Improving Short-Term Forecasting Models for Optimised Photovoltaic Curtailment Strategies

Bsc. Industrial Engineering and Management
Bachelor Integration Project Report

Carlos Rodriguez Ruiz-Canela

Student Number: s5129087

First supervisor: Prof. Dr. M. (Michele) Cucuzzella

Daily supervisors: Dr. S. (Sebastian) Trip and E. (Edoardo) Vacchini

Assessor: Dr. Ir. M. (Mozhdeh) Taheri

Contents

Chapter 1: Abstract

This project addresses the critical need for accurate short-term photovoltaic (PV) forecasting to support FIRN Energy's operational decision-making. Building on a prior ARIMAX-based approach, the work introduces enhanced forecasting pipelines for both minute-ahead and day-ahead horizons. A re-implemented ARIMAX(1,0,1) model was tuned for minute-ahead predictions, capturing rapid fluctuations in PV output using an autoregressive structure and exogenous environmental data. A Random Forest model was trained across data from over 20 FIRN sites for day-ahead forecasting, demonstrating superior generalisation and predictive accuracy compared to site-specific or previous statistical models. All models were developed in Python, enabling seamless integration with FIRN's infrastructure. Results show significant accuracy gains across evaluation metrics (MAE, RMSE, MAPE, R^2), with the day-ahead Random Forest achieving up to 68% MAPE reduction over the original ARIMAX baseline in worst-case sites. The final deliverable includes all forecasting code, input data templates, and API connectivity scripts, offering FIRN a scalable, ready-to-deploy solution.

Chapter 2: Introduction

2.1 Context and Problem Setting

As solar photovoltaics (PV) become increasingly prevalent in power systems, they present significant operational challenges. Solar generation's variable and non-dispatchable nature introduces uncertainty into grid operations and electricity markets. Grid operators and market participants rely on accurate forecasts to ensure system balance, making short-term PV forecasting essential. Short-term forecasting refers to predicting solar energy output over the next few minutes to hours, typically up to one day ahead. Forecasts are used to mitigate imbalances in real time and enable informed bidding, optimisation, and reduce operational penalties [Cormode et al., 2014].

For example, the 28 April 2025 Iberian blackout, triggered by frequency instability during approximately 60% renewable penetration, left Spain and Portugal in darkness for nearly ten hours. The event exposed the limitations of inverter-based sources, like PV, which do not naturally provide rotational inertia. Unlike traditional generators that stabilise frequency with their spinning mass, PV systems rely on inverters that lack this stabilising effect, making the grid more susceptible to sudden imbalances [Bousso, 2025].

FIRN Energy, a Belgian company managing a distributed portfolio of PV systems, is directly affected by these dynamics. Its operations span multiple sites with varying characteristics, making accurate forecasting operationally valuable and technically challenging. Two forecasting horizons are especially relevant for FIRN: the minute-ahead timeframe, which supports internal steering to avoid imbalance costs (financial penalties incurred when actual production deviates from what was scheduled in the energy market), and the day-ahead horizon, which informs market participation and curtailment strategies.

While many studies have focused on one-day-ahead PV forecasting, significantly less attention has been paid to minute-ahead predictions in operational environments, where most studies have focused on specific time horizons such as an hour ahead, a day ahead, and a month ahead [Asiedu et al., 2024]. The ability to forecast production just a few minutes in advance allows companies like FIRN to fine-tune their internal systems, reduce imbalance penalties, and improve overall system responsiveness. This use case presents new engineering challenges: the model must operate at high frequency, respond to abrupt changes, and perform under limited future weather visibility [Pelland et al., 2013].

2.2 Problem Statement

FIRN Energy relies on short-term photovoltaic (PV) forecasting to support both real-time operations and longer-term planning. Until now, its forecasting framework has focused on a day-ahead AutoRegressive Integrated Moving Average with eXogenous (ARIMAX) model developed for a single location [Clua, 2025]. While this model provided a helpful starting point, it was constructed with limited weather variables and lacked flexibility. Its performance beyond the original training site was never validated, which restricted its reliability for wider deployment across FIRN's portfolio of solar systems.

Acknowledging these limitations, FIRN identified two clear areas for improvement:

- The company needed an improved day-ahead forecasting solution that could scale across multiple sites while maintaining accuracy.
- FIRN sought to develop a new minute-ahead model to support real-time imbalance steering; an increasingly vital task as solar penetration increases and system dynamics become more volatile over short intervals.

These two requirements, scalability in day-ahead forecasts and the introduction of a responsive minute-ahead model, constitute the core motivation for this project. The aim is to extend and enhance FIRN's existing forecasting capabilities to improve their robustness, responsiveness, and applicability to their operational context.

Inaccurate short-term forecasts can lead to real-time imbalance penalties, inefficient curtailment strategies, and missed market opportunities. Grid operators rely on stable injections, while FIRN's commercial operations benefit from accurate predictions to optimising and revenue. As FIRN expands, scalable forecasting tools become a key operational requirement.

2.3 Research Objective

This project aims to enhance short-term PV forecasting for FIRN Energy by developing and validating two application-specific models tailored for operational use:

- **Minute-ahead forecasting model:** Adapted to predict PV output on a minute-level basis. It utilises production values and short-lag weather variables to support real-time imbalance management at the grid level.
- **Day-ahead forecasting model:** Trained on historical data from FIRN's sites. It is designed to replace the previous forecasting approach with a more accurate and scalable alternative, capable of generalising diverse locations and operational conditions.

Both models are developed using real production and weather data and are intended for seamless integration into FIRN's operations. Their performance will be evaluated using industry-standard metrics, ensuring that the proposed solutions meet the company's requirements for accuracy, robustness, and practical deployability.

2.4 Research Questions

The main research question for this project is:

> *What forecasting setup provides accurate and scalable short-term photovoltaic (PV) predictions across multiple FIRN Energy sites, utilising production and weather data?*

Sub-questions:

- What modelling approach is the most effective for generating reliable minute-ahead forecasts to facilitate real-time imbalance mitigation?
- What design framework ensures accurate day-ahead forecasts that maintain robustness across various sites?
- Is it more advantageous to create individual forecasting models for each site or to develop a unified model leveraging data from multiple sites?

2.5 Scope and Limitations

This project focuses exclusively on short-term forecasting of PV production for FIRN Energy sites. Two distinct forecasting horizons are addressed: minute-ahead predictions to support real-time grid balancing, and day-ahead forecasts for curtailment planning and market operations. All models are developed using historical PV production data provided by FIRN and historical weather forecasts obtained via Open-Meteo. Forecasts are limited to the resolution of the input data: one-minute resolution for minute-ahead models and hourly resolution for day-ahead forecasts.

Seasonal limitations apply. The models were trained and validated on data collected outside of the summer months, which may impact overall performance under higher irradiance conditions. Higher irradiance levels in summer may result in increased PV production and steeper ramps, possibly putting the models into untested operating regimes and increasing prediction errors if they are not retrained using summer data. Additionally, real-time system integration is beyond the current project scope.

Model performance is assessed using standard evaluation metrics, with testing performed on held-out periods for each use case. The emphasis is on accuracy, maintainability, and operational readiness.

Chapter 3: Literature Review

Short-term PV output forecasting has been approached using statistical and machine learning models. Among the statistical methods, AutoRegressive Integrated Moving Average (ARIMA) and ARIMAX are commonly applied due to their simplicity and effectiveness for short horizons. These models combine autoregressive terms with external inputs such as solar irradiance or cloud cover, enabling them to track temporal dependencies while accounting for external drivers of PV production. As demonstrated by a study, ARIMAX can deliver strong performance for very short-term horizons (5 minutes to 2 hours), particularly when utilising updates to model parameters and lagged PV observations as primary predictors [Bacher et al., 2009]. However, they struggle with rapid weather changes and nonlinear relationships.

ARIMAX models are a practical choice for minute-ahead forecasting due to its rapid computation, interpretability, and the significance of temporal correlations in ultra-short-term horizons. Its effectiveness in controlled conditions and minimal data requirements render it suitable for real-time operational use.

Machine learning models have gained attention for day-ahead forecasting due to their capacity to capture complex, nonlinear relationships between weather variables and PV output. Tree-based models such as Random Forest (RF) and Gradient Boosted Trees (GBT) effectively manage multivariate input data and are less susceptible to overfitting. A study by UPM demonstrated that RF significantly outperformed traditional models, achieving a normalised Root Mean Square Error (nRMSE) of 4.2% compared to 6.5% for ARIMA [Cantón Sánchez, 2020].

Direct comparisons between machine learning and statistical approaches confirm these findings. In a mini-review of forecasting methods, RF and Extreme Gradient Boosting (XGBoost) achieved RMSE reductions of 40–60% over ARIMA, alongside higher R^2 scores, showcasing superior adaptability under diverse weather conditions [Dou et al., 2023].

