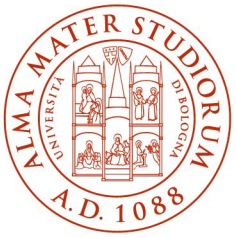


ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Department of Statistical Sciences "Paolo Fortunati"

Second Cycle Degree in Greening Energy Market and Finance

Curriculum: Renewable technologies

Deep Learning for Solar Power Forecasting: A Transformer-Based In-Context Learning Approach

Master's Thesis in Deep Learning and Artificial Intelligence

Supervisor:
Prof. Carlo Alberto Nucci

Candidate:
Chiara Simionato

Co-Supervisor:
Ing. Gabriele Piantadosi

Graduation session: March 2026
Academic year 2024/2025

Contents

List of Figures

List of Tables

Abstract

As the role of photovoltaic generation increases in the energy supply chain, forecast errors have larger operational and economic implications. Accordingly, accurate short-term PV forecasts acquire growing importance for the efficient management of power systems and for decision-making in electricity markets. In parallel, deep learning and transformers-based architectures have shown promising performance in forecasting tasks. Large Language Models (LLMs), that rely on a transformer architecture, have also shown benefits from providing as input example textual representations of input-output pairs, this practice is known as in-context learning.

This thesis investigates whether in-context learning, performed with transformer encoders, can be used as auxiliary component in day-ahead PV power forecasting when combined with a numerical Long Short Term Memory-based model (LSTM).

The empirical analysis uses inverter-level AC power from six ground-mounted PV plants in Italy, together with periodical meteorological inputs and the irradiance signal. The task is framed as a day-ahead forecast problem, leveraging a 168-hour multivariate context, with separate evaluation over full days and daylight hours. We chose Longformer, as encoder, to elaborate the structured prompt containing past k context exemplars with $k \in \{1, 2, 3\}$, whose representation is fused with the Long Short Term Memory (LSTM) output.

On a held-out test set, text conditioning yields small but systematic improvements over the purely numerical baseline: configurations with one or two exemplars modestly reduce absolute errors, while $k = 3$ leads to noisy representations that affect the model performance.

Overall, few-shot, text-based conditioning emerges as a complementary information channel for PV forecasting, providing modest benefits and suggesting scope for future work on probabilistic forecasts and richer prompt designs.

CHAPTER 1

Introduction

1.1 Problem statement

The rapid expansion of photovoltaic (PV) generation is reshaping the operation of modern power systems. As installed capacity grows at both transmission and distribution levels, system operators, market participants, and plant owners increasingly depend on short-term PV power forecasts to support unit commitment, reserve procurement, congestion management, and bidding in electricity markets. Forecast errors translate directly into imbalance costs and may compromise secure operation when PV penetration is high.

From a modelling standpoint, PV power forecasting is challenging for at least three reasons. First, PV production is driven by exogenous meteorological conditions and solar geometry rather than by controllable inputs; cloud dynamics can induce sharp fluctuations at short time scales, making the irradiance–power relationship highly non-linear. Second, the quality and resolution of available meteorological inputs, such as numerical weather prediction products, reanalysis data, or external APIs, impose an upper bound on forecasting accuracy. Third, inverter-level measurements are affected by plant heterogeneity, sensor imperfections, and operational anomalies, which introduce additional noise and uncertainty.

Recent years have seen a strong shift towards data-driven and deep learning approaches for PV forecasting. Recurrent neural networks (RNNs) and LSTM networks have become standard baselines for multivariate time-series forecasting in energy systems, while transformer-based architectures and LLMs have started to appear as flexible sequence models capable of capturing long-range dependencies and contextual information. However, most existing work treats these families of models separately: purely numerical architectures are applied to time series, whereas LLMs are used for language-like tasks.

1.2 Research question

Against this background, the thesis investigates whether long-context transformer encoders, originally developed for natural language processing, can be exploited as auxiliary components in PV power forecasting when combined with a standard numerical LSTM-based model.
The methodological novelty of this thesis lies in assessing whether conditioning the forecasts on an additional textual channel that encodes simple physical priors and a small number of past context–target examples leads to a measurable improvement in day-ahead inverter-level PV power predictions, compared to a purely numerical baseline trained on the same dataset and under the same preprocessing and validation protocol. A central aspect of this comparison is that performance is always evaluated in two complementary ways: over the entire 24-hour horizon and, separately, restricting the metrics to daylight hours only. The latter perspective isolates the period in which PV production is non-zero and operationally relevant, and therefore provides a more informative view of forecasting quality for system operation and market participation.

A second objective is to study how the amount of textual conditioning influences the results. In particular, the thesis examines how forecasting accuracy changes when the prompt includes one, two, or three past examples from the same inverter, again comparing performance over full days and over daylight hours. This analysis aims to clarify whether providing additional few-shot exemplars systematically benefits the model or whether gains saturate, or even deteriorate, as the prompt becomes longer and more complex.

Finally, the thesis investigates how these effects vary across different plants, using plant-level error and association metrics to characterize cross-site heterogeneity in the gains obtained from text conditioning. Issues related to meteorological input uncertainty, plant and inverter heterogeneity, and missing-data treatment are documented in the data and methodology chapters and are discussed qualitatively in light of the empirical results, but they are not the focus of dedicated robustness experiments.

1.3 Background about Deep Learning

The aim of this section is to provide a technical key for readers who are not familiar with modern deep learning architectures but need to interpret the methodological choices in later chapters.

1.3.1 Deep learning architectures for sequence modelling

In this thesis, PV power forecasting is framed as a sequence modelling problem: given a history of observations and covariates, the goal is to predict a sequence of future values. Deep learning models for sequence modelling are built by stacking multiple simple layers into deep architectures: the input $\mathbf{x} \in R^F$ is transformed by the sequence of layers, each consisting of an affine transformation followed by a non-linear activation, until the final output layer is reached. All parameters are trained jointly end-to-end: given an input sequence and the corresponding target, the network produces a prediction, a loss function is computed, and gradients are propagated through all layers so that the parameters are updated by gradient-based optimisation methods such as stochastic gradient descent or Adam (Goodfellow et al., 2016).

The intermediate vectors $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \ldots$ are called *hidden representations* and they are the intermediate outputs of the internal layers , i.e. *hidden layers*. The space they belong to is often referred to as a *latent space*. The term *latent* emphasises that these representations are not given a priori but are learned from data: during training, the network adjusts θ so that the latent vectors become useful for the downstream task, e.g. forecasting. In this sense, deep learning can be understood as *representation learning*: instead of manually designing features, one lets the model discover useful transformations of the input data (Goodfellow et al., 2016).

Latent spaces are typically lower-dimensional than the input space and tend to organise data points according to task-relevant notions of similarity. For instance, in time-series forecasting, two different context windows that correspond to similar weather patterns and production regimes may be mapped to nearby points in latent space, even if they differ in many individual covariate values.

1.3.2 Embeddings and latent spaces

The notion of a latent space is particularly useful for *embeddings*, which are vector representations of discrete objects such as words, plant identifiers, or inverter identifiers. An embedding layer associates each element in a finite vocabulary $\mathcal{V}$ with a dense vector in R^d, where d is the embedding dimension. Formally, an embedding is a matrix

$$E \in R^{|\mathcal{V}| \times d},$$

whose i-th row E_i is the embedding of the i-th element in the vocabulary.

Embedding layers replace high-dimensional one-hot encodings with compact con-

tinuous vectors that can encode semantic or structural relationships. In natural language processing, word embeddings learned by transformer encoders tend to place semantically similar words close to each other in latent space. In this thesis, embeddings play a role both in the numerical baseline and in the text-conditioned model:

- in the numerical LSTM baseline (Section 4.3.1), plant and inverter identifiers are mapped to vectors in a latent space and concatenated to the numerical covariates. In this way, the encoder can learn plant-specific and device-specific offsets and interaction patterns directly from the data;
- in the Longformer-based text encoder, textual prompts are tokenised into subword units and each token is mapped to a vector in a high-dimensional latent space. The transformer refines these initial token embeddings through its attention layers, producing contextualised representations that depend on the entire prompt.

1.3.3 Encoder–decoder architectures for sequence modelling

Many forecasting problems can be viewed as mapping an input sequence

$$(\mathbf{x}_{t-T+1}, \ldots, \mathbf{x}_t)$$

of length T to an output sequence

$$(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_{t+H})$$

of length H. A common design pattern for such tasks is the *encoder–decoder* architecture:

1. The *encoder* processes the input sequence and produces one or more latent representations that summarise its information content.
2. The *decoder* takes these latent representations and generates the output sequence, possibly in an autoregressive fashion, one step at a time, or in a single multi-step prediction.

In classical sequence-to-sequence models for machine translation, for example, an RNN or transformer encoder reads the source sentence and the decoder generates the target sentence word by word, conditioned on the encoder output.

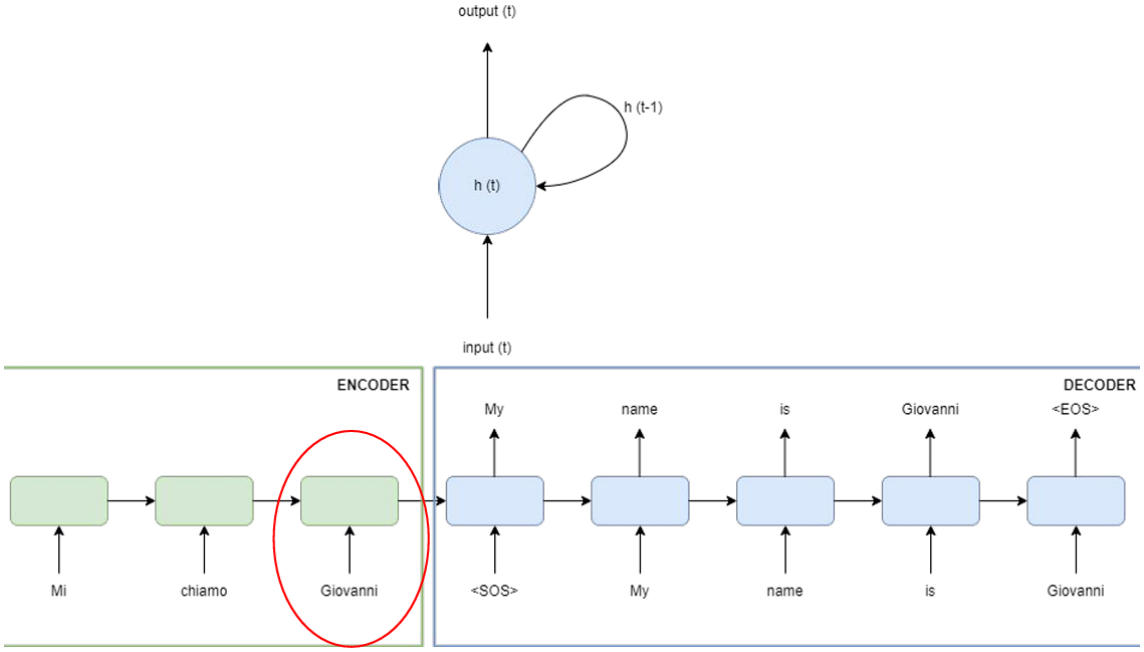


Figure 1.1: Encoder–decoder structure used for sequence-to-sequence modelling.

