pseudo range measurements associated with L1 and L2 signals, while f1 and f2 represent the respective high and low frequencies of the GPS signals. The TEC derived from pseudo range measurements tends to have inherent noise. Vertical Total Electron Content (VTEC) is determined from the Slant Total Electron Content (STEC) using the below equation 3, as outlined in the work of Sahu et al. (2020).

$$VTEC = (STEC - [b_R + b_S])/S(E)$$

In this expression, where S(E) denotes the obliquity factor at the ionospheric pierce point (IPP) with a zenith angle z, E represents the elevation angle of the satellite in degrees, and VTEC is the vertical Total Electron Content at the IPP. The biases for the satellite and receiver are denoted as $b_S$and b_R, respectively. The obliquity factor S(E) is mathematically defined by the following equation 4:

$$S(E) = \frac{1}{\cos(z)} = \{1 - (\frac{R_E \times \cos(E)}{R_E h_s})^2\}^{-0.5}$$

In the above equation 4, z is the zenith angle of the satellite, next $R_E$is denoted as the mean Earth radius, which is 6371km, h_S is the ionospheric shell height which holds an approximate value of 350 km and lastly E is the elevation angle. These are the equations involved in the GOPI-SEEMALA software in order to obtain the VTEC and STEC data in the ".cmn" file. [10]

2.4 Solar Activity

In scientific discourse, the term "solar activity" can be interpreted in various ways. Initially, it pertains to the degree of solar activity, signifying the strength of solar electromagnetic radiation, notably in solar X-rays and the extreme ultraviolet (EUV) spectrum (XUV). Another interpretation involves specific occurrences related to solar activity, such as coronal mass ejections and events involving solar protons.

Solar radiation is absorbed by the Earth's upper atmosphere, leading to processes such as heating, dissociation, and ionization. The ionosphere primarily forms through the ionization effects induced by solar Extreme Ultraviolet (XUV) radiation. It is well-established that solar XUV exhibits regular and irregular fluctuations across various timescales, ranging from minutes (associated with solar flares) and roughly 27 days (corresponding to solar rotation) to decades (linked to the

11-year solar cycle). These fluctuations can have amplitudes varying by more than 1000 times, with greater variability typically observed at shorter wavelengths.

The variability in solar activity has profound effects on several parameters within the ionosphere, including neutral density, temperature, ion and electron densities, ionospheric temperatures, neutral winds, and electric fields. The fluctuations initiated by solar activity result in significant variations in these ionospheric characteristics. [11]

2.4.1 Sunspot number (SSN)

The sunspot number is a measure of the solar activity on the Sun's surface, specifically the number of sunspots visible at any given time. Sunspots are temporary phenomena on the Sun's photosphere that appear darker than the surrounding areas due to the intense magnetic activity occurring in those regions.

The sunspot number is calculated based on a formula that considers the number of sunspot groups and individual sunspots visible on the solar disk. The formula was originally developed by Rudolf Wolf in the 19th century. Where the sunspot number equation, R_z is derived as:

$$R_z = k(10_g + f)$$

g is the number of sunspot group, f is the number of a single sunspot, and k is the correction factor for each observer, Since sunspot is observed through naked eye in the 19th century. [12]

A higher sunspot number generally indicates higher solar activity, as sunspots are associated with increased magnetic activity. Solar activity follows an approximately

11-year cycle known as the solar cycle, and the sunspot number tends to peak during periods of high solar activity and decrease during periods of low solar activity.

2.5 Geomagnetic Activity

Geomagnetic storms can lead to significant impacts on Earth, including the disruption of satellite communications, interference with the detection and tracking of aircraft, disturbance in navigation systems, and, in rare instances, interruptions in the flow of electrical energy across power grids. Geomagnetic variation encompasses both quiet variations, characterized by regular patterns mainly originating from solar electromagnetic radiation, and geomagnetic disturbance, displaying irregular patterns primarily driven by the solar wind. The Kp index, or Kp for brevity, serves the purpose of globally monitoring sub-auroral geomagnetic disturbance. [13]

2.5.1 Kp Index

Changes in the solar wind induce variations in electric currents within the magnetosphere and ionosphere. These alterations in the magnetic field can be monitored using ground-based magnetometers. Various geomagnetic indices, such as the Kp index, are utilized to track the intensity of geomagnetic disturbance associated with fluctuations in the solar wind (Mayaud, 1980).

The Kp index, ranging from 0 to 9 and expressed in units of thirds, is a widely adopted measure of geomagnetic activity, with higher values indicating more severe geomagnetic storms. Derived from observations at magnetic observatories worldwide, standardized measurements are consolidated to generate a singular, global Kp index value. Computed over a 3-hour span of geomagnetic data, the Kp index is updated every 3 hours. Due to its correlation with numerous phenomena in near-Earth space, the Kp index is valuable for parametrizing empirical Geo-space models.

Furthermore, the Kp index plays a crucial role in describing and predicting scintillations in signals from global navigational systems, especially at high latitudes. Studies, such as the work by Secan et al. (1997), utilize the Kp index to forecast the intensity, latitude, and local time of irregularities with the Wideband Ionospheric Scintillation Model. The effectiveness of the Kp index is evident in its high skill and impact on machine learning-based predictions of high-latitude scintillations (e.g., McGranaghan et al., 2018). [13]

2.6 Machine Learning based TEC forecasting.

Machine learning (ML) is a field of study that focuses on enabling machines to handle data more efficiently by teaching them algorithms and models. In some cases, when we encounter data which are difficult to interpret or extract information from, machine learning techniques can be applied. The increasing availability of large datasets has led to a growing demand for machine learning across various industries, where it is used to extract valuable insights from the data. [14]

The primary goal of machine learning is to enable machines to learn from data. Researchers have explored numerous approaches to develop algorithms and models that allow machines to learn autonomously without explicit programming. This area of study attracts mathematicians and programmers who work on finding solutions to the challenges posed by large datasets.[14]

The most common machine learning that is used in GPS TEC forecasting are the four machine learning algorithms which are the Neural Network approach, Long-Short Term memory (LSTM) approach, the adaptive neuro-fuzzy interference system (ANFIS), support vector machines (SVMs), random forests (RFs), convolutional neural network (CNN) and the Gradient Boosting Decision Tree (GBDT).

2.6.1 Neural Network

Neural networks have gained recognition as a robust and elegant method for approximating complex nonlinear functions by representing them as compositions of elementary nonlinear functions. [15] The neural network model has demonstrated successful applications in TEC modeling and forecasting. By adjusting the weights between network nodes, the neural network (NN) establishes the nonlinear relationship between input and output. A trained network inherently captures the implicit nonlinear relationship between input and output, enabling it to make predictions for ionosphere conditions using only input features. Figure 2.1 shows the neural network structure with one hidden layer.

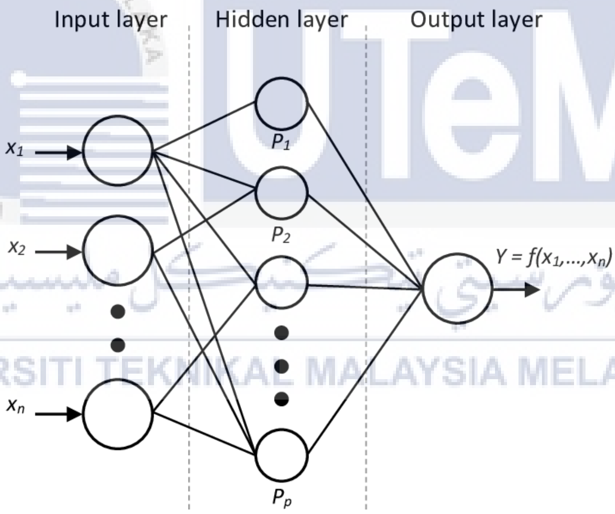

Figure 2.1: Neural Network Structure

The output of the network can also be expressed as the following equation:

$$y = k\left(\sum_{j=1}^{n}(w_j x_j + b)\right)$$

This is where y is output which is the predicted TEC value. K() is the activation function, while x is the input features related in order to predict TEC values, for example, sun activity and magnetic activity. With sufficient input set of data, the relationship between the parameters can be forecasted.

TEC forecasting using neural network methods has been previously conducted in various locations worldwide, including South Africa, Brazil, India, Cyprus, Malaysia, Japan, China, and the U.K. In Malaysia, the paper by [3] shows a previous study explored on the feasibility of using neural networks to predict TEC In the period of low to medium solar activity, utilizing data from 2005 to 2010 from a single station, it was found that the neural network demonstrated promise as a viable tool for TEC value prediction. The results indicated acceptable levels of RMSE, absolute and relative errors, as well as a favourable coefficient of correlation in specific network configurations. But the study also shows that increasing in number of neurons and hidden layers had minimal impact on the neural network's ability.

2.6.1.1 Levenberg–Marquardt algorithm

In terms of mathematics and computing, the Levenberg–Marquardt algorithm (commonly referred to as LMA or simply LM) is employed to tackle non-linear least squares problems which is also recognized as the damped least-squares (DLS) method and proves high reliance in solving minimization problems, particularly those encountered in least squares curve fitting. The basis of LM algorithm is a linear approximation to "f" in term of p. Using a small $||\delta_p||$, a Taylor series expansion will lead to the below approximation which is equation 1.

$$f(\boldsymbol{p} + \delta_p) \approx f(\boldsymbol{p}) + \boldsymbol{J}\delta_p$$

From the equation above, J is the Jacobian matrix $\frac{\partial f(p)}{\partial p}$. Bolded letters are used for denoting transposition. f is a functional relation which directs a parameter vector $p \in R^m$ to predict the measurement vector $\hat{x} \in R^2$. LM is a type of non-linear optimization method that is iterative that produces a series of vectors, example $p_0, p_1, p_2 \ldots,$ that leads to the local minimizer p^+ for f. The ideal $J\delta_p - \in$ is orthogonal to the column space of J which ultimately leads to $J^T(\boldsymbol{J}\delta_p - \in) = 0$, which then makes δ_p as the solution of equation 1.

$$J^T\boldsymbol{J}\delta_p = J^T \in$$

Above shows equation 2 and the matrix $J^T\boldsymbol{J}$ is the approximate Hessian, which is a rough estimation towards matrix of second order derivatives. The LM method is capable of solving a slightly difference of equation 2, known by the name *augmented normal equations*. Which is shown in below equation 3.

$$N\delta_p = J^T \in$$

This is a variation where there are off-diagonal elements of N that is almost the same as the elements of J^TJ, while the diagonal element is induced as the equation $N_{ii} = \mu + [J^TJ]_{ii}$, where $\mu > 0$. Alteration of J^TJ is referred to as *damping* and μ is *damping term.* If the new parameter vector $\boldsymbol{p} + \delta_p$ with δ_p obtained from equation 3 computed lower error $\in$, the new parameter is then accepted, and the process will be repeated with a decreasing damping term. If its proven to be false, the augmented normal equation will be solved again, and the iteration continues until a lower error value of δ_p is found. By repetitively solving equation 3 using different damping term, this leads to an acceptable update to the parameter vector that corresponds to a single iteration of the LM algorithm. [16] The LM algorithm is the algorithm used in MATLAB to train the artificial neural network.