Deep learning models such as Long-Short Term Memory (LSTM) and Convolutional Neural Networks (CNN) are also being investigated; however, they require extensive high-resolution data, rendering them less suitable for use cases with limited historical records. The SolNet study found that in contexts with fewer than six months of data, RF and XGBoost outperformed LSTM by 20–30% in RMSE while being faster and more interpretable [Depoortere et al., 2024]

Finally, hybrid models that combine statistical and machine learning techniques, such as the wavelet-ARMA-NARX model, have demonstrated promising performance. These models separate noise (wavelet), model linear structures (ARMA), and capture nonlinear patterns (NARX). However, their complexity and deployment requirements make them difficult to justify in practical settings with time constraints and limited resources [Nazaripouya et al., 2016]

3.1 Forecasting Horizons

Forecasting requirements vary by time horizon. Minute-ahead forecasting (under 5 minutes) demands high temporal resolution and fast response. ARX models, which use recent power output and short-term weather data, perform well at this scale. A study reported a 35% RMSE reduction over persistence models for horizons under 2 hours [Bacher et al., 2009]

In contrast, day-ahead forecasting must capture full-day variability. Machine learning models like Random Forest and Gradient Boosting consistently outperform statistical models like ARIMA, particularly when using rich weather data [Fara et al., 2021]

3.2 Site-Specific vs Generalised Training

Most forecasting approaches in the literature have focused on developing models tailored to specific photovoltaic installations. These site-specific models are often optimised for conditions, employing historical production and weather data from a single location. While this method can yield accurate results, it lacks scalability, since each new site necessitates dedicated data collection, preprocessing, and model training.

An alternative approach involves training a single, generalised dataset from multiple PV sites. In theory, this enables the model to capture broader patterns and weather-production relationships that can be transferred across locations. However, there is limited practical validation of whether such generalisation enhances performance in operational settings.

This project contributes to this question by testing a model trained on more than 20 real commercial sites. The results assist in assessing whether a single model can replace individual models without compromising accuracy and whether such a setup is viable for scalable deployment.

While multi-site models have been studied in academic contexts, few works have tested generalisation across commercial PV systems with operational variability and real deployment constraints. This project tests such a setup across more than 20 real FIRN sites.

3.3 Model Choice Justification

The ARIMAX model was selected for minute-ahead forecasting due to its proven success in short-term solar power predictions. Integrating autoregressive elements with external weather factors makes ARIMAX ideal for high-frequency time series data. In past research, this model demonstrated a 35% reduction in RMSE compared to persistence baselines for forecasts less than two hours, particularly when using power and irradiance values as predictors[Bacher et al., 2009]. Its simplicity, interpretability, and low computational burden make it well-suited for minute-level operational tasks.

Random Forest and Gradient Boosting were initially selected for day-ahead forecasting due to their proven ability to capture nonlinear relationships in weather and production data. RF has shown firm performance in various studies. A comparative study of RF and GBT in solar power predictive analytics found that RF achieved the highest R^2 (0.809), the lowest

Root Mean Square Error (RMSE) (1280.797), and the lowest Mean Absolute Error (MAE) (727.005), outperforming GBT on all key metrics [Aquino, 2025]

Both RF and GBT were implemented and evaluated under controlled conditions to identify the model that best meets FIRN's operational requirements. This comparison aimed to determine which model to use. The findings are presented in Section 4.4.2.

Chapter 4: Methodology

This chapter outlines the study's methodological framework, including the complete forecasting pipeline, data sources, preprocessing, model development approaches, and validation strategies.

4.1 End-to-End Forecasting Pipeline

The forecasting approach follows a structured pipeline to process raw operational data into accurate PV power predictions. It begins with retrieving PV production data from FIRN's internal systems and historical Open-Meteo API weather forecasts. These datasets are then aligned temporally and filtered to ensure consistency, with missing or corrupted entries.

The preprocessing stage enriches the datasets with engineered features, including lagged power values and selected weather indicators. These features are chosen based on established forecasting literature and tailored to each model's requirements (ARIMAX vs. RF).

Once prepared, the data feeds into model-specific training pipelines: ARIMAX for minute-ahead and Random Forest for day-ahead forecasting. Each model is trained using historical data and validated using a holdout test set.

The final stage involves evaluation using standardised performance metrics (RMSE, MAE, MAPE, R^2), with comparisons between forecasted and actual outputs performed under conditions that simulate real-world implementation. Figure 4.1 visually represents this structured pipeline.

Figure 4.1: Overview of the forecasting pipeline.

4.2 Data Sources

Two primary datasets were utilised for this project. FIRN's internal PV production data, available at a 1-minute temporal resolution, provided the foundation for all forecasting activities. This data was accessed directly from FIRN's internal systems.

Weather data was sourced from Open-Meteo via historical weather API calls, offering hourly historical data and forecasts. Operational deployment will utilise Open-Meteo's forecast API, while evaluations in this project were conducted using Open-Meteo's historical forecast API to simulate real-world forecasting performance.

4.3 Data Preprocessing

Data preprocessing involved precise alignment of PV and weather data to identical timestamps and investigations into integrity and consistency across the input datasets. Historical PV output data were transformed into lagged variables based on prior work for the ARIMAX model [Clua, 2025]. The selection of weather variables employed a differentiated approach. We selected shortwave and diffuse radiation for the ARIMAX minute-ahead model, consistent with previous successful investigation and implementations [Clua, 2025]. In contrast, a more extensive set of variables, including cloud cover, precipitation, wind speed, humidity, and atmospheric pressure, was used for the RF-based day-ahead forecasting. This choice was based on existing literature, suggesting that more variables benefit longer horizon predictions [AlSkaif et al., 2020]

4.4 Model Development

4.4.1 Minute-Ahead Forecasting

The minute-ahead forecasting model modifies the ARIMAX structure used in previous research [Clua, 2025], shifting from day-ahead predictions to real-time minute-level forecasts. Unlike the original model, which depended on a single 24-hour lag, this revised version includes multiple recent lag features (1 to 5 minutes) and a 15-minute power slope to capture local dynamics better.

The model retains the exogenous weather inputs (shortwave and diffuse radiation) and is implemented in Python using FIRN's high-resolution data, including quantitative accuracy metrics.

The ARIMAX structure was chosen due to its low-latency nature and ability to leverage recent production history. Despite its statistical simplicity, it remains competitive for ultra-short-term applications. Minor adjustments, as stated in Section 5.2, were made to improve robustness and integration within FIRN's Python-based environment.

4.4.2 Day-Ahead Forecasting

Following the literature review and initial exploratory testing between Random Forest and Gradient Boosting Trees(presented in the appendix), Random Forest was selected as the optimal modelling approach for day-ahead forecasting. Due to its greater robustness and generalisability, Random Forest outperformed Gradient Boosting Trees. This choice aligns with comparative studies, as stated in Section 3.3 and confirmed through the following:

The Random Forest (RF) model achieved superior forecasting accuracy across all key metrics compared to the Gradient Boosting Trees (GBT) model. Specifically, RF yielded a Mean Absolute Error (MAE) of 0.153, Root Mean Square Error (RMSE) of 0.301, Mean Absolute Percentage Error (MAPE) of 10.96%, and an R^2 score of 0.983. In contrast, GBT recorded an MAE of 0.211, RMSE of 0.365, MAPE of 13.28%, and R^2 of 0.975. These results reinforce the selection of Random Forest for its higher Accuracy.

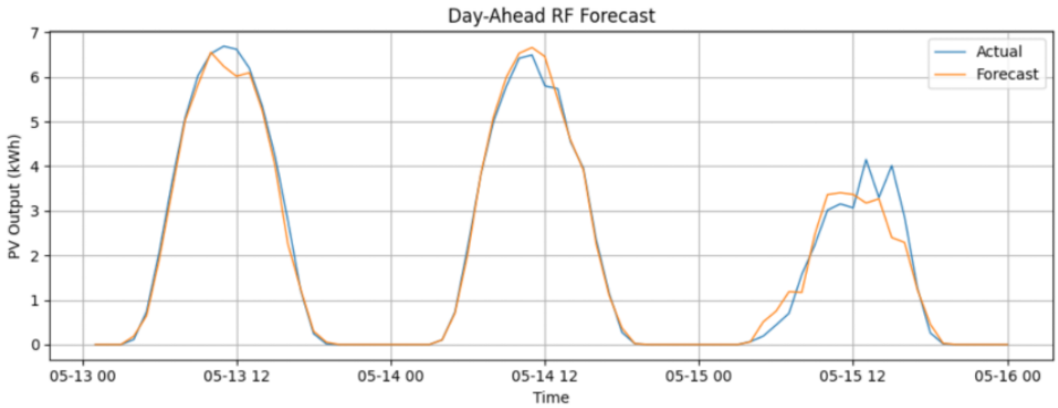

Figure 4.2: Random Forest Forecast vs Actuals

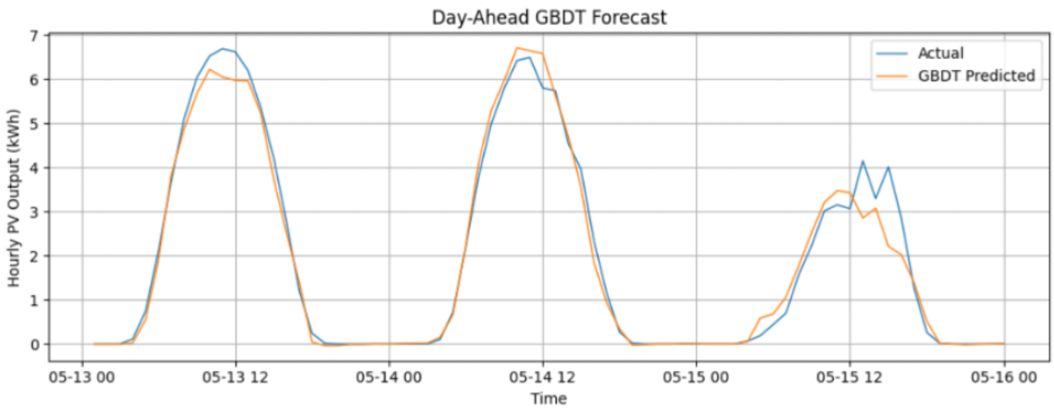

Figure 4.3: Gradient Boosting Trees Forecast vs Actuals

After selecting Random Forest as the model to be used, three distinct training strategies were subsequently evaluated for the RF model to determine the best operational approach for FIRN:

1. Site-specific model training, validated on the same site.
2. General model training on 19 sites, tested on an unseen site.
3. General model training on all 20 available sites.