(Gurioli, 2024)

In this thesis, the encoder–decoder perspective appears in a slightly specialised form:

- The *numerical encoder* is a bidirectional LSTM that reads a multivariate context window of PV power and meteorological covariates and maps it to a fixed-dimensional latent vector.
- The *text encoder* is a Longformer model that reads a structured prompt and maps it to a separate latent vector.
- A*fusion block* combines these two latent vectors into a single joint representation.
- A simple *forecasting head*, a feed-forward network, plays the role of decoder: it maps the fused latent representation to a 24-dimensional vector representing the day-ahead production profile.

Conceptually, the encoders transform heterogeneous inputs, i.e. time series and text, into points in their respective latent spaces; the decoder operates exclusively in latent space, without direct access to the raw inputs. This separation is important:

it allows the architecture to change the encoder modules, e.g. replace the LSTM with a transformer, without altering the downstream forecasting head.

1.3.4 Recurrent neural networks and LSTMs

RNNs are designed to process sequential data by maintaining an internal hidden state that is updated over time. Given an input sequence $(\mathbf{x}_1, \ldots, \mathbf{x}_T)$ with $\mathbf{x}_t \in R^F$, a simple RNN defines the hidden state $\mathbf{h}_t \in R^H$ through the recurrence

$$\mathbf{h}_t = \phi\big(W_h \mathbf{h}_{t-1} + W_x \mathbf{x}_t + \mathbf{b}\big),$$

where W_h and W_x are trainable weight matrices, $\mathbf{b}$ is a bias vector, and $\phi(\cdot)$ is a non-linear activation function, typically tanh or ReLU. The same transformation is applied at every time step, and the parameters are shared across the entire sequence.

For forecasting, the final hidden state $\mathbf{h}_T$ or the full sequence $(\mathbf{h}_1, \ldots, \mathbf{h}_T)$ can be fed to a feed-forward head that outputs the desired prediction horizon. Training is performed by backpropagation through time, which unfolds the recurrence over the sequence and applies standard backpropagation.

A well-known difficulty with simple RNNs is the *vanishing/exploding gradient* problem: as gradients are propagated backwards through many time steps, they may decay towards zero or grow very large, making it hard to learn dependencies spanning long horizons (Hewamalage et al., 2021). LSTM networks address this limitation by introducing a more structured hidden state and *gating mechanisms* that explicitly regulate information flow.

An LSTM cell maintains two vectors at each time step:

- a *cell state* $\mathbf{c}_t$, which acts as a memory that can carry information across many time steps;
- a *hidden state* $\mathbf{h}_t$, which is exposed to the next layer or to the output head.

The update at time t is governed by three gates and a candidate state:

$$\begin{aligned}
\mathbf{i}_t &= \sigma\big(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + \mathbf{b}_i\big) && \text{(input gate)},\\
\mathbf{f}_t &= \sigma\big(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + \mathbf{b}_f\big) && \text{(forget gate)},\\
\mathbf{o}_t &= \sigma\big(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + \mathbf{b}_o\big) && \text{(output gate)},\\
\tilde{\mathbf{c}}_t &= \tanh\big(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1} + \mathbf{b}_c\big) && \text{(candidate state)},\\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t,\\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t),
\end{aligned}$$

where $\sigma(\cdot)$ is the logistic sigmoid, $\odot$ denotes element-wise multiplication, and all $W_\cdot$, $U_\cdot$, $\mathbf{b}_\cdot$ are trainable parameters.

Intuitively:

- the forget gate $\mathbf{f}_t$ decides which components of the previous memory $\mathbf{c}_{t-1}$ should be retained;
- the input gate $\mathbf{i}_t$ controls how much of the candidate information $\tilde{\mathbf{c}}_t$ is written into memory;
- the output gate $\mathbf{o}_t$ determines how much of the updated memory $\mathbf{c}_t$ is exposed in the hidden state $\mathbf{h}_t$.

This structure makes it easier for the optimiser to preserve and update relevant information over longer horizons and explains why LSTMs have become standard baselines for time-series forecasting in energy systems (Hochreiter and Schmidhuber, 1997; Hewamalage et al., 2021). In this thesis, as described in Chapter 4.3, a bidirectional LSTM encoder is used as the numerical baseline to summarise 168 hours of plant-level context into a fixed-dimensional representation, which is then mapped to the 24-hour forecast horizon.

1.3.5 Transformers and the self-attention mechanism

Transformers provide an alternative way to model sequences that does not rely on recurrence. Instead of updating a hidden state step by step, transformers use *self-attention* layers in which each position in the sequence directly interacts with all other positions. This design allows the model to capture long-range dependencies while processing all time steps in parallel (Vaswani et al., 2017; Lin et al., 2022).

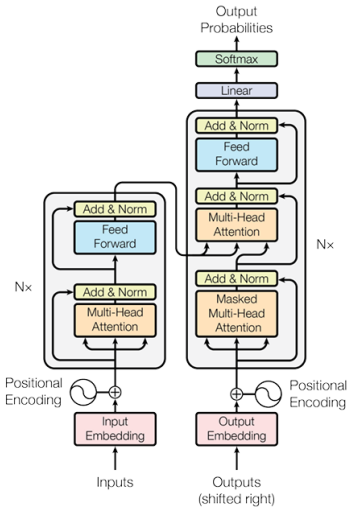
Consider an input sequence represented as a matrix

$$X \in R^{T \times d},$$

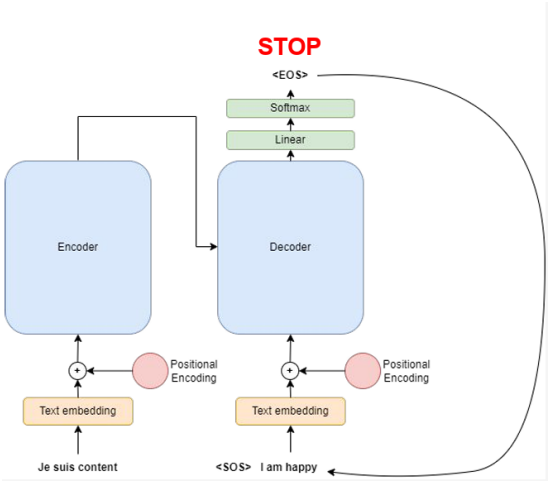
where T is the sequence length and d is the embedding dimension. The self attention mechanism operates on three vectors: Queries Q, keys K and values V, where $Q, K, V \in R^{d \times n}$, and d is the dimension of the feature space and n is the sequence lenght. The *scaled dot-product attention* operation is defined as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V.$$

At a conceptual level, each row of $QK^\top$ contains similarity scores between one position and all others; applying the softmax row-wise turns these scores into non-negative weights that sum to one. The product with V returns a weighted average of the value vectors, so that the representation at each position becomes an adaptive summary of the whole sequence. In other words, self-attention assigns to each time step a weighted combination of all other elements in the sequence, and the weights are learned as part of the model. This formulation enables direct interactions between distant time steps, overcoming the limitations associated with vanishing gradients, compressed hidden states, and fixed temporal kernels that characterise recurrent and convolutional models.



(a) Transformer architecture(Vaswani et al. (2017))



(b) Example of transformer token generation(Gurioli (2024))

Figure 1.2: Transformer encoder–decoder architectures.

In practice, transformers use *multi-head attention*, where the attention computation is replicated H times with different projection matrices:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(H_1, \ldots, H_H)\, W^O,$$

with

$$H_h = \text{Attention}(Q_h, K_h, V_h)$$

for each head h and a learned output projection W^O. Different heads can specialize in different dependency patterns, e.g. short versus long temporal ranges, or different

subsets of features (Vaswani et al., 2017).Overall, multi-head attention further enhances representational capacity by allowing the model to attend simultaneously to multiple temporal patterns, effectively capturing interactions across different time scales.

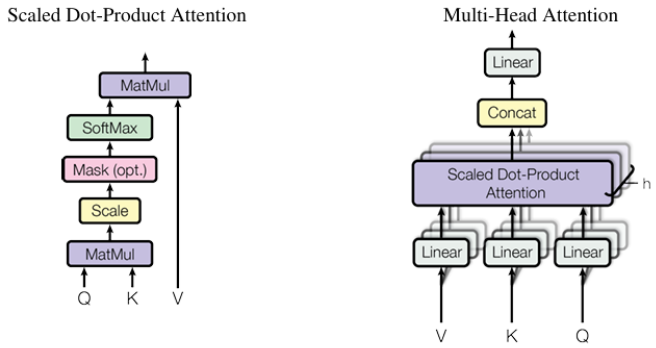


Figure 1.3: Scaled dot-product versus multi-head attention.

(Vaswani et al., 2017)

These features have proven particularly effective in multi-horizon forecasting settings, where relevant information may be distributed unevenly across the historical sequence

Because self-attention is permutation-invariant, transformers must encode positional information explicitly. This is typically done by adding a positional encoding P_t to each input embedding x_t. In the original formulation, these encodings are deterministic sinusoidal functions of the position index; many modern variants use learned positional embeddings instead (Vaswani et al., 2017; Lin et al., 2022). In both cases, the goal is to provide the model with enough information to distinguish between, for example, "one hour ago" and "one week ago".

1.3.6 Large Language Models architecture and training paradigm

LLMs are transformer-based architectures trained at scale using self-supervised learning objectives, most commonly next-token prediction. From an architectural perspective, modern LLMs rely almost exclusively on transformer variants, typically in a decoder-only configuration for autoregressive modeling. Each transformer

layer alternates self-attention mechanisms, which compute content-dependent interactions across sequence positions, with position-wise feed-forward networks that introduce non-linear feature transformations. Multi-head attention allows the model to represent multiple dependency patterns in parallel, while residual connections and normalization layers facilitate stable optimization at depth (Lin et al., 2022).

The training paradigm of LLMs is centered on large-scale pretraining over massive corpora, enabling the model to learn general-purpose representations that can be transferred to downstream.

LLMs can be adapted to numerical regression and time-series forecasting through two broad paradigms: *fine-tuning* and *in-context learning*. *Fine-tuning* involves updating model parameters using labeled data and task-specific loss functions, such as mean absolute error or mean squared error, whereas *in-context learning* keeps model parameters fixed and conditions predictions on example input–output pairs provided within the prompt. These two paradigms exhibit distinct behaviors under distribution shift, data scarcity, and sequence-length constraints (Brown et al., 2020; Bommasani et al., 2021).

1.4 Background about Photovoltaic Plants

PV plants convert incoming solar radiation into electrical energy through a sequence of physical transformations that connect atmospheric conditions to grid-injected power. Even though the forecasting task addressed in this thesis is entirely data-driven, a minimal understanding of this conversion chain is helpful to interpret the choice of target variable, the construction of covariates, and the sources of uncertainty that affect model performance. In modern power systems, utility-scale and distributed PV installations have expanded rapidly over the last decade, becoming a central pillar of decarbonisation strategies and significantly increasing the need for accurate short-term production forecasts (International Renewable Energy Agency (IRENA), 2019; Iheanetu, 2022).