2.6.1.2 Long short-term memory (LSTM)

LSTM (Long Short-Term Memory) networks are specifically designed to handle temporal sequences and capture long-range dependencies. They possess the ability to retain information from previous states and utilize it in the current state. This is achieved through a set of gates, including input and output gates, which act as filters. The network assigns weights to relevant past data while disregarding less significant information. The same process is applied to any new incoming data. The resulting weighted data, combining both old and new information, is then passed through the output gate to generate an updated vector.

As a result, LSTM has proven to be highly effective in various tasks involving sequence prediction and labelling, such as speech recognition, handwriting generation, and machine translation. Additionally, LSTM has been successfully implemented in time series forecasting across different sectors, including economics, finance, e-commerce, petroleum production, hydrology, pandemic transmission, air pollution, weather prediction, solar radiation, and ionosphere electron density analysis. Numerous studies have explored the application of LSTM in predicting ionosphere parameters, indicating that the model performs relatively well for short-term predictions. By utilizing large and diverse datasets encompassing various solar and magnetic conditions, the accuracy of LSTM predictions can be further improved.[16-20]

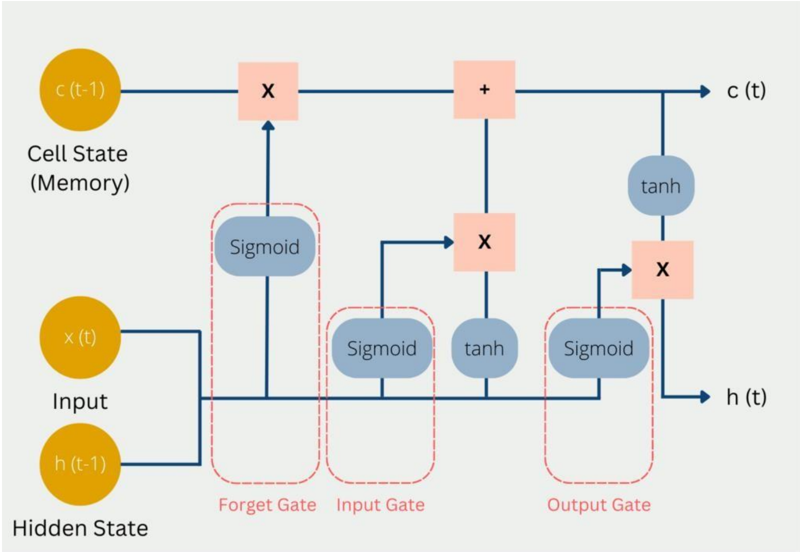

Figure 2.2: Long short-term memory

According to the paper by [19], the LSTM deep learning network model was trained and validated using observed TEC values spanning from 2009 to 2017. To achieve an accurate forecast, a suitable combination of solar and magnetic indices, along with hourly observed TEC variables, was carefully selected.

To assess the performance of the LSTM ionosphere forecast model, the results were compared with observed GPS-TEC data and estimations from the IRI-2016 model. The comparison analysis was conducted for the year 2018. The experimental performance of the proposed LSTM algorithm was evaluated by measuring error metrics (RMSE = 1.6149) and coefficients of correlation (CC = 0.992) during the test period. Findings indicated that the LSTM model exhibited a relatively improved performance compared to the IRI and GPS models, specifically at the low-latitude Indian location. The enhanced performance of LSTM was attributed to its high goodness of fit (r) value of 0.992. Thus, the deep learning model LSTM shows promise as a suitable forecasting model for estimating ionosphere conditions in the Indian low-latitude region. [19]

2.6.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is another highly popular and advanced machine learning technique. SVMs are supervised learning models equipped with learning algorithms that analyses the data for classification and regression analysis. Apart from linear classification, SVMs excel in non-linear classification by utilizing the kernel trick, which implicitly maps inputs into high-dimensional feature spaces. Essentially, SVMs draw margins between different classes. These margins are drawn in a manner that maximizes the distance between the margin and classes, effectively minimizing classification errors. [18]

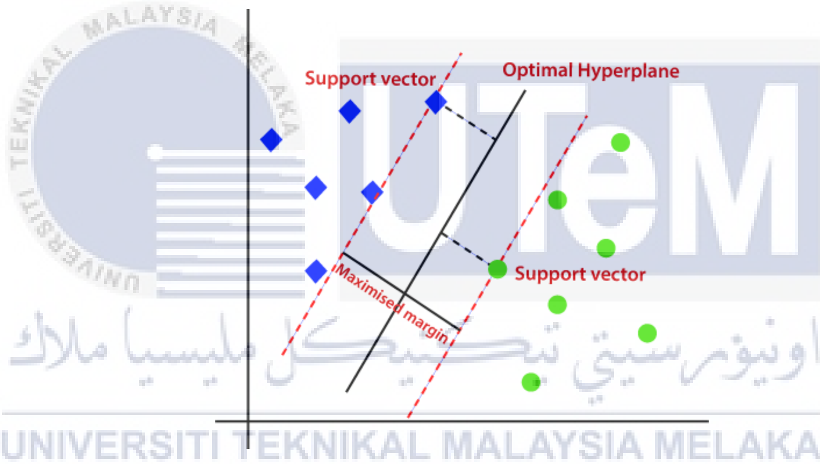

Figure 2.3: Support Vector machine

The three commonly used kernel functions are:

linear kernel function $(x_i, x_j)=\langle x_i, x_j\rangle$

polynomial kernel function $K((x_i, x_j) = \langle x_i, x_j\rangle + 1)^d$

Gaussian radial basis function kernel $K(x_i, x_j)= \exp(-g||x_i - x_j||^2)$.

The ionosphere undergoes complex variations influenced by factors like solar and magnetic activity. As a result, it is often characterized as a nonlinear system. This non-linearity makes the SVM method a suitable approach for predicting ionosphere parameters. [18]

In the paper by [18], it shows that SVM obtained excellent generalization performance and structural risk minimization. The input included contains the thermosphere wind, ionosphere diurnal variation, magnetic activity, solar activity, ionosphere seasonal variation, geographic location, and the recent values of TEC from 2016-2017. Prediction from the GPU accelerated SVM model on the 2018 datasets shows a short training time span at 7 minutes and it is able to provide good prediction values both in day and night.

2.6.3 Gradient Boosting Decision Tree (GBDT)

GBDT, which stands for Gradient Boosting Decision Tree, is a widely utilized machine learning algorithm renowned for its efficiency, accuracy, and interpretability. It excels in variety of machine learning tasks, such as multi-class classification, click prediction, and learning to rank, consistently achieving state-of-the-art performance. However, as big data has emerged with a significant increase in the number of features and instances, GBDT faces new challenges, particularly in striking the right balance between accuracy and efficiency. Traditional implementations of GBDT require scanning all data instances for each feature to estimate the information gain from all potential split points. As a result, the computational complexity of these implementations grows proportionally with both the number of features and the number of instances. [25]

The GBDT approach demonstrates high efficiency and remarkable capability in automatically detecting nonlinear interactions, making it a promising method for accurately forecasting TEC time series, especially in severe weather conditions. The algorithm operates as follows: Each sample, denoted as xi (where "I" = 1, 2, 3, ..., N), represents an individual data point, with N being the total number of samples. The training set is represented as T = {(x1, y1), (x2, y2), ..., (xN, yN)}, where y represents the predicted TEC value for each sample. GBDT aims to minimize a specific loss function to obtain the optimal model, with the loss function used to calculate residuals at each iteration. The most commonly employed loss functions include the square function and absolute error. [2]

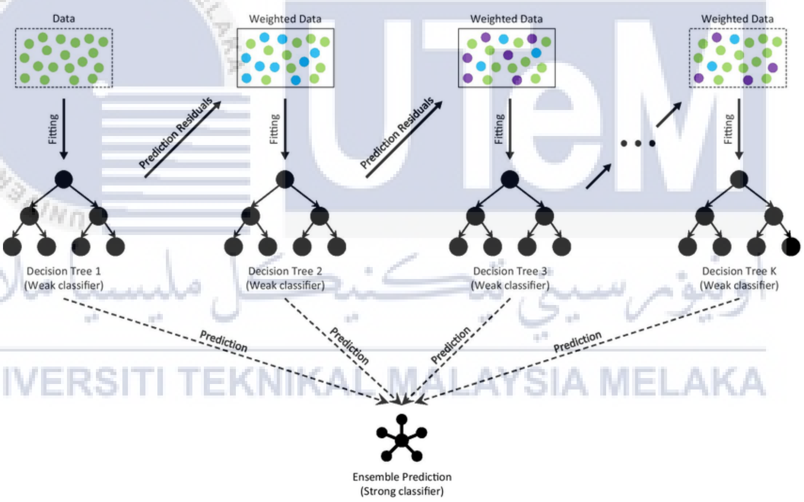

Figure 2.4: Gradient Boosting decision tree structure

In paper [2], three stations located at low magnetic latitudes were chosen, and data spanning a period of four years during the peak solar activity phase were utilized. This paper has utilized four different machine learning models namely NN, LSTM,ANFIS and GBDT. During the period of high solar activity, the machine learning-based approach demonstrates a prediction accuracy of approximately 2.9–4.7 TECU. Among the four models evaluated, the newly introduced GBDT approach exhibits the highest performance. In comparison to the other three models, GBDT shows an average improvement in prediction accuracy of around 5.6%. Additionally, GBDT performs exceptionally well during magnetic storms, with an average RMSE improvement of 12.7% when compared to the other models. This shows the GBDT has a high prediction accuracy and stability even when given severe space weather conditions. [2]

CHAPTER 3

METHODOLOGY

3.1 Introduction

The project is carried out for a span of one year time and is divided into multiple process. This chapter contains the methodology of the project. There are multiple stages for the purpose of training the machine learning model in order to forecast the future GPS TEC values. The very first stage includes data selection and preprocessing. This was followed by adjustments and fine tuning for the purpose of achieving the objective of the project. This project includes only software with no hardware involved and is under the category of emerging technology.

