SHORT-TERM FORECASTING OF GLOBAL HORIZONTAL IRRADIANCE USING STACKED ENSEMBLE MACHINE LEARNING ALGORITHMS

By

Fhulufhelo Walter Mugware
22020643

Mini dissertation for the Master of Science Degree in E-Science

in the

Department of Mathematical and Computational Sciences,
Faculty of Science, Engineering and Agriculture
University of Venda, Thohoyandou, Limpopo
South Africa

Supervisor: **Dr T Ravele**

Co-Supervisor: **Prof C Sigauke**

February 16, 2025

Declaration

I, Fhulufhelo Walter Mugware, with student number 22020643, affirm that this research is entirely my own creation and has not been presented for any academic qualification at any other educational institution. It does not incorporate the written work of others unless duly acknowledged and cited as appropriate.

Signed (Student): ..... Date: 16/02/2025

Abstract

In today's world, where sustainable energy is essential for the planet's survival, accurate solar energy forecasting is crucial. This study focused on predicting short-term Global Horizontal Irradiance (GHI) using data from the Southern African Universities Radiometric Network (SAURAN) at the Univen Radiometric Station in South Africa. Various techniques were evaluated for their predictive accuracy, including Recurrent Neural Networks (RNN), Support Vector Regression (SVR), Gradient Boosting (GB), Random Forest (RF), Stacking Ensemble, and Double Nested Stacking (DNS). The results indicated that RNN performed the best in terms of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) among the machine learning models. However, Stacking ensembles with XGBoost as the meta-model outperformed all individual models, improving accuracy by 67.06% in MAE and 22.28% in RMSE. DNS further enhanced accuracy, achieving a 93.05% reduction in MAE and an 88.54% reduction in RMSE compared to the best machine learning model, as well as a 78.89% decrease in MAE and an 85.27% decrease in RMSE compared to the best single stacking model. Furthermore, experimenting with the order of the DNS meta-model revealed that using RF as the first-level meta-model followed by XGBoost yielded the highest accuracy, showing a 47.39% decrease in MAE and a 61.35% decrease in RMSE compared to DNS with RF at both levels. These findings underscore the potential of advanced stacking techniques to significantly improve GHI forecasting.

Keywords: *Double Nested Stacking, GB, GHI, Machine learning, RNN, RF, SAURAN, Solar energy, Stacking ensemble, SVR.*

Dedication

This work is dedicated to my late father, Ntshengedzeni Jameson Mugware, and my entire family.

Acknowledgment

I would like to start by expressing my gratitude to God for His unwavering support throughout this journey. There were times when I felt drained, but His love and guidance gave me the strength to persevere.

I also want to thank my supervisor, Dr.T. Ravele, for the continuous support and invaluable guidance, and my co-supervisor, Prof. C. Sigauke, for constantly motivating me to learn and grow. This project would not have been possible without their dedication and expertise.

Also want to thank my sister, who helped me begin my academic journey, and my mother, Ntsieni Mugware, who has been nothing but supportive. I also want to thank my brother, Anza Mugware, who always pushes me to work hard and wants nothing but the best for me. Lastly, I offer my heartfelt dedication to my sister, Mpho Mugware, and my little brother, Arendwaho Mugware.

I also want to give special thanks to my fellow students, Ndivhuwo and Asakundwi from the e-Science group of 2023, for their continuous assistance and collaboration throughout this project. Lastly I want to sincerely thank the University of Venda for welcoming me in to their academic community and the DST-CSIR National e-Science Postgraduate Teaching and Training Platform (NEPTTP) for their vital financial support.

Table of contents

List of Figures

List of Tables

Abbreviations

GHI	Global Horizontal Irradiance
RNN	Recurrent Neural Network
SAURAN	Southern African Universities Radiometric Network
GB	Gradient Boosting
RF	Random Forest
SE	Stacking Ensemble
DNS	Double Nested Stacking
SVR	Support Vector Regression
LSTM	Long Short-Term Memory
ANN	Artificial Neural Network
GPR	Gaussian Process Regression
nRMSE	Normalized Root Mean Square Error
DMAE	Dynamic Mean Absolute Error
DNN	Deep Neural Network-based
DHI	Diffuse Horizontal Irradiation
BMI	Beam Normal irradiation
SP	Smart Persistence
FFN	Feed Forward Network
ARIMA	Auto Regressive Integrated Moving Average
AdaBoost	Adaptive Boosting
MLP	Multi-Layer Perceptron
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
XGBOOST	Extreme Gradient Boosting
LGBM	Light Gradient Boosting Machine
GBDT	Gradient Boosting Decision Tree
SVM	Support Vector Machines

RMAE	Relative Mean Absolute Error
RRMSE	Relative Root Mean Squared Error
ACF	Autocorrelation Function
PACF	Partial Autocorrelation Function
EDA	Exploratory Data Analysis
SE XG	Stacking Ensemble XGBOOST
SE RF	Stacking Ensemble Random Forest
DNS XG	Double Nested Stacking XGBOOST
DNS RF	Double Nested Stacking Random Forest
DNS XG RF	Double Nested Stacking XGBOOST Random Forest
DNS RF XG	Double Nested Stacking Random Forest XGBOOST

Chapter 1

Introduction

Renewable energy will play a crucial role in the future by offering sustainable, environmentally friendly alternatives to fossil fuels, promoting energy security, economic growth, and mitigating the impacts of climate change. Energy resources can be divided into three categories: fossil fuels, renewable, and nuclear energy (Demirbas, 2000). Renewable energy has a significant advantage over traditional fossil fuels since it can produce energy with considerably fewer carbon emissions. This makes it a crucial strategy for countries to reduce their carbon footprint.

Many countries are now emphasising the use of renewable energy as one of the key ways to reduce emissions further. Although fossil fuels have been the primary source of energy for our civilisation, rapid population growth and convenience have led to severe consequences. However, extensive research has been conducted over the past years, highlighting the adverse effects of using fossil fuels, making renewable energy a more viable and sustainable option (Armaroli and Balzani, 2011).

Although we have different types of renewable energy, there is one that stands out which is solar energy. This is due to many reasons, such as it being clean and available in most places. Solar energy is derived from the sun and is a key source of unlimited free energy available on Earth. (Sayigh, 2012). According to Guangul and Chala (2019) in 90 minutes, the sun can supply energy that is enough to meet the required demand of the whole planet for one year, this fact underscores the immense potential and significant impact of solar energy as a renewable and sustainable energy source.

While solar energy offers significant benefits, there are notable concerns such as the expense associated with harvesting it. However, this concern is gradually diminishing as nations and corporations globally invest substantial sums of money into solar energy initiatives. Consequently, advancements in technology are occurring, and the operational costs of solar energy are decreasing over time (Philibert et al., 2011).

1.1 Background

It is crucial to understand the generation of solar energy, as this knowledge can aid in various aspects such as storage, and integration. Moreover, it can help those in charge to be better prepared for any unexpected circumstances. Several studies have been conducted on solar energy, and they have revealed that Global Horizontal Irradiance (GHI) is one of the strongest predictors of actual solar energy generation hence, there is a need for forecasting GHI (Kumar Barik et al., 2021).

GHI refers to the total solar radiation per unit area received on a horizontal plane at the Earth's surface. Accurately predicting GHI has numerous benefits, including optimised energy production, improved grid integration, enhanced energy management, and cost reduction (Aliberti et al., 2021).

1.2 Problem Statement

Accurate forecasting of solar energy is essential for effective energy planning, grid integration, and the cost-efficient use of renewable energy. However, the current forecasting models for GHI in South Africa primarily depend on individual machine learning methods. These models often struggle to capture the complex, non-linear relationships present in solar radiation data. Therefore, there is an increasing demand to improve the accuracy of GHI forecasts by employing advanced ensemble techniques.

1.3 Purpose of the study

1.3.1 Aim

This research project aims to compare the predictive power of Double Nested Stacking(DNS), traditional stacking, RNN , SVR, RF, and GB on forecasting GHI at SAURAN Univen radiometric station.

1.3.2 Objectives

The main objectives of this study are to:

I. develop individual models which are RNN, SVR, RF, and GB,

II. create an ensemble stacking model and explore different base models to assess their impact on performance,

III. develop a DNS model integrating base and base models,

IV. evaluate and compare the performance of all developed models.

1.3.3 Significance of the Study

This study is significant because it contributes to efforts aimed at improving renewable energy forecasting, especially in South Africa. Accurately predicting GHI has direct implications for solar power generation, energy storage, and grid stability. By developing an advanced ensemble modeling approach, this research provides a novel solution to enhance forecasting accuracy. The findings can assist energy providers in optimising solar energy production, reducing reliance on fossil fuels, and improving overall sustainability in energy management.

1.3.4 Contribution to knowledge

This research advances the fields of machine learning and renewable energy forecasting by introducing a sophisticated stacking ensemble framework specifically designed for GHI prediction. While traditional stacking methods have been previously studied, this study enhances the approach by incorporating a DNS model, which has not been widely used in this context. The findings offer valuable insights into the effects of various meta-model combinations, presenting a new methodological framework that can be adapted

for other time-series forecasting applications within the energy sector.

1.4 Scope of the Study

The research is structured into five chapters that comprehensively cover all aspects of the subject. Chapter 2 presents a detailed analysis of literature related to GHI, RNN, SVR, RF, GB, and stacked ensemble algorithms, along with studies that have utilised these methods. Chapter 3 provides a general overview of the machine learning techniques and stacking ensemble methods that will be used in this research study. Chapter 4 covers research findings, and lastly, Chapter 5 will summarise the findings and make recommendations and improvements on future studies.

Chapter 2

Literature review

2.1 Introduction

This chapter will provide an overview of the studies conducted related to short-term forecasting of GHI using machine learning and stacking ensemble methods.

2.2 An overview of GHI forecasting using machine learning algorithms

2.2.1 State-of-the-art in short-term forecasting of GHI using machine learning.

Numerous researchers from South Africa and around the world have been studying the forecasting of GHI and experimenting with different approaches, such as machine learning and stacking ensemble models, to enhance the accuracy of forecasts.

One such study was conducted by Gbémou et al. (2021), who compared ma-

chine learning models with a scaled persistence model in forecasting GHI. The study analysed the performance of artificial neural networks (ANN), Gaussian process regression (GPR), SVR, and Long Short-Term Memory (LSTM) models. Two years' worth of GHI data was used to train the models. Following the application of the coverage width-based criterion, dynamic mean absolute error (DMAE), and normalised root mean square error (nRMSE), the results showed that the machine learning models performed better than the scaled persistence model, with the SVR, LSTM, and GPR models being the top performers.

A recent study conducted in India by Rajaprasad and Mukkamala (2023) focused on short-term forecasting of GHI using machine learning methods. The study proposed a new hybrid deep neural network-based (DNN) model that combined convolutional neural network bi-directional LSTM (CNN BiLSTM), and compared it with LSTM and BiLSTM, which are also deep learning models. The study used one-minute interval GHI data for January 2023 to train and test these models. The proposed hybrid model outperformed LSTM and BiLSTM. This suggests that combining different machine learning methods can enhance the accuracy of GHI forecasting (Rajaprasad and Mukkamala, 2023).

In a study conducted by Yamani and Alyami (2021), it was noted that deep learning models, particularly LSTM, have been successful in forecasting tasks. However, the accuracy of the forecasting results depends not only on the robustness of the model but also on the amount of training data fed

into the model. Most LSTM models have been found to require a relatively large amount of data. The study's authors experimented with the accuracy of LSTM depending on the training data used. The study utilised three datasets of historic GHI from Saudi Arabia. The aim was to determine the minimum amount of data required for forecasting one hour ahead. This was achieved by training the LSTM with five years of data and gradually reducing the amount of data fed into the model to compare the accuracy. Using nRMSE, the study found that LSTM can achieve excellent GHI forecasting accuracy with at least two years of training data.

In a recent study, researchers aimed to forecast not only the GHI but also the Diffuse Horizontal Irradiance (DNI) and Beam Normal Irradiance (BNI). They sought to predict hourly solar irradiation for different time horizons, including one hour and six hours ahead. The study utilised data from Odeillo, France, which exhibits high meteorological variability. The researchers trained the models using RF, Artificial Neural Networks (ANN), and Smart Persistence (SP) and assessed their performance using the Normalised Root Mean Square Error (NRMSE). The results indicated that RF accurately predicted GHI, DNI, and BNI compared to ANN and SP. Additionally, the study conducted seasonal analysis, revealing that winter and summer are the best seasons for forecasting, as opposed to spring and autumn. Furthermore, the predictability of the three components was compared, showing that GHI is less complex to predict compared to BNI and DNI, as the latter two are more sensitive to meteorological conditions (Benali et al., 2019).

