EXPLAINABLE AI BASED SOLAR IRRADIANCE PREDICTION FROM WEATHER DATA USING MACHINE LEARNING AND DEEP LEARNING WITH HYBRIDIZED CLUSTERING

SAIDUR RAHAMAN SAGOR (SN. 201914025)
CHOWDHURY FARJANA TUR SANTONA (SN. 202014010)
MUHAMMAD SAMEE SEVAS (SN. 202014038)

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MILITARY INSTITUTE OF SCIENCE AND TECHNOLOGY
DHAKA, BANGLADESH

MARCH 2024

EXPLAINABLE AI BASED SOLAR IRRADIANCE PREDICTION FROM WEATHER DATA USING MACHINE LEARNING AND DEEP LEARNING WITH HYBRIDIZED CLUSTERING

B.Sc. ENGINEERING THESIS

By

SAIDUR RAHAMAN SAGOR (SN. 201914025)
CHOWDHURY FARJANA TUR SANTONA (SN. 202014010)
MUHAMMAD SAMEE SEVAS (SN. 202014038)

Approved as to style and content by the Examiners in March 2024:

____

Nusrat Sharmin, PhD — Chairman(Supervisor)
Assistant Professor of Computer Science and Engineering
MIST, Dhaka, Bangladesh

____

Brigadier General Mohammad Sajjad Hossain — Head of The Department
Head of the Computer Science and Engineering
MIST, Dhaka, Bangladesh

Department of Computer Science and Engineering MIST, Dhaka

CANDIDATE'S DECLARATION

EXPLAINABLE AI BASED SOLAR IRRADIANCE PREDICTION FROM WEATHER DATA USING MACHINE LEARNING AND DEEP LEARNING WITH HYBRIDIZED CLUSTERING

DECLARATION

We hereby declare that the study reported in this thesis entitled as above is our own original work and has not been submitted before anywhere for any degree or other purposes. Further we certify that the intellectual content of this thesis is the product of our own work and that all the assistance received in preparing this thesis and sources have been acknowledged and cited in the reference Section.

SAIDUR RAHAMAN SAGOR
Student No. 201914025

CHOWDHURY FARJANA TUR SANTONA
Student No. 202014010

MUHAMMAD SAMEE SEVAS
Student No. 202014038

Department of Computer Science and Engineering MIST, Dhaka

ACKNOWLEDGEMENTS

We are thankful to Almighty Allah for his blessings for the successful completion of our thesis. We like to acknowledge and give our warmest thanks and deep respect to our supervisor, Assistant Professor Nusrat Sharmin, PhD of the MIST for her constant supervision, affectionate guidance and great encouragement and motivation. Her keen interest on the topic and valuable advices throughout the study was of great help in completing thesis. We are especially grateful to the Department of Computer Science and Engineering (CSE) of Military Institute of Science and Technology (MIST) for providing all their support during the thesis work.

Finally, we would like to thank our families and our course mates for their appreciable assistance, patience and suggestions during the course of our thesis.

ABSTRACT

EXPLAINABLE AI BASED SOLAR IRRADIANCE PREDICTION FROM WEATHER DATA USING MACHINE LEARNING AND DEEP LEARNING WITH HYBRIDIZED CLUSTERING

The solar revolution in Bangladesh stands as a symbol of hope and self-reliance, illuminating communities and steering the nation towards a more sustainable future. Highlighting the crucial importance of solar irradiance forecasting in attaining sustainable energy objectives, this study utilizes advanced machine learning techniques to enlighten solar irradiance prediction in Bangladesh. Ensemble machine learning has become a vital tool that significantly enhances the precision and reliability of solar radiation predictions by amalgamating diverse model outputs, providing a more robust and accurate forecasting framework. Previous research efforts have overlooked the exploration of this diverse range in solar radiation prediction using ensemble machine learning. This study addresses this gap by conducting a detailed experiment on ensemble artificial intelligence techniques. Furthermore, the research introduces Explainable AI (XAI) with ensemble machine learning, shedding light on factors influencing predictions and providing valuable insights for decision-makers in the solar energy sector. Notably, the study pioneers an XAI-based analysis specific to Bangladesh, marking a significant stride in solar radiation prediction. Additionally, a novel hybridized approach incorporating various clustering techniques and the LightGBM algorithm is introduced, offering an efficient framework for solar radiation prediction. As a result, this study contributes to our understanding and optimization of solar irradiance prediction by offering a comprehensive method that integrates XAI, ensemble approaches, and machine learning. We have developed an autoML tool based on XAI and Ensemble as a further contribution. We have validated our result with the low-code PyCaret machine learning package to see that, among all the methods, lightGBM has shown promising results in terms of solar irradiance prediction.

সারসংক্ষেপ

EXPLAINABLE AI BASED SOLAR IRRADIANCE PREDICTION FROM WEATHER DATA USING MACHINE LEARNING AND DEEP LEARNING WITH HYBRIDIZED CLUSTERING

বাংলাদেশে সৌর বিপ্লব আশা ও স্বনির্ভরতার প্রতীক হিসেবে দাঁড়িয়েছে, সম্প্রদায়কে আলোকিত করছে এবং জাতিকে আরও টেকসই ভবিষ্যতের দিকে নিয়ে যাচ্ছে। টেকসই শক্তির লক্ষ্য অর্জনে সৌর বিকিরণ পূর্বাভাসের গুরুত্বপূর্ণ দিক তুলে ধরে, এই গবেষণাটি বাংলাদেশে সৌর বিকিরণ পূর্বাভাসকে আলোকিত করতে উন্নত মেশিন লার্নিং কৌশল ব্যবহার করেছে। এনসেম্বল মেশিন লার্নিং একটি অত্যাবশ্যক হাতিয়ার হয়ে উঠেছে যা আরও শক্তিশালী এবং সঠিক পূর্বাভাস কাঠামো প্রদান করে বিভিন্ন মডেল আউটপুট একত্রিত করে সৌর বিকিরণ পূর্বাভাসের নির্ভুলতা এবং নির্ভরযোগ্যতাকে উল্লেখযোগ্যভাবে বৃদ্ধি করে। পূর্ববর্তী গবেষণা প্রচেষ্টা এনসেম্বল মেশিন লার্নিং ব্যবহার করে সৌর বিকিরণ ভবিষ্যদ্বাণীতে এই বৈচিত্র্যময় পরিসরের অন্বেষণকে উপেক্ষা করেছে। এই গবেষণাটি কৃত্রিম বুদ্ধিমত্তার কৌশলগুলির উপর একটি বিশদ পরীক্ষা পরিচালনা করে এই ব্যবধানটি সমাধান করে। তদুপরি, গবেষণাটি এনসেম্বল মেশিন লার্নিং সহ ব্যাখ্যাযোগ্য AI (XAI) প্রবর্তন করে, ভবিষ্যদ্বাণীগুলিকে প্রভাবিত করার কারণগুলির উপর আলোকপাত করে এবং সৌর শক্তি সেক্টরে সিদ্ধান্ত গ্রহণকারীদের জন্য মূল্যবান অন্তর্দৃষ্টি প্রদান করে। উল্লেখযোগ্যভাবে, গবেষণাটি বাংলাদেশের জন্য নির্দিষ্ট একটি XAI-ভিত্তিক বিশ্লেষণের পথপ্রদর্শক, যা সৌর বিকিরণ পূর্বাভাসে একটি উল্লেখযোগ্য অগ্রগতি চিহ্নিত করে। উপরন্তু, বিভিন্ন ক্লাস্টারিং কৌশল এবং LightGBM অ্যালগরিদমকে অন্তর্ভুক্ত করে একটি অভিনব হাইব্রিডাইজড পদ্ধতি চালু করা হয়েছে, যা সৌর বিকিরণ পূর্বাভাসের জন্য একটি কার্যকর কাঠামো প্রদান করে। ফলস্বরূপ, এই অধ্যয়নটি XAI, সমন্বিত পদ্ধতি এবং মেশিন লার্নিংকে একীভূত করে এমন একটি বিস্তৃত পদ্ধতির প্রস্তাব দিয়ে সৌর বিকিরণ পূর্বাভাস সম্পর্কে আমাদের বোঝার এবং অপ্টিমাইজেশানে অবদান রাখে। আমরা আরও অবদান হিসাবে XAI এবং Ensemble-এর উপর ভিত্তি করে একটি autoML টুল তৈরি করেছি। আমরা লো-কোড PyCaret মেশিন লার্নিং প্যাকেজের মাধ্যমে আমাদের ফলাফল যাচাই করেছি যে, সমস্ত পদ্ধতির মধ্যে, লাইটজিবিএম সৌর বিকিরণ পূর্বাভাসের পরিপ্রেক্ষিতে আশাব্যঞ্জক ফলাফল দেখিয়েছে।

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This chapter delves into the examination of the thesis, encompassing the review, motivation, objectives, outcomes, contributions, and distribution. To grasp the research's motivation and define its objectives, an initial assessment of the current state of the problem is conducted, along with an exploration of the existing research gap. The subsequent sections elaborate on the research's contributions. Conclusively, the chapter outlines the comprehensive distribution plan for the thesis.

1.1 Research Background

Solar irradiance in Bangladesh has a profound impact, harnessing the nation's abundant sunlight for economic growth, environmental sustainability, and resilience against climate change. Situated in the solar belt, the sun provides clean energy, reduces carbon emissions, and fosters a green conscience. In a land prone to climate change impacts, solar energy serves as a dependable lifeline, powering homes, schools, and hospitals. Beyond its practical applications, the solar revolution in Bangladesh symbolizes the nation's dedication to a cleaner and brighter future, fostering unity among communities and improving livelihoods. The transformative power of the sun underscores the resilience and adaptability of the people, making precise solar radiance forecasting crucial for harnessing this sustainable energy source in the context of Bangladesh. Forecasting solar irradiance plays a pivotal role in our quest for sustainable energy. It enables us to seamlessly integrate solar power into our energy grid, ensuring a consistent supply of eco-friendly electricity and decreasing our dependence on fossil fuels. This foresight involves deciphering atmospheric circumstances, cloud formations, and solar rhythms, harmonizing scientific knowledge with the natural world. It stands as a testament to human inventiveness, offering a pathway to tap into the universe's energy reservoir. This capacity to predict the sun's energy emissions resembles a celestial crystal ball, steering us towards a more eco-friendly and robust tomorrow. Solar radiation forecasting through machine learning is a cutting-edge approach that blends the power of AI with the sun's unpredictable nature. By analyzing historical data and real-time atmospheric conditions, machine learning models provide highly accurate predictions of solar radiation, enabling optimal energy production for solar systems. This technology transcends traditional weather forecasting, offering significant benefits for solar farms, businesses, and households, reducing costs and minimizing environmental impact. It represents a harmonious union of nature and innovation, driving us toward a cleaner, more efficient future where every ray of sunlight is maximized for energy generation.

Ensemble machine learning is a crucial technique in AI and data science, as it combines the predictions of multiple models to improve accuracy and model robustness. By leveraging the strengths of diverse algorithms, ensembles reduce overfitting and enhance generalization. Popular ensemble methods like Random Forests and Gradient Boosting assemble weak learners into strong ones, making them adaptable to complex real-world data. In the realm of solar radiance prediction, ensemble machine learning also plays a significant role in achieving successful outcomes. Ensembles are vital for feature importance and model interpretability, benefiting fields from finance to healthcare. They are pivotal in achieving more accurate and reliable predictive models in today's data-rich world. In the realm of solar radiance prediction in the context of Bangladesh, ensemble machine learning also plays a significant role in achieving successful outcomes. The ability of solar radiation (Rs) to support life on earth, regulate the climate and weather, and supply a renewable energy source is all dependent on it ("A Novel Machine Learning Approach for Solar Radiation Estimation", 2023). Prediction and reliable energy production are hampered by its erratic and intermittent nature. To estimate Rs, ("A Novel Machine Learning Approach for Solar Radiation

Estimation", 2023) suggests a novel framework that combines different machine learning models and takes into account overlooked factors. Recursive Feature Elimination (RFE) is employed by the framework in conjunction with algorithms like random forest, logistic regression, decision trees, Pearson correlation, and gradient boosting models. The models perform admirably, with the logistic regression model standing out for its exceptional capabilities ("A Novel Machine Learning Approach for Solar Radiation Estimation", 2023). The authors in (Alam, Al-Ismail, Hossain, & Rahman, 2023) employed a combination of ensemble machine-learning models, including Gradient-Boosting Regressor, Adaboost Regressor, Random Forest Regressor, and Bagging Regressor to forecast solar irradiation in Bangladesh.

Explainable AI (XAI) is a transformative concept in artificial intelligence that aims to enhance the transparency and interpretability of AI systems. It strives to provide human users with a clear understanding of how AI models reach their decisions, moving away from the traditional "black-box" approach. XAI utilizes various methods, including visualizations and feature analysis, to uncover the rationale behind AI predictions. This transparency is vital for applications where decision-making affects people's lives, such as healthcare and finance, and it aids in addressing biases and ensuring compliance with ethical standards. XAI plays a crucial role in building trust, accountability, and ethical responsibility in AI systems. A comprehensive examination of explainable artificial intelligence (XAI) was presented in (Arrieta et al., 2020), covering aspects such as concepts, taxonomies, opportunities, challenges, and the adoption of XAI tools. The authors (Lee, Oh, & Kim, 2020) utilized XAI methodologies to interpret the results of load forecasting generated by an XGBoost model. They then demonstrated their analysis using SHAP technique. In a study outlined in (Pierrot & Goude, 2011), researchers conducted short-term electricity load forecasting using generalized additive models. This approach facilitated the amalgamation of a regressive component incorporating explanatory variables (such as weather, calendar variables, and global trends) and an auto-regressive component encompassing lagged loads.

XAI in solar irradiance prediction introduces transparency and interpretability to the forecasting of solar energy output. It aims to clarify why AI models make specific solar irradiance predictions, providing valuable insights for decision-makers in the solar energy sector. XAI techniques reveal the factors influencing predictions, helping users understand the impact of meteorological variables and historical data on accuracy. This knowledge is instrumental in refining models and optimizing energy generation. XAI promotes trust, informed decisions, and the wider adoption of solar energy as a sustainable power source.

Clustering weather data serves the purpose of uncovering meaningful patterns, relationships, and variations within meteorological information. By grouping similar weather conditions, clustering enables meteorologists and researchers to identify regional and temporal trends, seasonal variations, and anomalies in the data. This process aids in understanding the complex dynamics of climate, allowing for more accurate predictions, targeted resource management, and improved responses to extreme weather events. Furthermore, clustering supports the identification of distinct climate zones, the spatial analysis of geographical regions with similar weather patterns, and the customization of forecasts for specific areas. Overall, clustering weather data is instrumental in extracting valuable insights that contribute to more effective decision-making in fields such as meteorology, climate science, and emergency preparedness. Bae et al. explored the utilization of a support vector machine (SVM) in conjunction with k-means clustering to predict solar irradiance one hour ahead (Bae, Jang, & Sung, 2016). Their study incorporated diverse meteorological factors, such as cloud cover, as inputs. The findings underscored the superior performance of the SVM regression model compared to both artificial neural network (ANN) and nonlinear autoregressive (NAR) approaches for solar irradiance forecasting.

1.2 Motivation

The motivation for our research stems from the urgent need to address the challenges associated with solar energy utilization, particularly in regions characterized by dynamic environmental conditions like Bangladesh. As the global community increasingly recognizes the imperative of transitioning towards sustainable and renewable energy sources, the role of solar power in mitigating climate change and promoting energy security has become increasingly prominent. However, the effective integration of solar energy into the energy grid and the optimization of renewable energy projects hinge on accurate forecasting of solar irradiance. In Bangladesh, a country abundant in solar resources, the variability of solar irradiance due to factors such as cloud cover, atmospheric conditions, and seasonal changes presents significant obstacles to realizing the full potential of solar energy. By developing innovative machine learning and deep learning techniques for solar irradiance prediction, our research seeks to bridge this gap and empower stakeholders in Bangladesh and beyond to make informed decisions regarding solar energy deployment and optimization. Our motivation is rooted in the belief that by enhancing the accuracy and reliability of solar irradiance forecasts, we can accelerate the transition towards a sustainable and greener future while simultaneously addressing pressing energy and environmental challenges facing our planet.

1.3 Problem Statement

The utilization of solar energy holds significant promise for addressing energy needs while reducing carbon emissions and promoting sustainability. However, accurately forecasting solar irradiance, particularly in regions like Bangladesh with dynamic environmental conditions, remains a challenge. The lack of precise predictions hampers the effective integration of solar power into the energy grid and the efficient planning of renewable energy projects. Bangladesh, a country with abundant solar resources, is actively seeking to harness solar energy to meet its growing energy demands and mitigate its reliance on fossil fuels. Yet, the variability of solar irradiance due to factors such as cloud cover, atmospheric conditions, and seasonal changes poses obstacles to the efficient deployment and optimization of solar energy systems.

1.4 Thesis Objectives

The following are the objectives of this study:

1. To develop a machine learning and deep learning model for accurate solar radiation prediction.
2. To incorporate explainable AI techniques to enhance model interpretability.
3. To balance prediction accuracy and model interpretability in solar radiation prediction.
4. To develop a hybridized clustering method for forecasting solar radiation.

1.5 Thesis Contributions

The research paper's contributions are as follows:

1. Solar irradiation data from 32 stations was gathered, and an in-depth investigation of ensemble learning algorithms was carried out. This investigation encompassed diverse techniques, including various averaging, boosting, bagging, stacking, and blending ensemble algorithms. To the best of our knowledge, such a comprehensive investigation has not been undertaken previously. Notably, recent studies (Alam et al., 2023) have only examined four machine-learning algorithms in their analysis.

2. Evaluating the performance of various deep learning techniques to determine the best-suited ones for solar radiation forecasting in Bangladesh. Moreover, Exploration of the underutilized CNN-LSTM models in the context of Bangladesh.

3. We incorporated nine distinct explainable AI models into our analysis. The data adaptation process for explainable AI yielded new insights, with a specific focus on identifying the most vital features from the Bangladesh perspective for predicting solar power output. As far as our knowledge extends, this marks the initial XAI-based analysis of solar radiation prediction utilizing a dataset from Bangladesh.

4. We have developed a hybridized method for forecasting solar radiation, which integrates clustering algorithms with the LightGBM algorithm. Our approach begins with the application of clustering algorithms, followed by the utilization of the LightGBM Regressor on each cluster. Specifically, we have assessed four clustering algorithms for this purpose: K-means, Mini-batch K-means, Fuzzy C-means, and Gaussian Mixture.

1.6 Thesis Distribution

The thesis consists of six main chapters. Chapter 2 offers an extensive review of the literature regarding the prediction of solar radiation. Chapter 3 explores the proposed methodology and the algorithms utilized in the research. In Chapter 4, the experimental results are presented. Chapter 5 addresses Engineering Considerations, Challenges, and Remedies. Finally, Chapter 6 discusses the findings, draws conclusions, and proposes directions for future research.

CHAPTER 2

LITERATURE REVIEW

This chapter explores the realm of Machine Learning (ML) and Deep Learning (DL) algorithms, focusing on their applications in predicting Solar Radiation. Additionally, it investigates various Explainable AI (XAI) techniques and clustering methods. Through a thorough review of existing literature, we aim to grasp the current landscape of research and advancements in these domains. The goal of this section is to furnish a thorough comprehension of ML, DL, XAI, and Clustering techniques, elucidating their roles in Solar Radiation Prediction and informing forthcoming endeavours in this arena.

We divided the literature review into three parts:

- Phase 1: Machine Learning and Deep Learning Techniques for Solar Radiation Prediction
- Phase 2: Explainable Artificial Intelligence (XAI) in the context of Solar Radiation Prediction
- Phase 3: Hybridized Clustering to Forecast Solar Radiation.

2.1 Phase 1: Machine Learning and Deep Learning Techniques for Solar Radiation Prediction

An ensemble feature selection method is proposed ("Solar Radiation Forecasting Using Machine Learning and Ensemble Feature Selection", 2022) to select pertinent input parameters and their past observation values. The paper compares the performance of various machine-learning algorithms for solar radiation forecasting. Forecasting accuracy is increased by the suggested ensemble feature selection method, and the Voting-Average algorithm outperforms other algorithms across all prediction time horizons. Hirata and Aihara employed an infinite-dimensional delay coordinates time series model for solar irradiance prediction ("Improving time series prediction of solar irradiance after sunrise: Comparison among three methods for time series prediction", 2017). Their research highlighted the effectiveness of this method, particularly in forecasting irradiance levels during the post-sunrise period. Frimane et al. proposed an innovative approach using a Dirichlet process Gaussian mixture model to generate synthetic time-series data for solar global horizontal irradiance (GHI) at resolutions as fine as 1 minute, utilizing input data with resolutions higher than 10 minutes ("Nonparametric Bayesian-based recognition of solar irradiance conditions: Application to the generation of high temporal resolution synthetic solar irradiance data", 2019). In ("Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM", 2018), an advanced machine-learning model has been utilized to anticipate solar irradiation based on weather measurements. This model relies on long short-term memory (LSTM) networks, renowned for their extended capability to capture temporal dependencies in time series data. Notably, it exhibited commendable predictive performance in contrast to traditional backpropagation neural networks. Hourly predictions of solar irradiation are delivered by this model. Deep learning models have become the center of attention in recent times due to their outperforming traditional shallow models in various applications (Kumari & Toshniwal, 2021). For precise solar radiation forecasting, numerous models and algorithms have been created, including wavelet long short-term memory (WLSTM) (Alizamir et al., 2023), attention-oriented long short-term memory (ALSTM) (Irshad et al., 2023), and modified bidirectional gated recurrent unit (MBGRU) (Venu et al., 2023). According to (Kumari & Toshniwal, 2021), in comparison to CNN, LSTM, GRU, RNN, and DNN, the fusion of CNN-LSTM demonstrates a substantial enhancement in prediction accuracy, with improvements of 3.62, 25.29, 34.66, 37.37, and 26.20 per cent, respectively. According to (Alkhayat & Mehmood, 2021), recurrent neural network models, including those with long short-term memories and gated recurrent units, are second most frequently used in this field after hybrid forecasting models, with convolutional neural networks coming in third. They also discovered that multistep-ahead and probabilistic forecasting techniques are gaining popularity (Alkhayat & Mehmood, 2021).In the proposed approach of (Aslam, Lee, Kim, Lee, & Hong, 2020), there is a comparison of contemporary deep learning and machine learning architectures, encompassing GRUs, LSTM, RNN, FFNN, and SVR models. Additionally, it stated that while all the models worked well, GRU performed more proficiently than the other models (Aslam et al., 2020). The effectiveness of the proposed models of (Aslam et al., 2020) is demonstrated by comparison with the current state-of-the-art method for long-term solar radiation forecasting, namely random forest regression (RFR), where the proposed models outperformed the traditional method. Deep neural networks are used in (Alzahrani, Shamsi, Dagli, & Ferdowsi, 2017) to present a method for predicting solar irradiance. To forecast irradiance, the DRNN is utilized (Alzahrani et al., 2017). According to the performance tests of (Alzahrani et al., 2017), the outcomes demonstrate that deep learning neural networks are superior to all other techniques. The study of (Irshad et al., 2023) presents a model for solar radiation prediction that combines arithmetic optimization with hybrid deep learning and predicts solar radiation with a high degree of accuracy. In (Faisal et al., 2022), the authors introduced an approach that utilizes neural networks and weather data from five distinct cities in Bangladesh to forecast solar radiation. Using the meteorological data, RNN, LSTM, and GRU were trained (Faisal et al., 2022). Among them, the GRU model outperformed the other two models, with a MAPE score of 19.28 per cent (Faisal et al., 2022). While in (Miskat et al., 2023), the authors discussed the importance of predicting solar radiation in Bangladesh and a brief discussion of various deep learning algorithms used. (Abedin et al., 2017) proposes using ANN to predict solar radiation in different locations in Bangladesh. The National Solar Radiation Database (NSRDB) was used by the authors of (Anwar, Islam, & Alam, 2023) to gather Bangladeshi data. They used state-of-the-art N-BEATS architecture for time series analysis, which produced impressive results with minimal computational time and cost (Anwar et al., 2023).

2.2 Phase 2: Explainable Artificial Intelligence (XAI) in the context of Solar Radiation Prediction

Based on the current state of the art, two major research gaps have been identified. To begin with, there has been little study into applying explainable AI to solar radiation forecasts. However, a comparison of explainable AI algorithms in the context of solar radiation prediction is also absent. Moreover, the explainable AI has yet to be thoroughly investigated. It is worth noting here that few papers explore XAI for Solar Photovoltaic (PV) fault energy forecasting (see Table 2.1), but not on the solar radiation prediction from weather data.

Table 2.1: Explainable AI in Solar Radiation Prediction field

No	References	Field Description	Models	Explainable AI Tools
1	(Sarp, Kuzlu, Cali, Elma, & Güler, 2021)	Uses a high-resolution dataset and an explainable AI (XAI) tool to demonstrate how solar photovoltaic (PV) energy forecasting can be done.	XGBoost	ELI5
2	(Kuzlu, Cali, Sharma, & Güler, 2020)	This article provides a number of XAI tool adoption use cases for smart grid applications, including LIME, SHAP, and ELI5, for forecasting solar PV energy.	Random Forest Regression (RFR)	LIME, SHAP, ELI5.
3	(Prasad et al., 2023)	an early warning system based on artificial intelligence that is specifically suited for UV index (UVI) forecasting over the short term and integrates ground-based and satellite-derived predictors for Australian hotspots with high UV exposure.	EJH-X-DNN	LIME, SHAP, Permutation feature importance (PFI).
4	(Utama, Meske, Schneider, Schlatmann, & Ulbrich, 2023)	This research employs explainable artificial intelligence (XAI) methodologies to derive explanations from a multi-layer perceptron (MLP) model, specifically in the context of detecting faults in solar photovoltaic systems.	MLP	Anchors, DiCE, SHAP

2.3 Phase 3: Hybridized Clustering to Forecast Solar Radiation

The authors in (Ayodele, Ogunjuyigbe, Amedu, & Munda, 2019) present a k-means-SVR method for solar irradiation prediction, emphasizing clustering's role in accuracy. In (Benmouiza, Tadj, & Cheknane, 2016), the authors optimize stand-alone PV systems with hourly solar radiation classification (fuzzy c-means). The paper utilizes a Gaussian mixture model to predict solar irradiance and optimize stand-alone PV systems, integrating hourly solar radiation classification through a genetic algorithm (Wahbah, EL-Fouly, & Zahawi, 2020).

2.4 Research Gap

Seven notable research gaps emerge from the current state of research.

1. There is a limited exploration of ensemble-based machine-learning methods for solar radiation prediction.

2. There has been limited investigation into the application of machine learning and deep learning techniques for predicting solar radiation specifically within the context of Bangladesh.

3. The utilization of CNN-LSTM models for forecasting solar radiation remains an underexplored area of study.

4. A few publications have addressed the regression problem of solar irradiance production independently, but none have examined it in-depth within the framework of different AI tools, with a primary focus on feature importance.

5. Understanding a feature's significance is essential to comprehend the behaviour of the model, figuring out what major variables affect predictions, and even enhancing model performance. XAI approaches assign weight to various input features to give

light on why a model generated a particular prediction.

6. The utilization of multiple clustering methods for solar radiation prediction has been limited in exploration.

7. There is a lack of investigation into season-specific clustering for predicting solar radiation.

CHAPTER 3

METHODOLOGY

This chapter investigates the use of Machine Learning (ML) and Deep Learning (DL) algorithms, such as Decision Tree regression, Linear Regression, and others. It also looks into clustering techniques and Explainable AI (XAI) approaches to solar radiation prediction. The chapter aims to comprehend the current research landscape and guide future endeavours in solar radiation prediction through a thorough review of the existing literature.

Figure 3.1 displays framework for proposed methodology of solar radiation prediction. We divided our proposed methodology into three parts:

- Phase 1: Machine Learning and Deep Learning Techniques for Solar Radiation Prediction
- Phase 2: Explainable Artificial Intelligence (XAI) in the context of Solar Radiation Prediction
- Phase 3: Hybridized Clustering to Forecast Solar Radiation.

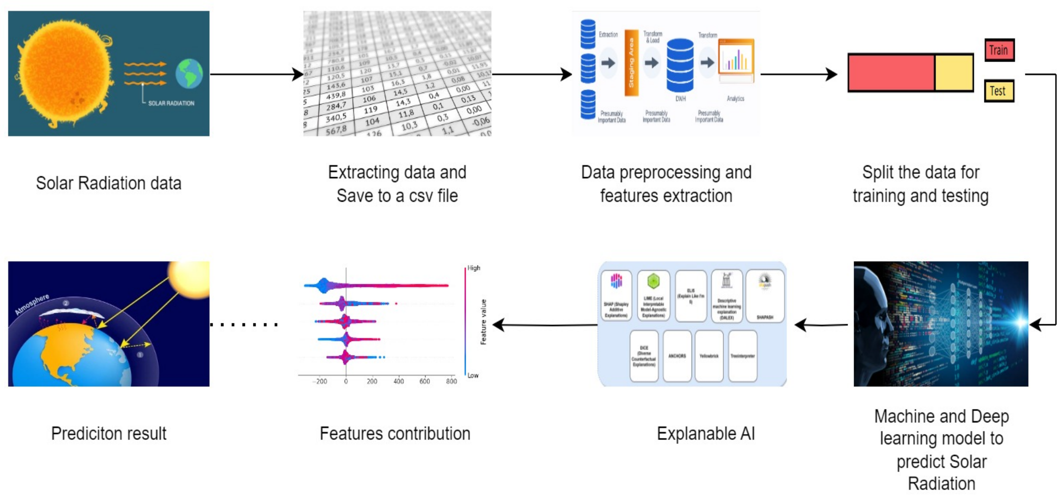


Figure 3.1: Framework for proposed methodology of solar radiation prediction

3.1 Dataset Description

3.1.1 Dataset I

The research station known as HI-SEAS (Hawaii Space Exploration Analog and Simulation) was chosen as the study area. The International MoonBase Alliance is currently in charge of running HI-SEAS, a research station that simulates Mars and Moon exploration. The 1,200-square-foot HI-SEAS habitat is situated in a Mars-like environment on Hawai'i Island's Mauna Loa volcano. In cooperation with dozens of space agencies, businesses, and groups from around the world, HI-SEAS has hosted tens of analog space missions, including nine successful NASA Mars simulation missions lasting four to twelve months.

The dataset consists of meteorological data collected over four months (September to December 2016) from the HI-SEAS weather station. This information was gathered between NASA Missions IV and V and is available on Kaggle under(, n.d.-a) There are eleven parameters and 32686 training samples in this dataset. Table 3.1 provides a summary of the attributes, data types used in the dataset, and their nomenclature used in this study. A heatmap to visualize solar radiation forecasting data is illustrated in Figure 3.2.