3.2 Project overview

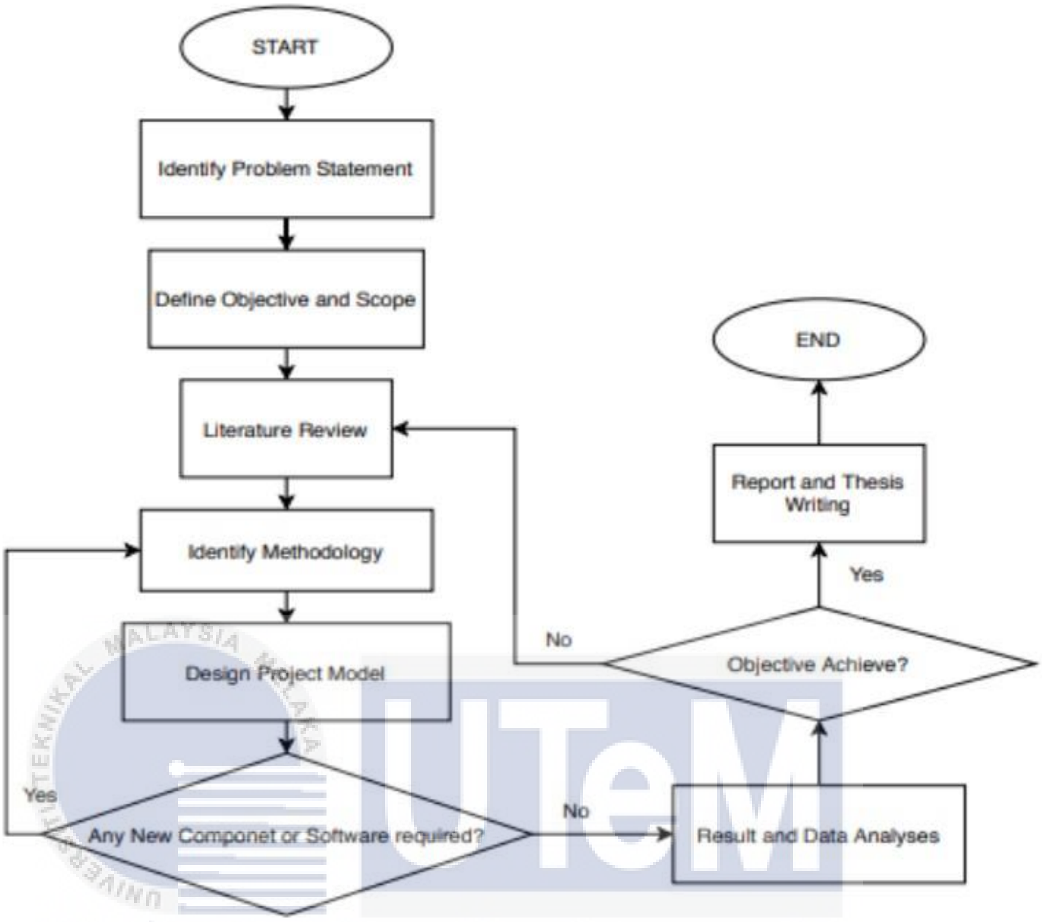

Figure 3.1: Project Flowchart

3.3 Neural Network in GPS TEC forecasting

In figure 3.2 will show the input used together with the aid of Neural Network machine learning language in order to forecast the TEC value.

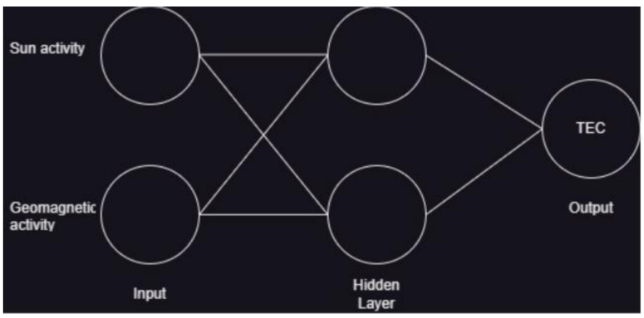

Figure 3.2: Neural Network for GPS TEC forecasting.

As shown in figure 3.2, it is seen that the input used for forecasting the TEC values are sun and magnetic activity. One hidden layer is used for forecasting the TEC values.

3.4 Project flowchart

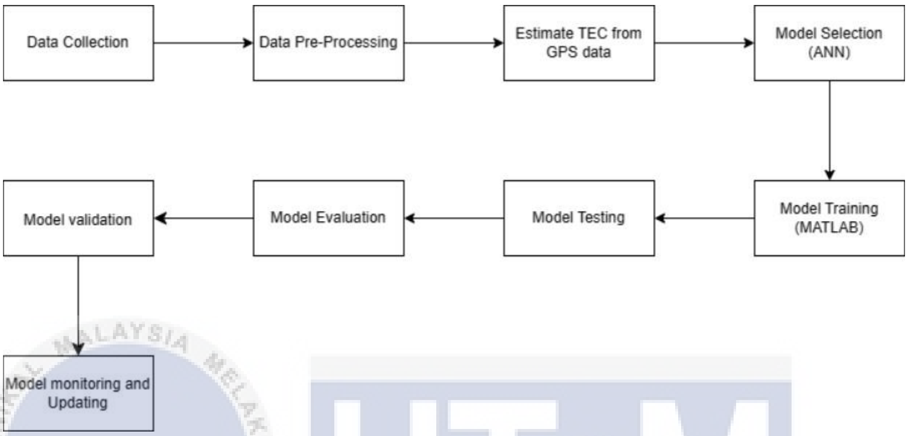

Figure 3.3: The flow chart of training the machine learning model.

The details and in-depth description of the process of training the machine learning model is shown below.

3.4.1 Data Collection

A total of 12 months' worth of GPS data collected from FKEKK station in the year 2022. This data is given through my final year project supervisor which is Dr. Ho YiH Hwa. On the other hand, the magnetic data and sun activity data is obtained from the NASA website which is through OMNIWeb Data Explorer (nasa.gov). The smallest data interval available on this website is hourly averaged shown in figure 3.4, hence this resolution is picked. In order to match the data from the NASA website, the TEC is also filtered to a resolution of hourly average.

Figure 3.4: Resolution selection in OMNI Web Nasa

The data obtained from the website is in “.txt” form, in order to better process and filter out the data it is converted into “.xlsx” form which is Microsoft excel.

For the GPS data it is processed using the GOPI-GPS-TEC software as shown in figure 3.5. There are a total of 365 compressed files initially each and within it consists of one “.o” file which is the file needed to extract the TEC values. In order to batch process, multiple programs are constructed using python or excel VBA to extract or modify the huge batch of data. In appendix A shows the code used to rename all the “.0” , “.N” , and “.G” file. Renaming is necessary due to the fact that the initial naming of each file does not match the rinex 3.0 file format hence cannot be processed by the GOPI-TEC software. After renaming the files comes the next step, which is to gather all the “.o” files into a single folder to allow the GOPI-TEC software to run everything in a batch instead of doing it manually one by one. In order to place all the files into a single folder the python code in appendix B is used. After grouping the data using the python code into a single folder, by clicking on “this year” in the GOPI-TEC user interface, all the “.std” and “.cmn” file will be produced into a single folder. The user interface is shown as figure 3.5.

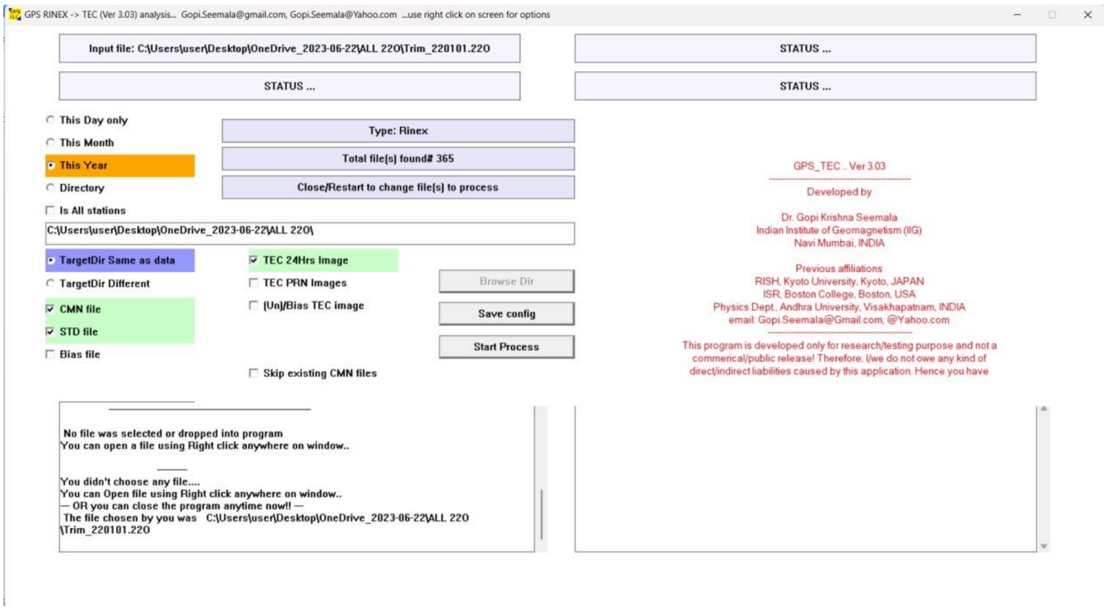

Figure 3.5: User interface of GOPI-TEC software

After extracting the “.std” files using the GOPI-TEC software, python code in appendix C is used to convert all the “.std” files into “.xlsx” file for further processing. This process eases the identification of columns of the needed data. By using the python code in appendix D, the hourly resolution of the mean TEC of each day is extracted into a new excel file. Since there is missing data in certain days, these missing data are automatically substituted with a “-”, which isn’t a numerical number hence this causes the MATLAB to be unable to obtain the data that is copied into the workspace. Hence the python code in appendix E is used to filter out days without a complete 24-hour data.

The last set of input data is the TEC seasonal and diurnal differences which consists of two parts naming the day number and hour number. These two criteria are elaborated in the equations below.

$$DNS = SIN\frac{2\pi \times DN}{365.25} \tag{1}$$

$$DNC = COS\frac{2\pi \times DN}{365.25} \tag{2}$$

$$HRS = SIN\frac{2\pi \times HR}{24} \tag{3}$$

$$HRC = COS\frac{2\pi \times HR}{24} \tag{4}$$

Where, DNS denotes the sine component corresponding to the day number of the year, while DNC signifies the cosine component associated with the day number of the year. HRS represents the sine component corresponding to the hour number of the day, and HRC represents the cosine component of the hour number of the day.

The above equation of (1)-(4) shows the cyclic components (sine and cosine fragments) of day number and hour number. In order to acquire these data, the equation is placed in Microsoft excel and the corresponding required input is entered.

3.4.2 Data Pre-Processing

The data is pre-processed and is analyzed in order to extract relevant features to forecast TEC values. Methods commonly used are filtering, smoothing, feature extraction and so on. But the main method used in this neural network model training is through data correlations, data normalization and filtering.

3.4.2.1 Data correlation

Data correlations refers to the statistical measure of the extent to which two or more variables vary in relation to one or more variables. This pre-processing method determines the robustness and direction of linear relationship between the two chosen variables. rrelation coefficients are usually used in expressing the degree of correlation. The most common correlation coefficients are known as the Pearson correlation coefficients which has a range from -1 to 1, where:

- 1 resembles a perfect positive linear correlation.
- -1 resembles a perfect negative linear correlation.
- 0 resembles no linear correlation.

After placing in all the input data into the workspace in MATLAB, a specific code line is needed to obtain the plot which is the MATLAB code “corrplot()”. By using this line of code, a correlation between TEC and input variables, (sun activity and geomagnetic activity) is obtained as shown in figure 3.6.

