



Faculty of Computer Science and Information Technology

***PREDICTION OF SOLAR ENERGY PRODUCTION USING
REGRESSION TECHNIQUES***

THASHEELA A/P MURGIS

**Bachelor of Computer Science with Honours
(Information Systems)**

2024

PREDICTION OF SOLAR ENERGY PRODUCTION USING REGRESSION TECHNIQUES

THASHEELA A/P MURGIS

This project is submitted in partial fulfillment of the
requirements for the degree of
Bachelor of Computer Science and Information Technology

Faculty of Computer Science and Information
Technology
UNIVERSITI MALAYSIA
SARAWAK

2024

RAMALAN PENGELUARAN TENAGA SOLAR MENGGUNAKAN TEKNIK REGRESI

THASHEELA A/P MURGIS

Project ini merupakan salah satu keperluan untuk
Ijazah Sarjana Muda Sains Komputer dan Teknologi
Maklumat

Fakulti Sains Komputer dan Teknologi Maklumat
UNIVERSITI MALAYSIA SARAWAK

2024

DECLARATION

I hereby declare that this research, inclusive with all related content is none other than that of my own work, with consideration of the exception of research-based information which were adapted and extracted from relevant source, which have evidently been stated respectively.

Signed,

THASHEELA

Thasheela A/P Murgis,

81378

Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak

28th July 2025

ACKNOWLEDGEMENT

First and foremost, I would like to extend my heartfelt gratitude to my final-year supervisor, Dr. Stephanie Chua Hui Li, for her unwavering guidance and support throughout this project. Her consistent encouragement and support have been invaluable and having her as my Data Mining lecturer has greatly enhanced my understanding and ability to apply the knowledge required for this project. I would also like to express my appreciation to Ts. Dr. Sarah Flora Samson Juan for her insightful guidance and practical advice on effectively completing this project. Additionally, I am deeply thankful to my friends and coursemates for their assistance. Their willingness to dedicate their time to discussions and provide feedback on my final-year project has been incredibly helpful. Also, I am profoundly grateful to my beloved partner and family for their unwavering encouragement and support, both emotionally and financially, which have been instrumental in enabling me to complete this project. Last but not the least, thank you Lord for everyone and everything that you arranged to assist me in making this project successful.

ABSTRACT

Accurate solar energy production forecasting is essential for optimizing energy usage, storage, and integration with the power grid. This project leverages machine learning regression techniques to develop a predictive model using weather data such as temperature, humidity, cloud cover, and wind speed. Historical and real-time weather data are processed and analyzed to train and evaluate various regression models, including Linear Regression, Random Forest, and Support Vector Regression, based on metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2). The Knowledge Discovery in Databases (KDD) methodology guides the project's structured approach, encompassing data selection, pre-processing, transformation, and modeling. This study identifies key weather attributes influencing solar energy production and demonstrates the application of machine learning to provide reliable predictions. The findings contribute to the broader adoption of solar energy by enhancing predictability, reducing dependency on fossil fuels, and supporting global sustainability efforts. The project's outcomes aim to empower users with actionable insights for efficient energy management and decision-making.

ABSTRAK

Ramalan pengeluaran tenaga solar yang tepat adalah penting untuk mengoptimumkan penggunaan tenaga, penyimpanan, dan integrasi dengan grid kuasa. Projek ini menggunakan teknik regresi pembelajaran mesin untuk membangunkan model ramalan berdasarkan data cuaca seperti suhu, kelembapan, litupan awan, dan kelajuan angin. Data cuaca sejarah dan masa nyata diproses dan dianalisis untuk melatih dan menilai pelbagai model regresi, termasuk Regresi Linear, Random Forest, dan Support Vector Regression, berdasarkan metrik seperti Ralat Mutlak Purata (MAE), Akar Min Kuasa Dua (RMSE), dan R-kuasa Dua (R^2). Metodologi *Knowledge Discovery in Databases* (KDD) digunakan untuk membimbing pendekatan berstruktur projek ini, merangkumi pemilihan data, prapemprosesan, transformasi, dan pemodelan. Kajian ini mengenal pasti atribut cuaca utama yang mempengaruhi pengeluaran tenaga solar dan menunjukkan aplikasi pembelajaran mesin untuk menyediakan ramalan yang boleh dipercayai. Penemuan ini menyumbang kepada penerapan tenaga solar yang lebih meluas dengan meningkatkan kebolehpercayaan, mengurangkan kebergantungan pada bahan api fosil, dan menyokong usaha kemampanan global. Hasil projek ini bertujuan untuk memperkasakan pengguna dengan pandangan yang boleh diambil tindakan bagi pengurusan tenaga dan pembuatan keputusan yang cekap.

TABLE OF CONTENTS

ACKNOWLEDGEMENT.....	i
ABSTRACT.....	ii
ABSTRAK.....	iii
LIST OF FIGURES	6
LIST OF EQUATIONS	6
LIST OF TABLES.....	7
Chapter 1	1
Introduction.....	1
1.1 Introduction	1
1.2 Problem Statement	1
1.3 Scope	1
1.4 Objectives.....	2
1.5 Methodology	2
1.5.1 Domain Understanding.....	2
1.5.2 Data Selection.....	3
1.5.3 Data Pre-processing	3
1.5.4 Data Transformation	3
1.5.5 Data Mining/Modelling	3
1.5.6 Model Evaluation	4
1.6 Significance of Project	4
1.7 Project Schedule.....	5
1.8 Expected Outcome	7
1.9 Thesis Outline	7
Chapter 1.....	7
Chapter 2.....	7
Chapter 3.....	7
Chapter 4.....	7
Chapter 5.....	8
Chapter 6.....	8
1.10 Summary	8
Background and Literature Review	9
2.1 Introduction	9
2.2 Solar Energy	9
2.3 Comparison of Previous Work	11

2.4 Model for Comparison	16
2.4.1 Linear Regression	16
2.4.2 Support Vector Machine (SVM)	16
2.4.3 Random Forest.....	17
2.5 Chosen Evaluation Metrics.....	17
2.5.1 Mean Absolute Error (MAE)	17
2.5.2 Root Mean Squared Error (RMSE)	17
2.5.3 R-squared (R^2)	18
2.6 Tools.....	18
2.7 Chapter Summary.....	19
Chapter 3	21
Research Methodology	21
3.1 Introduction	21
3.2 Process Explanation	21
3.2.1. Domain Understanding.....	22
3.2.2 Data Selection.....	22
3.2.3 Data Pre-Processing.....	22
3.2.4 Data Transformation	23
3.2.4 Data Mining/Modelling	24
3.2.5 Model Evaluation	25
3.3 Chapter Summary.....	27
CHAPTER 4.....	28
IMPLEMENTATION	28
4.1 Introduction	28
4.2 Implementation.....	28
4.2.1 Data Selection.....	28
4.2.2 Data Preprocessing.....	28
4.2.3 Data Transformation.....	31
4.2.4 Data Mining/Modelling.....	33
4.2.5 Model Evaluation	35
4.3 Summary	36
CHAPTER 5.....	37
RESULTS AND DISCUSSION	37
5.1 Introduction	37
5.2 Visualization of Actual and Predicted Solar Energy	37

5.3 Final Correlation Analysis.....	40
5.4 Model Performance and Calibration	41
5.5 Interpretation and Limitations.....	41
5.5 Automation and Real-Time Dashboard Refresh	41
5.6 Evaluation Metrics and Calibration Explanation	42
5.7 Chapter Summary.....	43
CHAPTER 6.....	44
CONCLUSION AND FUTURE WORK	44
6.1 Introduction	44
6.2 Contribution	44
6.3 Limitations	44
6.3 Recommendations for Future Work	45
6.4 Summary	45

LIST OF FIGURES

Figure 1.1: Diagram of Project Methodology	2
Figure 1.2 Overall Gantt Chart for Final Year Project	6
Figure 3.3 Mock-up of Power BI dashboard	26
Figure 4.1 Dataset before cleaning	29
Figure 4.2 Dataset after cleaning	29
Figure 4.3 Further preprocessing conducted	30
Figure 4.4 Temperature readings converted	30
Figure 4.5 Correlation matrix.	31
Figure 4.6 Splitting Python Script	33
Figure 4.7 Training the models	34
Figure 5.1 Script to fetch data from OpenMeteo API	37
Figure 5.2: Power BI dashboard showing average actual and predicted solar energy production, with environmental feature cards.	39
Figure 5.3: Power BI dashboard showing prediction for 24 hours resulting a higher MAPE.	39
Figure 5.4: New Correlation Matrix	40
Figure 5.5 Automation Flow	42

LIST OF EQUATIONS

Equation 3.1: Mean Absolute Error	25
Equation 3.2: Root Mean Squared Error	25
Equation 3.3: R-Squared	25
Equation 5.1: Equation used to calculate actual	37

LIST OF TABLES

Table 2.1: Comparison of Previous Work	11
Table 3.1: Hardware and Software Requirement	24
Table 4.1: Training Result	37

Chapter 1

Introduction

1.1 Introduction

Solar energy is a renewable energy source that is increasingly being adopted globally as part of efforts to reduce reliance on fossil fuels and mitigate environmental impacts. However, solar energy production is highly dependent on environmental conditions such as temperature, cloud cover, and sunlight duration. Predicting solar energy production accurately is critical to managing energy resources efficiently, particularly for optimizing energy usage, storage, and grid integration.

Machine learning models, especially regression techniques, offer powerful tools for predicting solar energy production based on historical weather data. By leveraging data on temperature, humidity, cloud cover, and other weather factors, these models can provide accurate forecasts, helping users make data-driven decisions to optimize energy use. This project aims to develop a predictive system using regression techniques for solar energy production, with the results visualized in Power BI and real-time notifications set up through Power Automate.

1.2 Problem Statement

Solar energy production is highly dependent on weather conditions such as sunlight intensity, temperature, humidity, wind speed, and cloud cover. However, the inherent variability and uncertainty of weather patterns pose significant challenges in accurately forecasting solar energy output. This project aims to develop a robust prediction model that uses weather variables to accurately forecast solar energy production. By leveraging advanced machine learning techniques and historical weather and solar energy production data, the model will provide precise and reliable forecasts.

1.3 Scope

This project focuses on predicting solar energy production using real-time weather data. The scope includes:

- Using historical solar energy production data and weather data (e.g., temperature, humidity, cloud cover).
- Applying multiple regression techniques to create and compare predictive models.

- Integrating the selected model with Power BI to provide visual insights into solar energy forecasts.
- Setting up Power Automate to send notifications when predicted energy production exceeds or falls below defined thresholds.

1.4 Objectives

The objectives of this project are:

1. To determine the attributes that influence solar energy production.
2. To compare the performances of various regression techniques for predicting solar energy production.
3. To determine the most effective regression model for predicting solar energy production.

1.5 Methodology

The methodologies used in this project are the knowledge discovery in databases (KDD)

as shown in Figure 1.1.

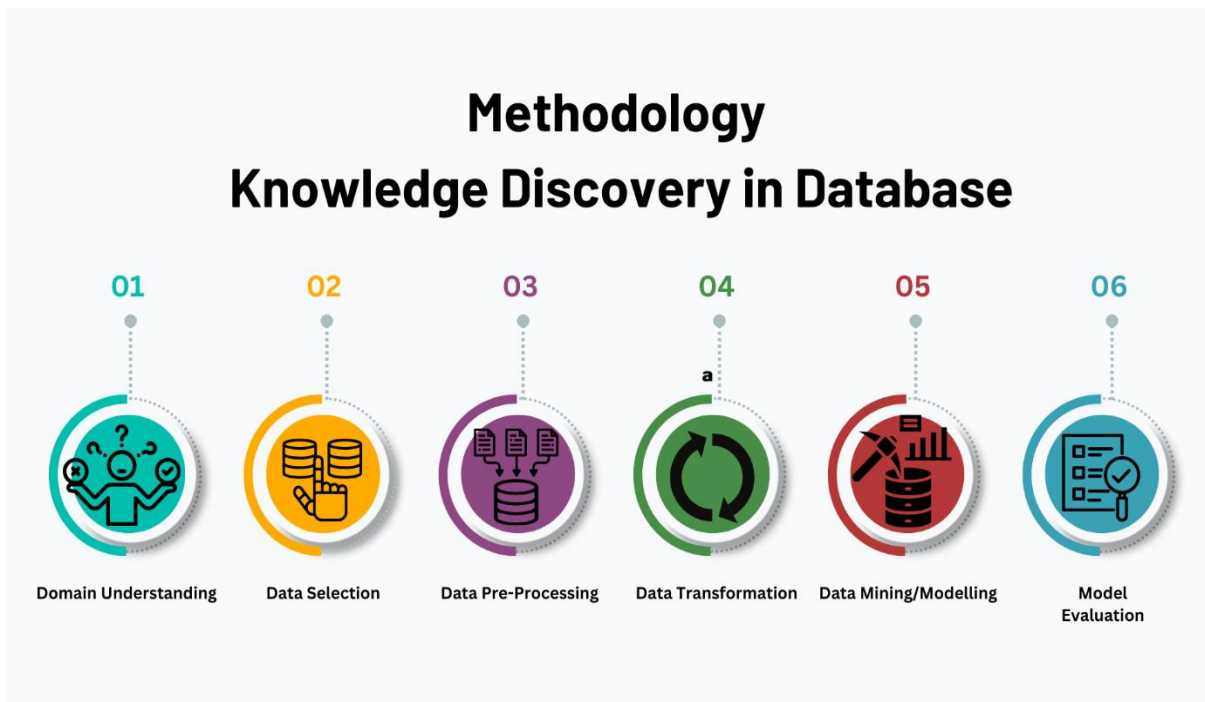


Figure 1.1: Diagram of Project Methodology

1.5.1 Domain Understanding

Supervised machine learning is used to analyse the relationship between weather conditions and solar energy production. Regression analysis is the primary data analysis technique

employed in this project. The goal is to create a predictive model that estimates future solar energy output based on real-time weather data (e.g., temperature, humidity, cloud cover, wind speed). The model will be trained on historical data where the target variable solar energy production is known. Through this process, the model will learn how weather patterns influence energy production and provide accurate predictions.