In a recent study conducted in India by Krishnan et al. (2024), researchers investigated solar radiation and focused on forecasting hourly GHI. The historic data used covered all climatic zones from 1995 to 2014. The primary goal of the study was to develop a high-performing model based on GB that also required less computational time. The GB model was benchmarked against a two-layer feed-forward neural network (FFN) and Auto-Regressive Integrated Moving Average (ARIMA). The results indicated that the GB model outperformed the FFN and displayed better accuracy compared to the ARIMA model, as MSE and MAE. The developed model and its architecture are intended to have practical implications.

2.3 An overview of forecasting using ensemble stacking techniques

Stacking ensemble models, similar to machine learning models, are being used for forecasting in various domains. One of the most significant areas of application for these models is in the field of renewable energy. However, there is a lack of research that has employed stacking ensemble methods for forecasting GHI.

Nziyumva et al. (n.d.) implemented stacking ensemble methods in forecasting to improve the accuracy of solar irradiance prediction. They evaluated the efficacy of the stacking ensemble technique against single models like the Multi-Layer Perceptron (MLP), Bootstrap aggregating (Bagging) regressor, and Adaptive Boosting (AdaBoost). The same models were combined through stacking ensemble, and the resulting ensemble method was evaluated

using determination coefficient, mean absolute error (MAE), and root mean squared error (RMSE). The researchers found that the stacking ensemble method outperformed all the single models, thus improving the prediction accuracy.

In a study by Guo et al. (2020), forecasting accuracy was investigated using the stacking ensemble principle. The study proposed using a single stacking ensemble model to predict PV power. This model was chosen for its ability to combine different principles and characteristics of various models to produce accurate predictions. Individual training was conducted for the models (XGBOOST, RF, CATBoost, LGBM) that would be stacked in the ensemble model to ensure a fair assessment of prediction accuracy. These models were combined using SVR for the second level and then compared with the stacking ensemble model. The data for training these models was collected from the Data Collection System (DCS) and included meteorological data. The prediction accuracy of power generation was considered in different weather settings, including rainy and sunny days. After combining these models and using RMSE to compare the results, it was found that XGBoost demonstrated superiority compared to each model. Furthermore, compared to the stacking ensemble, the RMSE of the stacking ensemble was 1.84% lower.

In a study conducted by Zhou et al. (2023), a new ensemble model called DNS was proposed for photovoltaic power forecasting. The DNS model used base models to generate predictions at the first level, which are then used as features to train higher-level meta-models. This results in a more sophis-

ticated ensemble with enhanced predictive capabilities, aiming to improve accuracy. The proposed DNS model outperformed the traditional stacking ensemble model, which was built based on a gradient boosting decision tree (GBDT), XGBoost, and SVR. The study used PV power station data from 2019 to the present day and validated the effectiveness of the DNS model in improving forecasting accuracy.

2.3.1 Research gap(s)

Upon examining the literature, it becomes evident that there is a significant gap in the application of stacking ensemble techniques in this field. While individual machine learning models have demonstrated effectiveness, the potential advantages of ensemble learning, particularly stacking, have largely been overlooked. Addressing this gap is crucial, as utilising stacking ensemble methods could greatly enhance the accuracy and reliability of GHI predictions. This improvement would advance renewable energy forecasting and make energy generation planning and management more dependable.

2.3.2 Contribution from this work

This study enhances GHI forecasting by introducing a DNS approach, which improves predictive accuracy beyond conventional stacking techniques. It examines the effects of the ordering of meta-models, utilises Bayesian optimisation for hyperparameter tuning, and refines model input selection through advanced feature selection methods. These findings provide a scalable methodology for improving solar energy predictions in the South African context.

2.4 Conclusions from literature

This chapter discussed the literature on using machine learning algorithms for predicting GHI and reviews studies that have implemented stacking ensemble methods in forecasting. Additionally, it highlights a literature gap that this study aims to address.

Chapter 3

Methodology

3.1 Introduction

This chapter will cover the different methods and techniques employed in this study to forecast GHI. This includes machine learning models such as RNN, SVR, GB, and Stacking ensemble models such as traditional Stacking ensemble and DNS. We will also explore techniques for parameter tuning, variable selection and assessing forecast accuracy.

3.2 Data source

The study will be utilising GHI data from the SAURAN Univen radiometric station. The data is minute-averaged and consists of various variables such as temperature, humidity, wind speed, cloud cover and many more. The data can be accessed through this website: https://sauran.ac.za/.

3.3 Models

The following is a list of machine learning techniques and stacking ensemble models that will be utilised to forecast GHI in this project these models are explained further and also explored how they function.

3.3.1 Recurrent neural networks

The ANN models have been extensively studied to achieve human-like performance, particularly in the field of pattern recognition. Among these, RNN are a type of ANN that excel as a powerful tool for processing time series data. This is due to their inherent ability to capture temporal dependencies within sequential data. The structure of RNN is shown in the Figure 3.1. Unlike traditional feedforward neural networks, RNNs have loops within their architecture, which allow them to maintain a memory of past inputs while processing new ones (Mandic and Chambers, 2001).

RNN is defined as folllows :

$$y_j(t+1) = \varphi(\sum_{i=1}^{m+n} w_{ij} z_i(t)) \tag{3.3.1}$$

$$z_i(t) = \begin{cases} y_i(t) & \text{if } (i \geq n) \\ u_{i-n} & \text{if } (i > n), \end{cases} \tag{3.3.2}$$

where m stands for proportion of inputs, and n for the hidden and output neurons, φ is the arbitrary differential component, generally a sigmoid function, y_j determines the output of the j^{th} neuron and w_{ij} the relationship between the i^{th} and the j^{th} neuron (Garcia-Pedrero and Gomez-Gil, 2010).

An RNN achieves its capability through recurrent neural connections. A fundamental equation for processing an input sequence $x = (x_1, x_2, \ldots, x_T)$ to determine the RNN hidden state h_t is:

$$h_t = \begin{cases} 0, & \text{if } t = 0 \\ \Phi(h_{t-1}, x_t), & \text{otherwise} \end{cases} \tag{3.3.3}$$

where the Φ function is non-linear. Recurrent hidden state update is realised as follows:

$$h_t = g(wx_t + uh_{t-1}), \tag{3.3.4}$$

where g is a function of the hyperbolic tangent. Generally, this type of recurrent neural network environment without neurons often suffers from gradient issues that are difficult to resolve.

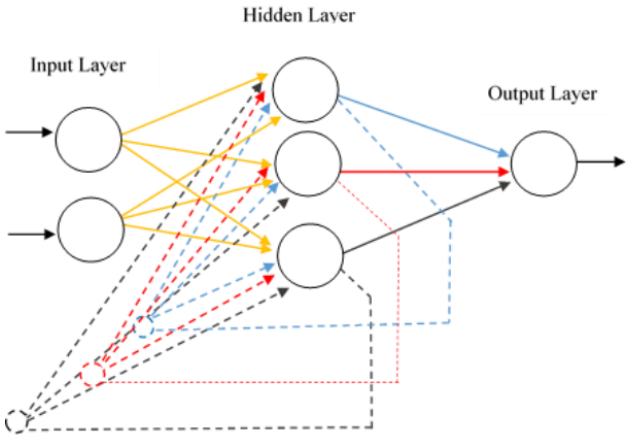


Figure 3.1: RNN structure. (Source:(Petneházi, 2019))

3.3.2 Support Vector Regression

Another model that will be considered in this study is SVR which was actually an extension from the model Support vector machines (SVM) which was introduced by Vapnik et al. (1998) in the 90s. SVM was designed for binary object classification then it was adapted into prediction model, the concept of this was done or initiated by Drucker et al. (1996) and the model became SVR.

SVR operates by identifying a hyperplane in a high-dimensional space that best fits the given data points while minimising the error within a specified margin known as epsilon. The goal is to strike a balance between the model's complexity and the accuracy of data fitting. SVR utilises support vectors to define the decision boundary, offering several advantages over other machine learning models.It is especially effective for high-dimensional data and performs well even when the number of dimensions exceeds the number of samples. Furthermore, SVR can manage non-linear relationships by employing kernel functions, enabling it to model complex patterns that linear regression might miss (Aghelpour et al., 2019).

The mathematical formulation of SVR is expressed as:

$$f(x) = \omega\phi(x) + b, \tag{3.3.5}$$

where ω represents the weight vector, $\phi(x)$ is a function that maps the input x into a high-dimensional feature space, which is a nonlinear transformation of the original input space. Additionally, b denotes the bias term.

To calculate the coefficients ω and b, it is required to reduce the regularised risk function which can be expressed as

$$\frac{1}{2}\|\omega\|^2 + C\frac{1}{l}\sum_{i=1}^{l} L_\epsilon(y_i, f(x_i)), \tag{3.3.6}$$

where $\|\omega\|^2$ is a regularised term which maintains the function capacity. C is a cost error. The empirical term from the second term in equation 3.3.6 can be defined as

$$L_\epsilon(y_i, f(x_i)) = \begin{cases} 0, & \text{if } |y_i - f(x_i)| = \epsilon, \\ |y_i - f(x_i)| \geq \epsilon. \end{cases} \tag{3.3.7}$$

Equation 3.3.7 expressed the transformation of the primal objective function in order to get the values of ω and b by introducing the positive slack variables $\xi_i^*(\cdot)$.

$$\text{minimise } \frac{1}{2}\|\omega\|^2 + C\frac{1}{l}\sum_{i=1}^{l}(\epsilon_i + \xi_i^*) \tag{3.3.8}$$

subject to

$$\alpha(x) = \begin{cases} y_i - \langle \omega, x_i \rangle - b \leq \epsilon + \xi_i, \\ \langle \omega, x_i \rangle + b - y_i \leq \epsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0. \end{cases} \tag{3.3.9}$$

The optimisation problem presented in equation 3.3.9 must be transformed into its dual formulation using Lagrange multipliers for a more efficient solution.

$$L = \frac{1}{2}\|\omega\|^2 + C\frac{1}{l}\sum_{i=1}^{l}(\epsilon_i + \xi_i^*) - \sum_{i=1}^{l}(\eta_i\epsilon_i + \eta_i^*\xi_i^*)$$
$$- \sum_{i=1}^{l} a_i(\epsilon + \xi_i + y_i + \omega \cdot \phi(x_i) + b)$$
$$- \sum_{i=1}^{l} a_i^*(\epsilon + \xi_i^* - y_i - \omega \cdot \phi(x_i) + b), \tag{3.3.10}$$

where L is the Lagrange and η_i, η_i^* are the Lagrange multipliers. Hence the dual variables in equation 3.3.10 have to satisfy positive constraints, $\eta_i^*, a_i^* \geq 0$.

The resulting SVR model can be expressed as follows:

$$f(x) = \sum_{i=1}^{l}(a_i - a_i^*)\phi(x_i)\phi(x) + b \tag{3.3.11}$$

3.3.3 Random Forests

A decision tree is a statistical model used for classification, which was introduced by Breiman and Ihaka (1984). Later Breiman (2001) introduced the concept of RF, which is an ensemble learning method. In RF, multiple decision trees are built and their predictions are combined to improve predictive accuracy and the structure of the RF can be seen below Figure 3.2. Each decision tree in a RF is trained on a random subset of the data and a random subset of features, which helps in reducing over-fitting and capturing diverse patterns in the data (Lahouar and Slama, 2015).

This study will focus on the regression aspect of the RF model. In a general

context, a random vector $X \in \mathcal{X} \subseteq \mathbb{R}^p$ is observed, and the objective is to predict the integrable square random response $Y \in \mathbb{R}$ by estimating the regression function using non-parametric regression approximation.

$$m(x) = \mathbb{E}[Y \mid X = x]. \tag{3.3.12}$$

Assuming a training sample $D_n = ((X_1, Y_1), \ldots, (X_n, Y_n))$, which consists of independent random variable pairs of prototypes (X, Y), the dataset D_n is utilized to construct the function $m_n : \mathcal{X} \to \mathbb{R}$. The estimate of the regression function m (mean squared error) is considered compatible if

$$\mathbb{E}[m_n(X) - m(X)]^2 = 0 \quad \text{as} \quad n \to \infty. \tag{3.3.13}$$

A RF is the predictor of the random M regression trees set. The expected value at query point X for the j-th tree in the family is denoted by

$$m_n(X : \Theta_j, D_n), \tag{3.3.14}$$

where $\Theta_1, \ldots, \Theta_M$ is a separate random variable Θ distributed as a generic random variable m and independent of D_n. Variable Θ is used to evaluate the training set before each tree grows and to choose the following instructions, more detailed definitions are given later. The j-th tree estimate takes the mathematical form:

$$m_n(x; \Theta_j, D_n) = \sum_{i \in D_n^*(\Theta_j)} \frac{\mathbf{1}_{x_i \in A_n(x; \Theta_j, D_n)} Y_i}{N_n(x; \Theta_j, D_n)}. \tag{3.3.15}$$

Where:

- D_n^* = the set of data points selected prior to the tree construction,

- $A_n(x;\Theta_j, D_n)$ = the cell containing x, and
- $N_n(x;\Theta_j, D_n)$ = the number of points that fall into $A_n(x;\Theta_j, D_n)$.