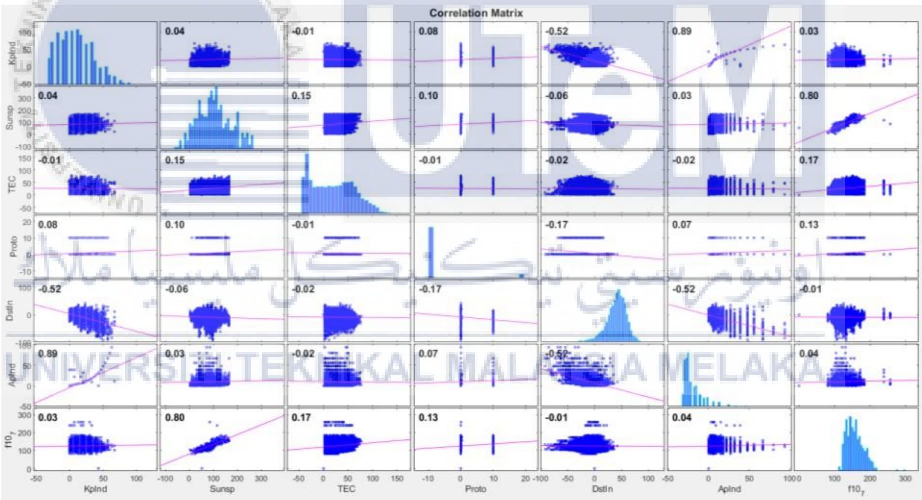

Figure 3.6: Data correlation between TEC and input

3.4.2.2 Data normalization

Data normalization is another preprocessing method that is commonly used in both statistic and machine learning to scale and make the input data features consistent. The ultimate goal of performing a data normalization is to transfigure the data under the same scale and without interrupting the difference in the range of the values in the dataset. Normalization is best when dealing with outlier data and as well as dealing with algorithms that are sensitive to the scale of input features. The method used for data normalization is through the minimum-maximum normalization method. This method uses the formula below:

$$X_{normalize} = \frac{X - X_{minimum}}{X_{max} - X_{minimum}}$$

Normalization is an important step in preprocessing pipeline, and to make sure that the input features contribute equally to the learning process of the neural network and also improves the output which is the performance and regression of the network.

3.4.2.3 Filtering

After processing the raw GPS data using the GOPI-TEC software, two different files will be prompted which are the ".cmn" and ".std" file. Among these two files the ".std" file is used and the data in column 3 which is the standard deviation of TEC is used as the target output of the TEC, which is also the output that is needed to be predicted. Going through the data, certain dates shows that there are missing TEC values at certain hours which do not fulfill the 24 hours requirement mark for each day. This process is done by using python code to run through every single ".std" file to determine if each file has a total of 1440 occupied data. The python code is attached in appendix E.

3.4.3 Estimate TEC from GPS data.

To estimate the TEC, Computing the ionosphere delay by measuring the difference between the signal's arrival time of the the receiver and the expected arrival time based on the GPS satellite's location is needed. Next, which is the extraction of the carrier phase, where it is used to eliminate errors caused by the GPS receiver's clock. Furthermore, calculation on the ionosphere is delayed by dividing it by the carrier frequency of the GPS signal. Lastly it is important to map the TEC value over a geographic area to analyze the ionosphere conditions and predict the impact of ionosphere disturbances on GPS signals.

3.4.4 Machine learning network selection

Artificial Neural Network (ANN) is used, ANN are biologically inspired computational networks, it is a computational model inspired by the human brain that can learn patterns and make predictions. It consists of interconnected layers of artificial neurons that process and transmit information, enabling tasks such as classification, regression, and pattern recognition. ANN comprises of three layers: input, hidden, and output. In this project, the feed-forward backprop neural network algorithm is used and Levenberg-Marquardt algorithm as training technique.

3.4.4.1 Neural network specifications

Neural networks are recognized as artificial intelligence mechanisms capable of learning non-linear input/output relationships in complex processes. These mechanisms comprise basic processing elements known as artificial neurons, storing the summation obtained by manipulating input signals using weights. The determination of these weights in an artificial neural network involves an iterative adjustment procedure during training until optimal weights are achieved. Once trained, the input signal passes through an activation function (transfer function) to produce neuron outputs. The sigmoid activation function, typically denoted as Equation 1, is commonly preferred in multi-layer perceptron models as a non-linear filter to generate output signals.

During training, a back-propagation algorithm is employed in both feed-forward and feed-backward processes. The biases of the neural network are iteratively adjusted until the Root Mean Squared Error (RMSE) reaches a specified threshold for the output signal. In this project, the activation function for all layers is the tangent sigmoid function, and the Levenberg-Marquardt backpropagation algorithm is used for network training. Due to its quick response for predictions and effectiveness in the training process, the final multi-layer perceptron neural network chosen for this project consists of one input layer, three hidden layers with numerous neurons, and one output layer.

The optimal number of neurons and layers was determined through trial and error. Various neural network designs with the same input neurons but different numbers of neurons and layers were considered to find the optimal architecture. TEC, associated with solar cycle variation, seasonal variation, diurnal variation, spatial variation, geomagnetic activity, and solar activity variation, was expected to be learned by the proposed neural network. Data on solar activity and geomagnetic criteria were obtained from the OMNIWeb Data Explorer website (nasa.gov). The input layer of this neural network comprises ten neurons representing different parameters. Several network designs, labeled as Net1, Net2, and Net3 are displayed in figure 3.7. Meanwhile a neural network from past research paper [27] is also included as shown in figure 3.8. The specifications of the networks are also displayed in table 3.1 below.

Table3. 1: Specifications of Neural Network

Neural Network Model	Input nodes	Cyclic component or day number and hour number	Solar Activity Parameters	Geomagnetic activity Parameters
Net1	10	DNS, DNC, HRS, HRC	Sunspot No, Proton Ratio, F10.7 index	Kp index, Dst Index, Ap index
Net2	6	-	Sunspot No, Proton Ratio, F10.7 index	Kp index, Dst Index, Ap index
Net3	6	DNS, DNC, HRS, HRC	Sunspot No	Dst index
Net2016	8	DNS,DNC,HRS,HRC	Sunspot No, F10.7 index	Ap index, Dst index

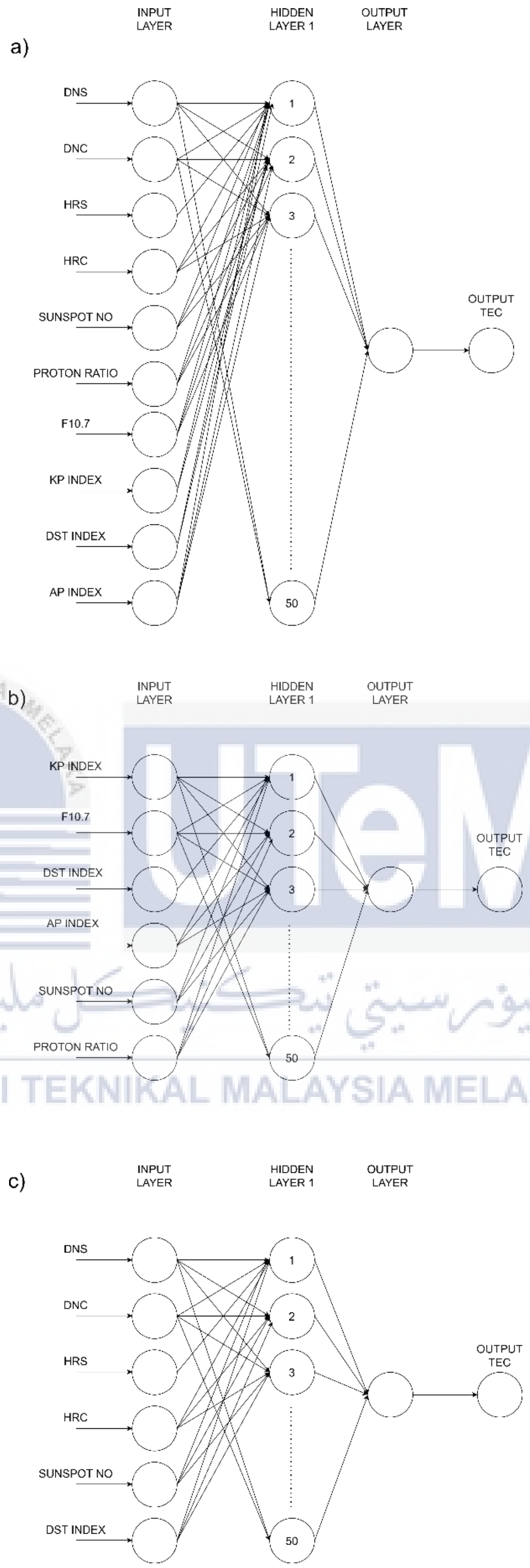

Figure 3.7: Schematic diagram of proposed neural network models, a) Net1, b) Net 2, and c) Net 3

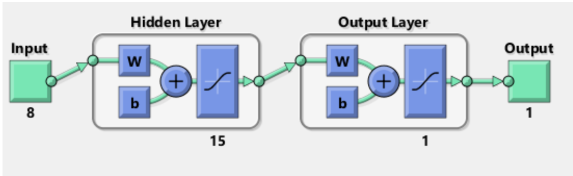

Figure 3.8: Network specification of past research

These neural networks were statistically tested using different combinations of input neurons to determine the optimal architecture. The decision criterion for the testing procedure was statistically the lowest expected RMSE parameter. The analysis in this research covers the year 2022.

It is necessary to implement neural network systems with different input layer neurons in order to evaluate the suitability of a particular input layer to the performance of the neural network. Satellite communication systems using radio frequencies are impacted by the ionosphere's dispersive characteristics, a phenomenon driven by the sun's ultraviolet radiation and geomagnetic activity. Disturbances in the ionosphere caused by space weather events, such as solar flares (specifically X-ray flares) from the Sun and geomagnetic storms, have the potential to disrupt GNSS (Global Navigation Satellite System) communications. The neural network model learns from the Sunspot Number (SSN) as an indicator of solar activity. An increase in the Sun's radiation levels corresponds to a higher number of sunspots. Solar flares significantly elevate these emissions, and upon interacting with Earth's magnetic field, they cause disturbances, leading to ionospheric storms that can degrade satellite-based radio communications. The F10.7 proxy, measuring solar flux, serves as a fundamental indicator of solar activity and radiation received from the Sun. Measured in solar flux units (sfu) at a frequency of 2800 MHz (10.7 cm), solar flux is closely tied to ionization levels and electron concentration in the F2 region. The Sun emits electromagnetic radiation, including X-rays, ultraviolet (UV) rays, protons, and electrons collectively known as solar wind. Changes in solar wind speed and proton density impact the space environment, generating substantial electric currents that disrupt satellite signals and communications. The Ap index serves as a global geomagnetic activity indicator, while the Dst index is widely used as a proxy for ring current strength, reflecting the intensity of geospace magnetic storms, a complex aspect of magnetospheric dynamics. The Kp index serves as a gauge for geomagnetic activity, ranging from 0 to 9, where elevated values signify increased levels of geomagnetic disturbance. It is derived from the observed magnetic field variations recorded at specific magnetic observatories worldwide. The average Kp index is influenced by multiple factors, such as the solar cycle. During phases of reduced solar activity, the typical Kp index tends to be lower, whereas in periods of heightened solar activity, it may register higher values. Generally, a Kp index of 2 or lower indicates calm geomagnetic conditions, while readings of 5 or higher suggest turbulent conditions.

3.4.5 Model training

MATLAB is used to train the model with the aid of the extension download which is “deep learning toolbox”. By writing the MATLAB code “nntool” in the workspace, the following user interface (ui) will appear as figure 3.9.