At a high level, a grid-connected PV plant comprises three main building blocks. First, PV modules convert plane-of-array solar irradiance into direct current (DC) electricity via the PV effect in semiconductor junctions. Second, one or more inverters transform DC power into alternating current (AC) at grid voltage and frequency. Third, balance-of-system elements, such as mounting structures, e.g. fixed-tilt or tracking, cabling, transformers, protection devices and monitoring equipment, complete the installation and condition the flow of power to the point of connection. From the perspective of forecasting and system operation, the most relevant elec-

trical quantity is the AC power measured at inverter level, since it represents the actual power injected into the grid and is the basis for market participation, imbalance settlement and system security assessments. For this reason, the dataset used in this thesis adopts inverter-level AC active power as the target variable; the corresponding measurement infrastructure and data structure are documented in detail in Chapter 3.

1.4.1 The role of the inverter

The inverter is the interface between the DC side of the PV array and the AC grid, and its behaviour strongly shapes the relationship between irradiance and delivered power. Beyond performing DC–AC conversion, modern PV inverters implement maximum power point tracking (MPPT), i.e. a control strategy that continuously adjusts the operating voltage and current of the array to keep the operating point close to the maximum of the current–voltage power curve under varying irradiance and cell temperature. As a consequence, the mapping from incident irradiance to AC power is not determined solely by module characteristics, but also by the MPPT algorithm, efficiency curves and control logic embedded in the inverter.

In addition, grid codes in many jurisdictions now require smart inverters to provide grid-support functionalities such as reactive power control, voltage and frequency support and ride-through capabilities during disturbances, particularly as PV penetration increases (Gui et al., 2024). These functions are typically implemented through additional control loops that can modify the active power set-point or temporarily curtail generation in order to respect operational constraints at the point of connection. Furthermore, when the DC capacity of the array exceeds the rated AC power of the inverter, episodes of *clipping* occur: during high-irradiance conditions, the inverter limits its output at its nominal rating, flattening the upper part of the daily power curve. All these effects introduce non-linearities and plant-specific patterns that are more naturally captured at the level of individual inverters than at a fully aggregated plant level. This consideration motivates the focus on inverter-level data in the empirical analysis of this thesis, and connects with the modelling choices described in Chapter 4.

1.4.2 Irradiance and solar geometry

The primary exogenous driver of PV production is solar irradiance, but the effective irradiance on the module plane depends on both astronomical and atmospheric factors. At any given time, the position of the sun in the sky can be described

by its zenith and azimuth angles, which are deterministic functions of geographic location, date and time. Combined with the tilt and azimuth of the PV modules, solar position determines the angle of incidence of beam radiation and therefore the magnitude of the direct component of irradiance on the plane of array. Atmospheric conditions, including clouds, aerosols and water vapour, modulate the partition of global irradiance into direct and diffuse components and introduce short-term variability and ramps, especially under broken cloud conditions.

In practice, the irradiance information available to a forecasting model may come in different forms. In some settings, high-quality measurements of global horizontal irradiance or plane-of-array irradiance are available from on-site sensors. In others, irradiance is reconstructed from numerical weather prediction outputs or reanalysis products and converted to plane-of-array values through transposition models and clear-sky corrections (Solar Resource Solutions, 2015). In this thesis, irradiance-related variables are represented through a combination of meteorological inputs and physically derived features that account for solar geometry and plant configuration. The specific variables, data sources and transformation steps used to construct these irradiance proxies are described in Section 3.3

1.4.3 Nominal power and normalisation

PV plants differ substantially in size, layout and inverter configuration. The *name-plate* or nominal power of a plant is usually defined as the sum of the module ratings under standard test conditions (STC), i.e. an irradiance of $1000\,\mathrm{W\,m^{-2}}$, a cell temperature of $25\,^{\circ}\mathrm{C}$ and a reference spectral distribution (Jordan et al., 2016). In real operation, output deviates from this ideal benchmark due to temperature effects, optical losses, module mismatch, soiling and ageing. In addition, the DC capacity of the array and the AC rating of the inverters are often intentionally mismatched, leading to plant-specific DC/AC ratios that influence clipping behaviour and normalisation choices in data analysis.

When working with multi-plant or multi-inverter datasets, it is therefore convenient to normalise power measurements by an appropriate nominal capacity. This yields a dimensionless quantity, often interpreted as a per-unit utilisation factor, that preserves temporal dynamics and relative variations while reducing trivial scale differences between units. Normalisation also facilitates the transfer of models across inverters and plants of different sizes. The specific normalisation strategy adopted in this thesis, including the choice between DC and AC nominal ratings and the treatment of missing or inconsistent metadata, is described formally in Section 3.2.1.

1.4.4 Sources of variability and uncertainty

PV generation exhibits variability across a wide range of time scales. Over the course of a year, deterministic seasonal patterns arise from the evolution of solar geometry and the climatology of cloudiness at the site. Within a day, the diurnal cycle of solar position produces a characteristic bell-shaped envelope for clear-sky production, upon which clouds and other atmospheric phenomena superimpose stochastic fluctuations and ramps. At sub-hourly scales, the passage of individual cloud cells over the plant can induce sharp power ramps, particularly in convective regimes. Superimposed on these exogenous drivers, plant- and inverter-specific factors, such as partial shading, string outages, degradation or temporary curtailments,generate additional idiosyncratic structure in the time series.

From a forecasting viewpoint, these features make PV power a typical example of a sequence with both strong deterministic components and substantial exogenous uncertainty. Even when sophisticated statistical or machine learning models are employed, forecast skill is ultimately constrained by the quality, spatial resolution and lead time of the meteorological inputs used as covariates (Iheanetu, 2022). In other words, errors and biases in the underlying weather information impose an upper bound on achievable accuracy, and this bound depends on site characteristics and forecasting horizon. A more detailed discussion of forecasting models, error metrics and the role of meteorological inputs is provided in Chapter 2, while Chapter 3 documents the specific covariates and data sources used in this work. Here the key point is that PV forecasting should be understood as a context-dependent sequence modelling problem, shaped jointly by plant heterogeneity, solar geometry and the stochastic nature of the atmosphere.

CHAPTER 2

State of the art

2.1 Photovoltaic power forecasting: applications and challenges

The rapid expansion of PV generation over the past decade has fundamentally altered the operational and planning paradigms of modern power systems. As installed PV capacity continues to grow across both transmission and distribution networks, system operators and market participants increasingly rely on accurate short-term power forecasts to ensure reliable operation, efficient resource allocation, and secure integration of variable renewable energy sources. In this context, PV power forecasting has transitioned from a secondary planning tool to a core component of power system decision-making (International Energy Agency, 2023b; Ahmed et al., 2020).

At the system level, short-term PV forecasts are used to support unit commitment and economic dispatch decisions, reserve dimensioning, congestion management, and real-time balancing operations. In electricity markets, forecast accuracy directly affects bidding strategies, price formation, and imbalance settlement mechanisms, particularly in systems with high shares of variable renewable energy. At the distribution level, where PV penetration is often concentrated and highly decentralized, accurate forecasts are increasingly required to manage voltage regulation, network congestion, and local flexibility resources (Ahmed et al., 2020).

PV power generation is inherently variable, as it is driven by exogenous meteorological conditions and solar geometry. At short forecasting horizons, PV output may exhibit pronounced fluctuations caused by cloud formation, movement, and dissipation, leading to rapid changes in solar irradiance over time scales ranging from minutes to hours. Empirical studies consistently identify this high-frequency variability as one of the dominant sources of uncertainty in PV forecasting, particularly under convective and partially cloudy atmospheric regimes (Wolff et al., 2016). These dynamics pose significant challenges for operational forecasting, as small errors in irradiance prediction can translate into large deviations in power output over

short time intervals.

Beyond atmospheric-driven variability, the relationship between meteorological inputs and PV power output is characterized by strong non-linearities. While global horizontal irradiance represents the primary explanatory variable, the conversion of solar energy into electrical power is modulated by a range of interacting factors, including module temperature, angle of incidence, system losses, inverter behavior, and plant-specific characteristics. The combined effect of these factors generates a complex and non-linear mapping between inputs and output, which varies across plants and operating conditions. As a consequence, simple linear or parametric models often struggle to capture the underlying dynamics, particularly during rapidly changing weather conditions (Benali et al., 2019; Ahmed et al., 2020).

Forecasting errors in PV power production have direct operational and economic implications. In market environments, deviations between forecasted and realized generation result in imbalance costs and inefficient dispatch outcomes, which become increasingly significant as renewable penetration rises. From a system operation perspective, inaccurate forecasts may lead to suboptimal reserve allocation, increased curtailment of renewable generation, and voltage regulation challenges in distribution networks with substantial distributed PV capacity (Ahmed et al., 2020; Wolff et al., 2016). These effects highlight that forecasting accuracy is not merely a statistical objective, but a critical determinant of system efficiency and reliability.

A further structural challenge arises from the nature of the meteorological data used for PV power forecasting. Many state-of-the-art approaches rely on numerical weather prediction models or reanalysis products that provide irradiance and weather variables at spatial and temporal resolutions that may not fully capture the local conditions experienced by individual PV plants. This spatial and temporal mismatch introduces an irreducible source of uncertainty, effectively imposing an upper bound on achievable forecasting accuracy regardless of the modeling approach employed (Wolff et al., 2016). In practice, this implies that improvements in model complexity cannot fully compensate for limitations in input data quality.

Taken together, the combined effects of atmospheric-driven variability, non-linear input–output relationships, and meteorological input uncertainty make PV power forecasting a fundamentally challenging task. Despite substantial progress in machine learning and deep learning methods after 2018, recent literature consistently emphasizes that forecasting performance is increasingly constrained by data availability and inherent uncertainty rather than by model capacity alone (Ahmed et al., 2020). These considerations motivate the exploration of advanced sequence-based modeling approaches, including transformer-based architectures and LLMs–based

frameworks, which aim to better capture complex temporal dependencies and contextual information in PV power forecasting.

2.2 Methods for time series forecasting in energy systems

Time series forecasting plays a central role in modern energy systems, supporting operational decisions, market clearing mechanisms, and long-term infrastructure planning. The growing penetration of variable renewable energy sources has substantially increased both the dimensionality and the complexity of forecasting problems, as energy production patterns are increasingly influenced by exogenous drivers such as meteorological conditions and human behaviour. In this context, forecasting methods must cope with non-stationarity, non-linear relationships, and multiple interacting temporal scales, motivating a progressive evolution from classical statistical models toward more flexible, data-driven approaches (Ahmed et al., 2020; Hong et al., 2015).

2.2.1 Statistical and classical approaches

Statistical forecasting methods have historically represented the backbone of time series analysis in energy systems. Classical approaches such as autoregressive integrated moving average (ARIMA) models, exponential smoothing techniques, and linear regression models with exogenous inputs (ARX) have been widely adopted due to their mathematical transparency, relatively low computational requirements, and well-understood statistical properties (Nowotarski and Weron, 2018). In many operational settings, these methods continue to serve as reference benchmarks against which more complex models are evaluated.

At their core, classical time series models rely on the assumption that the future evolution of a process can be expressed as a linear function of its past values and, when applicable, a limited set of external drivers. For instance, an ARX model for PV power production can be expressed as

$$P_t = \sum_{i=1}^{p} \phi_i P_{t-i} + \sum_{j=1}^{q} \beta_j X_{t-j} + \varepsilon_t,$$

where P_t denotes power output at time t, X_t represents exogenous variables such as irradiance or temperature, and ε_t is a stochastic error term. Similar linear formula-

tions underlie ARIMA and exponential smoothing models, with differences arising in how trends, seasonality, and stochastic components are modeled.

