Data-Driven Modelling of Solar PV Inverters

A Multi-Model Approach Combining Survival Analysis,
Digital Twins, Anomaly Detection, and Predictive Maintenance

CSCI E-599a – Data Science Capstone

Cade Wolfaardt, Payas Chatrath, Muhammad Ali, Trac Nguyen

Dr. Bruce Huang, Dr. Stephen Elston

Capstone Report in the Field of Data Science
for the Degree of Master of Liberal Arts in Extension Studies

Harvard University Extension School

December 11, 2025

GitHub Repository: https://github.com/CadeWolfaardt/solar-pv-inverter-modelling

Abstract

MN8 Energy operates a portfolio of utility scale solar plants where maintaining consistent energy generation is critical for meeting contractual obligations. This project investigates how data driven methods can support monitoring, anomaly detection, forecasting, and predictive maintenance of solar inverters using operational Supervisory Control And Data Acquisition (SCADA) data. The study focuses on five subproblems: Digital Twin development, anomaly detection, survival analysis, power forecasting, and predictive maintenance.

A Digital Twin of inverter active AC power was developed as a numerically driven model combining physics based principles with machine learning. The hybrid formulation achieved the best overall performance, demonstrating that inverter behaviour is largely linear but with meaningful nonlinear structure that is captured by boosted tree methods. The Digital Twin also provides standardized and reconstructed inputs for downstream models, reducing the impact of missingness and inconsistencies in the raw dataset.

The anomaly detection framework integrates multiple operational indicators to flag suspicious behaviour in real time. Survival analysis was explored using Kaplan Meier and Cox Proportional Hazards models, providing risk based insights despite the absence of explicit failure labels. The predictive maintenance analysis applied ensemble learning combined with density based clustering, achieving the strongest performance with XGBoost paired with DBSCAN and revealing that weak early warning patterns may exist in the data. The forecasting work implemented and compared classical statistical models, deep learning architectures, and literature based methods, with SARIMA emerging as the strongest performer under the available data conditions.

Across all tasks, data quality proved a major limiting factor. Missing values, inconsistent readings, unlabeled outages, and nighttime zeros constrained the use of supervised failure prediction methods and required substantial preprocessing. Nevertheless, the combined modelling framework demonstrates that meaningful operational insights can be extracted from imperfect real world data and that a hybrid Digital Twin can serve as a powerful foundation for future predictive maintenance systems.

The project delivers a full analytical pipeline and an accompanying graphical user interface that presents Digital Twin outputs, anomaly indicators, forecasts, and predictive maintenance signals in a unified dashboard. Together, these components illustrate how data science methodologies can support the identification of sub optimal inverter behavior and contribute toward more reliable, proactive solar plant operations.

Acknowledgements

Our project team would like to thank the Goldman Sachs MN8 team for their continuous technical support, the data they provided, and the time they spent meeting with us and answering our questions. We also express our gratitude to Dr. Bruce Huang, Director of the Harvard Extension School Master's Degree Programs in Information Technology, for leading this capstone project; to Dr. Stephen Elston of Harvard University for his technical insights and invaluable guidance; and to Dr. Leonardo Neves for his thoughtful questions and support. We thank James Tourkistas for his early contributions during the development of Chapters 1 and 2. Portions of this work benefited from the use of AI tools (e.g., ChatGPT) for brainstorming, phrasing support, and clarification of technical concepts. All analysis, modelling, interpretation, and final writing decisions were made by the authors for their respective contributions.

We are also grateful to the other MN8 teams whose contributions helped us shape and refine our ideas and solutions. Additionally, we thank Harvard Extension School for providing such an excellent program, of which this class serves as a meaningful culmination. Finally, we extend our appreciation to our employers for their financial support and for giving us the opportunity to complete this program.

"Education is not the filling of a pail, but the lighting of a fire." - William Butler Yeats

Contents

List of Figures

List of Tables

Acronyms

AC Alternating Current

AI Artificial Intelligence

ANN Artificial Neural Network

CNN Convolutional Neural Network

Cox PH Cox Proportional Hazards

DC Direct Current

DT Digital Twin

EDA Exploratory Data Analysis

EMA Exponential Moving Average

GUI Graphical User Interface

GW Gigawatts

KM Kaplan-Meier

kNN k-Nearest Neighbors

LSSPV Large-Scale Solar Photovoltaic

LSTM Long Short-Term Memory

MA Moving Average

MAE Mean Absolute Error

ML Machine Learning

MPP Maximum Power Point

MSE Mean Squared Error

MW Megawatts

NRMSE Normalised Root Mean Squared Error

O&M Operations and Maintenance

OLS Ordinary Least Squares

OT Operational Technologies

PdM Predictive Maintenance

PF Power Factor

POA Plane-of-Array Irradiance

PV Photovoltaic

RF Random Forest

RMSE Root Mean Squared Error

RNN Recurrent Neural Network

SCADA Supervisory Control And Data Acquisition

SEP Solar Energetic Particles

SHAP SHapley Additive exPlanations

SVM Support Vector Machine

SVR Support Vector Regression

WCSS Within-Cluster Sum of Squared Errors

XGBoost Extreme Gradient Boosting

Chapter I: Introduction

1.1 Problem and Setting

1.1.1 Setting

MN8 Energy, Inc. (MN8) is one of the largest solar energy production and storage companies in the United States (US) (MN8 Energy, Inc., 2023). The company has 3 core businesses: Solar Production, Solar Storage, and Solar powered electric vehicle charging stations. MN8 derives most of its revenue from selling energy, storage, and related services to customers to meet long term contract obligations. To meet these obligations, MN8 manages 875 solar projects producing 2.9 Gigawatts (GW) of energy across the U.S. Additionally, the company has 270 Megawatts (MW) of battery storage capacity for excess energy. The energy is eventually sold to a portfolio of 200 customers to aid in decarbonization efforts.

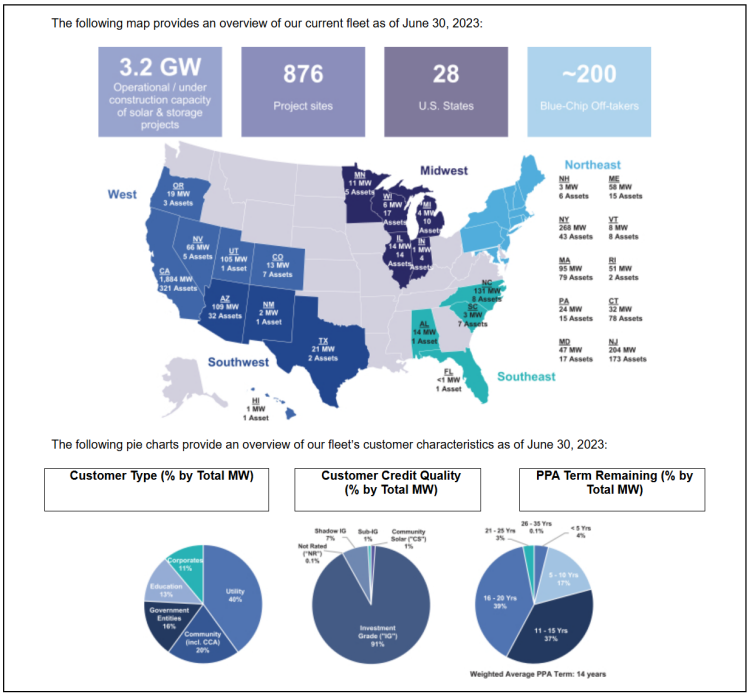

Figure 1: MN8 Map of Current Fleet (MN8 Inc., 2023)

Ultimately, this project addresses MN8's goal to "actively manage.. renewable energy assets to maximize revenue, minimize expenses and harness technological improvements... through proactive performance monitoring, customized preventative maintenance schedules and system design improvements" (MN8 Energy, Inc., 2023, p. 3).

1.1.2 Problem

Managing 100,000's of solar panels often in remote locations across 875 sites is extremely challenging. Solar energy production remains more expensive and availability is less predictable than energy from fossil fuels due to remote panel location and confounders like weather. To increase solar's value proposition, MN8 aims to develop methods that maximize energy output by proactively identifying required maintenance, reducing asset downtime, and balancing expenditures against lost revenue from decreased generation capacity.

In an attempt to identify and address inefficiencies in energy forecasting and panel maintenance, MN8 has identified two opportunities where the application of machine learning techniques may improve energy generation predictability and reduce asset downtime. The company hopes to develop solar plant compo-

nent Digital Twin (DT) to predict how much power a site should generate based on known inputs like weather. The company wants to use machine learning to proactively predict required maintenance. These applications can help maximize power generation and facilitate better long term contract management (R. Milan, personal communication, June 25, 2025)

1.1.3 Statement of Purpose

This project aims to utilize historical data and predictive algorithms to create a DT that constitutes a framework for simulating operational technologies, forecasting component failure, and predicting generated output; with the ultimate goal of reducing unplanned downtime, improving energy generation, and reducing the burden of meeting contractual energy agreements.

1.2 Subproblems

1. **Data Exploration and Processing**
   How can five years of high-frequency SCADA data be efficiently cleaned, preprocessed, and feature-engineered to produce a reliable analytical dataset suitable for downstream modelling?

2. **Predictive Maintenance Modelling**
   How can statistical and machine-learning techniques be used to identify early indicators of component degradation, emerging inefficiencies, or potential failure events?

3. **Survival Analysis of Component Health**
   How can survival models be applied to estimate failure likelihoods, quantify hazard rates, and identify operational or environmental factors that influence component longevity?

4. **Forecasting Inverter Output**
   How can time-series and machine-learning models be used to accurately forecast inverter active power under varying environmental and operational conditions?

5. **Anomaly Detection in Operational Behaviour**
   How can data-driven approaches identify anomalous patterns, non-standard operating regimes, and deviations from expected behaviour in near real time?

6. **DT Framework Construction**
   How can a DT framework be designed and implemented to replicate the physical and operational characteristics of the solar assets, enabling monitoring, forecasting, and anomaly detection in real time?

1.3 Theoretical and/or Conceptual Framework

1.3.1 Theoretical Framework

Digital Twin: A digital version of a physical system that reflects how it behaves and performs under live data from sensor inputs; this can be achieved through the use of classical methods or more recent machine learning.

Predictive Maintenance: A data-driven approach that uses analytics and machine learning to predict equipment failure beforehand; with the data input from the physical environment and DT, statistical or mathematical models are used to estimate the time/need of maintenance.

1.3.2 Conceptual Framework

Although solar farms require relatively less maintenance, components still fail, necessitating ongoing maintenance, which can result in reduced energy production and negatively impact the economic return from the solar farm. Thus, it is valuable to be able to identify when maintenance is required and, even better, to predict future maintenance needs.

Producing a DT that allows estimation of component performance (e.g., inverter, solar panel, combiner, etc.) allows for verification of component behavior, insight into system health, and anomaly detection

(comparing expected performance vs. actual). PdM enables scheduling planned intervention or maintenance before device failure. These applications enable MN8 to reduce downtime, thereby increasing the total energy output of the solar farm.

This research into DT-enabled monitoring and PdM for the MN8 solar farm involves several key stakeholders. MN8, as the plant owner, benefits from increased return on investment (ROI) through improved asset utilization and decreased downtime. Investors and shareholders benefit from improved company performance and more substantial annual returns, which can potentially lead to increased stock value. The research community and renewable energy industry can also benefit, as the study contributes to ongoing innovation in intelligent energy systems. Ultimately, society at large stands to benefit from improved reliability and efficiency in renewable energy systems, which helps meet environmental goals and reduce fossil fuel dependence.

1.4 A Priori Hypotheses

1. Machine learning techniques will be better at predicting expected energy output and failure than naive methods like simple averages.
2. Machine learning techniques can help us account for confounding variable impacts to improve our predictions.
3. Machine Learning (ML) techniques will allow us to account for time dependence of features.
4. Required maintenance events can be discerned from energy output and sensor data.

1.5 Variables and Key Concepts

- **Independent Variables (sensors)**
  - Date
  - Time
  - Morning Start Time (Impacts Energy Production/Delay May Indicate Failure)
  - GPS Location
  - Device ID / Brand
  - Weather
  - Shadow Times
  - Amperage
  - Temperature
  - Humidity
  - Orientation of Panel (Panel Angle Can Impact Energy Produced)
  - Inverters (Direct Current (DC) to Alternating Current (AC) conversion)
  - Position of Inverter (Where Its Located)
  - Panel Useful Life (Solar Panel Output Decreases Over 3 Years)
  - Other Solar Panel Sensors
  - Component characteristics (obtained from data sheets)
  - Component (inverter, combiner) input and output power
- **Potential Dependent Variables (Outcome)**
  - Expected Energy Output (watts)
  - Device Failure / Sub Optimal Performance (<65% Production)
  - Life Cycle of Devices / Parts Longevity

1.6 Assumptions, Limitations, Delimitations

1.6.1 Assumptions

1. The data accurately and sufficiently captures the real world system.
2. Failure events are discernible from the data.
3. The provided features are significant enough to predict energy generation.
4. Collinearity between features like temperature, weather, and shade can be reasonably controlled.
5. Economics of maintenance depends on continued energy demand.

1.6.2 Limitations

1. The accuracy of predictions is dependent on the quality of the data.

2. The prediction horizon is limited by knowledge of future variables and the level of variance explained by accessible data.
3. Equipment failures are low frequency events increasing the difficulty of prediction.
4. The impacts of weights and variables on model predictions will not necessarily translate to causality. The model must effectively represent reality.
5. Some older panels may not have all sensors.

1.6.3 Delimitations

1. We only model one project, despite the existence of 875 solar projects.
2. We focus on operational technologies and not the broader environment.

1.7 Importance of the Study

Solar energy offers a cleaner source of energy than fossil fuels. Increasing reliance on solar energy remains a crucial component in decreasing our environmental impact and CO2 reduction plans. The best way to improve solar adoption is to improve the reliability and commercial viability. Research into machine learning applications that help manage maintenance / decrease asset downtime, increase power / revenue generation, and improve energy production forecasting for easier contracting are critical to this goal (Milan, 2025).

Chapter II: Literature Review

2.1 Context

Fossil fuels are not a sustainable option for long-term energy production; consequently, renewable energy sources have become increasingly important for electricity generation (Hernández-Callejo et al., 2019). Solar energy is well-suited to meet this increasing demand, as it is the most abundant renewable energy source, and its utilization has no harmful impact on ecosystems (Kannan & Vakeesan, 2016). This has contributed to the rise in the adoption of Photovoltaic (PV) systems, which harness solar energy (Hernández-Callejo et al., 2019). Projections support this growth, indicating that solar energy, as well as wind, is projected to lead growth in U.S. power generation over the years 2024 and 2025 (Antonio, 2024).

As PV systems become more widespread, ensuring their consistent and reliable performance is critical; correspondingly, the importance of Operations and Maintenance (O&M) in PV systems has increased (Keisang et al., 2021). PV systems inherently require relatively low maintenance; however, research indicates that utilizing O&M strategies can still provide a clear benefit, increasing yield and improving overall profitability (Orosz et al., 2024; Keisang et al., 2021). In industrial and commercial settings, it is essential to ensure "reliability, efficiency in supplying power, safe system operation over the years, and return on investments," further reflecting the need for O&M (Keisang et al., 2021, p. 3).

To understand what is being maintained, it is important to examine how solar power is generated at the system level. Solar plants generate consumable electricity through several important components. At a high level, sunlight shines on solar panels connected by wiring that converts solar irradiance into DC electricity. The DC power is then converted to AC by an inverter for distribution (Peters & Madlener, 2017). PV system "components include the racking structure, DC/DC converter, inverter, switches, DC and AC protection devices, wiring, and others" (Peters & Madlener, 2017, p. 9). In large-scale setups, such as MN8, panels can cover hundreds of acres, include numerous inverters, and have multiple panels connected to each inverter (Peters & Madlener, 2017; Sarkar, 2024). Given the size and remoteness of plants, operators often struggle to manage maintenance properly (Sarkar, 2024). For example, Pereira et al. (2023) report that inverters are common causes of service calls, fail most often (4–6% of the time), and are costly to maintain.

To implement effective maintenance strategies, particularly predictive ones, it is essential to understand the components that are at risk of failure. There are three key Operational Technologies (OT) utilized in PV systems (Rajesh & Carolin Mabel, 2015): the solar panel (or PV panel), which is a collection of PV cells that convert solar radiation into electricity (Sodhi et al., 2022); the power converter, which allows for the extraction of useful generated electricity from the solar panels; and the tracking controller, which is used to harness maximum power from PV panels while tracking PV parameters and producing a control signal to the power converters (Rajesh & Carolin Mabel, 2015).

This paper focuses on the maintenance aspect of O&M, which includes several key strategy types: corrective/reactive, preventative, predictive, and condition-based maintenance (Keisang et al., 2021; Orosz et al., 2024). Corrective/reactive maintenance is a form of unscheduled intervention that occurs only when equipment fails, breaks down, or is unable to continue its primary function; in the long term, this can lead to increased downtime and higher costs (Dhillon, 2017; Orosz et al., 2024). Preventative maintenance focuses on the precautionary actions taken to reduce the probability of failure or degradation, involving regularly scheduled maintenance. Although this approach can be more cost-effective than reactive maintenance, it requires a greater initial investment of time and resources (Dhillon, 2002; Orosz et al., 2024).

Predictive Maintenance (PdM) involves forecasting failure events to enable mitigation measures to be taken (Keisang et al., 2021), relying on data analytics and monitoring technologies to forecast when maintenance is needed. This approach can be highly cost-effective; however, it typically demands substantial investment in "data collection and analysis tools" (Orosz et al., 2024, p.3). Condition-based maintenance involves utilizing real-time data streams to monitor performance and system condition, with maintenance scheduled in response to these insights. Similar to predictive maintenance, this strategy can be highly cost-effective; however, it requires a substantial investment in monitoring tools and sensors (Orosz et al., 2024).

Recent technological advances have shifted the focus toward more proactive approaches, particularly predictive maintenance (Murtaza et al., 2024). Prior to the implementation of sensors and machine learning strategies like Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) neural networks, and anomaly detection assistance, Sarkar (2024) notes his maintenance team struggled with fault detection taking 72 hours on average instead of the industry standard of 4 hours (or less). Unplanned maintenance was common, and the most remote areas were ignored.

2.2 Taxonomies of Faults in Photovoltaic Systems

To support the application of predictive maintenance and other strategies, researchers have developed taxonomies to classify faults in PV systems. Firth et al. (2010) distinguish three classes of PV system performance: ideal performance, relating to system specifications; normal performance, relating to average performance over time; and actual performance, relating to deviations from average performance. From these, the authors identify six faults tending to cause actual performance degradation from normal: "component failure; system isolation [for less than a day or more than a day]; inverter shutdown; shading; inverter Maximum Power Point (MPP) tracking failure; and other faults" (Firth et al., 2010, p. 4). The authors further categorize these faults into four classes caused by "shading; non-zero efficiency, non-shading[;] ... sustained zero efficiency faults (when the PV system stopped generating for long periods), ... [and] brief low or zero efficiency faults, with durations of around 10 minutes or less" (Firth et al., 2010, p. 4, 16). To distinguish between transitory events, such as cloud cover, and system faults, the authors advocate for evaluating the watts generated per unshaded square meter of irradiation (Firth et al., 2010, pp. 5-7). Although this conversion normalizes energy data, it may fail to identify non-mechanical events.

While Firth et al. focus on system-level performance deviations, other researchers have categorized faults based on physical failure modes within specific components. Orosz et al. (2024) state that PV panel failures can be classified as "optical degradation, electrical inadequacy, and unclassified faults" (p. 4). Furthermore, recent literature suggests that PV system reliability is inversely proportional to system size, with smaller systems being more reliable for longer periods. Typical causes of PV panel failures include overgrown vegetation, which can lead to shading or component compromise (e.g., wiring or inverter); cracks in panels caused by wildlife activity or weather events like hail, which can result in up to 30% performance loss; occlusion or shade from dirt, dust, and snow buildup (leading to annual energy production losses of up to 35%), and bird droppings; and panel aging or compromise such as silver paste (a component used for grid lines on PV cells) discoloration, which can decrease panel performance by up to 20%, delamination (where adhesive bonds allow moisture ingress), and discoloration (browned or discolored areas caused by poor-quality polymers), all of which reduce incident light.

Beyond PV panels, faults in power conversion components, such as inverters, have also been systematically classified. Golnas (2012) reports that inverters account for the majority of subsystem failures, contributing up to 28% of energy losses. The root cause is often the failure of parts or materials, excluding issues captured by control software, which frequently logs unclassified faults. Identifying the specific cause requires specialized component analysis. However, four key types of parts/materials failures are defined: "1) out-of-specification operating conditions, 2) substandard manufacturing or specification, 3) upstream construction-related error, or 4) stochastic failure" (Golnas, 2012. p. 3). Golnas also suggests that inverter failures follow the "bathtub curve" typical of electrical devices: an initial phase of high infant mortality failures (often related to manufacturing or installation issues), followed by a long period of low, stochastic failures typically due to unpredictable events and operational stresses; and finally, an increasing failure rate due to wear-out mechanisms.

While inverter failures remain the dominant failure point in PV systems (Golnas, 2012), their impact extends upstream, disrupting PV modules and limiting overall energy production. The majority of the documented and studied faults are concentrated at the inverter level (Dhoke et al., 2020; Golnas, 2012), providing limited visibility into failures across other OTs. Golnas (2012) notes that other areas of general failures include the AC subsystem (accounting for up to 20% of energy losses), external factors (beyond the scope of the PV system and unpredictable) contributing up to 20%, and additional areas such as support structure, DC subsystem, planned outages, and miscellaneous categories collectively accounting for up to 23% of energy losses. Another study, classifying 63 PV plant failures, found that failure types were distributed as follows: monitoring systems (30.6%), communication systems (26.3%), inverters (12.5%), grid (11.6%), and PV generators (8.5%). (Gallardo-Saavedra et al., 2019, p. 830). The lack of

monitoring at a suitable resolution in these different areas limits the ability to report faults.

2.3 Key Variables Influencing Solar Energy Systems Performance and Fault Detection

Given the mechanisms involved in producing solar energy, Al-Humairi et al. (2025) identified "temperature, humidity, solar irradiation, panel orientation, spectral losses, solar spectrum variations, ... manufacturing variability, ... [panel soiling (i.e., dust), and]... shading" (p. 3) as key factors impacting solar energy generation. Moreover, Springer et al. (2022) confirm that features like these are known to be system stressors leading to maintenance issues. In contrast, Zhao et al. (2012) achieved high fault detection accuracy on a limited sample from co-located solar panels by applying a tree model relying on Current ("I") and Voltage ("V") as key indicators of fault. In these fault detection tests, the authors concluded that the co-location of panels at similar tilts could enable I and V to act as proxies for features such as irradiance and temperature. Researchers have further identified other key variables, such as cloud coverage and movement, sensor reliability, PV power output (Orosz et al., 2024), and panel age (Hernández-Callejo et al., 2019), as critical features influencing PV system performance.

One of the articles by Rehman et al. (2025) focused on providing a global perspective on PV System Performance. It provides a comprehensive synthesis of numerous studies examining the impact of key environmental factors, including dust, tilt angle, temperature, and humidity, on PV performance worldwide. An interesting quantification provided by the article mentioned losses due to dust were up to ~40%, temperature was up to ~0.05% per degree Celsius increase, humidity was up to ~34.2% and optimal tilt angles. This article stands out because of its global scope and data-rich comparisons. However, it lacks original data analysis and omits a deeper exploration of factors such as shading, electrical degradation, and economic trade-offs. Moreover, regions such as America receive less detailed coverage compared to other regions. The operational goal of the review was to compile and evaluate global research on PV system degradation and improvements. This article mainly relied on previous studies and field studies. The findings presented in the article were analyzed using performance metrics such as energy yield losses, degradation rates, and efficiency restoration percentages, but no modeling frameworks were applied. This article can be helpful for projects aiming to optimize PV performance under environmental stress, providing the integration of mitigation strategies and highlighting the areas for further research (Artificial Intelligence (AI)-driven prediction and cost-benefit modeling).

2.4 Handling Imbalanced Data in Energy Fault Detection

Fault detection in solar energy systems is critical for achieving high prediction accuracy. However, a common challenge in such systems is the highly imbalanced nature of the data: normal operational records significantly outnumber fault data. This imbalance can negatively affect the performance of machine learning models, leading to inaccurate predictions. Therefore, addressing the class imbalance problem is essential when constructing datasets for predictive modeling.

The study by Wan et al. (2021) explores the issue of data imbalance in the context of forecasting solar flares and proposes three main strategies to mitigate it: (1) resampling techniques, (2) adjusting decision boundaries, and (3) assigning different weights to training samples.

2.4.1 Type I Resampling Techniques

This method involves modifying the dataset by either under-sampling the majority class or over-sampling the minority class to achieve a more balanced distribution during training. Under-sampling removes some data from the majority class while over-sampling duplicates data from the minority class. These approaches help reduce bias toward the dominant class and improve the model's ability to detect minority class events (e.g., solar flares). The authors note that the performance of this method can be further improved by applying an intelligent distributing method on data.

2.4.2 Type II Adjusting the Decision Boundary

This approach modifies the decision-making process by assigning different weights to the prediction errors of each class. Errors related to the minority class are given higher value, effectively biasing the model toward better detection of rare events. The paper suggests a simple formula for calculating these weights:

the total number of samples divided by the number of samples in each class. This technique makes it easier for the model to recognize minority class instances within a largely imbalanced dataset.

2.4.3 Type III Weighted Training Samples

Unlike Type II, which adjusts prediction error weights, Type III assigns different weights directly to the training samples. Samples from the minority class are given higher weights than those from the majority class. This increases their influence during training and helps the model learn patterns associated with rare events more effectively.

2.4.4 Data and Experimental Result

The dataset used in the study consists of the cadence of solar magnetogram data spanning three years from 1996, covering 185 solar regions. The imbalance is clearly shown in the sample ratio of 1,472 positive cases to 8,528 negative cases. The results indicate that among the three approaches, Type I (over-under sampling method) yields the best overall performance.

2.4.5 Application to the MN8 Project

Similar to solar flare forecasting, the MN8 Energy project is also expected to face significant data imbalance, as instances of solar unit or cell failure are likely to be rare. Most of the dataset will consist of normal, error-free operational data. Given this, the Type I (over-under sampling) approach to balance the dataset for each training epoch appears to be the most applicable strategy for our model training. This technique will be adopted in the MN8 Project to ensure more balanced learning and improved fault detection accuracy.

2.5 Feature Engineering in Time-Series Energy Data

Solar power systems generate output in the form of time series data, sequences of measurements collected at fixed time intervals. One of the main challenges in forecasting solar power is the high volatility and nonlinearity of the data, primarily driven by recurring seasonal trends. Seasonal fluctuations can significantly impact the accuracy of predictions, often leading to increased errors. To address these challenges, time series decomposition methods are commonly employed.

In a recent study, Gurcan et al. (2023) propose a novel approach aimed at improving forecasting accuracy independent of the machine learning algorithm used. Their method decomposes solar power generation time series data by incorporating irradiance and seasonal features as exogenous inputs. Specifically, the authors predict solar output two days ahead using a Moving Average (MA) filter for decomposition and Ordinary Least Squares (OLS) regression to estimate trends.

The study uses data collected from a solar plant in Turkey and explores two data preparation strategies: combined and separate. In the combined approach, data from multiple hours is aggregated (e.g., via averaging), while in the separate approach, individual data points are retained for different time segments. Two modeling strategies are applied to enrich the feature set: Feature-Based Modeling and Trend-Based Modeling.

In Feature-Based Modeling, forecasting accuracy is enhanced by extracting summary features that characterize the time series, such as the mean and trend. The mean represents the average over a selected time window, while the trend is derived using OLS linear regression. In Trend-Based Modeling, the time series is expressed as a combination of linear and non-linear components, aiming to minimize the effects of directional trends. Forecasting stability is achieved through machine learning algorithms such as Artificial Neural Network (ANN), Random Forest (RF), Support Vector Regression (SVR), and k-Nearest Neighbors (kNN), while trend forecasting is supported by OLS-based decomposition.

The proposed method shows significant improvements in prediction accuracy, reducing error metrics by 5% to 39% compared to baseline models. The study highlights ANN and SVR as particularly effective for solar power forecasting.

The promising results achieved with real-world data from Turkey suggest that this approach is also well-suited for fault prediction. For the MN8 Energy project, which aims to forecast system failure up to

seven days in advance, adapting this feature engineering and decomposition strategy could be a valuable next step.

2.6 Digital Twins for Solar Energy Systems: Architecture and Failure Simulation Approaches

There is no universally standardized definition of a DT, as its interpretation varies across the many domains in which it is applied (Abdelrahman et al., 2025). However, Yao et al. (2023) identify three core functions that characterize DTs: "(1) data fusion of various features of physical objects and high-fidelity real-time mapping of physical objects; (2) coexistence and coevolution throughout the lifecycle of physical objects; and (3) description, optimization, and control of physical objects" (p. 1). While DTs have also been defined from model, function, and linkage-based perspectives, this review focuses on data-driven DTs, as these align most directly with the goals of this project: using sensor data and analytics to simulate and manage PV systems. From a data-focused perspective, DT has been defined as "an approach that uses data-driven analytical algorithms and other physical models to simulate the operational state of an entity." (Lee et al., 2013, as cited in Yao et al., 2023, p. 3)

Yao et al. (2023) also highlight the broad functional relevance of DTs across several applications, including "simulation, monitoring, evaluation, prediction, optimization, [and] control" (p. 1). These functions are beneficial for anticipating and diagnosing early signs of underperformance in operational systems. Massel et al. (2021) utilize DT for the planning and design of a solar power plant, employing ontological engineering techniques. They model a solar power plant using a mathematical approach initially proposed by D.N. Karamov. and Naumov I.V., as cited by the authors, which calculates the generated power, efficiency, temperature, and output current and voltage of a PV panel using solar radiation, air temperature, wind speed, and panel characteristics. They utilized PostgreSQL DBMS as a database to organize and store the vast volume of data (e.g., characteristics of weather conditions, characteristics of equipment, etc.). The authors define the architecture of the implemented DT to have three blocks: Digital shadow, which is the backend pertaining to the database and interface for querying and storing data; the digital model, including the application interface, mathematical model, and system for collecting operational information; and control system, which is used to action a control command on a real object, which the DT prototypes. They highlight the limitations, stating the need for refinement of the DT to enable real object data interaction.

Pimenta et al. (2020) developed a hybrid DT for an onshore wind turbine, combining simulated structural models (using third-party technologies such as FAST by NREL and ANSYS Fluent) with SCADA and sensor data for real-time fatigue tracking and operational analysis. Their approach demonstrates how data-driven and physics-based components can collaborate to validate system behavior and improve fault detection. They note that further work can benefit from the use of higher-accuracy data with more validation.

Jain et al. (2020) put forward a DT for rooftop and building integrated PV systems, specifically modeling the PV source and source-level power converter, used for generating an estimated output that is compared and evaluated against the measured output for fault diagnosis. They utilize a mathematical model-based approach for their DT, which computes the measurable characteristic outputs of the modeled system. The authors posit that previous works were heavily specific and lacked generality, further stating that their developed DT possesses generality and can be applied to other energy systems.

Arafet and Berlanga (2021) also discuss a DT for fault detection in PV systems. Instead of a mathematical model, they opt to use a deep learning (DL) approach, specifically an autoencoder that combines convolutional layers, LSTM, and TimeDistributed layers to reconstruct the inverter errors and utilize the reconstruction error as a proxy for system health. The model was trained on a dataset obtained from an inverter manufacturing firm, which contained time-series data of 288 daily measurements for 259 metrics. The authors utilized dimensionality reduction techniques and data analysis methods to reduce the number of working features. The authors identify an explicit limitation of their work as the lack of meteorological variables that aid in understanding the temporal relationships between signals.

Yalçin et al. (2023) present a DT design that focuses on O&M through the integration of ML techniques. Trained models are used to predict system behavior given real-time data as input. The data used to train their models is synthesized using a simulated power plant in MATLAB/Simulink. The authors

propose defining the DT as a set of subsystem models, each representing a specific component of the solar plant. DNN performed best for PV panels, RF for DC-DC converters, and CatBoost for grid connection when considering both accuracy and model complexity. Their final system reported a global accuracy of 98.3%. However, the authors acknowledge limitations due to the synthetic nature of the data and the need for further validation with real-world systems. While various DT architectures have been proposed for PV systems (including mathematical, hybrid, and ML-driven approaches), there remains a clear gap in the deployment of real-world, adaptive DTs that operate on live sensor data and support continuous retraining.

2.7 Forecasting Techniques for Predictive Maintenance in Solar Applications

It is important for any solar plan to have a predictive maintenance capability. Having an accurate prediction of a system will minimize downtime, reduce costs, and optimize the performance of units.

2.7.1 Survival Analysis

Survival analysis is a suite of longitudinal statistical methods designed to study time-to-event data, where the outcome is the time until a particular event occurs. It is particularly useful for understanding how events unfold over time. In the study of the paper by Jackson et al. (2024), the authors apply various survival analysis algorithms to predict the occurrence of Solar Energetic Particles (SEP), high-energy particles ejected from the Sun during explosive solar events. The authors explore a comprehensive set of statistical models encompassing nonparametric, semiparametric, and parametric approaches. Specifically, they utilize the Kaplan-Meier (KM) estimator, the Cox Proportional Hazards (Cox PH) model, and several parametric models, including Exponential, Weibull, Lognormal, and Log-Logistic distributions. The KM estimator is used to estimate the probability of SEP occurrence over time, while the Cox PH model evaluates the impact of covariates on the timing of SEP events. Together, these methods capture both overall survival patterns and the influence of key variables. Unlike the KM and Cox models, the parametric approaches assume that the survival times follow a specific statistical distribution. The paper indicates the process of collecting data. The raw data collected from satellites undergoes several stages of transmission and processing before being cataloged in a SEP database. Using this data, the authors demonstrate how survival analysis can predict the time from a solar flare to the actual occurrence of a SEP event, measured in seconds.

These same techniques can be adapted for predicting failures in PV systems. By shifting the focus from SEP events to system degradation or failure, the same survival models can be applied to forecast when a failure might occur to a solar unit, from the present moment to when it will fail. This approach is highly relevant to the MN8 Energy project; this is one of a few methods to use for predictive maintenance.

Beyond survival analysis, three other popular methods for predicting maintenance are solar energy creation anomaly detection, binary decision models, and Monte Carlo / Markov Chain simulations.