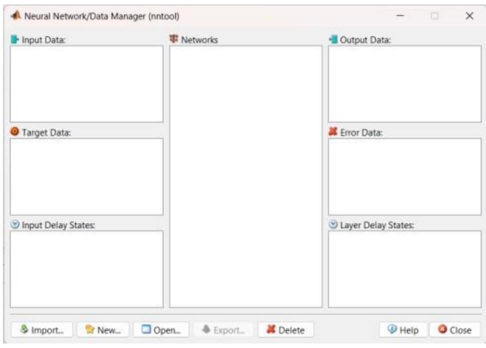

Figure 3.9: User interface of nntool.

The data is split into training and validation sets. The training set is used to train the model, and the validation set is used to evaluate the model's performance. During the training of the data, a total of 8688 data set was used. Among all the given data, only 70% of the input data was taken for training where 6082 input was taken randomly into account as the training data. The remaining 30% of the input data is split equally which is 15% each for the validation and testing data. The model will be trained with the processed data, this will aid the learning process of mapping between the input features and TEC values. In this project one year span worth of data in 2022 is used for training. Based on previous studies, it is shown that the TEC values vary according to the diurnal variation, seasonal variation, solar activity, and geomagnetic activity. Based on the previous research, these parameters will be used as input in order to predict the output which is the TEC.

3.4.6 Model Testing

Model testing is the process of evaluating the performance of a trained machine learning model using the same set of input data. This is done to estimate how accurate the model is likely to perform in the real-world, outside of the training environment. The same input data is fed back into the trained module to generate the predicted output TEC to show the comparison of the actual versus the predicted value. This process is done in the UI of nntool shown in figure 3.10.

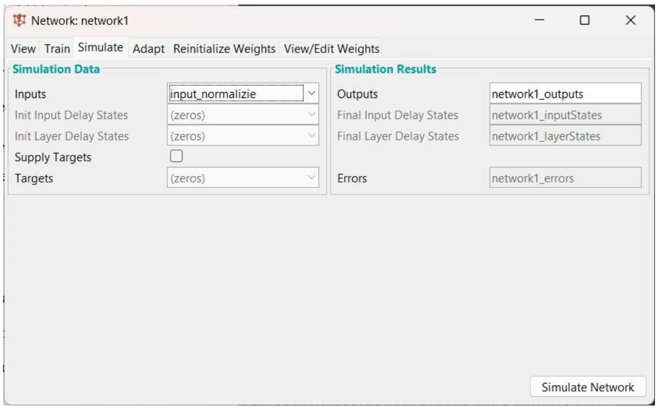

Figure 3.10: User Interface for network simulation in MATLAB

Using the simulate network button, MATLAB will be able to produce the output data from the prediction of the neural network model. This can aid the process of comparing resulting outputs in order for a better visual representation using graphs.

3.4.7 Model Evaluation

The performance of the model is evaluated using metrics. “Metrics” are standards used for analyzing and evaluating the aspects of a certain system, performance, or process. By using metrics, it can clearly identify the efficiency, quality, or effectiveness of something. In order to produce an optimum Neural Network, a comparison of the Root-Mean-Square Error (RMSE) was used. RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(TEC_{pred} - TEC_{meas})^2}$$

In this equation, N represents the number of data points, TEC_{pred} is the predicted total electron content by the trained model and TEC_{meas} is the mean total electron content estimated by the UTeM GPS station in Malacca.

Next is the coefficient of determination, R^2. This metric measures the proportion of the dependent variable that is forecasted from the input variables, which in all shows how does the predicted values match the actual values. R^2 has a range from 0 to 1, where the closer the value it is to 1 the more perfect it predicts the dependent variable. If the value shows closer to 0, the model shows no variability and relation with the output variable. The formula of R^2 is denoted as below.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(TEC_{meas} - TEC_{pred})^2}{\sum_{i=1}^{n}(TEC_{meas} - TEC_{mean})^2}$$

Where in this equation, n represents the number of data points, $TEC_{meas}$is the actual measurement of the total electron content value, TEC_{pred} is the predicted total electron content value from the trained neural network module, TEC_{mean} is the mean of the actual total electron content value.

3.4.8 Model Validation

Evaluation on how well a statistical model fits the data it was designed to explain or predict. This is important to ensure that the model is not overfitting or underfitting the data and that it can be used to make accurate predictions on new data. Based on the evaluation metrics, refine the model. This can involve changing the features, adjusting hyperparameters, or selecting another alternate models altogether. After the model is refined, test it on a separate test set to evaluate its performance on new data.

3.4.9 Model monitoring and updating

Regular monitoring and updating is done to ensure that the model continues to perform well.

3.5 Software selection

3.5.1 MATLAB

Figure 3.11: MATLAB logo

MATLAB is a powerful language designed for technical computing, offering a user-friendly environment that combines computation, visualization, and programming. It allows users to express problems and solutions using familiar mathematical notation. Common applications of MATLAB include math and computation, algorithm development, modelling, simulation, prototyping, data analysis, visualization, scientific and engineering graphics, and application development with graphical user interfaces.

The application name MATLAB originated from "matrix laboratory" and was initially created to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which were considered cutting-edge in matrix computation.

Over the years, MATLAB has evolved through user feedback. It has become the standard instructional tool for mathematics, engineering, and science courses in universities, while also being widely adopted in industry for high-productivity research, development, and analysis.

MATLAB offers a range of specialized solutions called toolboxes, which are essential for many users. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the capabilities of MATLAB for solving specific classes of problems. Various areas, including signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and more, have dedicated toolboxes available. [21]

3.5.2 GPS_TEC

Figure 3.12: GPS_TEC logo

The GPS TEC software developed by Gopi seemala is a useful application in deriving the GPS-TEC data from the RINEX 2.1 & RINEX 3.02 observation files. But it has a certain few limitation which supports only TEC derivations from GPS, while TEC from other sources such as GNSS is still currently unavailable.

This application is capable of batch processing all input files of the month and year, as well as every stations and files in the directory. It is also capable of getting ephemeris from the IGS navigation file once connected to the internet. Offline can be done as well but only if the same file is available in the same directory as the data. The software is capable of calculating the TEC from the observation data (from the input file) of GPS Rinex, Novatel (only capable for ID43 records), SCINDA (.scn files) and Leica. The output of the data is written in ascii output files which are the (*.CMN and *.STD) files. The vertical TEC values will be plotted on the software UI for every processed input data.

3.5.3 Python

Figure 3.13: Python logo

Python stands as a high-level, versatile programming language renowned for its emphasis on code readability, achieved through the use of significant indentation. It operates as a dynamically typed, garbage-collected language that accommodates various programming paradigms, including structured, object-oriented, and functional programming. Often referred to as a "batteries included" language, Python boasts a comprehensive standard library.

As a multi-paradigm language, Python fully supports object-oriented and structured programming, with additional backing for functional programming and aspect-oriented programming, including metaprogramming and metaobjects. It extends support for various paradigms through extensions, encompassing design by contract and logic programming.

Python employs dynamic typing, utilizing reference counting and a cycle-detecting garbage collector for memory management. The language employs dynamic name resolution (late binding), binding method and variable names during program execution.

Python's design incorporates aspects of functional programming in the Lisp tradition, featuring functions such as filter, map, and reduce, alongside list comprehensions, dictionaries, sets, and generator expressions. The standard library includes modules like itertools and functools, implementing functional tools borrowed from Haskell and Standard ML.

Designed for readability, Python's formatting is visually clean and often utilizes English keywords where other languages rely on punctuation. In contrast to some languages, it avoids using curly brackets to denote blocks, and while semicolons after statements are permitted, they are rarely used. Python exhibits fewer syntactic exceptions and special cases compared to languages like C or Pascal.

3.5.4 Excel Visual Basic for Application (VBA)

Figure 3.14: Microsoft excel logo.

Microsoft Excel is a spreadsheet editor created by Microsoft for use on Windows, macOS, Android, iOS, and iPad Operating system. This software encompasses powerful calculation and computation capabilities, graphing tools, pivot tables, and includes a macro programming language known as Visual Basic for Applications (VBA).

Microsoft Excel serves as a comprehensive spreadsheet tool with fundamental features common to all spreadsheets. It utilizes a grid layout of cells, organized in rows and columns, identified by numbers and letters, respectively, facilitating data manipulations like arithmetic operations. The software is equipped with a diverse range of built-in functions designed to address statistical, engineering, and financial requirements. Users can visualize data through line graphs, histograms, charts, and a limited three-dimensional graphical display.

One distinctive aspect is its programming functionality called Visual Basic for Applications (VBA), enabling users to employ various numerical methods. For instance, users can solve differential equations of mathematical physics and report the results back to the spreadsheet or external applications. Programming with VBA allows for spreadsheet manipulations that may be cumbersome or unfeasible using standard techniques. The Visual Basic Editor (VBE) provides a dedicated environment for code writing, debugging, and code module organization.

VBA facilitates the implementation of numerical methods and the automation of tasks such as formatting or data organization. Users can guide calculations using intermediate results reported back to the spreadsheet. The Macro Recorder is a user-friendly method for generating VBA code. It records user actions and translates them into VBA code, forming a macro that can be automatically repeated. Macros can be linked to triggers like keyboard shortcuts, command buttons, or graphics. The function of VBA and excel spreadsheet has greatly aid in batch processing of data.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

This chapter will be discussing on the result obtained which are the predicted TEC values from the trained neural network, optimum settings in MATLAB for training the neural network, Types of input and input nodes in the neural network, for the best regression and mean square error. The performance of the neural network and the result will be evaluated throughout this chapter.

4.2 Transfer function of neural network

The total electron content (TEC) data is obtained from UTEM (2.3122° N, 102.3195° E), Melaka for the year 2022 for training and experimental purposes. The standard deviation TEC dataset for the year 2022 is obtained using the Gopi-TEC software with a time resolution of 1 hour interval. The solar and geomagnetic activity data also has a 1-hour interval time resolution in order to match the TEC time span. All input data are paced at an interval of 1 hour. Initially three different transfer functions are tested, which are the functions available in MATLAB which are PURELIN, TANSIG, and LOGSIG. These 3 functions are tested with the same input data and figure 4.1 below shows the prediction vs actual TEC graph.

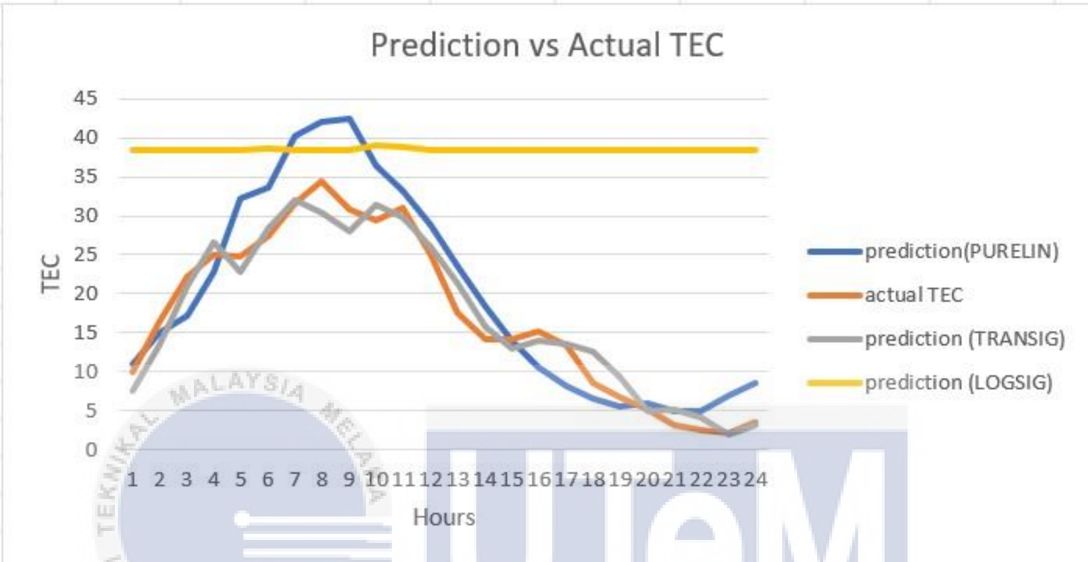

Figure 4.1: Prediction VS Actual TEC graph in 24 hours

As observed from the figure, we can conclude that the TANSIG function has the highest accuracy in comparison to all 3 of the different functions.