1.5.2 Data Selection

The first step after understanding the domain is collecting the relevant data. Historical solar energy production data and corresponding weather data (e.g., temperature, humidity, wind speed, cloud cover, and sunlight duration) will be gathered from a Kaggle dataset. Real-time weather data for predictions will be fetched using a weather API. This data will be used to build and test the predictive model for solar energy production.

1.5.3 Data Pre-processing

Data pre-processing techniques will be applied to ensure the data is ready for modelling. This includes handling missing values, normalizing continuous variables, and converting any categorical weather variables (e.g., weather condition descriptions) into numerical form. Feature selection will also be applied to identify the most important weather factors influencing solar energy production. By reducing irrelevant features, the model's performance can be improved in terms of accuracy and efficiency.

1.5.4 Data Transformation

Before the model is trained, data transformation techniques will be used to prepare the data for analysis. For instance, scaling numeric variables (such as temperature and wind speed) ensures that all features are on a similar scale. Additionally, temporal features like time of day, day of the year, and seasonal factors may be derived from the date and time fields in the weather data. These transformed variables will help the model better capture time-dependent patterns in solar energy production.

1.5.5 Data Mining/Modelling

The dataset will be used to train several machine learning regression techniques to predict solar energy production based on weather inputs. Various regression techniques will be explored.

Each technique will be trained using historical weather and energy production data to capture patterns between weather factors and energy output. The performance of the models will be

tested and compared to determine which provides the most accurate and reliable solar energy predictions.

1.5.6 Model Evaluation

Once the models are trained, their performance will be evaluated using common regression evaluation metrics, including:

- **Mean Absolute Error (MAE)**
- **Root Mean Squared Error (RMSE)**
- **R-squared (R^2)**

These metrics will help assess the accuracy of the predictions and the overall effectiveness of each model. The best-performing model will be selected for deployment, where it will use real-time weather data to predict solar energy production. The results will be integrated into Power BI for visualization, and Power Automate will be used to send real-time notifications if predicted energy production exceeds or falls below specific thresholds.

1.6 Significance of Project

This project has significant implications for both energy users and the environment. By providing accurate predictions of solar energy production, the model will enable users to optimize their energy consumption, storage, and integration with the power grid. The ability to forecast solar energy production based on real-time weather conditions will empower users to make proactive, data-driven decisions regarding when to store excess energy, reduce consumption, or switch to alternative power sources. This enhances the overall efficiency of energy management, reduces reliance on external energy supplies, and can lead to significant cost savings.

Additionally, the real-time integration of weather data with machine learning models and visualization tools like Power BI will allow users to monitor energy trends, respond quickly to changes, and prevent energy shortages or overproduction. This not only optimizes energy use for individuals and businesses but also contributes to a more stable and reliable renewable energy infrastructure.

From an environmental perspective, the project promotes the adoption of renewable energy by making solar power more predictable and reliable. By encouraging optimal use of solar energy, it reduces dependency on fossil fuels, thus lowering greenhouse gas emissions and contributing

to global sustainability efforts. This project also supports the broader transition to cleaner energy by making solar energy systems more attractive and accessible to users, helping to mitigate climate change and preserve natural resources for future generations.

1.7 Project Schedule

Figure 1.2 shows the Gantt Chart planned for Final Year Project.

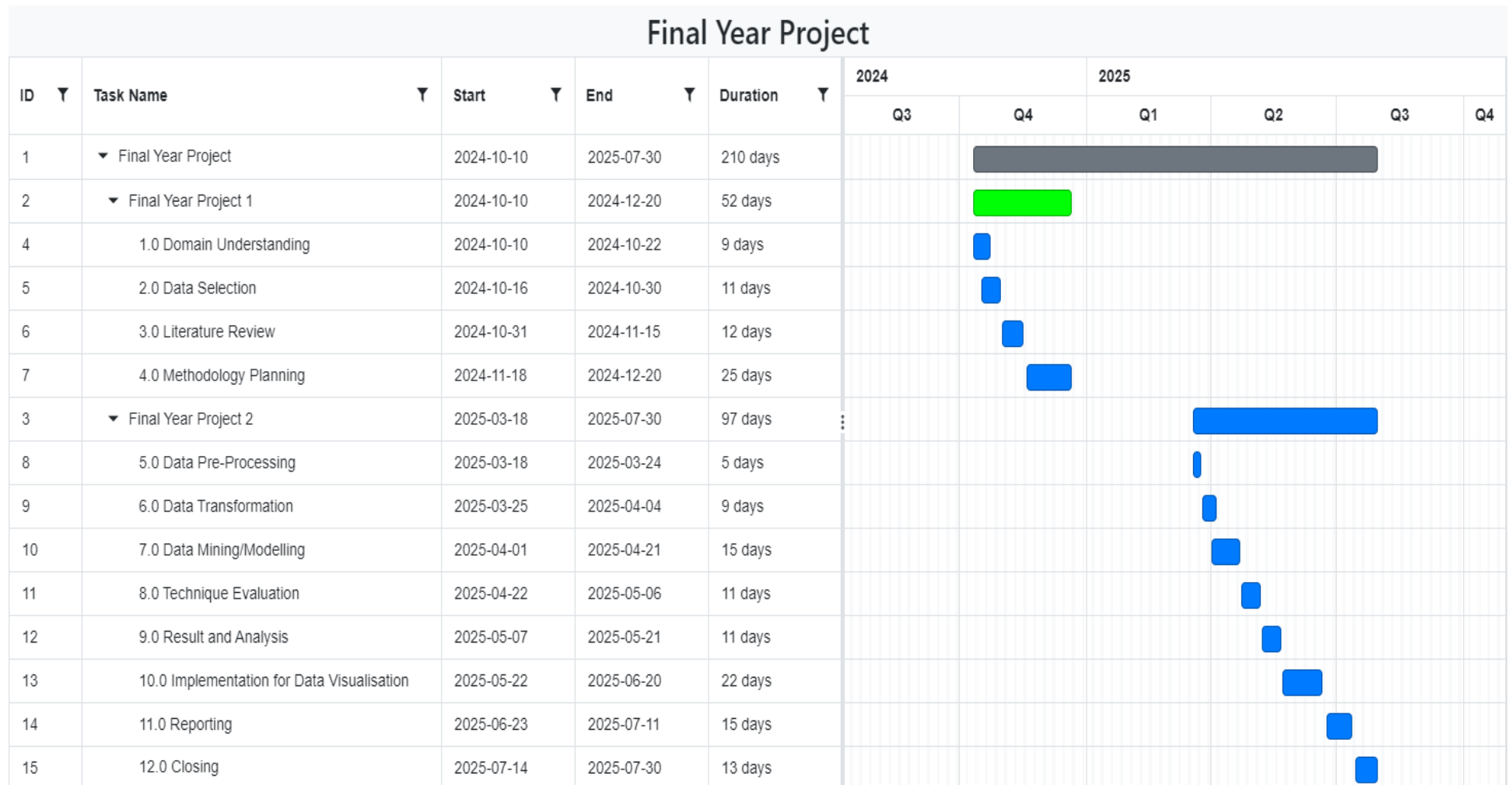


Figure 1.2 Overall Gantt Chart for Final Year Project

1.8 Expected Outcome

The outcome of this project is to determine the best regression technique used for predicting solar energy production. The study aims to evaluate and compare the performance of these models in accurately forecasting solar energy output based on weather data. The best-performing regression model will be selected and deployed to provide real-time predictions of solar energy production, which can then be visualized through Power BI and integrated with Power Automate for real-time notifications.

1.9 Thesis Outline

Chapter 1

Chapter 1 provides a general overview of the project, including the problem statement, objectives, project scope, methodology, significance of the project, project schedule, expected outcomes, thesis outline, and a summary.

Chapter 2

Chapter 2 discusses the background of solar energy production and the influence of weather conditions on energy generation. The literature review also covers related work in the application of machine learning and regression techniques for energy prediction, as well as the use of real-time data and predictive analytics in energy management.

Chapter 3

Chapter 3 describes the methodology used to address the problem. This section explains the regression techniques employed to predict solar energy production, details of the training environment, data preparation processes, and the procedures for testing and comparing the performance of different regression models.

Chapter 4

Chapter 4 provides a detailed explanation of the data mining and modelling implementation. It includes the steps taken for data collection, pre-processing, feature selection, and model training. The chapter also contains step-by-step instructions on how to integrate real-time weather data, and how the regression models were developed and tested.

Chapter 5

Chapter 5 presents the results of the regression models for solar energy prediction. A comparative study will be conducted to evaluate the performance of different regression techniques. The best-performing model will be selected and analysed in detail, with discussions on its accuracy and real-time prediction capabilities.

Chapter 6

Chapter 6 summarizes the project, outlines its limitations, and discusses potential areas for future work, such as expanding the model to predict solar energy across multiple locations, or incorporating additional weather factors and real-time optimization strategies.

1.10 Summary

This chapter discusses the increasing global use of solar energy as a renewable resource to reduce dependence on fossil fuels and its environmental impact. Accurate solar energy production predictions are crucial due to the dependence on weather conditions. Machine learning, specifically regression models, can help predict solar energy output based on weather data. The project aims to develop a prediction model using regression techniques, integrate the results with Power BI for visualization, and set up notifications via Power Automate to address the unpredictability of solar energy due to weather variability, which makes it difficult for potential users to adopt solar power systems.

Key objectives include determining factors that influence solar energy production, comparing regression techniques, and selecting the best model for accurate prediction. The methodology follows the Knowledge Discovery in Databases (KDD) process, involving data collection, pre-processing, transformation, modelling, and evaluation. Several regression techniques such as Linear Regression, Random Forest, and Support Vector Regression will be tested.

The project's significance lies in improving energy management by providing accurate solar energy forecasts, thereby enabling optimized energy use, reduced reliance on external power sources, and environmental benefits by promoting renewable energy. Expected outcomes include selecting the best regression model for solar energy prediction, which will be visualized in Power BI and integrated with real-time alerts. The chapter also outlines the project's schedule and provides an overview of the thesis structure.

Chapter 2

Background and Literature Review

2.1 Introduction

Solar energy is a renewable resource that plays a crucial role in sustainable energy solutions. It is essential for powering various applications, especially in rechargeable wireless sensor networks, where reliable solar radiation knowledge is vital for optimal management and deployment planning. Recent advancements in machine learning have significantly improved solar irradiance prediction. Various machine learning models have been applied to historical solar intensity data to enhance forecasting accuracy.

Accurate forecasting of solar energy is essential for effective management and security in the energy sector. It involves predicting solar radiation and converting it into power using photovoltaic (PV) panel characteristics. Forecasting methods can be categorized into indirect and direct models, with various approaches including statistical, physical, artificial intelligence (AI), and hybrid methods. Machine learning (ML) techniques have gained prominence in solar energy prediction due to their high learning and regression capabilities.

This chapter discusses the related work that researchers have done in recent years then made a comparative analysis of various research papers to acquire new knowledge. The year range of the research papers found for this project is within 5 years, with the oldest being 2021 and the latest being 2024. Most of the research papers found were looking at using different techniques to predict solar energy production. Before starting to discuss the related works, having a fundamental knowledge about the background of knowledge of solar energy and regression techniques is a must.