The trees are now combined to make the (finite) forest estimate as written:

$$m_{M,n}(x;\Theta_1,\ldots,\Theta_M, D_n) = \frac{1}{M}\sum_{j=1}^{M} m_n(x;\Theta_j, D_n). \tag{3.3.16}$$

Since M can be arbitrarily larger, modelling it makes sense to allow M to be infinitely large and to take into account instead of the forest estimates

$$m_{\infty,n}(x; D_n) = \mathbb{E}_\Theta[m_n(x;\Theta, D_n)]. \tag{3.3.17}$$

Where in equation 3.3.17:

- $\mathbb{E}_\Theta$ = the expectation with respect to the random parameter Θ, conditioned on D_n.

Operation $M \to \infty$ is justified by large numbers, which almost definitely asserts that it is subject to D_n.

$$\lim_{M\to\infty} m_{M,n}(x;\Theta_1,\ldots,\Theta_M, D_n) = m_{\infty,n}(x; D_n). \tag{3.3.18}$$

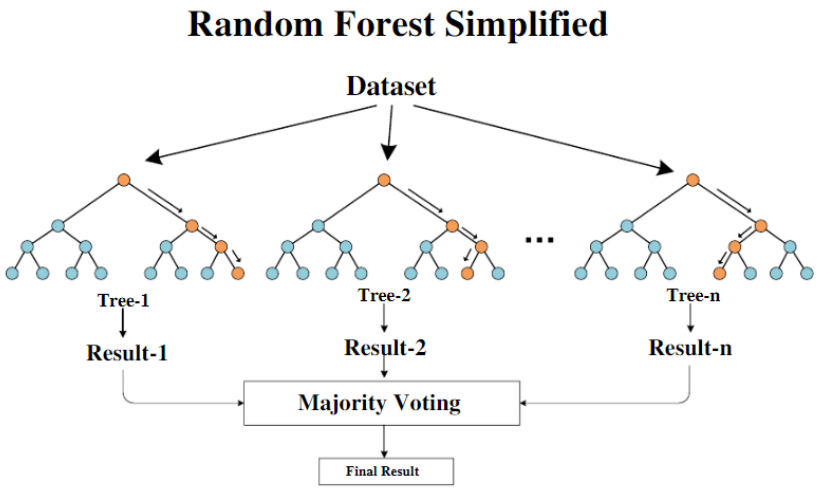


Figure 3.2: Random forest. Source: Dang et al. (2023)

3.3.4 Gradient Boosting Model

History or origin of GB can be traced back to a paper by Friedman (2001) titled *Greedy function approximation: a gradient boosting machine*. GB is an ensemble machine learning technique that can be used for both classification and regression tasks. Its objective is to create an ensemble model that minimises a specified loss function by sequentially adding weak models that correct the errors from previous iterations. GB is particularly effective for forecasting because it can capture complex nonlinear relationships and temporal dependencies in data, making it especially suitable for time series forecasting (Nemalili et al., 2023).

The GB model is defined by:

$$F_M(x) = \sum_{j=M}^{j=1} \rho_i h_i(x), \tag{3.3.19}$$

where $h_i(X)$ are the members of the ensemble and ρ_i, h_i are the weight of each in the ensemble and M is the size of the ensemble (Alcántara et al., 2023).

Training

Based on Friedman (2001), the algorithm of the GB model takes the following steps for the input data, $\{(x_i, y_i)\}_{i=1}^{n}$, and a differentiable loss function, $L(y, F(x))$, which is a squared regression in this study.

Step 1: Initialise the model with a constant value:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma), \tag{3.3.20}$$

where y_i is an observed value, and γ is a predicted value. $F_0(x)$ is the average of the observed values.

Step 2: For $m = 1$ to M:

(A) Compute

$$\gamma_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n \tag{3.3.21}$$

(B) Fit a regression tree to the γ_{im} values and create terminal regions R_{jm} for $j = 1, \ldots, J_m$

(C) For $j = 1, \ldots, J_m$, compute

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \tag{3.3.22}$$

(D) Update

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} \mathbb{I}(x \in R_{jm}), \tag{3.3.23}$$

where ν is the learning rate.

The loss functions can be customised by adjusting the learning rate, ν. This flexibility enhances the model's adaptability and helps reduce overfitting by allowing it to learn more gradually from each iteration. (Hastie et al., 2009).

Step 3: Output:

$$\hat{F}(x) = F_M(x) \tag{3.3.24}$$

After completing all M iterations and updating the $F_m(x)$ function, the final model, $\hat{F}(x)$, represents an approximation of the relationship between the independent and dependent variables.

3.3.5 Stacking ensemble

The concept of stacking was developed by Wolpert (1992), but the theoretical guarantees for stacking were not formally proven until the publication of a paper titled: *Super Learner* by Laan (2007). Stacking ensemble is a technique that enhances overall performance by combining predictions from multiple base models using a meta-model. This approach leverages the unique strengths of individual models and mitigates their weaknesses, resulting in improved predictive accuracy and robustness. Stacking has gained widespread popularity in various domains due to its effectiveness in ensemble learning, outperforming individual models and producing superior results.

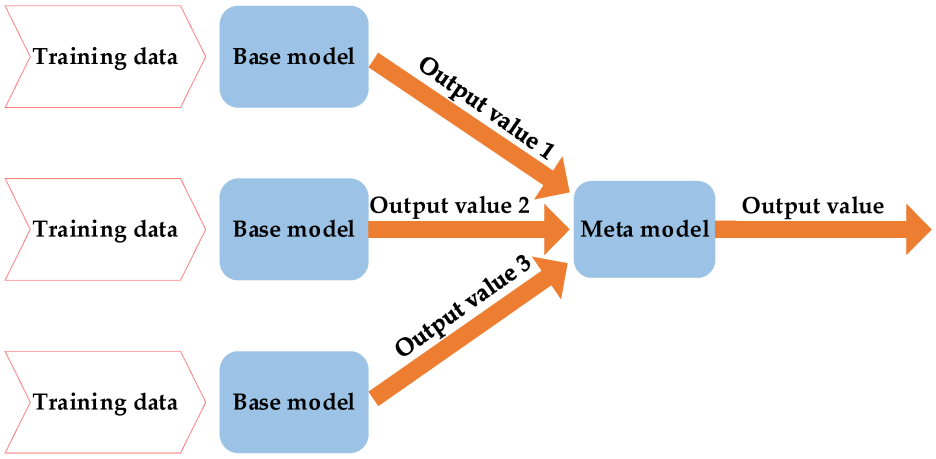


Figure 3.3: Stacking ensemble structure. (Source: (Zhou et al., 2023)

Figure 3.3 displays the architecture of a stacking ensemble model. The training data is used to train the base models, which generate separate predictions on the input data. These predictions are then fed into the meta model, which learns how to combine them to produce the final output value, representing the prediction of the stacking ensemble model (Khandelwal, 2021).

3.3.6 Double nested stacking

Traditional stacking models face challenges because they operate under different assumptions about the distribution of input data, which arises from the use of various underlying models. The simple structure of traditional stacking limits its flexibility and range, often leading to suboptimal performance of the meta-model and an inability to capture important interaction features. To address these issues, we propose the DNS model. In contrast to traditional stacking, DNS offers a more sophisticated and adaptable approach, allowing it to better model complex relationships within the data

and improve overall generalisation performance (Nziyumva et al., n.d.).

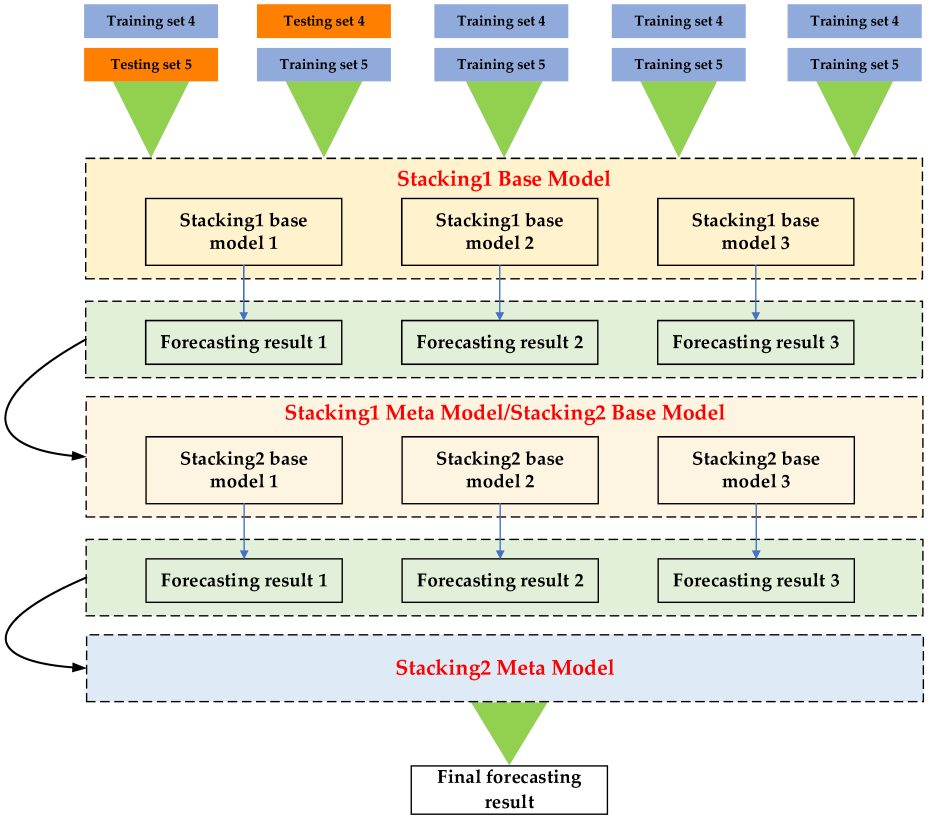


Figure 3.4: DNS model training process. Source: (Zhou et al., 2023)

Figure 3.4 shows the training process of DNS model. The process of DNS model training involves training multiple base models on the given training data. These models make predictions on the data. Next, we repeat the process with another set of base models, which generates another set of predictions. Finally, we use all these predictions as new features and train a meta-model to combine them into one final prediction. This approach leverages the strengths of multiple base models in two separate layers to enhance the accuracy of predictions. (Zhou et al., 2023).

Base models and meta model

In order to stack models, we need to first have base models - these are individual models that have been trained on the original dataset. It's important to have diverse base models in order to leverage the strengths of different models. This diversity helps to reduce errors by averaging them out, thereby reducing the overall variance and bias of the final predictions. In this study, we will use RNN, SVR, GB, and RF as our base models. This will allow us to effectively compare the performance of the individual models with that of the stacking ensemble (Divina et al., 2018).

Another essential component is the meta-model or level-2 model, which is trained on the output of the base models. This meta-model can identify which base models are more reliable for certain parts of the data, leading to improved final predictions. It can also correct errors that were not well handled by the base models.

In order to improve the reliability of the study, we will conduct the experiment using various meta-models to assess their effectiveness. The base models will be trained under the same conditions, with different meta-models such as Ridge regression, XGBOOST, RF, and Elastic Net being. The top two performing models will also be used and further experimented on to see how the meta-models impact the performance of stacking methods. This will help us evaluate the influence of the meta-models on the entire ensemble of models.

3.4 Bayesian optimisation

Parameter tuning is a crucial process in machine learning that helps optimise the performance of models across different tasks. This study will utilise Bayesian optimisation, which is a powerful technique for hyperparameter tuning in machine learning applications. This method of parameters tuning is highly efficient, flexible, and robust. It involves selecting hyperparameter configurations intelligently based on a probabilistic surrogate model, which leads to fewer evaluations of the objective function (Joy et al., 2016).

Bayesian optimisation helps to reduce the time required to obtain an optimal parameter set and improves the performance of test set generalisation tasks. Unlike methods such as grid search or random search, which evaluate hyperparameter sets independently of past trials, Bayesian optimisation considers previously tested hyperparameter combinations to determine the next set of hyperparameters to evaluate. This approach balances the exploration of the search space and the exploitation of known promising regions, leading to faster convergence on an optimal or near-optimal parameter set (Gardner et al., 2014).