2.7.2 Anomaly Detection

In an attempt to normalize energy data for outlier detection and find panels operating outside of their expected range, we can use non-parametric regression prediction to subtract actual output from the expected output based on available features. Evaluating residual energy may improve our capacity to identify malfunctioning panels/inverters based on output without the influence of confounders. Using "Humidity, Ambient temperature, Wind speed, Visibility, Cloud ceiling and Pressure" (p. 1) as predictors and regression-based algorithms like "Gradient Boosting Regressor (GB), XGB Regressor (XGBoost), kNN, LGBM Regressor (LightGBM), and CatBoost Regressor (CatBoost)," (p. 1) after a 70/30 split and 10 fold cross-validation, Nguyen et al. (2025) achieved a respectable .546 test set R-squared using CatBoost to predict power output (pp. 12-14). Prediction errors were within +/- 10 W (p. 15). To tune model parameters, the authors used Grid search cv and Bayesian optimization (.1 learning rate) (p. 13). Finally, CatBoost SHapley Additive exPlanations (SHAP) values confirmed positive impacts from AmbientTemp and negative impacts of humidity (p. 16).

Although dew from humidity is indeed known to adversely impact solar panel performance, the SHAP findings about ambient temperature are somewhat incomplete. The authors originally explained as ambient temperature increases, panel efficiency decreases. Additionally, high ambient temperature is

known to "induce thermal stress within the materials of solar panels, potentially leading to material degradation and reduced performance over time" (p. 6). CatBoost did not appear to identify this relationship. In our evaluation, we must weigh the importance of model results following known system properties. The strategies employed must also account for differences between immediate impacts and long-term impacts. In contrast, given Zhao et al. (2012)'s aforementioned assertion that co-location proxies some confounders like temperature, it is also worth exploring whether simpler models are useful that compare energy production for panels/inverters close in time and proximity to identify outliers.

Another article reviewed by us focusing on anomaly detection for predictive maintenance in a Large-Scale Solar Photovoltaic (LSSPV) plant used operational data from a centralized PV monitoring system in Malaysia. The researchers used K-Means clustering to group electrical current data and evaluated the clusters using the Within-Cluster Sum of Squared Errors (WCSS). They then applied LST and ANN models for anomaly prediction, assessing their accuracy using MSE, MAE, and RMSE, which resulted in showing LSTM to be slightly more effective than the ANN model. The approach used by the team was well structured; limitations include insufficient historical data and the complexity of PV systems, which lead to errors in feature extraction. This study aligns closely with our project goals, focused on predictive maintenance using time series data and machine learning. Some improvements could include testing alternative clustering algorithms apart from K-Means and comparing more time series models against LSTM to enhance the accuracy of predicting.

Other authors like Ahmed et al. (2024) trained Support Vector Machine (SVM), RF, Multivariate Regression Splines, and Gradient Boosting Machines to create hybrid models of stationary hourly ambient temperature, beam irradiance, cell temperature, diffuse irradiance, and wind speed to predict energy generation both 3 and 7 days in advance (pp. 5011-5012). Key features were beam irradiance, diffuse irradiance, and cell temperature (p. 5016).

Finally, Liu & Sun (2019) discuss the use of other machine learning techniques to improve the accuracy of solar power prediction. The authors talk about using PCA for dimension reduction, combined with k-means clustering and RF algorithm optimized by the differential evolution grey wolf optimizer for improved predictive modeling performance. This combination yielded the best results using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) reported in 10^{-2} as benchmarks.

Moreover, predictions were made in three distinct regions, looking out one, two, and three hours ahead. Using the hybrid model of PCA + K-means + HGWO (differential evolution grey wolf optimizer), the authors achieved an average error (MAE) between .04 and .06. The error unit is unclear, but the paper estimates mean PV output between .22 and .25. These hybrid models outperformed baseline models like ANN, decision trees, and Gaussian regression.

The paper, while reporting strong numerical results, lacks important methodological details. The data was split into test and train, but the proportion of the split is not mentioned. There is also no mention of cross-validation. The evaluation of the model is based on a simple test train split.

Different weather scenarios, such as stable and unstable irradiance conditions, have been classified using clustering. Then, different RF models have been trained for each scenario class. Importance scores from the RF models were used to improve efficiency and reduce redundancy.

This approach out performed SVMs and ANNs. Using RF is a good choice in this scenario as it handles nonlinearity and feature interactions well. The use of real world hourly data helps with model relevance.

The use of PCA to solve various problems comes at the cost of interpretability and explainability because once the data has been converted into its principal components via PCA, a direct link back to the original data has been lost. Since the data was collected hourly, we believe that the data missed events that would help explain reading (or changes therein). Despite these shortcomings, the authors offer additional modeling approaches that can be applied to predict expected energy output and identify anomalies.

2.7.3 Binary Models

Some practitioners report success using cross-validated binary prediction models. Qureshi et al. (2024) report 91%+ accuracy and, more importantly, 90-95% true positive rate using logistic regression, decision trees, and RFs to predict equipment failure (p. 37). Additionally, authors advocate for linear regression

and Gradient Boosted Machine to model equipment lifespans (p. 34). Regarding the shortcomings of the paper, although a 90-95% error rate sounds good, we must carefully discuss with stakeholders the costs associated with false positives and hazards associated with false negatives. These can add up quickly. Another key aspect of modeling mentioned but not fully discussed by the authors was data preprocessing and feature engineering (p. 35). As previously discussed, to create the best binary prediction model possible, we should carefully evaluate any known relationships of predictors to future maintenance based on existing literature and ensure we create variables for them. For example, Zhang et al. (2012) highlight the age of the system, core temperature's connection to capacitor failure, voltage and temperature's connection to switch failure, and temperature's relationship to inverter failure (pp. 826 - 827).

2.7.4 Markov Chains / Monte Carlo Simulations

Authors like Zhang et al. (2012) and Singh et al. (2022) attempt to assess PV reliability by estimating the "net rate of failure, mean time between failure (MTBF), and mean time to failure (MTTF)" (Singh et al., 2022, p. 1867) using reliability block diagrams, exponential distributions, and Markov models. This modeling strategy can incorporate the inherent probabilistic nature of failures. At its core, this method requires the estimation of exponential distribution parameters (i.e., rates over a period of time) for component failure and component repair. Based on these parameters, we can then estimate breakage and downtime. More complex adaptations incorporate conditional probabilities relating to historical data like weather. Moreover, similar to reinforcement learning, some authors have illustrated the possibility of incorporating rewards like revenue, capacity, or costs by mapping values to each state (pp. 1867-1869). Although time is limited, the benefit of examining these methods is they will require careful evaluation of key failure influencers. The downside is models can quickly get complex. For example, 2^n states are required to get estimates for n components.

2.8 Validation and Uncertainty in Predictive Maintenance Models

For model validation, it is standard practice to split available data into a train and test set, perform k-fold cross-validation (i.e., repeated train/test splits) to tune parameters using grid search or other methods and evaluate training model predictions against a test set using a chosen metric (average error /the distribution of test residuals/accuracy/false positive rate) (Nguyen et al., 2025, pp. 12-16; Qureshi et al., 2024, p. 34). The whole process can be repeated with cross-validation to estimate the error range. Beyond validation during modeling, although not fully discussed by authors, post-implementation validation is also suggested using data from another solar farm, post-implementation randomized experiments, and monitoring actual post-implementation energy output improvement over time.

2.9 Conclusion

Although earlier work has made significant strides in fault classification, predictive maintenance modeling, and DT architectures for PV systems, discernible gaps remain. The reviewed literature often relied on static datasets, failed to consistently communicate data aggregation strategies, used different pre-processing and variable sets, and identified different modeling strategies as superior. Imbalanced datasets and insufficient feature abstraction may pose challenges to the accuracy of anomaly and fault detection. Additionally, few DTs operate on live sensor data or accommodate automated retraining. However, despite these challenges, we remain hopeful available techniques will add some value. Papers achieved relatively low fault detection error rates; since PV faults are physical processes, the literature identified clear mechanical precursors, the large number of sensors at MN8 PV farms should improve data availability, a range of machine learning/probabilistic methods are available for our task, and MN8's large portfolio should offer opportunities to validate model performance.

To further address gaps, time permitting, this project may explore a modular architecture that implements real-time anomaly detection (enabled by a DT) separately from longer-term predictive maintenance models. This separation could enhance adaptability, simplify evaluation, and, importantly, aligns with best practices identified in prior works, instilling confidence in our proposed approach.

Chapter III: Methodolgy

3.1 Introduction and Purpose

Solar energy is an environmentally friendly, renewable alternative to fossil fuels, and MN8 alone produces an astounding 2.9 GW. However, stable energy production is continually challenged by factors such as weather variability and component failures.

This project leverages historical SCADA data and predictive algorithms to build a DT that reflects the expected operational output of inverters across the PV farm which supports anomaly detection and situating current asset performance. We further implement several modelling streams forecasting inverter output, performing survival analysis, and developing predictive-maintenance frameworks. Across all modelling components in this study, we focus specifically on inverter-level behavior and use active AC power as the primary output variable of interest, as it is the clearest indicator of operational performance and is consistently available across all devices. Together, these modelling components aim to reduce unplanned downtime, improve energy generation, and reduce the burden of meeting contractual energy agreements.

3.2 Research Questions

The main research question addressed by this project is:

> Can Data Science methodologies be used to identify sub-optimal solar inverter energy generation and future maintenance events?

Predicting future maintenance is important because it reduces system downtime and helps stabilize energy output to meet customer and grid commitments.

3.3 Research Design

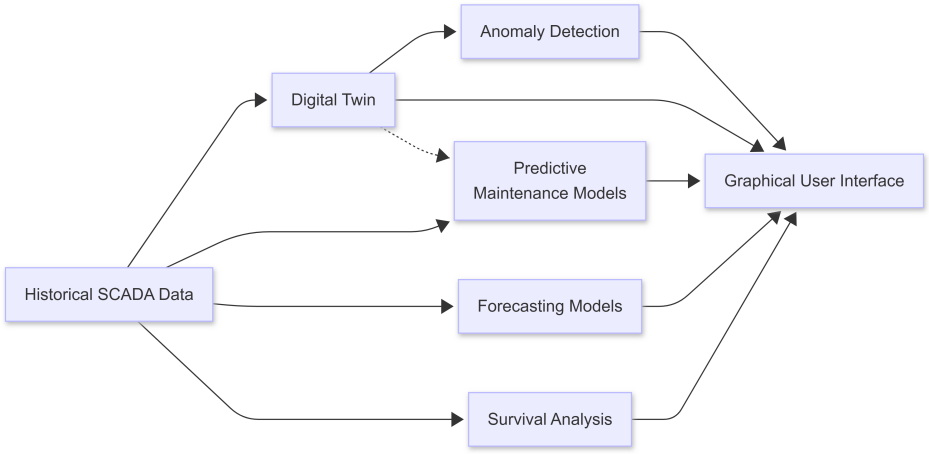

Figure 2: Component Integration Overview

We adopt a modular, data-driven framework that supports the modelling components introduced in the previous section. The study is quantitative in nature, relying on historical SCADA data to observe inverter behavior through measurable inputs and outputs. The design is observational and data-driven, with each modelling stream addressing a specific operational objective.

The DT ingests sensor data to estimate the inverter's expected active AC power at the current time step, providing a benchmark against which anomaly detection can compare real measurements to identify abnormal behavior. Forecasting models assist in projecting future energy output and bounding economic expectations. The predictive-maintenance module performs inference on degradation signals to estimate

when maintenance may be required within an appropriate time frame, and survival-analysis models inform on component lifetime and long-term reliability.

Outputs from these modelling streams are aggregated into a Graphical User Interface (GUI), providing a single point of access for visualisation, monitoring, and operational decision-making.

3.4 Data Collection and Storage

Our sponsor, MN8, collects data using a SCADA system across multiple PV plants. At the PV plant level, each project has several PV panels that feed information to an inverter, then several inverters are connected to a logger that uploads the collected data to the SCADA system. This happens once every 5 minutes.

High-frequency data collection results in rich, granular data sets that capture minute changes in weather conditions and the environment, such as changes in irradiance and cloud cover. In turn capturing anomalies in environmental conditions; something that prior models failed to do so (due to the longer intervals between data points). As a result this high resolution dataset allows us to train and develop more accurate models.

In PV systems, SCADA enables real time monitoring of energy generation, equipment performance, and weather conditions via a central interface. This helps with the collection of coherent data from across different geographical regions, which are subject to different environmental conditions.

The utilized data is confidential as it provides insight into MN8's performance, capacity, and infrastructure. Therefore, we follow these guiding principles when handling data:

- Ethical Use of Data: Data is used solely for the intended academic purposes, in line with the sponsor's goal of predicting failure.
- Anonymization: Any labels or metadata that identify the PV plant, its location, or other sensitive attributes are removed.
- Secure Storage and Encryption: Data is stored in encrypted, access-controlled environments, ensuring confidentiality and preventing unauthorized access.
- Access Limited to Our Team: Data access is restricted to the four members of our team only.

3.5 Exploratory Data Analysis

Before modelling the subproblems, we conduct exploratory data analysis on an initial sample of the dataset to gain general insight into inverter-relevant devices and metrics. The goal is to understand the structure of the data, identify broad patterns, and ensure alignment between the observed behavior and expected PV-system theory.

Our EDA examines the overall layout of the dataset (size, format, and available metrics), basic statistical properties of each candidate feature (mean, median, quantiles, range, and data types), and the event-logging structure. We also compute aggregation statistics across devices (such as mean-of-means, between-device standard deviation, and device-level minima and maxima) to evaluate consistency across units. Missingness is assessed both globally and per feature to understand data availability. Visual inspection of key signals is performed through time-series plots, and we compute Pearson correlation matrices for raw features, per-device correlations, and inter-device correlations to identify broad relationships.

The insights from this exploratory analysis guide subsequent preprocessing choices and shape the feature considerations used across the modelling components. Not all results of initial EDA is reported due to data confidentiality.

3.6 Data Processing

Data management is critical to any machine learning pipeline. As mentioned, provided data is in 5 minute increments from sensors, including energy output by inverter. According to the literature reviewed, sensor errors are not uncommon. Additionally, energy produced varies significantly based on numerous confounders like temperature and weather patterns.

To the greatest extent possible, our data processing attempts to distinguish between true errors and helpful outliers. For example, we avoid a policy that blindly imputes values when large amounts of data is missing or sensor outputs are uncommonly high/low because these could be indicative of mechanical errors. Gedde-Dahl (2022) suggests filtering nighttime data, obvious sensor errors, missing/stale data, inverter tracking errors, weather/irradiance/incidence angle based filters, and separating cleaning issues from faults (p.7).

3.6.1 Data Imputation Strategy

The dataset reflects real-world conditions, including unexpected anomalies which introduce periods of no reporting or erroneous reporting. There missing values are handled in two ways: imputation (filling nulls with 0) or interpolation (linear, derivative aware, etc.). Utilizing these methods the missing values are effectively handled throughout the dataset.

3.7 Predictive Maintenance

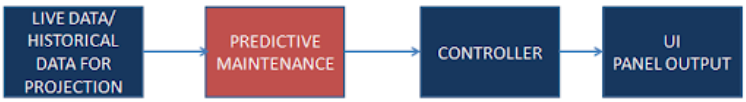

Figure 3: High-level flows of Predictive Maintenance

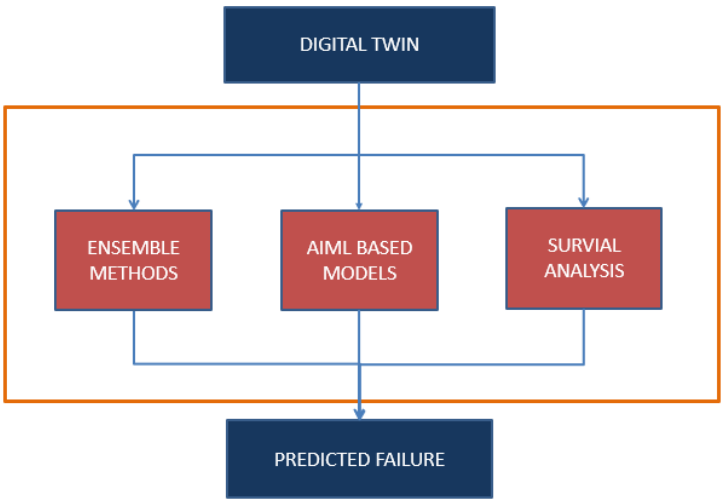

Figure 4: Overview of Predictive Maintenance Sub-Components

Because energy production is fundamentally a mechanical process, our goal is to develop a system-failure prediction methodology that relies less on raw time-series patterns and more on identifying meaningful precursors to failure. Several machine-learning–based approaches have been proposed for predictive maintenance, and in this study we explore two primary strategies: traditional survival analysis methods (Cox Proportional Hazards and Kaplan–Meier) and time-series models for failure prediction.

A major challenge for MN8 is the absence of labeled failure events, specifically, when and why a system failed. Despite having five years of data that include periods of disrupted power generation, it is difficult to determine the true cause of each disruption. For example, if a value of zero were interpreted as a failure, the model would incorrectly flag every night as a system failure, even though the system is operating normally but producing no power. Because of this lack of reliable labels, applying survival-analysis-based machine learning is not feasible.

Instead, we apply statistical monitoring techniques to the model's predicted values for future days. By comparing predictions against established thresholds, we can detect when the system begins to degrade or perform abnormally. This approach allows us to identify potential failures without requiring explicit failure labels.

Although the predictive-maintenance and forecasting models share similar methodological approaches, the key difference lies in the data source: Predictive Maintenance relies on daily outputs generated by the DT, whereas Forecasting uses raw operational data taken directly from the physical system to perform predicting of 1 day or 2 days in the future

3.7.1 Key Modeling Strategies from Literature Review

We initially planned to evaluate a wide range of models commonly used in the literature, including decision trees, random forests, linear regression, deep neural networks, gradient-boosted machines, k-nearest neighbors, LightGBM, CatBoost, XGBoost, and others. During implementation we decided to focus on two methodological frameworks:

1. **Time-Series Modeling for Predictive Maintenance**: We employ a LSTM model to forecast inverter output active power. The model operates continuously, using several days of historical data to predict the subsequent two days of power-generation values. From the predicted output, we compute a simple statistical mean and compare it against a predefined threshold. If the predicted value falls below this threshold, a notification is triggered.

2. **Survival Analysis**: We also apply survival-analysis techniques by structuring recent days of data as inputs to the Cox Proportional Hazards model and the Kaplan–Meier estimator. These methods provide estimated probabilities of failure over time and indicate when a failure event becomes likely.

Each day, both predictive approaches generate updated results based on the most recent data. These predictions can then be presented to analysts to support maintenance scheduling. Detailed explanations of each method are provided in the sections that follow.

3.7.2 General Modeling Steps

1. We begin by splitting data into a train and test set. We hold out the test set and perform the following steps on the train set.
2. If the model requires a reduced feature set to work properly, we will first apply a variable selection technique like Lasso.
3. If the selected model is sensitive to collinearity or requires normalized data, we will apply a method to combine highly correlated features like averaging/selecting one and standardize data.
4. If the modeling technique requires parameter optimization like Random Forest or XGBoost, we will do so using grid tuning or Bayesian Optimization methods.
5. For model comparison, we plan to use cross-validation. This process involves repeatedly splitting the training set into a new training set and validation set, training our model on the new training set, making predictions on the validation set, and tracking test error metrics. After many splits, the range of errors can be plotted and we can compare models.

3.7.3 Long Short Term Memory (LSTM) for Predictive Maintenance

A LSTM model is a specialized type of Recurrent Neural Network (RNN) designed to learn patterns in time-series data, particularly when long-term dependencies are important. LSTMs are widely used in predictive maintenance because they can effectively model system behavior and forecast future signals. In our case, we use an LSTM network to predict the inverter's active power output for the next two days based on the previous seven days of observed data.

The architecture of this model is shown in Table 1. This model is a sequence-to-sequence LSTM forecasting model designed to predict 2 days of future time-series data based on the previous 7 days.

Table 1: LSTM Model Architecture for Predictive Maintenance.

Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 2016, 128)	66,560
dropout_2 (Dropout)	(None, 2016, 128)	0
lstm_3 (LSTM)	(None, 64)	49,408
dropout_3 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 256)	16,640
dense_3 (Dense)	(None, 576)	148,032

3.7.3.1 Data Preparation The input to the LSTM model is constructed from a 7-day window, where each day consists of 288 measurements (one reading every 5 minutes). In total, the dataset includes the beginning to late summer in 2025 days of observations, corresponding to 6 to 8 months. These data form a 2-D matrix that is flattened into a single sequence of 44,064 time steps.

The LSTM uses a sliding window approach:

- Input window: 7 days × 288 samples = 2,016 time steps
- Output window: 2 days × 288 samples = 576 future time steps

For training, the data are reshaped into the standard LSTM format: (samples, timesteps, features). The model uses only one feature, measured active AC power, to learn the temporal patterns. Thus, each training example consists of 2,016 timesteps of AC power measurements (one feature), and the model predicts the next 576 timesteps of AC power output.

3.7.3.2 LSTM Predictive Model The model architecture consists of two stacked LSTM layers. The first LSTM layer contains 128 units, followed by a second LSTM layer with 64 units. To reduce overfitting, a Dropout layer is applied after each LSTM layer. Following the recurrent layers, a Dense layer with 256 units and a ReLU activation function is added. The final output layer is a Dense layer with 576 units, corresponding to the 2-day (576-timestep) prediction horizon. In total, the model contains 280,640 trainable parameters, occupying approximately 1.07 MB of memory (as shown in the figure above).

3.7.3.3 Training and Testing LSTM Predictive Maintenance Model With the dataset from the beginning to late summer in 2025 days of observations available from one inverter, we adopt an 80/20 split, using 80% of the data for training and the remaining 20% for testing. Each training sample consists of a 7-day input window, and the corresponding target is the subsequent 2-day output window. We slide this 7-day window forward by one day at a time, meaning that each new training sample replaces only the oldest day with a newly added day.

The model is trained using the Adam optimizer, and its performance is evaluated using Mean Squared Error (MSE) and MAE. Although the training process is computationally intensive and requires a considerable amount of time, the model ultimately achieves predictions that closely match the true observed data. The result is discussed in Chapter 4.

3.7.4 Ensemble Methodologies with Clustering for Predictive Maintenance

3.7.4.1 Data Preparation The goal of our data preparation was to reduce noise, engineering features, and standardize our data. The original data spanned over multiple files, and measurement types that required consolidation, filtering, and transformation. The following sections describe and define our end-end approach to process the available data that eventually is used for clustering, which later becomes the foundational dataset for machine learning.

3.7.4.1.1 Data Aggrication Our first step was to consolidate the available data into a single file. Focusing on the most recent complete year data, which we believe to be more relevant, we selected the years 2022, 2023, and 2024. Since the size of our dataset was huge, we used Polars LazyFrames, which only executes computation well explicitly told to do so. This allowed us to manipulate the data without loading the whole dataset into memory. The resulting file served as the primary input for all

of the following work, such as filtering, cleaning, feature engineering, clustering, and eventually model development

3.7.4.1.2 Data Filtering and Data Clearing We decided that the most suitable data for our use belonged to inverter X (ommited for data confidentiality). We filtered the data down to inverter X data, thereby reducing the size of the data making it more managable. We dropped the data that we identified as redundant, such as alternate timestamps, and the data not carrying any useful signal, such as constant metadata fields.

Measured active power was identified as our target variable, and we ensured that it was present in every timestamp. So every timestamp retained had to have a non-null value for active power. Any duplicate entries on device ID, timestamps were also removed to ensure unique entries, and we only retained columns that would be useful to us, namely device ID, event timestamp, metric, and the value of the metric.

Our next goal was to only use the data within the irradiance hours of the day. Using the Astral library we generated a suntable, which provided us with site specific sunrise and sunset times for every single entry. Using this suntable, we were able apply a daylight filter to drop any data that was outside of the hours of sunrise and sunset.

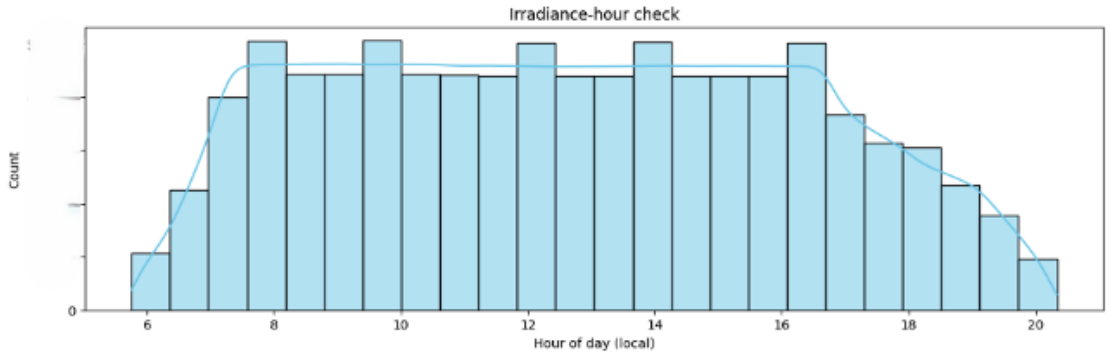

Figure 5: Irradiance by hour of day.

Lastly, we use a filter to remove any outliers, remaining inconsistencies, and/or measurements that are physically unrealistic. At this point we end up with a clean dataset that is filtered on measurements only recorded with the irradiance hours of the site location.

3.7.4.1.3 Data Transformation: Long to Wide format Originally, our data was in a long format, which is good for data storage and exploratory work. It was not suitable for machine learning models that need a consistent set of numerical features per observation. Therefore, to preprocess our data to be good input for machine learning models we pivot from the current long format to a wide format. In the wide format each metric, under value, becomes an independent column, while each row corresponds to a device ID, time stamp pair. The wide format also highlights when a metric does not appear for a given time stamp, which is useful later on for assessing missing values.

3.7.4.1.4 Feature Engineering We create a number of features to try and capture behavior and patterns of power generation. These features are at the center of our transformation process.

Normalization and Operational Context Features

We create two checks, norm_power and is_limited, that we use later for labeling.

Normalize AC Power: Since raw AC power varies across different weather conditions, temperature, irradiance, configuration we standardize these measurements to land between a value of 0-1. This helps

in simplifying compassion across different sessions, days, time of day while maintaining interpretability. We do this by dividing the target AC_POWER.MEASURED by the max capacity.

$$\text{norm\_power} = \frac{\text{Measured AC Power}}{\text{Max Capacity}}. \tag{1}$$

Limited V. Non Limited operation: Inverters may limit their power output due to numerous reasons, such as environmental conditions, grid constraints, etc. We risk falsely marking something as a failure if the drop in power is intentional. We keep a buffer of 10% so that we accidentally don't flag a normal fluctuation is_limited.

$$\text{is\_limited} = \begin{cases} 1, & \text{Measured AC Power} \leq 0.9 \cdot \text{AC Power Limit Setpoint}, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

A measurement of below 60% and an is_limited flag of 0 was considered as detecting a true drop in power. This is recorded as a fail0 column. This produced a failure rate of 10.76% which seems reasonable.

Capturing Daily and Seasonal Cycles

The way we capture time and day of the year do not capture the true relationship of distance between them. We needed to capture these in a way that would be useful for machine learning.

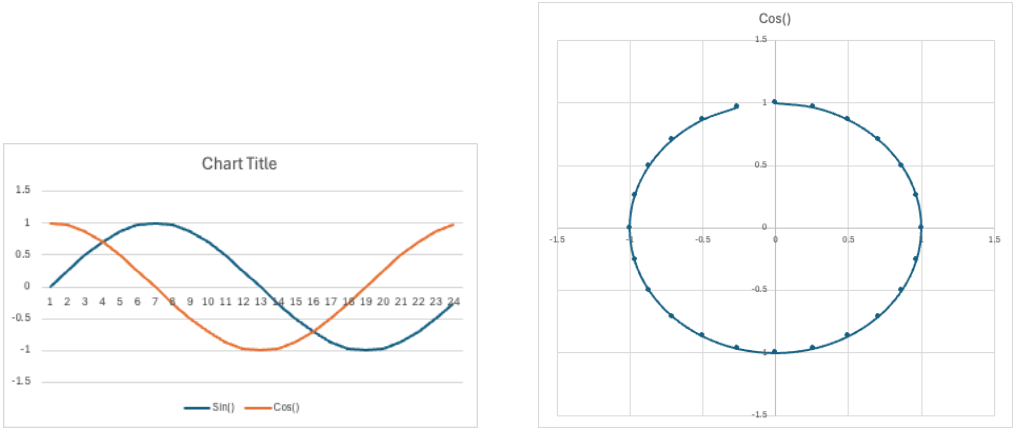

Figure 6: Foundation for seasonality.

Hour of day encoding: Solar power generation depends on the time of day. However , the way we capture time does not capture the true relationship between time. For example, 23:00 is closer to 0:01 than it is to 19:00, this does not help in machine learning. To resolve this we encode time using sine and cosine and the idea of a unit circle. That way we have a representation of time that respects cyclical continuity.

$$\text{hour\_sin} = \sin\left(2\pi \cdot \frac{\text{hour}}{24}\right), \tag{3}$$

$$\text{hour\_cos} = \cos\left(2\pi \cdot \frac{\text{hour}}{24}\right). \tag{4}$$

Day of year encoding: Similar to the time of say we need to encode the day of year that respects cyclical continuity and captures seasonal effects, such as summer maxima and winter minima in irradiance, that effect power generation

$$\text{doy\_sin} = \sin\left(2\pi \cdot \frac{\text{Day of the year}}{365}\right), \tag{5}$$

$$\text{doy\_cos} = \cos\left(2\pi \cdot \frac{\text{Day of the year}}{365}\right). \tag{6}$$

The slot5 Feature and Monthly Grouping

Since our data is captured at a regular fixed five minute interval, every day has 288 consistent slots. We need to be able to compare power generation across time therefore we created the following two

Slot5 (0-287): This allows for us to compare sloths across days. Slot 22 today can be compared to slot 22 from yesterday.

Monthly grouping: Because power generation is dependent on the season we create a month column. Grouping timestamps by month makes it easier for machine learning models to learn generation patterns.

K & M Metrics
We use the K & M metrics to identify failure. To avoid marking natural drops in power, we only mark a failure when the drop in power is continuous over a window of time.

K = rolling window size in 5 minute intervals
M = how many of these k time stamps must show low generation

So,

K = 24 → look at the last 2 hours (120 minutes 5 * 24)
M = 20 → at least 20 of those 24 slots must show low generation (fail0 =1)

If this condition is true mark a failure

Reference AC POWER, ref_ac_95m

For each time slot, we calculate ref_ac_95m by grouping the data by month and slot5 and then computing the 95th percentile of AC_POWER.MEASURED over all historical days in that group. This provides a realistic upper bound for expected performance under similar solar conditions.

Handling Missing Values and Feature Reduction

Because of the enormity of the size of our data we decided to keep good data and drop any missing values. Were able to do this because we had a sizable dataset after dropping any values with Nans

Correlated metrics: We generated a correlation matrix and removed correlated metrics with a threshold of 60%, except protected columns such as AC_POWER.MEASURED and AC_POWER_LMIT_SETPOINT.MEASURED. This improves the stability of the machine learning models and also helps eliminate multicollinearity and reduce redundancy.

Failure Labeling

We use two features to label something as a failure namely fail0 and failure

Fail0: This is a baseline flag used to detect a dip in power. The flag is marked as 1 if normalized power,norm_power, is $< 60\%$ and is_limited is false.

Failure: In general, failure is defined using a K–M persistence rule: over the last K five-minute slots, at least M of them must have fail0 = 1 for a failure to be marked. In our case, we use K = 24 and M = 20. This means that in the past two hours, at least 20 slot5 values must be marked as fail0 = 1.

3.7.4.2 Clustering We apply two different unsupervised clustering algorithms to our data, namely K means and DBSCAN to discover any patterns in our data. K means is a centroid based approach

and requires the user to specify the numbers of clusters. These clusters are sensitive to outliers and assume that clusters are roughly spherical. Whereas DBSCAN, as opposed to assuming spherical shaped clusters, can discover arbitrary shaped clusters, and are also good at detecting noise and outliers. Also, it does not require the user to specify data. By using these two different clustering algorithms we feel like we have approached two different directions.

3.7.4.2.1 K Means Clustering (Centroid Based) The first clustering algorithm we used was K means. We used K = 4 to cluster data into four clusters hoping to capture four different times of day, namely "morning ramp up", "early mid day", "peak irradiance hours" and "late afternoon."

3.7.4.2.2 DBSCAN (Density Based) DBSCAN was used as a second unsupervised clustering algorithm. It groups dense regions automatically and labels sparse or irregularities as noise. The parameters that we used were eps = 0.8 and min_samples = 20. This resulted in 128 clusters in total with a dominant cluster with 16,940 points and it identified noise with a cluster size of 6,187 points.

3.7.4.3 Model Development and Hyperparameter Sweeps After we have performed feature engineering, labeled, and clustered our data we train several machine learning models, namely Extreme Gradient Boosting (XGBoost), RF, AdaBoost (AdaBoost did not complete its full sweep), and CatBoost. We wanted to use at least one bagging, random forest, and one boosting, XGBoost, algorithm. For every model we ensure that our test train split keeps the chronological order by using shuffle=False. This ensures that there is no leakage and that the model is evaluated on the data from later timestamps and now from the timestamps used in training. We did not use the grey wolf optimizer because of the size of the sweeps we used.

All non predictive columns, which included device identifiers, timestamps, and intermediate labels were dropped. Performance was measured using AUROC and PR-AUC, and because our data was highly imbalanced we emphasize more on PR-AUC.

3.7.4.4 Predictive Anomaly Risk Flagging: In the literature review the authors Liu & sun(2019) limited their future horizon window looking out one, two, and three hours ahead to identify abnormal behaviour. In contrast we ask a more ambitious question: Can we detect signals in the data that point to anomaly-like behaviour 48 hours ahead?

This is a substantial stretch as the 48 shift naturally weakens the signal, as the device behaviour two days ahead of failure is subtle. Our intent here is primarily academic curiosity: we did not expect high predictive performance, but we wanted to determine whether the models could capture anything above the baseline failure rate, which for our dataset was 10.76%.

If the model can perform anything above the baseline failure then maybe some useful information is captured, however faint. To explore this we developed a flag called failures_future_48h by simply moving it ahead by 48 hours from the actual failure. This shifted label uses the same engineered features and passes through the same methodology as our original failure modeling pipeline. We should note that this flag does not signal a failure but only a possible risk of future failure.