4.5 Validation Strategy

The validation approach was carefully tailored to the two forecast horizons' specific requirements and intended operational contexts.

For minute-ahead forecasting (ARIMAX), validation was carried out using standard train-test splits with dedicated 5-minute holdout periods. Given the ultra-short-term nature of these predictions, forecasts were validated using historical PV data withheld from model training.

Two validation procedures were employed for day-ahead forecasting (RF). First, a holdout validation approach with a holdout 72-hour period was used to compare site-specific and general model performance. Additionally, to realistically simulate operational conditions, forecasts generated by the generalised RF model were evaluated against actual PV outputs and ARIMAX forecasts, using historical weather forecasts from Open-Meteo's API conducted on randomly selected dates. Using historical weather forecasts simulates real-world scenarios.

Performance metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2) were employed to assess forecast accuracy thoroughly. According to existing comparative studies on forecast evaluation, RMSE is particularly valuable in short-term scenarios due to its sensitivity to prediction spikes. At the same time, MAE and MAPE offer useful indicators for evaluating longer-term planning forecasts. Normalised metrics such as R^2 allow for meaningful comparisons between models across different datasets [Nguyen and Müsgens, 2022].

Chapter 5: Minute-Ahead Forecasting

5.1 Purpose

Minute-ahead photovoltaic (PV) forecasting is vital in maintaining real-time grid stability. For FIRN Energy, anticipating production fluctuations on a minute-level scale allows for timely dispatch adjustments, reducing imbalance costs, and avoiding grid disturbances. The operational need for rapid, reliable predictions makes this forecast horizon particularly sensitive to model responsiveness and deployment speed.

5.2 ARIMAX Model

A statistical ARIMAX (AutoRegressive Integrated Moving Average with eXogenous inputs) model was implemented and explicitly tuned for minute-ahead forecasting to meet this need. The selected structure, ARIMAX(1,0,1), includes one autoregressive term, no differencing, and one moving average term, making it both lightweight and effective at capturing short-term temporal patterns in PV output.

Mathematically, the model is represented as:

$$y_t = c + \phi y_{t-1} + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \theta \varepsilon_{t-1} + \varepsilon_t \tag{5.1}$$

Here, y_t denotes PV output at time t; y_{t-1} is the one-minute lag of output; $x_{1,t}$ and $x_{2,t}$ represent shortwave and diffuse radiation respectively; c is the intercept; ϕ and θ are the AR and MA coefficients; and ε_t is the residual error term.

The ARIMAX(1,0,1) model consists of three components: an autoregressive (AR) term of order 1, meaning the model uses the previous time step's PV output to inform the next prediction; an integration (I) term of 0, which indicates that the data does not need differencing to become stationary; and a moving average (MA) term of order 1, which helps the model adjust for recent forecast errors. The exogenous variables act as external inputs, providing real-time environmental context.

The model was developed and executed entirely within a Python-based environment. This choice facilitates direct integration with FIRN's existing data processing infrastructure and allows for smooth deployment, fast retraining, and automated evaluation as new data becomes available.

Input Features and Engineering

The model's input structure was designed to balance simplicity with predictive power. It integrates the following real-time features:

- **Lagged PV output (1 to 5 minutes):** These values capture the short-term autocorrelation inherent in PV generation. Including multiple lags improves the model's ability to recognise trends or ramps over a short time window.
- **Shortwave radiation:** Used as a direct proxy for solar irradiance, this feature is crucial for tracking immediate power output fluctuations caused by changing sunlight conditions, such as sudden cloud coverage [Clua, 2025].
- **Diffuse radiation:** This complements the shortwave input by accounting for indirect irradiance effects, which become particularly relevant in partially cloudy conditions [Clua, 2025].
- **Slope of power change over 15 minutes:** This engineered feature quantifies how rapidly power increases or decreases. It is calculated as the difference in PV output between t and $t - 15$, divided by the period, and strengthens the model's responsiveness.

All features were aligned using a time-index merging approach with a backwards "as-of" strategy to match PV output with the most recent weather data. Additional care was taken to remove duplicates, address missing values, and ensure no leakage across prediction windows. Minute-ahead and day-ahead forecasts were separated, utilising future exogenous data solely for forecasting and not during the training phase.

Differences from Previous Work

The model extends the work of a previous student who used an ARIMAX framework. However, that model was configured for day-ahead forecasts and relied on coarser lag structures. In contrast, this implementation is explicitly designed for minute-ahead forecasting, with key modifications:

- **Forecast Horizon:** Shifted from daily to 5-minute intervals, demanding higher temporal resolution in feature engineering and validation.
- **Lag Handling:** Instead of a fixed one-day lag, this model uses multiple recent lags (up to 5 minutes), better suited for ultra-short-term forecasting and allowing the model to pick up on immediate output dynamics.
- **Feature Expansion:** The addition of the 15-minute slope and careful variable selection.
- **Deployment Readiness:** Unlike the prior implementation, this version was written with modularity and portability in mind, allowing direct transfer into FIRN's operational workflows without external dependencies.

These changes make the model more responsive, lightweight, and suited to the high-frequency demands of real-time PV prediction.

5.3 Results

The ARIMAX model was validated using a five-minute holdout window, aligned with FIRN's operational use case. The following metrics were computed:

Metric	Value
MAE	0.013
RMSE	0.015
MAPE	1.24%
R^2	0.716

Table 5.1: Minute-Ahead Forecasting Performance (ARIMAX)

These results confirm that the model accurately tracks minute-level fluctuations and maintains consistent performance even under short evaluation horizons.

A forecast vs. actual plot was generated and included below to visually illustrate how closely the model's predictions track real PV output. The x-axis reflects the five-minute forecast horizon, covering the next 5 minutes (t+1 to t+5) based on the most recent available data. The values begin at 5 due to how the final data slice was indexed, though they represent consecutive forecast steps rather than clock time. The y-axis shows the PV power output in kilowatt-hours (kWh), allowing for a direct comparison between predicted and observed production levels over this short window.

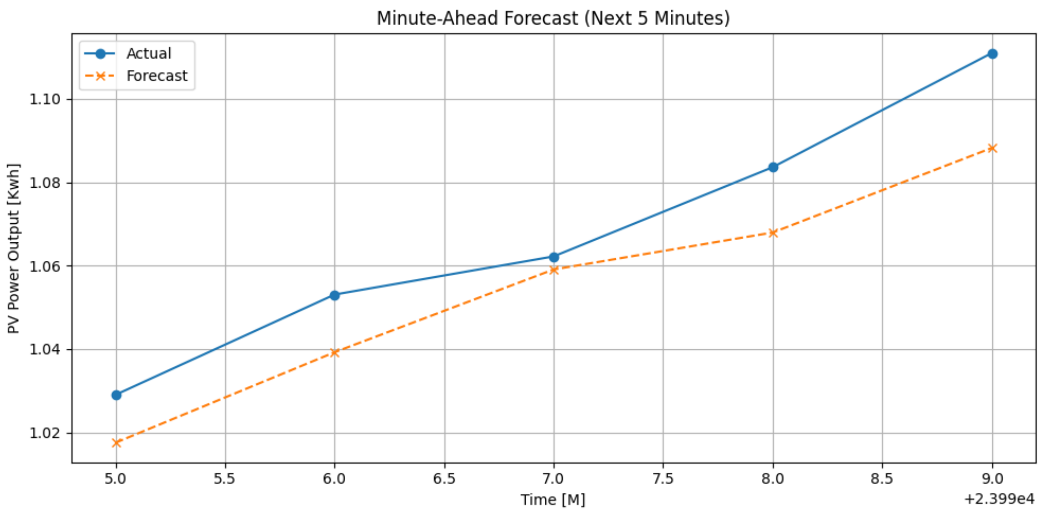

Figure 5.1: Minute-Ahead Forecast (Next 5 Minutes)

This test confirms the model's readiness for real-time deployment and effectiveness in supporting FIRN's imbalance mitigation goals.