4.3 Hidden layer specification

Determining the number of neurons in the hidden layers plays a crucial role in defining the overall architecture of your neural network. Despite their lack of direct interaction with the external environment, these layers wield substantial influence over the ultimate output. It is imperative to carefully consider both the quantity of hidden layers and the number of neurons within each layer. Insufficient neurons in the hidden layers can lead to underfitting, a situation where the network struggles to discern signals in intricate datasets.

Conversely, an excessive number of neurons in the hidden layers may give rise to various issues. Firstly, it can result in overfitting, characterized by the network having surplus information processing capacity, making it challenging to train all neurons adequately with the limited data from the training set. Even with sufficient training data, an excessively large number of neurons in the hidden layers can extend the training time, potentially reaching a point where effective training becomes impractical for the neural network.

From the trial and error through placing the same input but varying the number of hidden layer and number of neurons, the following table 4.1 is acquired. This table show cases the result of metrics R^2 and RMSE. The input is fixed with 10 neurons namely HRS, HRC, DNS, DNC, AP index, KP index, DST index, proton ratio, sunspot number, and F10.7 index.

Table 4. 1: hidden layer and number of neurons in relation to R^2 and RMSE

No of neurons (layer 1)	No of neurons (layer 2)	No of neurons (layer 3)	No of layers	R	MSE	R^2	RMSE
-	-	-	0	0.906	49.006	0.8208	7.0004
10	-	-	1	0.97116	15.9704	0.9432	3.996
20	-	-	1	0.9791	12.3373	0.9586	3.524
30	-	-	1	0.9808	11.8779	0.9620	3.4464
40	-	-	1	0.97916	13.8061	0.95887	3.7156
50	-	-	1	0.98571	8.1353	0.97162	2.8522
60	-	-	-	0.98244	9.5488	0.96518	3.09
10	10	-	2	0.97263	13.943	0.9460	3.734
50	50	50	3	-0.02	966.5187	-	31.088

From the table, it is shown that without a single layer or neurons, the R^2 and RMSE criteria proves to be at its worst. In comparison, when a single layer of neuron is applied, the performance metrics R^2 and RMSE shows improvement, especially the RMSE which is one of the key in acquiring an accurate output. With an increment of 10 neurons for every trial and error to determine the best number of neurons for the neural network it is seen that the value of R^2 steadily increases while the RMSE decreases. The performance metric of the neural network peaked at 50 neurons with a single hidden layer. Up until the number of hidden layer hits 60, the performance starts to show a decrement, hence the trial and error is stopped and moved on to increasing the number of hidden layer. It is shown that increasing the number of hidden layers does not actually benefit the performance metrics and this is especially true when the hidden layer is increased up to 3 layers with high number of neurons. A negative R^2 is produced and an MSE of close to 1000 is acquired. The following trial and error experiment proves the fact that excessive neurons and layers will cause over fitting and a poor result in performance metrics which in all produce a bad neural network.

4.4 Input neurons

After choosing the best number of neurons in a hidden layer, three different models with different numbers and type of input neuron are placed in comparison to one previous neural network model from the research paper [27]. This model also holds a different number of hidden neurons which is 15 in comparison to the number of neurons in this experiment which is 50. The neural network models are trained with different input parameters that affect and correlate to the ionospheric TEC variability. Figure 4.2 shows the input plots against the number of days in a year.

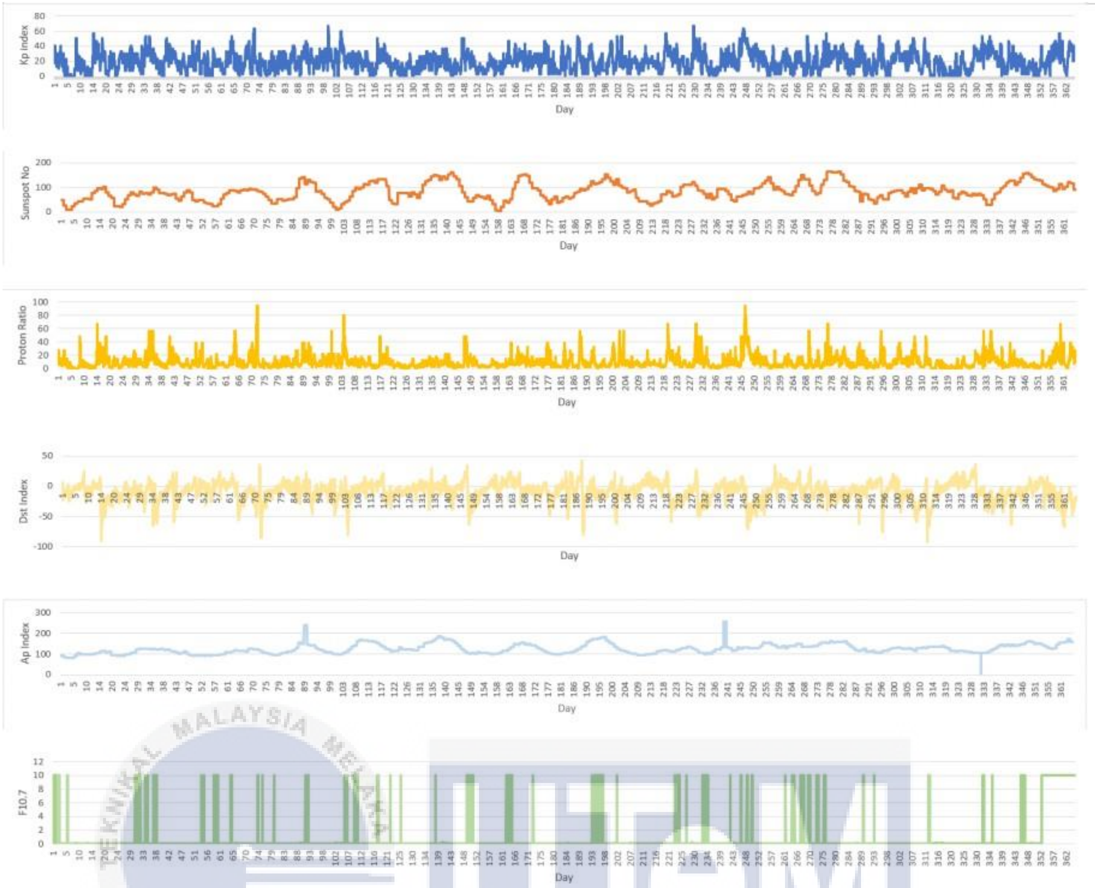

Figure 4.2: Input plots of solar and geomagnetic activity in year 2022

The input data as shown in figure 4.2 shows the time span that is included in this machine learning training model which is a total of 365 days in 2022. The daily data has an interval of 1 hour, which allows this machine learning model to predict TEC that is one hour ahead of time. These input sets aid the neural network model in acquiring an accurate ionospheric TEC result when influenced by the solar and geomagnetic activity.

The input as shown in figure 4.2 is used for training the neural network from past research paper in 2016 [27] where the researched neural network model has a different amount of hidden layer and hidden neurons, as well as training algorithm. The key performance metrics that is evaluated in this project are the RMSE and R^2. Both these criteria for all four of the neural network model are shown in figure 4.3 and figure 4.4. The input specifications of the models are shown in table 3.1.

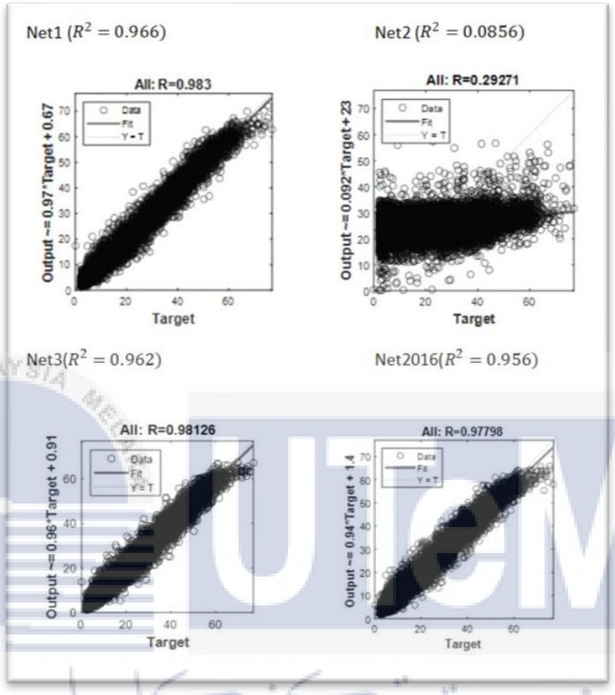

Figure 4.3: Scatter plot of all neural networks

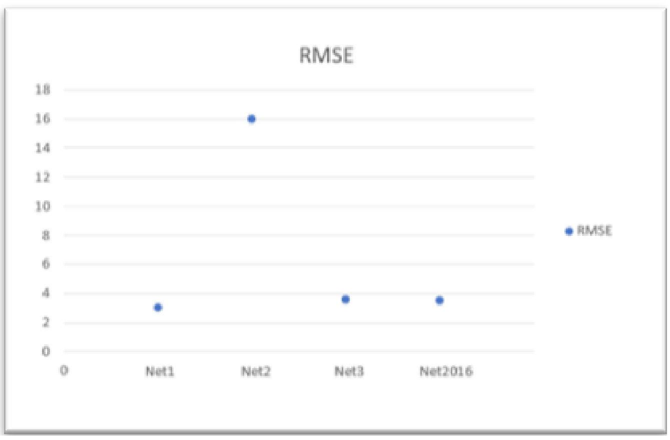

Figure 4.4: RMSE of all neural networks

Table 4. 2: Mean Absolute Error for each individual neural network.