3.8 Survival Analysis

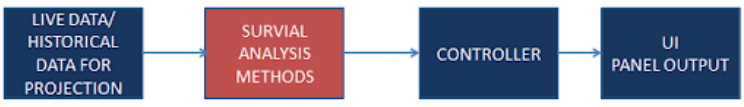

Figure 7: High-level flows of Survival Analysis Methods.

Survival analysis is used in this project to estimate the time until a fault occurs. Three models are employed: Kaplan-Meier (KM), Cox Proportional Hazards (Cox PH), and parametric models (Exponential, Weibull, and Log-Normal). Using time-series data collected from environmental and system sensors, the KM and Cox PH models utilize all available variables to estimate survival probabilities. In contrast,

the parametric models use only seven key variables identified by the sponsor. This technique has been mentioned in the paper Jackson et al. (2024). These survival analysis models will be used in this project by shifting the focus from Solar Flare events to system degradation or failure.

All these survival models can be applied to predict when a failure might occur to a solar unit, from the beginning when it reaches the marked level (70% of power) to when it will reach the warning level (65% of power). As new data arrives to the survival analysis component, each survival model predicts the time remaining until failure, based on the system output's power. The time it takes for the power to drop from approximately 70% to 65% is considered the "warning time," which serves as an estimate for required maintenance. As a sub-component living inside the Predictive Maintenance component, the output of Survival Analysis will be sent as an alert to the User Interface Component

For the KM model, we use the AC_Power.Measured values from the prediction dataset. Using the same dataset includes the beginning to late summer in 2025 days of observations (with 288 measurements per day), the KM model first constructs its event labels. We define an alarm watermark corresponding to 30% productivity. To set the numerical threshold, we multiply this 30% watermark by the median daily power output computed across the dataset from the beginning to late summer in 2025 days. Any day with productivity below this threshold is labeled as a "bad day."

Based on these labels, the KM model forms two required series:

1. Duration: the number of days until a failure occurs.
2. Observation: whether the current day is a failure (i.e., its productivity falls below the threshold).

These two series allow the KM estimator to evaluate the survival probability and estimate how soon a failure is likely to occur.

The KM model acts on these steps:

1. Converts daily sensor data into daily productivity.
2. Defines failure when productivity drops below a threshold (30% of median).
3. Calculates, for each day, how long until such a failure happens (or is censored).
4. Fit the model and apply the model to the sample input data to find:

- How long the inverter stays healthy (days, hours).
- When survival probabilities cross warning levels (e.g., 10%).

Provides an early-warning signal based on survival statistics. The output of this model is discussed in detail in the analysis section.

3.9 Forecasting

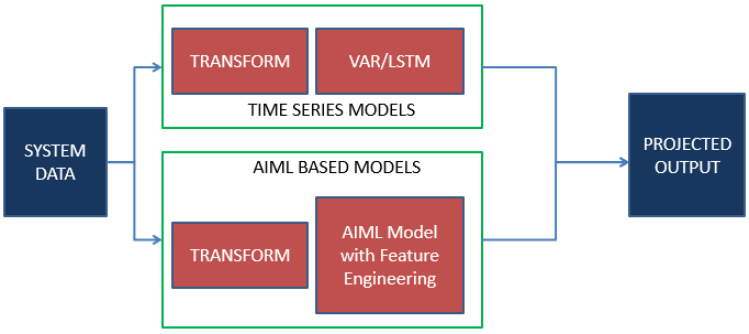

Figure 8: Time Series and AIML model Forecasting Sub-Components.

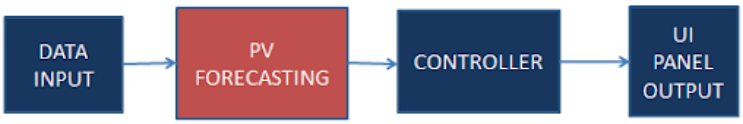

Figure 9: High-level flow of Forecasting.

3.9.1 Multivariate Time Series Forecasting (Time Series Model)

In this section, we perform multivariate and univariate forecasting of PV active power output using a diverse set of models to benchmark performance and capture different temporal dynamics.

- Data Preprocessing & Feature Engineering:
  - Aggregation: Data is aggregated to the site level or device type level by summing target metrics (AC Power) and averaging exogenous metrics (Irradiance, Temperature, etc.) over consistent time intervals.
  - Filtering: To focus on productive hours, the dataset is filtered in to include only daylight periods where AC Power or Irradiance is positive.
  - Cyclic Features: Time of day information is encoded using sine and cosine transformations to capture daily periodicity. Wind direction is similarly transformed.
  - Scaling: The target variable (AC Power) is scaled to a percentage (0-100%) based on the maximum value observed in the training set to ensure interpretability and training stability.
- Forecasting Models: We employed a multi-model approach to evaluate different forecasting strategies:
  - Baseline Decomposition: A hybrid approach that decomposes the series into a linear trend(modeled via Linear Regression) and a stable residual component (modeled via Gradient Boosting or XGBoost).
  - Vector Autoregression (VAR): A multivariate statistical model that captures linear interdependencies between the target variable and exogenous features like irradiance and Temperature.
  - LSTM: A Deep Learning Recurrent Neural Network (RNN) implemented in PyTorch, designed to learn complex, non-linear dependencies in sequence data.
  - Chronos: A pre-trained, transformer based time series foundation model applied in a zero-shot setting to leverage transfer learning capabilities.
  - SARIMA (Seasonal ARIMA): An extension of ARIMA that explicitly models seasonality, suitable for data with strong periodic patterns.
- Evaluation Strategy:
  - Train/Test split: The data is split chronologically, with the first 70% used for training and the remaining 30% for testing.
  - Metrics: Model performance is evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R^2) to assess predictive accuracy and goodness of fit.

3.9.2 Forecasting Failure With Trend Decomposition and Machine Learning

In a recent paper by Gurcan et al. (2024), propose a novel method to improve solar power forecasting. Their approach involves decomposing the data and incorporating irradiance and seasonal patterns as exogenous inputs. This preprocessing step helps better capture the key factors affecting overall output. The forecasting is performed using a machine learning algorithm. This is the approach we plan to integrate into the forecasting component of our project. As new data arrives, both linear and non-linear components are considered. The algorithm begins by decomposing the time-series data into seasonal and residual components. Stable (non-trend) values are predicted using a machine learning model, while trends that extend beyond the training range are extrapolated using Ordinary Least Squares (OLS); otherwise, the trend value is kept as a projected value. The final forecasted output is generated by merging the projected stable and trend components. The algorithm detail from the paper is in the following figure:

Algorithm 1. Trend decomposition and forecasting.

Input: time_series

Output: prediction

```
trend, resid = TrendDecompose(time_series)
stable = Subtractor(time_series, trend)
predicted_stable = PredictWithML(stable)
predicted_trend = trend
if (isExtrapolatedPrediction = True) then
    predicted_trend = PredictWithLinear(trend)
end if
prediction = Merge(predicted_stable, predicted_trend)
return prediction
```

Figure 10: Trend Decomposition and Forecasting (Gurcan Kavakci et al., 2023, p.6).

The machine learning model used is a Deep Neural Network. According to Gurcan et al., this method significantly improves forecasting accuracy, reducing error rates by 5% to 39% in testing with the data from a PV plant in Turkey. As mentioned in the paper, the accuracy is particularly for projections made two days ahead in the future. For this project, the projection will be 7 days. The output is delivered to the UI component for simulation purposes. This simple yet effective algorithm is one of the core models implemented in the forecasting component of this project.

The purpose of this Forecasting Model is to estimate the expected power output for the next day. The algorithm described in the literature leverages the time-series nature of the data by decomposing it into two components: seasonality and trend. Separate AI/ML models are then applied to each component. This approach is theoretically more accurate than training a single model on the full time-series directly, because seasonality and trend often exhibit distinct patterns that can be learned more effectively in isolation and then combined.

The referenced paper provides limited detail on the specific models used, but it outlines a general strategy of applying AI/ML methods to the seasonal component and linear regression to the trend component. Using the same inverter dataset as in the Predictive Maintenance work, we tested several models, including Random Forest, Linear Regression, Deep Neural Networks, LSTM, and RNN architectures, and found that LSTM provided the most accurate next-day predictions. It is important to note that, unlike the Predictive Maintenance model, the Forecasting model uses raw system data rather than outputs from the DT.

For both the seasonal and trend components, we use a 7-day historical window to predict 1 day ahead. First, the time-series is decomposed into its seasonal and trend datasets. The seasonal component is modeled using an LSTM network. Although the paper suggests using linear regression for the trend component, our data revealed multiple distinct linear segments occurring at different times of day, and these segment patterns varied across inverters. Due to this segmented structure, a single linear regression model could not capture the overall trend behavior. Consequently, we replaced the linear regression approach with a Deep Neural Network, which is capable of modeling the entire trend pattern across the full day.

The LSTM model for Seasonal Prediction is designed as in Table 2:

Table 2: LSTM Model Architecture for Seasonal Prediction in Predictive Maintenance.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 2016, 128)	66,560
lstm_1 (LSTM)	(None, 64)	49,408
dense (Dense)	(None, 288)	18,720
reshape (Reshape)	(None, 288, 1)	0

The Deep Neural Network designed for Trend data is as described in Table 3:

Table 3: Deep Neural Network Architecture for for Trend in Predictive Maintenance.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 2010, 64)	512
conv1d_1 (Conv1D)	(None, 2006, 64)	20,544
max_pooling1d (MaxPooling1D)	(None, 1003, 64)	0
conv1d_2 (Conv1D)	(None, 1001, 128)	24,704
global_average_pooling1d (GlobalAveragePooling1D)	(None, 128)	0
dense_1 (Dense)	(None, 512)	66,048
dense_2 (Dense)	(None, 288)	147,744

Both the seasonal and trend models generate a one-day-ahead prediction. The sum of these two predictions represents the forecasted power output for the next day. We refer to this combined output as the expected system data. This forecast is valuable not only for performance assessment and analytics, but also for validating the DT and supporting anomaly detection when used alongside the DT's own predictions.

3.10 Digital Twin

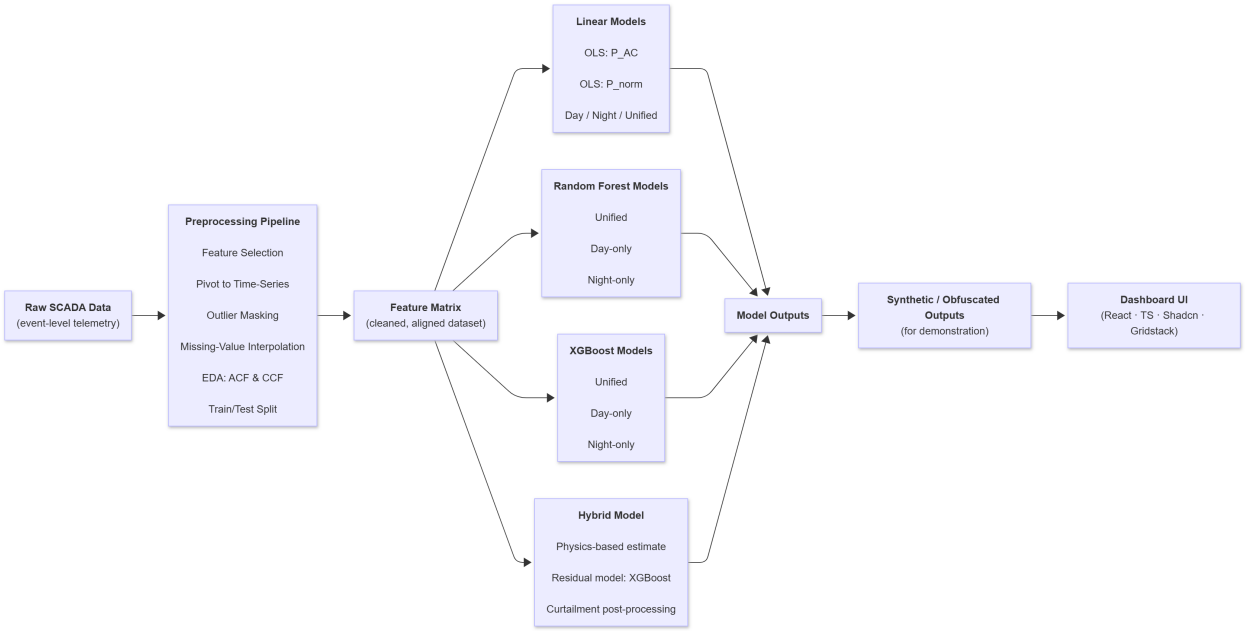

Figure 11: Digital Twin system architecture diagram.

A DT for PV power estimation offers multiple applications, including solar PV site planning and improved situational awareness in O&M. Comparing measured and estimated power enables performance monitoring by alerting operators to sustained discrepancies. Additionally, DTs can account for PV plant performance degradation over time through progressive learning. System aging may be tracked by periodic estimation logging and comparison (Walters et al., 2023).

The DT developed in this project is a numerical model representing an inverter within the solar PV plant. Rather than adopting a 3D-geometry representation or a comprehensive physics-based simulation suite, this work employs a numerical formulation tailored to O&M objectives. The DT's primary function is to ingest inverter sensor data and estimate the active AC power output at each corresponding time step.

Two modelling approaches are used: purely data-driven machine learning models and a hybrid model that combines foundational inverter physics with a machine learning component. All machine learning models (including the ML component of the hybrid model) are trained separately for each inverter rather than on a pooled dataset. This choice reflects device-specific behavior arising from hardware tolerances, ageing, thermal characteristics, and local micro-environmental conditions. Training per inverter avoids averaging these differences across units and allows the DT to capture each device's unique operational profile.

3.10.1 Preprocessing

Developing a reliable DT requires a well-defined preprocessing pipeline to ensure the input data is clean, consistent, and physically meaningful. The raw SCADA telemetry contains heterogeneous device-level metrics, irregular event logs, missing values, and sensor-specific behaviors that must be reconciled before modelling. The preprocessing pipeline therefore consists of:

(i) physically motivated feature inclusion and exclusion criteria,

(ii) restructuring event-level data into a unified time-series format,

(iii) chronological train–test partitioning to prevent leakage,

(iv) controlled interpolation for replacing missing values,

(v) general and regime-aware outlier removal, and

(vi) exploratory analysis to guide later modelling decisions.

Each step is described below.

3.10.1.1 Initial Feature Inclusion Criteria The DT focuses strictly on reproducing inverter-level behavior. Accordingly, most upstream and downstream device metrics were excluded. Downstream metrics largely reflect behavior already captured by inverter sensors, and upstream metrics do not directly influence the inverter's internal power-conversion process. The feature set is therefore limited to inverter-level electrical, thermal, and operational measurements. The only external variable included is the farm-wide plane-of-array (POA) irradiance, aggregated as the per-timestamp median across available sensors to obtain a robust site-level irradiance estimate.

Grid-tied inverters operate in two distinct regimes. During daytime, active power generation depends on irradiance and associated thermal conditions. After sunset, no DC generation occurs and the inverter operates in standby unless providing reactive-power support, where behavior is governed by grid-voltage interactions (Tharuka Lulbadda & Hemapala, 2022). This necessitates distinguishing daytime and nighttime behavior and motivates the inclusion of irradiance metrics.

Domain understanding also motivates the exclusion of several metrics. Communication-link status and general health flags do not influence active power when valid data is present. Positive and negative DC pole voltages were removed because the same information is already captured by retained DC metrics, and modelling ground-fault behavior is out of scope. Hardware version contributes no predictive value, as hardware-specific differences manifest through operational measurements. AC and DC insulation resistances similarly provide no utility for predicting active power. Pressure readings from low- and medium-voltage compartments were excluded because their influence on cooling performance is already

reflected in retained thermal metrics. Finally, output apparent power and phase-level AC currents were included only as potential lagged inputs; when used contemporaneously, they are algebraically dependent on the target variable and would introduce leakage.

A subset of inverter-level SCADA metrics was therefore selected based on physical relevance to power conversion and consistency across the dataset. Preliminary exploratory assessments were used only to confirm signal stability and operational meaning. The final list of metrics considered for modelling is provided in **Appendix A: Digital Twin Initial Feature List**.

3.10.1.2 Data Structuring and Pivoting The raw SCADA telemetry is event-oriented, with each metric logged as an individual time-stamped record. For modelling, the selected metrics were pivoted into a unified time-series structure in which each timestamp corresponds to a single row and each metric to a column. This representation aligns all signals on a common 5-minute interval and produces a consistent feature matrix suitable for supervised learning.

3.10.1.3 Train-Test Split To ensure a realistic evaluation and avoid leakage, the dataset was divided chronologically into training and test segments. Because PV output is seasonal, the final year of data (from late July 2024 onward) was held out entirely as the test set. A one-month buffer was inserted between training and testing, with the training period ending in late June 2024. The training portion was later subdivided into training and validation sets for model development.

3.10.1.4 Handling Missing Values For model development, missing values cannot be retained, so missing entries were dropped or interpolated depending on context. Only univariate interpolation methods available in standard time-series libraries were considered, as multivariate interpolation risks introducing leakage between predictors and the target. High-order global polynomial methods were excluded due to known instability on long time series. The interpolation families evaluated include simple, piecewise, spline-based, shape-preserving, and derivative-aware methods. A complete list appears in **Appendix B: Digital Twin Considered Interpolation Methods**.

To ensure interpolation methods were evaluated on representative data, outliers were removed prior to comparison. First, a robust-Z filtering procedure was applied independently to each metric using only finite values and rows where the inverter was producing non-zero active power. This provides a general removal of extreme deviations while avoiding leakage across variables.

Second, regime-aware outlier masking was applied to account for the fact that broad, per-metric filtering does not reflect the underlying relationship between DC input power and AC output power. Operating points were first assigned to physically motivated regimes (such as near-rated operation, standard daytime generation delineated usingPlane-of-Array Irradiance (POA), and remaining conditions) and within each regime, efficiencies were computed to tie input and output behavior. A robust-Z filter was then applied to these regime-specific efficiency distributions, allowing outlier detection to reflect the inverter's expected conversion performance under different operating conditions rather than applying a single global criterion.

Our regime-aware outlier masking requires a temperature-dependent power rating curve to establish physically meaningful bounds near inverter rating. This curve is derived from apparent-power capability, reactive-power behavior, and curtailment constraints. Active power near rating is influenced by reactive power generation due to the relationship between active, reactive, and apparent power:

$$S = \sqrt{P^2 + Q^2}, \tag{7}$$

where S is apparent power, P is active (AC) power, and Q is reactive power. Under unity Power Factor (PF) ($S = P$), the rated power corresponds directly to the apparent-power limit. When reactive power is generated, the maximum feasible active power is instead given by:

$$P = \sqrt{S_{\text{rated}}^2 - Q^2}, \tag{8}$$

where S_{rated} is the rated apparent power. Because inverter rating varies with temperature, we derive a temperature-dependent apparent-power curve $S_R(T)$ by filtering the data to near-ideal operating conditions (no curtailment, unity PF, high POA, and high efficiency). Measured cabinet admission temperatures are then binned in 2 °C intervals, and within each bin we take the 95th quantile of observed apparent power as the empirical rating. This produces a smooth, queryable estimate of $S_R(T)$.

Using this temperature-dependent apparent-power rating, we compute the temperature-dependent rated active power as:

$$P_R(T(t)) = \min\Big(\sqrt{S_R(T(t))^2 - Q(t)^2},\, \alpha(t)\, S_R(T(t))\Big), \tag{9}$$

where $\alpha(t)$ represents the minimum of the relevant active- and apparent-power limit setpoints. After masking outliers, missingness patterns were quantified across each metric. To evaluate interpolation robustness under realistic missingness behavior, synthetic gaps were injected using a bursty Markov-style process calibrated to the observed distribution of gap lengths each candidate interpolation method was applied to reconstruct the masked values, and reconstruction error was assessed using RMSE and MAE. A voting procedure across inverters and metrics was then used to identify the most suitable interpolation method for each metric category. The selected interpolation methods were subsequently applied to the outlier-masked data to produce the final cleaned dataset used for model development. Full mapping of each feature to its chosen interpolation strategy is provided in **Appendix B: Digital Twin Final Interpolation Methods**.

3.10.1.5 Exploratory Data Analysis To understand temporal structure, autocorrelation and cross-correlation functions were computed for relevant metrics. These analyses provide insight into temporal dependencies and potential lead–lag relationships relevant to model design. Empirical findings are presented in Chapter 4.

3.10.1.6 Train-Validation Split For hyperparameter tuning, a secondary split was created within the training period. Ninety percent of the pre-test data was used for training and the remaining ten percent for validation. A one-day buffer was inserted between the sets to minimise leakage due to temporal autocorrelation.

3.10.2 Benchmark Models

To contextualise model performance, several naïve forecasting models were evaluated: a persistence model (predicts the last value), a moving-average model, and an exponential moving-average model. These simple baselines serve as reference points for assessing the value added by the DT. Results are presented in Chapter 4.

3.10.3 Machine Learning Models

We evaluate several machine learning models for estimating inverter active AC power. All features are standardised by removing the mean and scaling to unit variance. Because inverter behavior differs between daytime and nighttime, each modelling approach is evaluated under two strategies:

(i) a unified model trained on all time periods, and

(ii) regime-specific models trained separately on daytime and nighttime subsets.

This structure allows the models to reflect the different physical drivers governing operation across regimes.

3.10.3.1 Feature Engineering Several engineered features are included to support model learning and encode known physical relationships. These include the temperature-dependent rated active power $P_R(T(t))$, per-unit DC bus voltage to capture MPPT deviation and DC-side loading, temperature-based residuals relative to a reference temperature to capture sub-threshold thermal efficiency trends, and per-unit line-to-line voltage deviation to encode grid-side conditions. In addition, rolling averages are

applied to selected variables to improve stability and reduce noise; the choice of which variables receive smoothing is informed by their variability and role in the model. The full list and method of construction is shown in **Appendix C: Digital Twin Engineered Features**.

3.10.3.2 Feature Selection The extended feature set (including both the physically motivated and engineered features) is refined using forward sequential selection. Candidate features are grouped by collinearity, and each group is evaluated before introducing features from other groups. A gradient-boosted model is used to assess predictive contribution at each step, with validation R^2 used as the performance measure. The resulting feature sets for each model variant are reported in Chapter 4.

3.10.3.3 Linear Models We develop two forms of linear regression for each inverter: one predicting active AC power directly, and one predicting a normalized value using a temperature-dependent rated-power curve:

$$P_{\text{norm}}(t) = \frac{P_{\text{AC}}(t)}{P_R(T(t))}, \tag{10}$$

where $P_{\text{norm}}(t)$ is the normalized active AC power at time-step t, $P_{\text{AC}}(t)$ is the measured active AC power, and $P_R(T(t))$ is the temperature-dependent rated active power computed from the power rating curve described earlier. This transformation partially linearises the influence of temperature and irradiance. Predictions of the normalized target can be denormalized using:

$$\hat{P}_{\text{AC}}(t) = \hat{P}_{\text{norm}}(t) \cdot P_R(T(t)), \tag{11}$$

where $\hat{P}_{\text{AC}}(t)$ is predicted active AC power, and $\hat{P}_{\text{norm}}(t)$ is predicted normalized active AC power. For each inverter, five linear models are trained: unified for both the raw and normalized targets; and regime-specific variants for raw targets.

3.10.3.4 Random Forest (RF) A Random Forest regressor is used to capture nonlinear relationships. For reproducibility, a fixed random state is used, with 300 trees, a minimum of 20 samples per leaf, and $\sqrt{\text{features}}$ considered at each split. These defaults are appropriate for structured datasets. For each inverter, unified, daytime-only, and nighttime-only models are trained. Only the raw target P_{AC} is modelled, as tree-based methods are insensitive to monotonic rescaling.

3.10.3.5 Extreme Gradient Boosting (XGBoost) To model more complex nonlinearities, we use XGBoost with 600 estimators, a learning rate of 0.05, maximum depth 6, minimum child weight 5, subsample 0.8, column-sample 0.8, and L2 regularisation. For each inverter, unified, daytime-only, and nighttime-only models are trained for the raw target P_{AC}. As with Random Forests, the normalised target is not used.

3.10.4 Hybrid Model

Physics-based models are stable and interpretable but cannot capture device-specific characteristics such as sensor bias, ageing, or inverter-specific inefficiencies. The hybrid model addresses this by combining a physics-based estimate with a machine-learning correction. The hybrid formulation is:

$$\hat{P}_{\text{hybrid}}(t) = \hat{P}_{\text{phys}}(t) + \hat{r}_{\text{XGB}}(t), \tag{12}$$

where $\hat{P}_{\text{phys}}(t)$ is the physics-based prediction and $\hat{r}_{\text{XGB}}(t)$ is the ML-predicted residual. Because operational characteristics vary across devices, a separate residual model is trained for each inverter.

A secondary investigation examines whether curtailment effects are adequately captured by the ML residual. To enforce curtailment limits explicitly, we evaluate an augmented prediction:

$$\hat{P}_{\text{curt}}(t) = \min\big(\hat{P}_{\text{hybrid}}(t),\, P_{\text{rated AC}}(t)\big), \tag{13}$$

where $P_{\text{rated AC}}(t)$ is the active-power setpoint.

3.10.4.1 Physics-Based Component

The physics model estimates output AC power as:

$$\hat{P}_{\text{phys}}(t) = \eta_{\text{elec}}(\lambda(t)) \cdot \eta_{\text{thermal}}(T(t)) \cdot P_{\text{DC}}(t), \tag{14}$$

where η_{elec} and η_{thermal} capture electrical and thermal efficiency effects respectively, and $P_{\text{DC}}(t)$ is input DC power. Electrical efficiency is defined as:

$$\eta_{\text{elec}}(\lambda(t)) = \eta_{\max}\big(1 - ae^{-b\lambda(t)}\big), \qquad \lambda(t) = \frac{P_{\text{DC}}(t)}{P_{\text{rated DC}}}, \tag{15}$$

where $\eta_{\max}$ is the maximum efficiency, (a, b) are empirical fitting constants (obtained using non-linear least squares fitting), e is Euler's number, $\lambda(t)$ is the load ratio, and $P_{\text{rated DC}}$ is the rated DC power. Thermal efficiency is:

$$\eta_{\text{thermal}}(T(t)) = \begin{cases} 1, & T(t) < T_{\text{threshold}}, \\ 1 - k\big(T(t) - T_{\text{threshold}}\big), & T(t) \geq T_{\text{threshold}}, \end{cases} \tag{16}$$

where $T(t)$ is measured temperature, $T_{\text{threshold}}$ is the derate threshold temperature, and k is the derating coefficient.

3.10.4.2 Machine Learning Residual/Correction Component

The ML component uses XGBoost to model the residual term $\hat{r}_{\text{XGB}}(t)$ between the physics-based estimate and the measured active power. The same hyperparameters as in Section 10.3.5 are used. The residual model uses the same feature pool defined in Sections 10.1.1 and 10.3.1 and is subjected to the same forward sequential feature-selection procedure described in Section 10.3.2.

3.10.5 Testing

All final model evaluations are performed on the held-out test set described in Section 10.1.3. No tuning or model adjustment is conducted using test data. For each inverter and each model variant (unified, daytime-only, nighttime-only), predictions are generated across the full test period and compared against the measured active AC power. Performance is summarized using the Normalised Root Mean Squared Error (NRMSE), computed as RMSE normalized by the range of the target variable, and the coefficient of determination (R^2), which provide scale-independent error evaluation and goodness-of-fit assessment. All numerical results and comparative analyses are presented in Chapter 4.

3.11 Anomaly Detection

In the anomaly section we employed a multi-stage anomaly detection framework to identify irregularities in PV inverter performance. The methodology combines statistical outlier detection with physics-based performance modeling to distinguish between genuine system faults and environmental variability.

- Statistical Outlier Detection (Broad Masking): A robust statistical approach was initially applied to filter gross data errors across all sensor telemetry. The Robust Z-score method was utilized, which is resilient to outliers compared to standard z-scores. For each relevant sensor metric X:

$$Z_{robust} = \frac{X - \text{median}(X)}{1.4826 \times \text{MAD}(X)} \tag{17}$$

Where MAD is the Median Absolute Deviation. Observations with |Z robust| > 3.5 were flagged as broad outliers and masked to prevent downstream contamination.

- *Composite Anomaly Detection.* A rigorous "Overall Anomaly" flag was constructed by intersecting three independent indicators to minimize false positives:
  - AC Power Outlier: Significant deviation in AC Power output (Robust Z-score > 1.3).
  - Power Flow Inconsistency:Instances where DC input current physically exceeded AC output power, indicating sensor mismatch or severe loss.
  - Performance Ratio (PR) Anomaly: Deviations in the Performance Ratio, calculated as:

$$PR = \frac{P_{AC} \times G_{ref}}{POA \times P_{rated}} \tag{18}$$

Where Gref is referenced irradiance (1000 W/mˆ2) and P rated is nominal capacity. Anomalies were flagged via Robust Z-score > 1.5.

An event was classified as a confirmed anomaly only if all three conditions in the composite Anomaly Detection section were simultaneously satisfied.

3.12 Integration/UI

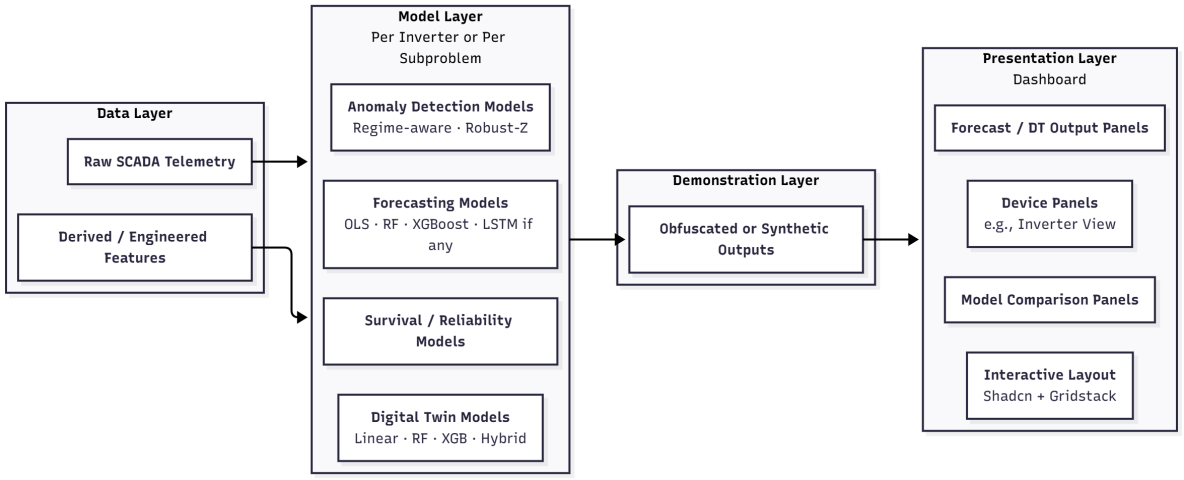

Figure 12: UI system architecture diagram.

To provide a unified interface for presenting the various components of the project, we implement a dashboard-style user interface. The current version of the system is designed for demonstration and integration purposes rather than for operational deployment. It replicates the intended behavior of a functional monitoring tool (displaying model inputs, outputs, and device-level summaries) but does not execute the underlying models in real time. Instead, representative or intentionally obfuscated datasets are used for illustration in accordance with confidentiality requirements.

The interface is developed as a React–TypeScript web application and makes use of Shadcn UI components and Gridstack for layout management. This combination enables a configurable, modular dashboard in which different visual elements (plots, summaries, tables, or device-level panels) can be arranged and resized interactively. The design supports future extension to real-time inference once model-serving infrastructure is integrated, but in its current form serves as a presentation layer for the models and related analyses. The outline of the UI can be can be found in **Appendix D: Graphical User Interface**.

3.13 Ethical Considerations

As with any project involving operational data from critical infrastructure, we reviewed the ethical and practical implications of our work before beginning analysis. MN8 confirmed that the dataset was internally generated and that our team had permission to use it for the academic purposes defined in the capstone agreement. We verified that no legal or regulatory constraints prohibited our use of the data and ensured that all storage and handling practices complied with MN8's confidentiality requirements, including restricted-access storage, encryption, and the removal of identifying metadata.

Throughout the project, we remained mindful of potential adverse impacts that could arise from the development and deployment of data driven tools:

1. Avoiding overreliance on models: The DT, forecasting models, survival analysis, and predictive maintenance outputs are intended to support operational insight, not replace the judgment of qualified engineers. We emphasize that no model should be treated as a definitive indicator of failure or system performance, and physical inspection remains essential.
2. Alignment with environmental and operational goals: The motivation for this work is to support efficient operation of clean, renewable energy systems. While the models aim to reduce downtime and improve generation efficiency, operational recommendations must not inadvertently encourage practices that conflict with sustainability goals. For example, any maintenance strategy inferred from model outputs must consider potential environmental cost, not solely energy yield.
3. Clear communication of model limitations: We ensured that MN8 is informed of the limitations inherent to each modelling approach, including the effects of missing data, unlabelled failures, and the conditions under which predictions may be unreliable through reporting in the paper. This transparency helps prevent misuse of model outputs and ensures they are interpreted as decision-support tools rather than definitive operational directives.

By addressing these considerations, the project maintained responsible data use, realistic expectations of model capability, and alignment with MN8's operational and environmental objectives.

3.14 Trustworthiness of Data

From the provided dataset, we recognize the challenges of making accurate predictions due to missing or NaN values. Labeling failures can be difficult, as a low or missing measurement does not always indicate a true system failure; it could also result from network issues or data transmission drops. Nevertheless, we trust that the data is genuine and directly reflects the system's output. Based on this, we decided to focus initially on a single inverter at a particular farm to develop and validate our models. Once these models are robust, they can be generalized and applied to data from other inverters across the farm.

3.15 Limitations and Delimitations

Imbalanced Data: Across all subproblems, the dataset contained very few confirmed maintenance events and no explicit failure labels. This created a structural limitation for supervised approaches in predictive maintenance and survival analysis. Techniques such as oversampling or SMOTE could theoretically help, but they risk introducing artificial patterns and increasing false positives. Since true failure labels were unavailable, we restricted our analysis to methods that operate under partial or weak supervision, such as DBSCAN-based clustering for predictive maintenance and non-parametric survival models.

Missingness and Data Quality Constraints: The inverter data included substantial missing values, zero readings, and occasional invalid measurements. These issues affected all components of the project. The DT was designed to mitigate some of this through interpolation and physically informed reconstruction, but long gaps introduce uncertainty that cannot be resolved without more complete data. Forecasting models, which require continuous temporal windows, were particularly affected. The presence of nighttime zeros and intermittent drops also complicated the distinction between normal operating behavior and genuine anomalies.