2.2 Solar Energy

Solar energy is pivotal in advancing sustainable development and achieving global clean energy objectives. It is primarily harnessed through two technologies: photovoltaic (PV) systems, which convert sunlight directly into electricity, and concentrated solar power (CSP), which uses mirrors or lenses to focus sunlight to generate thermal energy for power turbines. Both PV and CSP produce clean, emissions-free electricity that can help reduce greenhouse gas emissions (Izam et al., 2022).

The rapid development of solar technology has been driven by environmental and economic factors. Between 2010 and 2020, the cost of generating electricity from solar PV decreased by

85%, making it competitive with, and often cheaper than, traditional fossil fuel sources (International Renewable Energy Agency [IRENA], 2020). This significant cost reduction has led to a substantial increase in global installed capacity, which grew more than tenfold from 2008 to 2018 (IRENA, 2020).

The widespread adoption of solar energy aligns with international commitments, such as the Paris Agreement and the United Nations' Sustainable Development Goals (SDGs), particularly Goal 7, which aims to ensure access to affordable, reliable, sustainable, and modern energy for all (United Nations, n.d.).

Despite its benefits, challenges remain in solar energy deployment. Factors like weather conditions can affect efficiency, and effective storage solutions are needed to manage energy supply during periods without sunlight. Innovations in battery technology and hybrid systems are addressing these limitations, enhancing the reliability and adaptability of solar energy for continuous, large-scale use (International Energy Agency [IEA], n.d.).

In summary, solar energy is a vital component of sustainable development. By replacing fossil fuels with clean, renewable energy, solar power not only helps reduce carbon emissions but also offers significant economic and social benefits. The global push toward sustainable energy systems will continue to drive the expansion of solar technology, ensuring that it plays an increasingly prominent role in our energy landscape and contributes to a healthier, more sustainable future (Maka & Alabid, 2022).

2.3 Comparison of Previous Work

Table 2.1: Comparison of Previous Work

Citation	Description	Model/Algorithm Used	Evaluation Metrics	Results
(Narvaez et al., 2023)	Developed machine learning methodology for improving satellite-based solar radiation forecasting using site-adaptation and forecasting models.	• Random Forest • Neural Networks • AdaBoost • Linear Regression • LSTM • GRU	• R^2 • RMSE • MAE	Random Forest achieved R^2 of 0.96 for site adaptation. LSTM was best for daily and weekly forecasts, with R^2 of 0.96 and 0.88, respectively.
(Ledmaoui et al., 2023)	Conducted a comparative study of machine learning models for daily solar energy production forecasting using data from Benguerir, Morocco.	• Support Vector Regression (SVR) • Artificial Neural Network (ANN) • Decision Tree (DT) • Random Forest (RF)	• RMSE • MAE • MASE • R^2	ANN performed the best with the lowest RMSE and MAE and highest R^2 , showcasing its robustness for solar energy forecasting.

		• Generalized Additive Model (GAM) • Extreme Gradient Boosting (XGBoost)		
(Sahin et al., 2023)	Predicted efficiency of a 500 kWp solar power plant in Igdir, Turkey, considering both environmental and solar module characteristics.	• Artificial Neural Network (ANN) • Multiple Linear Regression • Principal Component Analysis (PCA)	• R^2 • RMSE	ANN outperformed Linear Regression with higher R^2 values. PCA effectively reduced data dimensions while retaining prediction accuracy.
(Erten & Aydilek, 2022)	Explored four regression models for solar power prediction and examined feature extraction's impact using Principal Component Analysis (PCA).	• Linear Regression • Logistic Regression • Lasso Regression • Elastic Net Regression	• R^2 • RMSE	Elastic Net Regression emerged as the most effective model, achieving superior RMSE and R^2 values over other regression models.

(Munawar et al., 2023)	Proposed a hybrid model using Linear Regression (LR) and a Knowledge-Based Neural Network (KBNN) to predict solar energy, addressing linear and nonlinear relationships in the data.	• Linear Regression (LR) • Knowledge-Based Neural Network (KBNN)	• R^2 • RMSE	KBNN achieved the highest R^2 (99%) and lowest RMSE (44), outperforming LR for both linear and nonlinear datasets.
(Lee et al., 2020)	Investigated the reliability of ensemble learning-based models for short-term solar irradiance prediction, focusing on combining weak regressors for improved accuracy.	• Boosted Trees • Bagged Trees • Random Forest (RF) • Generalized Random Forest (GRF)	• RMSE • MAE • R^2	Ensemble learning methods outperformed individual models, achieving more consistent and reliable predictions across different datasets.
(Komar et al., 2024)	Presented a regression-based method for real-time solar power plant efficiency forecasting, evaluating models for	• Gradient Boosting Regressor • ANN • Time-Series Models	• R^2 • RMSE	Gradient Boosting Regressor demonstrated the best performance, providing accurate real-time forecasts essential for

	different forecasting horizons and their integration into Smart Grid systems.			integrating solar power plants into energy management systems.
(Jebli et al., 2021)	Proposed a machine learning approach for solar energy prediction guided by Pearson correlation to select the most relevant meteorological inputs.	• Linear Regression (LR) • Random Forest (RF) • Artificial Neural Network (ANN) • Support Vector Regression (SVR)	• R^2 • RMSE	ANN and RF outperformed other models, achieving the highest prediction accuracy for real-time and short-term solar energy forecasting.

Table 2.1 shows the comparisons of previous works. By integrating satellite data with in-situ measurements, Narvaez et al. (2023) demonstrated a machine learning-based approach to enhance solar radiation predictions. The study showed how machine learning methods can be used to improve solar radiation forecasts in a variety of geographic settings. With an R^2 of 0.96, Random Forest proved to be the most effective model for site adaptation, greatly improving forecasting accuracy. For daily and weekly forecasts, Long Short-Term Memory (LSTM) models performed best, with R^2 values of 0.96 and 0.88, respectively.

Six machine learning models were thoroughly compared by Ledmaoui et al. (2023) to predict Morocco's solar energy production. The study found that the Artificial Neural Network (ANN) had the best R^2 and the lowest error rates (RMSE, MAE), making it the most accurate predictive model. The study offered useful insights for practical applications by determining the best ANN algorithm configurations, especially for predictive maintenance and photovoltaic power plant optimization.

The efficiency forecast of a 500 kWp solar plant was investigated by Sahin et al. (2023) by taking solar module and environmental variables into account. With higher R^2 values and improved RMSE, Artificial Neural Networks (ANNs) outperformed Multiple Linear Regression in terms of predictive power. By lowering the dimensionality of the data without sacrificing performance, Principal Component Analysis (PCA) was used to significantly improve prediction accuracy. The results help to maximize the efficiency of solar power plants and energy investment plans.

Elastic Net Regression proved to be the most successful method in Erten & Aydilek's (2022) evaluation for solar power prediction because it was able to strike a compromise between model complexity and accuracy. The study showed that Principal Component Analysis (PCA) feature extraction greatly enhanced model performance. The usefulness of Elastic Net Regression in solar energy forecasting was demonstrated by its superior RMSE and R^2 scores when compared to Linear, Logistic, and Lasso regression techniques.

Munawar et al. (2023) suggested a hybrid method for predicting solar energy that combines Linear Regression (LR) and Knowledge-Based Neural Networks (KBNN). KBNN performed exceptionally well in managing nonlinear relationships, attaining an R^2 of 99% and an RMSE of 44, whereas LR was efficient for linear datasets. The study demonstrated how the model

provides a solid answer for solar energy forecasting by producing accurate predictions with less data.

Lee et al. (2020) assessed ensemble learning techniques for short-term solar irradiance prediction, emphasizing how well they could combine weak regressors. In terms of consistency and dependability, Random Forest, Bagged Trees, and Boosted Trees performed better than individual regressors. The benefits of ensemble learning in raising forecast accuracy across various meteorological datasets were highlighted by the study.

A regression-based method for real-time solar power plant efficiency forecasts was presented by Komar et al. (2024), with a focus on integration with Smart Grid technologies. The most successful was the Gradient Boosting Regressor, which produced incredibly precise forecasts that are necessary for good energy management. The study demonstrated how sophisticated regression models can be used to maximize renewable energy systems.

The most pertinent meteorological inputs were identified by Jebli et al. (2021) using machine learning models for solar energy prediction, which were led by Pearson correlation. In terms of short-term and real-time prediction accuracy, ANN and Random Forest models performed better than Linear Regression and Support Vector Regression. The results underscored the significance of feature selection in improving model performance.

2.4 Model for Comparison

2.4.1 Linear Regression

One fundamental method that sheds light on the connection between predictors and the target variable is linear regression. It frequently serves as a reference model for comparison. Erten and Aydilek (2022), for example, compared linear regression to more sophisticated regression models and demonstrated how well it predicts solar power. Furthermore, Jebli et al. (2021) assessed linear regression's efficacy in conjunction with more intricate models as part of their solar energy forecast study.

2.4.2 Support Vector Machine (SVM)

Support Vector Machines have demonstrated strong performance in capturing nonlinear relationships in solar energy prediction. For example, Ledmaoui et al. (2023) included SVM in their comparative study, showcasing its ability to handle complex data structures. Furthermore,

Jebli et al. (2021) emphasized SVM's role in enhancing prediction accuracy for solar energy forecasting when applied to diverse meteorological inputs.

2.4.3 Random Forest

Random Forest is well known for its accuracy and resilience in regression tasks, especially when it comes to predicting solar energy. Several studies have shown that Random Forest regularly produces good prediction accuracy, beating other models in a variety of settings (Ledmaoui et al., 2023). Random Forest was also proven to be quite effective by Jebli et al. (2021), especially for short-term and real-time solar energy projections. Furthermore, by attaining exceptional performance metrics, Narvaez et al. (2023) demonstrated the usefulness of Random Forest in enhancing site-adaptation and solar radiation projections.

2.5 Chosen Evaluation Metrics

2.5.1 Mean Absolute Error (MAE)

MAE, which can be simply understood as the average absolute difference between observed and anticipated values, is frequently utilized in solar energy prediction studies. To assess model performance, MAE was used in several of the examined articles. To illustrate its applicability in evaluating model accuracy, Ledmaoui et al. (2023) employed MAE as a crucial criterion in their comparative study of machine learning models for solar energy forecasting. Similarly, models such as Random Forest did well in minimizing this measure when Narvaez et al. (2023) used MAE to analyse prediction errors for site adaptation.

2.5.2 Root Mean Squared Error (RMSE)

Many studies choose RMSE because it penalizes large mistakes more severely than MAE, which makes it especially useful for discovering models that manage outliers well. In their assessments, Narvaez et al. (2023) and Ledmaoui et al. (2023) both used RMSE; the former showed that Random Forest could attain low RMSE values in site adaptation tasks, while the latter highlighted the superior performance of ANN in lowering RMSE for daily solar energy production forecasts. Furthermore, RMSE was used to compare regression models by Erten and Aydilek (2022), highlighting its significance in assessing prediction precision.

2.5.3 R-squared (R^2)

R^2 is a crucial measure of model success since it is a common metric for determining the percentage of variance that a model explains. R^2 has been used in numerous research to compare models and choose the most effective predictors. For instance, Random Forest demonstrated its predictive accuracy for site adaptation with a high R^2 of 0.96, as reported by Narvaez et al. (2023). In a similar vein, Ledmaoui et al. (2023) showed how well ANN works to achieve the highest R^2 values among rival models. The function of R^2 in evaluating regression models was further emphasized by Erten and Aydilek (2022).

The chosen evaluation metric encompasses a variety of model evaluation facets. While RMSE highlights the significance of big deviations, R^2 measures the percentage of variability explained by the model, and MAE offers an intuitive estimate of average error. A thorough examination of model performance is ensured by the constant validation of these metrics in the examined literature as crucial instruments for assessing solar energy forecast models.

2.6 Tools

Machine learning tools and libraries in Python will be used in this project to build predictive models. Below are some of the tools used by researchers above:

A. Scikit-learn

Scikit-learn is a widely used open-source Python library for machine learning. It supports essential scientific and mathematical Python libraries, including NumPy and Matplotlib. Scikit-learn includes various classification and regression algorithms, such as Support Vector Regression (SVR), Random Forest (RF), and Gradient Boosting (XGBoost), which have been utilized in several solar energy prediction studies.

B. NumPy

NumPy is a core Python library that provides support for large multidimensional arrays and a vast collection of mathematical functions. It plays a fundamental role in numerical computations required for machine learning and solar energy prediction models

C. Pandas

Pandas is a powerful data manipulation and analysis library built on top of NumPy. It provides data structures such as DataFrames, which are widely used for handling structured data in solar energy forecasting research

D. Matplotlib

Matplotlib is a visualization library for Python that allows for plotting graphs and charts based on numerical data. In many solar energy forecasting studies, researchers have used Matplotlib to generate static plots that illustrate model performance and trends. However, for this study, Power BI will be preferred over Matplotlib for visualization due to its ability to create interactive, real-time dashboards. Power BI enables seamless integration with external data sources, such as weather APIs, making it a more effective tool for demonstrating solar energy predictions and weather conditions side by side.