Chapter 6: Day-Ahead Forecasting

6.1 Purpose

Day-ahead PV forecasting is pivotal in FIRN Energy's operational strategy, primarily supporting curtailment decisions and participation in electricity market bidding. Unlike minute-ahead forecasting, which addresses immediate operational responses, day-ahead predictions enable longer-term planning, providing FIRN with actionable insights to optimise grid interactions, mitigate curtailment risks, and strategically manage market activities.

Reliable day-ahead forecasts directly influence FIRN's ability to accurately schedule production, maximise revenue, and minimise financial penalties resulting from forecast errors.

6.2 Model Used

Random Forest Overview

Random Forest is an ensemble-based machine learning technique widely used for regression and classification tasks. During training, it builds multiple decision trees independently, each of which predicts the target variable based on a subset of randomly selected features and observations. The final prediction is obtained by aggregating the predictions from all individual trees.

Each decision tree in the ensemble is trained on a different bootstrap sample of the training data, and at each split in the tree, only a random subset of the available features is considered. This combination of bootstrapping and feature randomness introduces diversity among the trees, which helps prevent overfitting and reduces variance. As a result, the Random Forest model tends to generalise well, especially when dealing with high-dimensional or noisy datasets.

This ensemble approach significantly reduces overfitting and enhances the model's ability to capture complex, nonlinear relationships between input features and the target variable, making it well-suited for day-ahead PV forecasting [Sal, 2024].

Model Implementation and Features

The RF model developed for FIRN's day-ahead forecasting utilises an extensive set of input features to predict daily PV production accurately. The implementation pipeline consists of several clear steps:

Data Preparation and Feature Engineering The raw input data, comprising historical PV production and weather observations, was first loaded and processed to ensure consistent timestamps. Autoregressive (lagged) features of PV power output were created at intervals of 1, 2, 3, 6, 12, and 24 hours, explicitly capturing short-to-medium term dependencies.

The selected weather inputs included shortwave radiation, temperature, humidity, wind speed, cloud cover, precipitation, atmospheric pressure, and dew point, chosen based on previous research indicating their importance in influencing PV output, as stated in Section 4.3. Additionally, calendar-based features (hour and day of the week) were integrated to capture daily and weekly cyclical patterns in solar production.

Data Cleaning and Splitting Rows containing missing values in any essential feature were removed to maintain data quality. The dataset was then split into training and holdout validation sets. Specifically, the last 72 hours (3 days) of data were reserved for model testing.

Model Training A basic Random Forest regression model was configured and trained on the processed training dataset. The final RF configuration included 300 decision trees (`n_estimators=300`) with a maximum depth of 15 (`max_depth=15`). These parameters were selected through iterative testing to achieve the best balance between model complexity and generalisation capability.

6.3 Model Testing Strategy

To identify the most effective forecasting approach for FIRN's day-ahead needs, three distinct Random Forest (RF) model training strategies were implemented and evaluated:

Site-Specific Model: Models were initially trained and tested independently on data from individual sites. This approach provided a baseline to gauge the predictive capacity achievable when models are tailored exclusively to local conditions.

Generalised Model (Leave-One-Out): A general model was trained on a group of sites, excluding a specific test site. The model's accuracy on this previously unseen site was then assessed. This strategy tested the robustness of generalised training. Still, it revealed lower accuracy than the entirely site-specific model, likely due to limited exposure to the site-specific characteristics of the excluded location.

Generalised Model (All Sites): A comprehensive general model was trained using combined data from all available sites, including the evaluation site itself. This approach, leveraging a broader dataset and increased feature variability, achieved superior accuracy and generalisation across all sites compared to the other two methods.

This testing sequence clarified the benefits of generalised training. It demonstrated that combining data from multiple PV installations creates a more robust forecasting tool that leverages patterns common across sites while retaining sensitivity to individual site conditions.

6.4 Final Model Results

Table 6.1 summarises the forecasting accuracy for the three tested sites, comparing the site-specific models, the general models excluding each site ("Leave-One-Out"), and the general model including all sites.

Site	Training Strategy	MAE	RMSE	MAPE (%)	R^2
EDSS_KERCKHOVE	Site-Specific	0.265	0.437	13.82	0.958
	General (Excl. site)	0.289	0.501	16.62	0.945
	General (All sites)	0.215	0.373	11.20	0.969
GULLEGEM_DEMAN	Site-Specific	0.299	0.612	16.88	0.929
	General (Excl. site)	0.381	0.642	26.37	0.922
	General (All sites)	0.232	0.430	14.71	0.965
WERVIK_DEDECKER_POWERBLOCK	Site-Specific	5.657	9.754	16.82	0.941
	General (Excl. site)	7.584	13.166	20.66	0.893
	General (All sites)	2.217	4.077	5.82	0.990

Table 6.1: Day-Ahead Forecasting Accuracy Comparison

The results demonstrate the superior accuracy and generalisation performance of the final general model trained on data from all available sites. Across all evaluated sites, the *General (All sites)* approach consistently outperformed both the single-site and the leave-one-out generalised approach. Notably, the fully generalised model significantly reduced forecast errors, substantially improving site-specific and leave-one-out model accuracies.

Visual Evaluation

To illustrate the performance of the selected general model, Figure 6.1, Figure 6.2 and Figure 6.3 below present a detailed forecast versus actual PV production plot for the `EDSS_KERCKHOVE` site as an example. Additional plots for the other sites (`GULLEGEM_DEMAN` and `WERVIK_DEDECKER_POWERBLOCK`) can be found in the Appendix.

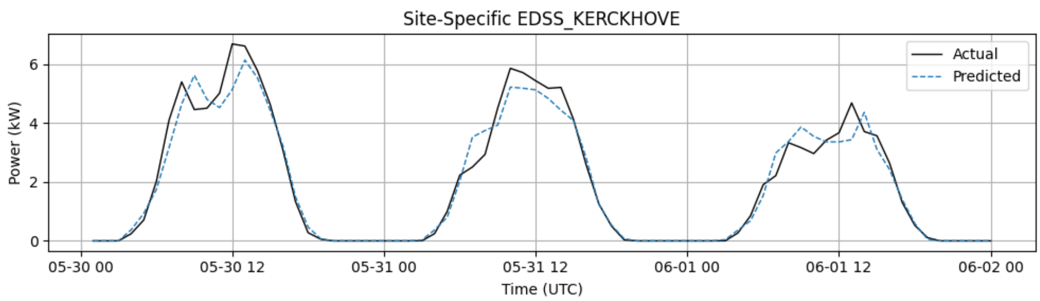

Figure 6.1: Day-Ahead Forecast vs. Actual PV Production (`EDSS_KERCKHOVE`)

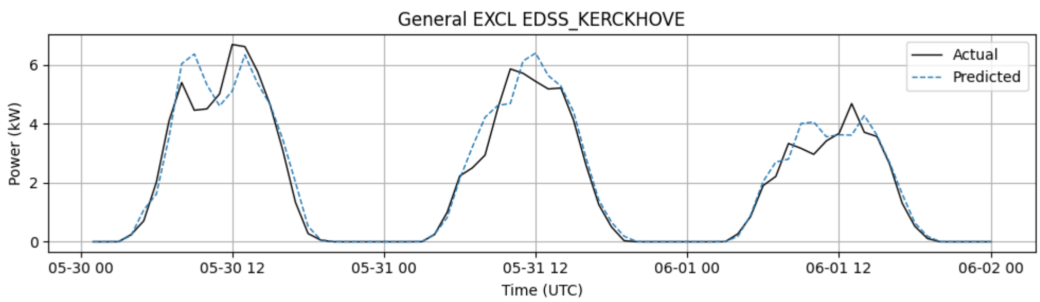

Figure 6.2: Day-Ahead Forecast vs. Actual PV Production (`EDSS_KERCKHOVE`)

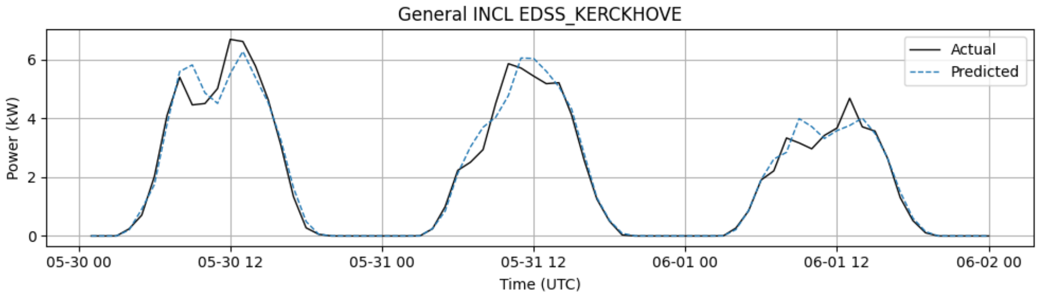

Figure 6.3: Day-Ahead Forecast vs. Actual PV Production (`EDSS_KERCKHOVE`)

6.5 Final Model Recommendation

The generalised Random Forest model, developed using data from all available PV sites, is recommended for FIRN Energy's operational day-ahead forecasting based on extensive comparative analysis. This model balances accuracy and operational simplicity, offering robust and consistent performance across various installations. It also facilitates model management and significantly reduces forecasting errors compared to the other evaluated approaches. FIRN should utilise this General Model for sites whose data contributed to training the model for best performance.