Table 3.1: Attributes of the dataset

No	Nomenclatures	Features
1	U_t	UNIXTime
2	D_{ata}	Data
3	T_{tt}	Time
6	R_{rad}	Radiation
7	T_{temp}	Temperature
8	$P_{ressure}$	Pressure
9	$H_{umidity}$	Humidity
4	Tsr_{tt}	Time of Sun Rise
5	Tn_{sn}	Time of Sun Set
10	W_{dir}	Direction of wind(degrees)
11	W_{spd}	Speed of wind

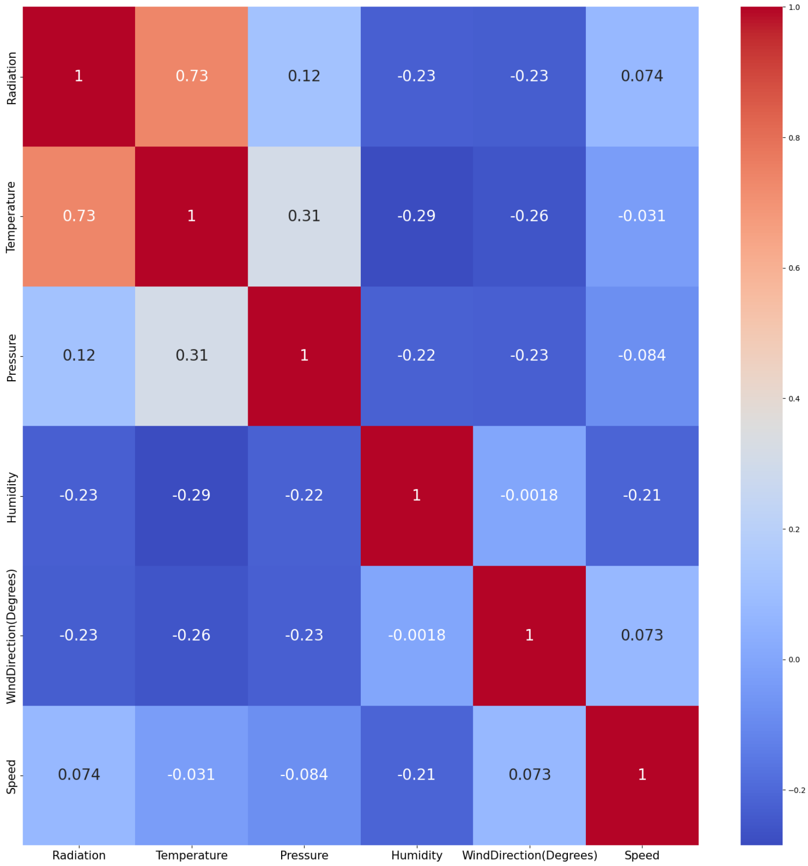


Figure 3.2: Visualizing solar radiation prediction data using heatmap for dataset I.

3.1.2 Dataset II

We have gathered a set of individual meteorological data variables spanning from 1961 to 2017 from the Bangladesh Agricultural Research Council (BARC) website (, n.d.-b). These data were obtained from 32 stations spread across the entirety of Bangladesh and were provided by the Bangladesh Meteorological Department (BMD). The dataset includes 15,063 training samples and 8 columns. Table 3.2 represents the features of the dataset. A heatmap to visualize solar radiation forecasting data is illustrated in Figure 3.3.

Table 3.2: Features of the dataset

No	Nomenclatures	Features
1	S_{rad}	Solar Radiation
2	Max_{temp}	Maximum Temperature
3	Min_{temp}	Minimum Temperature
4	$S_{humidity}$	Humidity
5	S_{cloud}	Cloud Coverage
6	S_{wind}	Wind Speed
7	$S_{sunshine}$	Sunshine (Hours)
8	$S_{rainfall}$	Rainfall

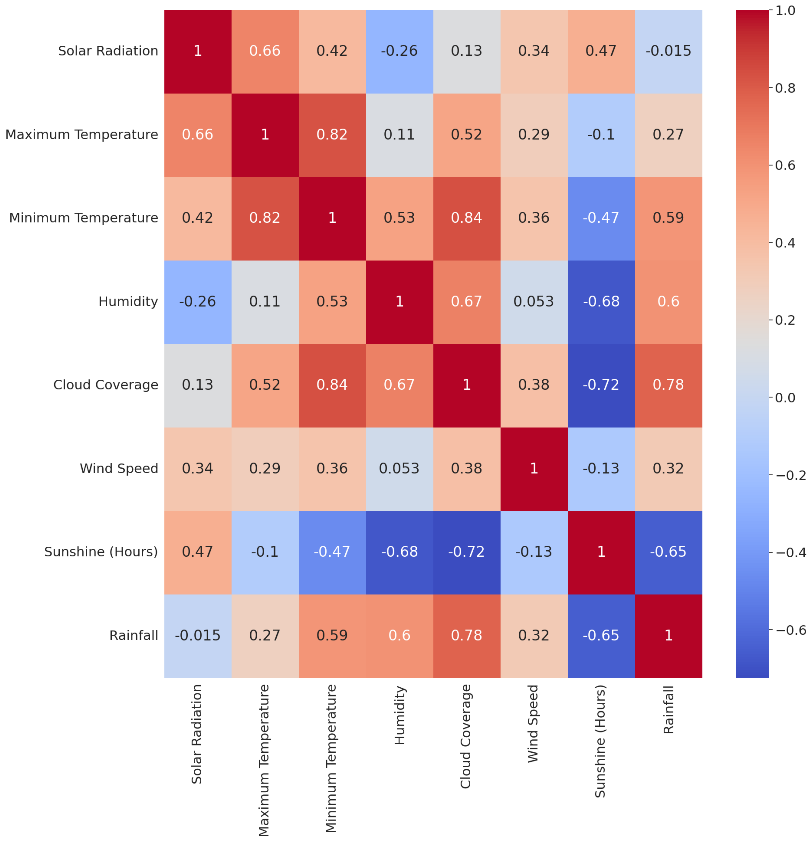


Figure 3.3: A heatmap to visualize solar radiation forecasting data for dataset II.

3.2 Terminology

3.2.1 Machine Learning Models

The realm of predictive modeling within machine learning prioritizes minimizing model error and maximizing prediction accuracy, often at the expense of interpretability. In applied machine learning, practitioners draw upon a diverse array of algorithms borrowed, reused, and adapted from various domains, including statistics, to achieve these objectives. In this study, we employed several machine learning algorithms to analyze the data.

1. **Decision Tree Regressor:** Decision Tree Regressor is a versatile machine-learning algorithm used for solving regression problems. It is a crucial learning algorithm that is primarily used for data analysis ("Comparative study of regressor and classifier with decision tree using modern tools", 2022). It is capable of solving problems with regression and classification. It constructs a tree-like structure in which each internal node serves as a point of decision or testing based on the input features, while each leaf node yields a continuous numerical prediction. It follows a recursive process of partitioning the dataset based on the values of these features to minimize prediction errors ("Comparative study of regressor and classifier with decision tree using modern tools", 2022). Decision trees offer remarkable transparency, enabling users to comprehend the decision-making process and the importance of various features. However, they are vulnerable to overfitting, a situation where they fit the training data too closely, and this can be mitigated by applying pruning techniques and restricting the depth of the tree (Myles, Feudale, Liu, Woody, & Brown, 2004). Decision Tree Regressors are frequently employed as fundamental building blocks in ensemble methods and have a wide range of applications in domains like finance, healthcare, and environmental science. They excel in capturing complex relationships and delivering accurate numerical forecasts (Myles et al., 2004).

2. **Linear Regression:**

Linear Regression is a type of supervised machine learning model that determines the linear relationship between the independent and dependent variables by constructing the best fit linear line between them. Mathematically, we can represent a linear regression as (Groß, 2003):

$$S_{\text{pred}} = \beta_0 + \beta_1 \text{fea}_1 + \beta_2 \text{fea}_2 + \ldots + \beta_n \text{fea}_n + \varepsilon \tag{3.1}$$

Where:

S_{pred} : the predicted or dependent variable ,

β_0 : the y-intercept,

$\beta_1, \beta_2, \ldots, \beta_n$: the regression coefficients for the independent variables

$\text{fea}_1, \text{fea}_2, \ldots, \text{fea}_n$ are the independent variables (features),

ε : the error term, representing the unexplained variation in S_{pred} that is not accounted for by the independent variables.

Linear Regression derives its name from its emphasis on showcasing a direct link between independent variables and dependent variables. It essentially unveils how alterations in independent variable values impact the dependent variable (Groß, 2003). This association is illustrated by a straight-line slope. The core premise of linear Regression hinges on the assumption of a linear connection between these variables. It presupposes minimal to no multicollinearity among independent factors (Seber & Lee, 2012). In the context of linear Regression, the error term should exhibit a typical bell-shaped distribution. Deviating from this distribution could lead to challenges in estimating coefficients, resulting in excessively wide or narrow confidence intervals

(Seber & Lee, 2012). Linear Regression doesn't assume autocorrelation within error terms. Any hint of error term correlation can substantially diminish the model's accuracy, typically stemming from residual errors depending on each other (Seber & Lee, 2012).

3. **Ridge Regression:** Ridge Regression, also referred to as Tikhonov regularization or L2 regularization, is an advanced method in linear regression that extends the traditional least squares approach by incorporating a new dimension. Addressing the complexities arising from multicollinearity, which involves high correlations among independent variables, Ridge Regression serves as a stabilizing force, enhancing the robustness and dependability of the model in the presence of intricate data relationships (McDonald, 2009). In machine learning, ridge regression reduces the standard error by incorporating a penalty term into the regression approximations. It helps to obtain estimates that are more accurate. By penalizing the weights of the feature's coefficients and lowering the difference between the actual and predicted observations, this Regression carries out L2 regularization.

4. **Lasso Regression:**

   The acronym LASSO represents the Least Absolute Shrinkage and Selection. It is a method of regularization. It is preferred over regression methods for more precise prediction. This model makes use of shrinkage. Shrinkage is the process by which data values are shrunk towards a central point known as the mean. The lasso procedure encourages the use of simple, sparse models (those with fewer parameters) (Wang, Li, & Tsai, 2007). This specific kind of Regression works well for models that exhibit high levels of multicollinearity or when you wish to automate specific steps in the model selection process, such as parameter elimination and variable selection (Wang et al., 2007). L1 regularization is used by Lasso Regression. The reason it is used is that it automatically selects features when we have more features.

5. **Elastic Net:** Elastic net linear Regression regularizes regression models by combining penalties from the lasso and ridge techniques (Liang & Jacobucci, 2020). By taking into account the shortcomings of both lasso and ridge regression methods, the technique improves statistical model regularization. So Elastic Net is (Liang & Jacobucci, 2020):

$$\text{Elastic Net Loss} = \text{Lasso Regression} + \text{Ridge Regression} \tag{3.2}$$

   Elastic Net Regression is useful when a dataset has a large number of features and the goal is to prevent overfitting and perform feature selection. It is a balanced combination of Lasso and Ridge Regression, making it a suitable choice when dealing with datasets where many features may be irrelevant or multicollinear. By adjusting the two regularization parameters, λ_1 and λ_2, one can fine-tune the trade-off between feature sparsity and coefficient size. This method is particularly effective for handling high-dimensional data and for building models that select essential features while controlling the magnitudes of the coefficients (Liang & Jacobucci, 2020).

6. **BaggingRegressor:** The baggingRegressor model functions as an ensemble of distinct predictors, working in unison to create a robust and dependable forecasting system. Think of it as a diverse group of experts, each offering a unique perspective, collaborating to provide you with precise predictions (Kadiyala & Kumar, 2018).

   BaggingRegressor constructs multiple foundational regression models, often utilizing Decision Trees, although other regressors can be employed as well. Each of these models learns from a slightly different view of the data. These base models are trained on varying subsets of the dataset, allowing them to capture different subtleties and idiosyncrasies within the data. When you require a prediction, all these individual models contribute their predictions, and the BaggingRegressor amalgamates their insights to produce a final, comprehensive prediction that mitigates the risk of overfitting and enhances prediction accuracy (Amin et al., 2023).

7. **Random Forest Regressor:** Random Forest Regressor is a potent predictive tool that employs decision tree principles to boost precision and trustworthiness (Biau & Scornet, 2016). It acts as a diverse committee of decision-makers, with each tree offering a distinct viewpoint. Together, they collaborate like a council, voting to determine the most probable outcome, reducing the risk of mistakes, and yielding precise predictions. This method benefits from diversity, as each tree is trained on different data subsets and examines random sets of features. This diversity prevents excessive reliance on specific data patterns and increases resilience to noise and outliers (Biau & Scornet, 2016). Furthermore, it can gauge feature importance, revealing influential variables and providing insights into predictive factors.
Random Forest Regressor operates as a collective of experts, pooling their wisdom to provide robust forecasts. This makes it a popular choice in fields requiring accurate predictions, showcasing its exceptional ability to harness individual decision trees' strengths for superior predictive performance. The equation for Random Forest Regressor can be expressed as follows:

$$\hat{y}(\mathrm{inp}) = \frac{1}{\mathrm{tot\_dt}} \sum_{d=1}^{\mathrm{tot\_dt}} f_d(\mathrm{inp}) \tag{3.3}$$

Where:

$\hat{y}(\mathrm{inp})$: the predicted output for the input inp.

$\mathrm{tot\_dt}$: the total number of decision trees (estimators) in the random forest.

$f_d(\mathrm{inp})$: the prediction made by the d-th decision tree in the forest.

8. **Gradient Boosting Regressor:** Gradient Boosting Regressor stands out as a formidable tool in the realm of machine learning, particularly suited for tackling regression tasks. Its underlying mechanism revolves around the iterative amalgamation of predictions from numerous feeble learners (Natekin & Knoll, 2013). With each iteration, the model zeroes in on rectifying the mistakes made in previous rounds, making adjustments to its forecasts in an effort to minimize those errors. This dynamic process persists until a predetermined number of iterations is reached or until the model's performance levels off (Natekin & Knoll, 2013). The Gradient Boosting Regressor is renowned for its exceptional accuracy, resilience, and proficiency in handling intricate data relationships, rendering it a sought-after choice for predictive modeling across diverse domains.

9. **XGBoost Regressor:** The XGBoost Regressor, also known as Extreme Gradient Boosting, is a potent machine-learning algorithm designed for regression tasks (Chen & Guestrin, 2016). It operates as an ensemble learning technique that amalgamates the forecasts generated by multiple decision tree models, resulting in precise and resilient predictions. What sets XGBoost apart is its exceptional efficiency, speed, and aptitude for handling intricate, non-linear associations within data. It leverages the principles of gradient boosting, which entails the continuous reduction of the error from prior models by training new models on the residual errors from their predecessors (Chen & Guestrin, 2016). XGBoost enjoys widespread popularity in diverse fields of data science and machine learning, largely owing to its exceptional performance and adaptability.

10. **LightGBM Regressor:** LightGBM Regressor is a state-of-the-art machine learning model tailored for regression tasks. It is well-known for its remarkable speed and precision, resembling a high-performance sports car in the field of machine learning (Ke et al., 2017). What sets LightGBM apart is its ability to efficiently process data

with minimal memory usage, enabling it to handle large datasets and complex feature spaces adeptly. It operates akin to a turbocharged engine for regression, swiftly and accurately navigating through the data. However, its appeal extends beyond swiftness; LightGBM is also a finely tuned instrument. It optimizes decision trees, adapting to the intricacies of the data to construct a model that comprehends intricate relationships. LightGBM shines in handling both numerical and categorical features, showcasing its versatility across a wide spectrum of regression tasks. Additionally, its parallel computing capacity further elevates its efficiency, delivering a seamless user experience (Ke et al., 2017).

11. **AdaBoost Regressor:** AdaBoost Regressor stands out as a potent ensemble learning technique tailored for building resilient regression models. Its operation can be likened to an orchestra, where individual weak learners, akin to musicians, contribute their part (Schapire, 2003). In this analogy, AdaBoost assumes the role of a conductor, orchestrating and synchronizing these elements. Notably, it emphasizes the learners that have previously made errors, resembling a conductor rectifying discordant notes. The ensemble hones its performance through iterative refinement, aiming to minimize prediction inaccuracies (Schapire, 2003). AdaBoost amalgamates the strengths of these weak learners, crafting a robust regression model adept at capturing intricate data patterns. The ultimate model mirrors a seamless blend of diverse contributions, enabling precise predictions for the target variable. AdaBoost's proficiency in handling intricate data structures and delivering precise regression outcomes positions it as an invaluable asset in the realm of machine learning.

12. **CatBoost Regressor:** CatBoost Regressor is a cutting-edge machine learning algorithm tailored specifically for regression tasks. It stands out for its seamless handling of categorical features, eliminating the need for extensive preprocessing and enhancing user-friendliness, especially for mixed data types (Prokhorenkova,

Gusev, Vorobev, Dorogush, & Gulin, 2018). Inspired by a cat's adaptability, it strikes a balance between bias and variance, ensuring reliable predictions and preventing overfitting by learning iteratively through gradient boosting, akin to a cat mastering a skill through practice.

This algorithm excels in uncovering intricate patterns within large and complex datasets, offering valuable insights that might elude other methods. Its speed and accuracy make it ideal for real-time applications where quick and precise predictions are crucial (Prokhorenkova et al., 2018). CatBoost Regressor acts as a vigilant cat, meticulously exploring data and using its analytical ”claws” to extract meaningful information effortlessly. Its robustness and versatility have made it a preferred choice among data scientists, ensuring reliable results across various regression tasks. In essence, CatBoost Regressor combines adaptability, precision, and efficiency, simplifying the complexities of regression modeling and data analysis for professionals and enthusiasts alike.

13. **PyCaret:** PyCaret (Ali, 2020) is a groundbreaking Python library designed for data scientists and machine learning practitioners, simplifying the journey from raw data to predictive models. As an open-source, low-code framework, PyCaret efficiently manages the entire machine-learning pipeline, offering a user-friendly interface for tasks like data preprocessing, feature engineering, model selection, hyperparameter tuning, evaluating multiple machine-learning models, and deployment. By automating routine tasks such as missing value imputation and categorical variable encoding, PyCaret allows users to focus on core aspects of model creation. Its standout feature is automated model selection, facilitating quick comparison and selection of the most effective algorithms for specific datasets. PyCaret goes beyond conventional libraries by incorporating model interpretation and providing insightful visualizations of the

importance of features and SHAP values. It extends its utility to anomaly detection, ensuring a comprehensive approach to data analysis. In essence, PyCaret is more than a library; it is an innovation catalyst, democratizing access to machine learning by offering a user-friendly yet robust toolkit. It makes the art and science of predictive modeling accessible to both novices and seasoned practitioners, emphasizing simplicity in navigating the complexities of machine learning tasks.

3.2.2 Deep Learning Models

A deep learning model is a computational architecture inspired by the structure and function of the human brain's neural networks. It consists of multiple layers of interconnected nodes, or neurons, organized in a hierarchical fashion. These models learn to perform tasks by automatically identifying patterns and features in large amounts of data. Deep learning has revolutionized various fields such as computer vision, natural language processing, and speech recognition, achieving remarkable performance in tasks like image classification, language translation, and voice synthesis. Its effectiveness stems from its ability to learn complex representations from raw data, adapt to diverse domains, and generalize well to unseen examples. In this study, we have used several deep learning algorithms.

1. **Multi-Layer Perceptron (MLP)**: Multi-Layer Perceptron (MLP) in deep learning can be thought of as an orchestra, with each neuron resembling a unique instrument, contributing its own tune to the intricate computation symphony. These neural networks consist of distinct layers, akin to different sections of the orchestra, processing information hierarchically. Just as a symphony's beauty emerges from the harmonious interplay of instruments, an MLP's strength lies in its capacity to harmonize complex patterns and relationships within data, forming a sophisticated, expressive model capable of diverse tasks, from image recognition to natural language understanding. Similar to a conductor guiding musicians, training an MLP involves chore-

ographing the data flow through its layers, allowing the network to refine its collective performance and unveil the hidden nuances within the data it encounters. Therefore, MLPs serve as both artists and scientists, unraveling the intricate compositions of information that shape our digital world.

2. **AutoRegressive Integrated Moving Average (ARIMA)**: ARIMA method stands as a versatile and widely employed technique for the analysis and prediction of time series data. It encompasses three fundamental components for modeling sequential information: the AutoRegressive aspect (AR) captures the associations between current and preceding data points; the Integrated element (I) transforms the data to ensure stationarity by using differencing to eliminate trends and seasonality; and the Moving Average component (MA) accounts for the influence of previous prediction errors on future values. ARIMA models are extensively applied in various domains, including finance, economics, weather forecasting, and more, providing a robust framework for understanding, modeling, and predicting time series data by addressing both short-term patterns and long-term trends.

3. **CNN-LSTM**: Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) stand as fundamental pillars in deep learning, each contributing distinct advantages in the analysis of sequential data. LSTM serves as a vigilant guardian, preserving context and insights across time, whereas CNN specializes in identifying patterns and trends within sequences. When combined, they forge a formidable alliance: CNN swiftly discerns immediate cues, while LSTM maintains a broader context, facilitating the decoding of intricate temporal patterns. This synergy proves invaluable across tasks such as forecasting and anomaly detection, underscoring the adaptability and potency of their collaborative prowess in deep learning domains. Figure 3.4 illustrates the architecture of CNN-LSTM for solar radiation prediction.

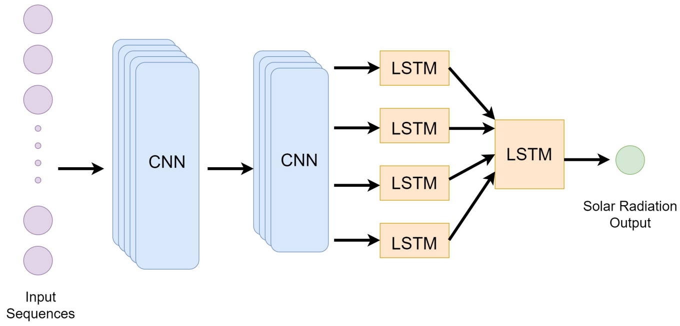


Figure 3.4: CNN-LSTM Architecture for Solar Radiation Prediction

3.2.3 Explainable Artificial Intelligence

The development of artificial intelligence systems that can offer clear reasons for their choices and outputs is known as explainable AI (XAI). By enabling humans to understand the reasoning behind AI model predictions, it seeks to enhance transparency and trust while enabling better accountability and decision-making in important fields (Holzinger, Saranti, Molnar, Biecek, & Samek, 2022). In this study, we have used several explainable artificial intelligence tools.

1. **Local and Global Interpret-ability** The extent of explanation is what separates local interpretability from global interpretability. While global interpretability is concerned with describing the model's overall performance in the prediction job, local interpretability is concerned with providing the reasoning behind a single prediction. Both types of interpretation are important, and different tools can be used to handle each, depending on whether one needs to understand the behavior of the model as a whole or just individual predictions.

   SHAP (Shapley Additive Explanations): is a highly versatile and potent technique for elucidating the predictions of machine learning models (Sushanth, Mishra, Mukhopadhyay, & Singh, 2023). It allocates significance scores to input features, providing a comprehensive understanding of their contributions to model predictions. SHAP distinguishes itself by offering both local and global interpretability, making it invaluable for explaining individual predictions and overall model behavior (Holzinger, Saranti, Molnar, Biecek, & Samek, 2020).

At its core, SHAP draws from game theory and the concept of Shapley values, initially developed for equitable reward allocation in cooperative games. In machine learning, Shapley values are employed to attribute each feature's impact on a specific prediction (Rozemberczki et al., 2022). One of SHAP's primary strengths is its model-agnostic nature, allowing it to seamlessly work with a wide range of AI models, from decision trees to neural networks. This flexibility ensures that SHAP can be applied universally, making it an indispensable tool for model interpretation (Misheva, Osterrieder, Hirsa, Kulkarni, & Lin, 2021). SHAP also ensures fairness and consistency when evaluating feature importance across different feature combinations and data distributions. This is especially crucial in domains like healthcare and finance, where model fairness and trust are paramount concerns (Holzinger et al., 2020).

The central outcome of SHAP's analysis is the computation of Shapley values for each feature, representing their average contribution across all potential feature combinations. This reveals how much each feature influences predictions when considered in isolation. Visualizations like SHAP summary plots and waterfall charts simplify these explanations for data scientists and stakeholders (Rozemberczki et al., 2022). SHAP finds applications across diverse domains, such as healthcare for explaining medical diagnoses, finance for justifying loan approval decisions, natural language processing for interpreting text-based models, and image recognition for understanding neural network outputs. In each domain, SHAP enhances transparency and interpretability, promoting responsible AI and informed decision-making (Sim et al., 2022).

The Shapley value for feature *it* is calculated as (Sushanth et al., 2023):

$$\phi_{it} = \sum \frac{V(S_{hap} \cup \{it\}) - V(S_{hap})}{(|S_{hap}|+1)!}$$

Where:

$$\begin{aligned} \phi_{it} &: \text{The Shapley value for feature } it \\ V(S_{hap} \cup \{it\}) &: \text{The prediction when feature } it \text{ is included in the subset } S_{hap} \\ V(S_{hap}) &: \text{The prediction when feature } it \text{ is not included in the subset } S_{hap} \\ |S_{hap}| &: \text{The number of features in subset } S_{hap} \\ (|S_{hap}|+1)! &: \text{The number of all possible permutations of subsets that include feature } it \end{aligned}$$

LIME (Local Interpretable Model-Agnostic Explanations): is a vital tool in Explainable AI (XAI) that focuses on providing local, understandable explanations for specific predictions made by complex machine learning models (Dieber & Kirrane, 2020). Instead of explaining the entire model, LIME zooms in on why a particular prediction was made for a specific input, fostering transparency and user comprehension.

Interpret ability is a central tenet of LIME, ensuring that users can grasp and trust the model's decision-making process. Rather than revealing the black-box model's intricate inner workings, LIME approximates its behavior using interpretable models like linear regression or decision trees (Hase & Bansal, 2020). To achieve this, LIME generates modified versions of input data, introducing slight variations, and observes the corresponding predictions from the black-box model (Dieber & Kirrane, 2020). It then constructs an interpretable model, such as a decision tree, based on these perturbed instances, providing a simple explanation for the specific prediction in question (Dieber & Kirrane, 2020).

This local explanation approach enhances decision-making and trust in AI systems, especially in high-stakes domains like healthcare and finance. LIME's power lies in its ability to bridge the gap between the complexity of advanced models and the need for understandable decision justifications(Zhao, Huang, Huang, Robu, & Flynn, 2021). By offering users a comprehensible surrogate model for individual predictions, LIME empowers them to evaluate and validate the model's decisions on a case-by-case basis (Kamal et al., 2021). Ultimately, LIME contributes to the broader goal of making AI both powerful and accountable, ensuring that AI systems are not just enigmatic black boxes but tools that people can trust and understand (Kamal et al., 2021).

The mathematical representation of LIME is as follows (Zhang, Damiani, Al Hamadi, Yeun, & Taher, 2022):

$$\mathrm{Exp}(p) = \arg\min_{g} \left[L(f,g,p) + \lambda(g)\right] \tag{3.4}$$

Where:

$\mathrm{Exp}(p)$: the explanation provided by LIME for a specific instance p.

$\arg\min_{g}$: the interpretable model g that minimizes the following expression.

$L(f,g,p)$: the loss function quantifying the difference between the predictions of the black-box model f and the interpretable model g for the instance p.

$\lambda(g)$: a complexity term assessing the complexity of the interpretable model g.

2. **Permutation Feature Importance** In machine learning, a method called permutation feature importance is used to determine how important each feature is in a predictive model. It entails assessing the effect on the model's performance metrics, like accuracy or mean squared error, after methodically permuting the values of a single feature. The significance of the disturbed feature is revealed by the performance loss when comparing the model's initial performance with the performance following feature value randomization. This approach offers a model-independent means of assessing feature significance, adaptable to different algorithms, and is especially helpful for intricate models or datasets with high-dimensional features.

ELI5-Explain Like I'm 5: The **ELI5 (Explain Like I'm 5)** method is a powerful communication approach that simplifies complex AI concepts and models, making them understandable to people of all ages, regardless of their technical background. It is essential in a world where artificial intelligence (AI) and machine learning are driving innovation across various industries.

ELI5 breaks down intricate AI ideas into simple language, avoiding technical jargon and mathematical equations to ensure inclusivity and accessibility (Kuzlu et al., 2020). This method utilizes analogies and visualizations, likening AI processes to familiar concepts, and often employs real-world examples to make AI relatable. For instance, it might describe a recommendation algorithm as a friendly librarian suggesting books based on your previous preferences. One of ELI5's primary purposes is bridging the gap between technical experts and non-experts, fostering meaningful discussions about AI's potential benefits and risks (El-Sappagh, Alonso, Islam, Sultan, & Kwak, 2021). It addresses fundamental questions such as ”Why did the AI make this decision?” and ”How does it work?” clearly and concisely, satisfying the curiosity of both experts and laypersons (El-Sappagh et al., 2021).

Transparency is a key element in responsible AI use, and ELI5 contributes to it by making AI systems more transparent and comprehensible (Chadaga et al., 2023). When people understand how AI functions, they are more likely to trust its decisions and assess its ethical implications accurately (Chadaga et al., 2023). In conclusion, the ELI5 method simplifies complex AI concepts through plain language, analogies, visual aids, and real-world examples. It promotes transparency, encourages meaningful discussions, and ensures that AI benefits are accessible to a broader audience, playing a vital role in shaping the discourse on this transformative technology.

3. **Model-Specific and Model-Agnostic** The distinction between model-specific and model-agnostic interpretability lies in the interpretation. Linear models, for instance, have model-specific interpretations tied to regression weights, and decision trees have their unique interpretations based on splits. Conversely, model-agnostic tools apply to any machine learning model and are implemented after the model has undergone training, serving a post-hoc purpose.

4. **Theoretical Background**

   Partial Dependency Plot: The marginal impact that one or two features have on the anticipated result of a machine learning model is displayed by the partial dependence plot, often known as the short PDP or PD plot (J. H. Friedman 200130). If there is a complex link between a feature and the target, a partial dependence plot can demonstrate it. Plots of partial dependence (PDP) are a helpful tool for understanding how features and forecasts relate to one another. It enables us to comprehend how variations in a given feature's value affect the model's predictions.

   Accumulated Local Effects (ALE): Accumulated Local Effects (ALE) is an innovative interpretation method introduced by Apley in 2018. Distinguished from other techniques, ALE focuses on differences in predictions rather than averages, enhancing its effectiveness in mitigating the impact of correlated features. ALE calculates local effects by dividing a feature into intervals and determining the differences in predictions within each interval, providing a more precise representation of the feature's impact on the model's prediction. ALE plots offer a more efficient and unbiased alternative to partial dependence plots (PDPs), making them an excellent tool for visualizing feature impact on model predictions. The visualization of accumulated local effects enables a deeper understanding of how features influence the model, facilitating more informed decision-making in the interpretability of machine learning models.

Individual Conditional Expectation (ICE): One visualization method employed in the field of machine learning interpretability is called Individual Conditional Expectation (ICE). It is especially helpful for comprehending how one input variable affects a model's expected results. Plotting the expected reaction for each instance in a dataset when the value of a particular variable changes while maintaining the constant value of all other features gives ICE a comprehensive and personalized picture. In essence, when the selected characteristic varies, ICE creates a set of curves, each of which represents the model's predictions for a particular instance. Through the visualization of these distinct conditional expectations, analysts can have a better understanding of a model's local behavior, spot trends, and comprehend how the model's predictions react to changes in particular input features.

5. **Tools**

Descriptive machine learning explanation (DALEX): emerges as a robust toolkit tailored for seamless integration within the R programming environment, primarily dedicated to enhancing the interpretability facets of machine learning models (Ekanayake et al., 2023). The central aim of this toolkit is to simplify the comprehension of complex model behaviors by equipping users with tools that facilitate the creation of easily understandable explanations (Ekanayake et al., 2023). This objective is achieved through a variety of methods, including visualizing feature contributions, computing variable importance, and actively assisting in model evaluation. These functionalities empower users to deeply explore the reasons behind model predictions (Ekanayake et al., 2023).