E. TensorFlow/Keras

Some studies on solar energy prediction have implemented deep learning models using TensorFlow and Keras for improved accuracy. While deep learning models provide high accuracy, this study will focus on regression techniques for better interpretability.

2.7 Chapter Summary

Chapter 2 provides a detailed background and literature review of solar energy production and prediction. It begins by emphasizing the importance of solar energy as a renewable resource essential for sustainable development and energy management. Solar energy technologies such as photovoltaic (PV) systems and concentrated solar power (CSP) have significantly advanced, contributing to global efforts in reducing greenhouse gas emissions and combating climate change. However, challenges related to weather dependency and variability highlight the need for accurate solar energy forecasting to optimize its utilization.

The chapter reviews various forecasting models and approaches, categorizing them into statistical, physical, artificial intelligence (AI), and hybrid methods. Machine learning (ML) techniques, particularly regression models, are noted for their high accuracy and ability to handle complex relationships in solar energy prediction. A comparative analysis of recent studies evaluates models such as Random Forest, Support Vector Machines (SVM), and Artificial Neural Networks (ANN), highlighting their strengths and performance in different scenarios.

The chapter also discusses the chosen models for the project such as Linear Regression, SVM, and Random Forest selected for their proven effectiveness in solar energy forecasting as demonstrated in the reviewed studies. For model evaluation, key metrics such as Mean

Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2) are justified based on their widespread use and relevance in assessing prediction accuracy.

Predicting solar energy is essential for maximizing the use of renewable energy sources, enhancing system stability, and lowering reliance on fossil fuels. The examined research emphasizes that environmental conditions including temperature, humidity, and cloud cover cause variations in solar energy generation. Reducing waste and increasing productivity, accurate forecasting can assist energy providers, companies, and homes in effectively managing power usage and storage. Machine learning models are an appropriate method for this investigation because prior research has shown how well they forecast solar energy. This work aims to improve solar prediction accuracy using regression models considering the growing reliance on renewable energy, supporting sustainability and energy efficiency objectives.

Chapter 3

Research Methodology

3.1 Introduction

The Knowledge Discovery in Databases (KDD) methodology provides a systematic approach to extracting meaningful insights from complex datasets. It consists of five key phases: selection, preprocessing, transformation, data mining, and interpretation/evaluation. This structured process is particularly suitable for this project as it aligns seamlessly with the goals of developing a predictive model for solar energy production.

KDD's iterative nature ensures that the project accommodates the challenges associated with solar energy data, such as handling missing or inconsistent data, engineering features relevant to environmental variables, and rigorously evaluating predictive models. By breaking the methodology into well-defined phases, KDD provides the flexibility to address data quality issues early on while ensuring the results are both accurate and actionable. Additionally, its emphasis on data transformation and model evaluation supports the integration of diverse attributes like weather conditions and energy production into robust predictive analytics. This makes KDD an ideal framework for uncovering patterns and relationships essential for achieving the project's objectives.

3.2 Process Explanation

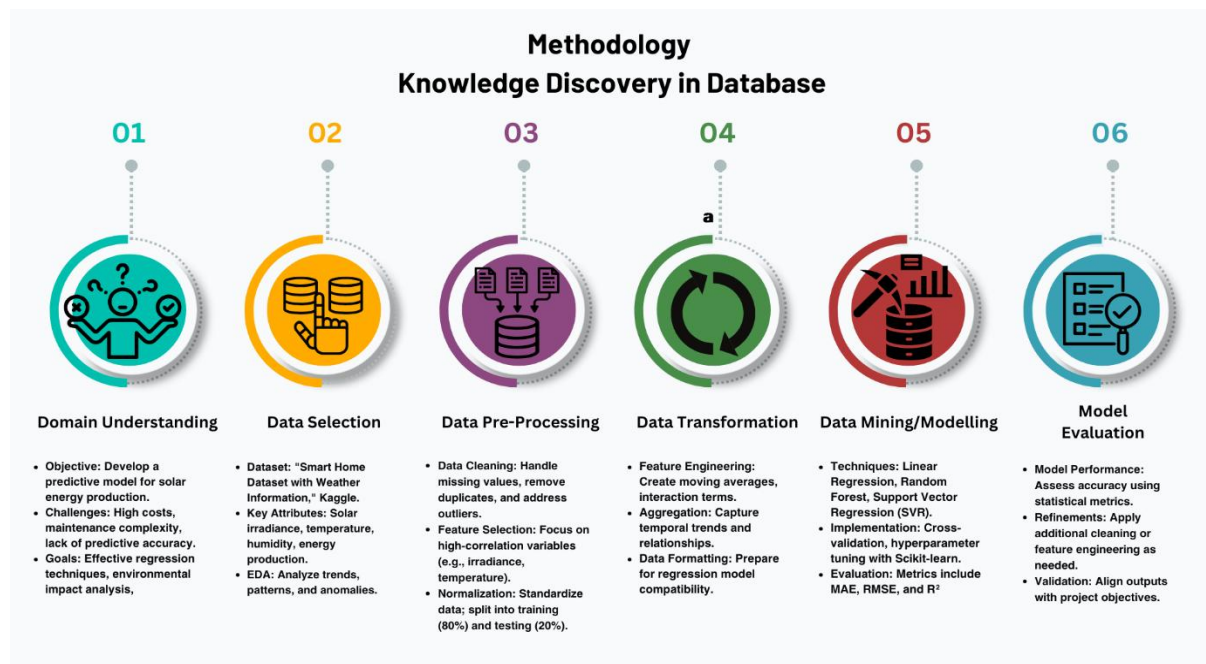


Figure 3.1 Diagram for Project Methodology

3.2.1. Domain Understanding

The primary objective is to develop an accurate predictive model for solar energy production. This model aims to encourage the adoption of solar energy by providing reliable forecasts that optimize energy usage. An evaluation of current solar energy management systems highlights challenges such as high costs, maintenance complexity, and lack of predictive accuracy. This assessment underscores the importance of an efficient forecasting solution. The specific goals are identifying the most effective regression technique for predicting solar energy production and analyzing the influence of environmental factors such as temperature, sunlight, and weather conditions. A comprehensive project plan is created, including timelines, milestones, tools and expected deliverables.

3.2.2 Data Selection

This phase involves collecting and exploring the dataset to gain insights and assess its quality. The dataset, Smart Home Dataset with Weather Information (2019) by Taranveer Singh Anttal, is obtained from Kaggle. It includes solar energy production data and associated weather conditions. Key attributes include temperature, solar irradiance, humidity, wind speed, and energy production values. Summary statistics and data distributions are analyzed to understand the dataset's structure. Exploratory data analysis (EDA) is conducted using visualization tools like Python's Matplotlib and Seaborn. This identifies trends, patterns, and potential anomalies. The dataset is checked for missing values, inconsistencies, and outliers. Data quality is essential to ensure accurate modeling results.

3.2.3 Data Pre-Processing

This phase focuses on preparing the dataset for modeling by cleaning, transforming, and integrating data. Relevant features such as solar irradiance, temperature, and historical energy production are selected based on their correlation with the target variable. Missing values are handled using imputation techniques. Duplicate records are removed to maintain data integrity. Additional features, such as moving averages or interaction terms, may be engineered to improve model performance. Data from different sources such as Weather API is merged into a single, cohesive data set. The data is standardized or normalized to ensure compatibility with regression algorithms. It is then split into training and testing datasets. The date and time involved in the dataset will also be ensured to be in minute format to ensure smooth integration with the chosen Weather API.

3.2.4 Data Transformation

Data transformation is a crucial step in ensuring the dataset is properly structured for predictive modelling. Since the dataset originates from Kaggle, it includes raw solar energy production values along with relevant weather attributes. Several key transformations are applied to enhance model performance. Feature engineering is employed to create new variables, such as moving averages and interaction terms, which capture complex relationships between factors like temperature and humidity. Additionally, temporal features such as time of day, seasonal variations, and daily solar cycle changes are extracted to improve the model's ability to identify trends.

To ensure consistency across all numerical inputs, data standardization and normalization techniques are applied. This step is critical for models such as Support Vector Regression (SVR) and Linear Regression, which are sensitive to scale variations. Missing data is handled using statistical imputation techniques, ensuring that incomplete observations do not introduce biases into the model. Furthermore, real-time weather data is incorporated using an external weather API, enabling the system to compare past predictions with current weather conditions. This integration ensures that Power BI can dynamically display how solar energy production aligns with evolving environmental factors, making the model highly adaptable for real-world applications. [Figure 3.2 shows a flowchart on the full data transformation that will be conducted.](#)

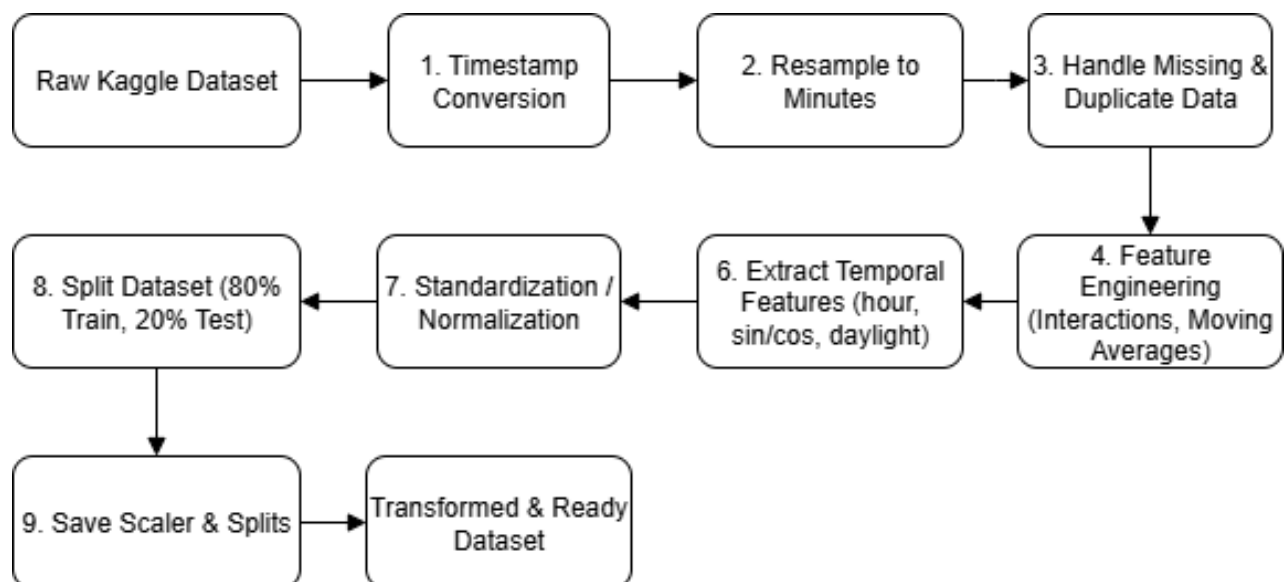


Figure 3.2 Data Transformation Flowchart

3.2.4 Data Mining/Modelling

Three regression techniques are considered, including Linear Regression, Random Forests and Support Vector Regression (SVR). The dataset is split into training (80%) and testing (20%) sets. Cross-validation is used to evaluate model robustness. Each regression model is implemented using Python libraries such as Scikit-learn. Hyperparameter tuning is performed to optimize model performance. Models are evaluated using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared. The best-performing model is selected for deployment.

The hardware and software requirements to ensure an effective data mining or modelling technique are listed in Table 3.1 below.

Table 3.1: Hardware and Software Requirement

Category	Requirement
Hardware	
Processor	Intel Core i5/i7 or AMD Ryzen 5/7
RAM	Minimum 8GB, Recommended 16GB or higher for faster model training
Storage	Minimum 256GB SSD (Recommended 512GB SSD or higher for handling large datasets)
Software	
Operating System	Windows 10/11
Integrated Development Environment (IDE)	Visual Studio Code
Programming Language	Python 3.7 or higher
Python Libraries	Pandas, NumPy, Scikit-learn
Visualization Tools	Power BI (for data visualization and demonstration)

3.2.5 Model Evaluation

The evaluation phase ensures that the predictive model meets the business objectives by assessing its performance using statistical metrics and validating its alignment with the project goals. The performance of the regression models is measured using the following metrics:

3.2.5.1 Mean Absolute Error (MAE)

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

Equation 3.1: Mean Absolute Error

This metric calculates the average magnitude of errors between predicted and actual values, providing an intuitive measure of prediction accuracy. A lower MAE indicates a better-performing model.