Model Generalizability: All models were developed for a single inverter. This was a deliberate delimitation, since inverters differ in hardware, age, irradiance exposure, and operational regimes. The methodology can be applied to other inverters, but model parameters cannot be assumed to generalize

without retraining. This applies to the DT, the forecasting models, the anomaly detector, and the predictive maintenance pipeline.

Constraints on Physics-Based Modeling: The DT incorporates a hybrid structure, but physically modelling all inverter components was not feasible with the provided dataset. Important thermal and electrical state variables were unavailable. As a result, the physics component uses simplified representations, and the learned residual model must compensate for missing complexity. While this approach performed well, its accuracy depends heavily on the specific inverter used for training.

Black Box Models and Interpretability: Several models used in the project, such as XGBoost, LSTMs, and the residual component of the DT, offer high predictive accuracy at the cost of interpretability. For anomaly detection and predictive maintenance, this raises the question of how much trust to place in models that do not provide clear causal explanations. In an operational setting, predictions must complement, not replace, engineering judgment. Interpretability constraints limit the extent to which these models can be used for automated decision-making without human oversight.

Temporal and Environmental Delimitations: Environmental variables such as temperature, irradiance dynamics, cloud cover, and wind were not consistently available. This restricted our ability to build models that incorporate complete environmental context. Trend forecasting algorithms and survival analysis would likely improve with richer environmental metadata. Similarly, thermal effects captured in inverter specification sheets could not be validated against internal sensor readings, limiting the depth of physics-based modelling.

Delimitation on Scope: The project focuses on detecting performance degradation and predicting maintenance-relevant behavior but does not attempt to diagnose specific failure modes, assess hardware condition, or provide economic optimization. The DT is used to simulate expected performance, but not to simulate internal inverter control behavior or module-level interactions.

Chapter IV: Analysis

4.1 Overview of the Data Environment

The dataset provided by MN8 includes several fields:

- A unique identifier for the device,
- the name of the device,
- the type of device (inverter, combiner, etc.)
- the name of the solar PV plant,
- the longitude and latitude of the Pv plant,
- the generation capacity of the plant,
- event timings (timestamps),
- metrics (active power, apparent power, DC power, etc.),
- and the measured value.

The analysis was conducted on data ranging from early December 2021 to late June 2024 with the held-out test spanning late July 2024 to late July 2025. The data is reported in 5-minute intervals, providing rich source.

4.1.1 Initial Observations

The initial exploratory analyses provide several high-level insights into inverter behaviour and data characteristics. The correlation matrix (see **Appendix E: Correlation Matrices**) shows that AC-side electrical variables have strong correlation with active power, which confirms their relevance as predictors of inverter output. AC and DC feature families are also highly correlated and often linked through algebraic relationships, so using one or the other provides similar information content. DC-side variables typically offer a clearer physical interpretation of power conversion.

Daily operation varies across inverters. Some units exhibit smooth and continuous 5-minute measurements that reflect stable daytime generation. Others show irregular patterns, including extended periods of zero production or abrupt deviations from expected behaviour. Figures 13 and 14 present representative examples of stable and disrupted daily profiles. Axis labels are intentionally omitted.

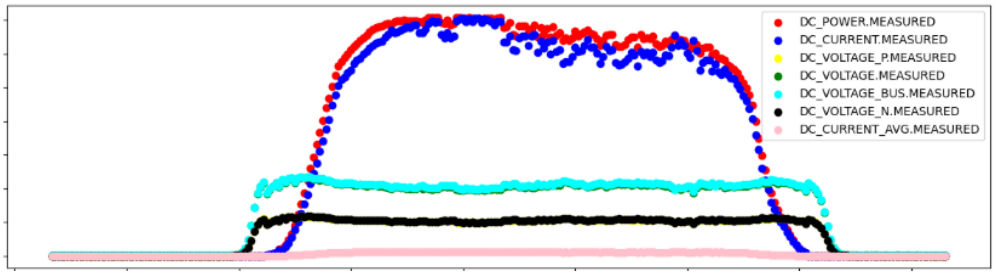

Figure 13: Representative day with stable generation.

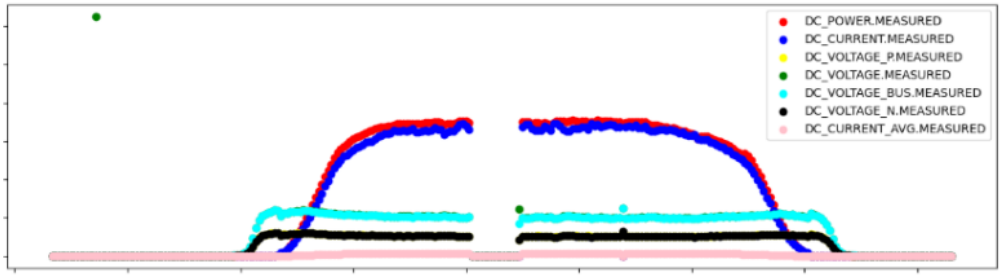

Figure 14: Representative day containing gaps and outliers.

These disruptions can appear at any time and across all inverters. They often take the form of sudden drops, spikes, or missing intervals. Detecting such events and understanding their potential causes is an important goal of the overall software system, since these patterns may reflect sensor issues, thermal behaviour, temporary derating, or early signs of degradation.

Missing values occur throughout the dataset. They arise from normal environmental variation, communication interruptions, or short-lived hardware conditions. These gaps increase the difficulty of predictive modelling, which is why controlled interpolation, outlier masking, and physically informed preprocessing steps (discussed in Chapter 3) are essential.

4.2 Analysis of Subproblem 1: Predictive Maintenance

4.2.1 EDA Relevant to Predictive Maintenance

4.2.1.1 Analysis of LSTM for Predictive Maintenance The historical graph below (see Figure 15) displays the distribution of the dataset from the beginning to late summer in 2025 observations-day dataset. The data include a number of zero values, representing periods of no power generation. The higher power outputs (toward the right side of the graph) contain more data points compared to the lower daily power generation values (on the left), reflecting the typical distribution of power production across the observation period.

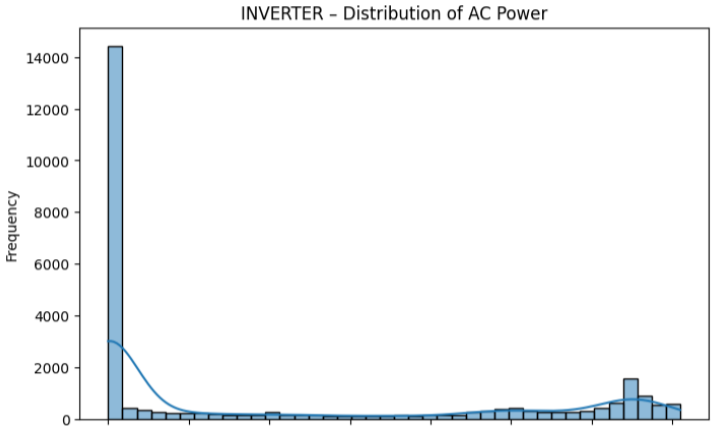

Figure 15: Distribution of inverter output active power.

The data shows that power generation drops to zero each night, while reaching peak values during the daytime.

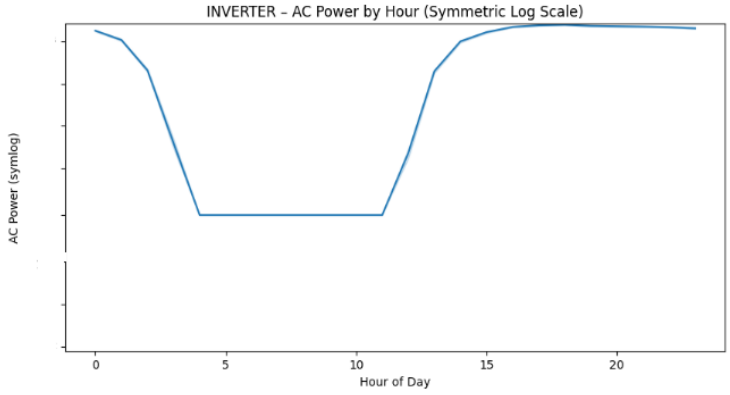

Figure 16: Distribution of inverter output active power by hour.

For the Predictive Maintenance AI/ML model, we use a single dataset consisting of 288 data points per day, representing clean and complete data. Similar to the preprocessing in the survival analysis, outliers, NaN values, and missing data are handled through linear interpolation or replaced with zeros. This preprocessing ensures the dataset is ready for training and testing the LSTM model for Predictive Maintenance.

4.2.1.2 Analysis of Ensemble Method Our exploratory analysis focuses on understanding the structure of the raw dataset and evaluating how suitable it is for clustering and machine learning models. The original data is in long format, where each timestamp contains several individual metrics rather than a single consolidated record, which is not conducive to machine learning. The structure of the data is not usable for machine learning in its current form. So, needed to pivot the dataset into a wide format where each row is a complete feature vector.

The data, used for ensemble methods, spans multiple years (2022-2024) with the same 5 minute resolution (288 points per day). The chronological order is not preserved which will cause temporal leakage if left uncorrected. Our pipeline enforced a forward moving chronological order for the data.

The data is collected 24 hours a day which introduces information that is of no use to us. We filter out the values that are not within the irradiance hours.

Because the data is collected around the clock, half the observations occur at nighttime, where the irradiance is zero. Not only do these readings provide no useful signal but can distort the model, so we filter the dataset to only contain irradiance hour data.

4.2.2 Model Performance and Results

4.2.2.1 Analysis of LSTM for Predictive Maintenance The LSTM model uses outputs from the DT to predict future power generation, with a 7-day historical window as input. Model performance is evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) by comparing predicted values with the true values in the test dataset. For example, on the test data, the model achieves:

- MAE: 5.32% of the mean
- RMSE: 8.72% of the mean

The MAE is about 5.32% of the mean AC power. This reflects good overall accuracy. The RMSE is higher at 8.72%, as expected, because RMSE penalizes larger errors more heavily. The RMSE percentage of 8.72% suggests that some predictions deviate more significantly, likely during rapid changes such as sunrise, sunset, or load spikes.

Overall, these results show that the model is reasonably accurate, generally staying close to actual values, though it struggles somewhat during periods of rapid change.

We ran a sample prediction using the first 7 days from the test dataset. The model's 2-day forecast is graphed alongside the true values for comparison, as shown in the Figure 17.

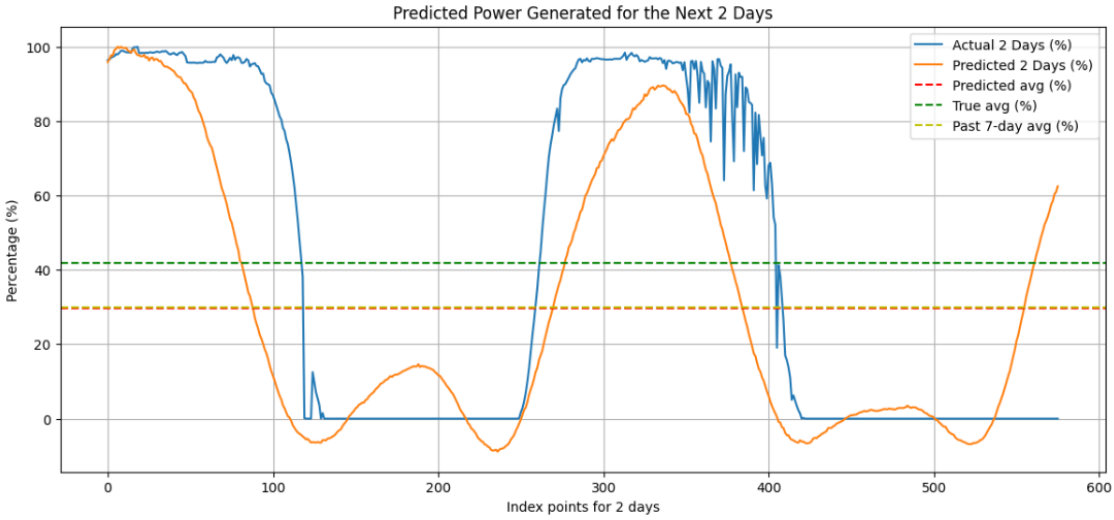

Figure 17: Two-day prediction of power generated.

We also include in the graph the average of the last 7 days of true data and the average of the predicted 2 days ahead. This provides a reference to assess whether the predicted values are higher or lower than the actual measurements from the field. Based on the comparison between these two averages, maintenance notifications can be generated if the predicted output falls below the expected threshold.

4.2.2.2 Analysis of Ensemble Method for Predictive Maintenance We trained four ensemble algorithms namely, XGBoost, Random Forest, AdaBoost, and CATBoost on datasets clustered on K Means and DBSCAN. Extensive hyperparameter sweeps were performed on each algorithm, details of which are listed in Table 4.

Table 4: Ensemble method sweeps.

	Clusters	
Algorithm	DBScan Models	K-Means Models
XGBoost	5,400	5,400
Random Forest	160	160
AdaBoost	500	1000
CATBoost	1944	1944

Because our data is imbalanced we evaluate our model performance on Precision–Recall AUC (PRAUC), the most informative metric for imbalanced data. (Note: The top five performing model performances are shared in the *Appendix E: Ensemble Model Performance.*)

XGBoost Performance

XGBoost had the strongest performance among all four methods. The best model was trained on data clustered using DBSCAN.

- PRAUC (test): 0.8151

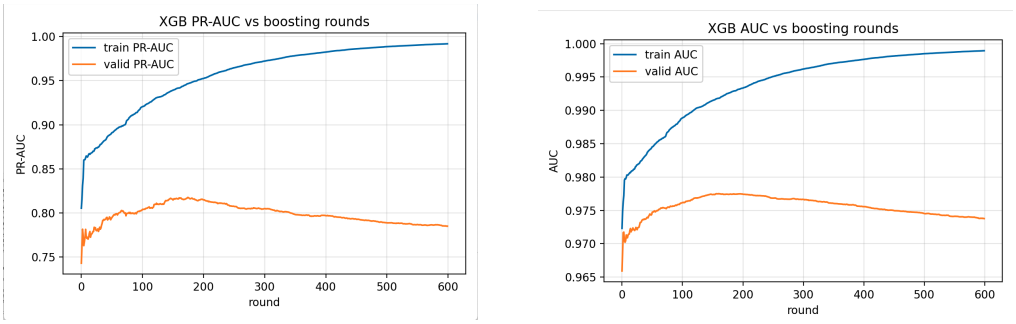

Figure 18: AUC and PR-AUC Curves.

K Means best PRAUC (test) score was a respectable 0.802 but it lagged behind DBSCAN

AdaBoost Performance

Adaboost performed second best, and the best model was trained on data clustered using DBSCAN.

- PRAUC (test): 0.8136

K Means best PRAUC (test) score was a respectable 0.802 but it lagged behind DBSCAN

CATBoost Performance

CATBoost performed third best, and the best model was trained on data clustered using DBSCAN.

- PRAUC (test): 0.796

K Means best PRAUC (test) score was a respectable 0.787 but it lagged behind DBSCAN, however K Means result was much closer to the DBSCAN

Random Forest Performance

Random Forest, the only bagging algorithm in our ensemble methods, performed moderately, and the best model was trained on data clustered using DBSCAN.

- PRAUC (test): 0.73

K Means best PRAUC (test) score was a respectable 0.728, which was almost matching results clustered on DBSCAN.

	Algorithm and Rank			
Algorithm	**Best PRAUC (DBSCAN)**	**Best PRAUC (K-Means)**	**Difference**	**Rank**
XGBoost	**81.51%**	80.20%	**0.0131**	**1st**
AdaBoost	**81.36%**	80.20%	**0.0116**	**2nd**
CATBoost	**79.60%**	78.70%	**0.009**	**3rd**
Random Forest	**73.00%**	72.80%	**0.002**	**4th**

Figure 19: Ranked ensembling methods.

Predictive Anomaly Risk Flagging Using the 48-Hour Shifted Label

To explore whether the model could detect any early signals that might precede abnormal behaviour, we created a new flag called failure_future_48 by moving the engineered failure label forward by 48 hours. As noted in Chapter 3, this is not an anomaly detector, but a simple future-risk marker indicating that a possible anomaly may occur two days later. Because the relationship between present behaviour and

outcomes 48 hours ahead is weak, this experiment is intentionally exploratory: the goal is to see whether the models can perform meaningfully above the baseline anomaly rate of 10.76%.

The best model, XGBoost+DBSCAN, achieved a PRAUC(test) of 0.33291, representing roughly an improvement 3 times above baseline, although this came with a large overfitting gap, which is expected for such a long-horizon signal. Figure 20 summarizes the performance of the top models tested on the 48-hour shifted label.

		Algorithm Performance					
Algorithm	**Cluster**	**AUROC Train**	**AUROC Test**	**AUROC Gap**	**PRAUC Train**	**PRAUC Test**	**PRAUC Gap**
XGBoost	DBSCAN	0.994698	0.780958	0.21374	0.948471	0.316032	0.63244
XGBoost	K-Means	0.995086	0.77269	0.2224	0.952389	0.304567	0.64782
Random Forest	DBSCAN	0.987768	0.793922	0.19385	0.8905	0.312692	0.57781
Random Forest	K-Means	0.987211	0.800395	0.18682	0.884955	0.319236	0.56572

Figure 20: Summarized performance of top performing models.

Impact of Clustering Methods

Across all algorithms and sweeps DBSCAN outperformed K Means on all metrics with smaller overfitting gaps. Models trained on datasets that were clustered with DBSCAN showed higher PRAUC scores and smaller overfitting gaps. This was expected as K Means assumes spherical clusters while data patterns from PV plants are not linear where we have the morning ramp up and mid-day peak, which too change with seasons. DBSCAN is density-based and deals better with this non-linearity.

This pattern held even in the exploratory 48-hour risk-flag experiment, where DBSCAN continued to show slightly stronger generalization and higher PRAUC(test) values than K Means.

Summary Results:

The highest-performing model across all algorithms and sweeps was XGBoost + DBSCAN, achieving a PRAUC of 0.8151. AdaBoost + DBSCAN followed closely with a PRAUC of 0.8136. Tables of the Top five models, hyperparameter sweeps and top-performing configurations are listed in the Appendix.

In addition to these main results, the long-horizon 48-hour risk-flag experiment showed that the models could still perform above the baseline anomaly rate, although with expected overfitting due to the weak future signal.

4.2.3 Interpretation of Findings

4.2.3.1 Findings of LSTM for Predictive Maintenance As mentioned in the previous section, the data provided by MN8 Energy does not include labels indicating whether a sample represents a failure or normal operation. This is reasonable, as a drop in power does not always correspond to a true system failure, it may simply reflect short-term hardware glitches or communication interruptions. Due to this uncertainty, applying AI/ML-based survival analysis is not feasible. Instead, we focus on predicting future power output and using statistical measures as monitoring indicators to assess system performance.

As the DT continues to supply daily inputs to the LSTM Predictive Maintenance model, the average expected power output for the next two days is calculated and reported. In Figure 21, we plot the averages computed over the most recent 30 consecutive days. Each point represents the two-day predicted average, derived from the preceding 7 days of input data. This expected power level is continuously compared against a predefined threshold. If the newly predicted two-day average falls below this threshold, a maintenance notification is triggered so that an appropriate maintenance schedule can be planned.

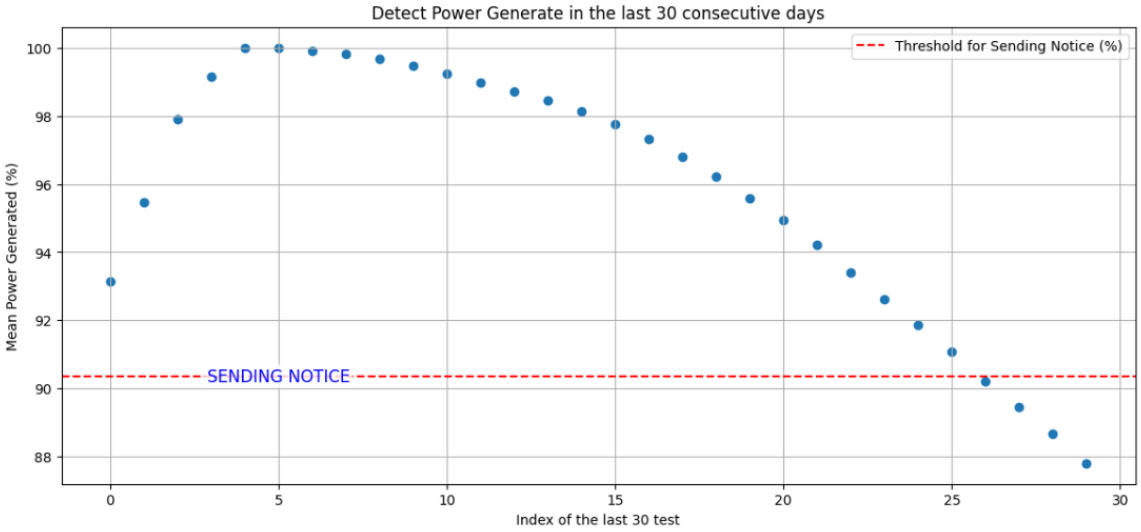

Figure 21: Monitoring Predict Continuous Average Power Output For Predicting System Failure.

4.2.3.2 Findings of Ensemble for Predictive Maintenance Boosting Vs Bagging

Boosting algorithms (XGBoost, AdaBoost, CATBoost) outperformed bagging algorithms (Random Forest). Boosting is better at picking up on small patterns and subtle signals associated with changes in behavior, while bagging averages many independent trees and does not adapt, as well to, or pick up on subtle changes in the signal. These results suggest that boosting methods are more useful for identifying unusual or unexpected patterns in output.

DBSCAN Vs K-Means

Across all algorithms DBSCAN out performed K Means, particularly in boosting algorithms. We expected this to be the case as power generation throughout the day and across seasons do not exhibit a linear pattern. Instead the position of the sun at different times of the day creates uneven patterns in data. The additional complexity of weather patterns and seasons add to the unevenness of the data.

K Means, which is centroid based, assumes roughly even sized clusters which is not the case in solar power output. Whereas DBSCAN, groups data points based on density and can deal with irregular shapes. It also can identify data points that do not belong to any cluster which is useful in identifying unusual behavior

Reliability of PRAUC Scores

High PRAUC scores such as the 0.81 for XGBoost with DBSCAN show that the models are able to identify unusual patterns and behavior in the dataset. Since the dataset did not contain explicit failure labels, these results also support the validity of our engineered failure labels.

Generalization and Overfitting Behavior

The best models show a small gap between training and test score, which means that the models are not simply memorizing noise but actually learning patterns. This is encouraging for real world use.

Algorithm	Cluster	AUROC Train	AUROC Test	AUROC Gap	PRAUC Train	PRAUC Test	PRAUC Gap
XGBoost	K-Means	0.989	0.976	**0.013**	0.917	0.802	**0.115**
XGBoost	DBSCAN	0.993	0.977	**0.016**	0.952	0.815	**0.137**
Random Forest	K-Means	0.984	0.971	**0.013**	0.895	0.728	**0.167**
Random Forest	DBSCAN	0.987	0.971	**0.016**	0.917	0.73	**0.187**
AdaBoost	K-Means	0.984	0.976	**0.008**	0.89	0.802	**0.088**
AdaBoost	DBSCAN	0.983	0.977	**0.006**	0.887	0.814	**0.073**
CATBoost	K-Means	0.983	0.972	**0.011**	0.91	0.787	**0.123**
CATBoost	DBSCAN	0.983	0.973	**0.01**	0.913	0.796	**0.117**

Figure 22: Generelazition of ensemble models across data set.

Findings from the 48-Hour Risk-Flag Experiment

The exploratory 48-hour risk-flag experiment revealed that models can capture some early signals associated with possible abnormal behavior, even when the predictive horizon is pushed far beyond what is typically studied. Although overall performance was modest, the best model, XGBoost+DBSCAN, achieved a PRAUC(test) of 0.33291, which is approximately three times above the baseline anomaly rate of 10.76%. This suggests that even faint long-horizon patterns may be detectable.

As expected, the models exhibited substantial overfitting due to the weak relationship between present conditions and outcomes two days later. However, the fact that the models performed meaningfully above baseline indicates that the engineered failure_future_48 label carries useful information and that the ensemble methods can extract weak early-warning signals. DBSCAN again showed slightly better stability and generalization than K-Means, aligning with the results from the main predictive-maintenance experiments.

4.2.4 Discussion in the Context of Existing Literature

Our ensemble-based predictive maintenance models draw inspiration from existing work in the PV forecasting literature. We originally considered the approach of Liu & Sun (2019) for anomaly detection, but their insight that structuring data with a clustering algorithm before modeling can strengthen predictive signals was well suited for predictive maintenance. Unlike their use of PCA and K Means for short term regression forecasting, we apply both DSCAN and K Means clustering as inputs to classification algorithms aimed at detecting failure-related behavior. Finally, to remain aligned with our initial interest in early anomaly indication, we also explore primarily for academic curiosity whether any meaningful early-warning patterns can be detected 48 hours in advance. We want to point out that because the signal is shifted so far ahead, this is not anomaly detection, but rather an anomaly-risk flagging experiment, marking possible future risk rather than current abnormal behavior

4.3 Analysis of Subproblem 2: Survival Analysis

4.3.1 EDA Relevant to Survival Analysis

4.3.1.1 EDA Relevant to Kaplan-Meier Function The Kaplan–Meier function categorizes failures based on a threshold defined by the user. As described in the previous section, the algorithm calculates both the number of observed failures and the time until each failure occurs. The resulting time-to-event distribution (see Figure 23) illustrates how failures (events) are spread over time. Based on the current data, the mean time to failure is estimated to be more than 3 days.

```
=== Time-to-Event Summary ===
Total observations: 153
Number of events (failures): 1
Number of censored observations: 8
Mean time to event: 3.37 days
Median time to event: 2.00 days
Proportion of events: 0.65%
Proportion censored: 5.23%
```

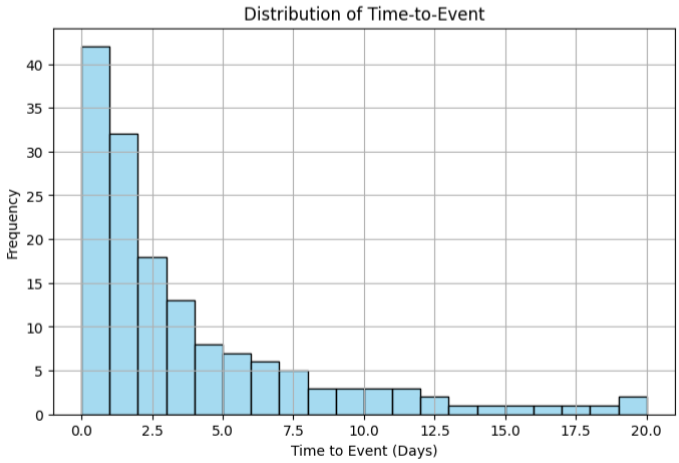

Figure 23: Time-to-event distribution.

The pie chart (see Figure 24) illustrates the proportion of devices that experienced a failure versus those that were censored (still operational at the end of the observation period). The current chart shows a high censoring rate, indicating that most devices are not expected to fail within the 2-day observation window.

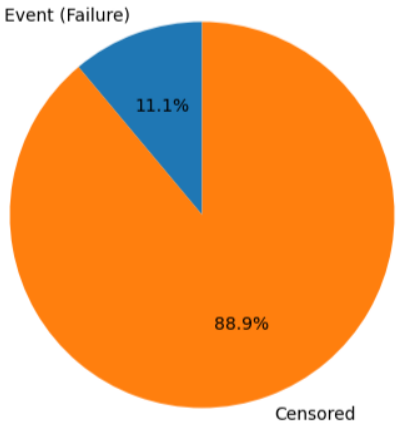

Figure 24: Proportion of devices that experienced a failure versus those that were censored.

4.3.1.2 EDA Relevant to COX PH Function A similar chart (see Figure 25) is presented for the Cox Proportional Hazards (Cox-PH) model. The status indicates that 68% of events (137 counts) are predicted to occur, while 32% (64 censored) of observations are censored. Using the full AC-related data as features, the Cox-PH model suggests that 68% of the systems could potentially fail within the next 2 days. While this serves as a warning, the data also show that the majority of systems remain under observation within the expected failure estimation range, providing a useful risk-monitoring perspective.

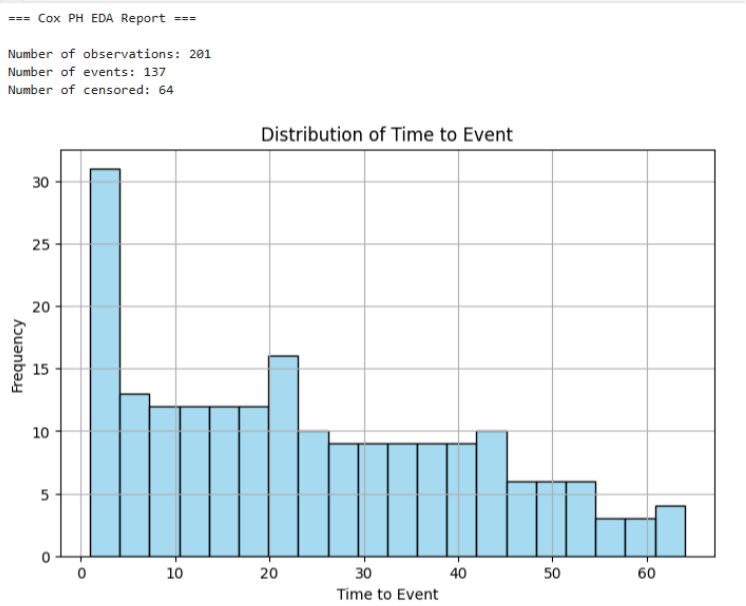

Figure 25: Time-to-event distribution.

4.3.2 Model Performance and Results

4.3.2.1 Kaplan-Meier Function Performance and Results The Kaplan–Meier function results indicate a high system survival probability, with over 70% likelihood of survival within the next 2 days. This suggests that the system is healthy and does not require immediate maintenance. It is important to note that this assessment is based solely on a single feature, AC_Power.Measured, and the predictions are derived from DT outputs. Historical data from the previous 7 days, as provided by the DT, are used as input for this function to estimate the probability of failure over the 2-day horizon (see Figure 26).

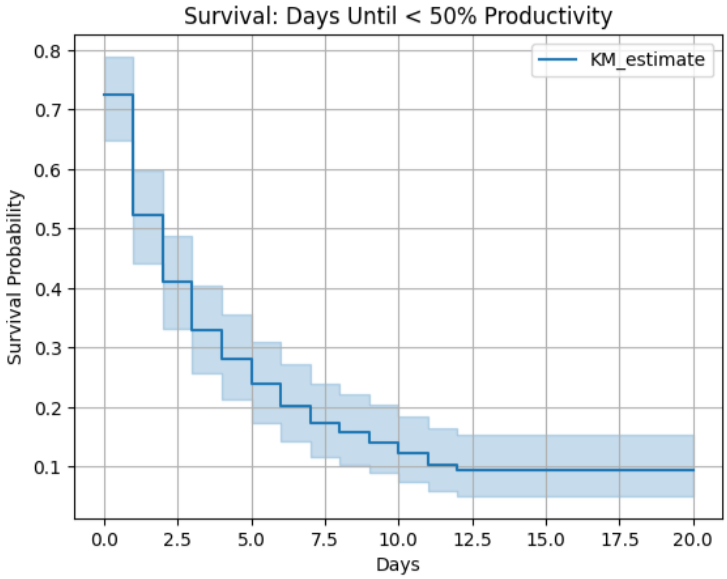

Figure 26: System survival probablity within the next 2 days.

4.3.2.2 COX PH Function Performance and Results The Cox-PH uses all the output of AC from Predictive Maintenance to use for predicting the failure within the next two days. The total observation is 201 in which there are 48 (failures) events. The number of sensors is 154 leading to about 13 days. There is a high probability of the system to have no issues in 24 hours or 2 days. However, 80% predictive failure might occur after 4 days (48 hours).

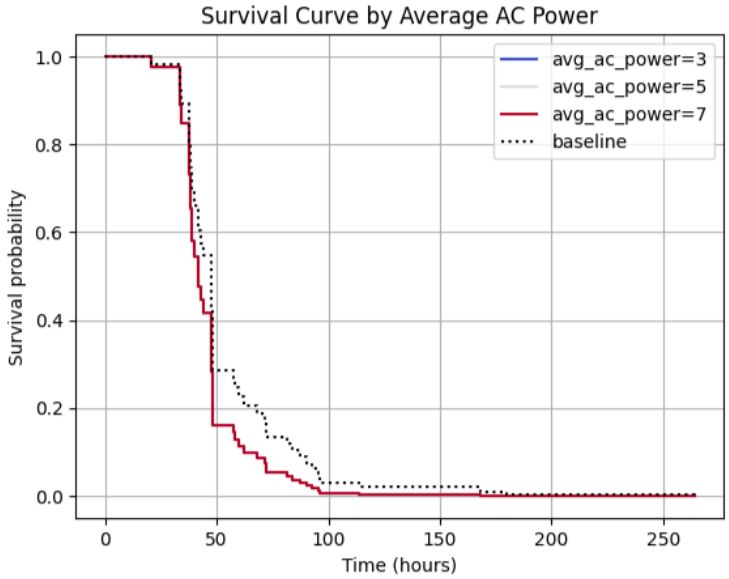

Figure 27: System survival probablity within the next 2 days.

4.3.3 Interpretation of Findings

The Cox-PH Hazard ratio given : avg_ac_power 0.999425, avg_current_max 1.005023, avg_power_limit 0.965808. This shows that all three covariates have HRs close to 1, so individually, they have weak effects on component hazard. The avg_power_limit has the strongest effect among these three (HR = 0.9658 → ~3.4% hazard reduction per unit).

The Cox-PH curve shows an early decline, suggesting failures occur quickly.

Kaplan–Meier Survival Analysis Interpretation:

- Median survival time: 2.0 days
- Time when survival probability drops to 10%: 12.0 days
- Number of observed failures: 133
- Number of censored cases: 20

Half of the units (or days) fail by around 2.0 days, and 10% of units remain healthy at day 12.0. For this inverter, the Kaplan-Meier indicates the system needs to be looked at around 12 days from the current time.