Chapter 7: Comparative Evaluation with Original Model

7.1 Evaluation Methodology

To objectively compare the newly implemented Random Forest model against the previously developed ARIMAX approach, identical evaluation procedures were applied. Both models were tested on the same three representative FIRN sites (`EDSS_KERCKHOVE`, `GULLEGEM_DEMAN`, and `WERVIK_DEDECKER_POWERBLOCK`). Each site's historical weather forecasts were retrieved from the Open-Meteo historical forecast archive to simulate realistic forecasting conditions, ensuring no data leakage and fair, consistent comparisons.

For each site, a 72-hour (3-day) evaluation period was selected randomly from available dates, and forecasts from ARIMAX and RF models were generated. Both models utilised lagged PV output and historical weather forecasts. RF used comprehensive lag structures (1, 2, 3, 6, 12, and 24 hours) and additional calendar-based features, while ARIMAX relied on a single 24-hour lag with shortwave and diffuse radiation inputs.

Performance was evaluated using MAE, Root Mean Square Error RMSE, MAPE, and R^2.

7.2 Performance Comparison

Table 7.1 summarises the forecasting accuracy metrics for ARIMAX and Random Forest across the three evaluated sites.

Site	**Model**	**MAE**	**RMSE**	**MAPE (%)**	$\mathbf{R^2}$
EDSS_KERCKHOVE	ARIMAX	0.573	0.996	40.56	0.793
	RF	0.105	0.199	9.19	0.992
GULLEGEM_DEMAN	ARIMAX	0.312	0.595	30.74	0.882
	RF	0.154	0.296	20.83	0.971
WERVIK_DEDECKER_POWERBLOCK	ARIMAX	10.249	16.860	86.66	0.175
	RF	3.343	7.775	27.55	0.825

Table 7.1: Forecast Accuracy Comparison (ARIMAX vs. Random Forest)

As clearly shown, the Random Forest model consistently outperformed ARIMAX on all metrics across each site. Notably, the improvement in forecasting accuracy is most substantial at the WERVIK site, where the ARIMAX model struggled significantly compared to the RF model. Reducing MAPE from 86.66% to 27.55% represents a percentage decrease of 68% in MAPE.

7.3 Summary of Model Improvements

The comprehensive comparative evaluation demonstrates several key improvements delivered by the Random Forest model:

- **Increased Accuracy:** RF achieved significantly lower MAE, RMSE, and MAPE values, indicating tighter forecasting accuracy and reduced prediction errors.
- **Enhanced Generalisation:** RF showed consistently strong performance across different FIRN sites, demonstrating its superior ability to generalise and adapt to various operational conditions compared to ARIMAX.
- **Python Implementation:** The new RF model and pipeline are fully implemented within a Python environment, facilitating seamless integration into FIRN's existing data processing and deployment infrastructure.
- **Deployment Readiness:** The generalised RF model trained across multiple sites provides FIRN with a robust, ready-to-deploy forecasting tool, streamlining future site integrations and significantly reducing operational overhead.

7.4 Visual Comparison

For a visual representation of the comparative performance, Figure 7.1 illustrates the forecast accuracy for the `EDSS_KERCHKHOVE` site. Additional comparative plots for the other two sites are included in the Appendix in Figures 7 and 8.

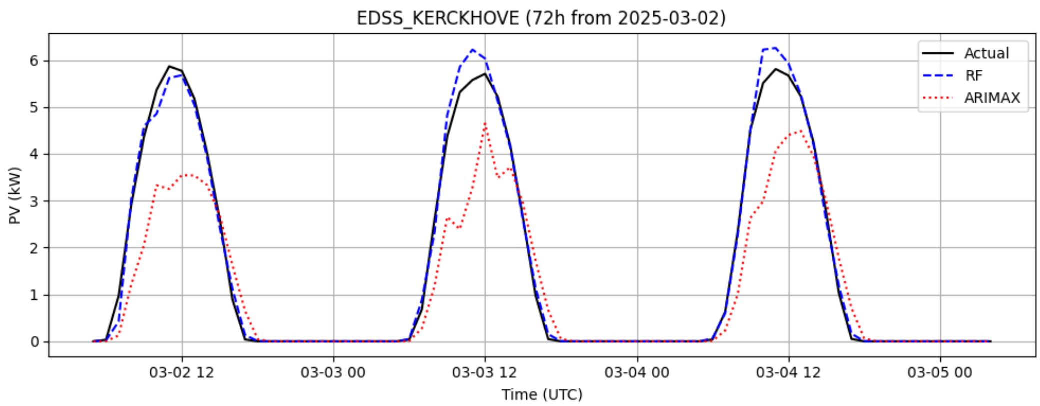

Figure 7.1: Comparative Forecasts for `EDSS_KERCHKHOVE` – ARIMAX vs. RF

This thorough evaluation establishes the Random Forest model as a superior choice for FIRN's day-ahead forecasting, offering measurable accuracy improvements and operational advantages over the previously developed ARIMAX approach.

Chapter 8: Discussion

The main research question throughout this project was: What forecasting setup provides accurate and scalable short-term predictions for photovoltaic (PV) energy across multiple FIRN Energy sites, utilising both production and weather data? The results from this project demonstrate that this is achievable.

For the first sub-question, which asked which modeling approach is the most effective for generating reliable minute-ahead forecasts to facilitate real-time imbalance mitigation, the adapted ARIMAX model proved suitable, offering strong short-term accuracy and responsiveness.

The second sub-question focused on identifying a design framework that ensures accurate day-ahead forecasts while maintaining robustness across various sites. The Random Forest model—trained on diverse site data—stood out for its generalisation capabilities and stability.

Finally, the third sub-question inquired whether creating individual forecasting models for each site is more advantageous than developing a unified model that leverages data from multiple sites. The comparative results showed that the unified general model consistently outperformed the site-specific versions, confirming its value for scalable deployment.

8.1 Strengths

The forecasting models developed in this project demonstrate substantial improvements in accuracy over the previously implemented ARIMAX approach. The significant reduction in forecasting errors across multiple key metrics (MAE, RMSE, MAPE, and R^2) indicates the effectiveness and robustness of the Random Forest model.

Furthermore, the developed models are immediately deployable by FIRN Energy, having been built explicitly within FIRN's existing Python-based infrastructure. The provided codebase's transparency and modular structure enhance usability, allowing FIRN to easily adapt and extend the models for future sites or additional operational requirements.

Comprehensive testing on over 20 real-world PV installations ensures broad applicability and high generalisation capability. This extensive validation across diverse operational conditions significantly strengthens confidence in the model's real-world performance.

8.2 Limitations

Despite these substantial advancements, certain limitations should be acknowledged. The models have been trained exclusively on data from recent months, predominantly covering autumn and winter conditions. Thus, they lack exposure to summer weather dynamics characterised by higher irradiance variability and increased temperature impacts on PV efficiency. This seasonality gap may temporarily reduce predictive performance until addressed.

Additionally, the current model relies on hourly-resolution weather forecasts, potentially missing valuable insights from minute-level granularity. A higher-resolution dataset could enhance minute-ahead model responsiveness, particularly during rapidly changing conditions like sudden cloud coverage or solar ramps.

While highly effective across most scenarios, the generalised Random Forest model may underperform in certain extreme edge cases—such as rare meteorological events, prolonged periods of unusual weather, or sites with atypical system configurations—due to the limited historical representation of these scenarios in training datasets.

8.3 Future Work

Several promising avenues of future work are recommended to refine and enhance forecasting accuracy and operational usability.

Collecting comprehensive summer data will strengthen the model's robustness across all seasonal variations. Retraining models on an extended dataset will improve their ability to generalise effectively throughout the year.

Developing automated workflows for regular retraining will allow the models to adapt continuously to changing operational conditions and new site-specific data, minimising manual intervention and sustaining optimal forecast performance.

Exploring real-time retraining approaches using rolling windows could enhance minute-level predictions by continuously integrating the most recent data, rapidly adapting to transient operational changes, and improving responsiveness to short-term weather variations.

By pursuing these targeted improvements, FIRN Energy can solidify and expand upon the forecasting successes demonstrated in this project, achieving sustained operational efficiency and accuracy in solar production forecasting.

Chapter 9: Conclusion and Recommendations

9.1 Conclusion

This project successfully developed robust forecasting models tailored to FIRN Energy's operational requirements. An improved ARIMAX model was implemented for minute-ahead forecasting, specifically adapted to leverage immediate historical data and selected environmental predictors, significantly enhancing short-term prediction accuracy and responsiveness.

For day-ahead forecasting, extensive comparative evaluation demonstrated that a generalised Random Forest model, trained on data from over 20 different PV installations, provides the highest overall predictive accuracy and robustness across diverse operational conditions. This generalised approach proved superior to site-specific models and substantially improved over the previous ARIMAX-based forecasts.

9.2 Recommendations for FIRN

The following key recommendations are presented for immediate implementation based on thorough analysis and extensive validation.