When the underlying process exhibits relatively stable dynamics and weak non-linearities, such models can deliver competitive forecasting performance while retaining a high degree of interpretability. In particular, their parameter estimates often admit a clear physical or economic interpretation, which is appealing in operational contexts where transparency and explainability are valued. Moreover, the modest data requirements of classical models make them suitable for applications where historical records are limited or incomplete.

However, the applicability of classical statistical approaches becomes increasingly constrained in environments characterized by strong non-linearities, non-stationarity, and rapidly changing external conditions. PV power generation represents a paradigmatic example of such a setting. The relationship between meteorological variables and power output is highly non-linear and time-varying, especially under partially cloudy conditions where irradiance can fluctuate abruptly within short time intervals. Empirical studies consistently report that linear models struggle to adapt to these dynamics, leading to systematic forecast errors during periods of high atmospheric variability (Wolff et al., 2016).

Another limitation of classical models lies in their handling of temporal structure. While seasonal components and lagged dependencies can be incorporated explicitly, the choice of lag order and functional form is typically fixed ex ante and may fail to capture evolving temporal patterns. In the presence of regime shifts, structural breaks, or multi-scale temporal dependencies, the rigidity of these specifications can result in poor generalization beyond the calibration period.

As a consequence, classical statistical approaches are increasingly regarded as baseline or benchmarking tools in PV power forecasting rather than as state-of-the-art solutions. Their limitations in capturing non-linear input–output relationships and adapting to rapidly changing conditions motivate the adoption of more flexible modeling paradigms, such as machine learning and deep learning methods, which relax many of the restrictive assumptions underlying traditional statistical models. These limitations highlight the difficulty of relying on fixed linear structures in highly variable and weather-driven energy systems, motivating the adoption of more flexible, data-driven modeling approaches.

2.2.2 Machine learning approaches

Machine learning approaches have been widely adopted in energy forecasting to overcome the structural limitations of linear statistical models. By relaxing assumptions on functional form, these methods allow for flexible, data-driven representations of the mapping between inputs and outputs, making them particularly suitable for systems characterized by non-linear dynamics and interacting drivers (Benali et al., 2019; Ahmed et al., 2020). In contrast to classical approaches, model complexity in machine learning is primarily controlled by data and regularization rather than by explicit parametric assumptions.

In PV power forecasting, machine learning models are typically trained on feature-engineered representations that combine historical power measurements with meteorological variables, calendar effects, and lagged observations. Techniques such as support vector regression, random forests, and gradient boosting machines are able to capture non-linear dependencies between irradiance, temperature, and power output without relying on explicit physical modeling. Empirical evidence shows that, when high-dimensional meteorological inputs are available, these methods consistently outperform linear statistical baselines in terms of point forecasting accuracy (Benali et al., 2019; Ahmed et al., 2020).

A key conceptual advance introduced by machine learning lies in its ability to model interaction effects among explanatory variables. For instance, the impact of solar irradiance on power output may depend on temperature, time of day, or recent atmospheric conditions. Ensemble-based methods and kernel-based models can naturally represent such interactions, providing a more expressive approximation of the underlying input–output relationship than linear models. This property is particularly relevant under partially cloudy conditions, where PV generation exhibits abrupt and non-linear responses to rapidly evolving meteorological drivers.

Despite these advantages, most machine learning approaches retain a fundamental limitation in their treatment of temporal structure. Temporal dependencies are typically encoded through manually selected lagged features and fixed-size input windows, which are defined ex ante and remain unchanged during training. As a result, the model's capacity to exploit information distributed across longer or variable temporal horizons is inherently constrained (Wolff et al., 2016). This reliance on handcrafted temporal representations introduces sensitivity to design choices and limits adaptability under non-stationary conditions.

Moreover, once features are constructed, standard machine learning models effectively treat observations as conditionally independent samples. Temporal ordering

beyond the selected lag structure is not explicitly modeled, and dynamic context is only indirectly represented through engineered inputs. In forecasting tasks where temporal coherence, regime changes, and evolving patterns play a central role, such as PV power generation under variable weather conditions, this limitation can lead to reduced robustness and degraded performance during periods of high variability.

These considerations suggest that, while machine learning methods represent a substantial improvement over classical statistical models in terms of flexibility and predictive power, they do not fully resolve the challenges posed by sequential dependence and non-stationarity. This motivates the transition toward deep learning and sequence-based architectures, which aim to learn temporal representations directly from data rather than imposing them through predefined feature structures. While machine learning models substantially improve flexibility and predictive accuracy, their reliance on handcrafted temporal features and fixed input representations limits their ability to fully exploit sequential structure, motivating the development of deep learning architectures specifically designed for sequence modeling.

2.2.3 Deep learning and sequence-based models

Deep learning approaches represent a conceptual shift with respect to traditional machine learning methods by enabling the joint learning of feature representations and temporal dependencies directly from raw or minimally processed time series data. Rather than relying on handcrafted features and fixed lag structures, deep neural networks aim to learn hierarchical representations that encode both short-term dynamics and longer-term temporal patterns. (Kong et al., 2019; Lim et al., 2021).

RNNs constitute one of the earliest deep learning architectures specifically designed for sequential data. In their general formulation, RNNs update a latent hidden state recursively as new observations become available, allowing information from past inputs to influence future predictions. To address optimization issues related to vanishing and exploding gradients, gated architectures such as LSTM networks and gated recurrent units (GRU) have been introduced. These models incorporate gating mechanisms that regulate information flow across time, improving the ability to capture longer-range temporal dependencies (Kong et al., 2019).

In the context of PV power forecasting, LSTM- and GRU-based models have demonstrated strong performance in capturing temporal correlations and adapting to non-linear dynamics induced by meteorological variability. Their ability to integrate historical power output with weather-related inputs has made them a standard

benchmark in the post-2018 PV forecasting literature (Benali et al., 2019; Ahmed et al., 2020). Nevertheless, the inherently sequential nature of recurrent architectures limits parallelization during training and inference, leading to scalability challenges for high-frequency or large-scale datasets. Moreover, empirical evidence suggests that their effective memory may degrade when modeling very long sequences or multi-horizon forecasting tasks (Lim et al., 2021).

Convolutional neural networks (CNNs) provide an alternative mechanism for modeling temporal structure by applying convolutional filters to time series data. Temporal convolutions are well suited to capturing local patterns and short-term fluctuations, and CNN-based models have been successfully applied to energy forecasting tasks, including PV power prediction (Benali et al., 2019). In practice, CNNs are often combined with recurrent architectures in hybrid CNN–RNN models, where convolutional layers extract local temporal features and recurrent layers model longer-term dependencies.

Despite their empirical success, both recurrent and convolutional deep learning models rely on fixed receptive fields and implicit representations of temporal relationships. In CNN-based architectures, the temporal context is constrained by kernel size and network depth, while in recurrent models information is compressed into a latent state that may not fully preserve long-range dependencies. These architectural constraints can limit flexibility in highly heterogeneous forecasting settings, particularly when short-term stochastic fluctuations coexist with longer-term deterministic patterns, as is the case in PV power generation (Lim et al., 2021). These limitations have spurred growing interest in alternative sequence modeling paradigms that can better capture long-range dependencies, scale efficiently, and flexibly adapt to heterogeneous temporal patterns, leading to the development of attention-based and transformer architectures.

2.2.4 Toward attention-based and transformer models

Recent advances in time series forecasting have increasingly shifted attention toward attention-based architectures, which depart from the recursive and convolutional paradigms by explicitly modeling dependencies between different time steps through adaptive weighting mechanisms. Rather than processing observations sequentially or within fixed receptive fields, attention-based models compute pairwise interactions across the entire input sequence, allowing the relevance of past observations to be learned dynamically as a function of the forecasting task (Lim et al., 2021).

Originally developed for natural language processing, transformers have been successfully adapted to time series forecasting tasks, including applications in energy systems. Their reliance on self-attention mechanisms allows for improved scalability and parallelization during training and inference, as all time steps within a sequence can be processed simultaneously (Zhou et al., 2021).

In addition to computational advantages, transformer-based models offer increased flexibility in handling heterogeneous inputs and multi-horizon forecasting objectives. By embedding multiple covariates, into a shared latent space, transformers can jointly model cross-feature interactions and temporal dependencies. This property is particularly valuable in energy forecasting settings, where production patterns are shaped by both deterministic components, e.g. solar geometry, and stochastic drivers, e.g. cloud dynamics.

Empirical evidence reported in recent studies indicates that transformer-based architectures can outperform recurrent and convolutional deep learning models in scenarios characterized by complex temporal patterns and long forecasting horizons (Lim et al., 2021; Zhou et al., 2021). Their ability to selectively attend to relevant portions of the historical sequence allows them to adaptively exploit information at different time scales, an essential capability in highly variable renewable energy systems.

In the context of PV power forecasting, attention-based and transformer models represent a natural progression in the methodological landscape. They provide a flexible framework for integrating meteorological information, historical power measurements, and contextual features while mitigating many of the structural limitations associated with earlier sequence-based architectures. Overall, the progression from classical statistical models to machine learning, deep learning, and attention-based architectures can be interpreted as a shift from predefined functional forms towards data-driven models that learn their own internal representations and are able to capture more detailed temporal structure in PV time series. This progression provides the methodological foundation for the transformer-based and LLMs–based forecasting approaches examined in the following sections.

2.3 Transformer-based models for time series forecasting

The introduction of transformer architectures has marked a significant shift in sequence modeling, offering a flexible alternative to recurrent and convolutional ap-

proaches traditionally employed in time series forecasting. Building on attention mechanisms, transformers decouple temporal dependency modeling from recursive computation, enabling direct interactions between distant observations within a sequence. The transformer architecture is particularly relevant for energy forecasting applications, where meaningful temporal dependencies may arise across heterogeneous time scales (Lim et al., 2021; Zhou et al., 2021). In time series applications, transformer models typically operate on fixed-length input sequences that are embedded into a high-dimensional latent space. Since attention mechanisms do not intrinsically encode temporal order, positional encoding is introduced to preserve information about the relative or absolute position of observations within the sequence.e(Lim et al., 2021).

2.3.1 The "Photovoltaic Power Forecasting: A Transformer-Based Framework" paper

A representative application of transformer architectures to PV power forecasting is provided by the framework proposed in "Photovoltaic Power Forecasting: A Transformer-Based Framework" (Piantadosi et al., 2024). The study is motivated by the observation that PV generation time series combine deterministic components, driven by solar geometry and seasonal cycles, with stochastic fluctuations induced by rapidly evolving atmospheric conditions. A key empirical finding of their review is that most state-of-the-art PV forecasting models rely on locally measured irradiance as an input feature. While this effectively bypasses the complexity of airmass and cloud modelling, it also introduces an important limitation: the need for plant-level irradiance measurements reduces the spatial applicability of the models and may constrain the forecasting horizon. Moreover, many studies use data from a single plant, a single season, or from a small number of benchmark sites, such as DKASC, often combined with coarse or approximate weather information. As a consequence, forecasting performance is sometimes evaluated on narrow operating regimes and may not generalise well to unseen plants, locations, or times of the year. The authors also note that a substantial share of the literature trains models solely on past PV production, without explicit meteorological inputs, which can reduce robustness under changing weather regimes.