The implementation of the Bayesian optimisation in this study is as follows:

Algorithm 1 Bayesian Optimisation for Hyperparameter Tuning

Require: Initial hyperparameter configuration space $\mathcal{H}$, initial observation set $D = \emptyset$, objective function f (model performance metric), number of iterations T.

Ensure: Best hyperparameter configuration $\hat{h}$ found.

1: **for** $n = 1, 2, \ldots, T$ **do**
2: Select hyperparameter configuration h_n using acquisition function based on $\mathcal{H}$ and D.
3: Evaluate model performance: $y_n = f(h_n)$.
4: Augment observation set: $D = D \cup (h_n, y_n)$.
5: **end for**
6: **return** Best hyperparameter configuration $\hat{h}$ based on observed performances in D.

3.5 Variable selection

Choosing the right variables is crucial when building a strong model. There are various methods for variable selection, such as Lasso and Ridge regression, but they can assume linear relationships. RF, on the other hand, is ideal for complex and non-linear relationships which will be employed in this study.

RF constructs several decision trees during the training process and then merges their outputs to make a final decision. It assesses how each variable contributes to reducing impurity in the nodes of each tree. Variables that consistently appear at the top of the trees or play a significant role in splitting data into similar subsets are considered more important. The algorithm evaluates the importance of features by calculating the average reduction in impurity or accuracy when a particular variable is included (Fox et al., 2017).

By ranking the features, RF helps in selecting the most relevant variables.

This, in turn, reduces dimensionality, improves model performance, and helps interpret the underlying data patterns. Moreover, RF is robust against overfitting and can handle high-dimensional data with complex interactions, making it a versatile choice for variable selection in various applications (Genuer et al., 2010).

3.6 Metrics for evaluating forecasts

To evaluate the effectiveness of our models and determine which one is the most accurate, we will use several metrics: MAE, Relative Mean Absolute Error (RMAE), RMSE, and Relative Root Mean Squared Error (RRMSE). We will choose the model that has the lowest values across all these metrics. In the next section, we will provide the formulas used to calculate each of these metrics:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.6.1}$$

$$RMAE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \tag{3.6.2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.6.3}$$

$$RRMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2}. \tag{3.6.4}$$

3.7 Conclusion

This chapter provided an overview of the methodology that will be deployed in this study. It covered the data source and various machine learning models, explains their unique features, and introduced the concept of stacking ensemble models to leverage their collective predictive ability. The chapter also discussed parameter tuning and variable selection, which will essential to effectively train the models. The accuracy of the forecasts is assessed using specific metrics to ensure the models' reliability and robustness. By combining these methodologies, we aim to conduct this study with the utmost scholarly standards and precision.

Chapter 4

Results and Discussions

4.1 Introduction

This chapter will conduct a detailed analysis of the datasets. It will cover the data sources, variables within the data, and the Exploratory Data Analysis (EDA) that was performed together with the tools used to achieve the whole results. This chapter aims to provide a comprehensive understanding of the data characteristics. It will also discuss the techniques used for variable selection, with a specific focus on random forest. Additionally, the chapter will explain the process of model fitting and highlight the steps taken to ensure the creation of robust and accurate predictive models.

4.2 Dataset description

The data used in this study is obtained from SAURAN which can be accesed at this website https://sauran.ac.za/. The SAURAN have over twenty stations across different provinces in South Africa and other two other neighboring countries. The specific focus of this project is around the USAID Venda

which is situated in Limpopo Province in Vhembe district in Vuwani the stations locations is -23.131 latitude and 30.4239 longitude with the elevation of 628 meters.

The datasets span from 15 November 2023 at 15:58:00 to 15 March 2024 at 22:32:00. The data is minute-averaged with 99941 observations. The variables in the data include GHI (the response variable), Temperature, Relative Humidity, Wind Speed, the wind speed vector magnitude, Wind Direction, Wind Direction Standard Deviation, Wind Speed Maximum, Barometer Pressure, Calculated Azimuth Angle, Calc Tilt Angle, and five derived variables (diff1 to diff60). All these variables will help to predict the response variable, and their ranges can be seen in the Table 4.1.

Table 4.1: Summary of Variables: Ranges.

Variable	**Range**
GHI	0.0064 – 1666.7258
diff1	-973.2486 – 961.2202
diff2	-1031.5691 – 1052.9602
diff15	-1219.3918 – 1191.9289
diff30	-1281.9041 – 1239.3146
diff60	-1270.3455 – 1442.9946
Temp (°C)	14.07 – 41.14
RH (%)	15.0 – 100.0
WS (m/s)	0.0 – 10.5
WVec_Mag (m/s)	0.0 – 10.21
WD (°)	0.0 – 360.0
WD_StdDev (°)	0.0 – 77.37
WS_Max (m/s)	0.0 – 17.84
BP (hPa)	934.1265 – 953.3445
Calc_Azimuth (°)	-180.0 – 180.0
Calc_Tilt (°)	0.04 – 153.1

4.3 Software and packages

Python is the language of choice for analysing data and implementing machine learning models due to its simplicity, versatility, and extensive library ecosystem. The version being used is Python 3.x, which ensures compatibility with a wide range of modern data science and machine learning libraries. The essential libraries for data analysis and machine learning in Python include Pandas, NumPy, Matplotlib, Seaborn, scikit-learn (sklearn), XGBoost, TensorFlow, Keras, Statsmodels, Scipy, LightGBM, joblib, PyTorch, Optuna, and bayes opt.

4.4 Exploratory data analysis

In our data preprocessing, we focused on cleaning the dataset to ensure its accuracy and reliability. One challenge we encountered was the presence of zero values in the GHI variable, which measures the total amount of shortwave radiation received from above by a horizontal surface. Since data is collected throughout 24 hours, zero values are recorded during night time when there is no sunlight. To address this issue, we used a technique called listwise deletion to remove all the zero values based on GHI. This ensures that our study is robust. Imputation techniques were not considered, as simply replacing with a mean can introduce bias, as GHI depends on specific factors. Since GHI rises in the morning and peaks during the day, replacing it with mean or other values would reduce the quality of the study.

Table 4.2: GHI summary Statistics

Min	Q1	Median	Mean	Q3	Max	Std	Skewness	Kurtosis
0.006	103.463	314.610	415.996	692.425	1666.725	359.767	0.653	-0.811

The summary statistics for the GHI data are presented in Table 4.2. These statistics provide a summary understanding of the variable before making predictions. The table includes the minimum, first quartile (Q1), mean, median, third quartile (Q3), maximum, standard deviation, skewness, and kurtosis.

The GHI values in the dataset range from a minimum of 0.006 to a maximum of 1666.725, with an average value of 415.996 over the data period. This wide range of values indicates that the dataset captures various weather conditions, ranging from very low solar radiation during early mornings or overcast days to much higher levels during clear, sunny periods. The positive skewness of the data suggests that most of the GHI values are on the lower end, with fewer high values, which is typical for solar data as low irradiance occurs more often than peak values.

Additionally, the kurtosis value of -0.811 indicates that the distribution is platykurtic, meaning it has a flatter peak and fewer extreme outliers compared to a normal distribution. This suggests that while the GHI values vary, there are fewer extreme high or low values than one might expect, making the dataset more balanced and stable overall. Understanding these aspects of the data is crucial before proceeding with any predictive modelling.

4.4.1 Visualisations

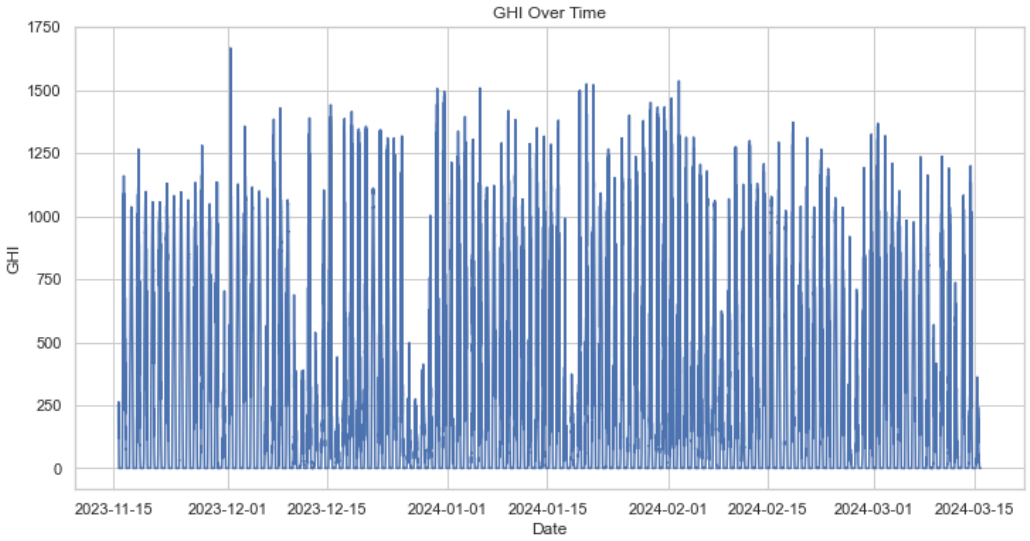


Figure 4.1: GHI for the sampling period 2023 to 2024.

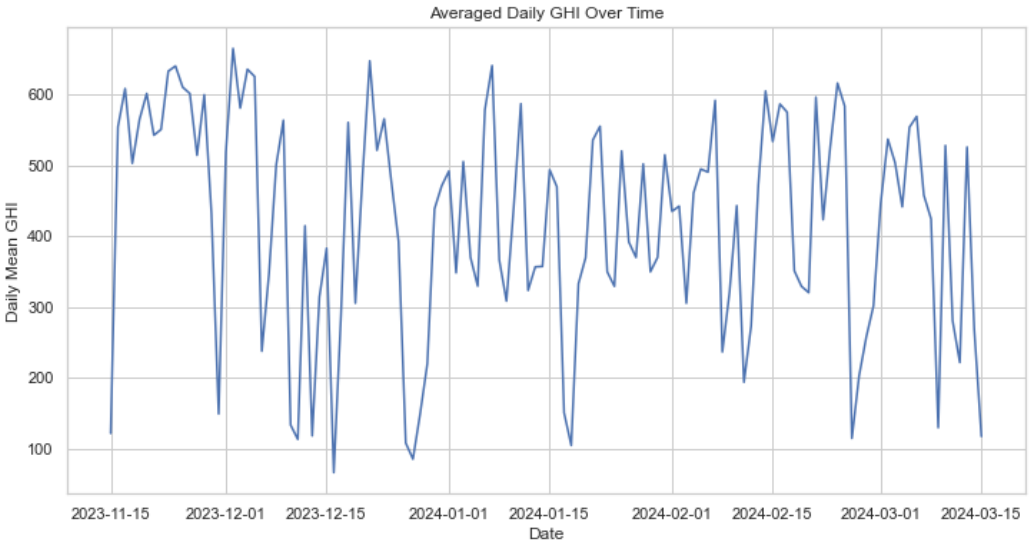


Figure 4.2: Averaged daily GHI.

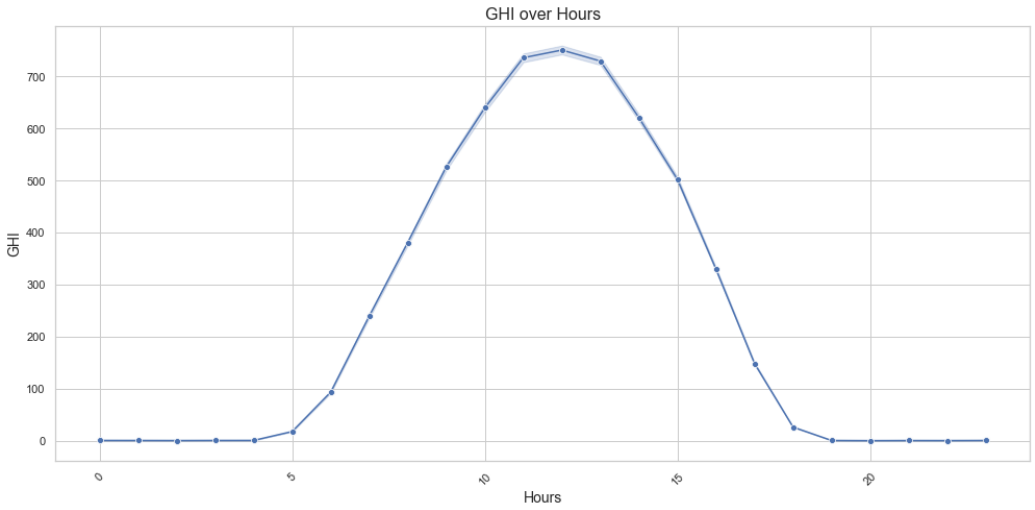


Figure 4.3: GHI over hours.