3.2.5.2 Root Mean Squared Error (RMSE)

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

Equation 3.2: Root Mean Squared Error

RMSE measures the standard deviation of the residuals (prediction errors). It penalizes larger errors more heavily than MAE, making it sensitive to outliers. Lower RMSE values indicate better model performance.

3.2.5.3 R-squared (R^2)

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Equation 3.3: R-Squared

R^2 , also known as the coefficient of determination, indicates the proportion of variance in the dependent variable that is predictable from the independent variables. R^2 values closer to 1.0 indicate a stronger predictive power of the model.

The evaluation results are thoroughly reviewed to ensure the model meets the project's objectives. Insights from the evaluation guide decisions on model refinement or further tuning. If the metrics indicate insufficient performance, additional feature engineering, data cleaning, or hyperparameter tuning may be performed. If the model performs well, it proceeds to deployment and integration with visualization and automation tools for demonstration. This rigorous evaluation ensures the chosen model is reliable and accurate in predicting solar energy production.

After evaluating multiple models, the best-performing regression model is deployed in Power BI for real-time visualization. The integration process involves embedding Python scripts directly into Power BI to allow dynamic predictions based on live data inputs. Additionally, a real-time weather API is connected to Power BI, enabling users to view live weather conditions alongside solar energy predictions. This setup is further enhanced with an interactive dashboard, which provides an intuitive way to analyse how different weather variables impact solar power generation. The combined use of Python for predictive modelling and Power BI for visualization ensures that this system is both analytically robust and user-friendly, making it an effective tool for energy monitoring and decision-making. Figure 3.3 below shows a simple mock-up of the expected outcome of the dashboard.

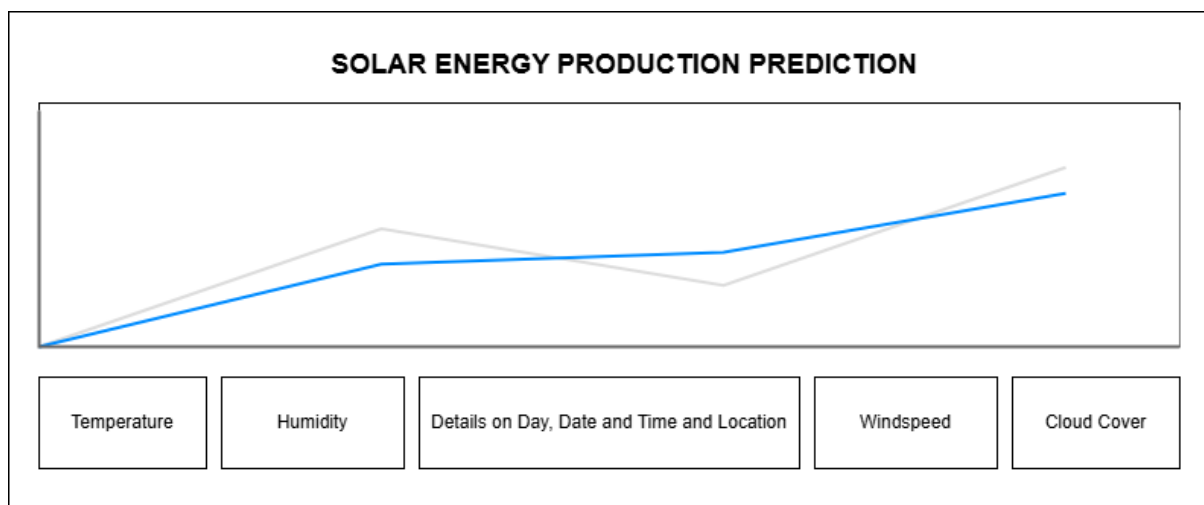


Figure 3.3 Mock-up of Power BI dashboard

3.3 Chapter Summary

This chapter described the Knowledge Discovery in Databases (KDD) framework-based technique that was applied in the project. The method's organized approach to data processing and the creation of trustworthy predictive models led to its selection.

With an initial focus on developing a precise solar energy forecast model to tackle issues such as high costs and poor predictive accuracy, the project started with a domain understanding. The quality and relevance of the data were ensured by selecting pertinent data from a trustworthy source and analysing important characteristics like temperature and solar irradiation.

To deal with missing values, eliminate duplicates, and get the data ready for modelling, the data will be pre-processed. Following the standardization and division of the dataset for training and testing, features will be chosen according to their influence on solar energy prediction.

Three regression techniques such as Linear Regression, Random Forest, and Support Vector Regression will be used throughout the modelling phase, and their effectiveness will be assessed using statistical measures including R-squared, MAE, and RMSE. Cross-validation and tuning will make sure the model performs at its best. The top-performing model is prepared for deployment.

Chapter 4

Implementation

4.1 Introduction

In this chapter, the implementation of the Knowledge Discovery in Database (KDD) process is described. The chapter covers the various steps involved in the KDD process, including data selection, pre-processing, transformation, and modelling using machine learning algorithms such as Linear Regression, Random Forest and Support Vector Regression (SVR). The main objective is to evaluate which model will be the best to be used to predict solar energy production.

4.2 Implementation

4.2.1 Data Selection

The foundation of this project is the Smart Home Dataset with Weather Information (2019), sourced from Kaggle. This dataset contains minute-level recordings of solar energy production, paired with corresponding weather parameters such as temperature, humidity, wind speed, and cloud cover from an unnamed country in 2016. The dataset is chosen for its large number of rows at 503910 rows and various features plus parameters. To enable real-time and out-of-sample testing, additional weather data was retrieved from the Open-Meteo API, which provides both historical and forecasted meteorological variables for Kuching, Malaysia. This dual data source approach ensures the robustness of the model by exposing it to both controlled historical conditions and dynamic real-world scenarios.

4.2.2 Data Preprocessing

Initially, the raw dataset comprised 32 features including items used by the solar energy and weather parameters involved. Through exploratory data analysis, only six features were identified as relevant to the task, and the rest were discarded. As the data is from a foreign country, snow-related features which are not applicable in Malaysia were systematically removed using Microsoft Excel's filter features. Figure 4.1 and 4.2 shows the raw data before and after cleaning respectively.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG
1	time	use [kW]	gen [kW]	House ow	Dishwash	Furnace 1	Furnace 2	Home offi	Fridge [kW]	Wine cella	Garage do	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1	Kitchen 1
2	1.5E+09	0.93283	0.00348	0.93283	3.33E-05	0.0207	0.06192	0.44263	0.12415	0.00698	0.01308	0.00042	0.00015	0	0.03135	0.00102	0.00407	0.00152	0.00348	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
3	1.5E+09	0.93433	0.00347	0.93433	0	0.02072	0.06382	0.44407	0.124	0.00698	0.01312	0.00042	0.00015	0	0.0315	0.00102	0.00407	0.00165	0.00347	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
4	1.5E+09	0.93182	0.00347	0.93182	1.67E-05	0.0207	0.06332	0.44607	0.12353	0.00698	0.01308	0.00043	0.00017	1.67E-05	0.03152	0.001	0.00407	0.00165	0.00347	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
5	1.5E+09	1.02205	0.00348	1.02205	1.67E-05	0.1069	0.06852	0.44658	0.12313	0.00698	0.013	0.00043	0.00022	0	0.0315	0.00102	0.00407	0.00162	0.00348	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
6	1.5E+09	1.1394	0.00347	1.1394	0.00013	0.23693	0.06398	0.44653	0.12285	0.00685	0.01278	0.00045	0.00033	0	0.0315	0.00102	0.00407	0.00158	0.00347	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
7	1.5E+09	1.39187	0.00343	1.39187	0.00028	0.50325	0.06367	0.44703	0.1223	0.00672	0.01245	0.00048	0.00057	0	0.03145	0.00102	0.00407	0.00158	0.00343	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
8	1.5E+09	1.36222	0.00345	1.36222	0.00028	0.4994	0.06372	0.44277	0.12295	0.00673	0.01242	0.00052	0.00055	0	0.03155	0.00103	0.00412	0.00153	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
9	1.5E+09	1.4319	0.00342	1.4319	0.00025	0.47787	0.17863	0.44428	0.1218	0.00678	0.01255	0.00048	0.00045	0	0.03173	0.00103	0.0042	0.00155	0.00342	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
10	1.5E+09	1.6273	0.00342	1.6273	0.00018	0.44765	0.3657	0.44147	0.12182	0.00695	0.01272	0.00047	0.0003	1.67E-05	0.03177	0.00102	0.0042	0.00157	0.00342	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
11	1.5E+09	1.73538	0.00342	1.73538	1.67E-05	0.17155	0.3625	0.43873	0.12183	0.00723	0.01335	0.00037	5.00E-05	0	0.03167	0.00102	0.0042	0.00162	0.00342	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
12	1.5E+09	1.58508	0.00342	1.58508	5.00E-05	0.0221	0.67873	0.4402	0.12145	0.00743	0.01358	0.00035	0.00012	3.33E-05	0.03167	0.001	0.0042	0.00157	0.00342	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
13	1.5E+09	1.51032	0.00343	1.51032	3.33E-05	0.02197	0.62067	0.43995	0.12125	0.00732	0.01383	0.00033	0.0001	0	0.03175	0.001	0.0042	0.00157	0.00343	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
14	1.5E+09	1.45987	0.00345	1.45987	5.00E-05	0.02188	0.57747	0.43995	0.12103	0.00723	0.01352	0.00037	8.33E-05	1.67E-05	0.03178	0.001	0.00422	0.00158	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
15	1.5E+09	0.84058	0.00343	0.84058	0	0.02095	0.1448	0.44478	0.05502	0.00703	0.01318	0.00065	0.00018	1.67E-05	0.03178	0.00102	0.00422	0.00162	0.00343	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
16	1.5E+09	0.7032	0.00343	0.7032	1.67E-05	0.02073	0.06197	0.44383	0.00478	0.00697	0.01312	0.00073	0.00023	0	0.03175	0.00102	0.00423	0.00155	0.00343	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
17	1.5E+09	0.57188	0.00345	0.57188	0	0.02065	0.06365	0.30778	0.00492	0.00705	0.0131	0.00073	0.00015	0	0.03173	0.001	0.00422	0.00158	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
18	1.5E+09	0.48573	0.00345	0.48573	1.67E-05	0.02062	0.06343	0.22045	0.00498	0.00703	0.01312	0.00075	8.33E-05	0	0.03183	0.001	0.00423	0.00162	0.00345	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0
19	1.5E+09	0.52317	0.00343	0.52317	0	0.02063	0.06212	0.26005	0.00495	0.007	0.01308	0.00073	0.0001	1.67E-05	0.03185	0.00102	0.00425	0.00172	0.00343	36.14	clear-nigh	0.62	10	Clear	29.26	1016.91	9.18	cloudCow	282	0	24.4	0

Figure 4.1 Dataset before cleaning

1	time	solar	temperature	humidity	windspeed	cloudcover
2	1/1/2016 5:00	0.7142	35.87	0.61	8.29	0.75
3	1/1/2016 5:01	0.497067	35.87	0.61	8.29	0.75
4	1/1/2016 5:02	0.465133	35.87	0.61	8.29	0.75
5	1/1/2016 5:03	0.512933	35.87	0.61	8.29	0.75
6	1/1/2016 5:04	0.651283	35.87	0.61	8.29	0.75
7	1/1/2016 5:05	0.71045	35.87	0.61	8.29	0.75
8	1/1/2016 5:06	1.195683	35.87	0.61	8.29	0.75
9	1/1/2016 5:07	1.184767	35.87	0.61	8.29	0.75
10	1/1/2016 5:08	1.157433	35.87	0.61	8.29	0.75
11	1/1/2016 5:09	0.815617	35.87	0.61	8.29	0.75
12	1/1/2016 5:10	0.61905	35.87	0.61	8.29	0.75
13	1/1/2016 5:11	0.754167	35.87	0.61	8.29	0.75
14	1/1/2016 5:12	1.042117	35.87	0.61	8.29	0.75
15	1/1/2016 5:13	1.324533	35.87	0.61	8.29	0.75
16	1/1/2016 5:14	1.359467	35.87	0.61	8.29	0.75

Figure 4.2 Dataset after cleaning

Although the dataset was expected to provide one-minute interval readings, an error in data collection led to entries recorded at one-second intervals. This inconsistency was addressed by resampling the data to uniform minute intervals, a frequency better aligned with both the granularity of weather APIs and the intended forecasting application.