Their approach adopts a sequence-to-sequence formulation in which historical PV power measurements are combined with exogenous meteorological variables to generate multi-step-ahead forecasts. Input sequences are constructed by concatenating power output observations with weather-related features and embedding them into

a shared latent representation. This design allows the model to learn complex, nonlinear interactions between meteorological conditions and power production in a purely data-driven manner, while avoiding the requirement of local irradiance measurements through the use of a modelled plane-of-array irradiance signal built from numerical weather prediction data.

A key methodological contribution of the framework lies in its use of self-attention to dynamically identify the most informative portions of the historical sequence for each forecasting horizon. Rather than compressing past information into a single latent state, as in recurrent architectures, the transformer explicitly evaluates pairwise interactions between time steps. This enables the model to emphasize observations that are most relevant for prediction, such as recent measurements under similar atmospheric regimes or temporally distant patterns associated with periodic solar behavior.

Positional encoding plays a crucial role in preserving temporal structure within the attention mechanism. By injecting information about time ordering into the model, the framework maintains sensitivity to temporal relationships despite the absence of recurrence. The resulting architecture supports parallel computation over the entire input sequence, improving training efficiency and scalability. Empirical results reported in the study indicate that the transformer-based framework achieves competitive performance compared to established deep learning baselines, particularly under conditions of elevated variability.
The authors consider two global numerical weather prediction (NWP) models: the Global Forecast System (GFS) and the European Centre for Medium-Range Weather Forecasts (ECMWF) as well as two data services built on top of NWP and observational products, OpenWeather and Open-Meteo (Piantadosi et al., 2024). GFS is a free global model provided by NOAA, delivering forecasts four times per day with a spatial resolution of about 27 km and a 3-hour temporal resolution up to 10 days ahead. ECMWF provides partially free global forecasts twice per day, with finer spatial resolution, around 14 km, and 1-hour temporal resolution, using a non-hydrostatic formulation in which altitude replaces pressure as the vertical coordinate.

OpenWeather offers a commercial Application Programming Interfaces (APIs) that fuses data from weather stations, satellite observations, radar and multiple global NWP models. According to the reference study, its outputs are updated every 10 minutes, with a spatial resolution of about 500 m and 1-hour temporal resolution up to 4 days ahead. Open-Meteo, by contrast, is an open-source data aggregator providing a free API for non-commercial use.. They show that the transformer

trained on Open-Meteo features achieves the best overall performance, with a day-ahead MAPE of approximately 2.2 % and competitive MAE and nRMSE values relative to the recent deep learning literature on PV forecasting(Piantadosi et al., 2024).

2.3.2 Key ideas and limitations of the baseline framework

The transformer-based framework introduced in the previous subsection encapsulates several methodological ideas that are central to recent advances in time series forecasting. First, the explicit modeling of temporal dependencies through self-attention enables adaptive weighting of historical observations, allowing the model to focus on the most relevant information for each forecasting horizon. This property is particularly advantageous in PV power forecasting, where the relevance of past observations may vary substantially depending on evolving meteorological conditions.

Second, the framework naturally supports the integration of heterogeneous inputs within a unified representation space. By jointly embedding power measurements and meteorological variables, the model can exploit cross-feature interactions without extensive manual feature engineering. This flexibility aligns with modern forecasting pipelines that rely on high-dimensional exogenous information and seek to minimize reliance on handcrafted temporal representations (Lim et al., 2021).

Despite these strengths, the baseline transformer framework also exhibits several limitations that are important to highlight. Most notably, the model remains purely data-driven and does not explicitly encode physical constraints related to PV generation processes. As a consequence, forecasting performance is inherently bounded by the quality, spatial resolution, and uncertainty of the meteorological inputs. Errors in weather forecasts or mismatches between grid-level meteorological data and plant-level conditions propagate directly through the model, limiting achievable accuracy (Piantadosi et al., 2024).

In addition, the framework focuses on deterministic point forecasting and does not explicitly address uncertainty quantification, which is increasingly relevant for operational decision-making in power systems.

These considerations suggest that, while transformer-based models represent a substantial methodological advance over earlier sequence-based approaches, further extensions are required to enhance robustness, efficiency, and contextual awareness. In particular, they motivate the exploration of architectures that leverage large-scale

pretraining and more flexible conditioning mechanisms.

2.4 Large Language Models for numerical and time-series tasks

LLMs are transformer-based foundation models trained at scale using self-supervised objectives, typically next-token prediction on massive corpora (Brown et al., 2020; Bommasani et al., 2021). While originally developed for natural language, their relevance to numerical and time-series tasks stems from two methodological facts: (i) transformers are general-purpose sequence models and can operate on any modality that can be serialized into tokens, and (ii) large-scale pre-training can yield transferable representations and implicit algorithmic behaviors that can be elicited through prompting or adapted through supervised updates (Bommasani et al., 2021; Wei et al., 2022).

For time-series forecasting and regression, LLM-based approaches should not be interpreted as "language solving numbers", but rather as a family of adaptation strategies that exploit (i) the attention mechanism as a flexible dependency model across positions, and (ii) pretraining at scale as a way to encode reusable priors over patterns, e.g. trend, seasonality, regime changes, once data are represented in a token space (Das et al., 2024; Ansari et al., 2024; Gruver et al., 2023).

Empirical evidence shows that increasing model size, data volume, and compute tends to improve downstream performance, following predictable scaling relationships (Kaplan et al., 2020). Subsequent work has refined this view by demonstrating that compute-optimal training requires scaling data alongside model parameters, as highlighted by the Chinchilla analysis (Hoffmann et al., 2022). These results are particularly relevant for numerical and time-series tasks, as the ability of a model to generalize from context and exhibit implicit computational behavior is strongly shaped by the training regime and scale.

Applying LLMs to numerical domains, however, introduces representation challenges that are largely absent in natural language processing. Standard subword tokenizers are not designed to handle numerical precision or arithmetic consistency, and different tokenizations may induce discontinuities in the representation of nearby values. In time-series forecasting, choices related to discretization, scaling, and the mapping between continuous values and tokens can have a dominant impact on predictive accuracy and calibration (Gruver et al., 2023; Ansari et al., 2024). As a result, LLM-style approaches to time-series forecasting often redesign tokenization

strategies explicitly, for example by quantizing real-valued observations into fixed vocabularies and training models using cross-entropy objectives on token sequences (Ansari et al., 2024).

Despite these challenges, there is growing evidence that large transformer models can support non-trivial quantitative reasoning when trained or adapted using suitable technical corpora. The Minerva project, for instance, demonstrates that continued training on mathematical and scientific data substantially improves performance on quantitative benchmarks (Lewkowycz et al., 2022). Importantly, such results do not imply that LLMs function as reliable calculators by default; rather, they indicate that scale and domain-appropriate data can shift learned representations toward behaviors that are more compatible with structured numerical reasoning.

A critical distinction relevant for this thesis is between general-purpose LLMs trained primarily on text and time-series foundation models trained directly on large corpora of numerical sequences. While both share the transformer architecture and language-model-style training objectives, time-series foundation models differ in their data modality and tokenization strategies. Examples include TimesFM, a decoder-only foundation model designed for strong zero-shot time-series forecasting performance (Das et al., 2024), Chronos, which employs explicit scaling and quantization for probabilistic forecasting (Ansari et al., 2024), and Lag-Llama, which leverages lag-based covariates to enable cross-domain transfer (Rasul et al., 2023). These models provide empirical evidence that language-model-style pretraining can be effective for forecasting tasks when representations and training data are aligned with the numerical nature of the problem.

In the context of PV power forecasting, these considerations suggest that the central question is not whether LLMs can handle numerical data in principle, but whether the chosen representation, pretraining regime, and adaptation strategy jointly yield models that are robust to meteorological uncertainty and non-stationarity. This perspective motivates the methodological choices explored in the subsequent sections of this thesis.

2.4.1 Fine-tuning vs in-context learning for regression tasks

Fine-tuning can be interpreted as supervised specialization of a pretrained representation. In the context of time-series forecasting, recent work explores fine-tuning LLM backbones or LLM-like transformers after mapping numerical sequences into

embedding space. Approaches such as TIME-LLM explicitly investigate fine-tuning strategies that adapt pretrained models to forecasting tasks (Jin et al., 2023). By updating model parameters, fine-tuning enables the internalization of task-specific structure, such as plant-level dynamics in PV systems, and can lead to improved predictive accuracy when sufficient labeled data are available. However, this approach also introduces potential drawbacks, including overfitting in low-data regimes, reduced cross-domain transferability, increased computational costs, and sensitivity to training hyperparameters (Bommasani et al., 2021).

From a regression perspective, fine-tuning corresponds to learning an explicit parametric mapping between inputs and outputs. While this often improves point forecast accuracy, it does not automatically ensure well-calibrated uncertainty estimates. In operational settings where risk-aware decision-making is required, such as reserve allocation or imbalance management, fine-tuning may need to be complemented with probabilistic objectives, ensembling strategies, or quantile-based approaches to avoid masking tail risks behind average performance metrics.

In-context learning offers a fundamentally different adaptation mechanism by enabling implicit learning at inference time. Originally demonstrated at scale in few-shot language modeling (Brown et al., 2020), subsequent theoretical and empirical studies have shown that transformers can implement learning algorithms within the forward pass. In particular, it has been argued that transformers can approximate gradient-based learning dynamics in-context (Von Oswald et al., 2023), and systematic analyses indicate that regression-like function classes can be learned through conditioning alone (Garg et al., 2022). These findings provide a conceptual justification for the use of ICL in numerical regression tasks without parameter updates.

For time-series forecasting, in-context learning typically involves providing the model with a window of past observations, optionally augmented with exogenous variables, and prompting it to generate future values. This setup can be interpreted as next-token prediction where the forecasting task is implicitly defined by the context. Empirical evidence shows that the effectiveness of this approach depends strongly on representation choices and the model's ability to encode numerical patterns. LLMTIME demonstrates that general-purpose LLMs can be repurposed for forecasting by representing time series as text, while highlighting sensitivity to tokenization and calibration (Gruver et al., 2023). In contrast, time-series-pretrained models such as Chronos and Lag-Llama are explicitly designed so that in-context conditioning aligns with the pretraining distribution over tokenized numerical sequences (Ansari et al., 2024; Rasul et al., 2023).

An important practical aspect of in-context learning for regression concerns prompt design. In this setting, prompts function as structured containers that specify the input sequence, optional covariates, output format, and sometimes normalization metadata. Performance may depend on factors such as value scaling, the inclusion of temporal markers, and the number of demonstrations provided. Recent work emphasizes prompt structure as a mechanism for controlling inductive bias in generative time-series forecasting models (Cao et al., 2023; Gruver et al., 2023).

The choice between fine-tuning and in-context learning is therefore closely tied to data availability, deployment constraints, and the nature of the forecasting task. In low-data or rapidly shifting regimes, in-context learning offers immediate adaptability without retraining, albeit with potential sensitivity to context length and representation choices. In more stable settings with abundant labeled data, fine-tuning may yield higher accuracy at the cost of reduced flexibility and increased computational overhead (Jin et al., 2023). For PV power forecasting, the methodological value of LLMs lies in evaluating how these adaptation strategies trade off accuracy, robustness, and interpretability under meteorological uncertainty, a question that is explored empirically in the following chapters.