	Model	MAE (TECU)
Highest error	Net2	15.9905
	Net3	3.0031
	Net2016	2.6574
Lowest Error	Net1	2.3459

In order to evaluate the performance of each individual neural network models, performance metrics such as R^2 and RMSE and error analysis parameters such as Mean Average Error (MAE), denoted by the formula:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |TEC_{MEAS} - TEC_{PRED}|$$

Where n is the number of data included, TEC_{MEAS} is the actual TEC provided by UTeM GPS station, TEC_{PRED} is the predicted value of TEC by the neural network. The MAE values are tabulated in table 4.2. First of all, according to the R^2 values shown in figure 4.3, all the R^2 values are close related to one another at a range of 0.9 which yields a good coefficient of determination, except for Net2, which yields a R^2 value of less than 0.1. Net2 consists of only the solar and geomagnetic activity without the cyclic component (sine and cosine components) of day number and hour number. This shows that the VTEC diurnal and seasonal differences make the most impact out of all the different solar and geomagnetic activity. Hence in order to properly train a functional neural network to predict the TEC output, all four of these inputs which are

DNS, DNC, HRS, and HRC are crucial. This statement held true even for the following criteria which are root mean square error (RMSE) and mean absolute error (MAE). The RMSE of net2 proves an outlier as it does not fall in the same range as the other 3 neural networks. Net2 has a huge RMSE which is around 16 TECU while others have a range of TECU that is less than 4. Same goes to the error analysis parameter, which is MAE, it also shows that Net2 has the highest error among all the neural networks which is nearly 5 times more in comparison to the lowest error neural network model. Net 3 shows to be the third lowest error value as shown in table 4.2. Net3 is driven by sunspot numbers and Dst Index which are significant features that are actively influencing the ionospheric TEC variations.

According to the MAE data in table 4.2 the next lowest error neural network model shows to be the network model proposed in the year 2016 by the research paper [27] known as Net2016. In comparison to all the proposed network models in this project the network model proposed previously does not fall short. The biggest difference maker among the best performing neural networks in this project lies in the number of hidden neurons and the amount of input neurons. Where there's only 15 hidden neurons and 8 input neurons for the past research paper [27], and the current neural network (Net1) consists of 50 hidden neurons and 10 input neurons. A total of 0.4% of increment was achieved based on the correlation of determination R^2. Net1 is referred to as the unified model of all the input data related solar and geomagnetic activity parameters. The R^2, RMSE, and MAE values for Net1 are 0.966, 3, and 2.3459 TECU. Thus, this shows that Net model is performing well forecasting with a high accuracy in comparison to all the remaining models and the past 2016 research model. This proves that additional correlated input neurons aid in having a higher precision reading.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In this paper, 1-h ahead prediction of ionospheric TEC using various neural network model is evaluated over the UTeM , Global Positioning System, Malaysia, Melaka. This paper focuses on varying the input neurons, hidden layers and hidden neurons and is trained using a 1-year span of TEC values in the year 2022.

The Neural network models utilize input parameters derived from factors known to impact ionospheric Total Electron Content (TEC), including solar activity, magnetic activity, seasonal variation (day number, DN), and diurnal variation (hour number, HR). The DN and HR represent seasonal and diurnal variations, respectively, expressed through cyclic components (sine and cosine) to capture the continuous trend of GPS Vertical Total Electron Content (VTEC) data. Input layers for various Neural Network models, such as Net1, Net3, and Ner2016 models, incorporate sets of solar and geomagnetic indices, along with DN and HR components. Notably, Net2 deviates from the others by excluding seasonal and diurnal variations, allowing observation of the significance of this specific input parameter.

The selection of input data for Neural Network (NN) models, particularly solar and geomagnetic indices, is noted to have a substantial impact on the predictions generated by the NN models. An analysis of errors, based on parameters such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2, is conducted to identify the most suitable solar and geomagnetic indices. This process aims to determine the indices that result in minimal prediction errors when comparing the predicted Vertical Total Electron Content (VTEC) values with those measured from GPS observations.

In conclusion, all the objectives of this paper were achieved. The chosen neural network, which is Net1, has a feed forward back propagation network using the tangent sigmoid transfer function. This neural network net1 consists of 1 hidden layer of 50 hidden neurons and 10 input neurons. Net1 was capable of predicting the TEC with high accuracy and low analysis errors. In the future, larger data size could be implemented for a higher accuracy in prediction for the reliability of the TEC forecasting.

REFERENCES

[1] Dana, P.H. Global Positioning System (GPS) Time Dissemination for Real-Time Applications. Real-Time Systems 12, 9–40 (1997). https://doi.org/10.1023/A:1007906014916

[2] Y. Han, L. Wang, W. Fu, H. Zhou, T. Li and R. Chen, "Machine Learning-Based Short-Term GPS TEC Forecasting During High Solar Activity and Magnetic Storm Periods," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 115-126, 2022, doi: 10.1109/JSTARS.2021.3132049.

[3]] X. Zhang, X. Ren, F. Wu, and Q. LU, “Short-term TEC prediction of ionosphere based on ARIMA model,” Acta Geod. Cart. Sin., vol. 2, pp. 118–124, 2014

[4] D. V. Ratnam, Y. Otsuka, G. Sivavaraprasad, and J. R. K. K. Dabbakuti, “Development of multivariate ionosphere TEC forecasting algorithm using linear time series model and ARMA over low-latitude GNSS station,” Adv. Sp. Res., vol. 63, no. 9, pp. 2848–2856, 2019.

[5] R. Tang, F. Zeng, Z. Chen, J.-S. Wang, C.-M. Huang, and Z. Wu, “The comparison of predicting storm-time ionosphere TEC by three methods: ARIMA, LSTM, and Seq2Seq,” Atmos. (Basel), vol. 11, no. 4, 2020, Art. no. 316.

[6] Mendillo, M. (2006), Storms in the ionosphere: Patterns and processes for total electron content, *Rev. Geophys.*, 44, RG4001, doi:10.1029/2005RG000193.

[7] Anderson, D. N., and J. A. Klobuchar (1983), Modeling the totalelectron content observations above Ascension Island,J. Geo-phys. Res.,88, 8020–8024.

[8] Anderson, D. N., et al. (1998), Intercomparison of physical modelsand observations of the

ionosphere,J. Geophys. Res.,103,2179–2192.

[9] Araujo-Pradere, E. A., T. J. Fuller-Rowell, and D. Bititza (2004),Time empirical ionosphere correction model (STORM) responsein IRI2000 and challenges for empirical modeling in the future,Radio Sci.,39, RS1S24, doi:10.1029/2002RS002805.

[10] Sahu, S., Trivedi, R., Choudhary, R. K., Jain, A., & Jain, S. (2021). Prediction of total electron content (TEC) using neural network over anomaly crest region Bhopal. Advances in Space Research, 68(7), 2919-2929.

[11] Liu, LiBo, et al. "Solar activity effects of the ionosphere: A brief review." Chinese Science Bulletin 56 (2011): 1202-1211.

[12] Vaquero, J. M. (2007). Historical sunspot observations: a review. Advances in Space Research, 40(7), 929-941.

[13] Matzka, J., Stolle, C., Yamazaki, Y., Bronkalla, O., & Morschhauser, A. (2021). The geomagnetic Kp index and derived indices of geomagnetic activity. Space weather, 19(5), e2020SW002641.

[14] Mahesh, Batta. "Machine learning algorithms-a review." *International Journal of Science and Research (IJSR).[Internet]* 9 (2020): 381-386.

[15] G. E. Hinton, “How neural networks learn from experience,” Sci. Amer., vol. 267, no. 3, pp. 144–151, 1992

[16] Lourakis, M. I. (2005). A brief description of the Levenberg-Marquardt algorithm implemented by levmar. Foundation of Research and Technology, 4(1), 1-6.

[17] Homam, Mariyam Jamilah. "Prediction of total electron content of the ionosphere using neural network." *Jurnal Teknologi* 78.5-8 (2016).

[18] Xia, Guozhen, et al. "Ionosphere TEC forecast model based on support vector machine with GPU acceleration in the China region." *Advances in Space Research* 68.3 (2021): 1377-1389.

[19] Hu, A.; Zhang, K. Using Bidirectional Long Short-Term Memory Method for the Height of F2 Peak Forecasting from Ionosonde Measurements in the Australian Region. Remote Sens. 2018, 10, 1658. [CrossRef]

[20] Moon, S.; Kim, Y.H.; Kim, J.-H.; Kwak, Y.-S.; Yoon, J.-Y. Forecasting the ionosphere F2 Parameters over Jeju Station (33.43◦ N, 126.30◦ E) by Using Long Short-Term Memory. J. Korean Phys. Soc. 2020, 77, 1265–1273. [CrossRef]

[21] Li, X.; Zhou, C.; Tang, Q.; Zhao, J.; Zhang, F.; Xia, G.; Liu, Y. Forecasting Ionosphere foF2 Based on Deep Learning Method. Remote Sens. 2021, 13, 3849. [CrossRef]

[22] Ren, X.; Yang, P.; Liu, H.; Chen, J.; Liu, W. Deep Learning for Global Ionosphere TEC Forecasting: Different Approaches and Validation. Space Weather 2022, 20, e2021SW003011. [CrossRef]

[23] Rao, T.V.; Sridhar, M.; Ratnam, D.V.; Harsha, P.B.S.; Srivani, I. A Bidirectional Long Short-Term Memory-Based Ionosphere foF2 and hmF2 Models for a Single Station in the Low Latitude Region. IEEE Geosci. Remote Sens. Lett. 2022, 19, 1–5.

[24] Tang, Jun, et al. "An Ionosphere TEC Forecasting Model Based on a CNN-LSTM-Attention Mechanism Neural Network." Remote Sensing 14.10 (2022): 2433.

[25] Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." *Advances in neural information processing systems* 30 (2017).

[26] Matlab, Starting. "Matlab." *The MathWorks, Natick, MA* (2012).

[27] Tebabal, A., Radicella, S. M., Nigussie, M., Damtie, B., Nava, B., & Yizengaw, E. (2018). Local TEC modelling and forecasting using neural networks. *Journal of Atmospheric and Solar-Terrestrial Physics*, *172*, 143-151.

[28] Panchal, Gaurang, et al. "Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers." *International Journal of Computer Theory and Engineering* 3.2 (2011): 332-337.