The data was also cleaned to remove missing, blank, or duplicate entries using pandas' data cleaning functions. The conversion of the time column to Python datetime format enabled the creation of additional features, such as hour of the day and a daylight indicator (set to 1 for 6 AM–6 PM, otherwise 0). Weather-related features were standardized or normalized to facilitate model convergence and fair comparison across variables. Figure 4.3 shows the actions taken.

```

# 1. Remove empty rows (any column is NaN)
before = len(df)
df = df.dropna()
after = len(df)
print(f"✅ Dropped {before - after} empty/missing rows. Remaining: {after}")

# 2. Parse time column
df['time'] = pd.to_datetime(df['time'], errors='coerce')

# 3. Remove night data (keep only 6AM-6PM)
df['hour'] = df['time'].dt.hour
df = df[(df['hour'] >= 6) & (df['hour'] < 18)].copy()

```

Figure 4.3 Further preprocessing conducted.

Temperature readings were converted to Celsius to match the input requirements of the machine learning models. Figure 4.4 shows the script used for conversion.

```

2ndpro.py
C: > Users > User > Desktop > SOLAR_PREDICTION_PROJECT > 2ndpro.py > ...

1  import pandas as pd
2
3  # Step 1: Load new cleaned dataset
4  df = pd.read_csv("time_consolidated_HomeC.csv")
5
6  # Step 2: Define conversion function
7  def fahrenheit_to_celsius(f):
8      |   return (f - 32) * 5.0/9.0
9
10 # Step 3: Convert relevant temperature columns
11 temp_cols = ['temperature']
12
13 for col in temp_cols:
14     |   if col in df.columns:
15     |       df[col] = df[col].apply(fahrenheit_to_celsius)
16
17 # Step 4: Save the updated dataset
18 df.to_csv("final_cleaned_dataset_celsius.csv", index=False)
19
20 # Step 5: Print confirmation
21 print("Fahrenheit to Celsius conversion completed successfully!")
22 print(f"Final dataset saved as 'final_cleaned_dataset_celsius.csv'")
23

```

Figure 4.4 Temperature readings converted

A correlation matrix was created to check the correlation between the new features. Figure 4.5 shows the correlation matrix.

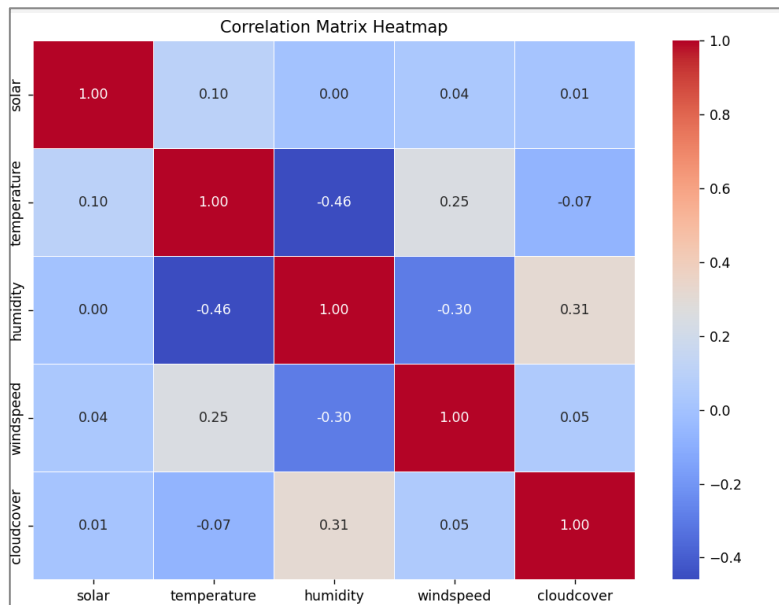


Figure 4.5 Correlation matrix.

The figure shows how strongly different weather factors (like temperature, humidity, wind speed, and cloud cover) may be related to solar energy production. Each square shows a number called the correlation, which ranges from -1 to 1. A value closer to 1 means a strong positive relationship, closer to -1 means a strong negative relationship, and around 0 means little to no relationship. In this diagram, all the values related to solar energy are close to 0, meaning that none of these weather factors have a strong direct impact on solar energy production in this dataset. Because of this, time-based features will also be considered to better capture patterns in solar energy production. Additionally, since the relationships between variables appear weak and likely non-linear, there is a high possibility that a non-linear model such as Random Forest or Support Vector Regression will perform better than a simple linear model.

4.2.3 Data Transformation

After data preprocessing, the dataset consisted of 94,878 of minute records of solar energy production and corresponding weather variables. The cleaned dataset was then split into two subsets:

- 80% for training (75,902 rows)
- 20% for testing (18,976 rows)

This split was performed using Scikit-learn's `train_test_split` function with a fixed random seed to ensure reproducibility.

Before model training, all numeric features were standardized using StandardScaler, which transformed the data to have a mean of 0 and standard deviation of 1. This preprocessing step is crucial for improving the performance of machine learning models especially those like Support Vector Regression (SVR) or Linear Regression, which are sensitive to feature scale. Both the scaled datasets and the scaler object were saved for consistency during real-time API-based prediction.

Initially, the model was trained using a combination of weather variables (temperature, humidity, wind speed, cloud cover) and time-based features (hour, is_daylight, hour_sin, hour_cos). However, during evaluation, it was observed that the model despite having a low Mean Absolute Error (MAE), produced a very high Mean Absolute Percentage Error (MAPE), around 50%.

This level of MAPE is problematic because it means that, on average, the model's predictions were 50% off from the actual values in relative terms, especially during hours of low energy production. This significantly limits the model's usefulness for decision-making and undermines its credibility when deployed in real-time applications.

Upon further analysis, it became clear that the time-based features were introducing noise rather than improving accuracy. One possible reason was that the dataset lacked sufficient temporal diversity, making it difficult for the model to learn generalizable time-related behaviours. Additionally, features like hour and hour_sin/cos although theoretically useful for capturing daily solar cycles had weak correlations with the target variable, and their inclusion led to overfitting or distortion in the model's learning process.

To address this, the final model was trained only using weather-related features. This decision was based on:

- Empirical results showing a lower MAPE when time features were excluded
- The physical relevance of weather variables, which are direct causal factors of solar energy production
- The improved stability and generalizability of the model when relying on environmental inputs alone

In the deployment pipeline, time-based features were reintroduced only during the prediction phase. This allowed the model's output to be interpreted and calibrated more accurately

especially to reflect real-world bell-curve patterns of solar generation throughout the day without compromising model performance during training.

This hybrid approach ensures that the model:

- Remains focused on the most relevant predictors during learning
- Still benefits from temporal context during real-time usage
- Avoids inflating errors due to inconsistent or weak time signals

. Figure 4.6 shows the script used for splitting into training and testing dataset.

```
# Load cleaned dataset
df = pd.read_csv("final_cleaned_dataset_celsius.csv")

# Specify features and target
columns_to_keep = ['temperature', 'humidity', 'windspeed', 'cloudcover']
X = df[[col for col in columns_to_keep if col in df.columns]]
y = df['solar']

# Feature scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split into training (80%) and testing (20%) sets
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42
)

# Convert arrays back to DataFrames for saving
X_train = pd.DataFrame(X_train, columns=X.columns)
X_test = pd.DataFrame(X_test, columns=X.columns)
y_train = pd.DataFrame(y_train, columns=['solar'])
y_test = pd.DataFrame(y_test, columns=['solar'])

# Save files
X_train.to_csv("X_train.csv", index=False)
X_test.to_csv("X_test.csv", index=False)
y_train.to_csv("y_train.csv", index=False)
y_test.to_csv("y_test.csv", index=False)
joblib.dump(scaler, "scaler.pkl")
```

Figure 4.6 Splitting Python Script

4.2.4 Data Mining/Modelling

For the predictive modelling phase, three regression techniques were implemented using Python's Scikit-learn library: Linear Regression, Random Forest Regressor, and Support Vector

Regression (SVR). Linear Regression and Random Forest models were trained on the full dataset using an 80:20 train-test split.

Due to computational limitations, SVR was initially trained on only a 20% subset. This limitation arose because SVR is computationally intensive and did not scale efficiently to the full dataset, even with GPU acceleration using Google Colab.

To streamline training, a limited set of hyperparameters was manually configured. For example, the Random Forest Regressor used a fixed number of trees (`n_estimators=20`). SVR was run with default settings and no formal hyperparameter optimization was performed.

Formal hyperparameter tuning methods like `GridSearchCV` were not used primarily due to hardware and runtime constraints. `RandomizedSearchCV` was used to balance performance improvement with feasible computation time for Random Forest and SVR. Figure 4.7 shows the script taking the train and test datasets and training the model.

```
# Step 1: Load the train/test data
X_train = pd.read_csv("X_train.csv")
X_test = pd.read_csv("X_test.csv")
y_train = pd.read_csv("y_train.csv")
y_test = pd.read_csv("y_test.csv")

# Flatten y values if needed
y_train = y_train.values.flatten()
y_test = y_test.values.flatten()

# Step 2: Train models

# Linear Regression
lr = LinearRegression()
start = time.time()
lr.fit(X_train, y_train)
lr_train_time = time.time() - start

# Random Forest (faster settings)
rf = RandomForestRegressor(n_estimators=20, random_state=42, n_jobs=-1)
start = time.time()
rf.fit(X_train, y_train)
rf_train_time = time.time() - start

# Support Vector Regression (only on 20% subset for speed)
X_train_svr, _, y_train_svr, _ = train_test_split(X_train, y_train, train_size=0.2, random_state=42)

svr = SVR(kernel='rbf')
start = time.time()
svr.fit(X_train_svr, y_train_svr)
svr_train_time = time.time() - start
```

Figure 4.7 Training the models

4.2.5 Model Evaluation

Upon training, all three models were evaluated to identify the best approach for solar energy prediction. The models were assessed based on three primary metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2). These metrics provide a comprehensive view of how closely the predicted values aligned with actual solar energy output. Table 4.1 shows the training results.

Table 4.1 Training Results

Model	MAE	RMSE	R^2
Linear Regression	0.1112	0.1346	0.0235
Support Vector	0.0878	0.1036	0.4214
Random Forest	0.0184	0.0408	0.9102

The Linear Regression model exhibited the poorest performance among the three, with an MAE of 0.1112, RMSE of 0.1346, and a very low R^2 score of 0.0235. This indicates that the model could explain only around 2.35% of the variance in the target variable. The high error values suggest that Linear Regression was unable to capture the underlying non-linear relationships between the input weather features and solar energy production. This is expected, as Linear Regression assumes a linear correlation between features and the output, which does not align well with the complex and dynamic nature of solar energy generation influenced by varying weather patterns.

The Support Vector Regression (SVR) model performed significantly better, achieving an MAE of 0.0878, RMSE of 0.1036, and an R^2 of 0.4214. This improvement is largely due to SVR's ability to model non-linear relationships through kernel functions. However, despite its better accuracy, SVR had limitations in terms of scalability and computational efficiency. Due to hardware constraints, it was only feasible to train the SVR model on a subset (20%) of the dataset. This limited training capacity likely impacted the model's ability to generalize well across the entire data range, preventing it from reaching its full potential.

The Random Forest Regressor emerged as the best-performing model, with an MAE of 0.0184, RMSE of 0.0408, and an impressive R^2 score of 0.9102. This means the model was able to explain over 91% of the variance in solar energy output, demonstrating a strong fit to the data. Random Forest excels in capturing complex, non-linear relationships and is relatively robust to noise and irrelevant features. It works by constructing an ensemble of decision trees and

aggregating their predictions, which reduces the risk of overfitting while maintaining flexibility. Its performance in this project suggests that the solar energy generation patterns were best captured through an ensemble approach that could handle the interactions and thresholds present in meteorological data.

In summary, the results show a clear progression in predictive capability from Linear Regression to SVR, and ultimately to Random Forest. The findings underscore the importance of using non-linear, ensemble-based models for solar energy forecasting, particularly when dealing with variables like cloud cover and temperature, which influence solar output in non-linear and time-sensitive ways. The Random Forest model's balance between accuracy and interpretability made it the most suitable choice for deployment in the real-time prediction system.

Model results were further validated through visual inspection of predicted vs. actual production curves, especially focusing on the model's ability to capture the expected daily bell curve shape of solar generation. Power BI is used to conduct the visual inspection. The next chapter further investigates the results obtained.

4.3 Summary

In summary, this chapter has detailed each stage of the implementation process: from the careful selection and cleaning of the dataset, through thoughtful feature engineering and transformation, to the development, training, and comparative evaluation of three regression models. The Random Forest Regressor emerged as the most suitable algorithm due to its high accuracy and strong alignment with physical solar generation patterns. The workflow culminated in the deployment of this model for real-time prediction using live weather data, supported by calibration techniques to further enhance interpretability and reliability. This robust and scalable approach provides a solid foundation for practical solar energy forecasting applications.