4.4 Analysis of Subproblem 3: Forecasting Inverter Output

4.4.1 EDA Relevant to Forecasting

4.4.1.1 EDA of Forecasting with Multivariate Time Series In the EDA section for this specific sub problem due to large volume of data and less compute resources we had focused majorly on the initial extract shared by the MN8 team. We mainly focused on the "Inverter" devices data. Our initial process involved examining the relationships between the different operational metrics using a correlation matrix. This helped identify features that move together, which could indicate functional dependencies or potential stressor variables. Key observations from the correlation analysis include:

- High Correlation between Power Metrics: There is an extremely high positive correlation between the "AC_Power" and "DC_Power", suggesting that these metrics are almost perfectly linearly related, as expected in an inverter system. This implies that one might be redundant if both are used directly as features, or they could serve as consistency checks.
- Voltage and Current Relationships: Similarly, various voltage and current measurements within the AC and DC systems eg. DC_Voltage_N and DC_Voltage_P, AC_Voltage_CA and AC_Voltage_BC showed very strong positive correlations. DC_Power and DC_Current also exhibit a high positive correlation.
- Temperature and Opertational Metrics: Several temperature-related metrics, such as Status_Internal_Temp, Status_Mod_Max_Temp and Status_IGBT_Max_Temp, are positively correlated with each other and with power/current metrics. For instance, Status_Internal_Temp had a high correlation with Status_Mod_Max_Temp. There are also notable correlations between Status_IGBT_Max_Temp and power metrics AC_Power, indicating that temperature increases with higher power generation, which is a crucial temperature/load relationship. These relationships could highlight potential thermal stress on components.
- Relationships with Setpoints and Power Factor: AC_Power_Limit_Setpoint shows significant positive correlations with power metrics(AC_Power) and voltage metrics(DC_Voltage_N). Power_Factor also showed strong correlations with power and current metrics. These setpoints and power factor can act as important "stressor variables" or control parameters. Their inclusion as features can help models understand how operational adjustments or grid conditions influence power output.
- Negative Correlations: Some metrics, such as AC_Power_Limit_Setpoint with Status_Word and DC_Voltage_P with Status_Word exhibit negative correlation these inverse relationships can be critical for identifying operational states or error conditions where a Status_Word changes significantly as operational parameters decrease or increase. While Status_Word might be more relevant for classification tasks, its inverse correlation with power metrics highlights periods of non-optimal operation that a forecasting model should ideally account for.

While this correlation analysis provides initial insights into feature behaviour, a more direct identification of 'degradation indicators' and 'stressor variables' would require domain specific knowledge and potentially feature engineering based on these observed relationships.

4.4.1.2 EDA of Forecasting with Decomposition Algorithm

In the literature, the algorithm divides the time-series data into two components: seasonal and trend. Predictions are performed separately on each component, and the results are combined at the end. To implement this approach, we choose the data of ONE inverter for the whole dataset of the first 7 months of 2025. At first, the data is decomposed into seasonal and trend portions. Using the same single-inverter dataset employed in the previous survival analysis, we apply a decomposition function to extract the seasonal and trend data. The heat map of the seasonal component is shown in Figure 28.

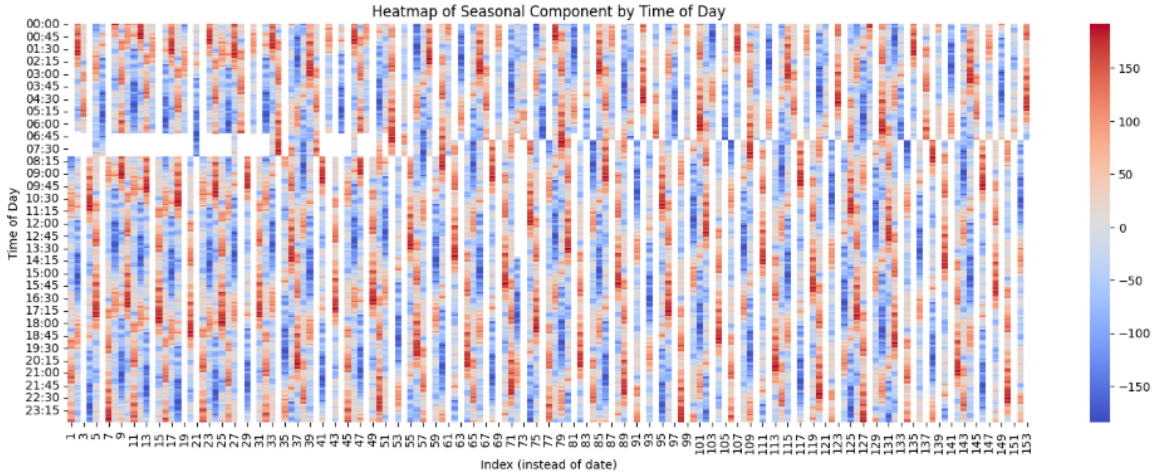

Figure 28: Heatmap of seasonal component by time of day.

There are some missing values in the seasonal data. For each day, the data transition from high (hot) to low (cold) values, representing periods of power generation and downtime. Some days have no power generation at all. The seasonal power generation for each hour in the dataset is illustrated in Figure 29.

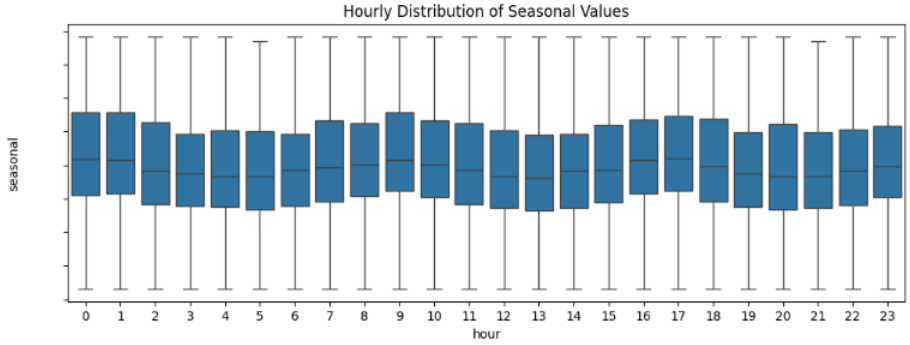

Figure 29: Hourly distribution of seasonal values.

For each hour of the seasonal data, the values fluctuate, reaching their highest levels around noon. Note that this represents an hourly summary across the entire 7-month dataset. The time axis may not start at 0 hours, but rather from noon of the previous day to noon of the current day. Nighttime is reflected on the right side of the chart, where power generation decreases to near zero. We can observe the oscillations in the data in the hourly chart of the seasonal dataset (see Figure 30).

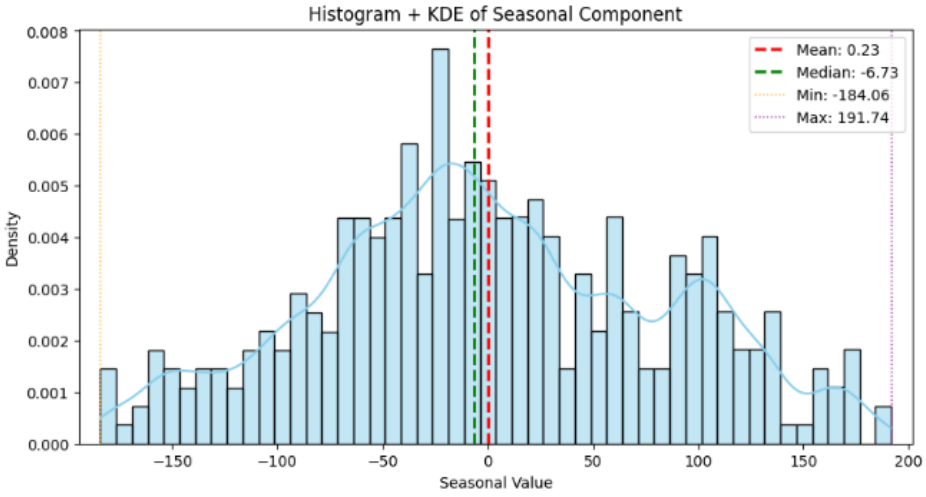

Figure 30: Histogram and Kernel Density Estimation of seasonal component.

When we plot the KDE (Kernel Density Estimate) of the seasonal component for the entire dataset, we can see that most of the values are concentrated near the trend (around 0). There are also periods when the seasonal data falls below the trend, as well as periods when it rises above the trend.

Similarly, for the trend data, the hourly plot shows that the trend values increase during certain hours of the day, as illustrated in the figure below. Across the entire dataset, this trend reflects the time when the sun is at its peak relative to the sensors, resulting in the highest energy output from the solar panels. The hourly distribution chart of the trend data (see Figure 31) shows that power generation rises during certain hours and decreases during others throughout the day.

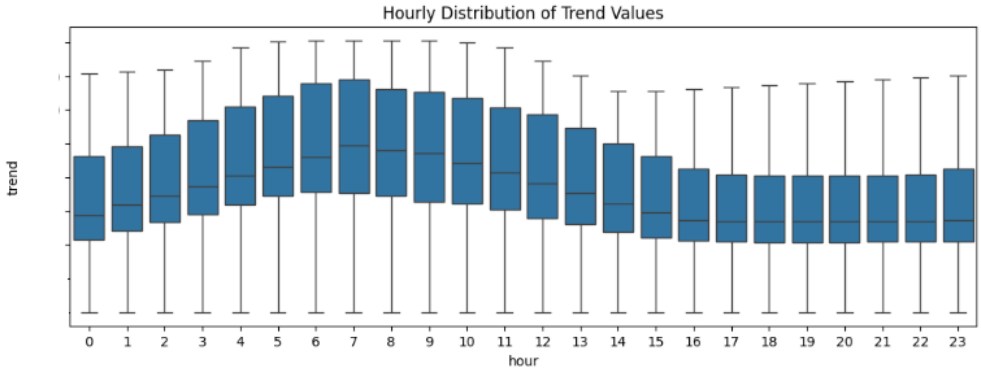

Figure 31: Hourly distribution of trend values.

When we map the trend component back to the original data (see Figure 32), it highlights periods when the inverter produces stable power, times when it drops to zero, and periods when the system performs well. This raises an interesting possibility: could the improvements in power output following a drop indicate that the solar panels were replaced or repaired, leading to higher performance in the subsequent months?

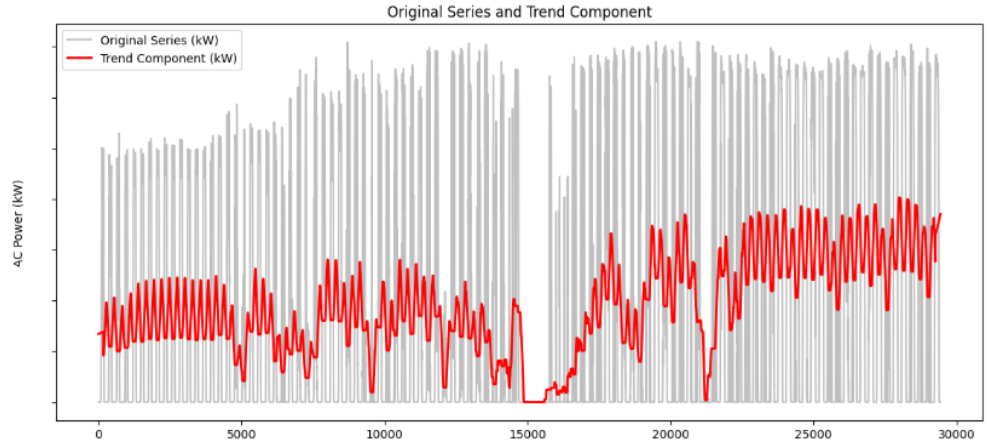

Figure 32: Mapped trend component to original data.

4.4.2 Model Performance and Results

4.4.2.1 Model Performance of Multivariate Time Series Forecasting We implemented a diverse set of forecasting models, each trained and evaluated independently. A common strategy we used for all the models involved training and testing on the same data across all the models. The metrics we used to evaluate these models are MAE(Mean Absolute Error), RMSE(Root Mean Squared Error), MAPE(Mean Average Precision Error) and R2(R squared). In the final version of the code we are able to produce the forecast for different device types in the PV plant. Our initial goal was to focus on the "Inverter" type but our reach goal was to develop a pipeline that would help forecast the power performance of all the devices. With the help of AI tools (Gemini and ChatGPT) we were able to expand our code and make it more general for it to be used for all the different device types and achieve our reach goal.

4.4.2.1.1 Baseline Model

- Data Handling: The Baseline model directly uses the aggregated time series data, either at a site

level or segmented by device type, derived from a large csv file of PV plant metrics. This preprocessing involves robust handling of missing values using forward-fill and backward-fill, ensuring a complete time series. Crucially, feature engineering is applied to create cyclical time of the day features (sine/cosine transformations of hour and minute) and wind direction features (sine/cosine of degrees). These engineered features are vital for capturing the inherent periodicities and directional influences common in solar power generation. Furthermore, a daylight filtering step ensures that the analysis and modeling focus only on periods of active power generation, where AC power is greater than zero or irradiance is detected. This prevents the model from being diluted by nighttime data, which would skew performance metrics and overall model learning.

- Approach: The baseline model employs a decomposition strategy, splitting the time series into two distinct, more manageable components:
  - Trend Component: This aspect primarily focused the long-term, underlying movement of AC power. It's modeled using LinearRegression on a smoothed version of the target, specifically its rolling mean. The choice of a rolling mean, with a window size, is designed to effectively smooth out short-term fluctuations and noise, thereby isolating the foundational linear progression. This technique is valuable given the inherent daily and seasonal patterns observed in solar power generation during the EDA step, allowing the model to capture the consistent directional changes over time.
  - Stable Component: After the linear trend is accounted for the remaining variability, or residuals form the stable component. This part of the model is responsible for capturing the complex, non-linear relationships that the simple linear trend cannot. It utilizes either an XGBRegressor (a powerful gradient-boosted decision tree algorithm) or as a fallback, a GradientBoostingRegressor. This model is trained on the full dataset including some of the critical environmental factors like Irradiance_Poa, Irradiance_Ghi, T_Amb, Humidity, and operational metrics like Tracker_Angle, DC_Current, DC_Voltage. Crucially , engineered cyclical features such as tod_sin, tod_cos (for time of the day seasonality), and wind_dir_sin, wind_dir_cos (for wind directionality) are also incorporated.
- Implications: This baseline model serves as a fundamental benchmark. Its performance provides initial insights into how well a relatively straightforward, yet feature-rich, approach can capture the complexities of solar power generation. If this model yields satisfactory results, it suggests that the power generation patterns include significant linear trends and explainable non-linear relationships with exogenous variables.Conversely, if its performance is poor, it points towards the need for more sophisticated models capable of discerning highly intricate temporal dependencies or uncaptured external influences. The inherent interpretability of its decomposed components (trend vs feature driven residuals) makes it easier to diagnose potential issues or identify areas for targeted improvement in subsequent, more complex models.

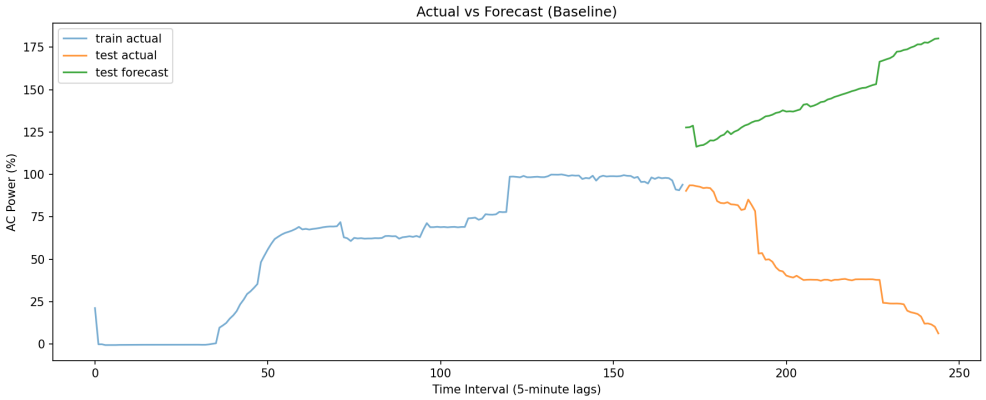

Figure 33: Baseline model performance.

4.4.2.1.2 Vector Autoregression Model (VAR)

- Data Handling: The VAR model utilizes a multivariate time series dataset constructed by combining the target AC_Power.Measured with a specific subset of highly correlated exogenous variables Irra-

diance_Poa.Measured, Irradiance_Ghi.Measured, DC_Current.Measured, DC_Voltage.Measured and T_Amb.Measured. This selection is grounded in the physical principles of solar generation, where irradiance and DC electrical parameters are direct precursors to AC Power output. The rationale is to leverage the interdependencies between these variables, allowing the model to capture how changes in one variable might propagate to affect the target variable over time. Missing values are handled similar to the baseline model implementation.

- Approach: The model employs the Vector Autoregression method from the statsmodels library, which treats every variable in the system as a function of the past values of all variables. The optimal lag order for the model is determined dynamically using the Akaike Information Criterion (AIC) with a specified maximum lag, ensuring a balance between model complexity and goodness of fit. Forecasting is executed recursively: the model predicts the next time step based on the historical window, effectively using the past trajectory of all included variables to forecast the future state of the AC power.
- Implications:The VAR model is particularly useful for identifying and exploiting dynamic, bi-directional relationships between system variables. If successful, it indicates that the historical variations in DC parameters and environmental factors provide significant predictive power for future AC power output beyond what a univariate model could offer. However, its linear nature means its may struggle with highly non-linear interactions or rapid shifts compared to non-linear models like LSTM. Its performance serves as a benchmark for determining the value of multivariate historical context.

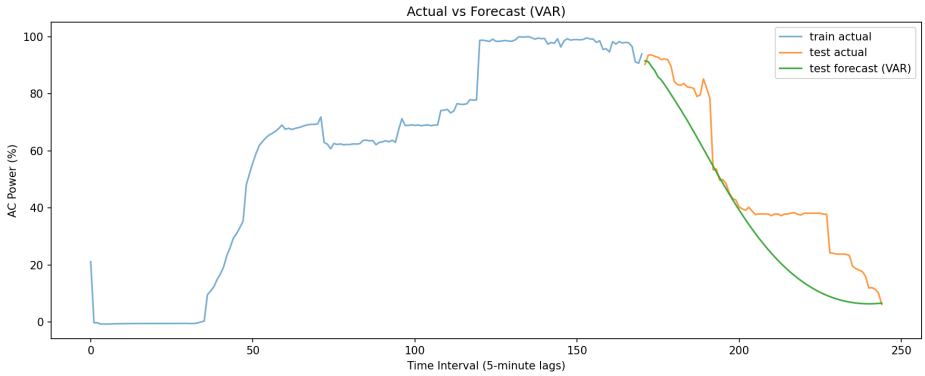

Figure 34: VAR model performance.

4.4.2.1.3 Long Short-Term Memory (LSTM)

- Data Handling: Most of the data handling in terms of features is very similar to the VAR model. One thing which was implemented to ensure stable and fast convergence during gradient descent. Furthermore, the data was transformed into sliding window sequences, creating the necessary temporal structure for the LSTM to learn dependencies from historical context.
- Approach: The model is implemented as a sequence-to-one regression neural network. It consists of an LSTM layer, which processes the input sequence to capture temporal dependencies and update its hidden state, followed by a fully connected linear layer that maps the final hidden state to a single continuous prediction (AC_Power). The model is trained using the Adam optimizer and Mean Squared Error loss function, iteratively adjusting weights to minimize the difference between predicted and actual power output. This architecture is specifically designed to model complex, non-linear dynamic behaviour that statistical models might miss.
- Implications: The LSTM's performance provides insight into the non-linearity and complexity of the underlying system. High accuracy suggests that the power generation process has significant temporal dependencies and non-linear interactions between the variables that the LSTM successfully encoded. Conversely, if the performance is similar to simpler models, it might indicate that the relationships are predominantly linear or the dataset size is insufficient for the deep learning model to generalize effectively. While powerful, the LSTM model offers less interpretability compared to baseline or VAR models regarding specific feature contributions.

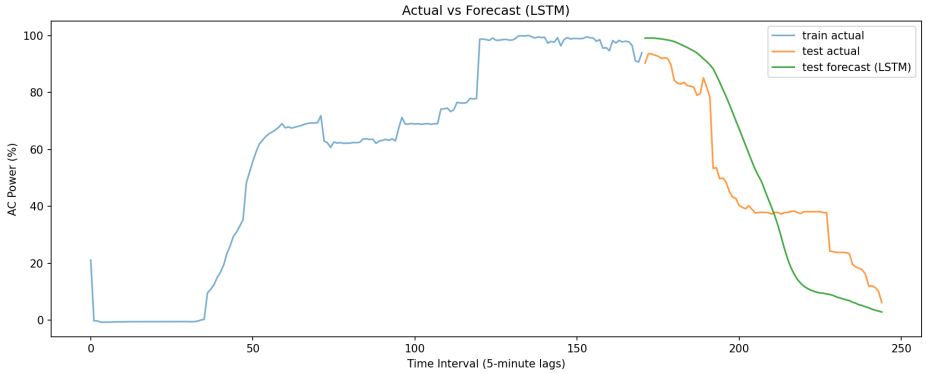

Figure 35: LSTM model performance.

4.4.2.1.4 Chronos Pipeline

- Data Handling: The Chronos model operates differently from the others by focusing primarily on the univariate target time series AC_Power.Measured or the fallback which is the DC. Here the Chronos model leverages the power of a foundational model pre-trained on a vast array of diverse time series data. The input data is split into a window, which provides the model with the recent historical trajectory of power generation, and a prediction. Here we have an assumption that the temporal patterns, seasonality, and distributions inherent in the target series itself are already learned by the model through its vast experience by being a pre-trained model and are sufficient for accurate forecasting without manual feature engineering.
- Approach: The model pipeline implemented here is using the chronos-forecasting library, specifically utilizing the amazon/chronos-t5-mini model (a transformer-based architecture). It functions as a generative model: it tokenizes the input time series values, processes them through the transformer network and autoregressively generates the future trajectory. The script draws multiple samples from the predictive distribution to capture uncertainty and uses the mean of these samples as the final deterministic forecast. This essentially represents a shift from the traditional statistical or supervised learning methods to a modern, "zero-shot" or "few-shot" learning paradigm.
- Implications: The use of Chronos introduces a state-of-the-art benchmark into the analysis. Strong performance here would indicate that the underlying temporal patterns of solar generation are robust and recognizable enough for a general purpose foundation model to predict effectively, even without specific environmental context. Conversely, if it underperforms compared to the multivariate models, it underscores the critical importance of exogenous variables (weather, sensor readings) in accurately forecasting solar power, which a univariate approach might miss. Like any deep learning model this model also offers limited interpretability regarding specific drivers of change.

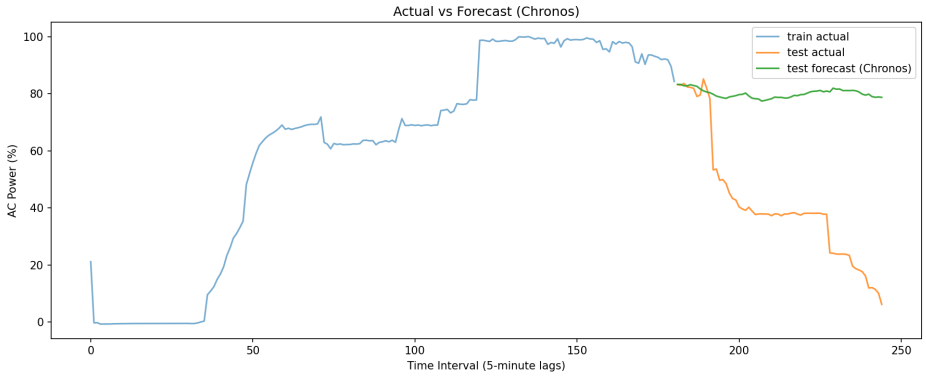

Figure 36: Chronos model performance.

4.4.2.1.5 Seasonal Autoregressive Integrated Moving Average (SARIMA) Model

- Data Handling: The SARIMA model leverages the target time series augmented with a robust set of exogenous variables to implement a SARIMAX framework. The feature set includes Irradiance_Poa.Measured, Irradiance_Ghi.Measured, Dc_Voltage.Measured, T_Amb.Measured, Humidity.Measured, Tracker_Angle.Measured, and the cyclical time features tod_sin and tod_cos. Missing values in these exogenous features are handled via forward and backward filling. The rationale for this selection is to ground the statistical model in the physical reality of power generation: solar irradiance and DC voltage are primary drivers, while environmental factors and time of day proxies help explain seasonal and daily variances that a pure autoregressive model might miss.
- Approach: The model utilizes the SARIMAX class, which extends the ARIMA framework to handle both seasonality and exogenous regressors. It is configured with a fixed non-seasonal order of (1,1,1) (AR = 1, I = 1, MA = 1) to handle local trends and autocorrelation, and a seasonal order of (1,0,1,12*24) (AR = 1, I = 0, MA = 1, Period = 288). The seasonal period of 288 explicitly assumes a minute level frequency, aiming to capture the repeating daily profile of solar generation. The model estimates parameters that best fit the historical data by minimizing the error term, simultaneously accounting for the linear influence of the provided exogenous variables.
- Implications: SARIMA offers high interpretability, as its parameters directly correspond to statistical properties like autocorrelation and seasonality. Good performance here would suggest that solar power data follows a structured, linear process driven by clear daily cycles and external inputs. However, if the seasonality order varies the model could suffer with issues in converging or generalizing. A failure to perform well compared to the baseline or LSTM would imply that the underlying system dynamics are too non-linear or complex for this classical statistical framework to capture effectively.

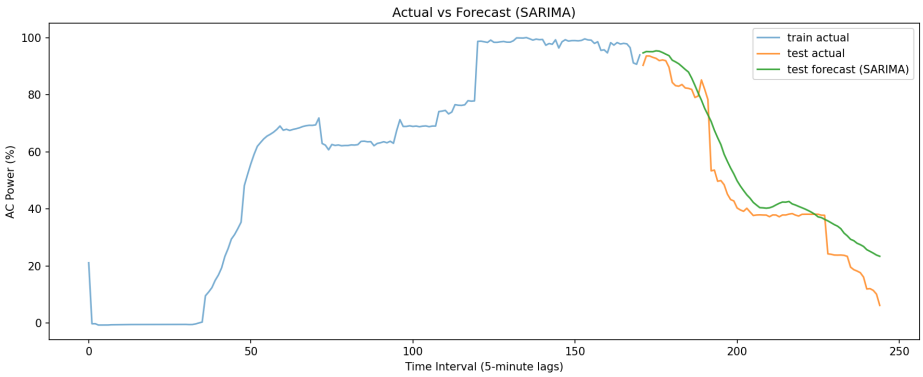

Figure 37: SARIMA model performance.

4.4.2.2 Model Performance of Forecasting with Decomposition Algorithm

We designed an LSTM model to predict the seasonal component and a Deep Neural Network (DNN) to predict the trend component. While the literature suggests using linear regression for trend prediction, this approach did not work with our dataset, as the trend shape varies for each day. Therefore, we opted for a DNN to capture the full variability of the trend data. The input for both models consists of the previous 7 days of solar panel data (directly from the system, not the DT). Following this approach, the LSTM produces a prediction closely matching the next day's seasonal values. The combined forecast from both models represents the expected power output of the inverter.

Figures 38, 39, and 40 show the results: the LSTM prediction based on 7 days of historical seasonal data, and the DNN prediction based on 7 days of historical trend data. The prediction output is the expected power generated tomorrow.

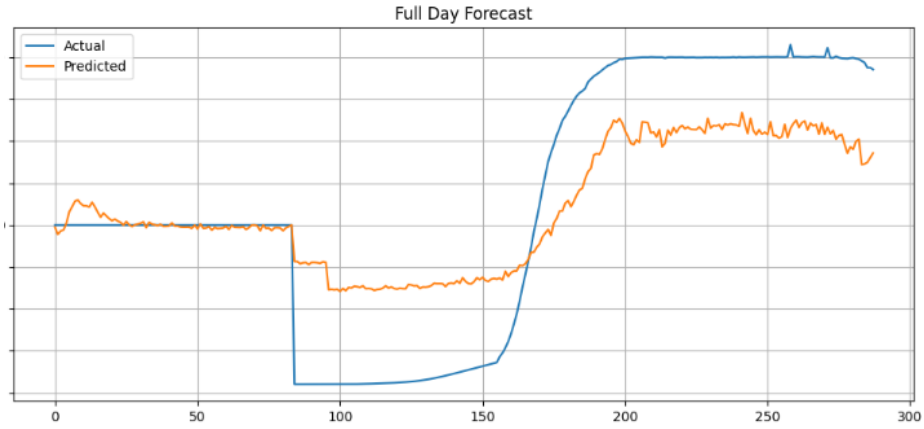

Figure 38: Seasonal prediction.

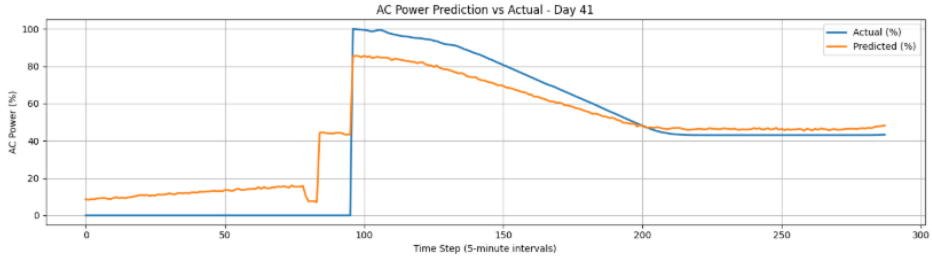

Figure 39: Trend prediction.

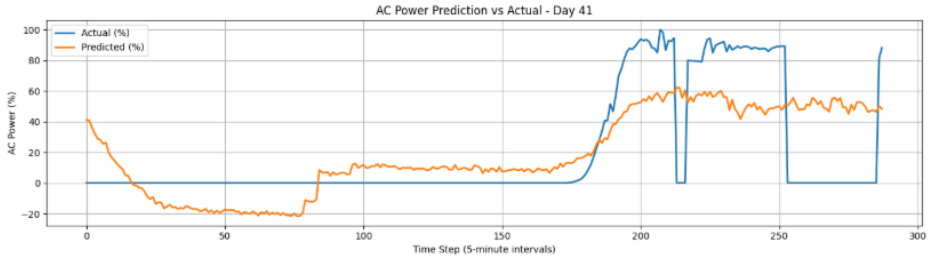

Figure 40: Combined Trend and Seasonal Predictions plotted with truth values.

With the data containing a lot of non-collected data spots and not valid numbers, the algorithm achieves good accuracy. Both RMSE and MAE values are relatively low. The modified algorithm gives on average the predictions fairly close to the true values. RMSE is higher than MAE indicates that large errors exist but are not extreme. This is due to the missing data; however, most predictions fall within a small error range. These metrics, taken alone, imply that the model is doing a reasonably good job capturing the general shape and magnitude of the target variable.

An R^2 show the models explains about 56% of the variance in the data. This shows almost half of the variability is not captured by the model. As mentioned above, the missing data beside missing consistently, the missing data and non-valid data occurred randomly on different days of the same dataset. In forecasting applications with noisy or non-linear signals (like solar power), an R^2 above 0.5 is common and opens room for improvement. The R^2 value indicates that the models capture some of the underlying structure, but not all the nuances or rapid changes.

Table 5: Model performance metrics.

Metric	Value
RMSE	0.173863010493861
MAE	0.153794851435234
MAPE	3042.90995136215%
R^2 Score	0.559003513333803

Figure 40 shows that the combined predictions of the trend and seasonal components align closely with the true data. This modified algorithm from the literature provides a reliable approach for forecasting the next day's power output. While the accuracy could be further improved, these results are useful for detecting potential anomalies and preventing sudden, unexpected stoppages of the inverter. By using this forecasting algorithm, an additional maintenance notification can be generated when necessary, enhancing the overall monitoring and preventive maintenance strategy.

4.4.3 Interpretation of Findings

model	MAE	RMSE	MAPE	R2
baseline	98.09824804494949	107.2337474376255	376.1424748594675	-16.050446974299785
var	12.091825680718712	14.603483032643778	33.562784818177064	0.6837826443128527
lstm	16.004358293086923	18.288043353722692	43.91264097768609	0.5040849516212894
chronos	38.58621814240472	43.497117567036405	164.10745075919775	-3.246073021682161
sarima	6.5221153070585505	7.8286744023486685	25.430994705068592	0.9091239872610138

Figure 41: Time-series forecasting performance.

4.4.3.1 Forecasting Time-Series

- Baseline: The baseline model, which decomposes the series into trend and stable components, performed poorly, as indicated by the high error metrics and extremely negative R2 score.This suggested it's not capturing the underlying patterns effectively.
- VAR: VAR showed significantly better performance compared to the baseline, achieving a reasonable R2 score. This indicates that considering the interdependencies between target and exogenous variables improved forecasting accuracy.
- LSTM: LSTM model performed decently, though slightly worse than VAR. I believe careful tuning and the more data we utilize with this model the better the accuracy will be.
- Chronos: This is a transformer-based model, which performed worse than VAR and LSTM but was similar to baseline. The performance of this model might be limited due to the limit of the data used on the model, a similar issue as the LSTM model.
- SARIMA: This model emerged as the best-performing model with the lowest error metrics and the highest R2 score. This suggests that the time series exhibits strong seasonality and autoregressive properties that SARIMA effectively captured, even with the exogenous variables.

metrics_summary_per_device_type_pct

device_type	model	MAE	RMSE	MAPE	R2
Combiner	baseline	119.5441551202510	129.8461737897840	25866.98303636990	-12.079604396611500
Combiner	var	67.44657960645380	76.23238693608480	15234.306439572900	-3.508326199479320
Combiner	lstm	18.11206103874710	22.270771750540700	3058.1622754701800	0.6152250157465110
Combiner	chronos	7.641045091848770	12.754713198941800	659.8822429204920	0.42183231749748600
Combiner	sarima	1426187.7087695400	3894497.871067490	429161121.7952680	-11766281563.692100
Inverter	baseline	121.6214507993270	132.6362396633300	8356553279.47011	-12.368513699407200
Inverter	var	65.60745168101520	75.24266570865910	4746875984.510950	-3.3021650777855000
Inverter	lstm	11.701060146782000	17.259535788377800	367368142.88345200	0.7736310036277880
Inverter	chronos	8.254390667574430	14.24730845287950	126284960.88243200	0.36365206046902000
Inverter	sarima	2758.235767101490	5209.7189957030200	268791067138.4560	-20623.68485796510
Inverter Module	baseline	123.2167495310860	133.7879191781010	8383811370.868720	-12.667700675161200
Inverter Module	var	52.263424489952	59.870298087028100	3695548069.5450700	-1.737061158934690
Inverter Module	lstm	18.479923321628900	25.145096947140600	758936835.8773430	0.5171988570018760
Inverter Module	chronos	7.524512864281860	13.093960333874900	100943597.82984200	0.4476183681284000
Inverter Module	sarima	47573.697617789500	110669.19099058000	4737841938288.880	-9352227.972613730

Figure 42: Summary metrics per device.