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

```
Sub Macro1()

Dim Year1 As Integer

Dim Year2 As Integer

Dim Day As Integer

Dim Trim As String

Dim Num(1 To 7) As String

Dim Num1(1 To 4) As String

Dim X As String

Num(1) = 1

Num(2) = 3

Num(3) = 5

Num(4) = 7

Num(5) = 8

Num(6) = 10

Num(7) = 12

Num1(1) = 4
```

```
Num1(2) = 6

Num1(3) = 9

Num1(4) = 11

Dim Month As Variant

Dim Month1 As Variant

Year1 = 20220

Year2 = 2022

Year3 = 220

Year4 = 22

Trim = "Trim"



'Months with 31 days

For Each Month In Num

    If Month = 1 Or Month = 3 Or Month = 5 Or Month = 7 Or Month = 8 Then

For Day = 1 To 31 'single digit months

    If Day <= 9 Then


        Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month & "0" & Day & "0000C" & "\" & Trim & Year1 & Month & "0" & Day & "0000C.22N" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month & "0" & Day & "0000C" & "\" & Trim & "_" & Year3 & Month & "0" & Day & ".22N"
```

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month & "0" & Day & "0000C" & "\" & Trim & Year1 & Month & "0" & Day & "0000C.22O" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month & "0" & Day & "0000C" & "\" & Trim & "_" & Year3 & Month & "0" & Day & ".22O"

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month & "0" & Day & "0000C" & "\" & Trim & Year1 & Month & "0" & Day & "0000C.22G" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month & "0" & Day & "0000C" & "\" & Trim & "_" & Year3 & Month & "0" & Day & ".22G"

Else

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month & Day & "0000C" & "\" & Trim & Year1 & Month & Day & "0000C.22N" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month & Day & "0000C" & "\" & Trim & "_" & Year3 & Month & Day & ".22N"

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month & Day & "0000C" & "\" & Trim & Year1 & Month & Day & "0000C.22O" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month & Day & "0000C" & "\" & Trim & "_" & Year3 & Month & Day & ".22O"

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month & Day & "0000C" & "\" & Trim & Year1 & Month & Day & "0000C.22G" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month & Day & "0000C" & "\" & Trim & "_" & Year3 & Month & Day & ".22G"

End If

Next Day

Day = 0

End If

If Month = 10 Or Month = 12 Then 'Double Digited months

For Day = 1 To 31

If Day <= 9 Then

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month & "0" & Day & "0000C" & "\" & Trim & Year2 & Month & "0" & Day & "0000C.22N" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month & "0" & Day & "0000C" & "\" & Trim & "_" & Year4 & Month & "0" & Day & ".22N"

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month & "0" & Day & "0000C" & "\" & Trim & Year2 & Month & "0" & Day & "0000C.22O" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month & "0" & Day & "0000C" & "\" & Trim & "_" & Year4 & Month & "0" & Day & ".22O"

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month & "0" & Day & "0000C" & "\" & Trim & Year2 & Month & "0" & Day & "0000C.22G" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month & "0" & Day & "0000C" & "\" & Trim & "_" & Year4 & Month & "0" & Day & ".22G"

Else

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month & Day & "0000C" & "\" & Trim & Year2 & Month & Day & "0000C.22N" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month & Day & "0000C" & "\" & Trim & "_" & Year4 & Month & Day & ".22N"

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month & Day & "0000C" & "\" & Trim & Year2 & Month & Day & "0000C.22O" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month & Day & "0000C" & "\" & Trim & "_" & Year4 & Month & Day & ".22O"

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month & Day & "0000C" & "\" & Trim & Year2 & Month & Day & "0000C.22G" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month & Day & "0000C" & "\" & Trim & "_" & Year4 & Month & Day & ".22G"

End If

Next Day

```
        Day = 0

    End If

    Next Month

'------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------'

'Months with 30 days

For Each Month1 In Num1

If Month1 = 4 Or Month1 = 6 Or Month1 = 9 Then

For Day = 1 To 30 'single digit months

    If Day <= 9 Then
```



```
        Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month1 & "0" & Day & "0000C" & "\" & Trim & Year1 & Month1 & "0" & Day & "0000C.22N" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month1 & "0" & Day & "0000C" & "\" & Trim & "_" & Year3 & Month1 & "0" & Day & ".22N"

        Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month1 & "0" & Day & "0000C" & "\" & Trim & Year1 & Month1 & "0" & Day & "0000C.22O" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month1 & "0" & Day & "0000C" & "\" & Trim & "_" & Year3 & Month1 & "0" & Day & ".22O"

        Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month1 & "0" & Day & "0000C" & "\" & Trim & Year1 & Month1 & "0" & Day & "0000C.22G" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month1 & "0" & Day & "0000C" & "\" & Trim & "_" & Year3 & Month1 & "0" & Day & ".22G"
```

Else

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month1 & Day & "0000C" & "\" & Trim & Year1 & Month1 & Day & "0000C.22N" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month1 & Day & "0000C" & "\" & Trim & "_" & Year3 & Month1 & Day & ".22N"

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month1 & Day & "0000C" & "\" & Trim & Year1 & Month1 & Day & "0000C.22O" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month1 & Day & "0000C" & "\" & Trim & "_" & Year3 & Month1 & Day & ".22O"

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month1 & Day & "0000C" & "\" & Trim & Year1 & Month1 & Day & "0000C.22G" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & Month1 & Day & "0000C" & "\" & Trim & "_" & Year3 & Month1 & Day & ".22G"

End If

Next Day

'November

Else

For Day = 1 To 30

If Day <= 9 Then

X = 1

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month1 & "0" & Day & "0000C" & "\" & Trim & Year2 & Month1 & "0" & Day & "0000C.22N" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month1 & "0" & Day & "0000C" & "\" & Trim & "_" & Year4 & Month1 & "0" & Day & ".22N"

Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month1 & "0" & Day & "0000C" & "\" & Trim & Year2 & Month1 & "0" & Day & "0000C.22O" As "C:\Users\user\Desktop\OneDrive_2023-

```
06-22\Utem gps data2022\" & Trim & Year2 & Month1 & "0" & Day & "0000C" & "\" & Trim & "_" & Year4 & Month1 & "0" & Day & ".22O"

            Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month1 & "0" & Day & "0000C" & "\" & Trim & Year2 & Month1 & "0" & Day & "0000C.22G" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month1 & "0" & Day & "0000C" & "\" & Trim & "_" & Year4 & Month1 & "0" & Day & ".22G"

        Else

            Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month1 & Day & "0000C" & "\" & Trim & Year2 & Month1 & Day & "0000C.22N" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month1 & Day & "0000C" & "\" & Trim & "_" & Year4 & Month1 & Day & ".22N"

            Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month1 & Day & "0000C" & "\" & Trim & Year2 & Month1 & Day & "0000C.22O" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month1 & Day & "0000C" & "\" & Trim & "_" & Year4 & Month1 & Day & ".22O"

            Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month1 & Day & "0000C" & "\" & Trim & Year2 & Month1 & Day & "0000C.22G" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year2 & Month1 & Day & "0000C" & "\" & Trim & "_" & Year4 & Month1 & Day & ".22G"

        End If

      Next Day

    End If

    Next Month1

    '--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------'
```

```
'February

For Day = 1 To 28

    If Day <= 9 Then

        Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & "20" & Day & "0000C" & "\" & Trim & Year1 & "20" & Day & "0000C.22N" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & "20" & Day & "0000C" & "\" & Trim & "_" & Year3 & "20" & Day & ".22N"

        Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & "20" & Day & "0000C" & "\" & Trim & Year1 & "20" & Day & "0000C.22O" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & "20" & Day & "0000C" & "\" & Trim & "_" & Year3 & "20" & Day & ".22O"

        Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & "20" & Day & "0000C" & "\" & Trim & Year1 & "20" & Day & "0000C.22G" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & "20" & Day & "0000C" & "\" & Trim & "_" & Year3 & "20" & Day & ".22G"

    Else

        Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & "2" & Day & "0000C" & "\" & Trim & Year1 & "2" & Day & "0000C.22N" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & "2" & Day & "0000C" & "\" & Trim & "_" & Year3 & "2" & Day & ".22N"

        Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & "2" & Day & "0000C" & "\" & Trim & Year1 & "2" & Day & "0000C.22O" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & "2" & Day & "0000C" & "\" & Trim & "_" & Year3 & "2" & Day & ".22O"

        Name "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & "2" & Day & "0000C" & "\" & Trim & Year1 & "2" & Day & "0000C.22G" As "C:\Users\user\Desktop\OneDrive_2023-06-22\Utem gps data2022\" & Trim & Year1 & "2" & Day & "0000C" & "\" & Trim & "_" & Year3 & "2" & Day & ".22G"

  End If

  Next Day

End Sub
```

Appendix B

```
import os
import shutil

# Set the source directory where you want to search for ".22O"
files
source_directory = "C:\\Users\\user\\Desktop\\OneDrive_
2023-06-22\\Utem gps data2022"

# Set the destination directory where you want to copy the
".22O" files
destination_directory = "C:\\Users\\user\\Desktop\\OneDrive_
2023-06-22\\ALL 22O"

# Create the destination directory if it doesn't exist
if not os.path.exists(destination_directory):
    os.makedirs(destination_directory)

# Iterate through the files in the source directory
for root, dirs, files in os.walk(source_directory):
    for filename in files:
        if filename.endswith(".22N"):
            # Build the full path to the ".22O" file
            file_path = os.path.join(root, filename)

            # Copy the ".22O" file to the destination
directory
            shutil.copy(file_path,
os.path.join(destination_directory, filename))

print(".22O files copied to the destination directory.")
```

Appendix C

```
import os
import pandas as pd

# Set the source folder path containing the ".std" files
source_folder = "C:\\Users\\user\\Desktop\\OneDrive_2023-06-22
\\ALL 22O"  # Change to your source folder path

# Set the destination folder path where the ".xlsx" files will
be saved
dest_folder = "C:\\Users\\user\\Desktop\\OneDrive_2023-06-22
\\New excel std"  # Change to your destination folder path

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

# Iterate through the ".std" files in the source folder
for root, dirs, files in os.walk(source_folder):
    for file in files:
        if file.endswith(".Std"):
            file_path = os.path.join(root, file)

            # Read the content of the ".std" file into a
DataFrame
            df = pd.read_csv(file_path, delimiter='\t',
encoding='utf-8')

            # Extract the file name without the extension
            file_name = os.path.splitext(file)[0]

            # Save the DataFrame to an Excel file in the
destination folder
            dest_file = os.path.join(dest_folder,
f"{file_name}.xlsx")
            df.to_excel(dest_file, index=False)

print("All '.std' files saved as '.xlsx' worksheets.")
```

Appendix D

```
import os
import openpyxl

# Set the source folder path containing the Excel files
source_folder = "C:\\Users\\user\\Desktop\\OneDrive_2023-06-22
\\New excel std"  # Replace with your source folder path
output_file = "consolidated_data.xlsx"  # Replace with your
desired output file name

# Initialize a new workbook for consolidation
consolidated_workbook = openpyxl.Workbook()
consolidated_worksheet = consolidated_workbook.active

# Define the indices of the cells to extract
cell_indices = [1, 61, 121, 181, 241, 301, 361, 421, 481, 541,
601, 661, 721, 781, 841, 901, 961, 1021, 1081, 1141, 1201,
1261, 1321, 1381]

# Function to extract specific cells, convert to numbers, and
consolidate data
def extract_convert_consolidate(file_path, consolidated_ws):
    source_workbook = openpyxl.load_workbook(file_path)
    source_worksheet = source_workbook.active

    for index in cell_indices:
        cell = source_worksheet.cell(row=index, column=2)
        if cell.value is not None and cell.value.replace('.',
'', 1).isdigit():
            consolidated_ws.append([float(cell.value)])

# Iterate through Excel files in the source folder
for root, dirs, files in os.walk(source_folder):
    for file in files:
        if file.endswith(".xlsx"):
            file_path = os.path.join(root, file)
            extract_convert_consolidate(file_path,
consolidated_worksheet)
```

Appendix E

```
import os
import openpyxl

# Set the source folder path containing the Excel files
source_folder = "C:\\Users\\user\\Desktop\\OneDrive_2023-06-22
\\New excel std"  # Change to your source folder path

# Function to check the number of rows in an Excel file
def check_rows(file_path):
    try:
        workbook = openpyxl.load_workbook(file_path)
        worksheet = workbook.active
        num_rows = worksheet.max_row
        if num_rows != 1440:
            print(f"File '{os.path.basename(file_path)}' does
not have 1440 rows.")
    except Exception as e:
        print(f"Error occurred while checking
'{os.path.basename(file_path)}': {e}")

# Iterate through Excel files in the source folder
for root, dirs, files in os.walk(source_folder):
    for file in files:
        if file.endswith(".xlsx"):
            file_path = os.path.join(root, file)
            check_rows(file_path)
```

Appendix F: Gantt Chart