FIRN is advised to employ the adapted ARIMAX model for minute-ahead forecasting in real-time operational contexts. This model supports rapid response to production fluctuations and improves grid stability by enhancing prediction responsiveness at the minute scale.

FIRN should implement the generalised Random Forest model at all operational sites for day-ahead forecasting tasks such as curtailment planning and market bidding. Its consistently high accuracy across varying conditions makes it an effective tool for enhancing operational planning and strategic market engagement.

An automated retraining pipeline should be established to ensure model performance remains aligned with evolving conditions. This process will allow seamless new data integration, adaptive model tuning, and continuous performance optimisation with minimal manual intervention.

Lastly, the modular forecasting engine and codebase developed in this project should be leveraged as the foundation for future forecasting infrastructure. Its transparency, scalability, and ease of integration make it an ideal platform for ongoing development and deployment across expanding operational scopes.

9.3 Final Deliverable for FIRN

The forecasting solution developed in this project is designed for use in a Python environment, making it user-friendly and ready for operation. It includes well-documented code for minute-ahead and day-ahead forecasting models that integrate smoothly with FIRN Energy's existing Python infrastructure.

The package also contains standalone scripts and instructions for accessing various Open-Meteo APIs, allowing for automated retrieval of real-time and historical weather data. Additionally, data folders store FIRN's proprietary photovoltaic production datasets, simplifying the integration of site-specific data or updates to the models.

The environment is modular and extensible, which supports easy deployment, retraining, and scalability. As FIRN expands its portfolio of photovoltaic sites, this forecasting platform serves as a solid foundation for its operational needs.

An overview of the codes used by FIRN will be included in the Appendix.

9.4 Final Takeaway

Accurate short-term PV forecasting strengthens both grid stability and market performance. This project delivers scalable, validated forecasting models that FIRN Energy can confidently deploy immediately to realise measurable operational and strategic advantages.

References

[Sal, 2024] (2024). Random forest algorithm overview. *Babylonian Journal of Machine Learning*, 2024:69–79.

[AlSkaif et al., 2020] AlSkaif, T., Dev, S., Visser, L., Hossari, M., and van Sark, W. (2020). A systematic analysis of meteorological variables for pv output power estimation. *Renewable Energy*, 153:12–22.

[Aquino, 2025] Aquino, B. (2025). Exploring machine learning models for predictive analytics in solar power generation. *Journal of Information Systems Engineering and Management*, 10:1096–1105.

[Asiedu et al., 2024] Asiedu, S. T., Nyarko, F. K. A., Boahen, S., Effah, F. B., and Asaaga, B. A. (2024). Machine learning forecasting of solar pv production using single and hybrid models over different time horizons. *Heliyon*, 10(7):e28898.

[Bacher et al., 2009] Bacher, P., Madsen, H., and Nielsen, H. A. (2009). Online short-term solar power forecasting. *Solar Energy*, 83(10):1772–1783.

[Bousso, 2025] Bousso, R. (2025). The path to cheap power will be very expensive. Accessed: 2025-06-12.

[Cantón Sánchez, 2020] Cantón Sánchez, Á. (2020). Development of a day-ahead pv power forecasting tool based on random forests and artificial neural networks techniques. Unpublished.

[Clua, 2025] Clua, M. M. (2025). Short-term forecasting model for optimizing photovoltaic electricity production.

[Cormode et al., 2014] Cormode, D., Lorenzo, A., Holmgren, W., Chen, S., and Cronin, A. (2014). The economic value of forecasts for optimal curtailment strategies to comply with ramp rate rules. In *2014 IEEE 40th Photovoltaic Specialist Conference (PVSC)*, pages 2070–2075.

[Depoortere et al., 2024] Depoortere, J., Driesen, J., Suykens, J., and Kazmi, H. S. (2024). Solnet: Open-source deep learning models for photovoltaic power forecasting across the globe.

[Dou et al., 2023] Dou, Y., Tan, S., and Xie, D. (2023). Comparison of machine learning and statistical methods in the field of renewable energy power generation forecasting: a mini review. *Frontiers in Energy Research*, 11.

[Fara et al., 2021] Fara, L., Diaconu, A., Craciunescu, D., and Fara, S. (2021). Forecasting of energy production for photovoltaic systems based on arima and ann advanced models. *International Journal of Photoenergy*, 2021:1–19.

[Nazaripouya et al., 2016] Nazaripouya, H., Wang, B., Wang, Y., Chu, P., Pota, H., and Gadh, R. (2016). Univariate time series prediction of solar power using a hybrid wavelet-arma-narx prediction method. pages 1–5.

[Nguyen and Müsgens, 2022] Nguyen, T. N. and Müsgens, F. (2022). What drives the accuracy of pv output forecasts? *Applied Energy*, 323:119603.

[Pelland et al., 2013] Pelland, S., Remund, J., Kleissl, J., Oozeki, T., and De Brabandere, K. (2013). *Photovoltaic and Solar Forecasting: State of the Art.*

Appendix

Additional Forecast Visualisations

The following figures include forecast vs. actual plots for sites not presented in the main report body but evaluated during testing. These graphs support the findings discussed in Chapters 6 and 7.