A distinctive feature of DALEX is its ability to vividly illustrate feature contributions, effectively displaying the influence of each specific feature on the ultimate model prediction (Baniecki, Kretowicz, Piatyszek, Wisniewski, & Biecek, 2021). This visual depiction serves as a potent instrument in identifying impactful features and grasping the key aspects of input data that drive the model's decision-making process (Baniecki et al., 2021).

Equally important is the toolkit's capability to quantify the importance of variables, shedding light on the variables that exert the most influence on a model's performance. This quantitative assessment helps users focus their attention on the factors that truly matter when dissecting and explaining the model's behavior (Biecek, 2018).

Beyond its other attributes, DALEX also offers robust support for model evaluation, providing users with the tools to gauge how well the model's performance aligns with actual observed outcomes (Biecek, 2018). This evaluative process instills confidence in the model's reliability and facilitates informed decision-making based on its predictions (Biecek, 2018). At its essence, DALEX aims to demystify the internal mechanisms of machine learning models, delivering accessible insights into the intricate processes that drive predictions. By delivering transparent and understandable explanations, it fosters a sense of trust in the results generated by the model, thus promoting responsible and informed decision-making across various applications of artificial intelligence.

SHAPASH: has emerged as an invaluable tool in the realm of machine learning, shedding light on the complex inner workings of models that often baffle users (Das, Sultana, Bhattacharya, Sengupta, & De, 2023). Through its user-friendly interface enriched with informative visualizations and explanatory tools, it acts as a vital bridge between intricate algorithms and human understanding. By employing advanced techniques such as Shapley values, permutation analysis, and partial dependence plots, SHAPASH exposes the hidden mechanisms of models, aiding in a deep comprehension of how different features contribute to predictions (Das et al., 2023).

One notable aspect is its ability to also detect biases, helping users uncover potential prejudices and take corrective actions to ensure fairness and impartiality in model outcomes (Yadav, Mahalle, Sathe, & Anerao, 2023). The tool's transparency-boosting capabilities are particularly valuable for users seeking well-grounded decisions based on insights from the model. The interactive visualizations provided by SHAPASH empower users to analyze model behaviors across various scenarios, allowing them to identify strengths and weaknesses (Yadav et al., 2023).

Moreover, by visually presenting the significance of features, SHAPASH aids in feature curation and dimensionality reduction, streamlining the process of refining models (Das et al., 2023). Its utility goes beyond mere model understanding; it empowers users to meticulously improve their models for greater precision, ensuring their applicability in real-world situations. For data scientists, analysts, and decision-makers striving to demystify machine learning models, SHAPASH proves to be an indispensable tool. Its amalgamation of sophisticated techniques and user-friendly interfaces marks a significant advancement in making AI and machine learning more accessible and transparent to a broader audience. Essentially, SHAPASH not only reveals insights but also imparts the ability to intuitively grasp models, enabling users to harness the potential of complex models for well-informed and impactful decision-making (Yadav et al., 2023).

6. **Counterfactual Explanations**

   Dice-Diverse Counterfactual Explanations (DiCE): Recent research enables the acquisition of conditions that would alter model predictions using counterfactual explanations, employing ”What if” hypothetical scenarios for a given input. One implementation, such as DiCE (Diverse Counterfactual Explanations), facilitates the exploration of counterfactual explanations for machine learning models, offering insights into how changes in input conditions could impact predictions.

   DiCE introduces counterfactual (CF) explanations, revealing feature-perturbed iterations of the same instance, and offering insightful ”what-if” scenarios for understanding model output. This capability serves as a valuable supplement to other explanation methods, catering to the needs of both end-users and model developers.

   In January 2020, the recent paper ”Explaining Machine Learning Classifiers through Diverse Counterfactual Example” and the implementation of DiCE were published on github by Microsoft Research’s Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. DiCE is one of the counterfactual explanations that their study has put into practice. Their recent research, which produces a variety of counterfactual explanations for any ML model, served as the foundation for this implementation.

7. **Visualization**

   Yellowbrick: The Python library Yellowbrick was created to improve and visualize the machine learning workflow. It offers a wide range of features to help practitioners assess and understand models. Yellowbrick is a collection of visualizers that make it easier to explore model behavior, feature importance, and diagnostic metrics. It was developed on top of the scikit-learn package. It contains feature analysis visualizations including scatter plots and parallel coordinates, as well as tools for evaluating model performance like ROC-AUC curves and confusion matrices. Yellowbrick is a useful tool for novice and expert data scientists alike since it streamlines the sometimes difficult process of choosing the best model, adjusting hyperparameters, and learning about the inner workings of machine learning models.

8. **Rule based interpretibability**

   Anchor: Anchor explanations represent a category of explanation techniques within the domain of Explainable Artificial Intelligence (XAI). They are designed to offer clear, interpretable, and comprehensible rationales for the predictions generated by machine learning models. Essentially, an anchor explanation constitutes a distinct rule or condition that, if satisfied, would lead the model to produce an identical prediction. These rules are presented in a human-readable format, enhancing the ease with which end-users can grasp the decision-making process employed by the model.

9. **Tree based interpretibability**

   TreeInterpreter: A Python package called TreeInterpreter offers resources for deciphering the forecasts of models based on trees, especially decision trees and random forests. Users can break down an ensemble of trees' predictions into the contributions of various characteristics using the library. TreeInterpreter makes complex tree-based models easier to comprehend by analyzing the contributions of each feature to the model's predictions. This collection of resources is very helpful for learning how ensemble models make decisions and how various features affect the final forecasts.

3.2.4 Clustering

1. **K-means** K-means clustering is a key unsupervised machine learning algorithm that partitions a dataset into K clusters based on similarities ("Prediction of global solar irradiation using hybridized k-means and support vector regression algorithms", 2019).

It starts by randomly selecting K cluster centroids and assigns each data point to the nearest centroid. The centroids are then updated by computing the average position of data points within each cluster, and this process repeats iteratively until convergence (“Prediction of global solar irradiation using hybridized k-means and support vector regression algorithms”, 2019). K-means is known for its computational efficiency and scalability, making it widely used in various applications (Sinaga & Yang, 2020). However, selecting the appropriate value for K can be challenging, often requiring domain knowledge or techniques like the elbow method. Despite assuming spherical, equal-sized, and non-overlapping clusters, K-means remains a powerful tool for uncovering patterns in data across diverse domains, such as marketing, image analysis, and natural language processing.

The equation for K-means clustering is represented as (Sinaga & Yang, 2020):

$$\text{Distance Calculation: } d(\text{point}_i, \text{cen}_z) = \sqrt{\sum_{k=1}^{n} (\text{point}_{ik} - \text{cen}_{zk})^2} \tag{3.5}$$

$$\text{Centroid Update: } \text{cen}'_z = \frac{1}{|\text{Set}_z|} \sum_{\text{point}_i \in \text{Set}_z} \text{point}_i \tag{3.6}$$

Where:

$$
\begin{aligned}
n &: \text{the number of features or dimensions in the data} \\
\text{point}_i &: \text{a data point} \\
\text{cen}_z &: \text{a cluster centroid} \\
\text{Set}_z &: \text{the set of data points assigned to cluster } z \\
|\text{Set}_z| &: \text{the number of data points in cluster } z
\end{aligned}
$$

2. **MiniBatchKMeans** MiniBatchKMeans serves as a clustering algorithm for segregating a dataset into distinct clusters based on the similarity between data points (Chavan, Patil, Dalvi, & Patil, 2015). It represents a modification of the conventional KMeans clustering method, specifically tailored to efficiently manage sizable datasets. In contrast to the standard KMeans, which process the entire dataset during each iteration, MiniBatchKMeans operates on randomly selected subsets or mini-batches of the data (Peng, Leung, & Huang, 2018). This approach enhances computational efficiency and enables the algorithm to handle large datasets that may exceed available memory constraints. The algorithm iteratively updates cluster centroids with each mini-batch, progressively converging towards a solution that accurately reflects the inherent structure of the data (Peng et al., 2018).

3. **Fuzzy C-Means** Fuzzy C-Means (FCM) clustering is a mathematical approach employed in data analysis and pattern recognition to segment a dataset into clusters based on similarity (Bezdek, Ehrlich, & Full, 1984). Unlike conventional crisp or "hard" clustering techniques, FCM introduces the concept of degrees of membership, allowing each data point to be associated with multiple clusters to varying extents (Bezdek et al., 1984). In FCM, data points are assigned membership values for each cluster, reflecting the degree to which the point belongs to that cluster (Cannon, Dave, & Bezdek, 1986). These membership values are expressed as fuzzy sets, offering

greater flexibility compared to traditional clustering methods. The algorithm iteratively adjusts cluster centers and membership values until convergence, striving to minimize an objective function that measures the overall fuzziness of the clustering arrangement (Cannon et al., 1986).

4. **Gaussian Mixture Model** Gaussian Mixture Model (GMM) clustering is a probabilistic approach for clustering and density estimation (Weber, Ray, Valverde, Clark, & Sharma, 2022). Unlike traditional methods, GMM views a dataset as a mixture of Gaussian distributions, each associated with a cluster. The model aims to estimate parameters like mean, covariance, and mixing coefficients for these distributions (Weber et al., 2022). The Expectation-Maximization (EM) algorithm is often used for fitting a GMM by iteratively computing the likelihood of data points belonging to clusters and updating distribution parameters. GMM is valuable for complex datasets with unclear cluster boundaries or overlapping clusters (Weber et al., 2022). It finds applications in diverse fields, such as image processing and speech recognition, where capturing the underlying probability distribution is essential.

3.3 Phase 1: Machine Learning and Deep Learning Techniques for Solar Radiation Prediction

3.3.1 Ensemble Machine Learning Techniques

Ensemble machine learning strategies harness the collective strength of multiple models to enhance predictive accuracy and resilience. By integrating diverse algorithms or subsets of training data, ensembles address individual model limitations and capture a more comprehensive understanding of the data patterns. Ensemble approaches have become indispensable across various sectors, such as finance and healthcare, where precision and dependability are paramount. In this study, we have used 5 ensemble machine-learning techniques.

1. **Averaging** An averaging ensemble can be likened to the harmonious convergence of knowledge within the domain of machine learning. Picture a council composed of a multitude of diverse experts, each bringing their unique perspectives to the table, united in their effort to make a collective and more well-informed decision. In this scenario, these experts are individual machine learning models, and their decision-making process involves consolidating their viewpoints through averaging to reach a final prediction that is more precise (Polikar, 2012). In our approach, we utilize a combination of five distinct models for Averaging Ensemble Machine-Learning Technique. These models include Decision Tree Regressor M_1, Linear RegressionM_2, Ridge RegressionM_3, Lasso RegressionM_4, and ElasticNet Regression M_5 and we have also calculated the weighted average of these five methods.

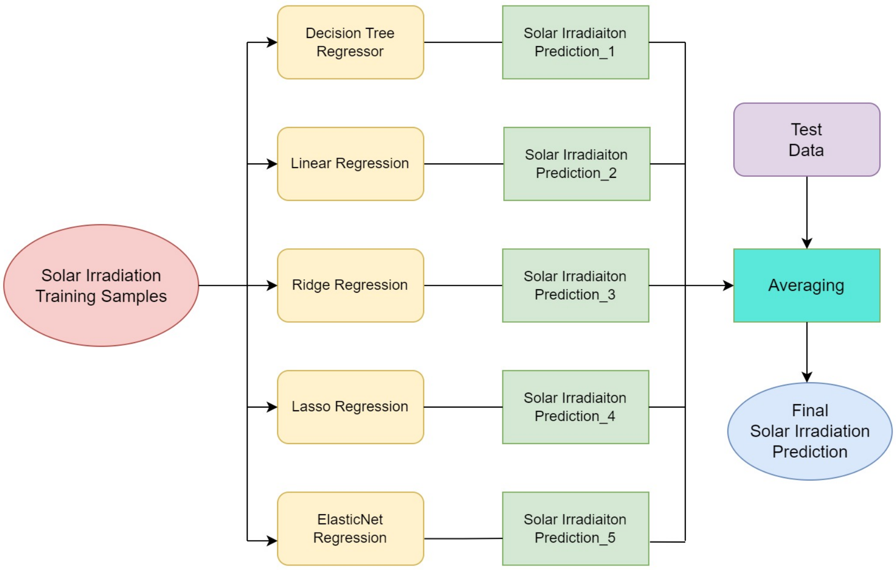


Figure 3.5: Averaging Ensemble Learning for Solar Irradiation Prediction

Figure 3.5 depicts the averaging ensemble learning for solar irradiation prediction. Algorithm 1 represents pseudo code for solar irradiance prediction using Averaging Ensemble Machine-Learning Technique.

Algorithm 1: : Solar Irradiance Prediction using Averaging Ensemble Machine-Learning Technique

Step 1: Load the dataset of Solar Radiation, *df*
Step 2: Perform Exploratory Data Analysis on the dataset with feature values denoted as $F = f_1, f_2, f_3, f_4, ..., f_n$
Step 3: Apply feature selection and split the dataset into df_train and df_test
Step 4: Training the train dataset df_train with different machine-learning models, $M_1, M_2, ..., M_k$
Step 5: Averaging the predictions, $pred_{final} = (pred_1 + pred_2 +, ..., +pred_k)/k$.
Step 6: set the weighted $weights = w1, w2, w3, w4, w5$
Step 7: Calculate the weighted average of predictions
Step 8: Calculate and store three different error values using $pred_{final}$ and df_test, E_{R^2}, E_{MAE} and E_{RMSE}.

2. **Bagging** Bagging is a powerful ensemble machine learning technique that combines ”bootstrap” and ”aggregating” to improve predictive models by leveraging diversity (Dietterich et al., 2002). Its key strength is mitigating variance by training multiple instances of a base model on different subsets of training data with replacement. This introduces heterogeneity and prevents fixation on training data idiosyncrasies and noise. During prediction, these independently trained models contribute their opinions, resulting in a more precise and reliable forecast than a single model could provide. Bagging democratizes the influence of individual models, reducing the risk of overfitting and enhancing generalization to new data (Dietterich et al., 2002).

   Bagging is valued for its simplicity and efficiency and is a fundamental tool in ensemble learning, with applications ranging from random forests, a collection of bagged decision trees, to other models benefiting from its stabilizing and predictive potential. Despite its unassuming name, bagging elevates the predictive capabilities of machine learning models. Figure 3.6 depicts the Bagging Ensemble Learning for Solar Irradiation Prediction.

   In our approach, we employ five different models as the base estimators for bagging: decision tree regression, linear regression, ridge regression, Lasso regression, and ElasticNet regression. Algorithm 2 represents pseudo code for solar irradiance

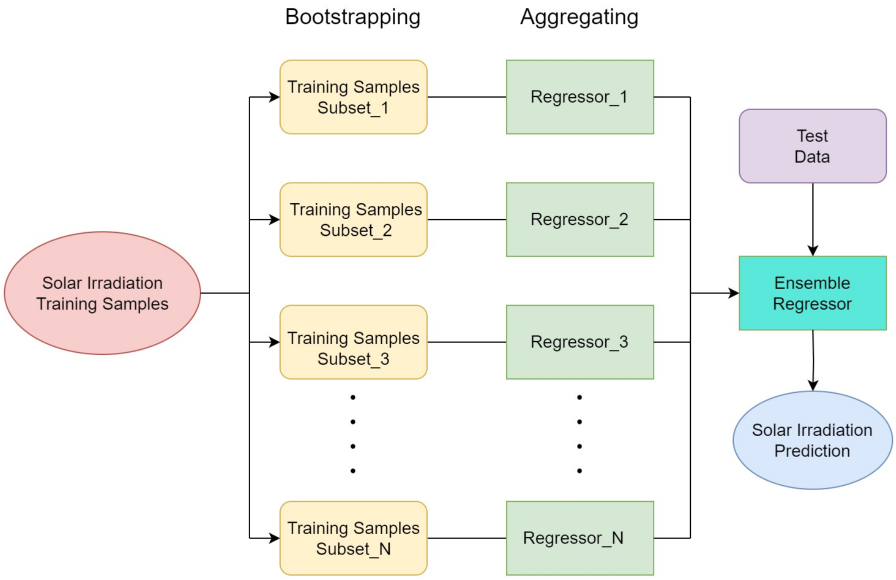


Figure 3.6: Bagging Ensemble Learning for Solar Irradiation Prediction

prediction using Bagging Regressor.

Algorithm 2: : Solar Irradiance Prediction using Bagging Regressor

---

Step 1: Load the dataset of Solar Radiation, *df*
Step 2: Perform Exploratory Data Analysis on the dataset with feature values denoted as $F = f_1, f_2, f_3, f_4, ..., f_n$
Step 3: Apply feature selection and split the dataset into df_train and df_test
Step 4: Training the train dataset df_train with different base models for $base\_estimator$ used in the bagging approach, $M_1, M_2, ..., M_k$
Step 5: Calculating errors using df_test for $M_1, M_2, ..., M_k$ models, $pred_1, pred_2, ..., pred_k$.
Step 6: Store three different error values, E_{R^2}, E_{MAE} and E_{RMSE} for each model.

Algorithm 3 presents pseudo code for forecasting solar irradiance utilizing a Random Forest Regressor.

3. **Boosting** Boosting is an ensemble machine learning technique that combines predictions from multiple weak learners to create a robust and accurate predictive model (Sagi & Rokach, 2018). Unlike traditional algorithms, boosting trains weak learners sequentially, with each learner focused on correcting the errors of its predecessors. In each iteration, boosting assigns greater importance to misclassified data points,

Algorithm 3: : Solar Irradiance Prediction using Random Forest Regressor

Step 1: Load the dataset of Solar Radiation, *df*
Step 2: Perform Exploratory Data Analysis on the dataset with feature values denoted as $F = f_1, f_2, f_3, f_4, \dots, f_n$
Step 3: Apply feature selection and split the dataset into df_train and df_test
Step 4: Training the train dataset df_train with Random Forest Regressor, *RF*
Step 5: Calculating errors using df_test.
Step 6: Store three different error values, E_{R^2}, E_{MAE} and E_{RMSE}.

compelling subsequent learners to pay more attention to challenging instances. This iterative process efficiently captures complex patterns in the data. Boosting utilizes a weighted voting system in the final model, where each weak learner contributes to predictions based on their precision, effectively elevating them to strong predictors. (Sagi & Rokach, 2018).

Figure 3.7 illustrates the Boosting Ensemble Learning method for Solar Irradiation Prediction and and Algorithm 4 represents pseudo code for solar irradiance prediction using Boosting Regressor.

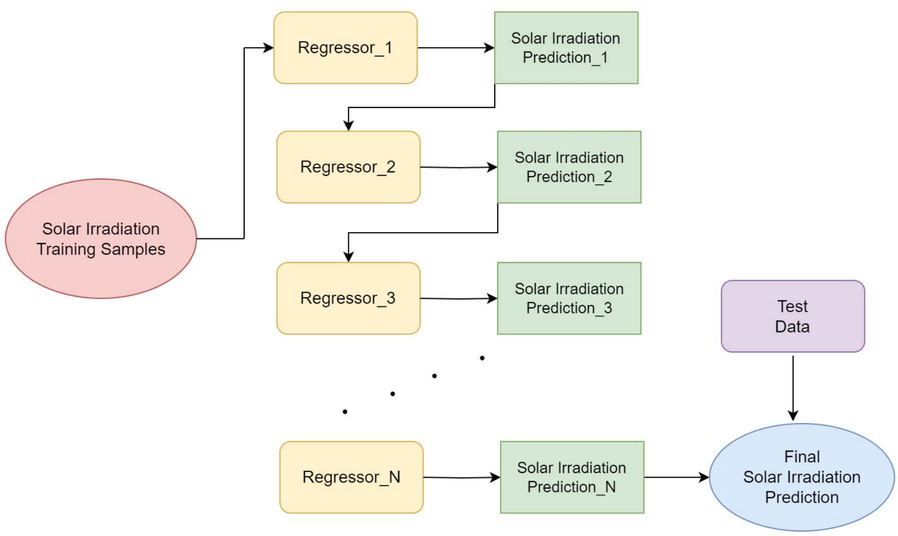


Figure 3.7: Boosting Ensemble Learning for Solar Irradiation Prediction

Algorithm 4: : Solar Irradiance Prediction using Boosting Regressor

Step 1: Load the dataset of Solar Radiation, *df*
Step 2: Perform Exploratory Data Analysis on the dataset with feature values denoted as $F = f_1, f_2, f_3, f_4, ..., f_n$
Step 3: Apply feature selection and split the dataset into df_train and df_test
Step 4: Training the train dataset df_train with different models for the Boosting approach, $M_1, M_2, ..., M_k$
Step 5: Calculating errors using df_test for $M_1, M_2, ..., M_k$ models, $pred_1, pred_2, ..., pred_k$.
Step 6: Store three different error values, E_{R^2}, E_{MAE} and E_{RMSE} for each model.

4. **Stacking** Stacking stands out as an advanced ensemble learning method that fuses forecasts from a variety of base models through the use of a meta-model (Pavlyshenko, 2018). This amalgamation leverages their strengths and mitigates their shortcomings. Unlike conventional ensemble techniques, Stacking doesn't simply take an average of predictions; instead, it learns how to assign weights to inputs from various models wisely. It assembles a diverse group of models, including decision trees, support vector machines, and neural networks, each adept at capturing distinctive data patterns (Pavlyshenko, 2018). These varied viewpoints are harmonized by the meta-model, resulting in a well-balanced and precise predictive system.

   Stacking enables the utilization of specialized models tailored to different sets of features, ensuring that no valuable information is left unexplored (Kwon, Park, & Lee, 2019). However, meticulous model selection and fine-tuning are imperative for its success, resembling the process of assembling an impeccable ensemble cast for a movie. Ultimately, stacking exemplifies the idea that the collaborative intelligence of diverse models can surpass individual endeavors, rendering it a potent tool in the realm of machine learning.

   In our methodology, we employ a fusion of five models to implement the Stacking Ensemble Machine Learning approach. These models comprise the Decision Tree Regressor, Linear Regression, Ridge Regression, Lasso Regression, and ElasticNet Regression. Finally, we utilize these five models as the meta-model and find the best model for Stacking. Figure 3.8 illustrates the Stacking Ensemble Learning approach for predicting Solar Irradiation. Algorithm 5 represents pseudo code for solar irradiance prediction using the Stacking Ensemble Machine-Learning Technique.

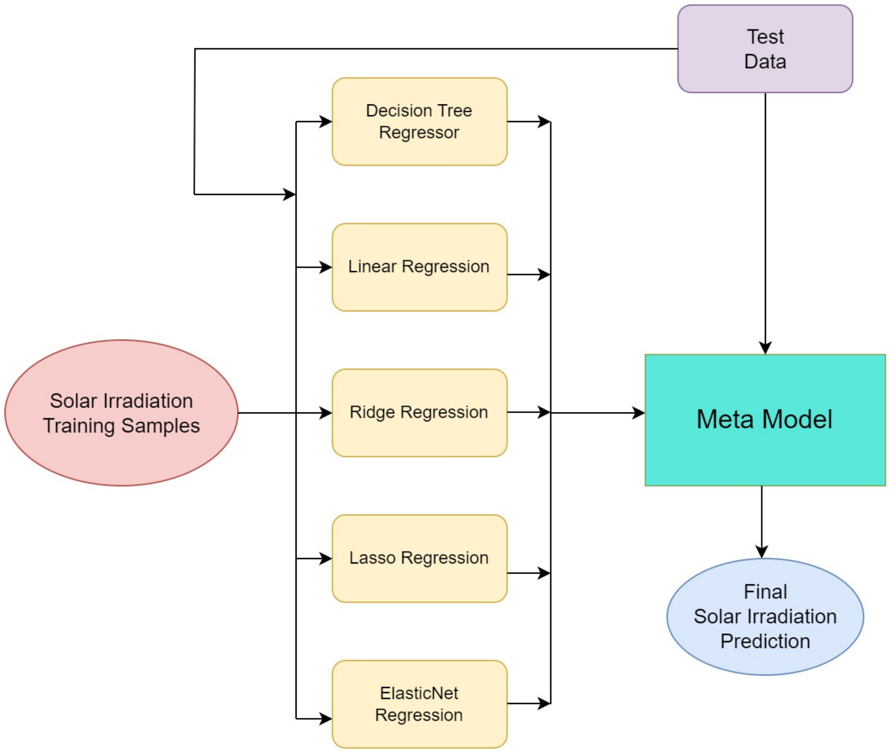


Figure 3.8: Stacking Ensemble Learning for Solar Irradiation Prediction

---

Algorithm 5: : Solar Irradiance Prediction using Stacking Ensemble Machine-Learning Technique

---

Step 1: Load the dataset of Solar Radiation, *df*

Step 2: Perform Exploratory Data Analysis on the dataset with feature values denoted as $F = f_1, f_2, f_3, f_4, ..., f_n$

Step 3: Apply feature selection and split the dataset into df_train and df_test

Step 4: Selecting Meta Model, M_{meta} for $final\_estimator$.

Step 5: Training the train dataset df_train with different machine-learning models, $M_1, M_2, ..., M_k$ with Meta Model, M_{meta}.

Step 6: Calculate and store three different error values using $pred_{final}$ and df_test, E_{R^2}, E_{MAE} and E_{RMSE}.

---

5. **Blending** Blending ensemble regression is a sophisticated technique in machine learning used to improve prediction accuracy in regression tasks. It begins by splitting the training data into a training set and a validation set, where multiple regression models, such as linear regression and decision trees, are trained on the training data. These models generate predictions for both the validation and test sets, and the actual target values of the validation set are retained. A meta-model, often a regression model like linear regression or ridge regression, is then trained using the validation set's true target values and the predictions from the base models on the validation set. The meta-model learns how to optimally combine the base models' predictions by determining the best coefficients (Wu, Zhang, Jiao, Guo, & Hamoud, 2021).
   Once the meta-model is trained, it is used to make the final predictions on the test set, leveraging the strengths of the base models to provide more accurate and robust results for continuous target variables. This approach is particularly valuable for handling complex and noisy datasets and helps improve model generalization while mitigating overfitting (Wu et al., 2021). Blending ensemble regression offers a powerful means to enhance predictive accuracy by drawing upon the collective insights of diverse regression models, resulting in more reliable and accurate predictions in regression tasks.

   In our method, we employ a blend of five unique models to create a Blending Ensemble Machine-Learning Technique. These models comprise the Decision Tree Regressor, Linear Regression, Ridge Regression, Lasso Regression, and ElasticNet Regression. Finally, we utilize these five models as the Meta Model and find the best model for Blending. Figure 3.9 demonstrates the Blending Ensemble Learning method for Solar Irradiation Prediction. Algorithm 6 represents pseudo code for solar irradiance prediction using the Blending Ensemble Machine-Learning Technique.

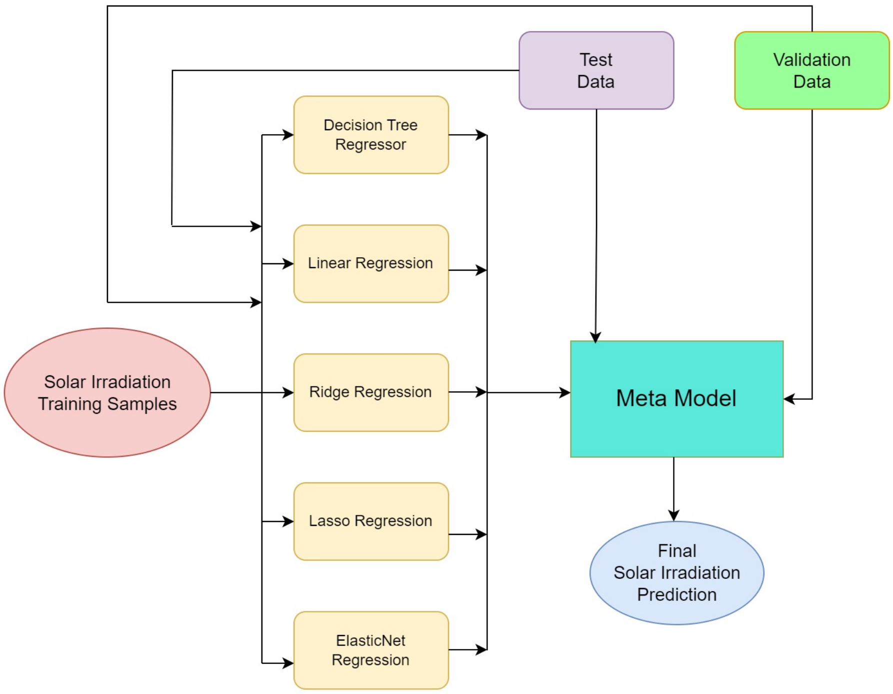


Figure 3.9: Blending Ensemble Learning for Solar Irradiation Prediction

Algorithm 6: : Solar Irradiance Prediction using Blending Ensemble Machine-Learning Technique

Step 1: Load the dataset of Solar Radiation, *df*
Step 2: Perform Exploratory Data Analysis on the dataset with feature values denoted as $F = f_1, f_2, f_3, f_4, ..., f_n$
Step 3: Apply feature selection and split the dataset into df_train , df_valid, df_test.
Step 4: Training the train dataset df_train with different machine-learning models, $M_1, M_2, ..., M_k$.
Step 5: Make predictions on df_valid and df_test for each model and combine their results in com_valid and com_test.
Step 6: Selecting Meta Model, M_{meta}.
Step 7: Make predictions on com_valid and com_test using M_{meta}.
Step 8: Calculate and store three different error values using $pred_{final}$ and df_test, E_{R^2}, E_{MAE} and E_{RMSE}.

3.4 Phase 2: Explainable AI in Solar Radiation Prediction

1. Local and Global Importance

Global Importance with SHAP: It bases its explanations on shapely values, which are measures of how much each feature contributes to the model. In our previous work (Sevas, Tur Santona, & Sharmin, 2023), we displayed the summary plot of the aforementioned features using XGboost model with SHAP where the contribution of each feature was displayed. The temperature showed the highest contribution around 700, and the wind direction showed the second nearly 400 in our previous work with the same dataset and features with the SHAP summary plot. In this analysis, we have shown the SHAP explanation of the summary plot of decision tree and MLP in figure 3.13, waterfall plot in Figure 3.14 and 3.15, the force plot in Figure 3.16, and the dependency plot in Figure 3.10, 3.11, 3.12 of XGBoost, Decision Tree and MLP model.