CHAPTER 3

Data

3.1 Photovoltaic systems and power generation basics

PV power forecasting depends critically on how the physical conversion chain from solar radiation to delivered electricity is conceptualized and represented in data. Even if irradiance is the dominant driver, the mapping from incident radiation to inverter-level AC output is shaped by solar geometry, plant configuration, e.g., fixed tilt versus tracking, and operational conditions that affect conversion efficiency, so treating it as a purely linear relationship can be misleading in practice (Holmgren et al., 2018). For grid-connected applications, inverter-level AC power is the most operationally relevant quantity because it corresponds to the energy injected into the grid and directly reflects inverter behaviour and control, consistent with the measurement design adopted in the reference dataset. The dataset used in this thesis follows the same design principles as in Piantadosi et al. (2024): production is measured at inverter level and normalised by nominal power, and auxiliary irradiance information is available both as an externally provided radiative variable and as a physically modelled signal. Accordingly, this section first summarizes the PV generation process and highlights the main physical drivers that motivate the use of irradiance-related features. It then describes plant locations and technical characteristics, emphasizing heterogeneity across sites and inverters that will matter for modelling choices later in the thesis.

3.1.1 Photovoltaic power generation process

PV systems convert solar radiation into electrical energy through the PV effect. In practical grid-connected applications, the electrical chain can be summarized as follows: incident irradiance on the module surface is converted into direct current (DC) power by the PV array, and DC power is then converted into alternating current

(AC) power by inverters. For forecasting and operational purposes, inverter-level AC power is particularly relevant because it corresponds to the electrical quantity delivered to the grid and is directly affected by inverter performance and control. This perspective is consistent with the dataset design adopted in Piantadosi et al. (2024), where power measurements are available at inverter level and then normalized by nominal power.

A key point for forecasting is that the irradiance-to-power mapping is not purely linear. Even when irradiance is the dominant driver, the effective irradiance at the modules depends on the position of the Sun, the system mounting configuration, i.e. fixed tilt or tracking, and the decomposition of radiation into components, i.e. direct, diffuse, and reflected. In addition, conversion efficiency varies with operating conditions and component characteristics. For this reason, forecasting pipelines typically combine (i) meteorological information describing the evolving atmosphere and incoming radiation, with (ii) physically grounded transformations that account for solar position and plant configuration (Holmgren et al., 2018).

In this thesis, the dataset includes both measured production and irradiance-related variables.
Irradiance can be represented in two complementary ways: as a measured or externally estimated radiative quantity (here denoted `S_RAD`), and as a physically modelled irradiance computed with `pvlib`. The latter uses plant location and configuration (including tracking vs. fixed mounting) together with solar geometry to compute an irradiance estimate that is physically consistent by construction. This approach aligns with the reference framework, which emphasizes modelling irradiance to reduce dependence on local pyranometer measurements that may be affected by faults(Piantadosi et al., 2024; Holmgren et al., 2018).

From a system-integration viewpoint, accurate PV forecasting becomes increasingly important as PV penetration grows, because forecast errors translate into imbalance costs and operational challenges related to reserve procurement and dispatch. Institutional assessments consistently highlight the relevance of improved forecasting for efficient integration of variable renewables, motivating careful documentation of data sources, measurement design, and uncertainty in the forecasting workflow(International Energy Agency, 2023a; IEA PVPS Task 16, 2020).

3.1.2 Plant location and technical characteristics

The dataset analysed in this thesis comprises six grid-connected, ground-mounted PV plants identified by city name. Plants differ in nominal power, number of invert-

ers and modules, and mounting configuration, resulting in heterogeneous production potentials across sites. This heterogeneity is explicitly documented in the reference study, which motivates inverter-level modelling and normalization choices by noting that plants and inverters may differ in ratings and effective capacity. (Piantadosi et al., 2024). A particularly relevant feature is that AC power data are available at inverter level. This enables each plant to be represented as a collection of sub-series. i.e. one per inverter, increasing the number of training instances and allowing the modelling framework to capture both cross-plant differences and intra-plant variability. In the reference dataset, for example, the Corato's plant comprises 27 inverters whose raw AC power profiles exhibit noticeable spread even under similar irradiance and weather conditions; once normalised by nominal power, their behaviour becomes much more homogeneous, supporting the use of a single forecasting model across inverters and plants (Piantadosi et al., 2024).

Nominal power is derived from manufacturer specifications of inverters and connected modules under standard test conditions (STC). Dividing measured AC power by nominal power yields a dimensionless normalised output that facilitates direct comparison across plants and inverters of different size. This normalisation is adopted throughout the thesis and plays a central role in allowing the model to generalise across heterogeneous systems.

Even within the same plant, inverter outputs may differ due to a combination of (i) structural factors, e.g., differences in inverter efficiency or aging, and (ii) local transient effects, e.g., partial shading or soiling patterns that are not perfectly uniform. While such effects are not always directly observable in the covariates, their presence motivates careful standardisation of the target and explicit discussion of uncertainty sources in later sections (Piantadosi et al., 2024).

The impact of normalisation at inverter level is illustrated in Fig. 3.2

Plant	P [kWp]	A [m^2]	Inv.	Mod.	Type	Tilt	Azimuth
S. Cristoforo	4830.80	22 738	28	7 320	Tracking	–	180°
Corato	4440.90	21 017	27	6 660	Tracking	–	180°
Ortona	2256.06	10 792	20	4 176	Fixed	25°	90°
Bisceglie	986.86	4 744	10	1 530	Tracking	–	180°
Modica	997.92	4 775	9	1 848	Fixed	25°	90°
Celso	7799.68	33 768	34	15 912	Tracking	–	180°

Table 3.1: PV plant characteristics (city name as plant label). P denotes the nominal DC power, A the total module area. For tracking plants the type refers to mono-axial tracking structures. Table reproduced from Piantadosi et al. (2024).

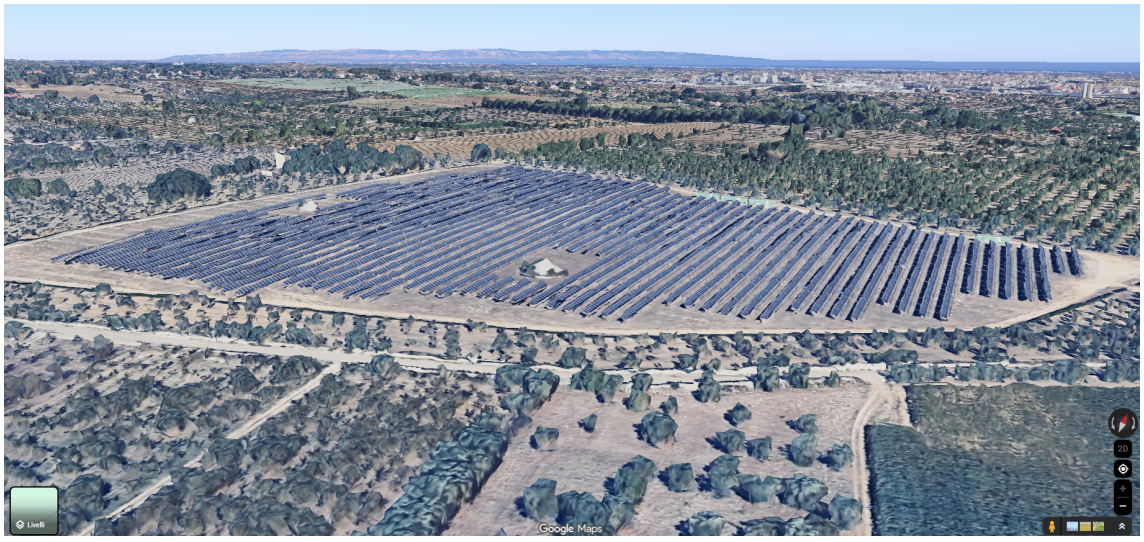


Figure 3.1: Picture of one of the PV plants analyzed in the project

(ENEA, 2026)

3.2 Measurements and monitoring infrastructure

A forecasting pipeline is only as interpretable as the measurement design underlying its target and covariates, especially when multiple plants and components are pooled into a single learning task. In this thesis, production is observed at inverter level and provided in normalized form, which improves comparability across

systems with different nominal capacities while still leaving room for residual heterogeneity due to component-level differences and operating conditions (Piantadosi et al., 2024). For this reason, the present section clarifies how the target variable is defined, how normalization is implemented, and how time indexing is handled in the working dataset, since these choices directly affect what a model can and cannot learn from the data. In addition, monitoring practice introduces a second, often underappreciated layer of complexity: irradiance measurements can be informative but are subject to uncertainty, calibration drift, and practical failure modes, so overreliance on on-site sensors may create avoidable fragility (Lee et al., 2026; Harrou et al., 2025). Documenting these aspects motivates the subsequent emphasis on robustness and on alternative irradiance representations discussed later (Lazzaretti et al., 2020; Piantadosi et al., 2024).

3.2.1 Power measurements

The target variable used in this thesis is inverter-level AC power provided in normalized form as `P_AC_RN`. Normalization is performed by dividing measured AC power by the nominal peak power associated with the inverter or plant, producing a dimensionless quantity that improves comparability across systems with different sizes. The reference framework explicitly adopts this normalization to standardise training data across plants of varying capacity and across inverters within each plant (Piantadosi et al., 2024).

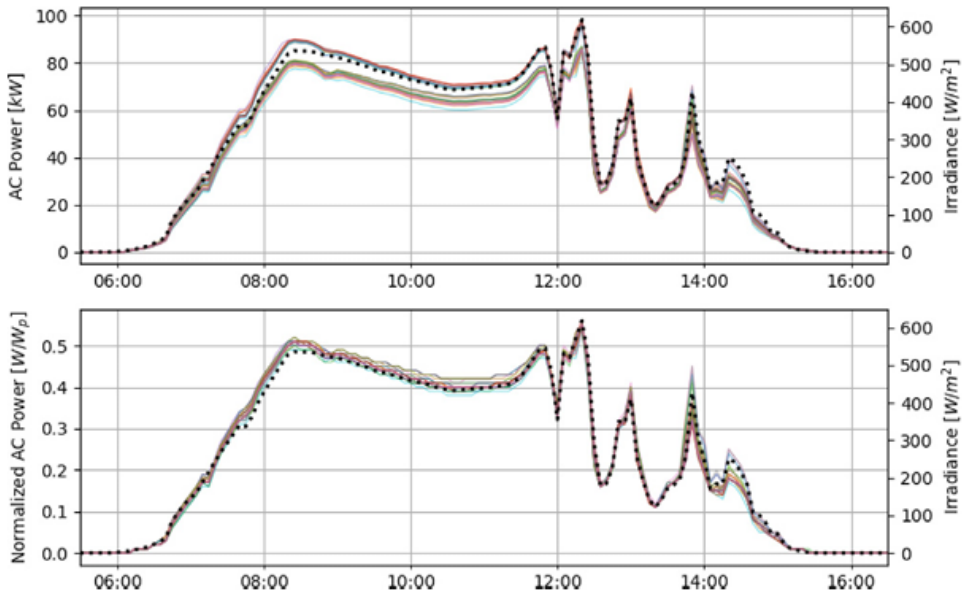


Figure 3.2: Example of inverter-level AC power profiles for the Corato plant. The upper panel shows the raw AC power of 27 inverters during a representative day, highlighting noticeable spread across devices even under similar irradiance conditions. The lower panel shows the same time series after normalisation by nominal power, revealing a much tighter clustering around a common diurnal pattern. Figure adapted from Piantadosi et al. (2024).