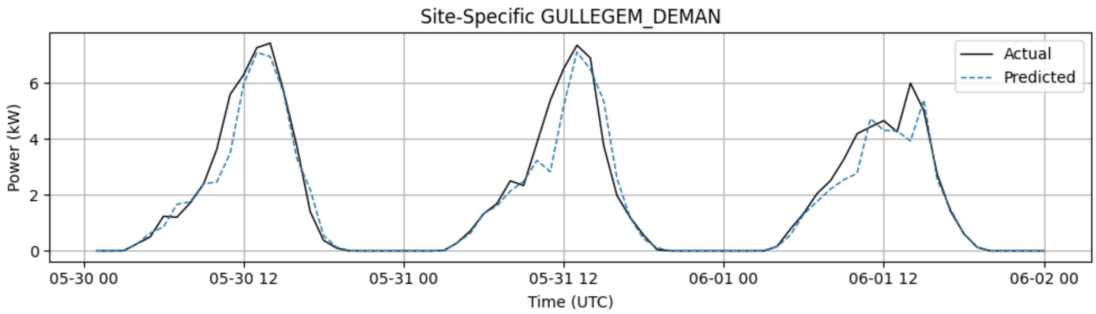

Figure 1: Site-Specific Forecast for ***GULLEGEM_DEMAN***

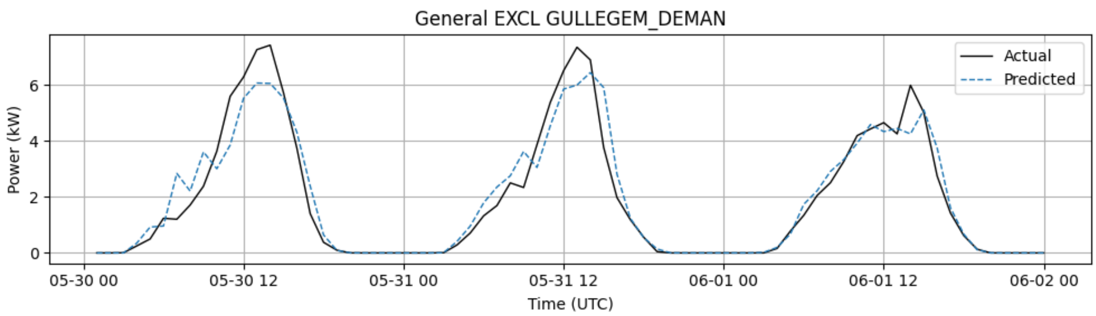

Figure 2: General Excluding ***GULLEGEM_DEMAN***

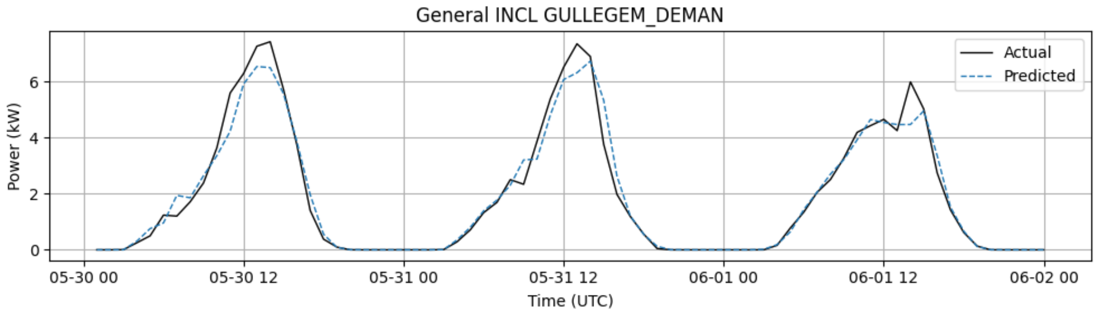

Figure 3: General Including `GULLEGEM_DEMAN`

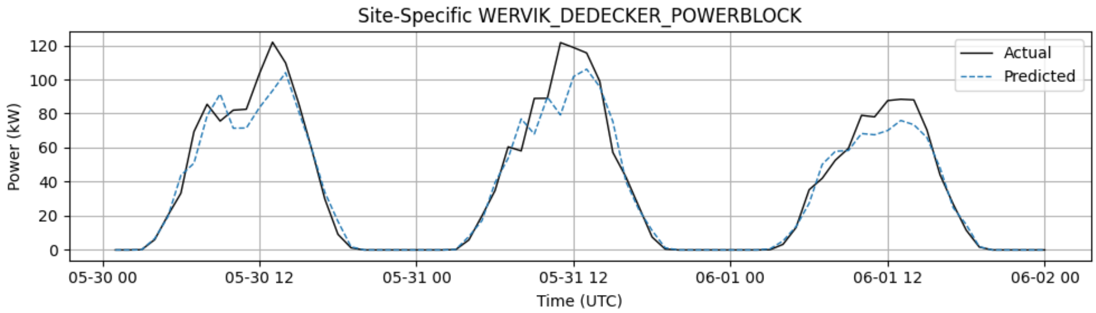

Figure 4: Site-Specific Forecast for `WERVIK_DEDECKER_POWERBLOCK`

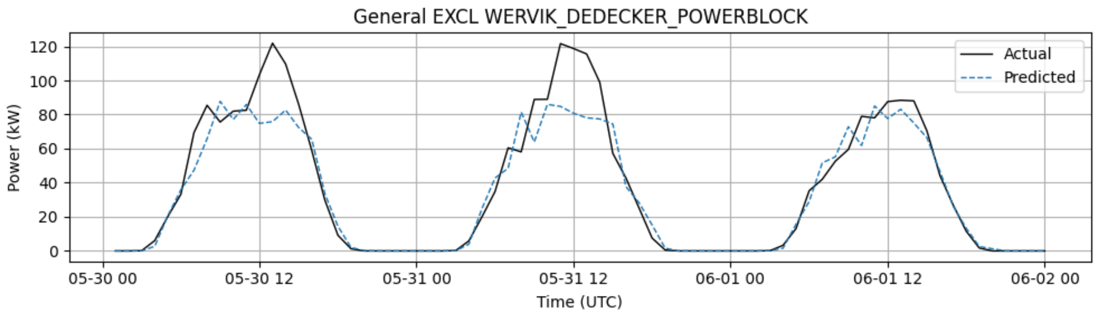

Figure 5: General Excluding `WERVIK_DEDECKER_POWERBLOCK`

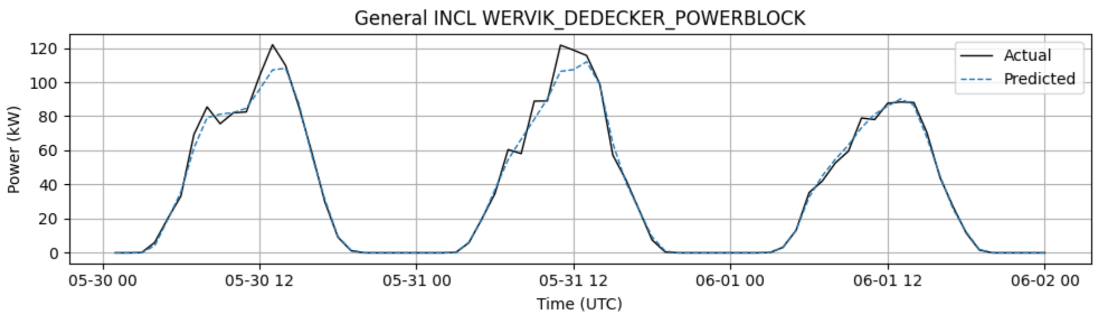

Figure 6: General Including `WERVIK_DEDECKER_POWERBLOCK`

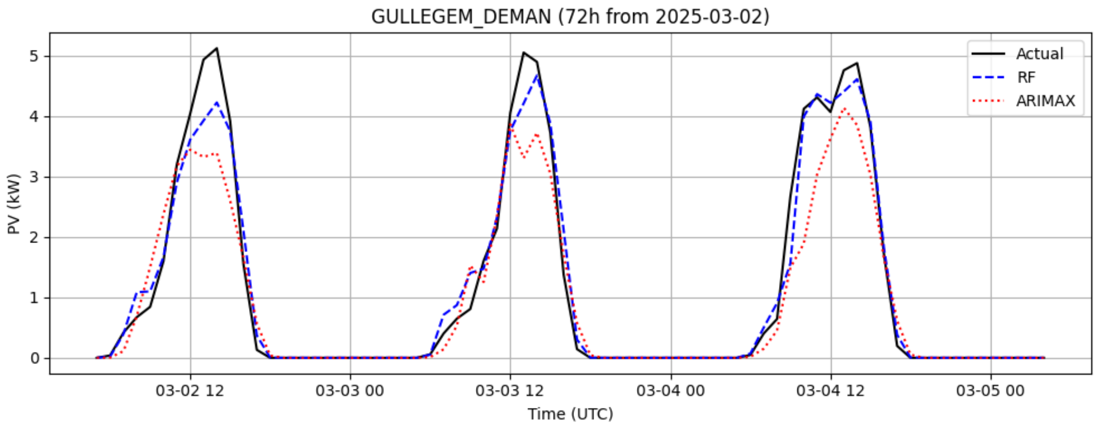

Figure 7: `GULLEGEM_DEMAN` *– Random Forest vs. ARIMAX (72h from 2025-03-02)*

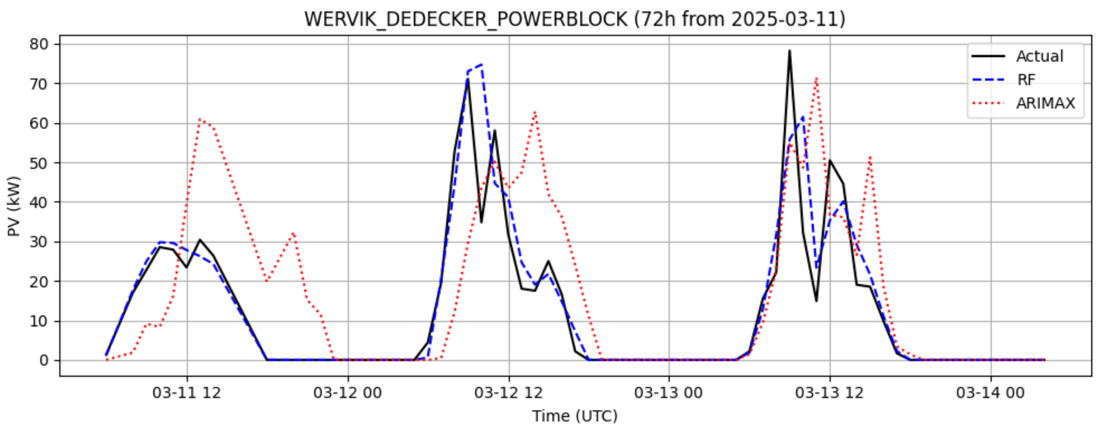

Figure 8: `WERVIK_DEDECKER_POWERBLOCK` *– Random Forest vs. ARIMAX (72h from 2025-03-11)*

Codes

Minute-Ahead

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import (mean_absolute_error,mean_squared_error,mean_absolute_percentage_error,r2_score)

def prepare_minute_data(pv_df, weather_df, max_lag=5):
    """
    Merge PV and weather minute data, create lagged PV features and a slope.
    """
    pv = pv_df.sort_values("time").set_index("time")
    w  = weather_df.sort_values("time").set_index("time")

    # merge nearest past weather onto each PV timestamp
    merged = pd.merge_asof(pv, w, left_index=True, right_index=True, direction="backward")

    # create lags power_lag_1m     power_lag_{max_lag}m
    for lag in range(1, max_lag + 1):
        merged[f"power_lag_{lag}m"] = merged["power"].shift(lag)

    # 15 min  slope feature
    merged["slope_15m"] = (merged["power"] - merged["power"].shift(15)) / 15

    merged.dropna(inplace=True)
    return merged

def train_and_forecast(df, exog_vars, endog_var="power", steps=5):
    """
    Train ARIMAX
    """
    df = df.set_index("time")
    df = df[~df.index.duplicated(keep="last")]

    # fit
    model = ARIMA(
        endog = df[endog_var],
        exog  = df[exog_vars],
        order = (1, 0, 1),
        trend = "n"
    )
    res = model.fit()

    # forecast the last `steps` rows
    exog_fc = df[exog_vars].iloc[-steps:]
    fc      = res.forecast(steps=steps, exog=exog_fc)
```