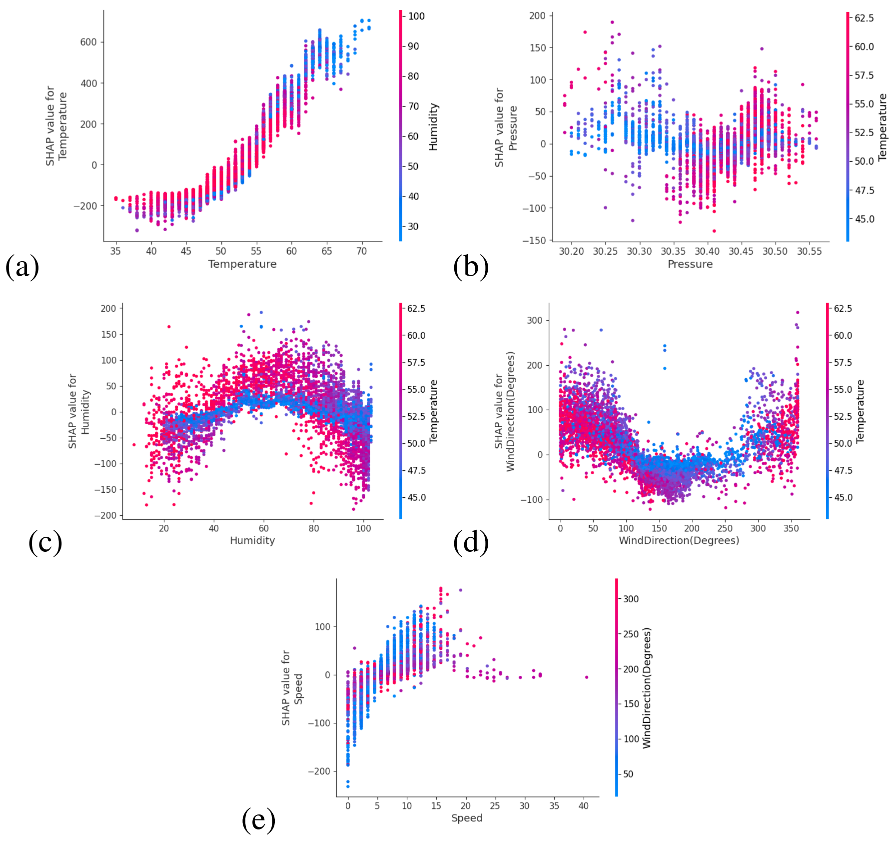


Figure 3.10: SHAP feature dependence plot using XGboost model with interaction visualization (a) temperature, (b) pressure, (c) humidity, (d) wind direction, and (e) wind speed

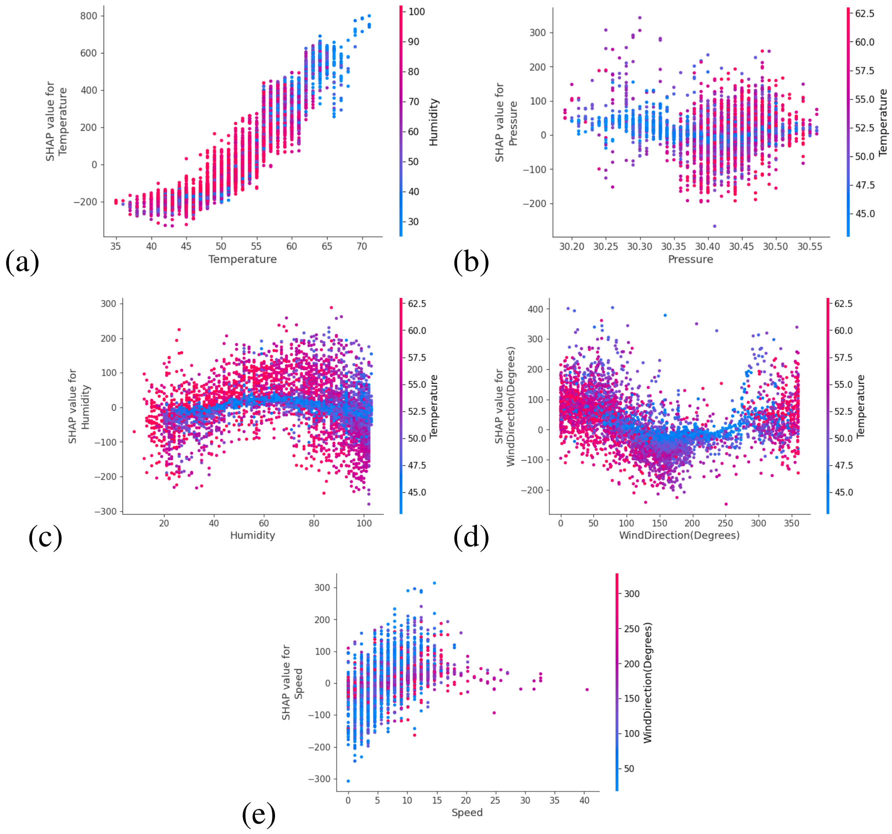


Figure 3.11: SHAP feature dependence plot using Decision Tree with interaction visualization (a) temperature, (b) pressure, (c) humidity, (d) wind direction, and (e) wind speed

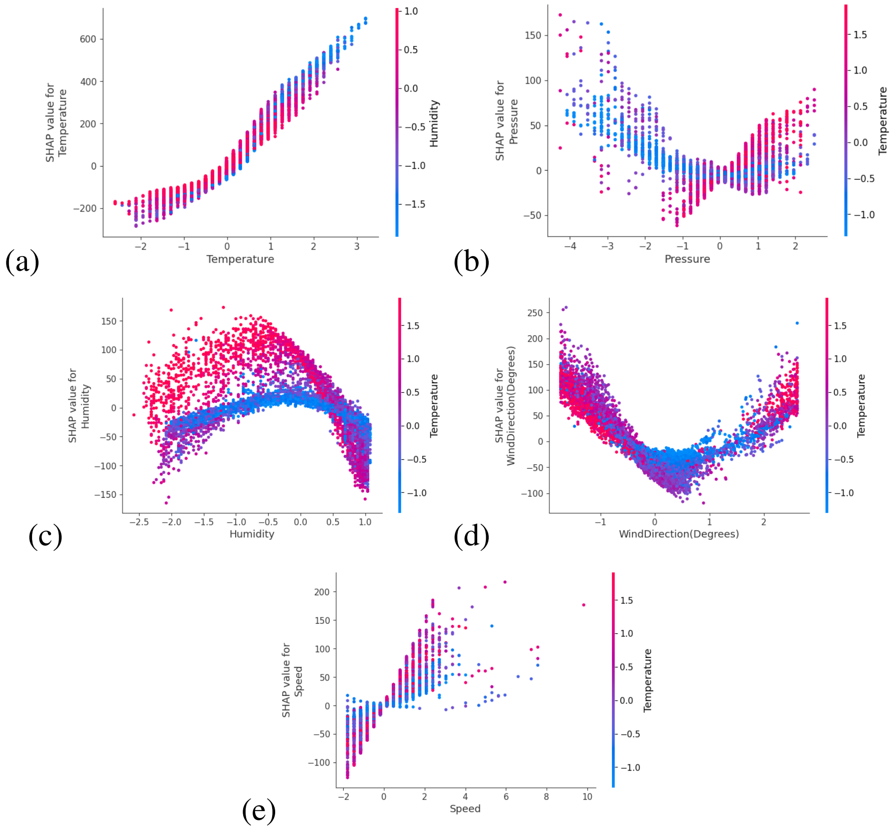


Figure 3.12: SHAP feature dependence plot using MLP model with interaction visualization (a) temperature, (b) pressure, (c) humidity, (d) wind direction, and (e) wind speed

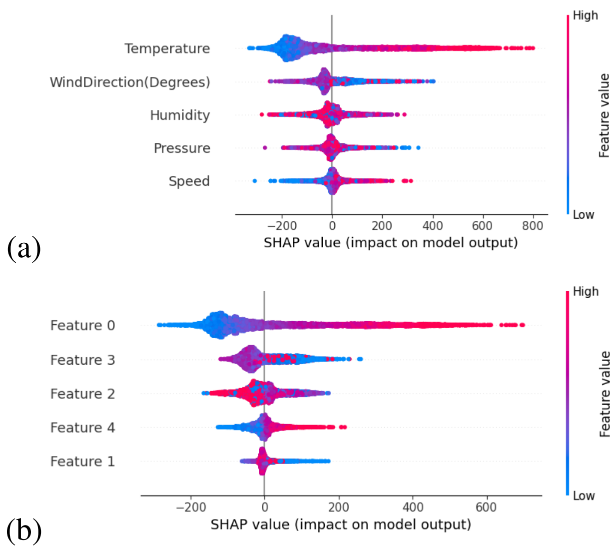


Figure 3.13: SHAP explanation of the summary plot using (a) Decision Tree (b) MLP model.

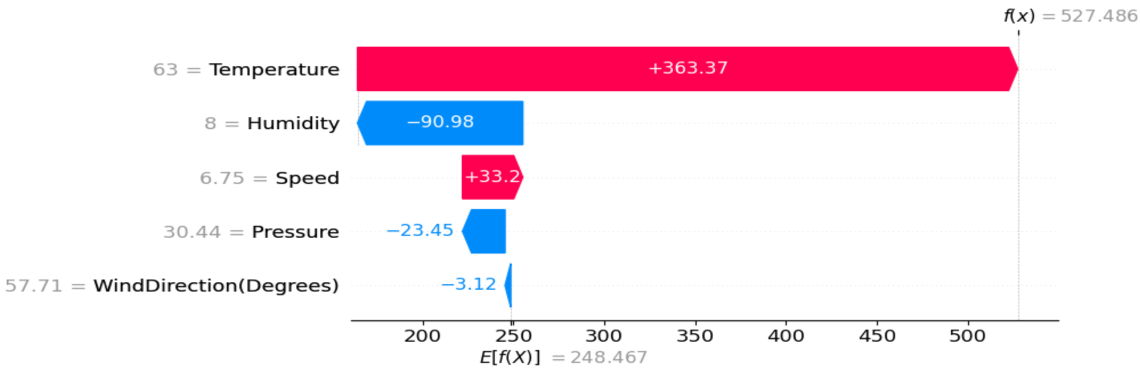


Figure 3.14: SHAP explanation of the waterfall plot with the boundary value(minimum) of humidity using XGBoost model.

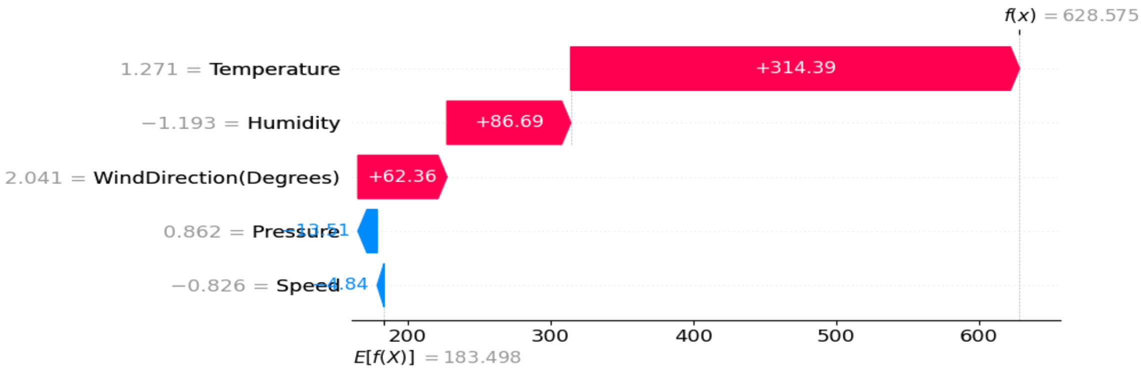


Figure 3.15: SHAP explanation of the waterfall plot for 0-th instance using MLP model.

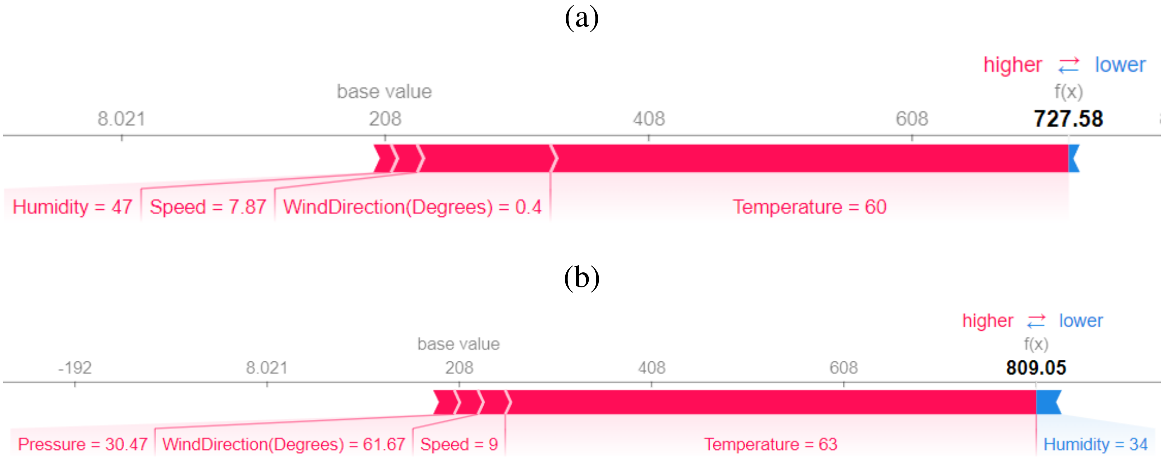


Figure 3.16: SHAP explanation of (a) 100-th instance, (b) 150-th instance using XGboost model.

In the waterfall plot of figure 3.14, we have taken the minimum boundary value of $H_{umidity}$ and see how each feature contributed to the prediction. The features are arranged in descending order based on their absolute SHAP values, highlighting the most influential features at the top. Positive contributions are represented by bars extending to the right, while negative contributions extend to the left. In the force plot of figure 3.16, we have used the 100-th and 150-th instances from the test dataset where the features are displayed vertically along the y-axis, and the SHAP values are represented by horizontal bars. The magnitude and direction of the feature's impact are shown by the length and direction of each bar. The color of the bars represents the feature's value (e.g., red for high values, blue for low values). We predicted 727.58 and 809.05 consecutively where the actual values were 671.97 and 1009.36 consecutively. Temperature shows similar behavior in both cases. The force plot allowed us to understand the individual contributions of each feature and how they collectively influence the prediction. In the dependency plot of figure 3.10, the shape of the plot provides insights into how the features impact the prediction showing whether the relationship is linear, monotonic or exhibits more complex patterns. From this figure, it is visible that all the other features interact with the temperature. While temperature itself interacts with humidity.

According to these three plots, the value of temperature plays the most important role in predicting solar radiation. From these SHAP explanations, it is clear that SHAP not only gives global interpretability but also gives local interpretability with a deeper understanding of the model's behavior. By visualizing the different plots, it can be seen that SHAP can provide insights into the overall feature importance and relationship across the entire dataset.

Local Importance with LIME: It determines an interpretable model that can explain the behavior of a machine learning model locally. After applying LIME to explain individual predictions, it displays the prediction along with the contribution of each feature. Figure 3.17 represents the LIME explanation of feature contribution on solar radiation prediction using three models. According to this numerical contribution of each feature, T_{temp}, $H_{umidity}$ and W_{dir}, W_{spd} has a positive contribution which means higher of these values will result in a higher likelihood of the desired outcome. On the other hand, $P_{ressure}$ shows a negative contribution to the prediction. Among all the six features T_{temp} has the higher positive contribution and $P_{ressure}$ has the higher negative contribution to the prediction which indicates, with influences of all six features, $P_{ressure}$ has a stronger impact and influence on the prediction. The prediction probability for the model is 715.52 including all six features.

As LIME is capable of local interpretability of models also, local interpretation of one instance has been shown in Figure 3.18, 3.19 and 3.20 which is obtained through LIME `explain_instance()` function. In this figure, the leftmost bar chart shows the probability of solar radiation prediction, the middle bar chart shows the lime explanation of the selected features and the rightmost table shows the original feature values. According to the results, the predicted value is 715.52 and the actual value is 675.86 for the first instance. It can also be extended with more instances. Here we have used the first instance of test datasets.

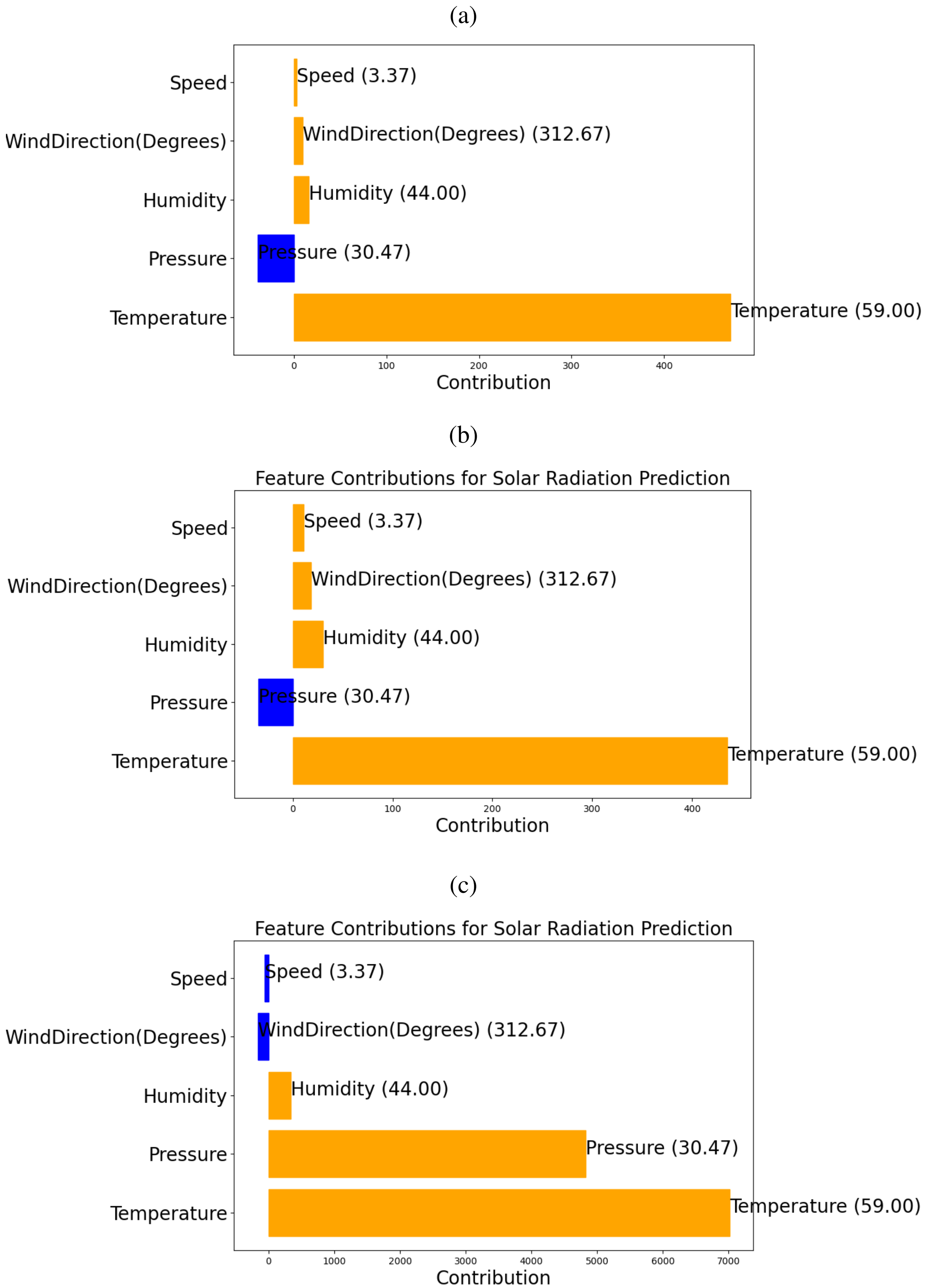


Figure 3.17: LIME explanation of feature contribution for solar radiation prediction with all features using (a) XGboost model (b) Decision tree (c) MLP.

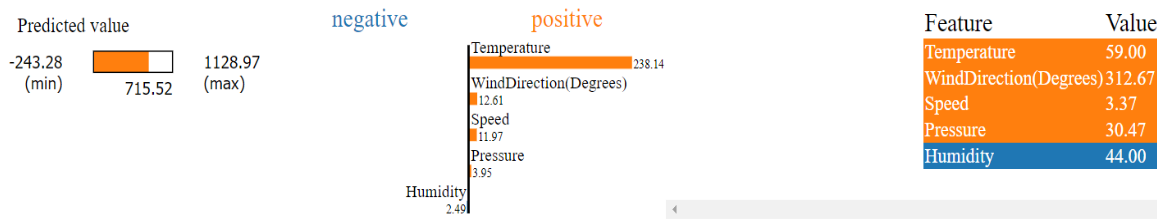


Figure 3.18: LIME explanation of the first instance from the test dataset where actual: 675.86, predicted: 715.52 using XGboost Model.

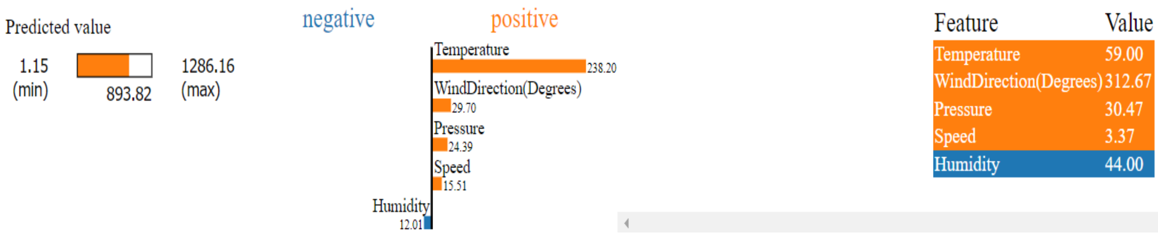


Figure 3.19: LIME explanation of the first instance from the test dataset where actual: 675.86, predicted: 893.82 using Decision Tree.

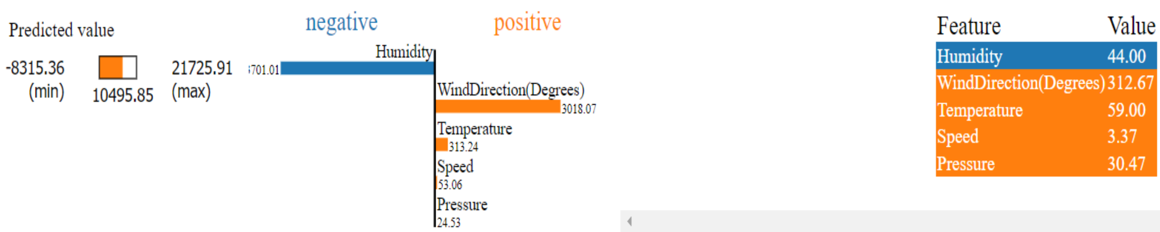


Figure 3.20: LIME explanation of the first instance from the test dataset where actual: 675.86, predicted: 10495.84 using MLP Model.

2. **Permutation Importance**

ELI5: Firstly, the model is trained and tested using the six of its features that we mentioned in section III. Secondly, the model is again trained and tested from scratch by removing one of its features i.e. W_{spd} to analyze the effect of removing some features on it. Then ELI5 is applied on both steps separately. Figure 3.21, 3.22**??** and 3.23 shows the feature weights obtained with all its features using ELI5 `show_weights()` function.

To get a better idea of the prediction, we run `eli5.explain_prediction()` function to see how the model works with all of its six features. The forecasting shows the sum of the feature contribution in addition to the ”BIAS” as shown in Figure 3.21, 3.22**??**. It displays the actual values of the input features X_test.iloc[0] for the given data point. Figure 3.24 shows the ELI5 explanation of feature importance on the prediction and their feature importance score. It can be seen that the RMSE score increases noticeably when applied to the subset of features. According to the results of both cases, T_{temp} has the highest weight. In both cases, humidity has given the second highest weight after temperature. It is clear that temperature has the highest impact on the prediction of the model and among the six parameters humidity gives the second highest weight after temperature.

(a)

Weight	Feature
1.0388 ± 0.0148	Temperature
0.1549 ± 0.0101	Humidity
0.1313 ± 0.0096	WindDirection(Degrees)
0.0686 ± 0.0032	Pressure
0.0606 ± 0.0032	Speed

(b)

Contribution?	Feature
+365.250	Temperature
+207.521	<BIAS>
+119.580	Humidity
+32.287	Pressure
+25.081	WindDirection(Degrees)
-34.696	Speed

Figure 3.21: With all six features, ELI5 explanation of feature (a) weights and (b) contribution on the prediction using XGBoost model.

(a)

Weight	Feature
1.1910 ± 0.0326	Temperature
0.3157 ± 0.0290	Humidity
0.2392 ± 0.0422	WindDirection(Degrees)
0.1599 ± 0.0210	Pressure
0.1004 ± 0.0236	Speed

(b)

Contribution?	Feature
+320.697	Temperature
+208.045	<BIAS>
+134.704	Humidity
+93.627	Pressure
+77.477	Speed
+59.270	WindDirection(Degrees)

Figure 3.22: With all six features, ELI5 explanation of feature (a) weights and (b) contribution on the prediction using Decision Tree.

Weight	Feature
0.8121 ± 0.0280	Temperature
0.1550 ± 0.0087	Humidity
0.1129 ± 0.0065	WindDirection(Degrees)
0.0587 ± 0.0100	Speed
0.0297 ± 0.0034	Pressure

Figure 3.23: With all six features, ELI5 explanation of feature weights on the prediction using MLP.

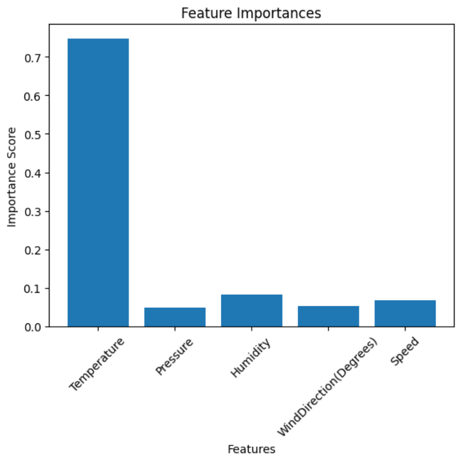


Figure 3.24: ELI5 explanation of feature importance on the prediction using XGBoost.

3. **Model Algoside**

ALE plots with Dalex: This offers global and local interpretation allowing us to understand the model's behavior at different levels of clarity. Dalex can give an overall model explanation of the whole structure of the trained model. In addition, it can show the functions and algorithms that build the model. The user can know the dataset that trained the model and how each variable in the dataset contributed to the model. Figure 3.25 shows the whole structure of the models and its explanation using the Explainer function and Figure 3.26 shows the model performance scores provided by Dalex using the model_perfromance() function. Dalex gives an explanation of the variable's importance along with drop-out loss and a full explanation of the model aspect importance and predicted aspect importance using the permutation-based variable importance method. Figure

3.27 shows the contribution of each feature on solar radiation prediction using the model_parts() function. Dalex also generates ceteris paribus profiles which explains how the model response depends on changes in a single input variable, keeping all other variables unchanged. Figure 3.29, 3.30 and 3.31 shows the Dalex explanation of the ceteris paribus profiles. Again, the partial dependence plot can also be visualized through Dalex from Figure 3.32, 3.33 and 3.34 using pdpplot().

Dalex also gives the local interpretation of the models same as LIME. Figure 3.28 shows the single interpretation of the 100-th instance through the predict_parts() function.

(a)

```
Preparation of a new explainer is initiated

  -> data              : 26148 rows 5 cols
  -> target variable   : Parameter 'y' was a pandas.Series. Converted to a numpy.ndarray.
  -> target variable   : 26148 values
  -> model_class       : xgboost.sklearn.XGBRegressor (default)
  -> label             : Not specified, model's class short name will be used. (default)
  -> predict function  : <function yhat_default at 0x7982ef9eeb90> will be used (default)
  -> predict function  : Accepts pandas.DataFrame and numpy.ndarray.
  -> predicted values  : min = -84.0, mean = 2.08e+02, max = 1.19e+03
  -> model type        : regression will be used (default)
  -> residual function : difference between y and yhat (default)
  -> residuals         : min = -7.45e+02, mean = 0.0235, max = 1.07e+03
  -> model_info        : package xgboost

A new explainer has been created!
```

(b)

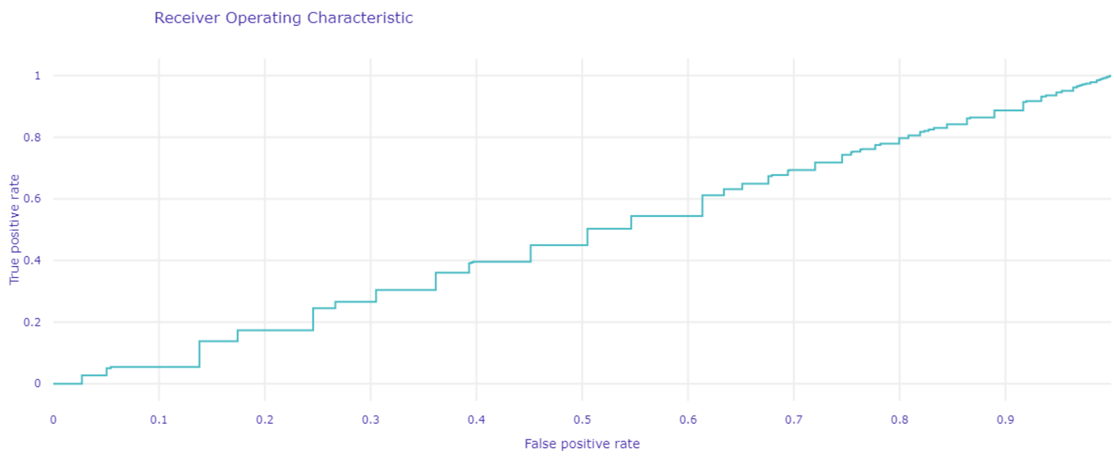
```
Preparation of a new explainer is initiated

  -> data              : 26148 rows 5 cols
  -> target variable   : Parameter 'y' was a pandas.Series. Converted to a numpy.ndarray.
  -> target variable   : 26148 values
  -> model_class       : sklearn.tree._classes.DecisionTreeRegressor (default)
  -> label             : Not specified, model's class short name will be used. (default)
  -> predict function  : <function yhat_default at 0x7c70b9f4a440> will be used (default)
  -> predict function  : Accepts pandas.DataFrame and numpy.ndarray.
  -> predicted values  : min = 1.13, mean = 2.08e+02, max = 1.6e+03
  -> model type        : regression will be used (default)
  -> residual function : difference between y and yhat (default)
  -> residuals         : min = -85.6, mean = 7.98e-19, max = 85.6
  -> model_info        : package sklearn

A new explainer has been created!
```

(c)

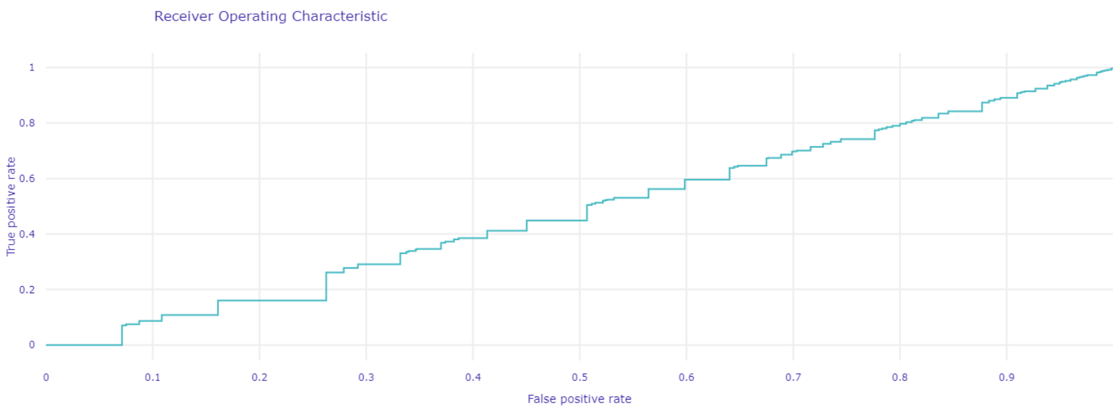
```
Preparation of a new explainer is initiated

  -> data              : numpy.ndarray converted to pandas.DataFrame. Columns are set as string numbers.
  -> data              : 26148 rows 5 cols
  -> target variable   : Parameter 'y' was a pandas.Series. Converted to a numpy.ndarray.
  -> target variable   : 26148 values
  -> model_class       : sklearn.neural_network._multilayer_perceptron.MLPRegressor (default)
  -> label             : Not specified, model's class short name will be used. (default)
  -> predict function  : <function yhat_default at 0x7b94f5117d00> will be used (default)
  -> predict function  : Accepts pandas.DataFrame and numpy.ndarray.
  -> predicted values  : min = -52.2, mean = 2.09e+02, max = 1.13e+03
  -> model type        : regression will be used (default)
  -> residual function : difference between y and yhat (default)
  -> residuals         : min = -7.92e+02, mean = -1.28, max = 1.27e+03
  -> model_info        : package sklearn

A new explainer has been created!
```

Figure 3.25: DALEX explanation of the whole structure of the trained model using Explainer function for (a) XGboost model (b) Decision tree (c) MLP model.