A key discovery was that Deep Learning (DL) models exhibited superior performance when data was isolated by device for forecasting. This success stems from the DL models' inherent ability to handle and learn patterns from data with missing values, an area where traditional statistical models typically struggle. Although we believe that in order to improve the models further we need more domain knowledge specific hyperparameter tuning on the models to achieve accurate predictions.

4.4.3.2 Forecasting with Decomposition Algorithm The two figures below (see Figure 43) show the trend data for two different days: one exhibits a linear trend but in varying regions with different shapes, while the other displays a non-linear trend across different periods. This demonstrates that the trend patterns in this dataset are more complex than what the original literature assumed. As a result, we adapted the algorithm to use a different AI/ML architecture, a Deep Neural Network (DNN), which provides a better fit for the trend data than the linear regression suggested in the original paper, at least for this dataset.

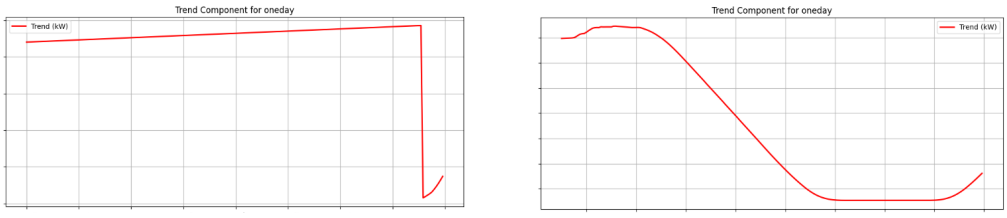

Figure 43: Different trend data.

4.4.4 Discussion in the Context of Existing Literature

As we implement the algorithm for Trend and Seasonal Decomposition described by Gürcan Kavakci et al. (2024, p. 6) using the current MN8 dataset, we observe that the method proposed in the literature does not perform well on this data. The extracted trend component does not exhibit global linearity;

rather, linear behavior only appears in certain segments. In addition, some portions of the trend are distinctly non-linear. Missing data also disrupt linearity, especially when linear interpolation is applied.

Furthermore, the linear pattern varies from day to day, and different inverters display different forms of non-linearity across various time intervals. Because of these characteristics, the use of Ordinary Least Squares (OLS) regression or a simple linear model,as recommended in the paper, is not suitable for this dataset. Instead, we replace the linear model with a Deep Neural Network (DNN) to better capture the behavior of all segments throughout the day.

The figure below shows the close prediction of the daily trend (288 data points) obtained using the Deep Neural Network

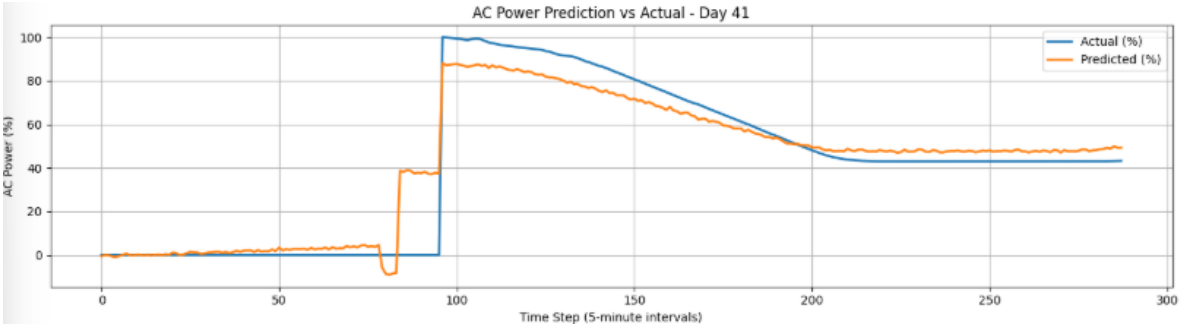

Figure 44: DNN prediction of daily trend.

4.5 Analysis of Subproblem 4: Anomaly Detection

Anomaly detection is something we experimented with as we wanted to get an insight on abnormal behaviour of the systems. Anomalies can affect our project's overall performance significantly. As early detection can lead to better response from the system.

4.5.1 Descriptive Patterns

- Efficiency Distributions: The OutlierDetector class was designed to compute regime-aware efficiencies by modeling temperature-dependent apparent power rating and active power rating.
- Temperature bins: The methods within the class were implemented to analyze the inverter performance as a function of temperature, deriving rating curves based on binned temperature data.
- Curtailment regimes: The calculation of active power rating factored in curtailment via alpha and allowing for a more accurate assessment of expected power output under varying grid conditions.
- Robust-z score landscape: Robust-Z scores were calculated across key metrics. Specifically, AC_Power.Measured was analyzed for broad outliers, and the performance_ratio was subjected to robust-Z analysis to detect deviations in system efficiency.

4.5.2 Anomaly Detection Framework Applied

- Threshold analysis: The anomaly detection framework combined multiple criteria:
  - Robust-A score on AC_Power.Measured with a sensitive z_thresh of 1.3
  - A binary flag for instances where DC input was observed to be greater than the AC output.
  - Robust-Z score on the calculated performance_ratio (The Performance Ratio is a crucial metric used in solar PV systems to evaluate their overall quality and efficiency taking into account all losses from the panel to the point of measurement. Unlike simply looking at AC power output, PR normalizes the output by the available solar irradiance, providing a more accurate picture of how well a system is performing relative to its potential.) with a z_thresh of 1.5
- Anomaly clusters: An overall anomaly flag was created, marking an observation as anomalous if it met all three of the above conditions. This method creates a clustered view of anomalies, requiring consensus across different detection techniques.

- Temporal patterns: Anomalies were visualized over a linear time interval scale, allowing the observation of temporal patterns. This approach highlights when anomalies occur relative to each other over a generalized timeline.

4.5.3 Interpretation of Findings

- Operational significance: The overall anomaly flag provides a robust indicator of periods requiring operational review. These anomalies suggest significant deviations from expected behavior, which could stem from sensor errors, inverter malfunctions, or external factors not accounted for in normal operation.
- Possible inverter states: Detected anomalies can point to various inverter states:
  - DC > AC: The DC input greater than AC output highlights potential data errors or highly unusual operational physics.
  - System under performance/over performance: Outliers in the performance ratio can indicate periods where the inverter is not converting solar energy to AC power as efficiently as expected, or conversely, is reporting unusually high efficiency.
  - Unexpected power output: Deviation in AC_Power.Measured can signal faults or unusual load conditions.
- Mismatch detection: The combination of AC_Power.Measured, DC_Current.Measured and performance_ratio allows for effective detection of mismatches within the inverter's energy conversion process. The DC>AC flag essentially targets one type of physical inconsistency.
- False Positives/ Negatives: The chosen z_thresh values (1.3 and 1.5) were selected for sensitive detection. While this ensures that most genuine anomalies are caught, it might lead to a higher rate of false positives compared to a less sensitive threshold. Further refinement of thresholds or the integration of more domain knowledge could help optimize the balance between false positives and false negatives.

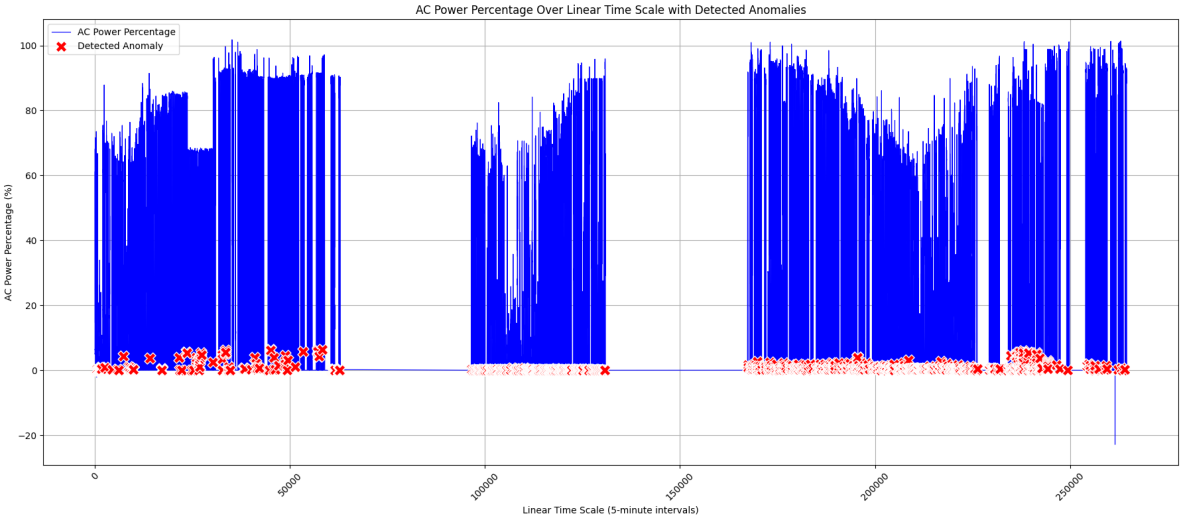

Figure 45: Detected anomalies across multiple days, displayed on a linear timescale.

4.6 Analysis of Subproblem 5: Digital Twin

This section presents the empirical results of the DT, which estimates the expected active AC power for each inverter and serves as the benchmark for anomaly detection. We evaluate the DT using the held-out test set and report performance across preprocessing choices, feature construction, and model variants. The following subsections summarise the data preparation, modelling outcomes, and key patterns observed in the DT behavior.

4.6.1 Data Preprocessing

We applied a two-stage outlier-masking pipeline consisting of (i) broad outlier masking and (ii) regime-aware masking. The broad stage used a robust-Z threshold of 3.5 and removed approximately 0-3% of values per inverter, with most removals concentrated during daylight periods and coinciding with high-POA intervals. This behavior is expected: daylight hours exhibit both higher variability and higher absolute magnitudes, making them more susceptible to large deviations when standardised. The distribution of percent change in null values across the time-of-day axis is shown in **Appendix F: Digital Twin Broad Mask Distribution**.

The second stage, regime-aware masking, applied a more stringent criterion: the 99.9th percentile of the absolute robust-Z score within each operational regime. This removed an additional 0-7% of values per inverter. Notably, the resulting null-distribution pattern is inverted compared with the broad mask: instead of concentrating removals during daylight periods, the regime-aware filter preferentially removed values in non-daylight (low irradiance or overnight) conditions. This behavior reflects how regime stratification isolates local behaviors: in low-variance night-time regimes, even small deviations can produce large robust-Z magnitudes, whereas daytime regimes tolerate greater variability before being flagged as anomalous. This inverted pattern is visible in **Appendix F: Digital Twin Regime Mask Distribution**.

Together, the two masking stages reveal complementary behaviors: the broad mask captures high-magnitude deviations during active operating periods, while the regime-aware mask identifies smaller but contextually inconsistent deviations during low-activity periods. This distinction is important for downstream modelling because it ensures that the null structure introduced by masking is not unintentionally biased toward a specific operational regime.

4.6.2 Autocorrelation

Autocorrelation functions (ACF) are computed for all candidate variables across the inverter fleet, and behavior is summarised using the median first-insignificant lag per metric. This approach provides a robust, project-wide view of temporal memory. Only the positive-memory horizon is considered when assessing predictive relevance. Negative autocorrelation observed at medium and long lags is attributed to daily cycles and inverter control oscillations rather than meaningful statistical dependence. Full ACF results are provided in **Appendix G: Autocorrelation Results**.

Three characteristic temporal-memory classes are identified across metrics. Short-memory variables decorrelate within a few hours and are dominated by rapidly changing physical drivers such as irradiance, temperature, inverter thermal response, and cloud-induced ramping. Medium-memory variables retain correlation over one to three weeks, reflecting slower system-level processes including grid-voltage regulation, transformer thermal inertia, operational setpoint drift, and weekly load patterns. Long-memory variables decorrelate only after several weeks, primarily due to seasonal cycles and slow inverter ageing effects.

These patterns are consistent with expected physical behavior. POA, temperature, and AC/DC power are governed by fast external forcing, resulting in lags beyond a few hours lacking predictive value. Grid frequency decorrelates within minutes due to rapid system balancing actions. In contrast, grid-side voltages exhibit medium-term drift and mean reversion, indicating that they encode broader system conditions relevant for stabilising baseline estimates. Internal temperatures and reactive-power behavior evolve slowly across seasons, offering contextual information about thermal loading and long-duration operational shifts.

For modelling purposes, these findings inform the incorporation of temporal information. Short-memory variables generally do not benefit from lagged features, as additional lags increase dimensionality without contributing meaningful signal. Medium-memory variables are better represented through rolling statistics or drift-based features rather than raw time-shifted values. Long-memory variables are most effectively captured using seasonal encodings or slow-varying trend indicators instead of explicit lag structures. These insights inform the design of rolling means and time-based feature engineering.

4.6.3 Cross-correlation

The cross-correlation functions (CCF) between each candidate metric and active AC power are computed to evaluate their direct temporal association and to determine whether any variables lead or lag the target. Full results are provided in **Appendix G: Cross-Correlation Results**. Across inverters, the peak correlations for the strongest predictors consistently occur at or near lag 0, indicating that these variables move contemporaneously with AC power and reflect the same underlying physical drivers.

Three empirical groupings are identified. Strong predictors (correlation > 0.7) include DC current, DC power, median POA, and IGBT maximum temperature. These metrics track the electrical and irradiance conditions that directly determine instantaneous inverter output, and their strong contemporaneous correlation aligns with operational physics. Moderate predictors (correlation > 0.3) include internal temperature and power factor. Internal temperature reflects slower thermal loading effects, often lagging irradiance changes by several minutes, while power factor captures control actions that adjust reactive power and therefore demonstrates moderate but meaningful coupling with real-power output. The remaining metrics fall into a weak-predictor group (correlations between –0.2 and 0.2), indicating limited contemporaneous explanatory value. However, this does not exclude them from consideration, as some may still contribute indirectly through interaction terms or through the engineered features introduced during sequential selection.

The CCF analysis does not reveal any practically useful leading indicators for active power, as strong predictors peak at lag 0. This suggests that real-time irradiance and electrical loading dominate system behavior. These findings reinforce the modelling decision to prioritise contemporaneous features and engineered summaries rather than large lag windows. The CCF results therefore serve as an empirical baseline that supports the sequential feature-selection process and frames expectations for model performance.

4.6.4 Feature Selection

Using an XGBoost model, we perform forward sequential feature selection by adding intra-family variables first and evaluating predictive contribution using validation R^2. We begin with the DC-side variables. Measured DC current alone achieves a high R^2 of approximately 0.99 for both daytime and nighttime models, indicating that the majority of variation in active power is explained solely by DC current. Adding DC power reduces performance; which is expected, as DC current is algebraically embedded within DC power, making the two features strongly collinear and therefore harmful to tree-based model stability. The same effect is observed for median POA, whose contribution is largely redundant once DC current is known.

Per-unit DC bus voltage yields a small improvement in daytime R^2, suggesting that internal electrical conditions (e.g., bus regulation and MPPT behavior) encode additional non-redundant information during generation hours. No improvement is observed in the nighttime model, which is consistent with the inverter being idle: DC-side electrical dynamics are effectively static, and bus voltage carries little predictive value. Rolling means of DC current and DC power improve predictive performance for the daytime model, indicating that windowed smoothing reduces high-frequency noise in the raw signals (particularly in DC power) and helps the model capture physically meaningful trends. Neither rolling feature improves nighttime performance, again consistent with the absence of DC generation.

Moving to the grid-side features, only power factor and the temperature-dependent rated active power contribute positively to daytime model performance. This suggests that curtailment events and grid-imposed reactive-power behavior influence real-power production enough to justify their inclusion. Frequency, per-unit line-to-line voltage, and rolling means of power factor and frequency provide no benefit. This is expected because these variables remain tightly regulated by grid constraints and exhibit limited within-device variability, making them weak predictors of real-power output. For nighttime conditions, none of the grid-side variables improve model performance; with real power near zero and governed primarily by standby behavior, grid-side variations have minimal explanatory value.

Among the thermal features, none improve R^2 for either daytime or nighttime models. This is expected for nighttime operation, as thermal derating does not occur and cooler temperatures do not impact efficiency when the inverter is not converting power. For the daytime model, the lack of improvement suggests that temperature-dependent effects are already fully captured through the temperature-dependent

rated active power feature included earlier in the selection sequence; adding raw or transformed thermal measurements therefore provides no additional predictive benefit.

The final feature sets selected for the ML models are shown in Table 6.

Table 6: Final feature set for ML models.

Model	Features
Daytime Model	Measured DC current; per-unit DC bus voltage; 90-minute rolling means of DC current and DC power; temperature-dependent rated active power
Nighttime Model	Measured DC current

These selections align closely with the earlier cross-correlation analysis: DC current consistently emerged as the strongest predictor; while proxies for other strong predictors, DC power (via its rolling mean) and thermal effects (captured indirectly through the temperature-dependent rated active power), also contributed meaningfully during daytime operation. The nighttime model retains only DC current, reflecting the absence of DC-side or thermal dynamics when the inverter is not generating.

4.6.5 Baseline Model Performance

The baseline forecasting models (persistence, Moving Average (MA), and Exponential Moving Average (EMA)) provide a simple reference point for evaluating more advanced models.

Table 7: Baseline model performance.

Model	R^2	NRMSE
Persistence	0.8297	0.1109
Moving Average	0.2498	0.2327
Exponential Moving Average	0.8308	0.1105

From Table 7, the persistence model achieves relatively strong performance ($R^2 \approx 0.83$), indicating that AC power is highly autocorrelated at 5-minute intervals. This reflects the physical smoothness of inverter power output under typical operating conditions, where changes between adjacent timesteps are gradual.

In contrast, the MA model performs poorly across both metrics. With a 79-point window (~6.5 hours, motivated by the perviously conducted autocorrelation analysis), the MA heavily smooths the signal, causing it to react too slowly to rapid irradiance-driven changes such as cloud transients. The resulting predictions lag behind the true dynamics, yielding a much lower R^2 and substantially higher NRMSE.

The EMA model performs best among the baselines. Although its performance is similar to persistence, the slightly higher R^2 and lower NRMSE arise because the EMA (with a 1/79 smoothing factor) captures longer-term trends while retaining modest responsiveness to recent changes. This balance allows the EMA to track the gradual drift in AC power more effectively than both the overly rigid MA and the strictly one-step persistence baseline. Overall, these baselines show that short-horizon AC power prediction is dominated by strong temporal dependence, and that models capable of adapting smoothly to recent history (like the EMA) provide the most representative simple benchmark for comparison.

4.6.6 Linear Model Performance

The linear models show strong performance when predicting raw active AC power but substantially weaker performance when using the normalized formulation.

Table 8: Unified linear model performances.

Model	Validation R^2	Test R^2	Validation NRMSE	Test NRMSE
OLS: Active Power	0.9685	0.9424	0.0477	0.0545
OLS: Normalized Active Power	0.8543	0.9549	0.1026	0.0482

The raw-target Ordinary Least Squares (OLS) model achieves high accuracy and maintains good generalization, with only a modest drop in R^2 (≈ 0.026) and a small increase in NRMSE (≈ 0.007) when moving from validation to test. This indicates that a largely linear relationship exists between the selected features and active AC power, particularly driven by DC-side variables that capture most of the variance in inverter output.

In contrast, the normalized formulation behaves less predictably. While its validation performance is substantially lower than the raw-target model, it exhibits an unusually large improvement on the test set (increase of ≈ 0.10 in R^2 and decrease of ≈ 0.05 in NRMSE). This volatility reflects the sensitivity of the normalized target to the temperature-dependent rating curve: the normalization introduces additional variance and removes structure that the linear model otherwise captures directly. The inconsistency across splits suggests weak generalization and unstable behavior, rather than genuine superior performance.

Both linear variants outperform the naïve baselines; however, given the instability and lower predictive value of the normalized model, we elected not to train regime-specific variants for the normalized formulation. Regime-aware models show a slightly different pattern. Their performance is summarised in Table 9:

Table 9: Regime-aware linear regression model performance.

Model	Validation R^2	Test R^2	Validation NRMSE	Test NRMSE
OLS: Regime-Aware	0.9829	0.9274	0.0359	0.0682

The regime-aware model achieves stronger performance on the validation set but degrades more sharply on the test set compared with the unified raw-target model. This suggests that splitting by regime allows the model to capture more fine-grained structure during training but reduces stability and generalization. In principle, this behavior could be improved through regularization or more extensive hyperparameter tuning; however, nonlinear models evaluated in the same round of experimentation substantially outperformed the linear variants. As a result, further effort spent optimizing linear models was not justified, and attention shifted toward the more expressive nonlinear approaches.

4.6.7 Random Forest Performance

The RF models demonstrate excellent predictive performance across both unified and regime-aware configurations, substantially outperforming the linear models and approaching near-perfect fit on both validation and test sets.

Table 10: Random Forest model performance.

Model	Validation R^2	Test R^2	Validation NRMSE	Test NRMSE
RF: Unified	0.9950	0.9960	0.0190	0.0143
RF: Regime-Aware	0.9976	0.9985	0.0135	0.0099

The unified RF model achieves extremely strong performance, with validation and test R^2 values of 0.9950 and 0.9960, respectively. The small improvement from validation to test and the reduction in

NRMSE suggest that the unified model generalizes well and is not overfitting to transient behaviors in the training period. This stability reflects the ability of tree ensembles to capture nonlinear relationships across both daytime and nighttime regimes without requiring explicit regime separation.

The regime-aware RF model performs even better, achieving R^2 values of 0.9976 (validation) and 0.9985 (test), along with meaningful reductions in NRMSE. Splitting the data by operational regime removes conflicting patterns (such as nighttime zero-DC conditions and daytime MPPT variability) allowing the trees to specialize on more homogeneous behaviors. This leads to higher overall accuracy while preserving generalization.

Both RF variants outperform the linear models by a substantial margin. The improved performance arises from Random Forests' ability to capture nonlinear interactions between DC measurements, irradiance variability, voltage behavior, and rolling features; relationships that linear models can only approximate. Furthermore, tree-based models are robust to collinearity and do not require the target normalization that destabilized the linear-normalized formulation. Because RF models already deliver strong performance with default hyperparameters and without extensive tuning, they serve as a reliable nonlinear benchmark for the more expressive boosted-tree models evaluated next.

4.6.8 XGBoost Model Performance

The XGBoost models deliver the strongest performance among all machine learning approaches evaluated, achieving exceptionally high accuracy across both unified and regime-aware configurations.

Table 11: XGBoost model performance.

Model	Validation R^2	Test R^2	Validation NRMSE	Test NRMSE
XG: Unified	0.9992	0.9990	0.0076	0.0072
XG: Regime-Aware	0.9990	0.9996	0.0072	0.0050

The unified XGBoost model achieves extremely high predictive accuracy, with validation and test R^2 values of 0.9992 and 0.9990, respectively. The near-identical performance across splits indicates excellent generalisation and suggests that XGBoost is able to model the dominant nonlinear structure governing inverter behaviour using only standard hyperparameters. The consistently low NRMSE further highlights the stability of its predictions across both high-production and low-production periods.

The regime-aware model performs even better, achieving 0.99961 on validation and 0.9996 on the test set, along with meaningful reductions in NRMSE. By allowing the model to specialise separately on daytime and nighttime conditions, the boosted-trees capture subtle operational differences that remain even after feature engineering (such as distinct relationships between DC loading, efficiency behaviour, and low-generation nighttime noise). The improvements over the unified model, although small in absolute terms, reflect XGBoost's ability to exploit fine-grained patterns in homogeneous data subsets.

Across all machine learning models, XGBoost provides the highest level of accuracy and the most stable generalisation. Compared with Random Forests, the performance gains arise from boosting's capacity to model interactions sequentially and correct residual errors, leading to finer resolution of temperature effects, DC-side nonlinearities, and curtailment behaviour. These results strongly support the use of XGBoost as the nonlinear benchmark and motivate its selection as the machine-learning component in the hybrid modelling framework evaluated later in this chapter.

4.6.9 Hybrid Model Performance

The physics-based component produced a single pair of empirical fitting constants (a, b) across all inverters. Fitting the electrical-efficiency curve independently for each device yielded effectively identical values, indicating negligible device-level variation in the relationship between load ratio and electrical efficiency. The final fitted parameters were

$$a = 9.5824 \times 10^{-16}, \quad b = 760.23888,$$

suggesting an extremely sharp transition in the electrical-efficiency curve at low load ratios and confirming the well-known tendency of utility-scale inverters to operate near peak electrical efficiency across the majority of their generating range.

4.6.9.1 Feature Selection for the Residual Component The XGBoost residual model selected a feature set that differed from the one obtained for the standalone ML models. Following the same sequential inclusion order, we observe that DC current, DC power, median POA, and per-unit DC bus voltage each improved predictive performance. Notably, none of the DC-side rolling-mean features were included. This differs from the forecasting-oriented ML models and suggests that:

- the residual between measured active power and the physics model predominantly reflects steady-state DC-side discrepancies rather than short-term noise, and
- adding multiple collinear DC-side metrics is beneficial in the residual context, likely because the residual surface retains several distinct DC-side dependency structures not captured by the physics model.

Among grid-side features, only power factor and per-unit line-to-line voltage were retained. Frequency and its rolling mean, as well as rolling-mean power factor and the temperature-dependent rating curve, were excluded. This pattern indicates that the physics model already captures the primary load-dependent effects, and the remaining residual variation tied to grid behaviour is limited to instantaneous PF and voltage deviations; variables that directly influence real-power transfer but do not appear to require smoothed or derived representations.

For thermal variables, the residual model retained maximum IGBT temperature, internal temperature, and the 18-hour rolling mean of internal temperature. This implies that although the physics model incorporates a thermal-derating term, it does not fully capture slower thermal dynamics or inverter-specific thermal behaviour, including gradual heating and cooling trends tied to ambient conditions, ventilation variability, or unit-level ageing.

The final set of selected features is:

(i) DC Current

(ii) DC Power

(iii) Median POA

(iv) Per-Unit DC Bus Voltage

(v) Power Factor

(vi) Per-Unit Line-to-Line Voltage

(vii) Maximum IGBT Temperature

(viii) Internal Temperature

(ix) 18-Hour Rolling Mean of Internal Temperature

4.6.9.2 Model Performance Performance was evaluated at four levels: the physics component alone, the residual model alone, the combined hybrid model, and a curtailment-augmented hybrid model. Results are shown below.

Table 12: DT hybrid Model and component performance.

Model	Target Variable	Validation R^2	Test R^2	Validation NRMSE	Test NRMSE
Physics Component	Active power	0.98820	0.98950	0.02980	0.02590

Model	Target Variable	Validation R^2	Test R^2	Validation NRMSE	Test NRMSE
Residual Component	Residual	0.99900	0.99810	0.00081	0.00099
Hybrid	Active power	0.99999	0.99998	0.00081	0.00099
Curtailed Hybrid	Active power	0.99999	0.99998	0.00088	0.00117

The physics component alone already achieves strong performance, with $R^2 \approx 0.99$, indicating that the first-principles formulation captures the majority of inverter behavior. The remaining error represents both measurement noise and device-specific behavior not explained by the idealized efficiency curves. The residual model achieves near-perfect accuracy when predicting the physics-model residuals ($R^2 \approx 0.999$). This reflects two factors:

1. the residual surface is smooth and highly structured, and
2. the hybrid decomposition simplifies the machine-learning task by relegating complex nonlinearities to a smaller correction space.

The full hybrid model inherits the residual model's performance, reducing NRMSE by nearly two orders of magnitude compared with the physics model alone. The hybrid formulation therefore provides a substantial accuracy gain while preserving interpretability and physical meaning.

The curtailment-augmented hybrid model enforces an explicit physical constraint on the predicted output. As expected, this slightly increases error (because clipping introduces discontinuities), but ensures alignment with operational limits. This trade-off is typically desirable when the model is used in operational or anomaly-detection settings where violating curtailment caps would be unrealistic.

Overall, the hybrid architecture delivers the best performance of any modelling approach evaluated in this project, combining physical interpretability, strong generalization, and extremely low error on both validation and test sets.

4.6.10 Interpretation of Digital Twin Performance

To evaluate the overall DT system, we compare the performance of all modelling approaches (linear models, Random Forests, XGBoost, and the hybrid physics-ML architecture) and summarize their relative strengths, weaknesses, and implications for the research question.

Across all experiments, the linear models performed surprisingly well when predicting raw active AC power, achieving validation and test R^2 values above 0.94. This indicates that a large proportion of inverter behavior (approximately 80-90% of the explainable variance) is governed by fundamentally linear relationships, primarily driven by DC-side electrical variables. However, the linear models consistently failed to capture the remaining nonlinear structure, especially under conditions involving thermal effects, voltage deviations, and curtailment behavior. The normalized linear model, in particular, showed volatile generalization, reinforcing that linear methods are sensitive to target transformations and cannot reliably model the full operational envelope.

The tree-based nonlinear models provided far stronger performance. Random Forests reduced error substantially over the linear models, capturing local nonlinearities and interactions absent in the OLS formulations. XGBoost further improved performance, producing extremely low NRMSE values and test R^2 approaching 0.999. These results indicate that the final 10–20% of inverter behaviour (unexplained by linear models) is governed by nonlinear relationships, including device-specific thermal dynamics, actuator/controller behaviours, and subtle grid-side interactions.

The hybrid model achieved the best overall performance. The physics component alone accounted for roughly 99% of the variance, and the residual XGBoost model captured nearly all remaining nonlinear structure, pushing hybrid accuracy to effectively perfect levels (test NRMSE ≈ 0.001). This demonstrates two key insights:

1. Inverter behaviour is well-represented by known physics, and

2. The remaining deviations, those that matter most for anomaly detection and predictive maintenance, are highly structured and machine-learnable.

Taken together, these results indicate that the DT is able to reproduce inverter behaviour with extremely high fidelity, supporting its use for anomaly detection, operational benchmarking, and downstream modelling tasks.

4.6.11 Implications for the Research Question

The central research question of this project is:

> Can Data Science methodologies be used to identify sub-optimal solar inverter energy generation and future maintenance events?

The results across all modelling approaches strongly support an affirmative conclusion:

- The high predictive accuracy of the nonlinear and hybrid models indicates that departures from expected behavior can be detected at very fine resolution.
- The physics-guided decomposition ensures that deviations have interpretable physical meaning (e.g., thermal underperformance, DC/AC mismatch, efficiency loss).
- The machine-learning residual component learns inverter-specific patterns; exactly the type of behavior most indicative of degradation, component drift, or emergent faults.

Because the DT provides an accurate expected operational benchmark, any significant deviation from its prediction forms a reliable signal for anomaly detection and potential early-stage maintenance needs. Thus, the modelling framework directly supports the detection of sub-optimal generation and contributes to maintenance decision-making.

4.6.12 Comparison With Related Work

The DT developed in this project achieves extremely low prediction error, with the hybrid model attaining an NRMSE below 0.1% and an R^2 exceeding 0.999 on both validation and test data. While direct numerical comparison to prior work must be made cautiously, since the literature reports performance using different metrics (RMSE, NRMSE, R^2) and on different PV components, it is useful to situate our results against representative studies.

Walters et al. (2023) investigate PV DTs for realistic power estimation and report strong performance for neural-network-based models, with R^2 values of 0.97452 for an MLP and 0.97021 for an Elman recurrent network. Their results demonstrate that high accuracy is achievable in panel-level and system-level PV generation modelling using data-driven approaches.

Hueros-Barrios et al. (2025) focus on real-time hardware emulation for fault detection in DC–DC converters. Their virtual–physical hybrid model achieves an NRMSE of 5.93% when reproducing measured power signals. Although their target component differs from the inverter-level AC power considered here, their work underscores the importance, and difficulty, of capturing device-specific nonlinearities in PV power electronics.

Chicaiza et al. (2025) develop both physics-based and neuro-fuzzy models (ANFIS) for predicting PV system power output. Their best-performing neuro-fuzzy model achieves an NRMSE of 0.0282. While their modelling context differs (panel- and array-level prediction), their findings reinforce the advantage of hybrid and nonlinear methods for accurately capturing PV system behaviour.

Taken together, these studies highlight that:

- high-accuracy DTs for PV systems are attainable,
- nonlinear and hybrid models consistently outperform purely linear formulations, and
- component-specific modelling (panel, converter, inverter) benefits strongly from incorporating domain knowledge.

Within this broader landscape, the DT developed in this project performs at the upper end of reported accuracy ranges, even when compared qualitatively to state-of-the-art methods. The hybrid model's

extremely low NRMSE and near-perfect R^2 suggest that, at the inverter level, combining physics-based structure with machine-learning corrections is particularly effective. Although the underlying metrics differ across studies, the collective evidence supports the conclusion that the modelling framework developed here is consistent with, and in some respects exceeds, the predictive fidelity reported in contemporary PV DT research.

4.6.13 Limitations and Delimitations

Despite the strong performance, several limitations and delimitations remain:

- Scope limited to inverter-level modelling. Panel-level, string-level, or tracker-level behaviour is not modelled; conclusions apply strictly to inverter performance.
- Dependence on SCADA quality. Sensor errors, communication outages, and intermittent gaps in telemetry may limit performance in real deployments.
- No explicit modelling of rare fault conditions. Extremely rare events (e.g., catastrophic hardware failures) are unlikely to appear in training data and therefore not learnable by supervised models.
- Thermal and curtailment behaviour remain approximate. The physics model uses simplified derating and rating-curve approximations; real-world inverter control algorithms are proprietary and more complex.
- Generalisability across plants is not guaranteed. All models are trained per-inverter and per-plant; extending the DT to unseen plants would require retraining.
- ML components are observational, not causal. While highly predictive, the models do not establish causal mechanisms underlying degradation or performance loss.

4.6.14 Overall Interpretation

Across all modelling formulations, the results demonstrate that:

- inverter behaviour is highly predictable using a combination of physics and machine learning,
- nonlinear models capture essential behaviour missed by linear models,
- the hybrid model achieves the strongest performance by pairing physical interpretability with ML flexibility, and
- the model accuracy is sufficient to support anomaly detection and maintenance forecasting in operational settings.

These findings strongly affirm the feasibility of using Data Science to detect sub-optimal generation and support maintenance-related decision-making in utility-scale solar PV plants.