Inverter-level granularity is methodologically valuable because it increases the effective sample size and exposes systematic differences between sub-components of the same plant. However, it also implies that a forecasting model must handle residual heterogeneity even after normalization, because different inverters can exhibit distinct behavior due to hardware characteristics, degradation, and operational conditions. Monitoring and fault-detection studies in PV systems highlight that performance deviations can arise from multiple sources, including sensor issues and component-level anomalies, reinforcing the need to document measurement design and robustness choices(Lazzaretti et al., 2020).

Time stamps are recorded in Coordinated Universal Time (UTC). The reference study reports production data originally collected at a high temporal resolution; the working dataset used in this thesis is provided at an aggregated resolution, which is verified directly from the time index and consistently handled in the experimental design.

3.2.2 Irradiance measurements

In PV monitoring practice, irradiance is often measured on-site with pyranometers or reference cells. Such measurements can be highly informative, but they are also subject to uncertainty, calibration drift, and potential faults. Guidance on radiation measurement and instrument practices is provided in institutional standards such as the WMO *Guide to Instruments and Methods of Observation*, which discusses measurement principles and sources of uncertainty for radiation instruments(World Meteorological Organization, 2024).

A key design choice in the dataset and the reference framework is to avoid strict dependence on on-site irradiance sensors. Instead, irradiance information is provided through external sources and physically modelled features (Section 3.3), motivated by robustness considerations when local sensors are unavailable or unreliable. This rationale is consistent with recent research on sensor fault detection and diagnosis in PV monitoring, which treats pyranometer and temperature sensor faults as practically relevant failure modes (Harrou et al., 2025; Piantadosi et al., 2024). In the present work, irradiance- related inputs are therefore derived from external meteorological products and from pvlib-based irradiance models rather than from plant-specific pyranometers.

3.3 Irradiance and meteorological data sources

Beyond defining what is measured on the plant side, PV forecasting requires specifying how exogenous atmospheric information is sourced and transformed into model inputs. In operational settings, relying exclusively on on-site instrumentation can limit portability across locations and can make the workflow vulnerable to missing data or sensor faults; for this reason, recent frameworks increasingly combine external meteorological products with physically grounded feature construction (Piantadosi et al., 2024). This section documents the meteorological and irradiance-related inputs used in the thesis, with a focus on transparency about provenance and assumptions. It first describes the external weather variables retrieved through an open API called Open-meteo, which provides a standardized way to access hourly covariates relevant for PV forecasting (Open-Meteo, 2026a). It then introduces derived irradiance features computed with pvlib, which embed solar-geometry constraints and plant-configuration information into a consistent irradiance estimate (Holmgren et al., 2018). Finally, it discusses the main limitations and uncertainty sources implied by these choices—particularly resolution mismatch and residual plant/in-

verter heterogeneity—since these issues shape how forecasting performance should be interpreted in later chapters (Pierro et al., 2020; Piantadosi et al., 2024).

3.3.1 External meteorological data sources

Meteorological covariates are obtained from external data sources rather than exclusively from on-site instrumentation. This choice supports portability across plants and reduces sensitivity to local sensor outages. The dataset includes a broad set of atmospheric variables typically used in PV forecasting pipelines, such as air temperature, humidity, pressure, precipitation-related fields, wind speed and direction at multiple heights, and cloud cover components at different atmospheric layers. These covariates are aligned in time with the plant-level production data at hourly resolution.

The external meteorological variables used in this thesis are retrieved through the Open-Meteo API, an open-source weather API providing access to weather model outputs via a standardized interface(Open-Meteo, 2026b). This choice was made because of the comparative analysis described in the article *Photovoltaic power forecasting: A Transformer based framework* that showed that, across the considered machine learning models, using Open-Meteo data tends to yield the lowest forecasting errors (Piantadosi et al., 2024).

3.3.2 Derived irradiance features

Beyond raw meteorological variables, the dataset includes irradiance-related features designed to provide physically meaningful information for PV power forecasting. The variable `S_RAD` represents irradiance, while an additional irradiance estimate is computed using the `pvlib` Python library. The `pvlib` framework provides open, benchmarked implementations of PV system models and solar position and irradiance routines, and has been widely adopted in the research community as a reference toolbox for PV modelling workflows(Holmgren et al., 2018).

In the present context, `pvlib` is used to compute a physically consistent estimate of plane-of-array (PoA) irradiance by applying a set of mathematical models for solar geometry and radiative transfer, based on:

- *solar zenith and solar azimuth angles* derived from plant coordinates and time stamps;
- *Global Horizontal Irradiance (GHI); Direct Normal Irradiance (DNI); Diffuse*

Horizontal Irradiance (DHI), computed using a clear-sky and irradiance separation model;

- plant configuration, including tilt and azimuth for fixed-tilt systems, and tracking geometry. i.e. *axis orientation, maximum tilt and backtracking behaviour*, for mono-axial tracking plants.

Conceptually, PoA irradiance is obtained by decomposing the incoming radiation into three contributions: a direct component projected onto the module surface, a sky-diffuse component, and a ground-reflected term driven by surface albedo. In compact form, the model can be written as

$$E_{\mathrm{PoA}} = E_b + E_d + E_g,$$

where E_b is the beam (direct) component, E_d is the diffuse component and E_g is the ground-reflected component. `pvlib` implements this decomposition through explicit analytical transposition formulas and parameterised mathematical models of atmospheric turbidity and air mass, ensuring that the resulting PoA estimate respects basic radiative-transfer and geometric constraints. It is important to note that this PoA irradiance is a mathematical model output rather than a direct measurement.

In the dataset used for this thesis, the `pvlib`-based irradiance estimate is plant-specific but invariant across inverters within the same plant, consistent with its dependence on location and configuration rather than inverter hardware. This structure is useful because it separates (i) a shared plant-level radiative signal from (ii) inverter-level production variability that may reflect component heterogeneity or localised effects. The combination of Open-Meteo covariates and `pvlib`-based PoA irradiance is therefore intended to provide a rich, physically interpretable and fully model-based input space for the forecasting models.
The behaviour of the irradiance model is exemplified in Fig. 3.3

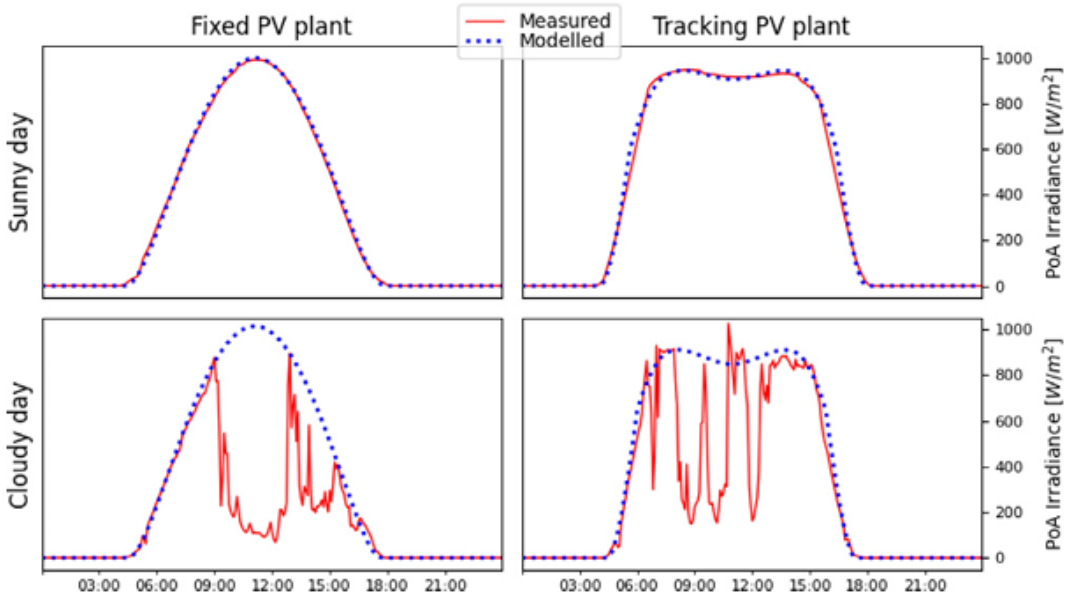


Figure 3.3: Measured (solid line) and modelled (dashed line) plane-of-array irradiance for a fixed-tilt plant (left column) and a tracking plant (right column), on a clear-sky day (top row) and on a cloudy day (bottom row). The irradiance model closely matches observations under clear-sky conditions, while discrepancies become evident on cloudy days because fast-changing atmospheric effects are not explicitly represented. Figure adapted from Piantadosi et al. (2024).

A central methodological motivation for using modelled irradiance is robustness: relying exclusively on on-site pyranometer measurements can expose the forecasting pipeline to missing values or sensor faults. The reference study explicitly uses modelled irradiance as an input feature to avoid dependency on pyranometer data, while still providing a physically grounded representation of solar resource conditions Piantadosi et al. (2024) and the same method applies to the present work.

3.3.3 Data limitations and uncertainties

Despite the advantages of external meteorological covariates and physically modelled irradiance, relevant sources of uncertainty remain. First, external weather data represent model outputs at a finite spatial and temporal resolution, which may not perfectly match the micro-scale atmospheric conditions experienced at the plant site. This mismatch can be particularly relevant under rapidly changing cloud conditions, where local variability may be only partially captured by gridded products. As a consequence, a portion of PV forecasting error may reflect input uncertainty rather

than modelling limitations, motivating explicit discussion of data resolution and robustness choices (Piantadosi et al., 2024; IEA PVPS Task 16, 2020).

Second, measurement uncertainty and sensor faults are a practical concern in operational PV monitoring. Institutional guidance, e.g., WMO documentation on radiation measurement, emphasizes that radiation measurements are affected by instrument characteristics, calibration, and environmental factors. Moreover, recent studies on sensor fault detection in PV systems explicitly consider injected faults in pyranometers and temperature sensors, confirming that sensor anomalies are non-negligible and can degrade downstream analytics if not addressed (Bouyeddou et al., 2022; Harrou et al., 2025).

Finally, inverter-level power series preserve residual heterogeneity even after normalization by nominal power. While normalization makes production signals comparable across plants and inverters of different sizes, it does not eliminate differences due to inverter-specific behavior or localised conditions (e.g., soiling or partial shading). These limitations justify the modelling choices developed in subsequent chapters, where the objective is to learn robust predictive relationships under measurement noise and meteorological uncertainty(Piantadosi et al., 2024).