(a)



(b)



(c)

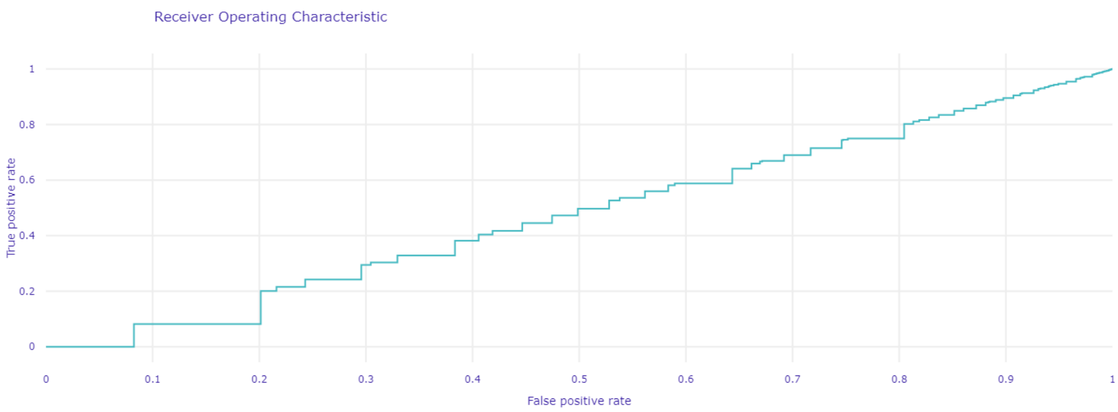
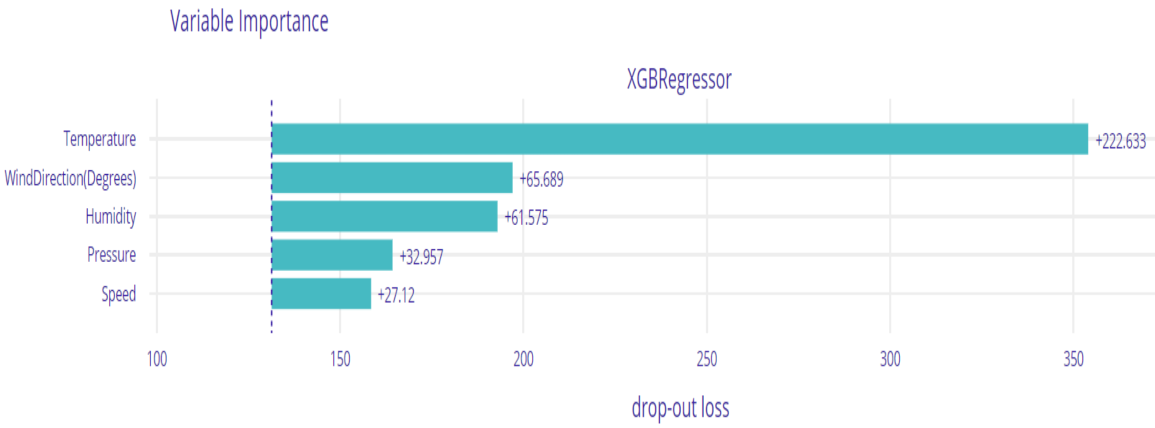
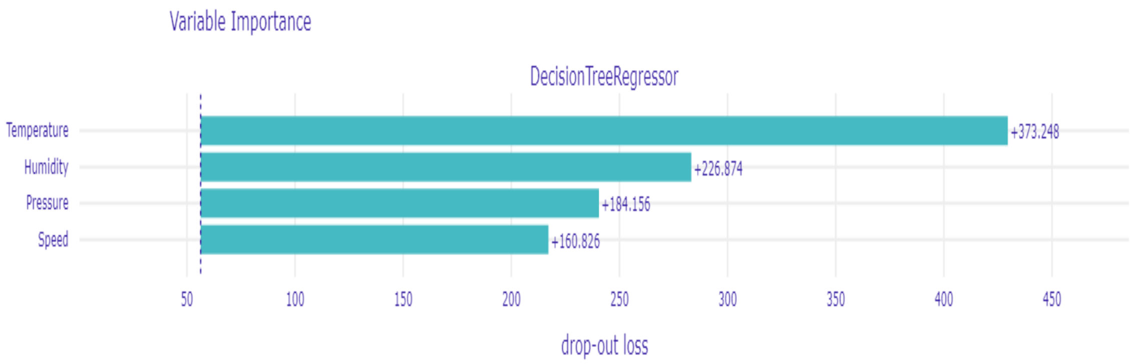


Figure 3.26: DALEX explanation of the model performance scores using (a) XGboost model (b) Decision tree (c) MLP model.

(a)



(b)



(c)

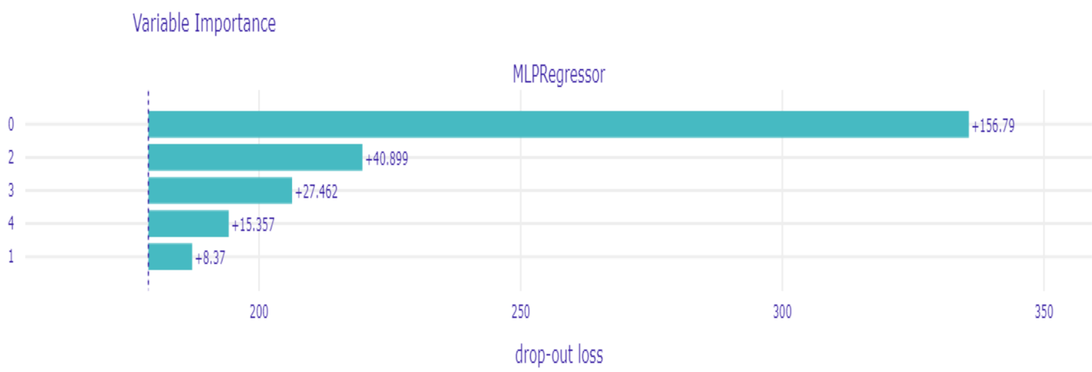


Figure 3.27: DALEX visualization of the contribution of each feature based on drop-out loss using (a) XGboost model (b) Decision tree (c) MLP model.

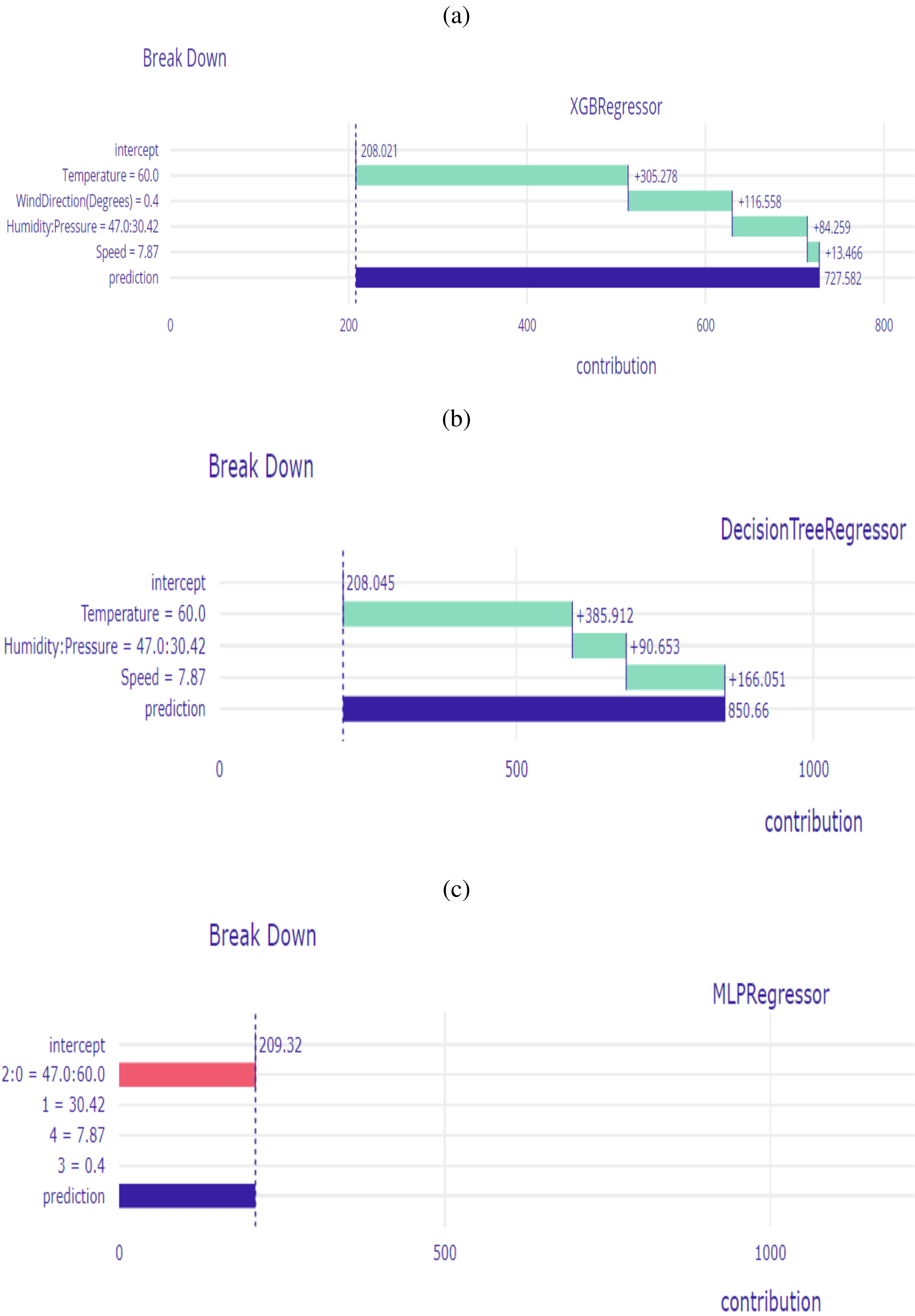


Figure 3.28: Feature importance values for 100-th instance of the input data using DALEX for breakdown. using (a) XGboost model (b) Decision tree (c) MLP model.

Figure 3.29: DALEX explanation of Ceteris Paribus profiles for XGBoost model.

Figure 3.30: DALEX explanation of Ceteris Paribus profiles for Decision tree.

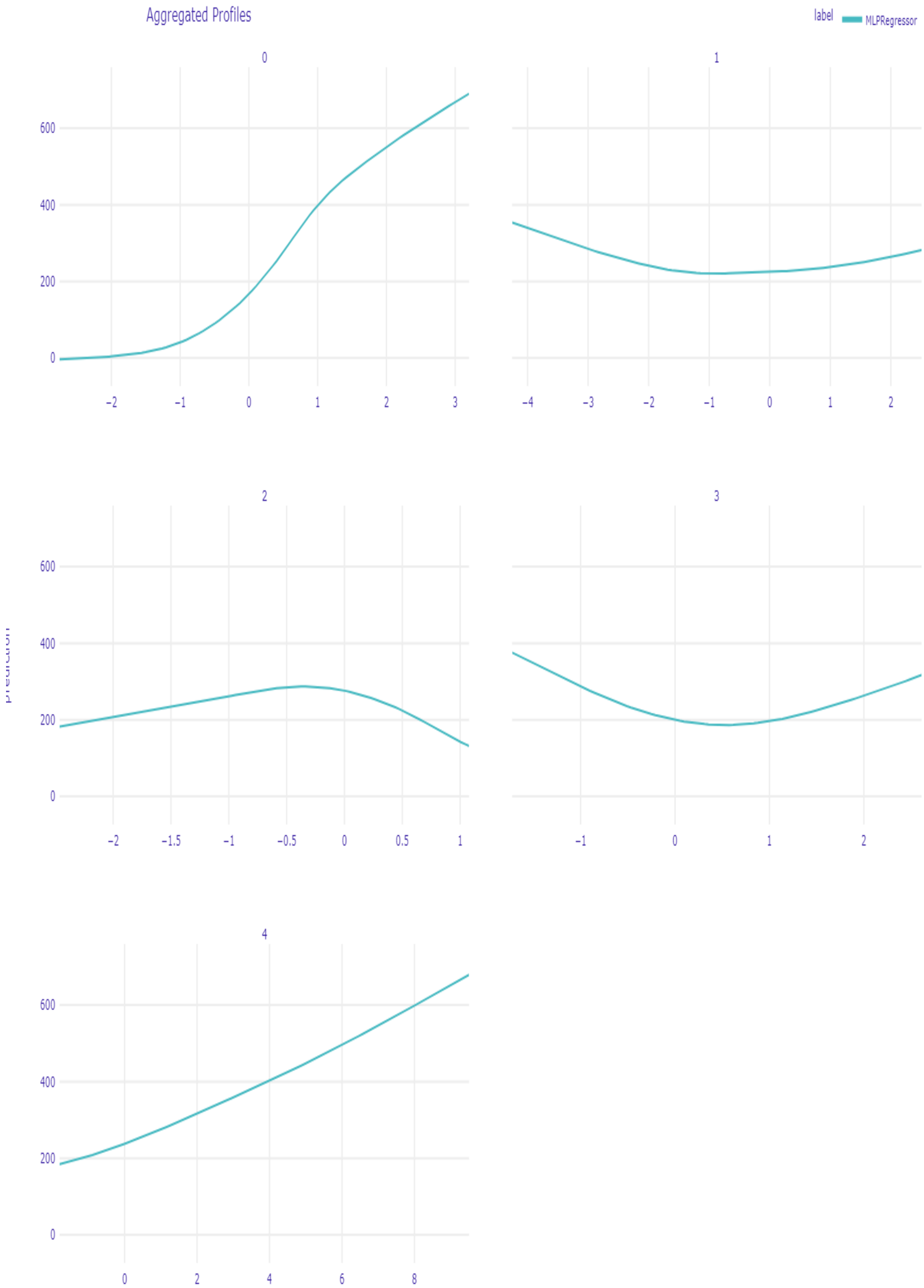


Figure 3.31: DALEX explanation of Ceteris Paribus profiles for MLP model.

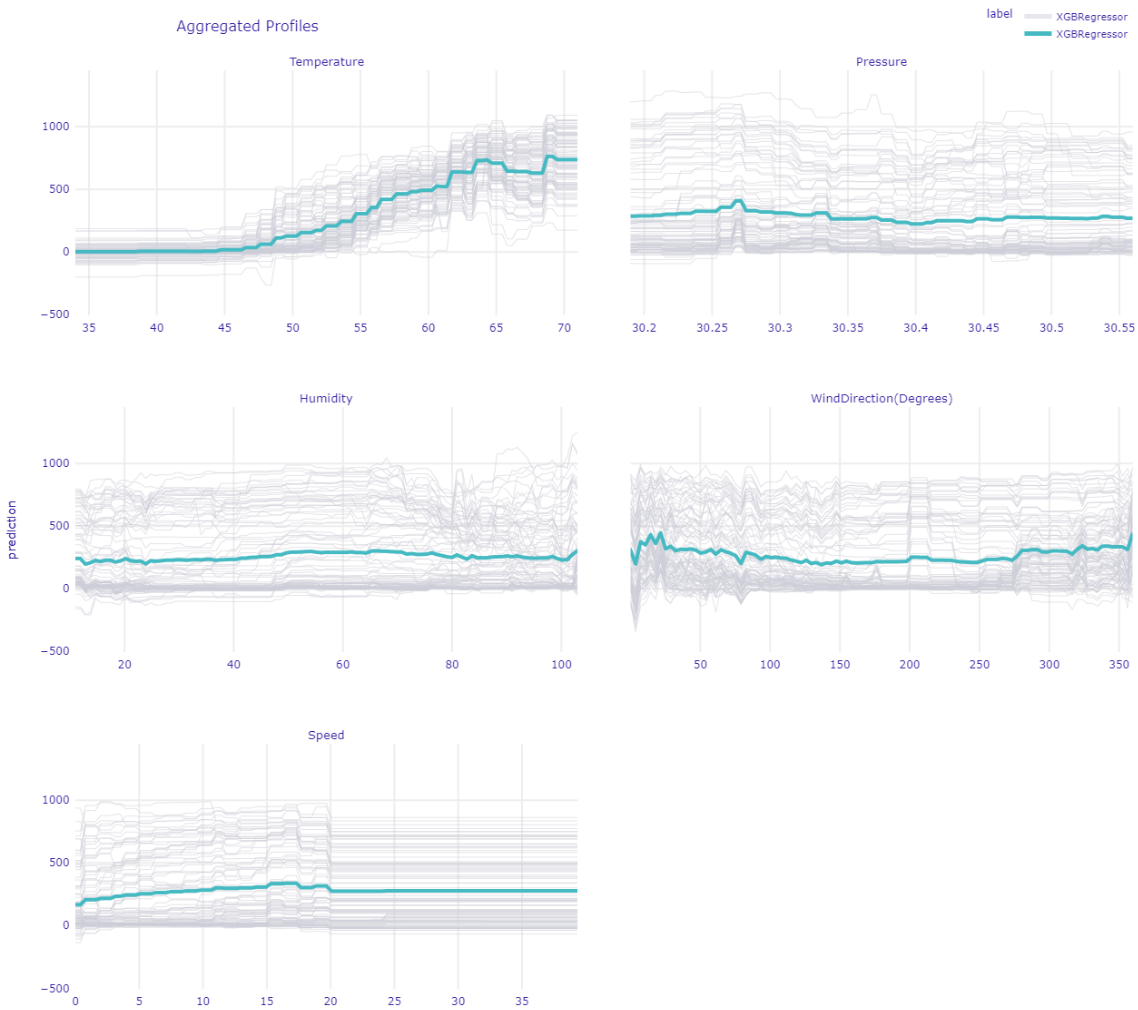


Figure 3.32: Partial dependence plots obtained from DALEX for XGBoost model.

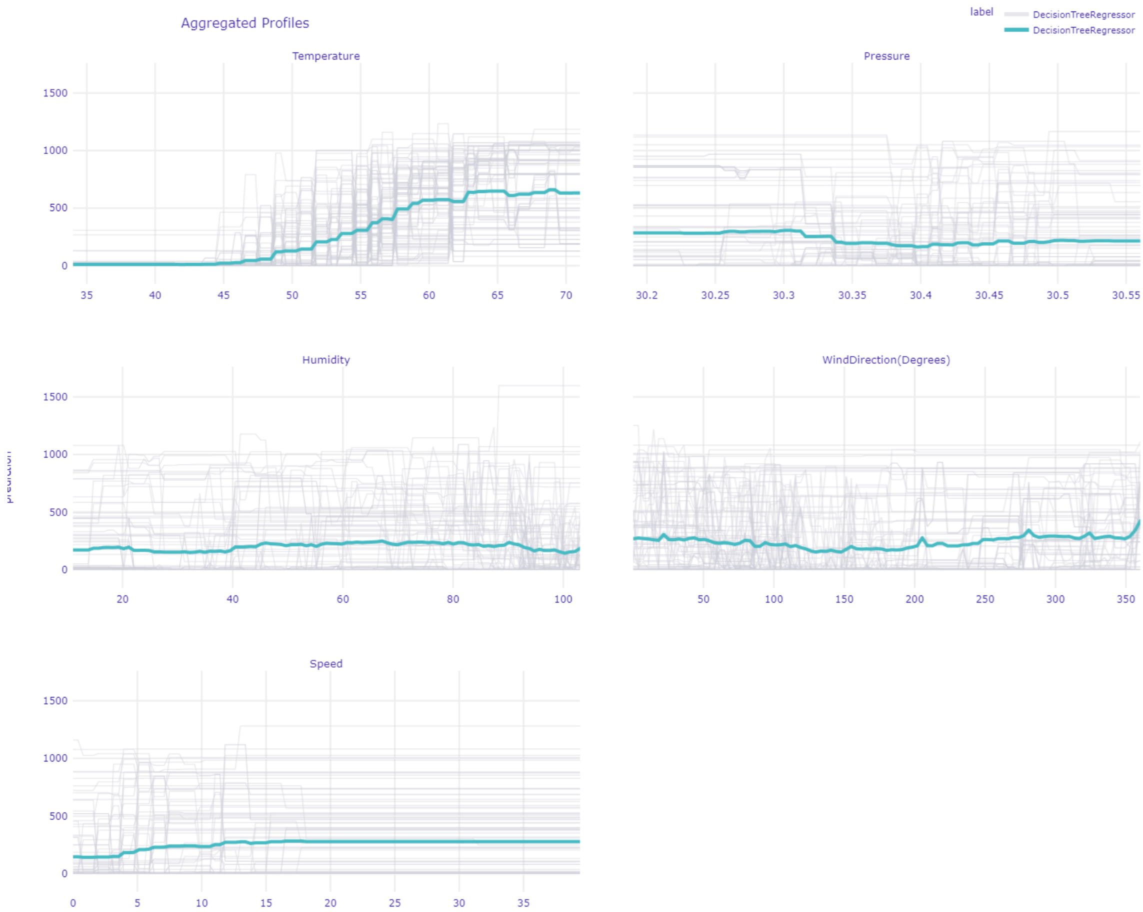


Figure 3.33: Partial dependence plots obtained from DALEX for Decision Tree.

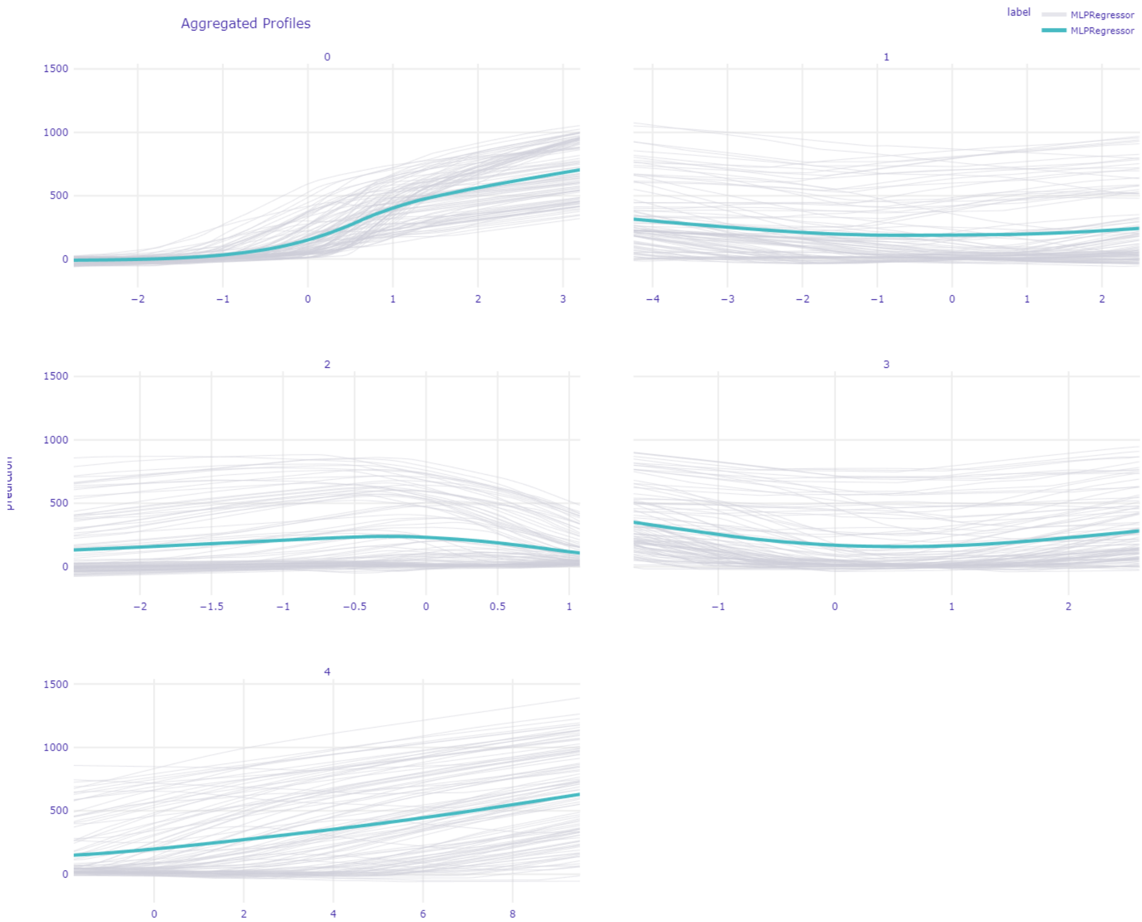


Figure 3.34: Partial dependence plots obtained from DALEX for MLP model.

PDA and ICE plots with SHAPASH: It provides easy-to-read visualization and a web app. By initializing a SmartExplainer object and running it with the run_app() function, we can visualize a web app with four parts. The four consist of a contribution plot, a feature importance plot where clicking on each feature will update the contribution plot, a local explanation of which feature contributes the most to the predicted value, and a selection table to select a subset and focus the exploration of that subset. Figure 3.35, 3.36 and 3.37 shows the SHAPASH visualization of feature contribution using three model. Again SHAPASH can compare the feature importance of a specific subset. Figure 3.38(left) shows the SHAPASH explanation of the feature importance of the subset [0,50,100,150,200,250,300,350,400,450,500,550] with plot.features_importance(selection=subset).

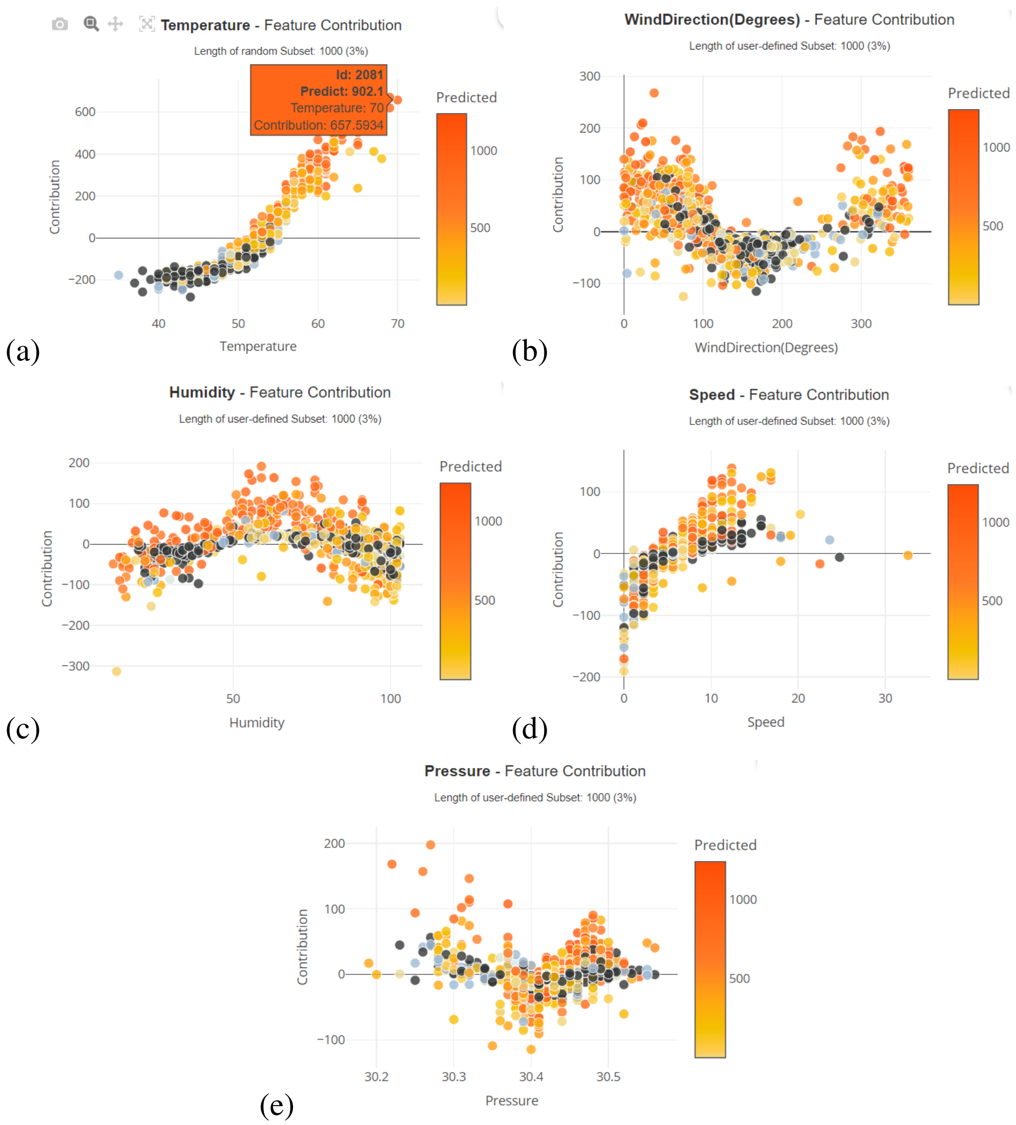


Figure 3.35: SHAPASH visualization of feature contribution of (a) temperature, (b) wind direction, (c) humidity, (d) wind speed, and (e) pressure using XGBoost model.