Chapter V: Findings, Conclusions, Implications, and Future Work

5.1 Findings

5.1.1 Data Quality and Operational Characteristics

Across the dataset, several consistent patterns affect all modelling tasks:

Missingness and Irregularity. Missing values occur frequently and sometimes in long contiguous gaps, especially around communication interruptions or inverter resets. These gaps introduce uncertainty and reduce stability in downstream models. Irregular operational behaviour, including abrupt drops in power or intermittent zero-generation periods, is common across devices.

Operational Variability. Daily and seasonal irradiance changes drive large variations in inverter behaviour. Weather and thermal conditions strongly influence output, and transient fluctuations can resemble fault-like patterns even when the inverter is functioning normally.

Together, these properties highlight the need for robust preprocessing and caution when interpreting short-lived anomalies as indicators of degradation.

5.1.2 Digital Twin Findings

The DT models exhibit consistently strong performance in reproducing inverter active AC power.

Dominant Linear Structure. Linear models capture most of the variance in AC power, confirming that the inverter behaviour is heavily governed by linear relationships driven primarily by DC-side inputs.

Nonlinear Refinement. Tree-based models and the hybrid model capture the remaining nonlinear structure (roughly the final 10-20 percent of unexplained behaviour) which includes device-specific quirks, thermal effects, and operational irregularities.

Hybrid Model as the Benchmark. The hybrid approach produces the most accurate representation of inverter output, closely matching measured AC power across validation and test sets. This provides a highly reliable benchmark for identifying suboptimal generation and supports downstream anomaly detection and maintenance decisions.

Overall, the DT demonstrates that data-driven and physics-informed methods together can reproduce inverter behaviour at high fidelity, enabling fine-grained operational monitoring.

5.1.3 Predictive Maintenance Findings

Because failure labels were not reliable, predictive-maintenance tasks relied on indirect indicators rather than explicit fault events.

Despite this, two key findings emerged:

- Ensemble Methods Are Most Effective.
- Boosting models, especially XGBoost combined with density-based clustering, achieved the best performance in identifying abnormal behaviour patterns. This indicates that unusual operating regimes form coherent structures in feature space.

Weak Early Warning Signals Exist. By shifting labels earlier, small but meaningful predictive signals were detected up to 48 hours before abnormal events. Even though these signals are faint, they suggest the potential for early-warning indicators when better-labelled data becomes available.

5.1.4 Forecasting Findings

The forecasting models reveal several broad insights:

- The time series exhibits strong seasonality and autoregressive structure; reflected in the superior performance of SARIMA.

- VAR models benefit from multivariate dependencies, outperforming simple baselines.
- Deep learning models improve when trained per inverter, suggesting that device-specific behaviour and idiosyncrasies are crucial.
- Transformer-based models underperform with the available data volume, indicating they require larger datasets to be effective.

These findings reinforce that inverter behaviour is structured and predictable but sensitive to device-level differences and data availability.

5.1.5 Anomaly Detection Findings

The combined anomaly flag, created by merging multiple independent checks, effectively identifies timestamps where inverter performance deviates from expectation. These anomalies frequently align with known operational disruptions such as rapid power drops, sensor inconsistencies, or curtailment events.

The multi-check approach substantially reduces false positives and provides a practical tool for operators to pinpoint unusual behaviour without relying on a single noisy indicator.

5.1.6 Overall Synthesis

Across all modelling streams, a clear theme emerges:

> Inverter behaviour is highly structured and largely predictable, and data science methodologies can reliably identify deviations from expected performance.

The DT offers a strong real-time benchmark, forecasting models quantify short- and medium-term expectations, and anomaly detection and predictive-maintenance analyses highlight departures from normal operation. Despite data-quality limitations, the collective evidence supports the project's main research question: data-driven and hybrid modelling can meaningfully assist in identifying suboptimal energy generation and informing maintenance decisions.

5.2 Discussion and Implications

The findings from the preceding chapters carry several practical, theoretical, and methodological implications for solar farm monitoring, predictive maintenance, and data-driven reliability assessment.

5.2.1 Practical Implications

A major practical insight is that data quality is the primary factor limiting model reliability. The inverter dataset contains missing values, zero readings at night, and occasional invalid measurements. All of these must be cleaned or reconstructed before they can be used effectively. Predictive accuracy would improve significantly if future datasets contained clearer event markers and well-defined failure labels. Without explicit labels, supervised failure prediction is not feasible.

In the absence of true failure labels, short-horizon forecasting provides a practical early-warning mechanism. When the predicted active power falls below the expected range, operators can be alerted to investigate anomalies or emerging faults. Kaplan-Meier and Cox proportional hazards models also provide risk-aware estimates without requiring explicit failure markers. These statistical tools support maintenance planning and help reduce unplanned outages.

The project showed that methods from the literature do not always generalize to real inverter behavior. For example, linear trend models that assume global linearity do not work well with the segmented and nonlinear daily structure observed in the MN8 data. Neural networks and other nonlinear models proved more capable of capturing these patterns and produced more reliable short-term forecasts.

The DT provides additional practical value. It produces a stable and physically informed estimate of active AC power, which supports anomaly detection and helps standardize inconsistent data streams. More accurate DT predictions lead to better performance in downstream forecasting and maintenance models. Continued development of the hybrid physics and machine learning approach will strengthen real-time monitoring and operational decision-making.

Ensemble-based predictive maintenance results show that meaningful patterns related to abnormal behavior can be extracted even without explicit failure labels. Boosting algorithms combined with density-based clustering produced the strongest performance. The exploratory 48 hour shifted-label experiment suggests that early signs of deterioration may exist, although these signals are weak. While not suitable for automated decision-making yet, they provide useful direction for future data collection and labeling practices.

5.2.2 Theoretical and Scientific Implications

A combined system that integrates DT modeling, forecasting, anomaly detection, survival analysis, and predictive maintenance provides the most complete representation of inverter behavior. Each module captures a different aspect of system performance. Together they form a comprehensive reliability framework.

The analyses show that inverter behavior is device specific. Therefore, individualized models yield better performance than a single generalized model across all devices. While the system architecture can be standardized, the model parameters should be learned for each inverter independently.

Environmental and operational factors influence the power time series strongly. Irradiance patterns, seasonal shifts, and temperature all contribute to daily variability. These findings indicate that future research should incorporate richer metadata such as panel-level temperature, sensor angle, module age, and high-resolution weather information.

The DT also carries scientific value because it combines mechanistic inverter physics with data-driven correction. This hybrid approach provides a pathway for future PV DT development, especially in settings where neither pure physics nor pure machine learning is sufficient on its own.

5.2.3 Methodological Implications

The project highlights that real-world operational datasets contain irregularities. Missing data, nonlinear regimes, and unexpected operational transitions require that algorithms adapt to the dataset rather than the dataset being forced into a particular method. Exploratory data analysis is therefore essential for selecting appropriate models.

The trend behavior in the MN8 data shows that linear regression models recommended in the literature cannot capture the nonlinear and segmented structure of real inverter output. Neural networks provide a more flexible modeling approach and are better suited to these characteristics.

Survival analysis remains possible even without true failure labels because Kaplan-Meier and Cox proportional hazards estimators can function under partial observability. However, the results should be interpreted as indicators of risk rather than precise predictions of failure. Labeling historical data accurately would require significant expert time and is not scalable across large datasets. For this reason, short-term forecasting is a practical alternative for early fault detection.

The DT contributes methodologically by standardizing and reconstructing data before it is passed to anomaly detection, forecasting, and predictive maintenance models. This leads to more consistent inputs and more reliable downstream performance. Improvements in the DT directly improve the accuracy of the entire modeling pipeline.

The predictive maintenance experiments show that clustering affects supervised learning performance. Density-based clustering produced more useful classes than centroid-based methods and should be considered in future maintenance modeling. The 48 hour shifted-label experiment also highlights that proxy labels may reveal early risk signals even when explicit failure labels are unavailable.

5.3 Conclusion

This project presented a comprehensive analytical examination of the MN8 inverter dataset across five subproblems: predictive maintenance, survival analysis, forecasting, anomaly detection, and DT modelling. Several overarching insights emerged.

Data quality proved to be a major constraint. Missing values, communication gaps, nighttime zeros, and

occasional invalid readings complicated all modelling tasks, particularly those requiring temporal continuity or explicit failure labels. As a result, forecasting, survival estimation, and predictive maintenance required careful preprocessing and could not rely on fully supervised failure-driven AIML methods.

Across the modelling tasks, we found that methods needed to be adapted to the characteristics of the data rather than applied directly from prior literature. Nonlinear and hybrid approaches generally outperformed linear or decomposition-based models, and daily trend patterns required flexible architectures such as neural networks. Survival analysis produced reasonable risk estimates but could not identify true failures due to the lack of explicit event labels.

Forecasting and anomaly detection performed reliably, demonstrating that short-horizon power prediction and identification of operational deviations are both feasible. Ensemble-based predictive maintenance experiments showed that meaningful structure relating to abnormal behaviour exists in the dataset even without labelled failures; boosting models paired with density-based clustering produced the strongest results and revealed hints of weak early-warning patterns.

The DT played a central unifying role by providing a stable, physics-informed reconstruction of inverter active AC power. Its outputs supported anomaly detection, improved the consistency of forecasting inputs, and supplied the most reliable historical baselines for maintenance-related analyses. The hybrid DT, combining a physics model with an XGBoost residual, achieved the best performance of all models, reflecting both the largely linear nature of inverter behaviour and the nonlinear device-specific effects best handled by machine learning. Continued development of this model will directly benefit the downstream predictive tasks that depend on its estimates.

Overall, despite challenges stemming from missingness, non-linearity, and the absence of explicit failure labels, the results show that data-driven and hybrid modelling approaches can meaningfully support operational monitoring and risk assessment in utility-scale PV plants. The combination of DT, forecasting, anomaly detection, and predictive-maintenance models provides a strong foundation for improving reliability, situating current asset performance, and reducing the burden of meeting contractual energy obligations.

5.4 Future Work

Several avenues exist for extending and strengthening the system developed in this project. These opportunities span improvements to the DT, data quality, predictive maintenance, forecasting, anomaly detection, and human interpretability of model outputs.

5.4.1 Advancing the Digital Twin

The DT can be further developed in several ways. First, incorporating more detailed physics would help the model better capture device specific behavior, particularly around thermal dynamics, clipping behavior, and efficiency transitions. Second, the residual learning component could be improved with richer feature sets, better temperature modeling, and more inverter specific operational metadata. Third, expanding the DT to ingest additional contextual information such as irradiance forecasts or real time weather conditions would allow it to produce more forward looking estimates suitable for operational planning. Finally, automating DT recalibration using online learning or drift detection would allow the system to adjust to changes in inverter aging, sensor degradation, or seasonal shifts.

The DT also has the potential to serve as the central data preparation layer for all downstream predictive maintenance, anomaly detection, and survival analysis models. Continued refinement of its interpolation, outlier masking, and reconstruction capabilities will directly improve the performance and reliability of all other modeling components.

5.4.2 Natural Language Summaries and Operator Facing Explainability

As DTs become more integrated in solar plant operations, there is value in generating human readable descriptions of system state. Recent work by Jan Sturm et al. (2024) explored the use of Large Language Models to convert DT outputs into short explanatory narratives. A similar approach could be incorporated into this project, where numerical outputs from the DT and predictive maintenance models are transformed into concise operational summaries. This would improve situational awareness for operators

and introduce a structured method for logging system behavior in readable form. Further research is needed to evaluate appropriate model size, latency constraints, and the reliability of generated text.

5.4.3 Improving Data Coverage and Sensor Context

Many limitations encountered in the project stem from missing data, irregular gaps, and the absence of labeled failure events. Several improvements would materially strengthen future modeling efforts.

- Integrating detailed weather metadata such as cloud cover, wind speed, humidity, and minute level irradiance would help distinguish environmental variability from equipment degradation.
- Collecting richer hardware information including temperature at panel or string level, inverter age, and expected end of life characteristics would provide more meaningful features for predictive maintenance models.
- Developing a rule based or predicate logic labeling system to automatically tag inverter outages, communication failures, and suspected degradation would substantially improve the quality of training data. Automated labeling represents a significant opportunity for future work because it reduces reliance on manual SME annotation and enables large scale model development.

5.4.4 Future Work for Predictive Maintenance

The predictive maintenance stream revealed promising early patterns but also highlighted several directions for further study.

- Clustering structure should be explored more fully. DBSCAN outperformed K Means, but only a single K value was tested. A sweep across different values of K, and potentially other clustering methods, may uncover additional structure.
- Failure label engineering deserves deeper investigation. The exploratory 48 hour future risk label demonstrated that early warning signals may exist in the data. Future work should test 12 hour, 24 hour, and rolling horizon labels, along with more refined definitions of failure precursors.
- With properly labeled data, more advanced supervised models could be evaluated including temporal attention models, transformer architectures, and survival based deep learning methods.

5.4.5 Future Work for Forecasting

Several improvements can be made to the forecasting models.

- Hyperparameter tuning across all forecasting models (VAR, SARIMA, LSTM, Chronos) would likely yield higher accuracy.
- Domain informed feature selection could improve exogenous variable choices, particularly for irradiance and weather related drivers.
- More robust error handling, especially around model fitting in periods with missing or inconsistent data, would create more stable pipelines.
- Ensemble forecasting that combines statistical and deep learning models could produce more robust predictions across changing environmental conditions.
- Physics Informed Neural Networks (PINNs) are the cutting edge of latest research in physics constrained prediction, this addition of physics may increase the possible time horizon.

Bibliography

Abdelrahman, M., Macatulad, E., Lei, B., Quintana, M., Miller, C., & Biljecki, F. (2025). What is a digital twin anyway? Deriving the definition for the built environment from over 15,000 scientific publications. *Building and Environment, 274*, 112748. https://doi.org/10.1016/j.buildenv.2025.112748

Ahmed, S. I., Bhuiyan, K. A., Rahman, I., Salehfar, H., & Selvaraj, D. F. (2024). Reliability of regression based hybrid machine learning models for the prediction of solar photovoltaics power generation. *Energy Reports, 12*, 5009–5023. https://doi.org/10.1016/j.egyr.2024.10.060

Al-Humairi, A., Khalis, E., Al Hemyari, Z. A., & Jung, P. (2025). The impact of data augmentation on AI-driven predictive algorithms for enhanced solar panel cleaning efficiency. *Processes, 13*(4), 1195. https://doi.org/10.3390/pr13041195

Golnas, A. (2012). PV system reliability: An operator's perspective. *2012 IEEE 38th Photovoltaic Specialists Conference (PVSC) PART 2*, 1–6. https://doi.org/10.1109/pvsc-vol2.2012.6656744

Antonio, K. (2024, January 16). *Solar and wind to lead growth of U.S. power generation for the next two years*. U.S. Energy Information Administration. https://www.eia.gov/todayinenergy/detail.php?id=61242

Arafet, K., & Berlanga, R. (2021). Digital twins in solar farms: An approach through time series and deep learning. *Algorithms, 14*(5), 156. https://doi.org/10.3390/a14050156

Chicaiza, W. D., Topa, A. O., Sánchez, A. J., Escaño, J. M., & Álvarez, J. D. (2025). Model design for photovoltaic facilities based on fuzzy neural network as core of its digital twin. *Energy Conversion and Management, 342*, 120001. https://doi.org/10.1016/j.enconman.2025.120001

Dhillon, B. (2017). *Engineering systems reliability, safety, and maintenance*. CRC Press. https://doi.org/10.1201/9781315160535

Dhillon, B. S. (2002). *Engineering maintenance*. CRC Press. https://doi.org/10.1201/9781420031843

Dhoke, A., Sharma, R., & Saha, T. K. (2020). A technique for fault detection, identification and location in solar photovoltaic systems. *Solar Energy, 206*, 864–874. https://doi.org/10.1016/j.solener.2020.06.019

Firth, S. K., Lomas, K. J., & Rees, S. J. (2010). A simple model of PV system performance and its use in fault detection. *Solar Energy, 84*(4), 624–635. https://doi.org/10.1016/j.solener.2009.08.004

Gallardo-Saavedra, S., Hernández-Callejo, L., & Duque-Pérez, O. (2019). Quantitative failure rates and modes analysis in photovoltaic plants. *Energy, 183*, 825–836. https://doi.org/10.1016/j.energy.2019.06.185

Gedde-Dahl, G. S. (2022). *Optimising maintenance operations in photovoltaic solar plants using data analysis for predictive maintenance* (Master's thesis). Norwegian University of Life Sciences. https://hdl.handle.net/11250/3027040

Kavakci, G., Cicekdag, B., & Ertekin, S. (2023). Time series prediction of solar power generation using trend decomposition. *Energy Technology, 12*(2). https://doi.org/10.1002/ente.202300914

Hernández-Callejo, L., Gallardo-Saavedra, S., & Alonso-Gómez, V. (2019). A review of photovoltaic systems: Design, operation and maintenance. *Solar Energy, 188*, 426–440. https://doi.org/10.1016/j.solener.2019.06.017

Hueros-Barrios, P. J., Rodríguez Sánchez, F. J., Martín Sánchez, P., Santos-Pérez, C., Sangwongwanich, A., Novak, M., & Blaabjerg, F. (2025). Digital twin approach for fault diagnosis in photovoltaic plant DC–DC converters. *Sensors, 25*(14), 4323. https://doi.org/10.3390/s25144323

Zulfauzi, I. A., Dahlan, N. Y., Sintuya, H., & Setthapun, W. (2023). Anomaly detection using K-Means and long-short term memory for predictive maintenance of large-scale solar photovoltaic plants. *Energy Reports, 9*(12), 154–158. https://doi.org/10.1016/j.egyr.2023.09.159

Jackson, I., & Martens, P. (2024). Advancing solar energetic particle event prediction through survival analysis and cloud computing: Kaplan–Meier estimation and Cox proportional hazards modeling. *The Astrophysical Journal Supplement Series, 272*(2), 37. https://doi.org/10.3847/1538-4365/ad3fba

Jain, P., Poon, J., Singh, J. P., Spanos, C., Sanders, S. R., & Panda, S. K. (2020). A digital twin approach for fault diagnosis in distributed photovoltaic systems. *IEEE Transactions on Power Electronics, 35*(1), 940–956. https://doi.org/10.1109/tpel.2019.2911594

Kannan, N., & Vakeesan, D. (2016). Solar energy for the future world: A review. *Renewable and Sustainable Energy Reviews, 62*, 1092–1105. https://doi.org/10.1016/j.rser.2016.05.022

Keisang, K., Bader, T., & Samikannu, R. (2021). Review of operation and maintenance methodologies for solar photovoltaic microgrids. *Frontiers in Energy Research, 9*. https://doi.org/10.3389/fenrg.2021.730230

Liu, D., & Sun, K. (2019). Random forest solar power forecast based on classification optimization. *Energy, 187*, 115940. https://doi.org/10.1016/j.energy.2019.115940

Massel, L. V., Shchukin, N., & Cybikov, A. (2021). Digital twin development of a solar power plant. *E3S Web of Conferences, 289*, 03002. https://doi.org/10.1051/e3sconf/202128903002

Milan, R. (2025, June 25). *Creating a virtuous cycle* [Presentation]. Harvard University. https://harvard.zoom.us/rec/play/...

MN8 Energy, Inc. (2023). *Form S-1/A*. Securities and Exchange Commission. https://www.sec.gov/Archives/edgar/data/1908233/000119312523209389/d366853ds1a.htm

Murtaza, A. A., Saher, A., Zafar, M. H., Syed, A., Aftab, M. F., & Sanfilippo, F. (2024). Paradigm shift for predictive maintenance and condition monitoring from Industry 4.0 to Industry 5.0: A systematic review. *Results in Engineering, 24*, 102935. https://doi.org/10.1016/j.rineng.2024.102935

Nguyen, H. N., Tran, Q. T., Ngo, C. T., Nguyen, D. D., & Tran, V. Q. (2025). Solar energy prediction through machine learning models: A comparative analysis. *PLOS ONE, 20*(1), e0315955. https://doi.org/10.1371/journal.pone.0315955

OpenAI. (2025). *ChatGPT*. https://chat.openai.com/

Orosz, T., Rassõlkin, A., Arsénio, P., Poór, P., Valme, D., & Sleisz, Á. (2024). Current challenges in operation, performance, and maintenance of photovoltaic panels. *Energies, 17*(6), 1306. https://doi.org/10.3390/en17061306

Pereira, F., & Silva, C. (2023). Machine learning for monitoring and classification in inverters from solar photovoltaic energy plants. *Solar Compass, 9*, 100066. https://doi.org/10.1016/j.solcom.2023.100066

Peters, L., & Madlener, R. (2017). Economic evaluation of maintenance strategies for ground-mounted solar photovoltaic plants. *Applied Energy, 199*, 264–280. https://doi.org/10.1016/j.apenergy.2017.04.060

Pimenta, F., Pacheco, J. F., Branco, C. M., Teixeira, C., & Caetano, E. (2020). Development of a digital twin of an onshore wind turbine using monitoring data. *Journal of Physics: Conference Series, 1618*(2). https://doi.org/10.1088/1742-6596/1618/2/022065

Qureshi, M. S., Umar, S., & Nawaz, M. U. (2024). Machine learning for predictive maintenance in solar farms. *ResearchGate*, 3. https://www.researchgate.net/publication/390178131

Rajesh, R., & Carolin Mabel, M. (2015). A comprehensive review of photovoltaic systems. *Renewable and Sustainable Energy Reviews, 51*, 231–248. https://doi.org/10.1016/j.rser.2015.06.006

Ramirez-Vergara, J., Bosman, L. B., Wollega, E., & Leon-Salas, W. D. (2022). Review of forecasting methods to support photovoltaic predictive maintenance. *Cleaner Engineering and Technology, 8*, 100460. https://doi.org/10.1016/j.clet.2022.100460

Rehman, T., Qaisrani, M. A., Shafiq, M. B., Baba, Y. F., Aslfattahi, N., Shahsavar, A., Cheema, T. A.,

& Park, C. W. (2025). Global perspectives on advancing photovoltaic system performance: A state-of-the-art review. *Renewable and Sustainable Energy Reviews, 207*, 114889. https://doi.org/10.1016/j.rser.2024.114889

Sarkar, R. (2024). AI-powered predictive maintenance for solar energy systems: A case study. *International Journal for Multidisciplinary Research, 6*(6). https://doi.org/10.36948/ijfmr.2024.v06i06.30731

Singh, S., Saket, R. K., & Khan, B. (2023). A comprehensive review of reliability assessment methodologies for grid-connected photovoltaic systems. *IET Renewable Power Generation, 17*(7), 1859–1880. https://doi.org/10.1049/rpg2.12714

Sodhi, M., Banaszek, L., Magee, C., & Rivero-Hudec, M. (2022). Economic lifetimes of solar panels. *Procedia CIRP, 105*, 782–787. https://doi.org/10.1016/j.procir.2022.02.130

Springer, M., Jordan, D. C., & Barnes, T. M. (2022). Future-proofing photovoltaics module reliability through a unifying predictive modeling framework. *Progress in Photovoltaics: Research and Applications, 31*(5). https://doi.org/10.1002/pip.3645

Šturm, J., Zajec, P., Škrjanc, M., Mladenić, D., & Grobelnik, M. (2024). Enhancing cognitive digital twin interaction using an LLM agent. *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, 103–107. https://doi.org/10.1109/mipro60963.2024.10569919

Tharuka Lulbadda, K., & Hemapala, U. (2022). Use of solar PV inverters during night-time for voltage regulation and stability of the utility grid. *Clean Energy, 6*(4), 646–658. https://doi.org/10.1093/ce/zkac042

Walters, M., Yonce, J., & Venayagamoorthy, G. K. (2023). Data-driven digital twins for power estimations of a solar photovoltaic plant. *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. https://doi.org/10.1109/ssci52147.2023.10371834

Wan, J., Fu, J.-F., Liu, J.-F., Shi, J.-K., Jin, C.-G., & Zhang, H.-P. (2021). Class imbalance problem in short-term solar flare prediction. *Research in Astronomy and Astrophysics, 21*(9), 237. https://doi.org/10.1088/1674-4527/21/9/237

Yalçin, T., Paradell Solà, P., Stefanidou-Voziki, P., Domínguez-García, J. L., & Demirdelen, T. (2023). Exploiting digitalization of solar PV plants using machine learning: Digital twin concept for operation. *Energies, 16*(13), 5044. https://doi.org/10.3390/en16135044

Yao, J., Yang, Y., Wang, X.-C., & Zhang, X.-P. (2023). Systematic review of digital twin technology and applications. *Visual Computing for Industry, Biomedicine, and Art, 6*(1). https://doi.org/10.1186/s42492-023-00137-4

Zhang, P., Li, W., Li, S., Wang, Y., & Xiao, W. (2013). Reliability assessment of photovoltaic power systems: Review of current status and future perspectives. *Applied Energy, 104*, 822–833. https://doi.org/10.1016/j.apenergy.2012.12.010

Zhao, Y., Yang, L., Lehman, B., de Palma, J.-F., Mosesian, J., & Lyons, R. (2012). Decision tree-based fault detection and classification in solar photovoltaic arrays. *2012 IEEE Applied Power Electronics Conference and Exposition (APEC)*. https://doi.org/10.1109/apec.2012.6165803

Appendix A

Digital Twin Initial Feature List

This appendix provides the complete list of raw and derived features considered for use in the Digital Twin model (see Table 13). These metrics correspond to the inverter-level SCADA signals selected based on physical relevance and data availability, as described in Section 3.10.1.1.

Table 13: Raw and derived features considered for the Digital Twin.

Feature Name	Category
`AC_POWER.MEASURED`, `AC_POWER_LIMIT_SETPOINT.MEASURED`	Measured and limited real power
`AC_VOLTAGE_AB.MEASURED`, `AC_VOLTAGE_BC.MEASURED`, `AC_VOLTAGE_CA.MEASURED`	Measured AC line-to-line voltages
`DC_CURRENT.MEASURED`, `DC_POWER.MEASURED`, `DC_VOLTAGE.MEASURED`, `DC_VOLTAGE_BUS.MEASURED`	DC-side voltage, current, and power
`FREQUENCY.MEASURED`, `POWER_FACTOR.MEASURED`	Grid frequency and power factor
`STATUS_AC_MOD_ADMISSION_TEMP.MEASURED`	Air-intake temperature (ambient to inverter)
`STATUS_INTERNAL_TEMP.MEASURED`	Internal enclosure/cabinet temperature
`STATUS_IGBT_MAX_TEMP.MEASURED`	IGBT junction (device) temperature
`SVA_LIMIT_SETPOINT.MEASURED`	Apparent power configured limit
`VAR.MEASURED`, `VAR_LIMIT_SETPOINT.MEASURED`	Reactive power and its configured limit

Appendix B

Digital Twin Considered Interpolation Methods

This appendix provides the complete list of considered interpolation methods (see Table 14) as well as the final mapping of features to interpolation methods for data used in the Digital Twin model (see Table 15), as discussed in Section 3.10.1.4.

Table 14: Considered interpolation methods for Digital Twin preprocessing.

Interpolation Methods	Description
Linear	Connects adjacent points with straight lines; simple and fast but may not capture curvature.
Time	Interpolates based on actual datetime spacing between samples.
Index	Uses the integer index position as the interpolation domain.
Values	Interpolates along the ordering of data values rather than index.
Nearest	Assigns the nearest observation's value; may introduce discontinuities.
Slinear	First-order spline interpolation; piecewise linear.
Quadratic	Fits piecewise quadratic polynomials; captures curvature.
Cubic	Fits piecewise cubic polynomials; smooth for physical signals.
Polynomial	Fits a single global polynomial; unstable for long series.
Spline	General spline interpolation; smooth but can overshoot.
Pchip	Piecewise cubic Hermite; monotonicity-preserving.
Akima	Spline method robust to outliers; reduces oscillations.
CubicSpline	Cubic spline with control over boundary conditions.
from_derivatives	Interpolates using values and derivatives when available.

Digital Twin Final Interpolation Methods

Table 15: Chosen Interpolation Method per Metric.

Metric	Chosen Interpolation Method
AC_POWER.MEASURED	akima
AC_POWER_LIMIT_SETPOINT.MEASURED	from_derivatives
AC_VOLTAGE_AB.MEASURED	pchip
AC_VOLTAGE_BC.MEASURED	from_derivatives
AC_VOLTAGE_CA.MEASURED	pchip
DC_CURRENT.MEASURED	pchip
DC_POWER.MEASURED	pchip
DC_VOLTAGE.MEASURED	akima
DC_VOLTAGE_BUS.MEASURED	pchip
FREQUENCY.MEASURED	spline
POA.MEDIAN	pchip
POWER_FACTOR.MEASURED	akima
STATUS_AC_MOD_ADMISSION_TEMP.MEASURED	from_derivatives
STATUS_IGBT_MAX_TEMP.MEASURED	akima
STATUS_INTERNAL_TEMP.MEASURED	pchip
SVA_LIMIT_SETPOINT.MEASURED	akima
VAR.MEASURED	from_derivatives
VAR_LIMIT_SETPOINT.MEASURED	akima

Appendix C

Digital Twin Engineered Features

This appendix provides the complete list of engineered features (see Table 16) for data used in the Digital Twin model, as discussed in Section 3.10.3.1.

Table 16: Engineered features used in Digital Twin modelling.

Engineered Feature	Description	Formulation	Reasoning
P_R_T	Active power rating curve	Derived from apparent power rating, reactive power, and minimum curtailment setpoint	Defines the current upper limit of AC power
Vdc_pu	Per-unit DC bus voltage	DC bus voltage / nominal DC voltage	Captures efficiency shifts with MPPT deviation and bus regulation
deltaT	Admission temperature residual	Admission temperature - reference temperature	Captures temperature-dependent efficiency loss below derate
Vll_pu	Line-to-line voltage deviation per-unit	(Vab + Vbc + Vca) / (3 × nominal Vll)	Captures grid-side conditions affecting output
I_DC_rm	DC current rolling mean	90-minute rolling mean	Improves stability and reduces noise
P_DC_rm	DC power rolling mean	90-minute rolling mean	Improves stability and reduces noise
POA_rm	Irradiance rolling mean	90-minute rolling mean	Improves stability and reduces noise
IGBT_rm	IGBT max temperature rolling mean	3-hour rolling mean	Improves stability and reduces noise
PF_rm	Power factor rolling mean	90-minute rolling mean	Improves stability and reduces noise
ITEMP_rm	Internal temperature rolling mean	18-hour rolling mean	Improves stability and reduces noise
F_rm	Frequency rolling mean	12-hour rolling mean	Improves stability and reduces noise

Appendix D

Graphical User Interface

This appendix shows captures of the GUI (see Figures 46 through 54) discussed in Section 3.12.

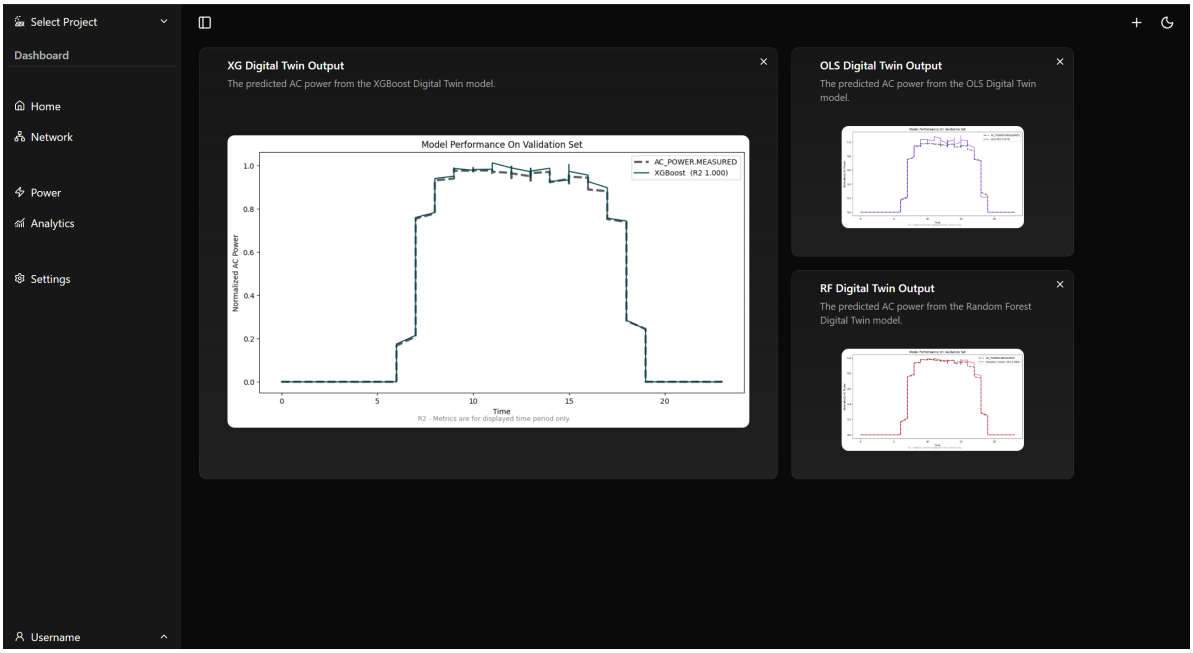

Figure 46: Overview of GUI dashboard.

Sidebar

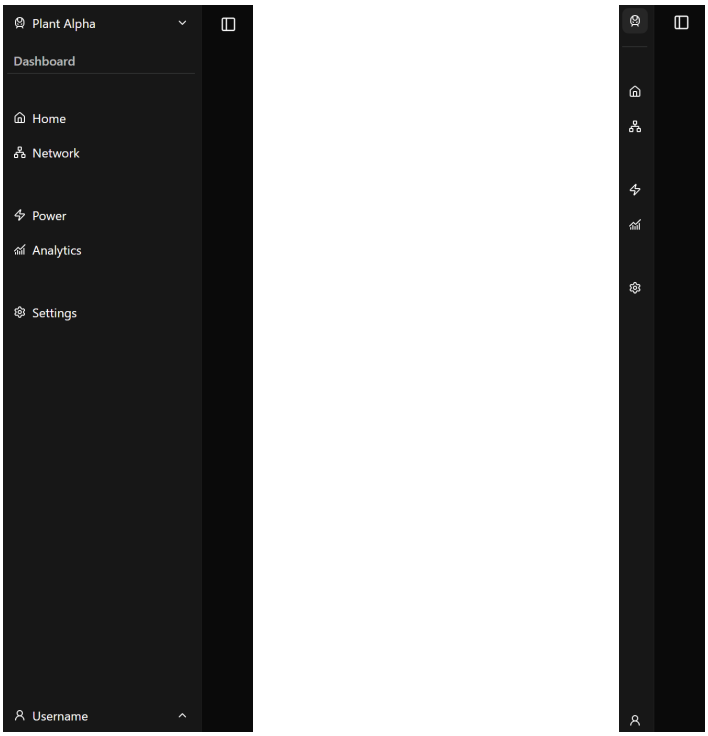

Figure 47: Expanded and collapsed views of the sidebar, used for navigation.