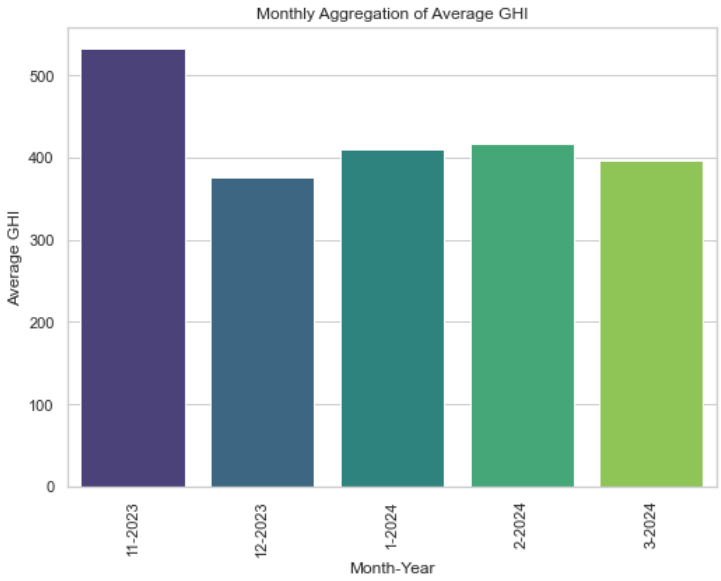


Figure 4.4: GHI over hours.

Figures 4.1, 4.2, 4.4, and 4.3 represent the GHI at different times, including the overall time, average daily aggregation, monthly aggregation, and by

the hour. These plots allow us to observe the behavior of GHI at different times. Figure 4.1 shows how GHI behaves minute by minute throughout the data, while Figure 4.2 displays the average daily GHI behavior. Figure 4.4 summarises the GHI over different months covered by the data, revealing the highest GHI in the eleventh month of 2023. Lastly, Figure 4.3 illustrates the behavior of GHI over hours, confirming that GHI is lowest during late hours and peaks during mid-hours of the data. Together, these plots offer a comprehensive understanding of GHI trends across different time frames

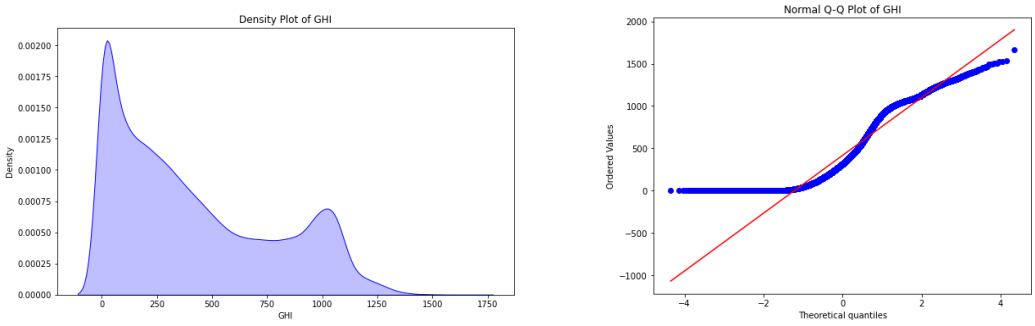


Figure 4.5: Density and QQ plot.

Figure 4.5 displays both the density and Q-Q plots, which confirm the earlier EDA findings. The density plot indicates a right-skewed distribution, with most GHI values concentrated between 0 and 200, and fewer high values above 1000. The Q-Q plot further illustrates that the GHI data deviates from normality, showing significant skewness and potential outliers, which reinforces that the data does not follow a normal distribution.

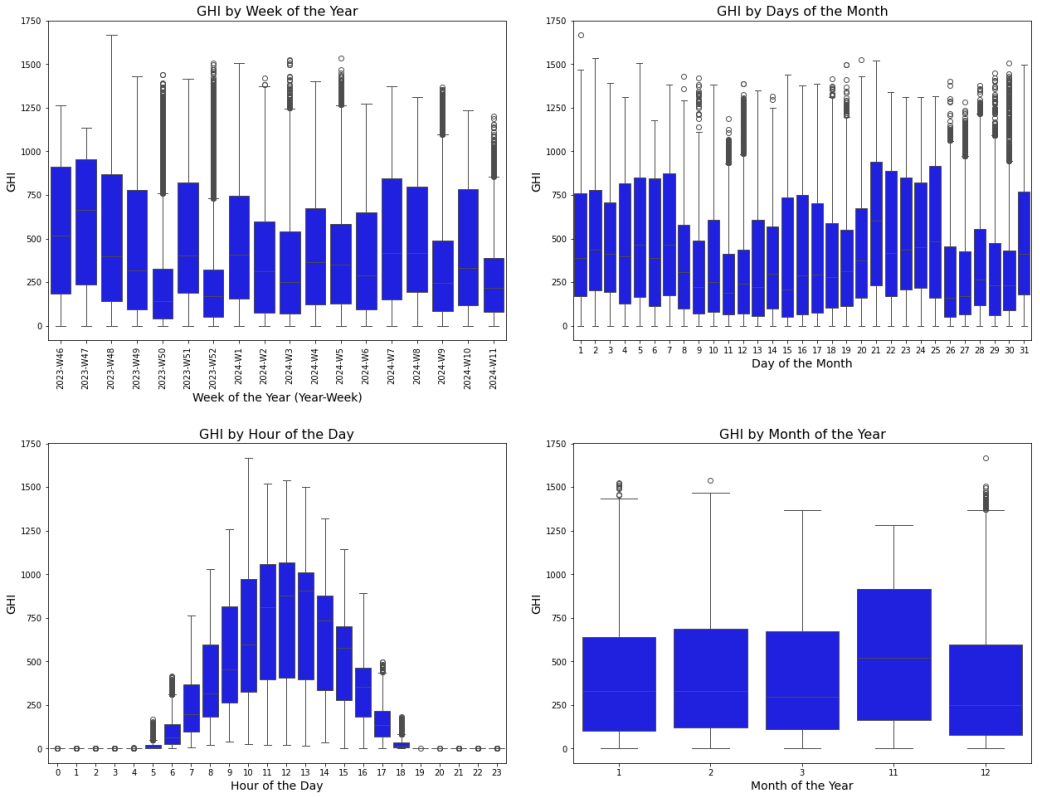


Figure 4.6: Distribution of GHI across the week, day, hour and month in the dataset.

In Figure 4.6, a box plot illustrates the distribution of GHI across different time frames. The "GHI by Week of the Year" plot indicates that some weeks have high median GHI values, suggesting consistently sunny conditions, while other weeks display lower medians, indicating more overcast conditions. The presence of numerous outliers, particularly in certain weeks, suggests occasional spikes in GHI, likely due to clear days occurring within otherwise variable weeks. The "GHI by Days of the Month" plot highlights the variation of GHI across individual days within a month. While the median GHI remains relatively stable, some days exhibit higher peaks or more outliers, indicat-

ing occasional increases in solar irradiance resulting from day-to-day weather fluctuations. Finally, the "GHI by Month of the Year" plot reveals seasonal trends, with some months demonstrating higher median GHI, reflecting sunnier seasons, while others show lower medians, indicative of cloudier periods. The outliers observed in certain months suggest that there are occasional days of high irradiance even during typically low GHI months.

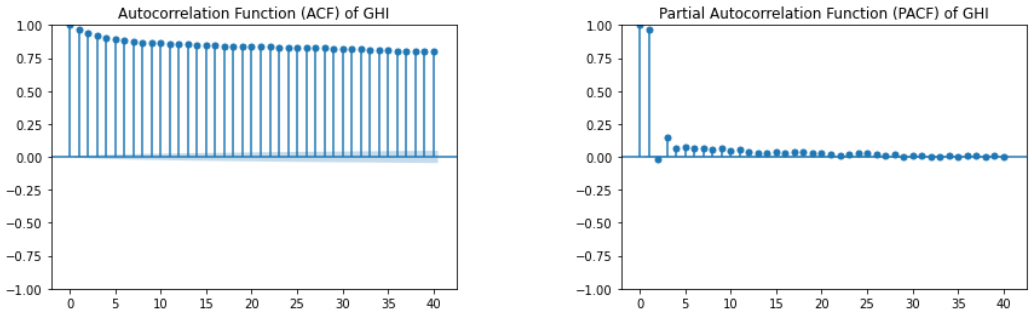


Figure 4.7: ACF and PACF plots.

The Figure 4.7 below displays the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) plot for GHI. The ACF plot shows a slow, gradual decay in correlation, indicating that the time series is non-stationary and has long-term dependencies or trends. The strong correlations at multiple lags suggest that GHI values over time are highly persistent. On the other hand, the PACF plot shows a sharp cutoff after lag 1, suggesting that after accounting for the first lag, the correlations with subsequent lags are negligible.

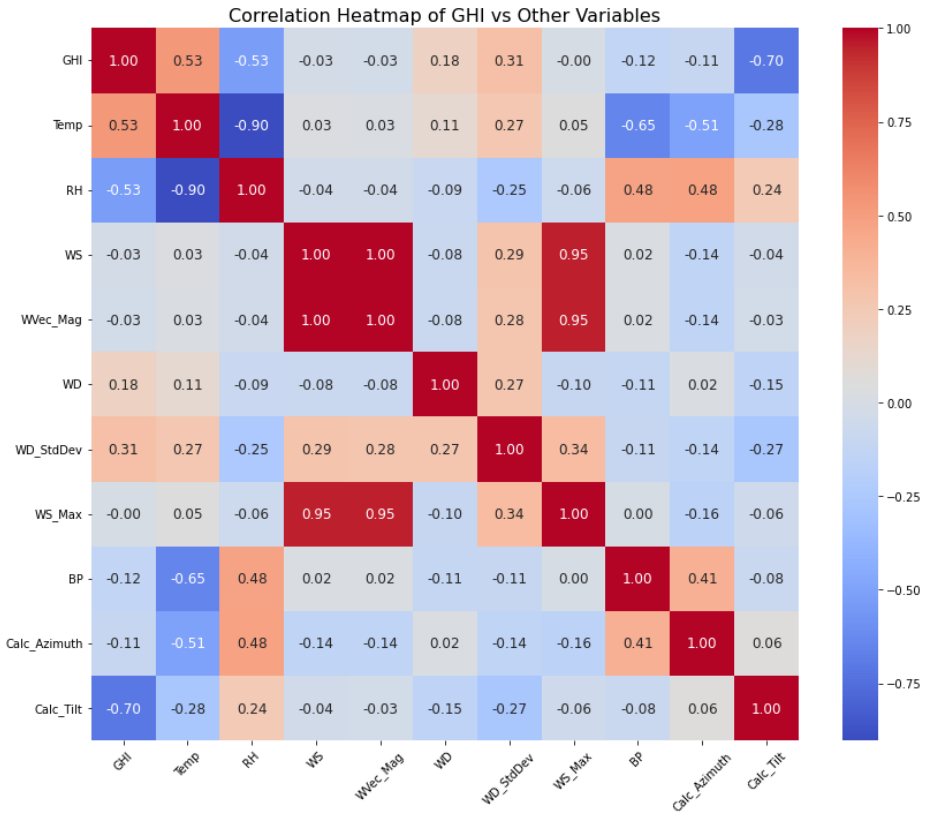


Figure 4.8: Correlation of the dataset.

In exploring the dataset, it is important to check how the dependent variable is correlated with other variables. After computing the correlation heatmap, Figure 4.8 shows that GHI is not correlated with any other variable. This insight prompts us to include all other variables in our analysis going further.

4.5 Variable selection

Figure 4.9 shows the feature importance that were selected. To achieve this, a random forest model was trained, and the feature importances were used to select the best set of features according to Recursive Feature Elimination

with Cross-Validation (RFE-CV). The metric that was optimised was the negative mean squared error, and using 5-fold cross-validation indicated that $WVEC\_mag$, WS_Max, and WS do not significantly contribute to predicting GHI. Hence, these features will not be used in any models going forward.