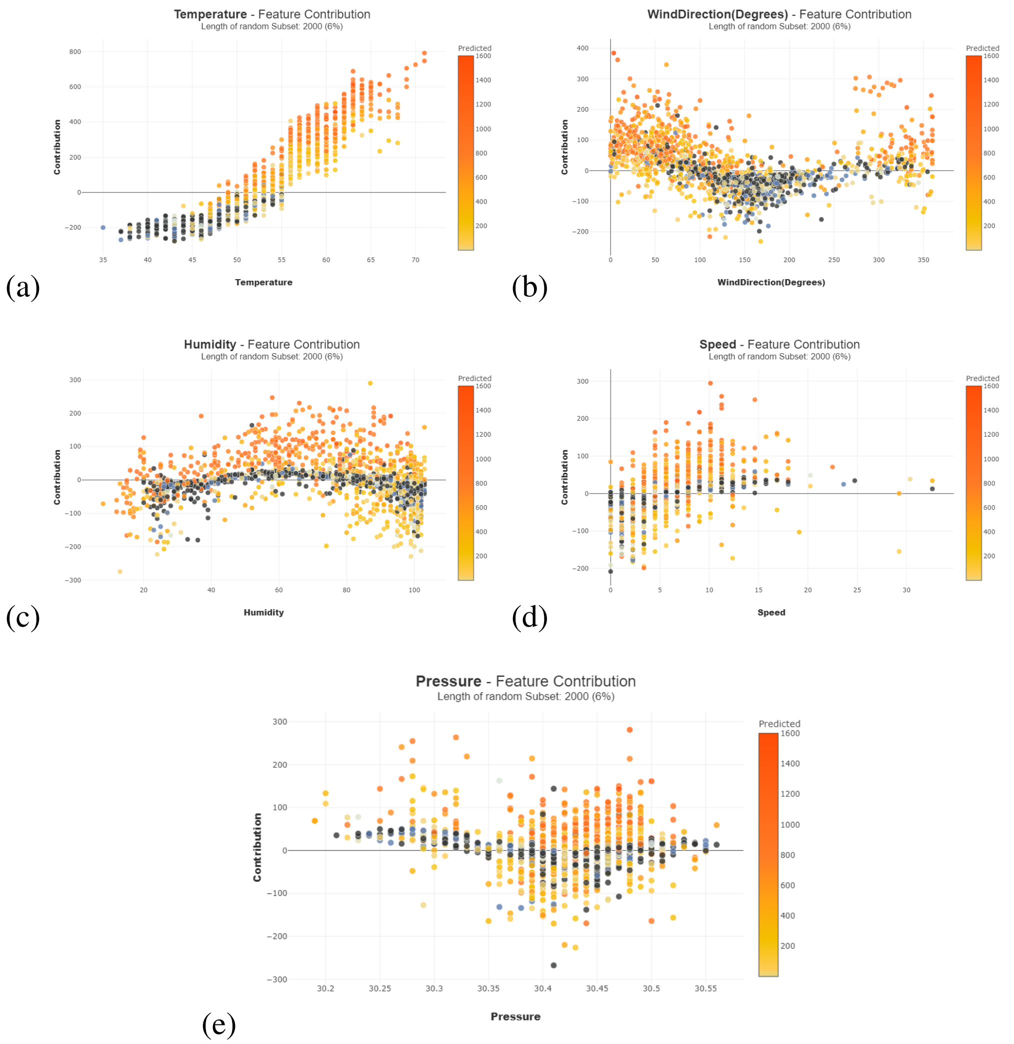


Figure 3.36: SHAPASH visualization of feature contribution of (a) temperature, (b) wind direction, (c) humidity, (d) wind speed, and (e) pressure using Decision Tree.

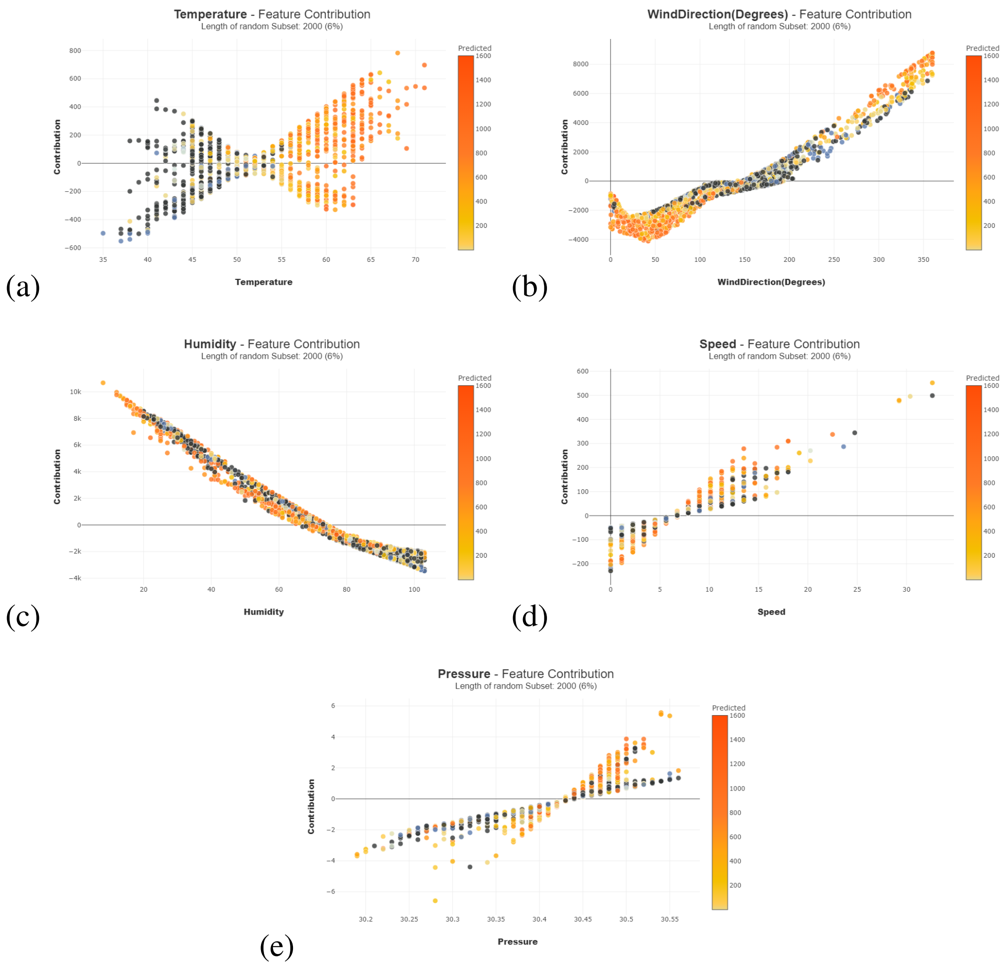


Figure 3.37: SHAPASH visualization of feature contribution of (a) temperature, (b) wind direction, (c) humidity, (d) wind speed, and (e) pressure using MLP.

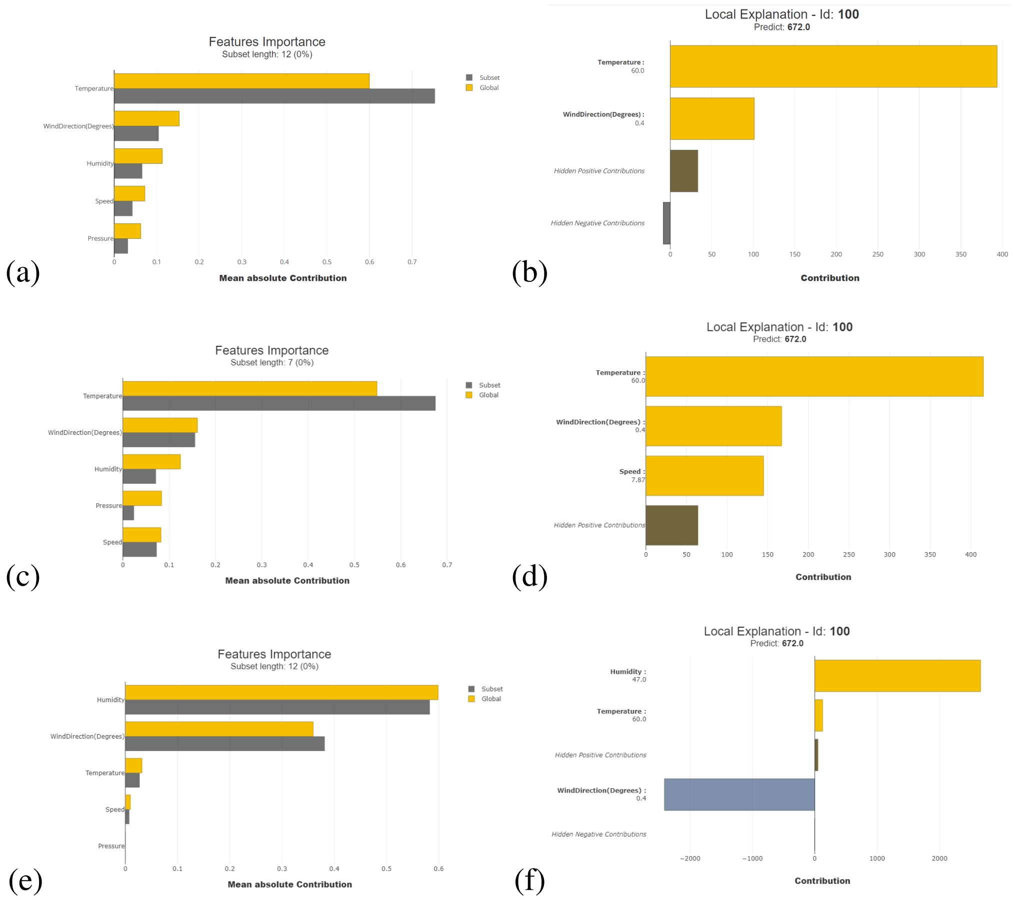


Figure 3.38: SHAPASH explanation of feature importance of a specific subset (left) and local explanation of the 100-th instance using local_plot() method (right) (a)XGBoost (b) XGBoost (c) Decision Tree (d) Decision Tree (e) MLP (f) MLP .

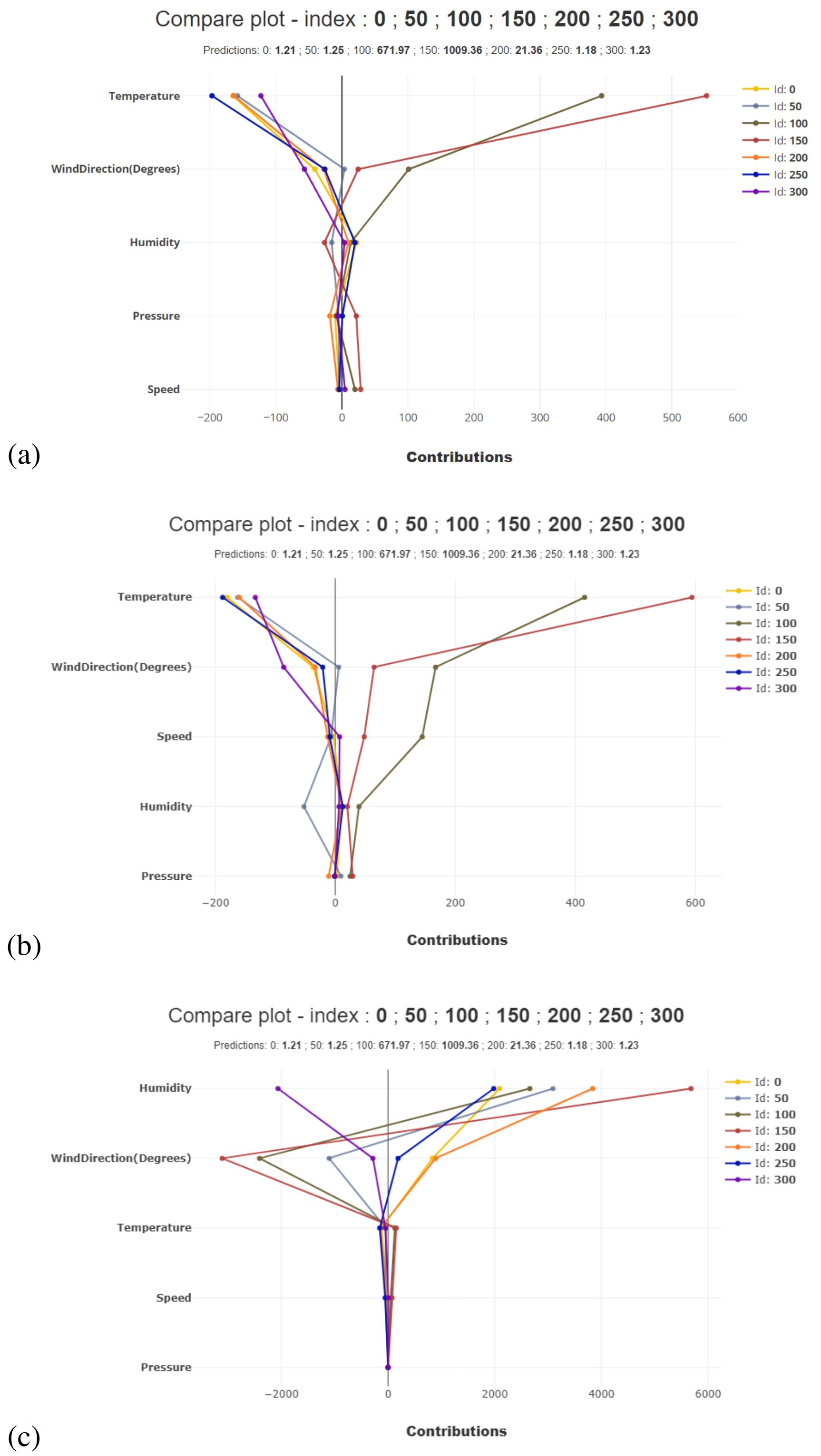


Figure 3.39: SHAPASH visualization of compare plot of 0-th,50-th,100-th,100-th,150-th,200-th,250-th,300-th instance using (a) XGBoost model, (b) Decision Tree (c) MLP

SHAPASH also gives the local explainability of models. The filter() and local_-plot() functions allow to summarize the local explainability. Figure 3.38(right) shows the local explanation of the 100-th instance which gives almost similar prediction as LIME and DALEX gives. Apart from individual local explanations, SHAPASH can compare between instances which can be visualized through the compare_plot() method. The SmartExplainer object explains why two or more individuals do not have the same predicted values. At the top of the plot, the most important criterion is visible. Figure 3.39 shows the SHAPASH compare plot of 0-th,50-th,100-th,100-th,150-th,200-th,250-th,300-th instance.

4. **DiCE**

The 'dice_ml' library has been used to generate counterfactual explanations for a specific instance. In order to maintain the desired outcome (in this fig 3.40 3.0 or 5.0), the diverse counterfactual set offers alternate input scenarios that could alter the model's prediction. These counterfactual justifications aid in comprehending the decision boundaries of the model as well as its sensitivity to various input feature combinations. Values with a '-' symbol denote features that are not specified during the counterfactual generation process and can have any value.

(a)

```
100%|██████████| 1/1 [00:00<00:00,  8.87it/s]Query instance (original outcome : 3.0)
```

	Temperature	Pressure	Humidity	WindDirection(Degrees)	Speed	Radiation
0	44	30.42	92	164.690002	6.75	3.0

```
Diverse Counterfactual set (new outcome: [3.0, 5.0])
```

	Temperature	Pressure	Humidity	WindDirection(Degrees)	Speed	Radiation
0	46.0	-	-	197.89	-	3.086700201034546
1	-	-	-	167.91	-	3.3767106533050537

(b)

```
100%|██████████| 1/1 [00:00<00:00,  6.43it/s]Query instance (original outcome : 870.0)
```

	Temperature	Pressure	Humidity	WindDirection(Degrees)	Speed	Radiation
0	59	30.469999	44	312.670013	3.37	870.0

```
Diverse Counterfactual set (new outcome: [3.0, 5.0])
```

	Temperature	Pressure	Humidity	WindDirection(Degrees)	Speed	Radiation
0	53.0	-	94.0	125.09	-	3.2799999713897705
1	43.0	-	92.0	165.45	23.18	3.440000057220459

Figure 3.40: Dice counterfactual explanation method applied on a specific instance using (a) XG-boost (b) Decision Tree

5. **Feature Visualization with Yellowbrick**

   The Yellowbrick 'ResidualsPlot' visualizer examines the residual distribution on both the training and test datasets to assist in diagnosing the model's performance in figure 3.41. The 'rank1d' visualizer (figure 3.42) aids in determining which features may be more predictive. The 'rank2d' visualizer facilitates the discovery of possible correlations or patterns between feature pairs. In both scenarios of figure 3.43, the visualizer offers a scatterplot matrix with each point denoting a pair of features and each point's colour indicating the degree of covariance or correlation. These visual aids can be used to select features, spot redundant features, comprehend possible multicollinearity problems in the dataset, and obtain insights into feature relationships.

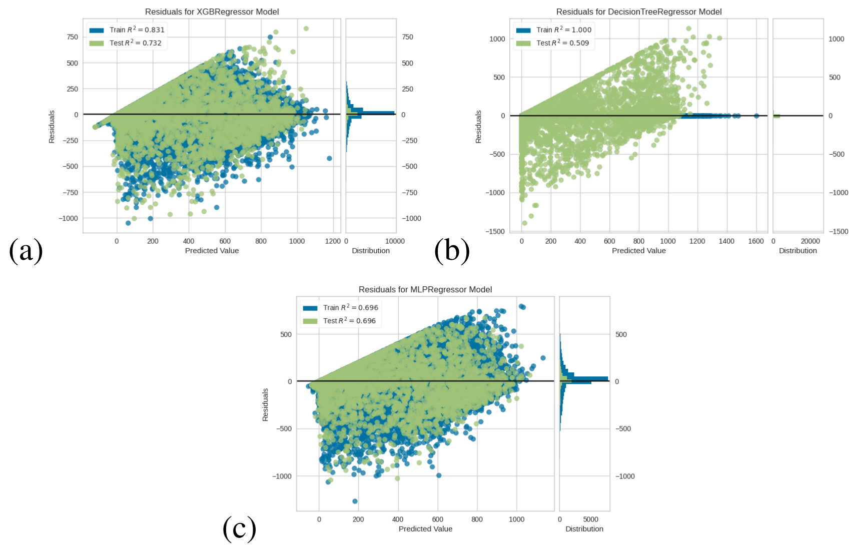


Figure 3.41: 'ResidualsPlot' visualizer from the Yellowbrick library to analyze the residuals of (a) XGBoost regression (b) Decision Tree (c) MLP model.

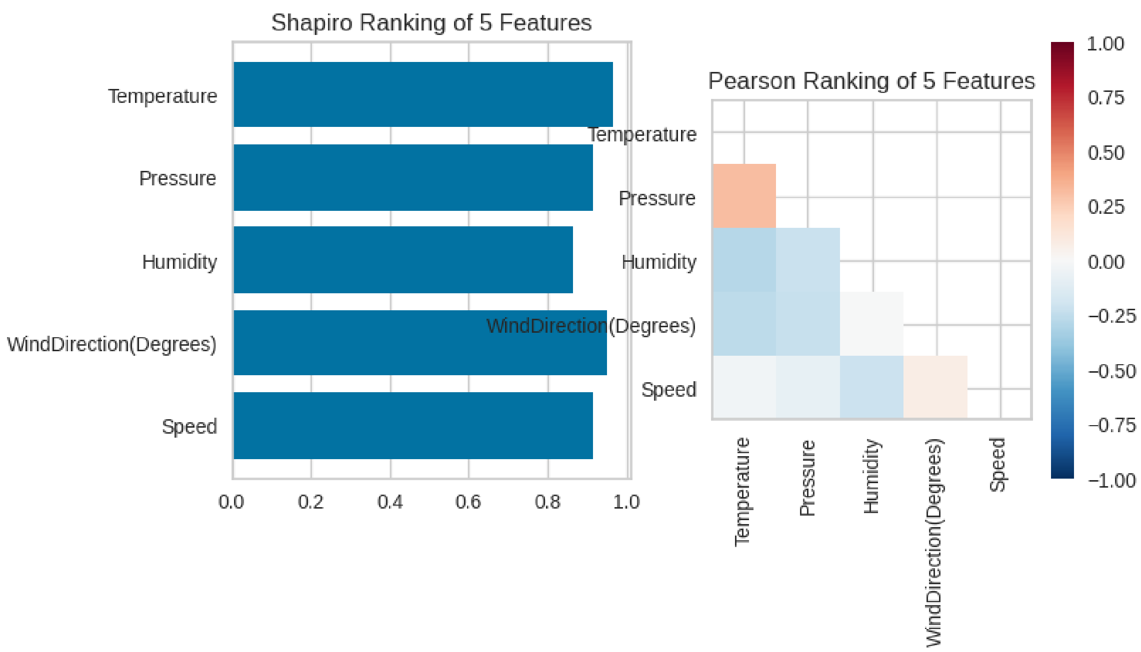
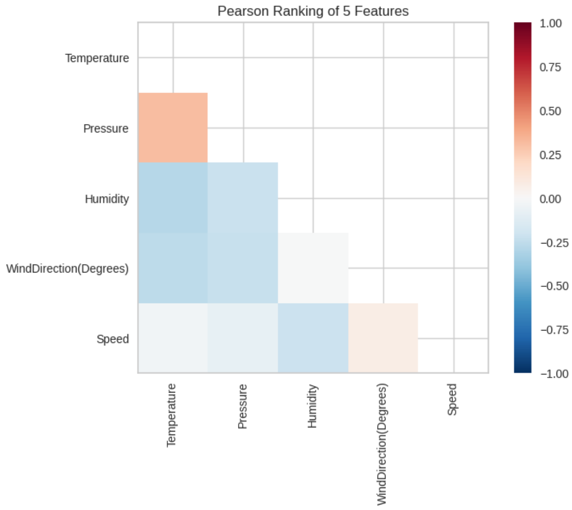
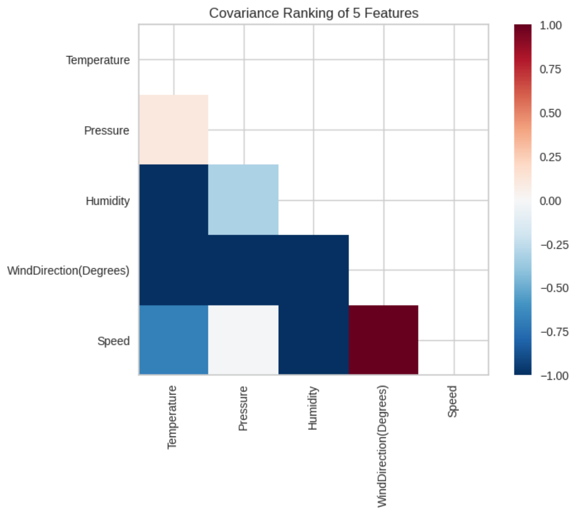


Figure 3.42: Yellowbrick's Rank1D and Rank2D visualizers to analyze and visualize the ranking of features



(a)



(b)

Figure 3.43: Yellowbrick's Rank2D Visualizer with (a) Pearson and (b) Covariance Ranking

6. **Anchor**

   Figure 3.44 provides the input used of the AnchorTabular explainer from the Alibi library using the XGBoost model and Decision tree. The output explained from Anchor is as follows:

```
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
predict_fn = lambda x: model.predict(x)
feature_names = X_train.columns
explainer = AnchorTabular(predict_fn, feature_names)
explainer.fit(X_train.values, disc_perc=(25, 50, 75))

idx = 0
print('Prediction: ', model.predict(X_test.iloc[[idx]].values)[0])
explanation = explainer.explain(X_test.iloc[[idx]].values, threshold=0.95)
print('Anchor: %s' % (' AND '.join(explanation.anchor)))
print('Precision: %.2f' % explanation.precision)
print('Coverage: %.2f' % explanation.coverage)
```

Figure 3.44: Use of AnchorTabular explainer from the Alibi library

for XGBoost Model :

Prediction: 708.8713 WARNING:alibi.explainers.anchors.anchor_base:Could not find an anchor satisfying the 0.95 precision constraint. Now returning the best non-eligible result. The desired precision threshold might not be achieved due to the quantile-based discretisation of the numerical features. The resolution of the bins may be too large to find an anchor of required precision. Consider increasing the number of bins in 'disc_perc', but note that for some numerical distribution (e.g. skewed distribution) it may not help. Anchor: Speed <= 3.37 AND Humidity <= 56.00 AND WindDirection(Degrees) > 179.22 AND Temperature > 55.00 AND Pressure > 30.46 Precision: 0.00 Coverage: 0.00

for Decision Tree :

Prediction: 870.45 WARNING:alibi.explainers.anchors.anchor_base:Could not find an anchor satisfying the 0.95 precision constraint. Now returning the best non-eligible result. The desired precision threshold might not be achieved due to the quantile-based discretisation of the numerical features. The resolution of the bins may be too large to find an anchor of required precision. Consider increasing the number of bins in ‘disc_perc‘, but note that for some numerical distribution (e.g. skewed distribution) it may not help. Anchor: Speed <= 3.37 AND Humidity <= 56.00 AND WindDirection(Degrees) > 179.22 AND Temperature > 55.00 AND Pressure > 30.46 Precision: 0.02 Coverage: 0.00

7. **TreeInterpreter**

   The Decision Tree model’s predictions for the test set (X_test) are broken down into three parts using the ’tree interpreter’ library: prediction: Each instance’s prediction made by the model.
   bias: The ensemble’s average prediction from each tree.
   contributions: How each characteristic affects the prediction for every instance.
   To make sure that the total of the feature and bias contributions matches the model’s prediction, the assertion check is carried out. The output is described as follows:
   prediction Shape:
   There are 6538 instances in the test set, and each prediction is a scalar (for regression), as indicated by the prediction’s shape of (6538, 1).
   bias Shape:
   There is one bias value for each occurrence in the test set, as indicated by the bias’s (6538,) shape.
   contributions Shape:
   Contributions have the form (6538, 5), which indicates that there are nine features and that each feature has a contribution value for each instance.
   Based on the quantity and quality of features, the output shapes match the expectations.

```
from treeinterpreter import treeinterpreter as ti
model = DecisionTreeRegressor()
model.fit(X_train, y_train)


prediction, bias, contributions = ti.predict(model, X_test)
preds, bias, contributions = ti.predict(model, X_test)
preds.shape, bias.shape, contributions.shape
print("Prediction Shape:", prediction.shape)
print("Bias Shape:", bias.shape)
print("Contributions Shape:", contributions.shape)
assert(numpy.allclose(model.predict(X_test), bias + np.sum(contributions, axis=1)))
```

```
Prediction Shape: (6538, 1)
Bias Shape: (6538,)
Contributions Shape: (6538, 5)
```

Figure 3.45: Use of 'treeinterpreter' on decision tree to interpret predictions

3.5 Phase 3: Hybridized Clustering to Forecast Solar Radiation

As an additional contribution of this study, this section focuses on the clustering aspect. Here, we detail the process of partitioning the dataset into clusters and employing the LightGBM model to generate predictions based on the clustered data.

Figure 3.46 demonstrates Hybridized Clustering for Solar Radiation Prediction. We initially segment the solar dataset into four clusters using the K-means clustering algorithm, categorizing them as "Very Sunny," "Sunny," "Cloudy," and "Very Cloudy." Subsequently, we proceed to partition each cluster data into separate training and testing sets. Following this, we employ LightGBM Regressor to generate predictions for each cluster's data.

Algorithm 7 represents pseudo code for Hybridized Clustering to forecast Solar Radiation.

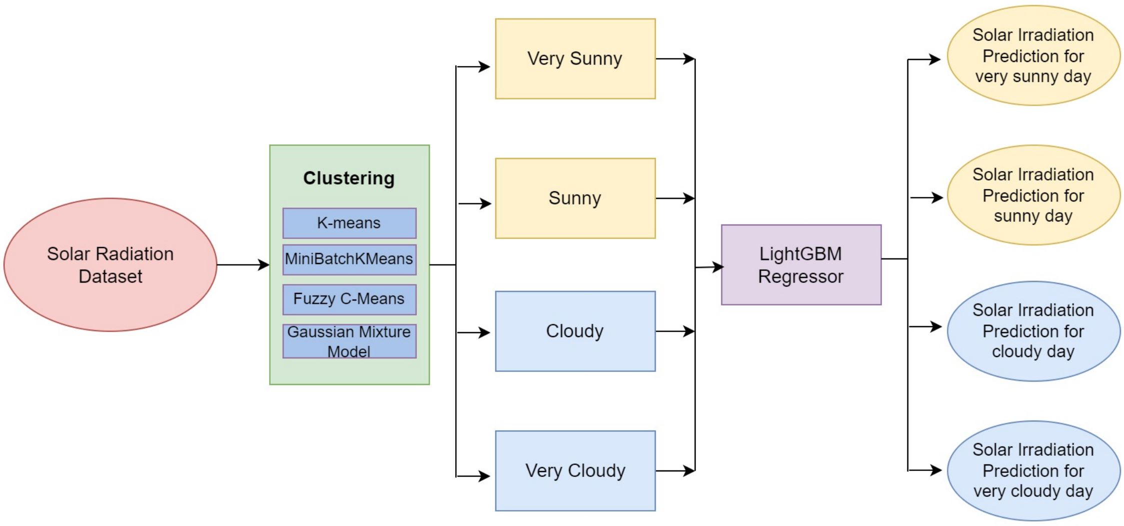


Figure 3.46: Hybridized Clustering to Forecast Solar Radiation

Algorithm 7: : Solar Irradiance Prediction using Hybridized Clustering

Step 1: Load the dataset of Solar Radiation, *df*
Step 2: Perform Exploratory Data Analysis on the dataset with feature values denoted as, $F = f_1, f_2, f_3, f_4, ..., f_n$
Step 3: Apply feature selection and set k=4 for k-means clustering
Step 4: Apply Clustering Algorithm
Step 5: Split the dataset into 4 clusters, C_1, C_2, C_3, C_4
Step 6: Split the 4 clusters dataset into $train_1, train_2, ..., train_k$ and $test_1, test_2, ..., test_k$
Step 7: Training the train datasets $train_1, train_2, ..., train_k$ with LightGBM
Step 8: Calculating errors using $test_1, test_2, ..., test_k$
Step 9: Store three different error values, E_{R^2}, E_{MAE} and E_{RMSE} for each cluster.

3.6 An interactive tool for predicting Solar Radiation based on weather data

We have developed an interactive tool with Streamlit (Khorasani, Abdou, & Hernández Fernández, 2022) that forecasts solar radiation using provided weather data. We employed the LightGBM machine learning model for making predictions. Additionally, we have incorporated SHAP to visualize the importance of features and contributions in the analysis of the predictions.

CHAPTER 4

EXPERIMENTS AND RESULTS

In this chapter, we present a comprehensive comparison between the classification results obtained from using ML and DL algorithms on augmented dataset while estimating tropical cyclones using satellite images or predicting tropical cyclones on numerical data. The experiments have been conducted on various types of ML and DL models including CNN, DenseNet, LSTM, VGG16, Boosting algorithms, SVM etc. The results have been analyzed and compared in terms of accuracy and error rate. The aim of this chapter is to evaluate the effectiveness of ML and DL algorithms while predicting tropical cyclones or estimating the intensity of tropical cyclones. The comparison provides a deeper insight into the performance of different ML and DL models and their contribution to the overall accuracy or error rate of the classification task.

4.1 Evaluation Metrics

R-squared (R^2) Score: R^2 score, also referred to as the coefficient of determination, is a statistical metric employed to evaluate the effectiveness of a regression model. It quantifies the proportion of the variability in the dependent variable that can be accounted for by the independent variables included in the model. Spanning a scale from 0 to 1, an R^2 score of 1 signifies a perfect fit of the model to the data, explaining all the variability. Conversely, an R^2 score of 0 implies that the model does not contribute any explanatory power. In practical terms, R^2 serves as a valuable tool for assessing how accurately a regression model captures and anticipates the underlying data patterns, making it a valuable measure in the context of regression analysis and model evaluation.

The equation for R^2 score is represented as follows:

$$R^2 = 1 - \frac{\sum_{obs=1}^{total} (z_{obs} - \hat{z}_{obs})^2}{\sum_{obs=1}^{total} (z_{obs} - \bar{z})^2} \tag{4.1}$$

Where,

$total$: the collective count of observations,

z_{obs} : the actual value of the obs-th observation,

$\hat{z}_{obs}$: for the forecasted value of the obs-th observation,

$\bar{z}$: the average of the actual values.

Root Mean Squared Error (RMSE): Root Mean Squared Error (RMSE) serves as the maestro in assessing the precision of a predictive model by calculating the square root of the average squared variances between predicted and actual values. Within the realm of regression analysis, RMSE orchestrates a symphony that balances accuracy and imperfections, producing a distinct tune that reflects how closely predictions align with reality. This metric transforms errors into a succinct measurement, articulating the performance rhythm through a harmonious fusion of squares and roots. Positioned as the pinnacle of model evaluation, RMSE composes a numerical climax, guiding practitioners through the intricate arrangement of predictive analytics.