PV Plant Drop Down Menu

To preserve confidentiality, we refer to the anonymized solar plants as Plant Alpha, Plant Beta, Plant Gamma, and Plant Delta. These labels are purely fictitious pseudonyms and do not correspond to any MN8 project names, off-takers, or locations.

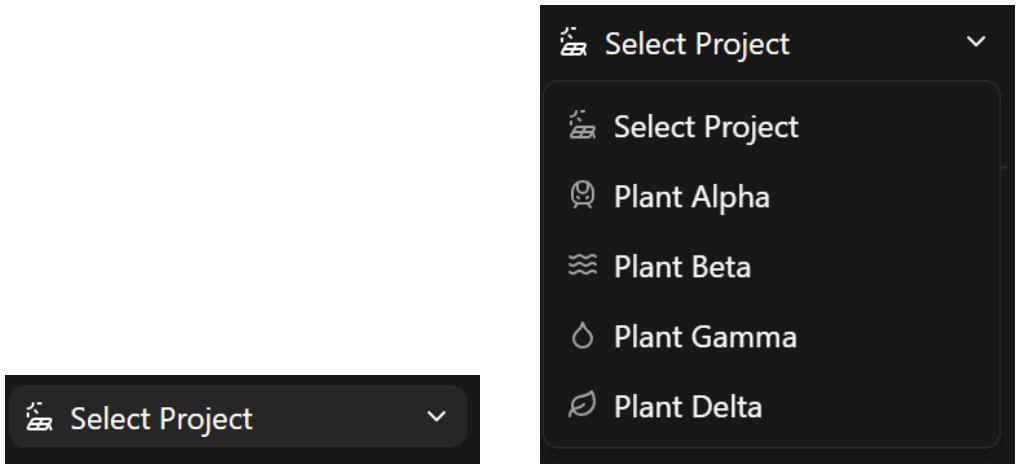

Figure 48: Expanded and collapsed views of the PV plant drop down menu, used for changing between PV plant data.

User Drop Down Menu

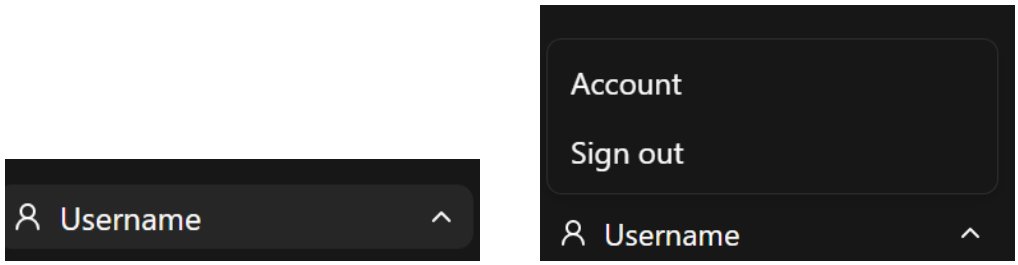

Figure 49: Expanded and collapsed views of the user drop down menu, used for accessing user related fields.

Dashboard Sheet

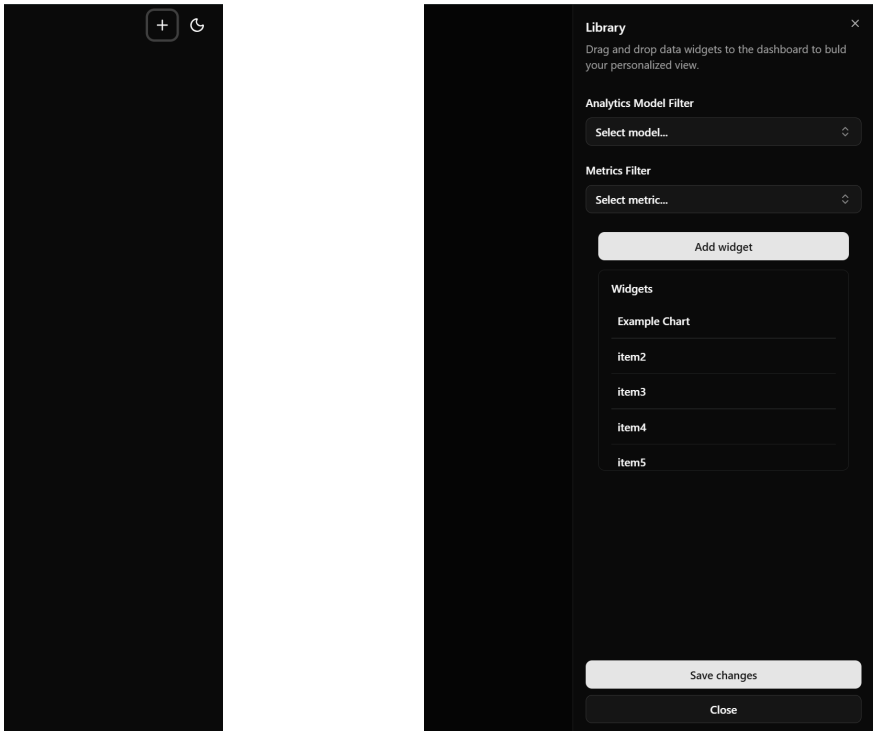

Figure 50: Expanded and collapsed views of the dashboard sheet, used for adding widgets to the dashboard.

Model Drop Down Menu

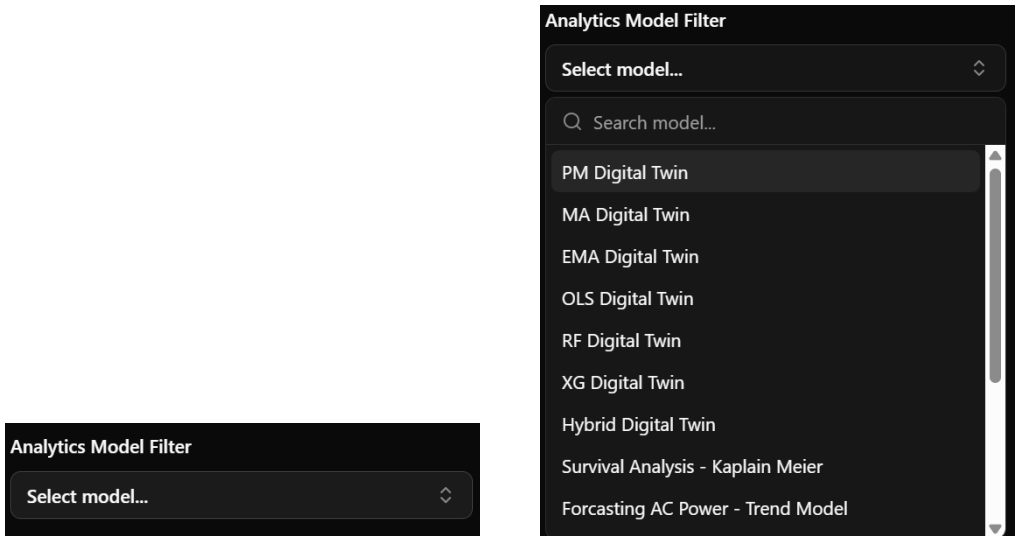

Figure 51: Expanded and collapsed views of the model drop down menu, used for adding model output plots to the dashboard.

Metric Drop Down Menu

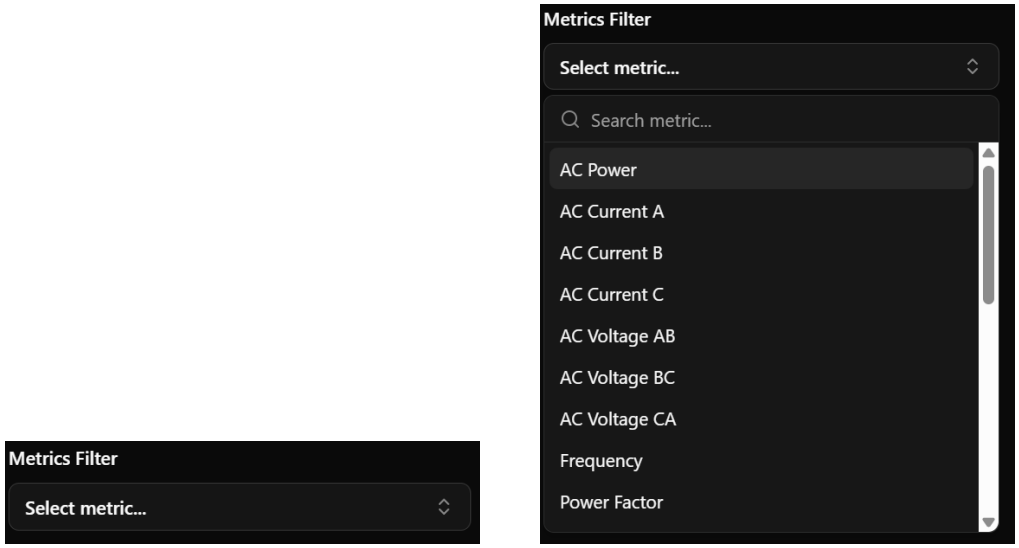

Figure 52: Expanded and collapsed views of the metric drop down menu, used for adding metric plots to the dashboard.

Theme Drop Down Menu

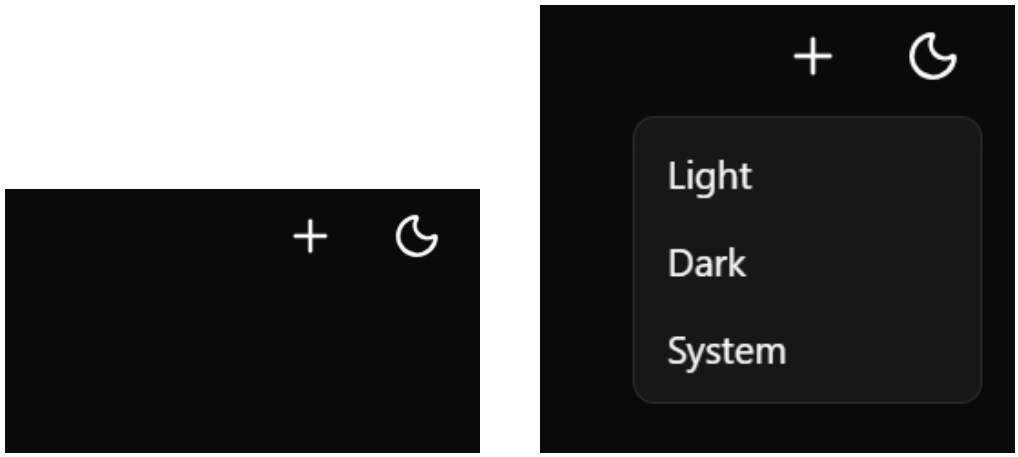

Figure 53: Expanded and collapsed views of the theme drop down menu, used for changing the theme of the dashboard.

Dashboard Widget

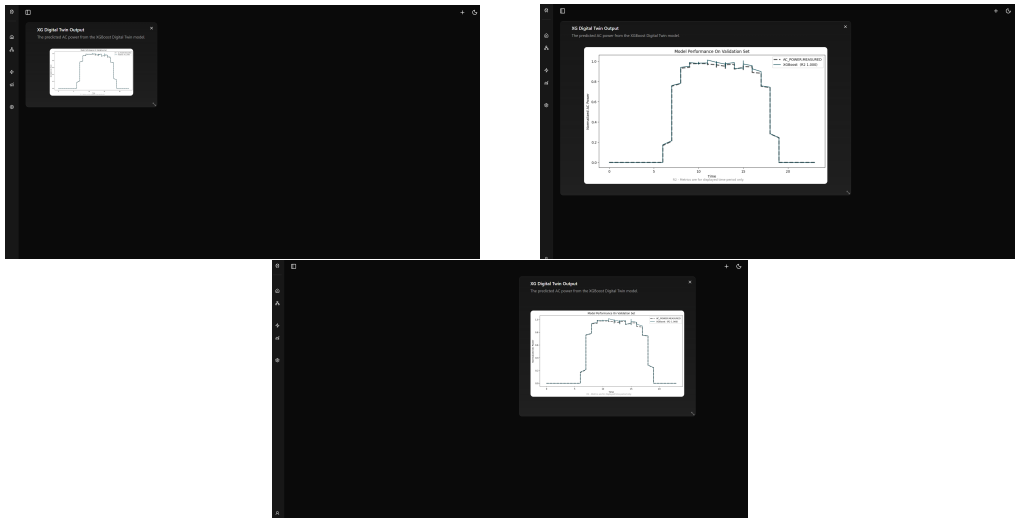

Figure 54: Static, resized, and translated views of dashboard widget.

Appendix E

Correlation Matrices

This appendix displays the results of the initial Exploratory Data Analysis (EDA) discussed in Sections 3.5 and 4.1.1.

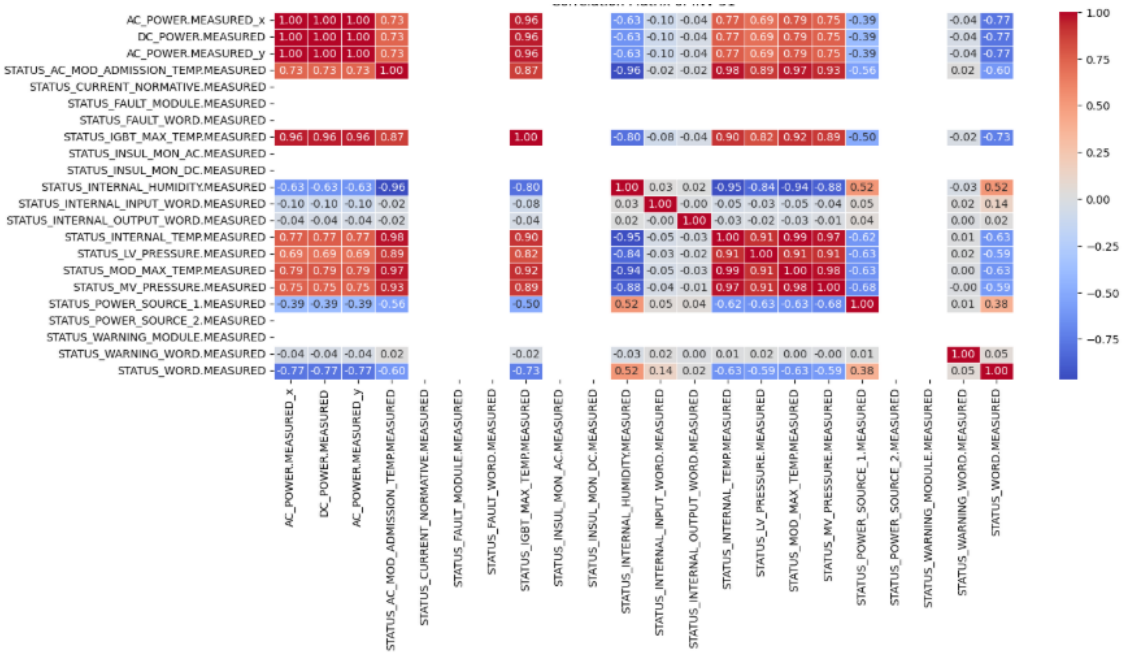

Figure 55: Correlation Matrix of all other status features and active power.

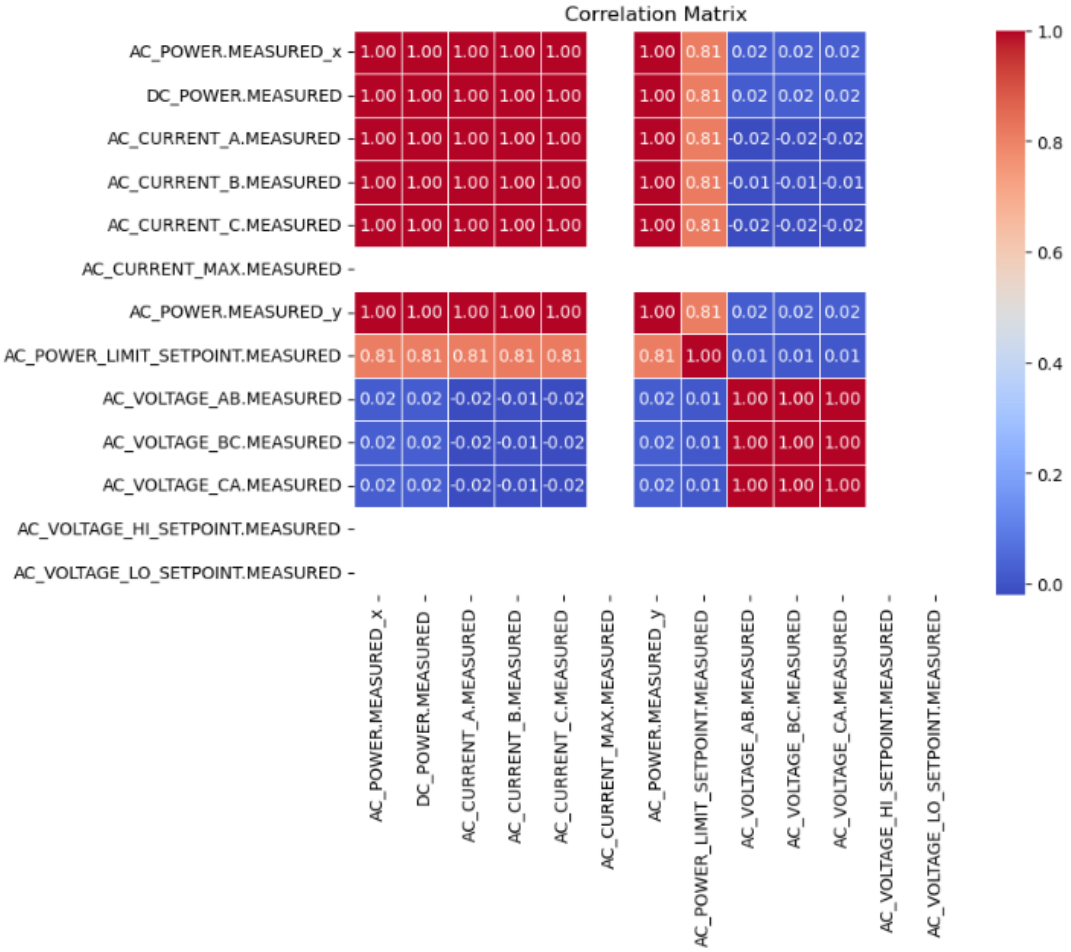

Figure 56: Correlation matrix of active power with other AC features of 1 inverter.

Appendix E

5.5 Ensemble Model Performance

This appendix details the top 5 model performances across each ensemble method.

XGB K means

	max_depth	learning_rate	n_estimators	min_child_weight	colsample_bytree	subsample	AUROC_train	AUROC_test	PRAUC_train	PRAUC_test
1	8	0.01	400	1	0.9	0.7	0.989	0.976	0.917	0.802
2	10	0.01	400	10	0.8	0.9	0.995	0.976	0.957	0.799
3	10	0.04	200	10	0.8	0.8	0.998	0.975	0.986	0.799
4	10	0.04	200	10	0.8	0.7	0.998	0.976	0.986	0.799
5	10	0.02	200	1	0.9	0.9	0.996	0.975	0.965	0.799

XGB DBSCAN

	max_depth	learning_rate	n_estimators	min_child_weight	colsample_bytree	subsample	AUROC_train	AUROC_test	PRAUC_train	PRAUC_test
1	10	0.03	200	40	0.8	0.8	0.993	0.977	0.952	0.815
2	10	0.02	200	40	0.9	0.8	0.990	0.977	0.928	0.812
3	10	0.02	200	40	0.9	0.7	0.990	0.977	0.928	0.812
4	10	0.03	200	40	0.9	0.8	0.993	0.977	0.951	0.811
5	10	0.03	200	40	0.9	0.7	0.993	0.977	0.947	0.811

Figure 57: XGBoost ensemble performance.

Random Forest K means

	n_estimators	max_depth	min_samples_leaf	max_features	class_weight	AUROC_train	AUROC_test	PRAUC_train	PRAUC_test
1	300	12	50	sqrt	balanced_subsample	0.984	0.971	0.895	0.728
2	300	18	50	sqrt	balanced	0.987	0.971	0.918	0.727
3	200	18	50	sqrt	balanced	0.987	0.971	0.917	0.725
4	200	16	75	sqrt	balanced	0.983	0.970	0.895	0.723
5	300	16	75	sqrt	balanced	0.983	0.970	0.895	0.723

Random Forest K means

	n_estimators	max_depth	min_samples_leaf	max_features	class_weight	AUROC_train	AUROC_test	PRAUC_train	PRAUC_test
1	200	18	50	sqrt	balanced	0.987	0.971	0.918	0.730
2	300	16	50	sqrt	balanced	0.987	0.971	0.917	0.729
3	600	16	50	sqrt	balanced	0.987	0.972	0.917	0.725
4	300	18	50	sqrt	balanced_subsample	0.987	0.972	0.918	0.725
5	800	18	50	sqrt	balanced_subsample	0.987	0.972	0.919	0.724

Figure 58: Random Forest ensemble performance.

AdaBoost K means

	n_estimators	learning_rate	max_depth	min_samples_	min_samples_	AUROC_train	AUROC_test	PRAUC_train	PRAUC_test
1	400	0.08	7	1	2	0.984	0.976	0.890	0.802
2	600	0.03	7	10	2	0.982	0.977	0.881	0.802
3	600	0.03	7	10	5	0.982	0.977	0.881	0.802
4	600	0.03	7	10	10	0.982	0.977	0.881	0.802
5	200	0.1	7	1	2	0.983	0.976	0.881	0.802

AdaBoost DBSCAN

	n_estimators	learning_rate	max_depth	min_samples_	min_samples_	AUROC_train	AUROC_test	PRAUC_train	PRAUC_test
1	400	0.03	7	1	5	0.983	0.977	0.887	0.814
2	400	0.03	7	10	2	0.981	0.977	0.873	0.810
3	400	0.03	7	10	5	0.981	0.977	0.873	0.810
4	400	0.03	7	10	10	0.981	0.977	0.873	0.810
5	400	0.03	7	5	2	0.982	0.977	0.875	0.809

Figure 59: AdaBoost ensemble performance.

CATBoost K means

	depth	learning_rate	iterations	l2_leaf_reg	random_strength	bagging_temperature	subsample	colsample_bylevel	AUROC_train	AUROC_test	PRAUC_train	PRAUC_test
1	4	0.1	300	3	1	0	0.9	0.6	0.983	0.972	0.910	0.787
2	4	0.1	300	3	1	1	0.9	0.6	0.983	0.972	0.910	0.787
3	4	0.05	600	1	1	0	0.9	0.6	0.983	0.972	0.914	0.786
4	4	0.05	600	1	1	1	0.9	0.6	0.983	0.972	0.914	0.786
5	4	0.1	300	5	1	0	0.5	0.9	0.983	0.973	0.913	0.785

CATBoost DBSCAN

	depth	learning_rate	iterations	l2_leaf_reg	random_strength	bagging_temperature	subsample	colsample_bylevel	AUROC_train	AUROC_test	PRAUC_train	PRAUC_test
1	4	0.1	300	5	1	1	0.5	0.8	0.983	0.973	0.913	0.796
2	4	0.1	300	5	1	0	0.5	0.8	0.983	0.973	0.913	0.796
3	4	0.1	300	3	1	0	0.9	0.8	0.983	0.973	0.913	0.795
4	4	0.1	300	3	1	1	0.9	0.8	0.983	0.973	0.913	0.795
5	4	0.1	300	1	1	1	0.5	0.6	0.983	0.971	0.912	0.794

Figure 60: CATBoost ensemble performance.

Appendix F

Digital Twin Broad Mask Distribution

This appendix visualizes the percent change in Null values across outlier masking steps outlined in Section 3.10.1.4 and discussed in Section 4.6.1 (see Figures 61 and 62).

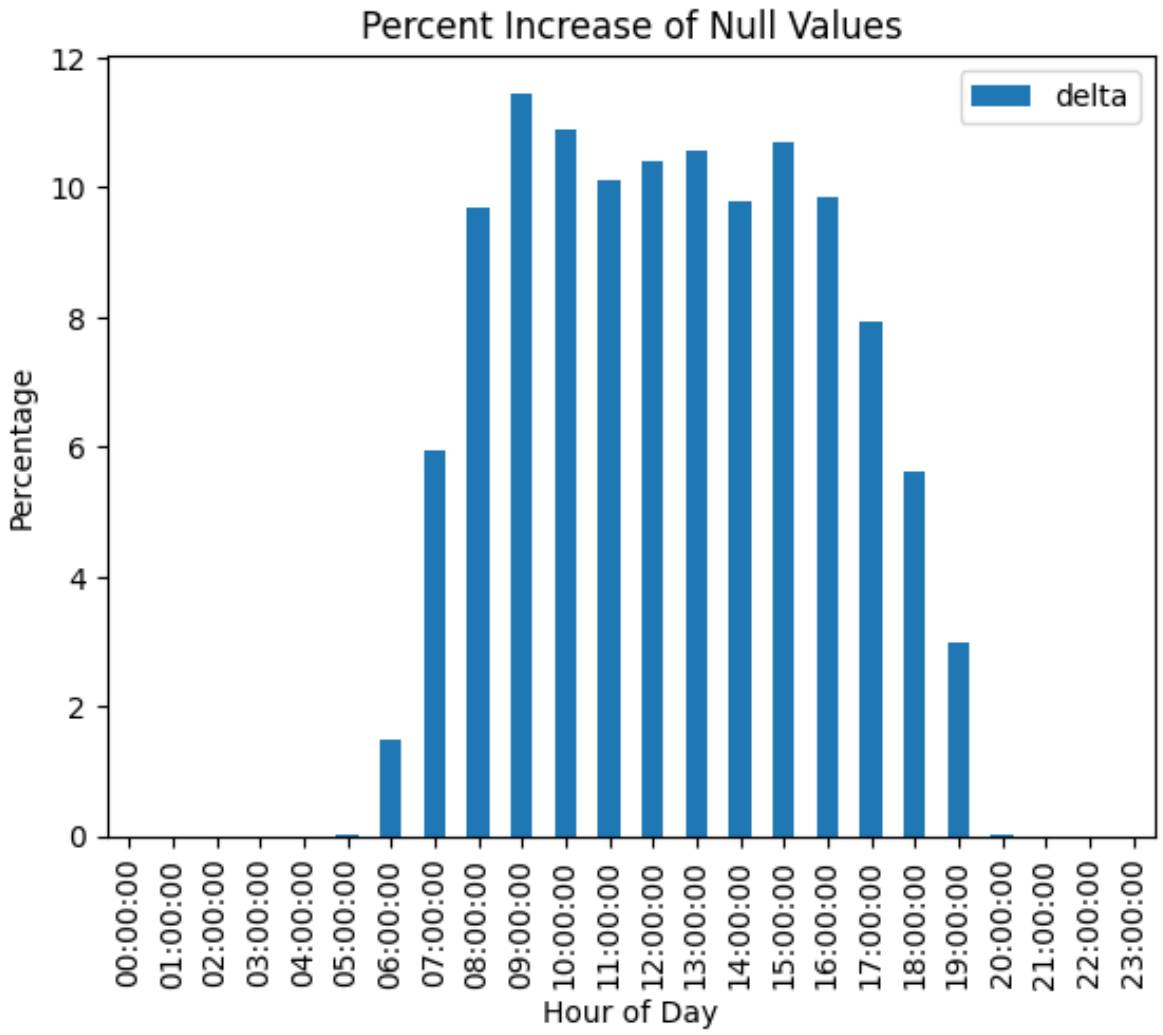

Figure 61: Digital Twin broad robust-z masking percentage increase of Null values.

Digital Twin Regime Mask Distribution

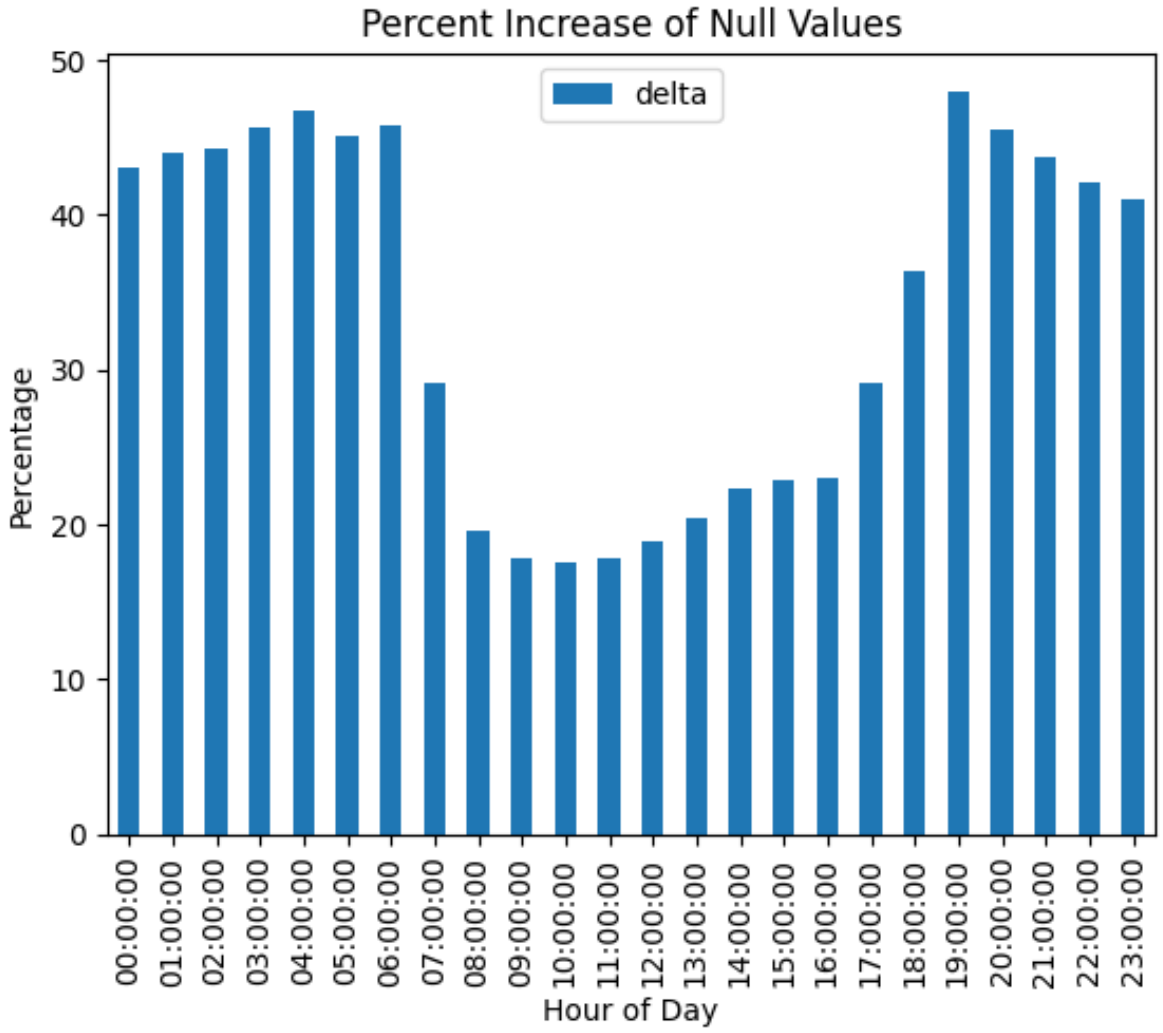

Figure 62: Digital Twin regime-aware masking percentage increase of Null values.

Appendix G

EDA Results

This appendix displays the results of the autocorrelation and cross-correlation studies executed (see Tables 17 and 18) on raw variables as discussed in Sections 3.10.1.5, 4.6.2, and 4.6.3

Table 17: Autocorrelation results per metric.

Metric	Lags	Equivalent Time
FREQUENCY.MEASURED	2	10.0 minutes
POWER_FACTOR.MEASURED	72	3.0 hours
DC_VOLTAGE_BUS.MEASURED	72	3.0 hours
AC_POWER.MEASURED	79	3.3 hours
DC_POWER.MEASURED	79	3.3 hours
DC_CURRENT.MEASURED	80	3.3 hours
POA.MEDIAN	80	3.3 hours
STATUS_IGBT_MAX_TEMP.MEASURED	110	4.6 hours
DC_VOLTAGE.MEASURED	123	5.1 hours
AC_POWER_LIMIT_SETPOINT.MEASURED	2732	1.4 weeks
STATUS_AC_MOD_ADMISSION_TEMP.MEASURED	5580	2.8 weeks
SVA_LIMIT_SETPOINT.MEASURED	5710	2.8 weeks
VAR_LIMIT_SETPOINT.MEASURED	5710	2.8 weeks
AC_VOLTAGE_CA.MEASURED	5711	2.8 weeks
AC_VOLTAGE_AB.MEASURED	5711	2.8 weeks
AC_VOLTAGE_BC.MEASURED	5711	2.8 weeks
STATUS_INTERNAL_TEMP.MEASURED	12722	6.3 weeks
VAR.MEASURED	15404	7.6 weeks

Table 18: Cross-correlation results per metric.

Metric	Median Peak Lag	Global Peak Correlation
DC_CURRENT.MEASURED	0	0.9767
DC_POWER.MEASURED	0	0.9554
STATUS_IGBT_MAX_TEMP.MEASURED	0	0.7703
POA.MEDIAN	0	0.7190
STATUS_INTERNAL_TEMP.MEASURED	12	0.6123
POWER_FACTOR.MEASURED	1	0.3613
FREQUENCY.MEASURED	12	0.1736
STATUS_AC_MOD_ADMISSION_TEMP.MEASURED	17	0.0474
DC_VOLTAGE_BUS.MEASURED	142	-0.0646
AC_POWER_LIMIT_SETPOINT.MEASURED	152	-0.0793
SVA_LIMIT_SETPOINT.MEASURED	261	-0.0821
VAR_LIMIT_SETPOINT.MEASURED	261	-0.0821
AC_VOLTAGE_BC.MEASURED	268	-0.0821
AC_VOLTAGE_CA.MEASURED	268	-0.0821
AC_VOLTAGE_AB.MEASURED	268	-0.0821
DC_VOLTAGE.MEASURED	1	-0.0968
VAR.MEASURED	5	-0.1160

Note:
***Median Peak Lag**: The typical lag at which this metric best predicts AC power at the inverter level.*
***Global Peak Correlation**: How strong this metric is as a predictor of AC power across the entire farm.*