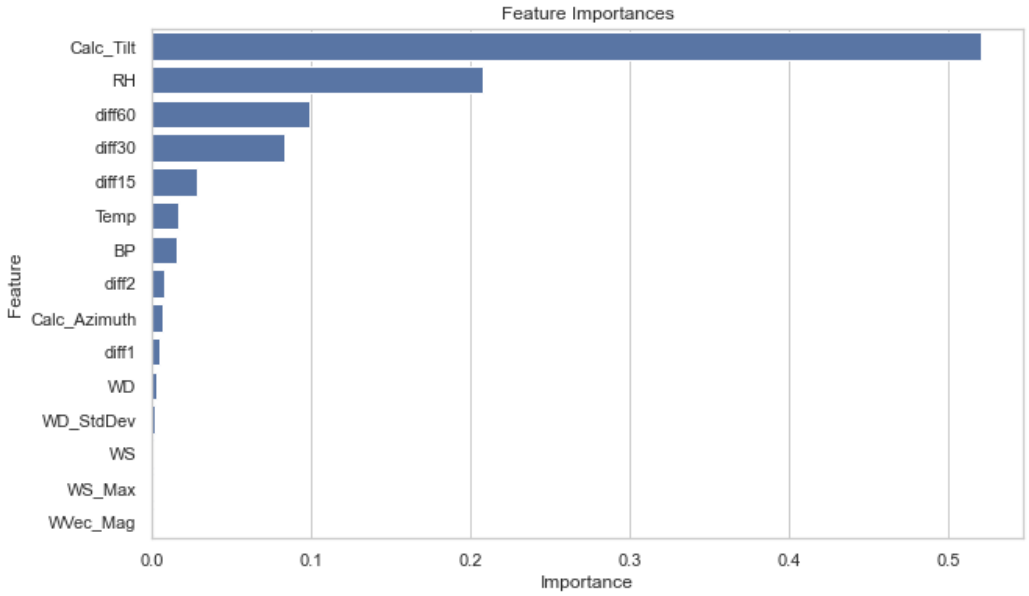
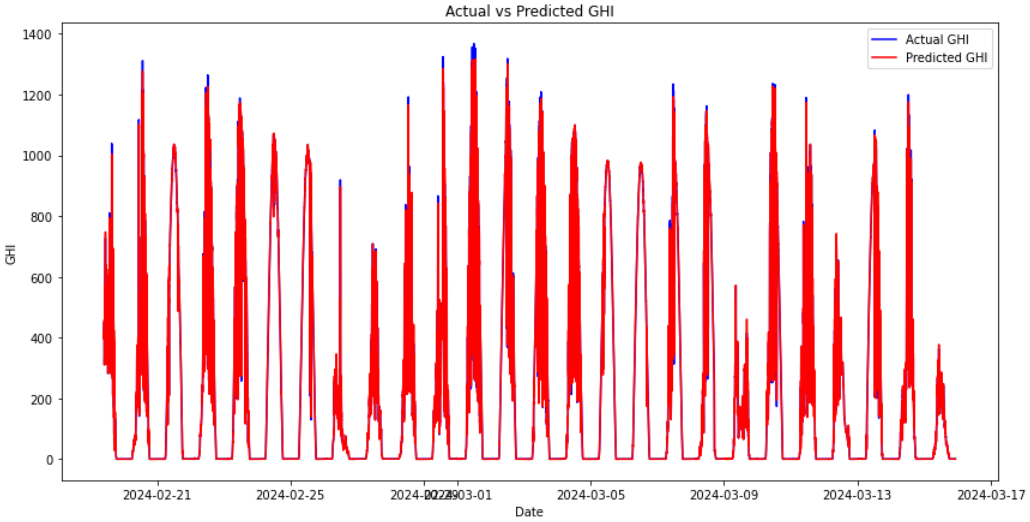


Figure 4.9: Feature selections.

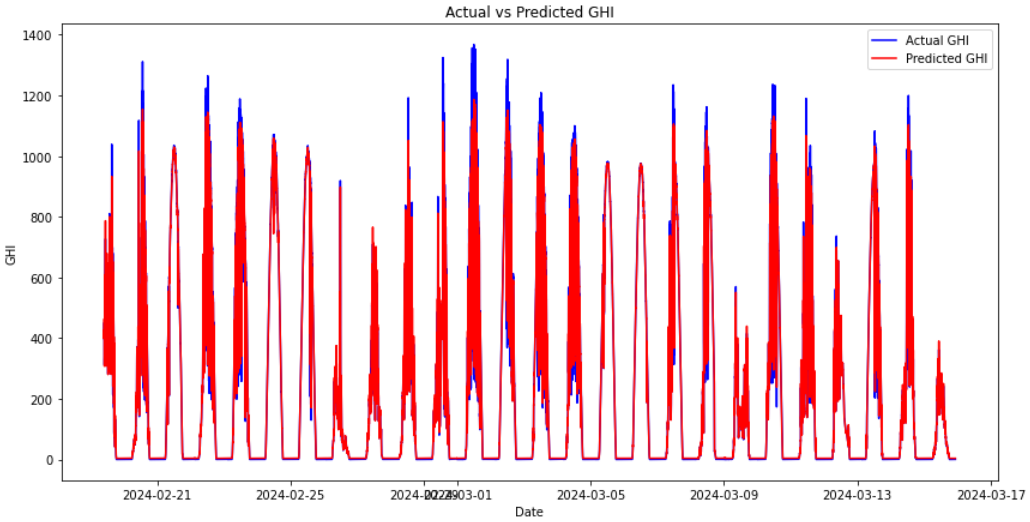
4.6 Machine learning models

After splitting the data into a 70-30 ratio, we began training various machine learning models. The data was normalised and Bayesian optimisation was implemented to help train the models with optimal hyperparameters. The results of the machine learning models' performance on the test set are presented in Figure 4.10. These plots depict the performance of the machine learning model on the test set.

RNN



RF



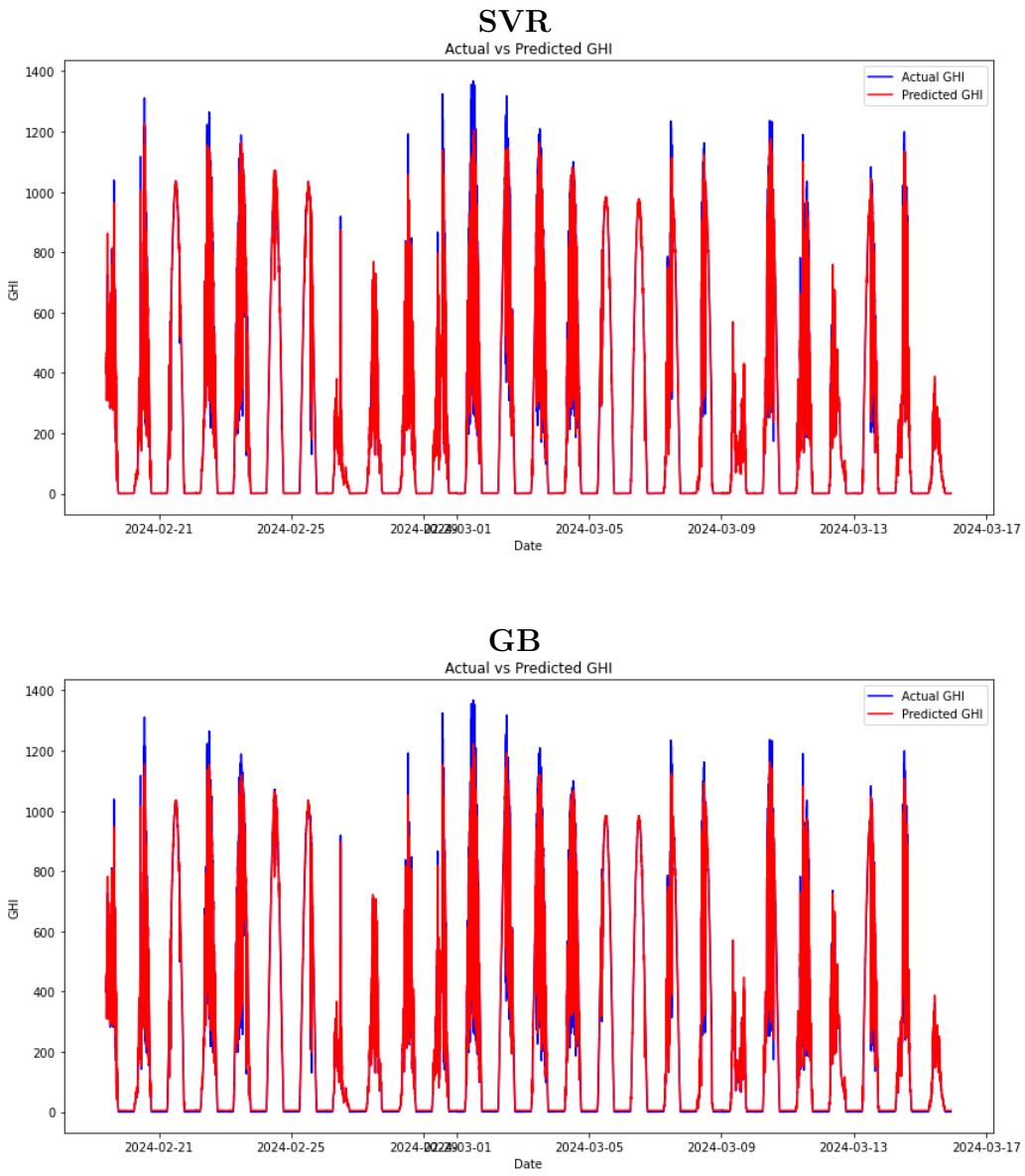


Figure 4.10: Machine learning model performance on test set.

4.6.1 Selected parameters

For the RNN model, the selected parameters from Bayesian optimisation are as follows: 32 units, a dropout rate of 0.2, a learning rate of 0.01, and 10 epochs.

For the RF model, the selected parameters are: max depth of 50, max features of 0.6897487361224869, min samples leaf of 10, min samples split of 10, and n estimators of 200.

For the GB model, the selected parameters are: learning rate of 0.01, max depth of 4, min samples leaf of 6, min samples split of 13, n estimators of 474, and a subsample of 0.8670322417190656. Finally, for the SVR model, the selected parameters are: Best C of 100.0 and Best epsilon of 0.01.

4.6.2 Model comparison

The table 4.3 compares four machine learning models: RNN, SVR, RF, and GB. It evaluates their performance using four key metrics: MAE, RMAE, RMSE, and RRMSE. These metrics help assess the predictive accuracy of each model, with lower values generally indicating better performance in terms of prediction error. We calculated these metrics on the test set after training the models.

Table 4.3: Machine learning model comparison.

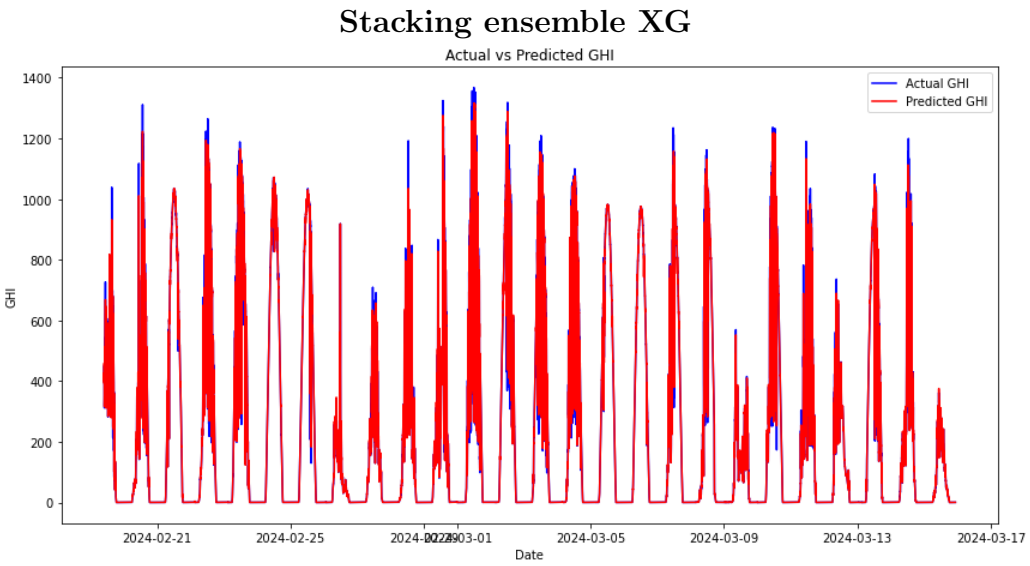
	MAE	RMAE	RMSE	RRMSE
RNN	22.1037	0.4290	25.6224	0.4973
SVR	26.4253	0.0668	81.9700	0.2073
RF	28.9002	0.0730	79.6583	0.2014
GB	29.6836	0.0750	78.0158	0.1973

Among the models, the RNN demonstrates superior performance, achieving the lowest MAE of 22.1037 and RMSE of 25.6224. These low error values indicate that the RNN is the most accurate model in terms of both absolute and squared prediction errors. Additionally, the RNN also exhibits the lowest relative errors, with an RMAE of 0.4290. These relative error metrics further highlight that the RNN model not only effectively minimises errors in absolute terms but also maintains consistency and reliability across different scales of prediction.