The equation for Root Mean Squared Error (RMSE) can be expressed as follows:

$$\text{RMSE} = \sqrt{\frac{1}{total}\sum_{obs=1}^{total}(z_{obs} - \hat{z}_{obs})^2} \tag{4.2}$$

Where,

$total$: the collective count of observations,

z_{obs} : the actual value of the obs-th observation,

$\hat{z}_{obs}$: for the forecasted value of the obs-th observation.

Mean Absolute Error (MAE): Mean Absolute Error (MAE) stands as a commonly employed metric within statistics and machine learning to gauge the precision of a predictive model. It calculates the average of the absolute differences between the predicted values and the actual values in a given dataset. MAE offers a straightforward means to evaluate how well a model performs, treating all errors as equally important and disregarding their direction. A smaller MAE reflects a superior model fit to the data, with values approaching zero, indicating a higher level of accuracy. MAE's ease of comprehension and interpretation renders it a pragmatic choice for appraising regression models and evaluating their predictive prowess.

The equation for Mean Absolute Error (MAE) can be calculated as follows:

$$\text{MAE} = \frac{1}{total}\sum_{obs=1}^{total}|z_{obs} - \hat{z}_{obs}| \tag{4.3}$$

Where,

$total$: the collective count of observations,

z_{obs} : the actual value of the obs-th observation,

$\hat{z}_{obs}$: for the forecasted value of the obs-th observation.

4.2 Phase 1: Machine Learning and Deep Learning Techniques for Solar Radiation Prediction

4.2.1 Ensemble Machine-Learning Techniques

1. **Averaging** Table 4.1 represents evaluation metrics for the Averaging approach. The ensemble machine learning approach using averaging performs well according to the evaluation metrics. It achieves an R^2 score of 0.86, indicating that 86% of the variance in the dependent variable is explained. The Mean Absolute Error (MAE) is 9.11, suggesting a relatively small average prediction error, and the Root Mean Square Error (RMSE) is 11.47, which is also a reasonably low error.

Table 4.1: Evaluation Metrics for Averaging approach

Ensemble Machine-Learning	R^2 Score	MAE	RMSE
Linear Regression	0.83	9.96	12.28
Ridge Regression	0.83	9.96	12.28
Lasso Regression	0.83	10.16	12.44
ElasticNet Regression	0.80	10.95	13.55
Decision Tree Regressor	0.80	8.05	13.41
Averaging	**0.86**	9.11	11.47
Weighted Average	0.85	9.14	11.51

2. **Bagging**

   Bagging Regressor: Table 4.2 represents different base models employed in the bagging approach. The Bagging Regressor utilizes multiple base models, with the Decision Tree Regressor standing out as the top performer, achieving an impressive

R^2 score of 0.89 along with low MAE and RMSE, indicating exceptional predictive accuracy. While the other base models employed in the Bagging approach demonstrate reasonable performance, with R^2 scores hovering around 0.83, they fall short of the outstanding results produced by the Decision Tree Regressor. In contrast to Averaging, Stacking, and Blending, the Bagging Regressor, especially when combined with the Decision Tree base model, delivers superior outcomes, boasting higher R^2 values and lower MAE and RMSE metrics. It's evident that the choice of the base model within the Bagging Regressor can have a profound impact on its performance when compared to other ensemble methods.

Table 4.2: Evaluation Metrics of different base models used in the Bagging approach

Base Models	R^2 Score	MAE	RMSE
Decision Tree Regressor	0.89	6.61	9.88
Linear Regression	0.83	9.96	12.28
Ridge Regression	0.83	9.97	12.28
Lasso Regression	0.83	10.17	12.45
ElasticNet Regression	0.80	10.96	13.55

Random Forest Regressor: Table 4.3 represents evaluation metrics using a Random Forest Regressor. The Random Forest Regressor demonstrates outstanding performance, with an impressive R^2 score of 0.90, a remarkably low MAE of 6.29, and an RMSE of 9.44. These metrics collectively illustrate its exceptional explanatory power and precise prediction capabilities. When compared to the Bagging Regressor with a Decision Tree base model, the Random Forest Regressor surpasses it by achieving a higher R^2 score and lower MAE and RMSE. This highlights the Random Forest's superior performance in this context. In contrast to Averaging, Stacking, and Blending, the Random Forest Regressor outshines these ensemble methods by offering superior R^2 and lower MAE and RMSE values.

3. **Boosting**

Table 4.3: Evaluation Metrics for Random Forest Regressor

Model	R^2 Score	MAE	RMSE
Random Forest Regressor	**0.90**	6.29	9.44

Table 4.4 represents evaluation metrics of different Boosting models. Boosting models, particularly LightGBM and CatBoost Regressors, consistently exhibit exceptional performance, as evidenced by their high R^2 scores of 0.91 and low MAE values of 6.35 and 6.05, along with minimal RMSE values of 9.15 and 8.81, respectively. These models not only excel in their ability to provide accurate predictions but also in their capacity to elucidate the underlying data relationships, rendering Boosting an appealing choice for ensemble learning.

Compared to other ensemble techniques like Averaging, Stacking, Blending, and Bagging, Boosting methods, such as XGBoost, CatBoost, and LightGBM, outperform the competition consistently. They consistently yield higher R^2 scores and lower MAE and RMSE values, underscoring their superiority in terms of accuracy and explanatory power. Hence, in the context of ensemble learning, Boosting methods emerge as the preferred options, offering superior performance.

Table 4.4: Evaluation Metrics of different Boosting models

Boosting Models	R^2 Score	MAE	RMSE
LightGBM Regressor	**0.91**	6.35	9.15
CatBoost Regressor	**0.91**	6.05	8.81
XGBoost Regressor	0.90	6.52	9.49
Gradient Boosting Regressor	0.89	7.23	10.15
AdaBoost Regressor	0.74	13.98	15.43

4. **Stacking**

Table 4.5 represents evaluation metrics for the Stacking approach. Compared to Averaging, Stacking demonstrates a marginal improvement in the R^2 score, reaching 0.87 for meta-models, including Linear Regression, Ridge Regression, Lasso Regression, and ElasticNet Regression. Additionally, Stacking yields reduced Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for these meta-models, indicating improved predictive accuracy compared to the Averaging technique. However, it's noteworthy that employing a Decision Tree Regressor as the meta-model leads to a decrease in performance, resulting in an R^2 score of 0.76, MAE of 10.52, and MSE of 14.79.

Table 4.5: Evaluation Metrics of different estimator and meta-models used in the Stacking approach

Estimator	Meta Model	R^2 Score	MAE	RMSE
Ridge Regression Lasso Regression ElasticNet Regression Decision Tree Regressor	Linear Regression	0.87	8.11	10.97
Linear Regression Lasso Regression ElasticNet Regression Decision Tree Regressor	Ridge Regression	0.87	8.11	10.98
Linear Regression Ridge Regression ElasticNet Regression Decision Tree Regressor	Lasso Regression	0.87	8.12	11.00
Linear Regression Ridge Regression Lasso Regression Decision Tree Regressor	ElasticNet Regression	0.87	8.13	11.02
Linear Regression Ridge Regression Lasso Regression ElasticNet Regression	Decision Tree Regressor	0.76	10.52	14.79

5. **Blending**

Table 4.6 represents evaluation metrics for the Blending approach. The Blending approach showcases strong performance, achieving an R^2 score of 0.86 for meta models including Linear Regression, Ridge Regression, Lasso Regression, and ElasticNet Regression. However, its efficacy diminishes to 0.77 when employing the Decision

Tree Regressor as the meta-model. In contrast to Averaging, Blending delivers comparable results with marginally lower Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) while maintaining the same R^2 score. When juxtaposed with Stacking, Blending exhibits a slightly lower R^2 score alongside higher MAE and RMSE values. Consequently, Blending occupies an intermediate position between Averaging and Stacking in terms of performance, striking a balanced compromise between simplicity and accuracy.

Table 4.6: Evaluation Metrics of different estimator and meta-models used in the Blending approach

Estimator	Meta Model	R^2 Score	MAE	RMSE
Ridge Regression Lasso Regression ElasticNet Regression Decision Tree Regressor	Linear Regression	0.86	8.36	11.27
Linear Regression Lasso Regression ElasticNet Regression Decision Tree Regressor	Ridge Regression	0.86	8.24	11.15
Linear Regression Ridge Regression ElasticNet Regression Decision Tree Regressor	Lasso Regression	0.86	8.30	11.20
Linear Regression Ridge Regression Lasso Regression Decision Tree Regressor	ElasticNet Regression	0.86	8.30	11.17
Linear Regression Ridge Regression Lasso Regression ElasticNet Regression	Decision Tree Regressor	0.77	10.00	14.19

4.2.2 Comparison of Various Ensemble Techniques

Figure 4.1 illustrates R^2 score comparison across ensemble machine learning techniques, and it shows that LighGBM gives the better result. We further continue the experiment with LightGBM for explainable AI and clustering techniques.

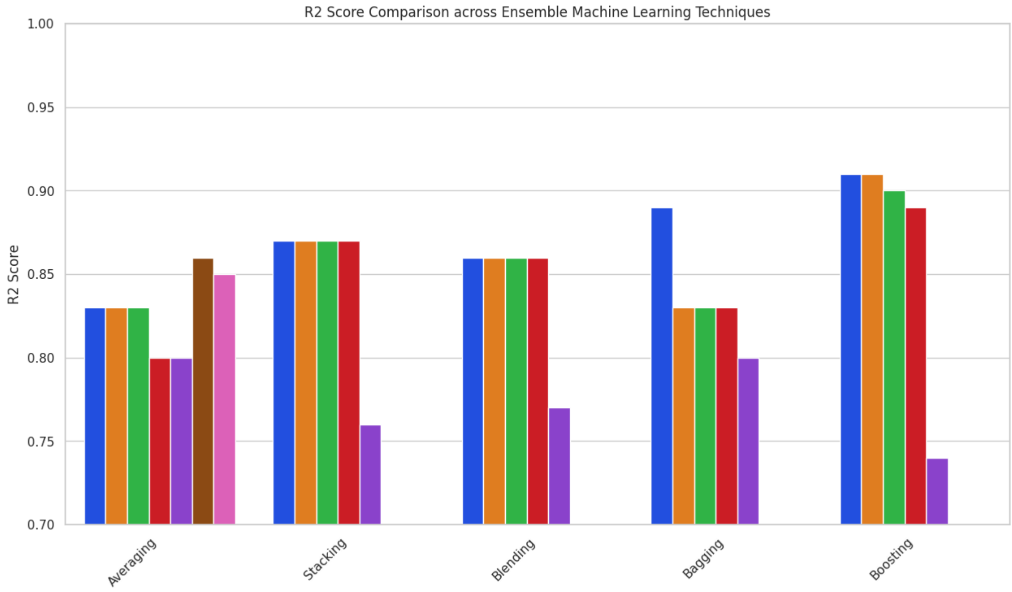


Figure 4.1: R^2 Score Comparison across Ensemble Machine Learning Techniques

PyCaret:

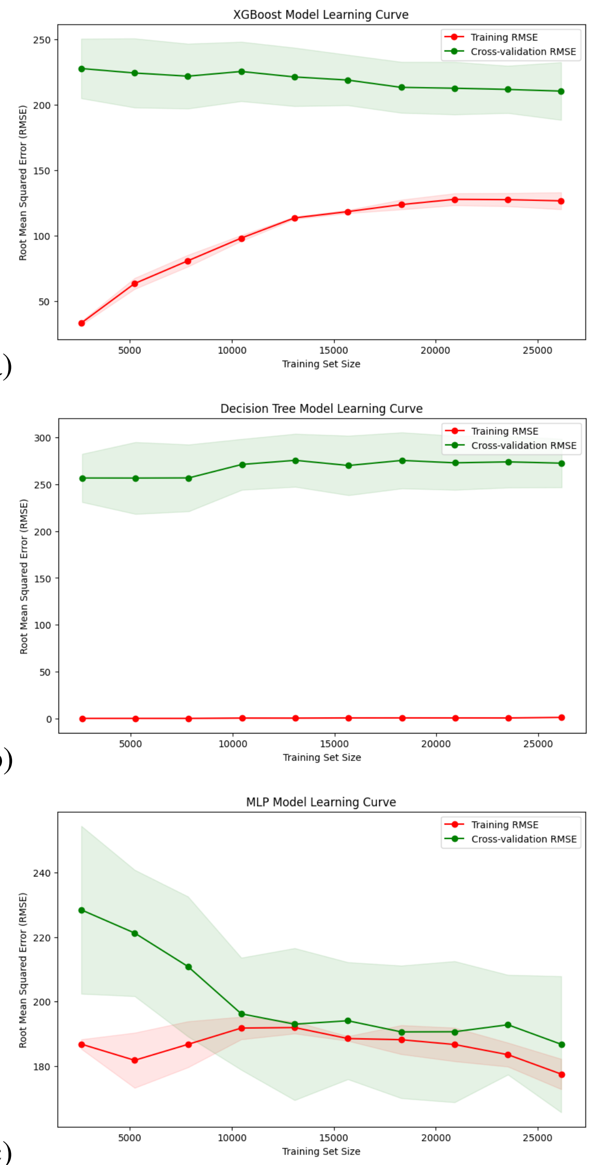
Table 4.7 represents evaluation metrics of machine learning models using PyCaret. Using PyCaret, the top-performing models, such as Extra Trees Regressor and Light Gradient Boosting Machine, achieve high R^2 scores (around 0.91), indicating a strong correlation between predicted and actual values. These models also exhibit relatively low MAE and RMSE values, suggesting accurate predictions. On the other hand, models like K Neighbors Regressor and AdaBoost Regressor have lower R^2 scores, indicating comparatively weaker performance in capturing the variance of the data. Linear regression-based models, while providing reasonable performance, exhibit slightly lower R^2 scores compared to some ensemble models.

Table 4.7: Evaluation Metrics of Machine Learning Models using PyCaret

ML Models	R^2 Score	MAE	RMSE
Extra Trees Regressor	**0.91**	6.17	8.91
Light Gradient Boosting Machine	**0.91**	6.30	9.03
Random Forest Regressor	0.90	6.26	9.24
Extreme Gradient Boosting	0.90	6.56	9.41
Gradient Boosting Regressor	0.89	7.18	9.86
Linear Regression	0.83	9.93	12.18
Ridge Regression	0.83	9.93	12.18
Lasso Regression	0.83	9.94	12.19
Bayesian Ridge	0.83	9.93	12.18
Huber Regressor	0.83	9.89	12.26
Lasso Regression	0.83	10.10	12.33
Lasso Least Angle Regression	0.83	10.10	12.33
Decision Tree Regressor	0.80	8.01	13.19
Elastic Net	0.80	10.81	13.37
K Neighbors Regressor	0.74	11.42	15.31
AdaBoost Regressor	0.73	13.87	15.39

4.3 Phase 2: Explainable AI in Solar Radiation Prediction

1. **Performance Analysis of Black Box Model** In Table 4.8 we have reported the performance of the performance of the three black box models used in this experiment. The decreasing trend of training and validation RMSE of the curve suggests that the model is learning and improving its predictions as more data is used. Figure 4.2 shows the model's performance on this dataset using the RMSE metric.



(a)

(b)

(c)

Figure 4.2: Learning curve of the performance of the model on the dataset using RMSE metric (a) XGBoost (b) Decision Tree (c) MLP.

Table 4.8: Evaluation Metrics of Machine Learning Models

Models	Evaluation Metrics		
	R^2 Score	MAE	RMSE
DT	0.514	108.620	219.672
XGBoost	0.732	96.037	163.076
MLP	0.696	106.939	173.660

2. **Feature Importance Score with Black Box model** XGBoost provides a built-in method to access feature importance scores, which can be extracted after training the model. These scores offer insights into which features are more influential in making predictions. Higher feature importance scores indicate a greater impact on the model's decision-making process.

   Figure 4.3 shows the feature importance plot of the three models for solar radiation prediction using feature_importances() function(for XGBoost and Decision Tree). XGBoost provides a built-in method to access feature importance scores, which can be extracted after training the model. These scores offer insights into which features are more influential in making predictions. Higher feature importance scores indicate a greater impact on the model's decision-making process. Decision trees use information gain and Gini impurity to determine the importance of a feature. Information gain measures the decrease in entropy or uncertainty when splitting data based on a feature, whereas Gini impurity measures how well a feature divides data into classes. Here, since we are using an MLPRegressor (a neural network model), it does not have a feature_importances_ attribute like tree-based models. So to analyze feature importance with a neural network, we considered using permutation importance for MLP. For a given dataset, the permutation_importance function determines the feature importance of estimators. The number of times a feature is randomly shuffled is set by the n_repeats parameter, which also yields a sample of feature importances.

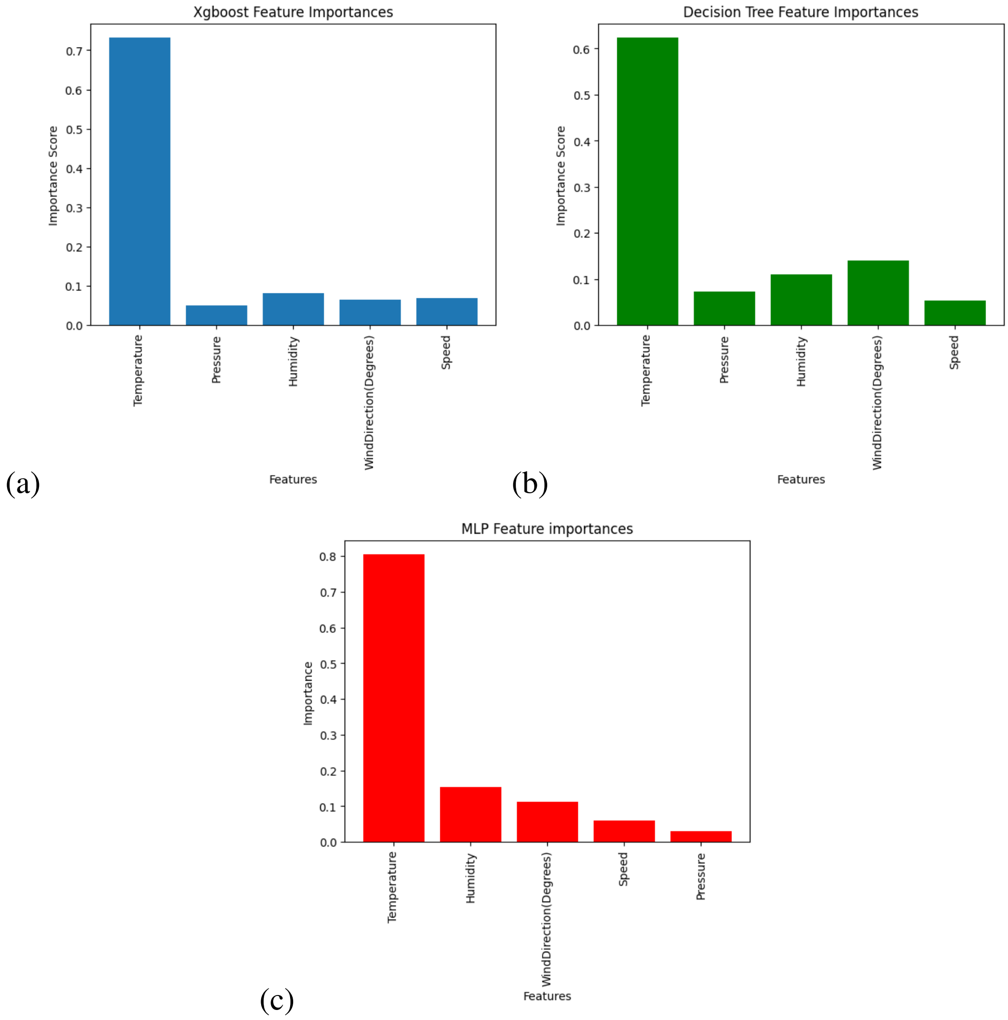


Figure 4.3: Feature importance plot (a) XGBoost (b) Decision Tree (c) MLP

3. **Visualizing Feature Importance With XAI Tools** This section delves into the role of XAI tools in comprehending complex black-box models by analyzing feature importance and contributions. It also discusses the process of removing specific features to visualize their effects on model performance, as evidenced by the RMSE metric.

   Complex black-box models, by nature, pose challenges in understanding their inner workings. XAI tools offer a valuable solution by providing insights into feature importance and contributions. These tools empower us to unravel the intricacies of these models, thus enhancing our understanding and confidence in their predictions.

The XAI tools used in this study have previously highlighted the importance and contribution of six selected features for solar radiation prediction. By leveraging the explanations provided by these tools, we can assess the relative significance of each feature and its impact on the model's predictive performance.

To further comprehend the effects of different features on the model, we conduct experiments by systematically removing each feature and observing the resulting change in RMSE. Table 4.9 presents the RMSE results with all six features and a subset of features, along with the removed features for visualization purposes.

The RMSE metric is a vital tool for evaluating model performance. A lower RMSE score indicates a more effective model, as it measures the closeness of predicted values to the actual values. The goal is to minimize RMSE to enhance the accuracy and effectiveness of the predictive model.

The experiments involving the removal of specific features provide valuable insights into the effects on model performance. By observing changes in RMSE, we can determine which features significantly impact the model's predictive accuracy and ascertain their relative importance.

Understanding the effects of feature subsets on model performance informs strategies for model optimization. By identifying non-critical features that can be excluded without substantial impact on predictive accuracy, we can streamline models, potentially reducing computational resources and improving efficiency.

Figure 4.4(a) shows the sum of the feature contribution in addition to the ”BIAS” with the subset of features. Figure 4.4(b), 4.4(c), 4.4(d), 4.4(e) explain the SHAP, SHAPASH, LIME, and DALEX report of the impact of the subset of the features on the model consecutively.

Table 4.9: RMSE results of each case

Models	Features removed	RMSE scores	RMSE (%)
with all six features	–	165.59	79.95%
LIME	Pressure	173.50	85.28%
SHAP	Humidity	179.94	86.88%
ELI5	Speed	171.92	84.50%
DALEX	Wind direction	170.48	83.79%
SHAPSH	Temperature	204.42	100.47%

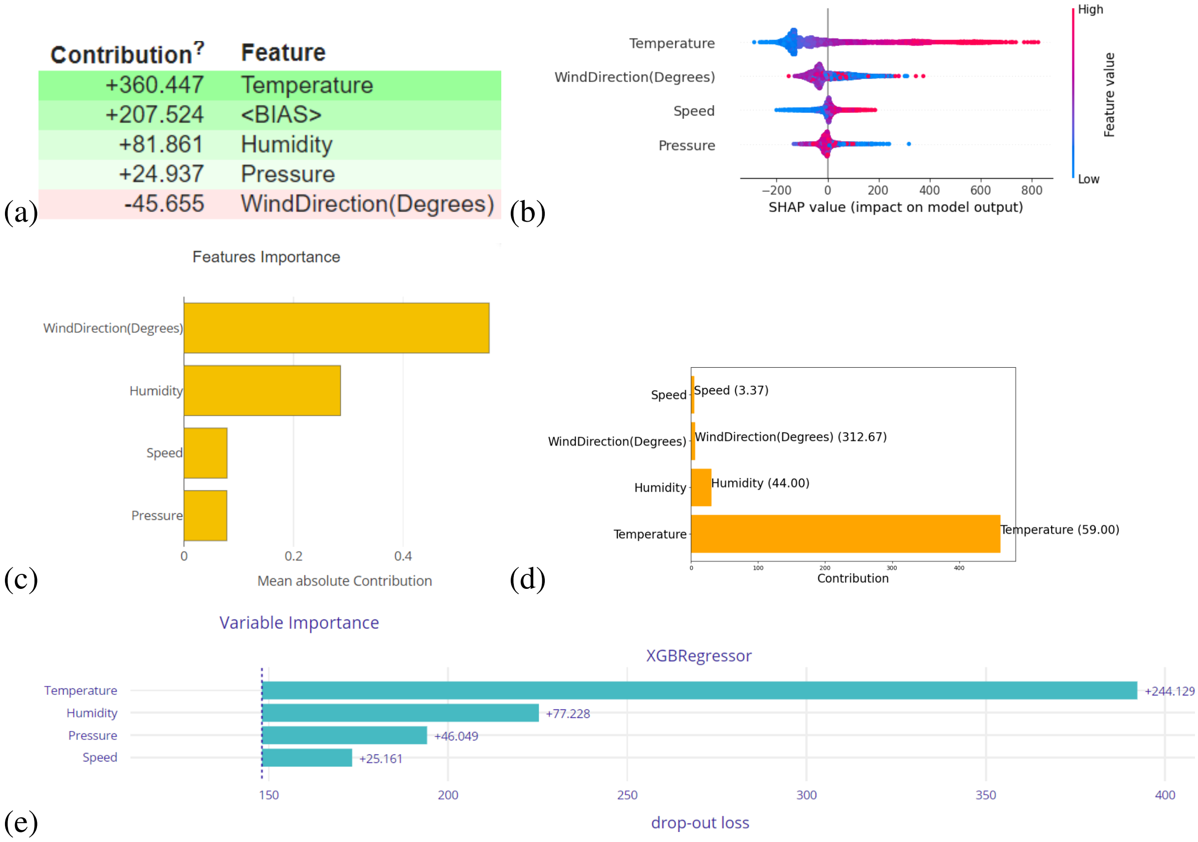


Figure 4.4: Feature contribution on the prediction with the subset of features (a) ELI5 explanation (b) SHAP explanation (c) SHAPASH visualization (d) LIME explanation (e) DALEX visualization

4. **Comparison of XAI tools**

All nine tools show that temperature has the greatest positive contribution and impact on solar radiation prediction. From the visualization, it is clear that temperature has an impact on other features also. SHAP provides more clearly detailed information than LIME and ELI5. It is easier to visualize through SHAP than LIME and ELI5. Again, DALEX and SHAPASH provide most of the things that LIME and ELI5 provide along with additional advantages. ELI5 is the simplest between LIME, SHAP, ELI5, DALEX and SHAPASH. SHAP provides a unified framework for explaining global model behaviour and individual prediction, where it computes shapely values to get the contribution of each feature. LIME provides an interpretable explanation in the form of weighted feature contribution. LIME shows the contribution of each feature, indicating which features harm the prediction and which have a positive impact. ELI5 provides an interpretable explanation through feature importance rankings. The significance of features can be understood by ELI5 with ease. ELI5 shows the weight of each feature with their ”bias” clearly. DALEX provides a systematic way to analyze model performance, feature importance, and interactions. It is useful for visualization and metrics for comparing model explanations across different algorithms. On the other hand, SHAPASH automates the generation of explanations and offers insights into feature contributions. A significant notice is that SHAPASH provides an interactive web app which makes easier for users to understand the feature contributions and feature importance of the model.

Importance of each feature and their contribution to the prediction can be identified and used for practical use through these XAI tools. In terms of integration, nine XAI tools can easily be integrated into the code as a Python library. All tools work well with the XGBoost regressor model.

Different Python libraries that are frequently used in the fields of machine learning and data visualisation include Dice, Anchor, TreeInterpreter, and Yellowbrick. The Dice library provides tools for decision tree visualisation and analysis, although its main purpose is to develop and assess decision tree models. Conversely, Anchor focuses on improving model transparency by developing interpretable machine learning models, particularly decision rules for individual predictions. As the name implies, TreeInterpreter's focus is on decision tree model interpretation, offering insights into feature contributions and model predictions. Last but not least, Yellowbrick distinguishes itself as a thorough visualisation library that facilitates a range of machine learning activities by providing tools for feature visualisation, model selection, and diagnostics. While Yellowbrick is a flexible tool, Dice and TreeInterpreter focus on decision trees, Anchor stresses interpretability, and versatile visualization toolkit for various machine learning applications, highlighting the diverse functionalities these libraries bring to the data science landscape.

4.4 Phase 3: Hybridized Clustering to Forecast Solar Radiation

1. **K-means-LightGBM**

   Table 4.10 represents evaluation metrics for K-means-LightGBM. The K-means-LightGBM methodology excels in weather prediction, particularly in the ”Very Sunny” and ”Very Cloudy” clusters. It achieves an outstanding R^2 score of 0.90 for very sunny days and an exceptional score of 0.93 for heavily overcast days. While still performing well in ”Sunny” and ”Cloudy” clusters, it demonstrates versatility and remarkable accuracy in predicting varying weather conditions.

2. **MiniBatchKMeans-LightGBM** Table 4.11 represents evaluation metrics for MiniBatchKMeans-LightGBM. The MiniBatchKMeans-LightGBM approach also exhibits exceptional performance in weather prediction in ”Very sunny” and ”Very

Table 4.10: Evaluation Metrics for K-means-LightGBM

Cluster	R^2 Score	MAE	RMSE
Very Sunny	0.90	6.05	8.47
Sunny	0.86	5.05	7.58
Cloudy	0.84	7.83	12.06
Very Cloudy	0.93	3.98	5.18

cloudy” scenarios. However, it maintains strong predictive capabilities in ”Sunny” and ”Cloudy” clusters, its versatility and remarkable accuracy shine through when forecasting diverse weather conditions.

Table 4.11: Evaluation Metrics for MiniBatchKMeans-LightGBM

Cluster	R^2 Score	MAE	RMSE
Very Sunny	0.88	6.67	9.42
Sunny	0.85	8.02	12.15
Cloudy	0.86	5.20	7.75
Very Cloudy	0.93	4.15	5.20

3. **Fuzzy C-Means-LightGBM** Table 4.12 represents evaluation metrics for Fuzzy C-Means-LightGBM. Like the K-means-LightGBM and MiniBatchKMeans-LightGBM methodologies, the Fuzzy C-Means-LightGBM approach demonstrates outstanding efficacy in predicting weather patterns, particularly excelling in scenarios characterized by ”Very Sunny” and ”Very Cloudy” conditions and slightly reduced performance in clustering of ”Sunny” and ”Cloudy” conditions.

Table 4.12: Evaluation Metrics for Fuzzy C-Means-LightGBM

Cluster	R^2 Score	MAE	RMSE
Very Sunny	0.89	6.06	8.62
Sunny	0.83	7.79	10.97
Cloudy	0.85	6.69	10.15
Very Cloudy	0.94	3.94	5.26

4. **GaussianMixture-LightGBM** Table 4.13 represents evaluation metrics for GaussianMixture-LightGBM. Similar to K-means-LightGBM, MiniBatchKMeans-LightGBM, and Fuzzy C-Means-LightGBM, the GaussianMixture-LightGBM approach demonstrates outstanding performance in weather prediction under ”Very Cloudy” conditions. However, its effectiveness diminishes when confronted with ”very sunny” clustering scenarios. Interestingly, the clustering performance for ”sunny” and ”cloudy” conditions remains relatively consistent.

Table 4.13: Evaluation Metrics for GaussianMixture-LightGBM

Cluster	R^2 Score	MAE	RMSE
Very Sunny	0.85	6.22	8.71
Sunny	0.86	7.33	11.05
Cloudy	0.86	5.40	8.21
Very Cloudy	0.94	3.86	4.87

4.5 Interactive Tool

In the Interactive Tool, Figures 4.5 and 4.6 depict the forecast for solar radiation and the corresponding feature contributions. The left sidebar allows users to modify input parameter values under ”Specify Input Parameters,” with the updated settings reflected in the ”Specified Input Parameters” section. The LightGBM machine learning model predicts solar radiation, and the results are displayed in the ”Prediction of Solar Irradiation” section. Additionally, the ”Feature Importance” section utilizes SHAP to illustrate the significance and contribution of different features.