Chapter 5

Results And Discussion

5.1 Introduction

This chapter presents and discusses the results of the solar energy production prediction models developed in this project. The focus is on comparing the actual solar energy production values calculated using meteorological data and solar physics with the machine learning model's predictions, after a simple calibration step. The results are visualized and discussed in terms of both predictive accuracy and practical implications.

5.2 Visualization of Actual and Predicted Solar Energy

As mentioned, OpenMeteo API will be used to get weather data that can be used to help with predicting the solar energy production. The API will be used to forecast prediction every week. Figure 5.1 shows the URL used to obtain the data from OpenMeteo.

```
# === 1. Date Setup: Start today ===
today = datetime.date.today()
start_date = today.strftime('%Y-%m-%d')
end_date = (today + datetime.timedelta(days=6)).strftime('%Y-%m-%d') # 7-day range

url = (
    f"https://api.open-meteo.com/v1/forecast"
    f"?latitude=1.5533"
    f"&longitude=110.3592"
    f"&start_date={start_date}"
    f"&end_date={end_date}"
    f"&hourly=shortwave_radiation,direct_radiation,temperature_2m,relativehumidity_2m,wind_speed_10m,cloudcover"
    f"&timezone=Asia%2FKuching"
)
```

Figure 5.1 Script to fetch data from OpenMeteo API

Unfortunately, API's that provide real solar energy data from panel was unable to be found in free tier API. Hence, to enable direct comparison between predicted and estimated actual energy, hourly solar production was expressed in kilowatt-hours (kWh) and calculated as follows in Equation 5.1,

$$actual_kWh = \frac{\text{shortwave radiation} \left(\frac{W}{m^2} \right) \times \text{panel area}(m^2) \times \text{efficiency}}{1000}$$

Equation 5.1 Equation used to calculate actual_kWh

where panel area and efficiency were set at the lowest for best fit.

Here, shortwave radiation is used because it accounts for the total downward solar radiation received at the surface, including both direct and diffuse components. It is a standard metric in meteorological datasets for estimating solar irradiance at the surface level and is often used in photovoltaic modelling (Vennila et al., 2022).

The panel area and efficiency were set to conservative minimum values to ensure general applicability and avoid overestimation.

The primary result is visualized in Power BI, which plots the average actual solar energy production (kWh) and the average calibrated predicted solar energy for each hour over a week-long period.

The blue line represents the “ground truth” or actual solar energy values, calculated from weather data using established solar irradiance formulas and known panel specifications. The white line shows the calibrated machine learning model prediction, adjusted with a scaling multiplier to correct for systematic under- or overestimation.

As observed, both curves display the expected bell-shaped pattern of solar energy generation, which corresponds to the natural solar cycle—rising in the morning, peaking around midday, and tapering off by evening. This behaviour validates the model's ability to capture the core structure of daily solar production.

However, the predicted curve typically has lower amplitude, and less variance compared to the actual values. This indicates that while the regression model can follow overall trends, it is limited in capturing short-term fluctuations and extreme peaks or dips, which are often influenced by localized or sudden weather changes such as cloud cover variability.

Importantly, the visualization is intentionally limited to the 8 AM to 5 PM time window. Outside of this range, both actual and predicted solar energy values approach or drop to zero due to the absence of sunlight. Including these near-zero values in evaluation metrics such as Mean Absolute Percentage Error (MAPE) would artificially inflate error percentages, especially because percentage-based errors become unstable or exaggerated when the actual value is close to zero.

To ensure a more meaningful and accurate representation, only the daytime hours where solar generation occurs are visualized and evaluated. This filtering approach helps focus the analysis

on the period where model performance truly matters during active solar production hours thereby improving the clarity and relevance of both visual and quantitative assessments. Figure 5.2 and 5.3 shows the charts difference when limited to only 8am to 5pm and when charted 24 hours respectively.

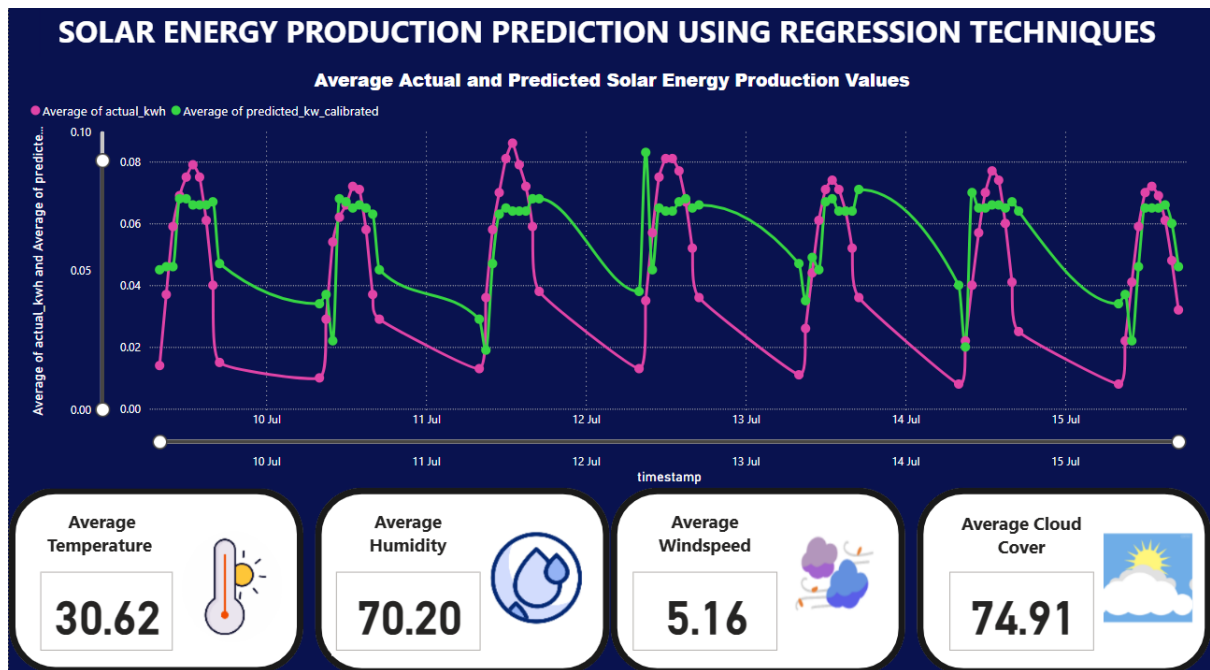


Figure 5.2: Power BI dashboard showing average actual and predicted solar energy production, with environmental feature cards.

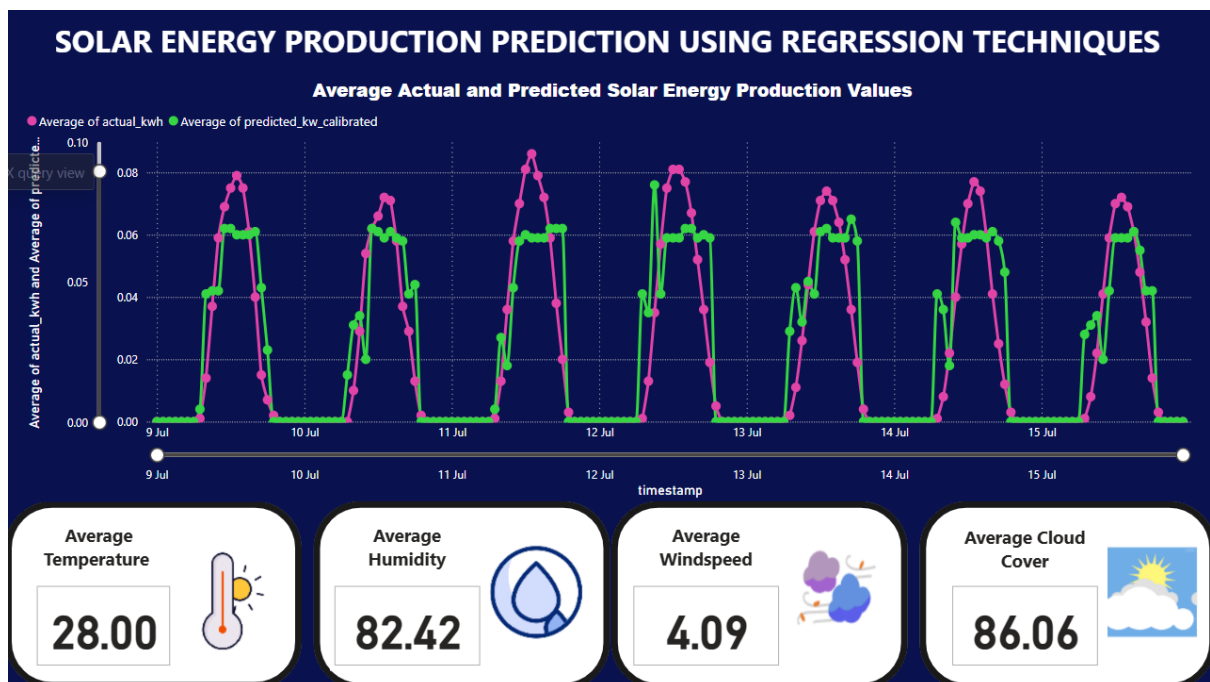


Figure 5.3: Power BI dashboard showing prediction for 24 hours resulting a higher MAPE

5.3 Final Correlation Analysis

Figure 5.4 presents the correlation matrix heatmap, which highlights the relationships between input features and the actual and predicted solar energy outputs.

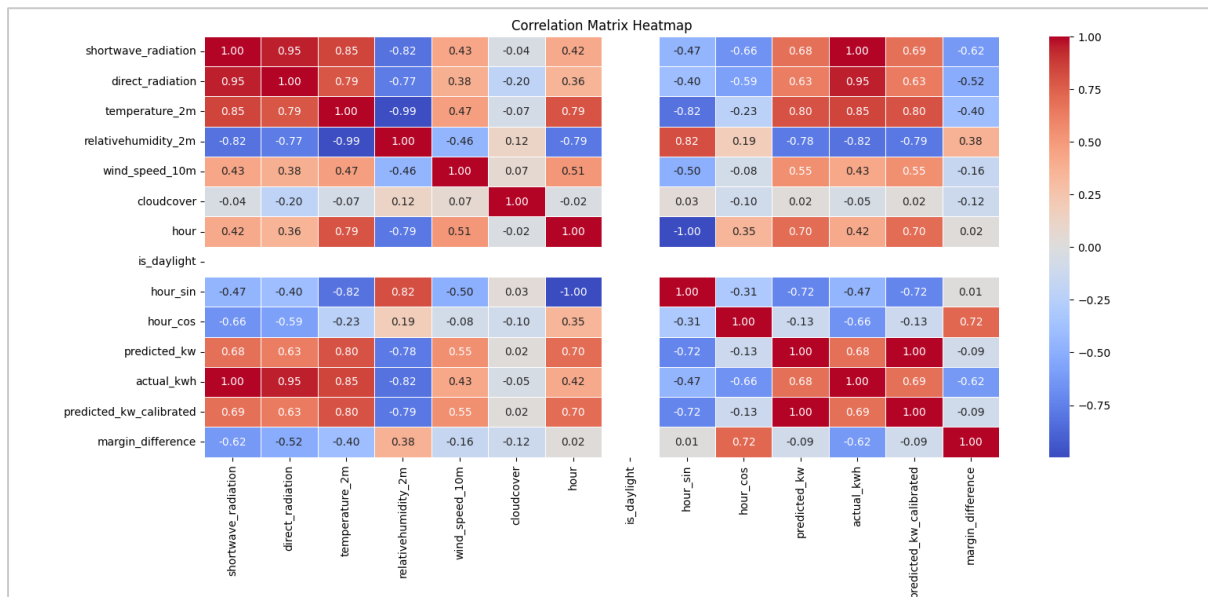


Figure 5.4: New Correlation Matrix

As expected, shortwave radiation and direct radiation show the strongest positive correlations with actual solar energy generation (1.00 and 0.95 respectively), confirming that irradiance is the most influential factor. Temperature also correlates positively (0.85), while relative humidity shows a strong negative correlation (-0.82), as higher humidity often corresponds to cloudier skies and reduced solar radiation. Wind speed has a moderate positive relationship, likely reflecting its indirect role in affecting weather conditions.

Time-based features such as hour, hour_sin, and hour_cos show moderate correlations with solar output and predictions, reflecting the natural daily solar cycle. However, their lower correlation compared to weather features supports the decision to exclude them during model training and use them only for post-prediction shaping. The calibrated model predictions (predicted_kw_calibrated) also show strong alignment with actual values (0.69), and the low correlation of margin_difference with other features indicates that prediction errors are not biased toward any specific input. Overall, the matrix validates the focus on weather-based features for training and confirms the soundness of the modelling approach.