In comparison, the SVR, RF, and GB models have higher error rates. The SVR model has a relatively low RMAE of 0.0668 but a much higher RMSE of 81.9700, indicating significant variability in its predictions and a higher penalty for larger errors. The RF and GB models have slightly better RMSE values (79.6583 and 78.0158, respectively) than SVR but still significantly lag behind the RNN model. The RF model's MAE is 28.9002 and the GB model's MAE is 29.6836. Their RMAE and RRMSE values (around 0.0730 and 0.0750 for RMAE, and 0.2014 and 0.1973 for RRMSE, respectively) indicate that while these models perform reasonably well, they do not match the predictive accuracy or consistency demonstrated by the RNN model.

4.7 Stacking ensemble models

The study proceeded to train stacking ensemble models. This process began by using the same hyperparameters that were used in the individual machine learning models, in order to provide a fair comparison of the models. In Chapter 3, there was a discussion of meta models, which are essential for use in stacking learning models. Out of the four models mentioned, a mini experiment was conducted to select only two: RF and XGBOOST, as they are the top two meta models. These models will be further explored in the stacking ensemble and even in the DNS model. The Figures 4.11 and 4.12 and shows the performance of the stacking ensemble and DNS models on the test set.



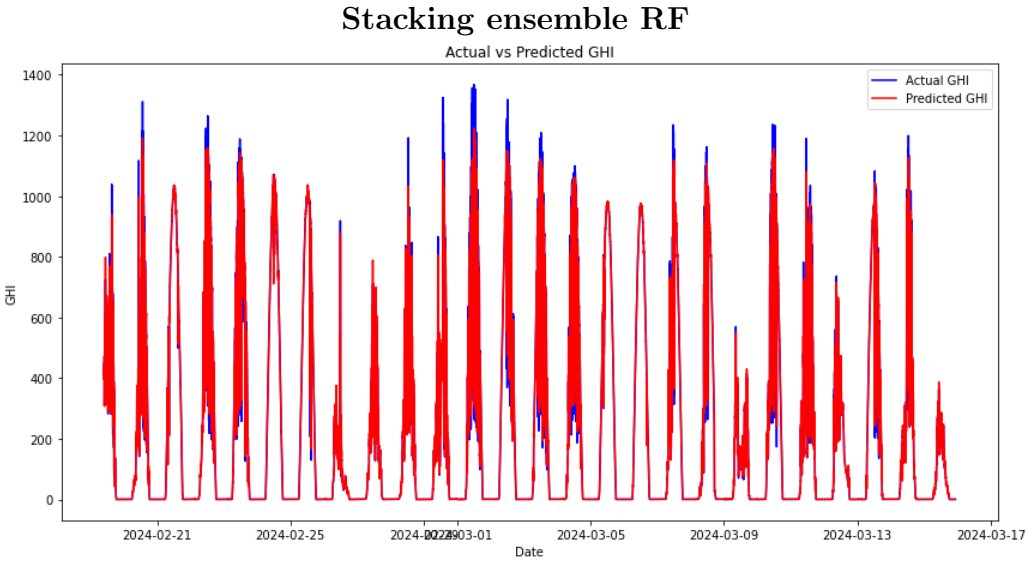
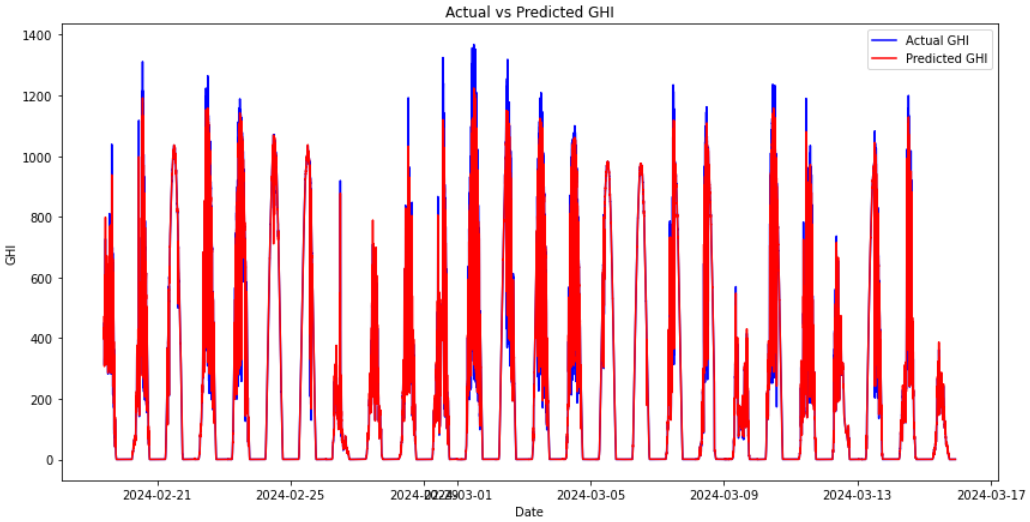


Figure 4.11: Stacking ensemble performance on the test set.



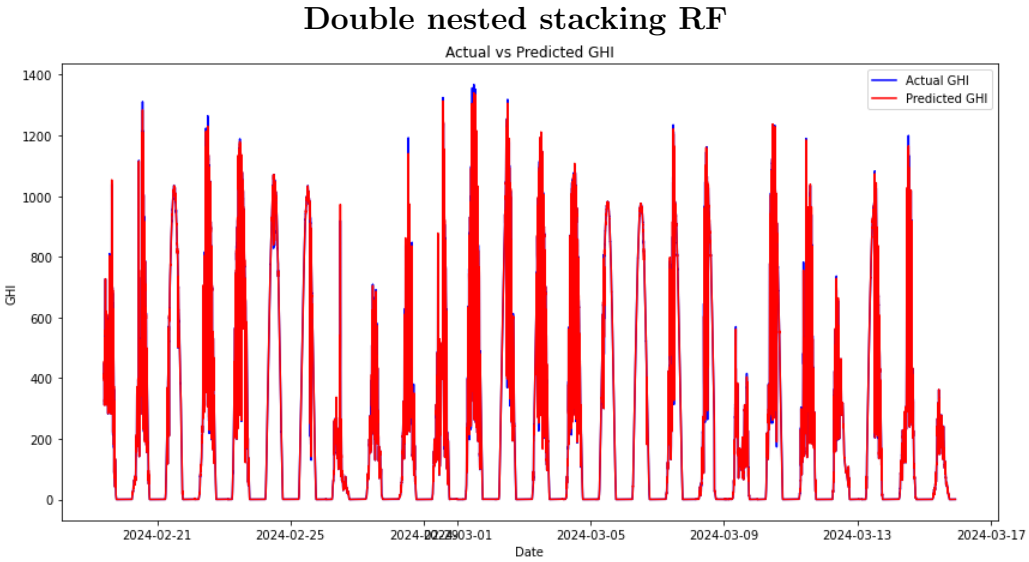


Figure 4.12: DNS models performance on the test set.

In Table 4.4, the error metrics for the stacking ensemble on the test set are displayed. The stacking ensemble with RF as the meta-model (SE RF) exhibits relatively higher MAE and RMSE values compared to other models. This suggests that the SE RF model has larger prediction errors on average. The RMAE and RRMSE values also indicate a relatively high error proportion in relation to the magnitude of the data. This might imply that the RF model is not as effective as other models in capturing the complexity or patterns in the data.

The stacking ensemble with XGBoost as the meta-model (SE XG) performed better than SE RF, showing significantly lower MAE, RMAE, RMSE, and RRMSE values. This suggests higher overall accuracy and lower error rates. The XGBoost meta-model may have utilised its gradient boosting mechanism

to better capture non-linear relationships and interactions among features, resulting in a more accurate model compared to the RF meta-model.

4.7.1 Model comparison

Table 4.4: Stacking models comparison.

	MAE	RMAE	RMSE	RRMSE
SE XG	7.2808	0.0184	19.9125	0.0504
SE RF	11.7188	0.0296	30.6534	0.0775
DNS XG	5.0633	0.0128	16.0328	0.0406
DNS RF	2.9217	0.0074	7.5926	0.0192
DNS XG RF	4.5771	0.0116	11.8832	0.0301
DNS RF XG	1.5369	0.0038	2.9341	0.0074

The DNS model with RF as the meta-model (DNS RF) shows significant improvement over the SE RF model. The MAE and RMSE are much lower, indicating a substantial reduction in prediction errors. Additionally, the RMAE and RRMSE values are also very low, suggesting high accuracy. This improvement may be attributed to the double nested structure, which enables the model to better integrate predictions from different base learners, resulting in more robust predictions.

The DNS XG model performs worse than the DNS RF model but better than the SE RF and SE XG models. Although XGBoost is generally a strong learner, the double nested stacking framework with XGBoost does not seem to outperform the RF meta-model in this particular setup. This could suggest that while XGBoost is powerful, it may not always yield the best results in ensemble frameworks involving multiple layers of modeling.

4.7.2 Meta model order impact

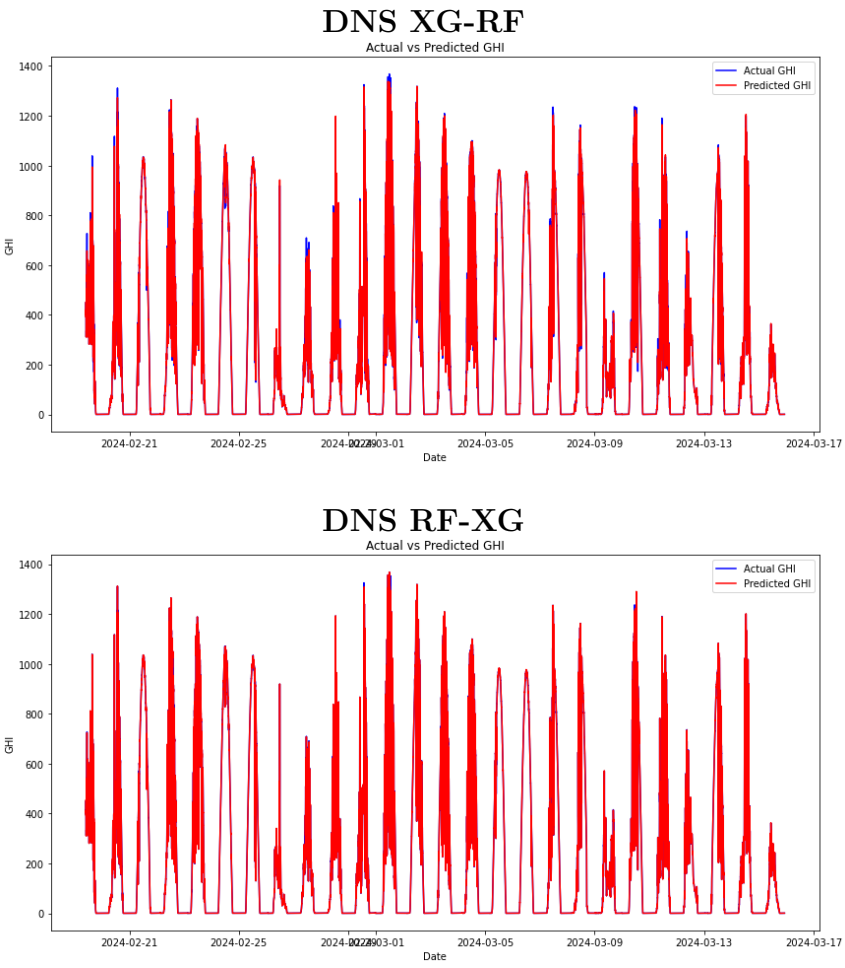


Figure 4.13: DNS XG-RF and DNS RF-XG models performance on the test set.

The order in which meta-models are arranged in DNS has a significant impact on model performance. This is because each meta-model processes the predictions from base models differently. The output of one meta-model serves as the input for the next, thereby affecting the overall results. We conducted

experiments to assess the effects of varying the sequence of meta-models in DNS, as displayed in Table 4.4.

The results of the experiments showed that the model using RF as the first meta-model, followed by XGBoost (DNS RF XG), performed better than the reverse order (DNS XG RF). Specifically, the MAE for DNS RF XG was 66.42% lower compared to DNS XG RF, and the RMSE was 75.31% lower. This indicates the significant impact of meta-model sequencing on the effectiveness of double nested stacking models.