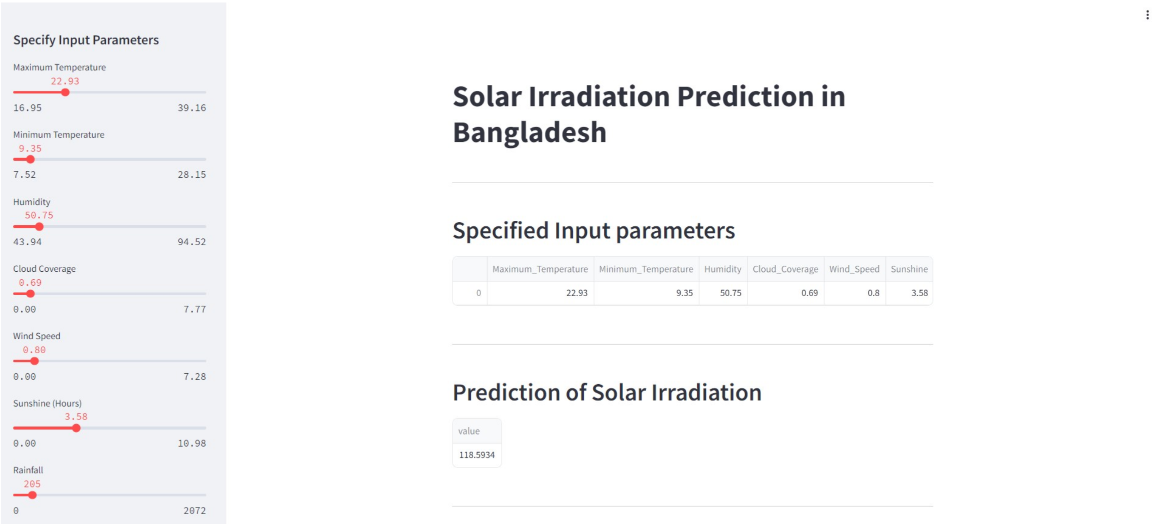


Figure 4.5: Interactive Tool with solar prediction

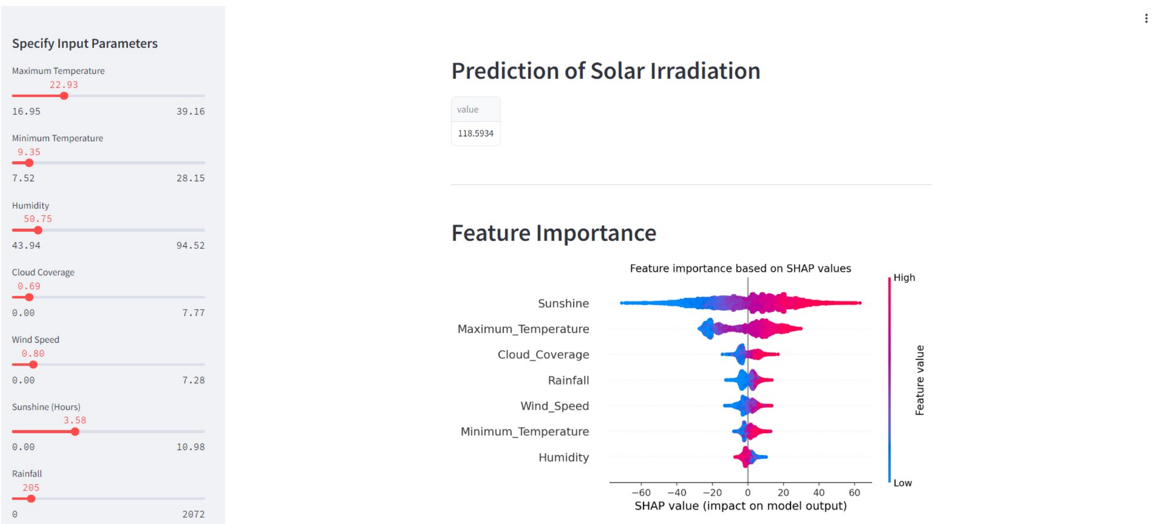


Figure 4.6: Interactive Tool with SHAP value

CHAPTER 5

ENGINEERING CONSIDERATIONS, CHALLENGES AND REMEDIES

5.1 Design Constraints

5.1.1 Ethical Consideration:

The data is collected for our thesis from the public websites. We are committed to transparency, disclosing any potential conflicts of interest that could influence our research process or outcomes, whether financial, personal, or professional. Throughout our research, we meticulously document methodologies, data sources, and analytical techniques employed to maintain transparency. We take responsibility for the accuracy and reliability of our research findings, acknowledging any limitations or uncertainties in the data or methodology used.

5.1.2 Environmental Consideration:

The thesis aims to optimize solar resource utilization using machine learning, enhancing the efficiency and reliability of solar energy systems by accurately predicting solar irradiance patterns. This contributes to global clean energy transition goals, aligning with initiatives like Sustainable Development Goals. The focus is on promoting environmental stewardship, reducing pollution, and conserving natural resources. Additionally, the research empowers communities by democratizing access to solar energy technologies, reducing energy poverty, and improving resilience to climate change. Overall, it seeks to create a sustainable future where clean energy plays a central role while preserving the planet for future generations.

5.1.3 Social Constraints:

The key social constraint in solar energy adoption is accessibility and affordability, especially for marginalized communities. Issues include inadequate grid connectivity, financial constraints, and regulatory barriers. Upfront costs of solar installations pose challenges, while concerns about reliability and maintenance deter potential adopters. To address these constraints, solutions include financial incentives, decentralized solar solutions, community empowerment through training, streamlined regulations, and public awareness campaigns. Overcoming these barriers can lead to a more inclusive and equitable energy transition, promoting sustainable development goals.

5.1.4 Sustainability in Environmental and Societal Context:

The thesis focuses on predicting and optimizing solar irradiance while integrating principles of sustainability, encompassing environmental conservation and societal well-being. Environmentally, it emphasizes responsible use of solar energy, reducing greenhouse gas emissions, and minimizing ecological footprint through innovative techniques. Societally, it promotes equity, access, and social inclusion in solar energy benefits, engaging communities and fostering local participation. Through multidimensional sustainability efforts, the research aims to advance global sustainable development goals by leveraging solar energy for positive environmental and social impact.

5.2 Complex Engineering Problem

5.2.1 Complex Problem Solving

1. **Depth of knowledge required:** Throughout our thesis, we have delved deeply into machine learning, deep learning, clustering, and explainable AI, acquiring profound knowledge in these areas. Our exploration has not only equipped us with a thorough grasp of the theoretical underpinnings and real-world implications of these fields but has also played a part in pushing the boundaries of AI research and development

forward.

2. **Depth of analysis required:** In our thesis, we conducted thorough analyses to select the most suitable machine learning and deep learning models, as well as the most effective explainable AI techniques. This involved delving deep into various options to ensure that we chose the models and techniques that best fit the requirements of our research and provide meaningful insights into our study domain.

3. **Extend of applicable codes:** Throughout our thesis, we prioritize adherence to professional engineering standards and codes of practice across all aspects of machine learning, deep learning, clustering, and explainable AI code. This commitment ensures that our methodologies and implementations meet the rigorous criteria set forth by the engineering profession, guaranteeing the reliability, integrity, and quality of our research and its practical applications.

4. **Interdependence:** Our thesis methodology comprises three interdependent phases. In Phase 1, we employ Machine Learning and Deep Learning Techniques to predict solar radiation. Building on this foundation, Phase 2 focuses on Explainable Artificial Intelligence (XAI) within the specific context of Solar Radiation Prediction, aiming to enhance the interpretability and transparency of our models. Finally, in Phase 3, we integrate Hybridized Clustering techniques to forecast solar radiation, leveraging the insights gained from both machine learning and explainable AI approaches. This sequential progression allows us to develop a comprehensive framework for accurate and understandable solar radiation forecasting.

5.2.2 Engineering Activities

1. **Range of resources:** For our thesis, we extensively utilize a diverse range of resources including journals, papers, and other scholarly materials to deepen our understanding and knowledge of machine learning, deep learning, clustering, and ex-

plainable AI. These resources serve as foundational sources that inform our research and help us develop robust methodologies and frameworks in these domains.

2. **Level of interaction:** Our thesis methodology unfolds across three levels. Initially, Phase 1 utilizes Machine Learning and Deep Learning Techniques for solar radiation prediction. Expanding upon this groundwork, Phase 2 delves into Explainable Artificial Intelligence (XAI) tailored to Solar Radiation Prediction, enhancing model interpretability and transparency. Subsequently, Phase 3 integrates hybrid clustering techniques to forecast solar radiation, drawing insights from both machine learning and explainable AI methodologies. This sequential advancement enables the development of a comprehensive framework for precise and comprehensible solar radiation forecasting.

3. **Inovation:** In our thesis, we leverage creative applications of engineering principles and research-based knowledge in innovative ways. Specifically, we implement explainable AI techniques for analyzing a Bangladeshi dataset to predict solar radiation. As far as our knowledge extends, this marks the pioneering use of explainable AI for solar radiation prediction within the context of Bangladeshi datasets, contributing novel insights to the field.

CHAPTER 6

DISCUSSION AND CONCLUSION

6.1 Discussion

In our research, we utilized two datasets to forecast solar irradiance and tried to enlighten solar energy future.

In the first phase of our study, we evaluated various ensemble machine learning methods, including Averaging, Stacking, Blending, Bagging, Random Forest, and Boosting, to assess their predictive performance. We found that Averaging and Stacking achieved strong R^2 scores of 0.86 and 0.87, striking a balance between simplicity and accuracy. Stacking performed slightly better than Averaging. Blending had a similar R^2 score (0.86) to Averaging but showed a slight edge in terms of lower Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). However, it fell slightly behind Stacking in R^2 and had higher MAE and RMSE, offering a trade-off between simplicity and accuracy. The Bagging Regressor, especially with a Decision Tree as a base model, achieved an impressive R^2 score of 0.89, indicating strong predictive accuracy. Other base models in Bagging performed reasonably with R^2 scores around 0.83. The choice of the base model significantly influenced Bagging's performance. The Random Forest Regressor outperformed Bagging with the Decision Tree base model, achieving an exceptional R^2 score of 0.90, low MAE, and RMSE. Boosting methods, such as LightGBM and CatBoost Regressors, consistently demonstrated outstanding performance with high R^2 scores (0.91) and low MAE, indicating accurate predictions and strong data relationship elucidation. XGBoost, CatBoost, and LightGBM consistently outperformed the competition, providing higher R^2 scores and lower MAE and RMSE values, highlighting their accuracy and explanatory power. Finally, PyCaret highlights strong performance from models like Extra Trees Regressor and Light Gradient Boosting Machine, yielding high R^2 scores around 0.91, while K Neighbors Regressor and AdaBoost Regressor show comparatively weaker results. Linear regression-based models perform reasonably but with slightly lower R^2 scores compared to ensembles.

The second phase of our study sheds light on the key drivers behind solar radiation prediction. Nine Explainable AI methods were employed to gain insights into how specific features impact the model's predictions. SHAP Explanations have shown that sunshine hours and maximum temperature are the most significant factors influencing the prediction. Meanwhile, ELI5's Feature Importance Scores underscore the importance of sunshine hours, closely followed by maximum and minimum temperatures, as well as cloud coverage. These scores provide a numerical measure of how each feature affects the prediction, confirming the importance of sunshine hours. LIME Explanations, on the other hand, delved deeper into the contributions of each feature and revealed that maximum temperature has the strongest negative impact, indicating its significant influence on the prediction. Taking all six features into account, the prediction probability was calculated to be 145.36.

Moving to our study's third phase, we used Hybridized Clustering to forecast Solar Radiation in Bangladesh. Notably, the K-means-LightGBM, MiniBatchKMeans-LightGBM, and Fuzzy C-Means-LightGBM models excelled in forecasting solar radiation levels within both "Very Sunny" and "Very Cloudy" clusters. However, the effectiveness of the GaussianMixture-LightGBM approach declined notably in scenarios characterized by "Very Sunny" clusters. Interestingly, the clustering performance remained relatively stable when predicting solar radiation in "Sunny" and "Cloudy" conditions. Finally, we've developed an interactive tool with Streamlit that forecasts solar radiation using provided weather data. We've utilized the LightGBM machine learning model for prediction, and to enhance interpretability, we've integrated SHAP to visualize the importance of features and contributions in the analysis of the predictions.

6.2 Thesis Limitations

While research on explainable AI, machine learning (ML), and deep learning (DL) shows promise for predicting solar radiation, a few problems still need to be worked out:

1. **Data Quality and Availability:** Obtaining sufficient high-quality data can be difficult, particularly in areas with little monitoring infrastructure, but it is essential for training deep learning models. The precision of forecasts may be hampered by this lack of information.

2. **Adaptability to Different Locations:** Models developed using data from one place may not perform well in other regions with dissimilar topography or climate. This implies that when predictions are used in unfamiliar settings, they might not be precise or trustworthy.

3. **Incorporating Other Significant Factors:** Although meteorological information is crucial for forecasting solar radiation, other elements such as land use and pollution levels may also have an impact. The intricacy of incorporating these elements into ML and DL models can impact prediction accuracy.

4. **Handling Uncertainty:** This is a necessary part of solar radiation prediction, particularly when circumstances change. In dynamic environments, the prediction accuracy of ML and DL methods is often compromised due to their inability to accurately account for this uncertainty.

5. **Data Access:** Accurate and relevant data for model training and validation can be challenging to obtain due to privacy laws and data ownership concerns. It is difficult to create and enhance ML and DL models for solar radiation prediction without enough data.

6.3 Future Work

Future studies on solar irradiance forecasting might look into a number of approaches to improve precision and usefulness. This includes developing more sophisticated temporal and spatial models that take geographic specifics into account, incorporating climate change factors, exploring real-time predictive models for practical applications, field testing for model validation, analyzing policy and economic implications, and developing community engagement initiatives to raise awareness. Also included are further refinements of ensemble machine learning techniques and the integration of advanced AI methodologies beyond the XAI models used. These options present viable means of enhancing solar irradiance forecasts, assisting Bangladesh in building a more sustainable and energy-secure future.

6.4 Conclusion

In our study, we assessed the predictive performance of various machine learning and deep learning methods. Following that, we ran the datasets through nine XAI models. Finally, our research concludes with a combined prediction based on hybridized clustering. In conclusion, our study used a multifaceted approach to forecasting solar irradiance in Bangladesh, employing a methodology aimed at understanding, predicting, and leveraging the region's solar resources. From employing collective machine learning strategies to incorporating Explainable AI approaches each stage has made a significant contribution to illuminating the path toward a more sustainable and energy-secure future for Bangladesh. This all-encompassing strategy, which combines environmental awareness with technological advancements, represents an ongoing commitment to maximizing the sun's potential for a greener, more sustainable world.

References

Abedin, Z., Barua, M., Paul, S., Akther, S., Chowdhury, R., & Chowdhury, M. S. U. (2017). A model for prediction of monthly solar radiation of different meterological locations of bangladesh using aritficial neural network data mining tool. In *2017 international conference on electrical, computer and communication engineering (ecce)* (p. 692-697). doi: 10.1109/ECACE.2017.7912993

Alam, M. S., Al-Ismail, F. S., Hossain, M. S., & Rahman, S. M. (2023). Ensemble machine-learning models for accurate prediction of solar irradiation in bangladesh. *Processes*, *11*(3), 908.

Ali, M. (2020). Pycaret: An open source, low-code machine learning library in python. *PyCaret version*, *2*.

Alizamir, M., Shiri, J., Fard, A. F., Kim, S., Gorgij, A. D., Heddam, S., & Singh, V. P. (2023). Improving the accuracy of daily solar radiation prediction by climatic data using an efficient hybrid deep learning model: Long short-term memory (lstm) network coupled with wavelet transform. *Engineering Applications of Artificial Intelligence*, *123*, 106199. Retrieved from doi: https://doi.org/10.1016/j.engappai.2023.106199

Alkhayat, G., & Mehmood, R. (2021). A review and taxonomy of wind and solar energy forecasting methods based on deep learning. *Energy and AI*, *4*, 100060. Retrieved from doi: https://doi.org/10.1016/j.egyai.2021.100060

Alzahrani, A., Shamsi, P., Dagli, C., & Ferdowsi, M. (2017). Solar irradiance forecasting using deep neural networks. *Procedia Computer Science*, *114*, 304-313. Retrieved from (Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017, Chicago, Illinois, USA) doi: https://doi.org/10.1016/j.procs.2017.09.045

Amin, M. N., Iftikhar, B., Khan, K., Javed, M. F., AbuArab, A. M., & Rehman, M. F. (2023). Prediction model for rice husk ash concrete using ai approach: Boosting and bagging algorithms. In *Structures* (Vol. 50, pp. 745–757).

Anwar, M. T., Islam, M. F., & Alam, M. G. R. (2023). Forecasting meteorological solar irradiation using machine learning and n-beats architecture. In *Proceedings of the 2023 8th international conference on machine learning technologies* (pp. 46–53).

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... others (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, *58*, 82–115.

Aslam, M., Lee, J.-M., Kim, H.-S., Lee, S.-J., & Hong, S. (2020). Deep learning models for long-term solar radiation forecasting considering microgrid installation: A comparative study. *Energies*, *13*(1). Retrieved from doi: 10.3390/en13010147

Ayodele, T., Ogunjuyigbe, A., Amedu, A., & Munda, J. (2019). Prediction of global solar irradiation using hybridized k-means and support vector regression algorithms. *Renewable Energy Focus*, *29*, 78–93.

Bae, K. Y., Jang, H. S., & Sung, D. K. (2016). Hourly solar irradiance prediction based on support vector machine and its error analysis. *IEEE Transactions on Power Systems*, *32*(2), 935–945.

Baniecki, H., Kretowicz, W., Piatyszek, P., Wisniewski, J., & Biecek, P. (2021). Dalex: Responsible machine learning with interactive explainability and fairness in python. *The Journal of Machine Learning Research*, *22*(1), 9759–9765. doi: 10.48550/arXiv.2012.14406

Benmouiza, K., Tadj, M., & Cheknane, A. (2016). Classification of hourly solar radiation using fuzzy c-means algorithm for optimal stand-alone pv system sizing. *International journal of electrical power & energy systems*, *82*, 233–241.

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & geosciences*, *10*(2-3), 191–203.

Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test*, *25*, 197–227.

Biecek, P. (2018). Dalex: Explainers for complex predictive models in r. *The Journal of Machine Learning Research*, *19*(1), 3245–3249.

Cannon, R. L., Dave, J. V., & Bezdek, J. C. (1986). Efficient implementation of the fuzzy c-means clustering algorithms. *IEEE transactions on pattern analysis and machine intelligence*(2), 248–255.

Chadaga, K., Prabhu, S., Bhat, V., Sampathila, N., Umakanth, S., & Chadaga, R. (2023). A decision support system for diagnosis of covid-19 from non-covid-19 influenza-like illness using explainable artificial intelligence. *Bioengineering*, *10*(4), 439. doi: 10.3390/bioengineering10040439

Chavan, M., Patil, A., Dalvi, L., & Patil, A. (2015). Mini batch k-means clustering on large dataset. *Int. J. Sci. Eng. Technol. Res*, *4*(07), 1356–1358.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Comparative study of regressor and classifier with decision tree using modern tools. (2022). *Materials Today: Proceedings*, *56*, 3571-3576. (First International Conference on Design and Materials) doi: https://doi.org/10.1016/j.matpr.2021.11.635

Das, S., Sultana, M., Bhattacharya, S., Sengupta, D., & De, D. (2023). Xai–reduct: accuracy preservation despite dimensionality reduction for heart disease classification using explainable ai. *The Journal of Supercomputing*, 1–31. doi: 10.1007/s11227-023-05356-3

Dieber, J., & Kirrane, S. (2020). Why model why? assessing the strengths and limitations of lime. *arXiv preprint arXiv:2012.00093*. doi: 10.48550/arXiv.2012.00093

Dietterich, T. G., et al. (2002). Ensemble learning. *The handbook of brain theory and neural networks*, *2*(1), 110–125.

Ekanayake, I., Palitha, S., Gamage, S., Meddage, D., Wijesooriya, K., & Mohotti, D. (2023). Predicting adhesion strength of micropatterned surfaces using gradient boosting models and explainable artificial intelligence visualizations. *Materials Today Communications*, *36*, 106545. Retrieved from doi: https://doi.org/10.1016/j.mtcomm.2023.106545

El-Sappagh, S., Alonso, J. M., Islam, S. R., Sultan, A. M., & Kwak, K. S. (2021). A multilayer multimodal detection and prediction model based on explainable artificial intelligence for alzheimer's disease. *Scientific reports*, *11*(1), 2660. doi: 10.1038/s41598-021-82098-3

Faisal, A. F., Rahman, A., Habib, M. T. M., Siddique, A. H., Hasan, M., & Khan, M. M. (2022). Neural networks based multivariate time series forecasting of solar radiation using meteorological data of different cities of bangladesh. *Results in Engineering*, *13*, 100365. Retrieved from doi: https://doi.org/10.1016/j.rineng.2022.100365

Groß, J. (2003). *Linear regression* (Vol. 175). Springer Science & Business Media.

Hase, P., & Bansal, M. (2020). Evaluating explainable ai: Which algorithmic explanations help users predict model behavior? *arXiv preprint arXiv:2005.01831*. doi: 10.18653/v1/2020.acl-main.491

Holzinger, A., Saranti, A., Molnar, C., Biecek, P., & Samek, W. (2020). Explainable ai methods-a brief overview. In *International workshop on extending explainable ai beyond deep models and classifiers* (pp. 13–38). doi: 10.1007/978-3-031-04083-2˙2

Holzinger, A., Saranti, A., Molnar, C., Biecek, P., & Samek, W. (2022). Explainable ai methods - a brief overview. In A. Holzinger, R. Goebel, R. Fong, T. Moon, K.-R. Müller, & W. Samek (Eds.), *xxai - beyond explainable ai: International workshop, held in conjunction with icml 2020, july 18, 2020, vienna, austria, revised and extended papers* (pp. 13–38). Cham: Springer International Publishing. doi: 10.1007/978-3-031-04083-2˙2

Hourly day-ahead solar irradiance prediction using weather forecasts by lstm. (2018). *Energy*, *148*, 461-468. doi: https://doi.org/10.1016/j.energy.2018.01.177

Improving time series prediction of solar irradiance after sunrise: Comparison among three methods for time series prediction. (2017). *Solar Energy*, *149*, 294-301. doi: https://doi.org/10.1016/j.solener.2017.04.020

Irshad, K., Islam, N., Gari, A. A., Algarni, S., Alqahtani, T., & Imteyaz, B. (2023). Arithmetic optimization with hybrid deep learning algorithm based solar radiation prediction model. *Sustainable Energy Technologies and Assessments*, *57*, 103165. Retrieved from doi: https://doi.org/10.1016/j.seta.2023.103165

Kadiyala, A., & Kumar, A. (2018). Applications of python to evaluate the performance of bagging methods. *Environmental Progress & Sustainable Energy*, *37*(5), 1555–1559.

Kamal, M. S., Northcote, A., Chowdhury, L., Dey, N., Crespo, R. G., & Herrera-Viedma, E. (2021). Alzheimer's patient analysis using image and gene expression data and explainable-ai to present associated genes. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–7. doi: 10.1109/TIM.2021.3107056

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, *30*.

Khorasani, M., Abdou, M., & Hernández Fernández, J. (2022). Streamlit basics. , 31–62.

Kumari, P., & Toshniwal, D. (2021). Deep learning models for solar irradiance forecasting: A comprehensive review. *Journal of Cleaner Production*, *318*, 128566. Retrieved from doi: https://doi.org/10.1016/j.jclepro.2021.128566

Kuzlu, M., Cali, U., Sharma, V., & Güler, Ö. (2020). Gaining insight into solar photovoltaic power generation forecasting utilizing explainable artificial intelligence tools. *IEEE Access*, *8*, 187814–187823. doi: 10.1109/ACCESS.2020.3031477

Kwon, H., Park, J., & Lee, Y. (2019). Stacking ensemble technique for classifying breast cancer. *Healthcare informatics research*, *25*(4), 283–288.

Lee, Y., Oh, J., & Kim, G. (2020). Interpretation of load forecasting using explainable artificial intelligence techniques. *The Transactions of the Korean Institute of Electrical Engineers*, *69*(3), 480–485.

Liang, X., & Jacobucci, R. (2020). Regularized structural equation modeling to detect measurement bias: Evaluation of lasso, adaptive lasso, and elastic net. *Structural Equation Modeling: A Multidisciplinary Journal*, *27*(5), 722-734. doi: 10.1080/10705511.2019.1693273

McDonald, G. C. (2009). Ridge regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, *1*(1), 93–100.

Misheva, B. H., Osterrieder, J., Hirsa, A., Kulkarni, O., & Lin, S. F. (2021). Explainable ai in credit risk management. *arXiv preprint arXiv:2103.00949*. doi: 10.2139/ssrn.3795322

Miskat, M. I., Sarker, P., Chowdhury, H., Chowdhury, T., Rahman, M. S., Hossain, N., ... Sait, S. M. (2023). Current scenario of solar energy applications in bangladesh: Techno-economic perspective, policy implementation, and possibility of the integration of artificial intelligence. *Energies*, *16*(3). Retrieved from doi: 10.3390/en16031494

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *18*(6), 275–285.

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, *7*, 21.

Nonparametric bayesian-based recognition of solar irradiance conditions: Application to the generation of high temporal resolution synthetic solar irradiance data. (2019). *Solar Energy*, *182*, 462-479. doi: https://doi.org/10.1016/j.solener.2019.02.052

A novel machine learning approach for solar radiation estimation. (2023). *Sustainability*, *15*(13), 10609-10609. doi: 10.3390/su151310609

Pavlyshenko, B. (2018). Using stacking approaches for machine learning models. In *2018 ieee second international conference on data stream mining & processing (dsmp)* (pp. 255–258).

Peng, K., Leung, V. C., & Huang, Q. (2018). Clustering approach based on mini batch kmeans for intrusion detection system over big data. *IEEE Access*, *6*, 11897–11906.

Pierrot, A., & Goude, Y. (2011). Short-term electricity load forecasting with generalized additive models. *Proceedings of ISAP power*, *2011*.

Polikar, R. (2012). Ensemble learning. *Ensemble machine learning: Methods and applications*, 1–34.

Prasad, S. S., Deo, R. C., Salcedo-Sanz, S., Downs, N. J., Casillas-Pérez, D., & Parisi, A. V. (2023). Enhanced joint hybrid deep neural network explainable artificial intelligence model for 1-hr ahead solar ultraviolet index prediction. *Computer Methods and Programs in Biomedicine*, *241*, 107737. Retrieved from doi: https://doi.org/10.1016/j.cmpb.2023.107737

Prediction of global solar irradiation using hybridized k-means and support vector regression algorithms. (2019). *Renewable Energy Focus*, *29*, 78-93. doi: https://doi.org/10.1016/j.ref.2019.03.003

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, *31*.

Rozemberczki, B., Watson, L., Bayer, P., Yang, H.-T., Kiss, O., Nilsson, S., & Sarkar, R. (2022). The shapley value in machine learning. *arXiv preprint arXiv:2202.05594*. doi: 10.48550/arXiv.2202.05594

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), e1249.

Sarp, S., Kuzlu, M., Cali, U., Elma, O., & Güler, (2021, 03). An interpretable solar photovoltaic power generation forecasting approach using an explainable artificial intelligence tool.. doi: 10.1109/ISGT49243.2021.9372263

Schapire, R. E. (2003). The boosting approach to machine learning: An overview. *Nonlinear estimation and classification*, 149–171.

Seber, G. A., & Lee, A. J. (2012). *Linear regression analysis*. John Wiley & Sons.

Sevas, M. S., Tur Santona, C. F., & Sharmin, N. (2023). Ensemble machine-learning model for solar radiation prediction using explainable ai. In *2023 14th international conference on computing communication and networking technologies (icccnt)* (p. 1-6). doi: 10.1109/ICCCNT56998.2023.10307694

Sim, T., Choi, S., Kim, Y., Youn, S. H., Jang, D.-J., Lee, S., & Chun, C.-J. (2022). explainable ai (xai)-based input variable selection methodology for forecasting energy consumption. *Electronics*, *11*(18), 2947. doi: 10.3390/electronics11182947

Sinaga, K. P., & Yang, M.-S. (2020). Unsupervised k-means clustering algorithm. *IEEE access*, *8*, 80716–80727.

Solar radiation forecasting using machine learning and ensemble feature selection. (2022). *Energies*, *15*(19), 7049-7049. doi: 10.3390/en15197049

Sushanth, K., Mishra, A., Mukhopadhyay, P., & Singh, R. (2023). Near-real-time forecasting of reservoir inflows using explainable machine learning and short-term weather forecasts. *Stochastic Environmental Research and Risk Assessment*, 1–21. doi: 10.1007/s00477-023-02489-y

(n.d.-a). ([Online; accessed 2023-08-11])

(n.d.-b). ([Online; accessed 2023-08-11])

Utama, C., Meske, C., Schneider, J., Schlatmann, R., & Ulbrich, C. (2023). Explainable artificial intelligence for photovoltaic fault detection: A comparison of instruments. *Solar Energy*, *249*, 139-151. Retrieved from doi: https://doi.org/10.1016/j.solener.2022.11.018

Venu, K., Indhu Prakash, K. V., Jayaram, S., Karan, N. S., Raja, M., & Renu, K. (2023). Solar radiation prediction using machine learning model. In *2023 international conference on sustainable computing and data communication systems (icscds)* (p. 52-57). doi: 10.1109/ICSCDS56580.2023.10104904

Wahbah, M., EL-Fouly, T. H., & Zahawi, B. (2020). Gaussian mixture model for estimating solar irradiance probability density. In *2020 ieee electric power and energy conference (epec)* (pp. 1–6).

Wang, H., Li, G., & Tsai, C.-L. (2007, 01). Regression coefficient and autoregressive order shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *69*(1), 63-78. doi: 10.1111/j.1467-9868.2007.00577.x

Weber, C. M., Ray, D., Valverde, A. A., Clark, J. A., & Sharma, K. S. (2022). Gaussian mixture model clustering algorithms for the analysis of high-precision mass measurements. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, *1027*, 166299.

Wu, T., Zhang, W., Jiao, X., Guo, W., & Hamoud, Y. A. (2021). Evaluation of stacking and blending ensemble learning methods for estimating daily reference evapotranspiration. *Computers and Electronics in Agriculture*, *184*, 106039.

Yadav, T., Mahalle, P., Sathe, S., & Anerao, P. (2023). Sonar signal prediction using explainable ai for iot environment check for updates. *Next Generation Systems and Networks: Proceedings of BITS EEE CON 2022*, *641*, 209. doi: 10.1007/978-981-99-0483-9

Zhang, Z., Damiani, E., Al Hamadi, H., Yeun, C. Y., & Taher, F. (2022). Explainable artificial intelligence to detect image spam using convolutional neural network. In *2022 international conference on cyber resilience (iccr)* (pp. 1–5). doi: 10.48550/arXiv.2209.03166

Zhao, X., Huang, W., Huang, X., Robu, V., & Flynn, D. (2021, 27–30 Jul). Baylime: Bayesian local interpretable model-agnostic explanations. In C. de Campos & M. H. Maathuis (Eds.), *Proceedings of the thirty-seventh conference on uncertainty in artificial intelligence* (Vol. 161, pp. 887–896). PMLR. Retrieved from