5.4 Model Performance and Calibration

The calibration approach applied here involves scaling the raw model predictions to better align with the range of the actual values. This is a common practice when the model captures the general trend but suffers from underestimation or overestimation due to training on limited or noisy data. No alpha blending with actual values was used. All evaluation metrics (MAE, RMSE, MAPE, R^2) were computed on the calibrated model output only. The environmental feature cards at the bottom of the dashboard such as temperature, humidity, windspeed and cloud cover provide additional context and reinforce the influence of weather on solar energy production (Balal et al., 2023).

5.5 Interpretation and Limitations

While the model captures the daily pattern of solar production, it does not perfectly fit the actual production curve. Key factors influencing this outcome include:

- **Dataset limitations:** The available data contained missing values and errors, which led to a small and imperfect training set, particularly after aggregating to minute data and removing night-time hours.
- **Model simplicity:** The regression techniques used including Random Forest can only learn patterns present in the training data. With limited samples, especially of rare or extreme events, the model may fail to capture the full diversity of real-world conditions.
- **Calibration necessity:** The need for calibration reflects these limitations and is a reminder that model outputs are only as good as the data used to train them.

Despite these challenges, the results demonstrate that a data-driven approach, when calibrated, can provide reasonable short-term predictions of solar energy production and useful insights for operational planning or educational purposes (Nie et al., 2022).

5.5 Automation and Real-Time Dashboard Refresh

To ensure the solar energy prediction model remains up to date with the latest weather data, a fully automated refresh pipeline was implemented using the following tools:

- **Python Script:** A script was developed to call the OpenMeteo API weekly and save the forecasted weather data to a .csv file. The script automatically sets the start date to the current date and forecasts a seven-day range of weather data. This data is processed and used to generate calibrated solar energy predictions. The predictions are then saved as `solar_prediction.csv`.

- **OneDrive Integration:** The Python script saves the .csv file directly into a specific OneDrive folder. This allows Power BI to continuously reference the latest data from a cloud-synced location.
- **Power BI Service:** A dashboard was created in Power BI Desktop and published to the Power BI Service. The dataset was configured to connect to the OneDrive file, enabling automatic updates when the source .csv is refreshed.
- **Power Automate:** A scheduled cloud flow was created using Power Automate to trigger a dataset refresh in Power BI once a week. This ensures the Power BI dashboard automatically reflects the latest prediction values from the most recent weather API data (Tirupati et al., 2023). Figure 5.5 shows the flow of the automation in Power Automate.

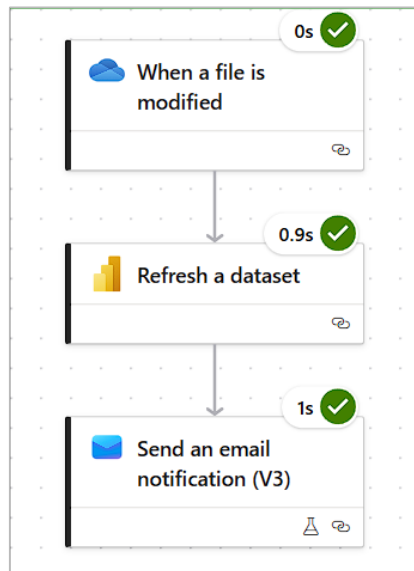


Figure 5.5 Automation Flow

This fully automated pipeline minimizes manual effort, maintains up-to-date solar prediction results, and ensures the dashboard is always ready for monitoring and decision-making.

5.6 Evaluation Metrics and Calibration Explanation

After the raw predictions were generated, a calibration step was introduced to scale the predictions closer to the actual solar energy values derived from weather-based solar radiation formulas. This is done by finding a scaling multiplier that minimizes the Mean Absolute Error (MAE) between the predicted and calculated values.

The following metrics were used to evaluate the calibrated model:

- **Mean Absolute Error (MAE):** Measures the average absolute difference between predicted and actual values. A lower MAE indicates better accuracy.
- **Root Mean Squared Error (RMSE):** Like MAE but penalizes larger errors more heavily. It is useful when high variance in predictions needs to be detected.
- **Mean Absolute Percentage Error (MAPE):** Expresses prediction error as a percentage of actual values. MAPE is particularly useful for interpretability, showing how much the prediction deviates from reality in percentage terms.

A MAPE of 12% means that, on average, the prediction is off by 12% compared to actual values (AIP Advances, 2024).

In this project, the calibration multiplier was determined by testing a range of values from 0.2 to 3.0 and selecting the one that produced the lowest MAE. The calibrated predictions were found to match the actual calculated values significantly better than the raw model output, especially during peak solar hours.

5.7 Chapter Summary

This chapter presented the results of the solar energy prediction model and explained its strengths and limitations. It was shown that the regression model was able to capture the diurnal pattern of solar energy production reasonably well. A calibration step using MAE minimization significantly improved accuracy. The dashboard, built with Power BI, effectively visualizes both predicted and actual solar output alongside relevant weather variables. An automated system using Python, OneDrive, Power BI, and Power Automate was implemented to ensure real-time refresh of the prediction pipeline with minimal human intervention. Evaluation metrics such as MAE, RMSE, and MAPE were used to assess model performance, demonstrating the practical value of this approach in forecasting energy trends.

Overall, the project successfully demonstrated a working prototype of an intelligent solar energy forecasting system that can be scaled and improved further with richer datasets and more advanced modelling techniques.

Chapter 6

Conclusion And Future Work

6.1 Introduction

This project implemented a data-driven framework for predicting solar energy production using weather data and regression modeling techniques. The approach involved: collecting and cleaning meteorological and solar generation data; engineering time and environmental features for machine learning; training and calibrating regression models to predict solar output; and visualizing results and performance in an interactive dashboard. While the final model required calibration to accurately follow the observed production pattern, it was able to capture the key daily trends in solar generation. This validates the potential of machine learning for renewable energy forecasting, even when data quality and quantity are limited (Vennila et al., 2022; Solano et al., 2022).

6.2 Contribution

The following are some ways that this project benefits data science practitioners as well as the renewable energy community:

- **Encourages Sustainable Energy Planning:** This model helps homes and energy suppliers optimize energy use and lessen dependency on fossil fuels by offering precise short-term estimates of solar energy generation based on weather data.
- **Makes Forecasting Tools More Accessible:** Users can obtain real-time visual insights and notifications without requiring sophisticated programming skills thanks to connectivity with Power BI and Power Automate.
- **Supports Educational and Research Purposes:** The project demonstrates how machine learning techniques can be applied to environmental datasets, serving as a template for future academic work or expansion by green tech startups.
- **Enables Proactive Energy Management:** The predictive system allows users to make informed decisions about energy storage and consumption, especially in areas where solar reliability is crucial for off-grid applications

6.3 Limitations

Several limitations were encountered and should be acknowledged:

- Limited dataset: The number of usable, high-quality data points after cleaning was small, restricting model learning capacity and robustness (Solano et al., 2022).
- Missing/extreme values: Gaps and errors in weather data, especially after aggregation, reduced the fidelity of both training and evaluation.
- Lack of advanced features: The models did not include certain potentially useful predictors, such as historical solar output, shading factors, or real-time irradiance sensor data.
- Calibration dependency: Reliance on post-processing calibration indicates the model was not fully generalized based only on the available features (Aouidad & Bouhelal, 2024).

6.3 Recommendations for Future Work

- Increase dataset size: Incorporate more months or years of data and/or additional data sources to strengthen model training and evaluation.
- Enhance Data Quality: Use advanced imputation techniques or APIs with better resolution and stability for more consistent inputs.
- Feature Enrichment: Include time-lag features, moving averages, weather type classifications, and IoT-based sensor data to enrich inputs.
- Model Optimization: Apply automated hyperparameter tuning and experiment with advanced models (e.g., XGBoost, LSTM) that generalize better.
- Automate and deploy: Develop a fully automated data pipeline that regularly updates the model and dashboard with new data for ongoing, real-time forecasting using tools like Power BI and Power Automate (Quevauvilliers, 2022; Tirupati et al., 2023).

6.4 Summary

This study has demonstrated the core principles and challenges of machine learning-based solar energy prediction. While not without limitations, the project provides a solid foundation for more advanced and scalable forecasting systems in the future. The integration of real-time automation and ongoing model evaluation can significantly enhance the reliability and utility of renewable energy predictions in smart grid systems (Vennila et al., 2022).

References

- AIP Advances. (2024). A systematic research survey on solar power forecasting using hybrid statistical and ML techniques. *AIP Advances*, 12(8). <https://doi.org/10.1063/5.0227616>
- Aouidad, H. I., & Bouhelal, A. (2024). Machine learning-based short-term solar power forecasting: A comparison between regression and classification approaches using extensive Australian dataset. *Sustainable Energy Research*, 11, 28. <https://doi.org/10.1186/s40807-024-00115-1>
- Balal, A., Jafarabadi, Y. P., Demir, A., et al. (2023). Forecasting solar power generation utilizing machine learning models in Lubbock. *Emerging Science Journal*, 7(4), 1052–1062. <https://doi.org/10.28991/ESJ-2023-07-04-02>
- Erten, M. Y., & Aydilek, H. (2022). Solar power prediction using regression models. *Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi*, 14(3), 333–342. <https://doi.org/10.29137/umagd.1100957>
- Jebli, I., Belouadha, F. Z., Kabbaj, M. I., & Tilioua, A. (2021). Prediction of solar energy guided by Pearson correlation using machine learning. *Energy*, 224, 120109. <https://doi.org/10.1016/j.energy.2021.120109>
- Komar, M., Lipianina-Honcharenko, K., Domanskyi, V., & Melnyk, N. (2024). Re based method for real-time solar power plant efficiency forecasting. *Proceedings of the 6th International Workshop on Modern Machine Learning Technologies*, 1–6. Retrieved from <https://ceur-ws.org>
- Lee, J., Wang, W., Harrou, F., & Sun, Y. (2020). Reliable solar irradiance prediction using ensemble learning-based models: A comparative study. *Energy Conversion and Management*, 208, 112582. <https://doi.org/10.1016/j.enconman.2020.112582>
- Ledmaoui, Y., El Maghraoui, A., El Aroussi, M., Saadane, R., Chebak, A., & Chehri, A. (2023). Forecasting solar energy production: A comparative study of machine learning

algorithms. *Energy Reports*, 10, 1004–1012.
<https://doi.org/10.1016/j.egy.2023.07.042>

Munawar, M. I., Alam, I., & Ali, M. S. (2023). An efficient energy prediction model for solar energy power system using artificial intelligence technique. *Proceedings of the 1st International Conference on Recent Academic Studies*, 139–140. Retrieved from <https://as-proceeding.com>

Narvaez, G., Giraldo, L. F., Bressan, M., & Pantoja, A. (2021). Machine learning for site adaptation and solar radiation forecasting. *Renewable Energy*, 167, 333–342.
<https://doi.org/10.1016/j.renene.2020.11.089>

Nie, Y., Paletta, Q., & Scott, A. (2022). Sky-image-based solar forecasting using transfer learning. arXiv. <https://doi.org/10.48550/arXiv.2211.02108>

Quevauvilliers, G. (2022, November 10). Using Power Automate to schedule a refresh of a Power BI dataset at the time you want! FourMoo.
<https://www.fourmoo.com/blog/2022/11/10/using-power-automate-to-schedule-a-refresh-of-a-power-bi-dataset-at-the-time-you-want/>

Sahin, G., Isik, G., & van Sark, W. G. J. H. M. (2023). Predictive modeling of PV solar power plant efficiency considering weather conditions: A comparative analysis of artificial neural networks and multiple linear regression. *Energy Reports*, 10, 2837
<https://doi.org/10.1016/j.egy.2023.09.097>

Solano, E. S., Dehghanian, P., & Affonso, C. M. (2022). Solar radiation forecasting using machine learning and ensemble feature selection. *Energies*, 15(19), 7049.
<https://doi.org/10.3390/en15197049>

Suanpang, P.; Jamjuntr, P. (2024). Machine learning models for solar power generation forecasting: Comparing LGBM and KNN in microgrid applications. *Sustainability*, 16(14), 6087. <https://doi.org/10.3390/su16146087>

Tirupati, K. K., Singh, S. P., et al. (2023). Leveraging Power BI for enhanced data visualization and business intelligence. *Universal Research Reports*, 10(2), 105–120.
<https://doi.org/10.36676/urr.v10.i2.1375>

Vennila, C., Titus, A., & Sri Sudha, T. (2022). Forecasting solar energy production using machine learning. *International Journal of Photoenergy*, 2022, 7797488. <https://doi.org/10.1155/2022/7797488>