4.8 Conclusion

In Chapter 4, we conducted a comprehensive analysis of the data, including EDA, variable selection, parameter tuning, and comparison of different machine learning models such as RNN, SVR, RF, and GB. We evaluated their performance using key metrics like MAE, RMAE, RMSE, and RRMSE. Our analysis revealed that the RNN model outperformed the others, demonstrating its ability to capture complex patterns in the data with the lowest prediction errors.

Additionally, we introduced Stacking ensemble models, including a DNS framework, which significantly improved prediction accuracy, especially when using the RF and XGBoost meta-models in the correct sequence. These results underscore the importance of meta-model selection and order in enhancing performance within ensemble learning. These findings lay a strong groundwork for further exploration of ensemble techniques in the upcom-

ing chapters, with the goal of refining and optimising model accuracy for forecasting.

Chapter 5

Conclusion

5.1 Introduction

In this chapter, we will provide a comprehensive summary of the research conducted and review the results discussed in the previous chapter. We will also present well-considered recommendations based on the findings of the study. Additionally, this section will outline the limitations encountered during the research process, providing a transparent account of any constraints that might affect the interpretation or generalisation of the results. Finally, we will explore potential areas for future research, suggesting directions that subsequent studies could take to build upon the foundations laid by this work and contribute to the broader field of knowledge.

5.2 Research findings

Forecasting GHI is crucial for measuring the potential solar energy production on a horizontal surface. This helps various sectors identify the best time to collect solar energy and plan for additional sourcing. The research study

focuses on forecasting GHI using minute-averaged data from the SAURAN database for USAID Venda, spanning from November 15, 2023, to March 15, 2024.

In the discussion in Chapter 4, we talked about modelling minute-averaged GHI forecasting using RNN, SVR, RF, GB, and two stacking ensemble methods. We also discussed the importance of variable selection and how we implemented RF, excluding three variables (wind speed, maximum wind speed, and wvec mag) from the GHI forecasting models.

According to the findings in Chapter 4, after all the data preprocessing, including the implementation of hyperparameter tuning for each model using Bayesian optimisation, the results show that the best-performing machine learning model is the RNN because it has the lowest values for both MAE and RMSE. The second-best performing model is SVR as it has the second-lowest MAE after RNN. Although its RMSE is higher, it still outperforms both RF and GB models overall.

When evaluating the performance of the stacking ensemble models, it was important to use the same parameters that were selected during the training of the individual models. Four meta models were used: Ridge regression, XGBOOST, RF, and Elastic Net. Out of these, the two best meta models, RF and XGboost, were further explored to understand the impact of the meta model in the stacking ensemble setting. The results showed that both stacking ensemble models with these meta models outperformed all other machine

learning models. The best stacking model was the one with XGboost as the meta model, showing a 67.060% and 22.2848% increase in accuracy compared to the MAE and RMSE of the best performing machine learning model RNN.

Implementing the same meta-models in another stacking technique, which is DNS, revealed that accuracy can be further improved. The results showed that both DNS XG and DNS RF outperform the single stacking model. The best performing results show a decrease in MAE and RMSE by 59.87% and 61.87% respectively when compared to the best performing SE XG. Further experimenting with the order and mixing of the meta-models can again improve the forecasting accuracy of the DNS model.

When the RF was the first-level meta-model followed by XGBOOST, the DNS model outperformed every model in the study, including the reverse order of the meta-models. The results show a 93.05% decrease in MAE and an 88.54% decrease in RMSE compared to the best performing machine learning model, and a 78.89% decrease in MAE and an 85.27% decrease in RMSE compared to the best single stacking model. Additionally, the DNS model outperformed the DNS RF, where RF was used at both levels, showing a decrease in MAE and RMSE by 47.39% and 61.35% respectively.

In conclusion, machine learning models can provide fairly accurate forecasts for GHI. However, implementing advanced techniques such as stacking ensembles with multiple levels and experimenting with different meta models can significantly improve performance. The results demonstrate the potential

of stacking in predicting GHI accurately, which has been validated in other sectors. This study highlighted its promising application in the renewable energy sector.

5.3 Recommendations

The results of this research show how important it is to accurately forecast GHI in the renewable energy industry. Having precise GHI forecasts is crucial for optimising the use of solar energy in power grids, which helps in generating cleaner energy and reducing the dependence on harmful power production methods. This is especially important in South Africa, where there is a growing need to implement strategies that reduce carbon emissions and address climate change.

Based on the findings of this study, we recommend using machine learning models in stacking ensemble techniques for short-term GHI forecasting. Our results indicate that increasing the stacking levels with diverse combinations of meta-models improves predictive accuracy.

5.4 Limitations of the study

The study had promising results, but faced several limitations during implementation. One major limitation was the dataset, which only covered part of the summer in South Africa and did not include data from all four seasons. Including data from all seasons could have provided more insights into the long-term seasonality of GHI. Additionally, the study only used data

from one out of nineteen radiometric stations owned by the SAURAN. Using data from multiple stations across different locations could have improved the model's applicability and reliability.

Another limitation was the complexity of the models. Although stacking ensemble techniques were used, the study only explored a limited number of meta-model combinations and levels. Investigating more complex ensemble architectures or hybrid models could potentially improve forecasting accuracy, but this was beyond the scope of the current research. Additionally, limited computational resources for training and tuning these advanced models constrained the extent of hyperparameter tuning and experimentation. Access to more powerful resources could have allowed for better optimisation and potentially improved results.

5.5 Future research

In future studies, it is important to include data from all seasons in South Africa's diverse climate. This will provide a more comprehensive understanding of the seasonal variations in GHI. Additionally, future research should focus on using more advanced machine learning models, such as hybrid models, and exploring a wider range of meta-models and their combinations in stacking ensemble methods. These efforts could improve the accuracy of GHI forecasts and provide more reliable predictions for renewable energy applications.

5.6 Conclusion

In this chapter, we summarised the findings of the study and outlined future research directions. The insights gained have the potential to significantly impact decision-making in the renewable energy sector by providing more reliable forecasts for GHI. This can contribute to improved planning and more efficient resource allocation for solar energy projects. Ultimately, the findings underscore the importance of using advanced machine learning techniques to integrate renewable energy into power grids and support the transition to cleaner energy solutions.

References

Aghelpour, P., Mohammadi, B. and Biazar, S. M. (2019), 'Long-term monthly average temperature forecasting in some climate types of Iran, using the models SARIMA, SVR, and SVR-FA', *Theoretical and Applied Climatology* **138**(3), 1471–1480.

Alcántara, A., Galván, I. M. and Aler, R. (2023), 'Deep neural networks for the quantile estimation of regional renewable energy production', *Applied Intelligence* **53**(7), 8318–8353.

Aliberti, A., Fucini, D., Bottaccioli, L., Macii, E., Acquaviva, A. and Patti, E. (2021), 'Comparative analysis of neural networks techniques to forecast global horizontal irradiance', *IEEE Access* **9**, 122829–122846.

Armaroli, N. and Balzani, V. (2011), 'The legacy of fossil fuels', *Chemistry–An Asian Journal* **6**(3), 768–784.

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

Breiman, L. (2001), 'Random forests', *Machine Learning* **45**, 5–32.

Breiman, L. and Ihaka, R. (1984), *Nonlinear discriminant analysis via scaling and ACE*, Department of Statistics, University of California Davis One Shields Avenue.

Dang, T. T., Hoang, K. N., Thanh, L. B., Thuy, T. N. T. and Quoc, C. N. (2023), 'Constructing and understanding customer spending prediction models', *SN Computer Science* **4**(6), 852.

Demirbas, A. (2000), 'Recent advances in biomass conversion technologies', *Energy Education Science and Technology* **6**, 19–41.

Divina, F., Gilson, A., Goméz-Vela, F., García Torres, M. and Torres, J. F. (2018), 'Stacking ensemble learning for short-term electricity consumption forecasting', *Energies* **11**(4), 949.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A. and Vapnik, V. (1996), 'Support vector regression machines', *Advances in neural Information Processing Systems* **9**.

Fox, E. W., Hill, R. A., Leibowitz, S. G., Olsen, A. R., Thornbrugh, D. J. and Weber, M. H. (2017), 'Assessing the accuracy and stability of variable selection methods for random forest modeling in ecology', *Environmental Monitoring and Assessment* **189**, 1–20.

Friedman, J. H. (2001), 'Greedy function approximation: a gradient boosting machine', *Annals of Statistics* pp. 1189–1232.

Garcia-Pedrero, A. and Gomez-Gil, P. (2010), Time series forecasting using recurrent neural networks and wavelet reconstructed signals, *in* '2010 20th International Conference on Electronics Communications and Computers (CONIELECOMP)', IEEE, pp. 169–173.

Gardner, J. R., Kusner, M. J., Xu, Z. E., Weinberger, K. Q. and Cunningham, J. P. (2014), Bayesian optimization with inequality constraints., *in* 'International Conference on Machine Learning', Vol. 2014, pp. 937–945.

Gbémou, S., Eynard, J., Thil, S., Guillot, E. and Grieu, S. (2021), 'A comparative study of machine learning-based methods for global horizontal irradiance forecasting', *Energies* **14**(11), 3192.

Genuer, R., Poggi, J.-M. and Tuleau-Malot, C. (2010), 'Variable selection using random forests', *Pattern Recognition Letters* **31**(14), 2225–2236.

Guangul, F. M. and Chala, G. T. (2019), Solar energy as renewable energy source: Swot analysis, *in* '2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)', IEEE, pp. 1–5.

Guo, X., Gao, Y., Zheng, D., Ning, Y. and Zhao, Q. (2020), 'Study on short-term photovoltaic power prediction model based on the stacking ensemble learning', *Energy Reports* **6**, 1424–1431.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Tibshirani, R. and Friedman, J. (2009), 'Unsupervised learning', *The elements of statistical learning* pp. 485–585.

Joy, T. T., Rana, S., Gupta, S. and Venkatesh, S. (2016), Hyperparameter tuning for big data using bayesian optimisation, *in* '2016 23rd International Conference on Pattern Recognition (ICPR)', IEEE, pp. 2574–2579.

Khandelwal, Y. (2021), 'Ensemble stacking for machine learning and deep learning', *Anal Vidhya* **9**.

Krishnan, N., Kumar, K. R. et al. (2024), 'Solar radiation forecasting using gradient boosting based ensemble learning model for various climatic zones', *Sustainable Energy, Grids and Networks* p. 101312.

Kumar Barik, A., Malakar, S., Goswami, S., Ganguli, B., Sen Roy, S. and Chakrabarti, A. (2021), Analysis of GHI forecasting using seasonal arima, *in* 'Data Management, Analytics and Innovation: Proceedings of ICDMAI 2020, Volume 2', Springer, pp. 55–69.

Laan, M. J. (2007), 'Super learner.', *Statistical applications in genetics and molecular biology* **6**.

Lahouar, A. and Slama, J. B. H. (2015), Random forests model for one day ahead load forecasting, *in* 'IREC2015 the Sixth International Renewable Energy Congress', IEEE, pp. 1–6.

Mandic, D. P. and Chambers, J. (2001), *Recurrent neural networks for prediction: learning algorithms, architectures and stability*, John Wiley & Sons, Inc.

Nemalili, R. C., Jhamba, L., Kiprono Kirui, J. and Sigauke, C. (2023), 'Nowcasting hourly-averaged tilt angles of acceptance for solar collector applications using machine learning models', *Energies* **16**(2), 927.

Nziyumva, E., Nsengimna, M., Niyogisubizo, J., Murwanashyaka, E., Nisingizwe, E. and Kwitonda, A. (n.d.), 'A novel two layer stacking ensemble for improving solar irradiance forecasting'.

Petneházi, G. (2019), 'Recurrent neural networks for time series forecasting', *arXiv preprint arXiv:1901.00069* .

Philibert, C. et al. (2011), 'Solar energy perspectives', *(No Title)* .

Rajaprasad, S. and Mukkamala, R. (2023), 'A hybrid deep learning framework for modeling the short term global horizontal irradiance prediction of a solar power plant in india', *Polityka Energetyczna-Energy Policy Journal* pp. 101–116.

Sayigh, A. (2012), *Solar energy engineering*, Elsevier.

Vapnik, V. N., Vapnik, V. et al. (1998), 'Statistical learning theory'.

Wolpert, D. H. (1992), 'Stacked generalization', *Neural Networks* **5**(2), 241–259.

Yamani, A. Z. and Alyami, S. N. (2021), Investigating hourly global horizontal irradiance forecasting using long short-term memory, *in* '2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)', IEEE, pp. 1–6.

Zhou, B., Chen, X., Li, G., Gu, P., Huang, J. and Yang, B. (2023), 'Xgboost–sfs and double nested stacking ensemble model for photovoltaic power forecasting under variable weather conditions', *Sustainability* **15**(17), 13146.