```python
    return res, fc

def evaluate_and_plot(df, forecast, steps=5):
    """
    Compute MAE, RMSE, MAPE, R  vs. the true last `steps` minutes,
    then plot Actual vs Forecast.
    """
    actual = df["power"].iloc[-steps:]
    forecast.index = actual.index  # align

    mae   = mean_absolute_error(actual, forecast)
    rmse  = np.sqrt(mean_squared_error(actual, forecast))
    mape  = mean_absolute_percentage_error(actual.replace(0, np.nan),
        forecast) * 100
    r2    = r2_score(actual, forecast)

    print("\nForecast Accuracy:")
    print(f"  MAE   : {mae:.3f}")
    print(f"  RMSE  : {rmse:.3f}")
    print(f"  MAPE  : {mape:.2f}%")
    print(f"  R      : {r2:.3f}")

    plt.figure(figsize=(10,4))
    plt.plot(actual.index, actual,   label="Actual",   marker="o")
    plt.plot(forecast.index, forecast, label="Forecast", marker="x",
        linestyle="--")
    plt.title(f"Minute Ahead Forecast (next {steps} minutes)")
    plt.xlabel("Time")
    plt.ylabel("PV Power [kW]")
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.show()

if __name__ == "__main__":
    # 1) load your minute resolution data for the specific site
    pv_df      = pd.read_csv("data/pv",parse_dates=["time"])
    weather_df = pd.read_csv("data/weather, parse_dates=["time"])

    # 2) merge & feature engineer
    df = prepare_minute_data(pv_df, weather_df, max_lag=5)
    print("Prepared data:", df.shape, "rows")

    # 3) define exogenous features
    exog_vars = ["shortwave_radiation", "diffuse_radiation"] + \
                [f"power_lag_{i}m" for i in range(1, 6)] + \
                ["slope_15m"]

    # 4) train & forecast
    steps = 5
    model_res, forecast = train_and_forecast(df, exog_vars, endog_var=
        "power", steps=steps)
```

```
    # 5) evaluate & plot
    evaluate_and_plot(df, forecast, steps=steps)
```

Day-Ahead

```
import os
import pandas as pd
import numpy as np
import requests
from datetime import datetime, timedelta
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor


def load_merged_site_dataframe(loc):
    """
    Reads a CSV with PV values and Weather values merged into a
        DataFrame indexed by time.
    Example path below Data/Weather/merged...
    """
    path = f"Data/Weather/merged_{loc}_hourly.csv"
    df = pd.read_csv(path, parse_dates=["time"]).set_index("time")
    return df

def add_features(df):
    """
    Adds power lags (1,2,3,6,12,24h) and calendar features (hour, dow)
        .
    Drops any rows with missing values.
    """
    df = df.copy()
    for lag in (1,2,3,6,12,24):
        df[f"power_lag{lag}"] = df["power"].shift(lag)
    df["hour"] = df.index.hour
    df["dow"]  = df.index.dayofweek

    weather_cols = [
        "sw_radiation","d_radiation","temperature","humidity",
        "wind_speed","cloud_cover","precipitation","pressure","
            dew_point"
    ]
    lag_cols = [f"power_lag{l}" for l in (1,2,3,6,12,24)]
    df = df.dropna(subset=weather_cols + lag_cols + ["power"])
    return df

def fetch_hourly_forecast(lat, lon, start_date, end_date):
    """
    Calls Open-Meteo to get an hourly forecast for [start_date..
        end_date].
    Ideally for the next day or 2
    """
    url = "https://api.open-meteo.com/v1/forecast"
```

```python
    params = {
        "latitude": lat,
        "longitude": lon,
        "start_date": start_date,
        "end_date":   end_date,
        "hourly": ",".join([
            "shortwave_radiation","diffuse_radiation","temperature_2m",
            "relativehumidity_2m","windspeed_10m","cloudcover",
            "precipitation","pressure_msl","dewpoint_2m"
        ]),
        "timezone": "UTC"
    }
    r = requests.get(url, params=params)
    r.raise_for_status()
    H = r.json()["hourly"]
    df = pd.DataFrame({
        "time": pd.to_datetime(H["time"]),
        "sw_radiation": np.array(H["shortwave_radiation"], dtype=float),
        "d_radiation":  np.array(H["diffuse_radiation"], dtype=float),
        "temperature":  np.array(H["temperature_2m"], dtype=float),
        "humidity":     np.array(H["relativehumidity_2m"], dtype=float),
        "wind_speed":   np.array(H["windspeed_10m"], dtype=float),
        "cloud_cover":  np.array(H["cloudcover"], dtype=float),
        "precipitation":np.array(H["precipitation"], dtype=float),
        "pressure":     np.array(H["pressure_msl"], dtype=float),
        "dew_point":    np.array(H["dewpoint_2m"], dtype=float)
    })
    return df.set_index("time")

#

# MAIN: TRAIN AND FORECAST FOR A SINGLE SITE
#

if __name__ == "__main__":
    # ----- USER INPUT -----
    loc        = ""     # site ID (filename suffix)
    latitude   =            # site latitude
    longitude  =            # site longitude
    forecast_days = 1       # how many days ahead to forecast, 1 or 2 ideally
    output_csv    = f"{loc}_dayahead_RF.csv"
    # -----------------------

    # 1) TRAIN: load all sites, build general RF
    weather_folder = "Data/Weather"
    files = sorted(f for f in os.listdir(weather_folder)
```

```
                    if f.startswith("merged_") and f.endswith("_hourly.
                        csv"))
all_frames = []
for f in files:
    site = f.replace("merged_","").replace("_hourly.csv","")
    df   = load_merged_site_dataframe(site)
    all_frames.append(add_features(df))

general_df = pd.concat(all_frames)
features = [
    "sw_radiation","d_radiation","temperature","humidity",
    "wind_speed","cloud_cover","precipitation","pressure","
       dew_point",
    "power_lag1","power_lag2","power_lag3","power_lag6","
       power_lag12","power_lag24",
    "hour","dow"
]
rf = RandomForestRegressor(n_estimators=300, max_depth=15,
   random_state=42)
rf.fit(general_df[features], general_df["power"])
print(f"[TRAINED] General RF on {len(general_df)} hours")

# 2) PREP: load the single sites historical data to get last 24
   h for lags
df_act = load_merged_site_dataframe(loc).sort_index()
df_feat = add_features(df_act)
last_time = df_feat.index.max()

# 3) FORECAST: fetch weather forecast for next `forecast_days`
   days
start_fc = (last_time + timedelta(hours=1)).strftime("%Y-%m-%d")
end_fc   = (last_time + timedelta(days=forecast_days)).strftime("%
   Y-%m-%d")
df_fc    = fetch_hourly_forecast(latitude, longitude, start_fc,
   end_fc)
df_fc    = df_fc.sort_index().iloc[:forecast_days*24]

# 4) BUILD feature DataFrame for forecast window
df_fc["power"] = np.nan
for lag in (1,2,3,6,12,24):
    # fill first `lag` rows from last known actuals, then NaNs
    vals = df_act["power"].iloc[-lag:].values.tolist() + [np.nan
       ]*(len(df_fc)-lag)
    df_fc[f"power_lag{lag}"] = vals

df_fc["hour"] = df_fc.index.hour
df_fc["dow"]  = df_fc.index.dayofweek

# drop any rows where lags are still missing
df_fc = df_fc.dropna(subset=features)

# 5) PREDICT & PLOT
df_fc["predicted_power"] = rf.predict(df_fc[features])
```

```
plt.figure(figsize=(10,4))
plt.plot(df_fc.index, df_fc["predicted_power"], label="Forecast",
    color="C0")
plt.title(f"{loc} {forecast_days}-Day Ahead RF Forecast")
plt.xlabel("Time (UTC)"); plt.ylabel("PV Power (kW)")
plt.legend(); plt.grid(True); plt.tight_layout(); plt.show()

# 6) EXPORT into a CSV for Curtailment Planning
df_fc[["predicted_power"]].rename(columns={"predicted_power": loc
    }) \
     .to_csv(output_csv)
print(f"[EXPORTED] {output_csv}")
```

Disclaimers

Legal

This report has been produced in the framework of an educational program at the University of Groningen, Netherlands, Faculty of Science and Engineering, Industrial Engineering and Management (IEM) Curriculum. No rights may be claimed based on this report. Citations are only allowed with explicit reference to the status of the report as a product of a student project.

For privacy and data security reasons, the whole Python project environment will not be publicly released. The codebase includes access credentials, internal data from FIRN Energy, and sensitive site-specific details such as geographic coordinates and database connection strings. To protect FIRN's operational integrity and confidentiality, this information will remain private and will not be included in any public version of this report or its associated deliverables.

Use of AI Tools

In accordance with the University of Groningen's guidelines on the responsible use of Generative AI (GenAI) in education, this project used AI tools in a limited and transparent manner to support the development of the report and associated materials.

Tools Used: ChatGPT (by OpenAI) and Grammarly Pro's AI-enhanced grammar suggestions were utilised during the writing and revision process.

Purpose of Usage: ChatGPT was used to assist in refining phrasing and improving paragraph structure. Grammarly Pro enhanced the document's grammar, spelling, tone, and clarity.

Prompts and Inputs: Prompts used with ChatGPT included instructions such as: "Convert this paragraph into formal academic writing," "Summarise this explanation concisely," and "Write this section in LaTeX without bullet points." In all cases, inputs to ChatGPT were written and designed by the student.

Review and Validation: All AI-generated outputs were critically reviewed, edited, and cross-checked to ensure academic integrity. No content was accepted without validation.

Data Protection: No proprietary data from FIRN Energy (e.g., site names, geographic coordinates, or credentials) was submitted to AI platforms. Any sensitive data handling was performed offline and locally within the secure development environment.