Table 3.2: Main variables included in the photovoltaic dataset

Variable	**Description**	**Unit**	**Source**	**Notes**
UTC	Timestamp of observation (UTC)	–	Monitoring system	Aggregated temporal resolution
P_AC_RN	Normalized AC power at inverter level	–	Inverter measurements	Normalized by nominal (peak) power W_p
device	Inverter identifier	–	Monitoring system	Distinguishes sub-plants
plant	Plant identifier (city name)	–	Metadata	Constant across inverters within the same plant
S_RAD	Irradiance	W/m^2	External source	Radiative input feature
pvlib	Modelled irradiance	W/m^2	pvlib modelling pipeline	Plant-level feature; invariant across inverters
shortwave_radiation	Global shortwave radiation	W/m^2	Open-Meteo	External meteorological data
direct_normal_irradiance	Direct normal irradiance (DNI)	W/m^2	Open-Meteo	Radiation component

CHAPTER 4

Methodology

4.1 Preprocessing

This section documents the preprocessing pipeline used to (i) construct a supervised learning dataset for day-ahead PV forecasting and (ii) ensure that model evaluation is comparable across plants, inverters, and diurnal regimes. The guiding principle is to apply a single, plant-agnostic set of rules grounded in basic physical consistency of PV production and standard precautions against information leakage in predictive modelling. The target variable is inverter-level normalized AC power, denoted $P_{\mathrm{AC\_\ RN}}$.

4.1.1 Temporal consistency, plant specific availability, fault removal, missing-data treatment, and robustness choices

Preprocessing begins with a lightweight set of temporal and quality-control steps that ensure internal consistency of each inverter time series before supervised windows are constructed. Timestamps are handled in a uniform reference (converted to UTC when timezone information is present and stored in a UTC-based timezone-naive format), and each plant is restricted to its predefined monitoring window in order to exclude out-of-coverage periods that would introduce structured missingness and distort learned relationships.

Observations flagged as faulty by the monitoring system are excluded prior to any window construction. This step is motivated by the well-known sensitivity of PV forecasting models to measurement artefacts and operational anomalies, which can otherwise dominate loss-driven training(Piantadosi et al., 2024; Ahmed et al., 2020).

Remaining missing values are treated within each inverter (device) time series to preserve chronological coherence when constructing sliding windows. The baseline

strategy is a conservative imputation through forward filling and backward filling (restricted within each device), which avoids creating artificial high-frequency variability while maintaining continuity of the input sequences. Here, *forward filling* replaces a missing value at time t with the most recent available observation of the same variable within the *same device* time series (i.e., it propagates the last known value forward in time). Conversely, *backward filling* replaces a missing value at time t with the next available observation within the same device series (i.e., it propagates the next known value backward in time). Applied sequentially, these rules fill interior gaps while also handling missing segments at the beginning of a device-specific record.

Since imputation may attenuate genuine variability if applied aggressively, the pipeline is designed to (i) apply a uniform rule across all devices, see Section 4.2.1, and (ii) keep the imputation mechanism simple and auditable, so that robustness can be assessed in ablations and plant-level diagnostics.

4.1.2 Window construction and day-ahead forecasting task

The forecasting problem is formulated as supervised learning via sliding windows at inverter level. Each sample is defined by a fixed-length historical context of $T = 168$ hourly observations, one week, and a prediction horizon of $H = 24$ hours, day-ahead.

We denote the input window by $\mathbf{x}_t$ and the corresponding target by $\mathbf{y}_t$:

$$\mathbf{x}_t = \big(\mathbf{z}_{t-T+1}, \ldots, \mathbf{z}_t\big), \qquad \mathbf{y}_t = \big(P_{\text{AC\_RN},t+1}, \ldots, P_{\text{AC\_RN},t+H}\big),$$

here, $\mathbf{z}_t$ collects the numerical covariates available at time t (including irradiance-related features and calendar encodings). Windows are constructed separately for each inverter to avoid mixing time series from different physical units within a single input–output pair, preserving the interpretation of a sample as a coherent temporal evolution of one device.

4.1.3 Day/night masking and physical consistency constraints

A central design choice links data construction to evaluation through an explicit day/night indicator. PV production is structurally near-zero during nighttime, so aggregate error metrics can be artificially optimistic if the evaluation is dominated

by trivial nighttime zeros(Ahmed et al., 2020; Piantadosi et al., 2024).

Daylight hours are identified using a horizon-level mask derived from `is_day`. In our implementation, for each forecast origin time t,the model predicts the next 24 hourly values $(t+1, \ldots, t+24)$, and we associate to the same 24 hours a binary mask

$$\mathbf{m}_t = \big(\texttt{is\_day}(t+1), \ldots, \texttt{is\_day}(t+24)\big).$$

The mask allows to compute the metrics for the daytime hours by restricting the evaluation to those horizon points for which $\texttt{is\_day} = 1$. This distinction is important because PV production is structurally near-zero at night; without a dedicated daytime filter, aggregate metrics may be inflated by trivial nighttime predictions and become less comparable across plants or data splits with different shares of night hours, without reflecting the model's ability to capture the diurnal production curve.

This step improves cross-device comparability because it reduces sensitivity of error metrics to plant-specific low-level night artefacts, which are not informative about diurnal dynamics.

4.1.4 Normalization, scaling, and leakage prevention

Numerical covariates are standardized to stabilize optimization and balance heterogeneous feature scales. Let x be a generic feature; the standardized value is

$$\tilde{x} = \frac{x - \mu_{\text{train}}}{\sigma_{\text{train}}}.$$

Crucially, the parameters $(\mu_{\text{train}}, \sigma_{\text{train}})$ are estimated on the training split only and then reused unchanged on validation and test splits, preventing information leakage from future data into preprocessing.

4.2 Validation strategy

4.2.1 Evaluation and metrics: 24 hours, daylight hours, per-plant

Performance is assessed using the same preprocessing techniques across all experimental conditions, including the baseline and the text-conditioned variants. This

strategy ensures that differences in reported accuracy can be attributed to modelling choices rather than to inconsistent scaling, encoding, or diurnal composition of the evaluation set.

Metrics are reported over the full 24-hour horizon and, separately, over daytime hours only, using a horizon-level mask derived from `is_day`, obtained with the method described in 4.1.3. Metrics are also disaggregated by plant to quantify cross-site heterogeneity in forecasting accuracy and to support diagnostic analysis.

We report complementary error and association measures to characterize both absolute accuracy and the ability to track the diurnal profile. Let y_{t+h} denote the observed inverter-level target at horizon step h from forecast origin t, and $\hat{y}_{t+h}$ its prediction, with $h = 1, \ldots, H$ and $H = 24$. Defining residuals $e_{t+h} = y_{t+h} - \hat{y}_{t+h}$, we then stack all forecast–observation pairs across origins and horizons into a single index $i = 1, \ldots, N$. For notational convenience, we write y_i and $\hat{y}_i$ for the observed and predicted values, respectively, and $e_i = y_i - \hat{y}_i$ for the corresponding residuals.

Mean Absolute Error (MAE) measures the average absolute deviation,

$$\mathrm{MAE} = \frac{1}{N} \sum_{i=1}^{N} |e_i|.$$

Median Absolute Error (MdAE) is the median of $|e_i|$ and provides a robust summary less sensitive to occasional spikes.

Mean Squared Error (MSE) measures the average squared deviation,

$$\mathrm{MSE} = \frac{1}{N} \sum_{i=1}^{N} e_i^2.$$

Root Mean Squared Error (RMSE) is the square root of MSE and is expressed on the same scale as the target,

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}.$$

Mean Absolute Percentage Error (MAPE) provides a scale-free measure,

$$\mathrm{MAPE} = \frac{100}{N_+} \sum_{i \in \mathcal{I}_+} \left| \frac{e_i}{y_i} \right|, \qquad \mathcal{I}_+ = \{i : y_i > 0\}, \quad N_+ = |\mathcal{I}_+|,$$

so that percentage errors are computed only when the observed value is strictly positive, avoiding divisions by zero or near-zero values that typically occur at night in PV series.

Symmetric MAPE (sMAPE) reduces asymmetry between over- and under-prediction by normalizing by the sum of magnitudes,

$$\text{sMAPE} = \frac{100}{N} \sum_{i=1}^{N} \frac{2|e_i|}{|y_i| + |\hat{y}_i|}.$$

To summarize linear association, we report the Pearson correlation coefficient r between predictions and observations,

$$r = \frac{\sum_i (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_i (y_i - \bar{y})^2} \sqrt{\sum_i (\hat{y}_i - \bar{\hat{y}})^2}},$$

and the coefficient of determination R^2,

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

Together, these measures capture absolute error in the original scale, relative error in percentage terms, and shape agreement of the predicted diurnal trajectory(Husein et al., 2024; Hewamalage et al., 2023; Sabir et al., 2024; scikit-learn developers, 2025).

4.3 Numerical forecasting baseline

4.3.1 LSTM encoder with plant/device embeddings and cyclical time features

In the baseline setting, the forecasting task introduced in Section 4.1 is tackled with a purely numerical encoder–decoder architecture based on a bidirectional LSTM network, which is a standard choice for multivariate time-series forecasting in energy applications (Hewamalage et al., 2021; Qu et al., 2021; Theocharides et al., 2021). For each inverter (device), the input at time t is a multivariate context window

$$\mathbf{X}_t = \big(\mathbf{x}_{t-T+1}, \ldots, \mathbf{x}_t\big) \in R^{T \times F},$$

constructed from covariate vectors $\mathbf{x}_\tau \in R^F$ at hourly resolution, where T denotes the length of the historical window (here $T = 168$ hours) and F is the total number of input features per time step. The prediction target is the H-dimensional day-ahead

production profile

$$\mathbf{y}_t = \big(y_{t+1}, \ldots, y_{t+H}\big) \in R^H, \qquad H = 24.$$

collecting the inverter-level AC power for the next 24 hours.

Each covariate vector $\mathbf{x}_\tau$ is obtained by augmenting a raw numerical feature vector $\tilde{\mathbf{x}}_\tau$ with: (i) a learned embedding for the plant identifier, (ii) a learned embedding for the device identifier, and (iii) a set of cyclical calendar features that encode the hours, the days and the months. Let $h_\tau \in \{0, \ldots, 23\}$ denote the hour of day at time τ, $w_\tau \in \{0, \ldots, 6\}$ the day of week, and $m_\tau \in \{1, \ldots, 12\}$ the month.

The corresponding cyclical feature vector $\boldsymbol{\phi}_\tau \in R^6$ is defined as

$$\boldsymbol{\phi}_\tau = \Big[\sin\big(2\pi h_\tau/24\big),\ \cos\big(2\pi h_\tau/24\big),\ \sin\big(2\pi w_\tau/7\big),\ \cos\big(2\pi w_\tau/7\big),$$

$$\sin\big(2\pi m_\tau/12\big),\ \cos\big(2\pi m_\tau/12\big)\Big],$$

so that temporally adjacent values, e.g. hour 23 and hour 0, remain close in feature space, avoiding artificial discontinuities at period boundaries and providing a smooth representation of daily and seasonal cycles (Hewamalage et al., 2021; Qu et al., 2021; Theocharides et al., 2021).

Let $e^{(p)}_{p(\tau)}$ and $e^{(d)}_{d(\tau)}$ denote the plant and device embeddings associated with the plant and inverter labels observed at time τ. The final covariate vector fed to the recurrent encoder is obtained by concatenation:

$$\mathbf{x}_\tau = \big[\,\tilde{\mathbf{x}}_\tau;\ e^{(p)}_{p(\tau)};\ e^{(d)}_{d(\tau)};\ \boldsymbol{\phi}_\tau\,\big],$$

The sequence $(\mathbf{x}_{t-T+1}, \ldots, \mathbf{x}_t)$ is processed by a bidirectional LSTM encoder, which summarizes the temporal dependencies in the context window. In the implemented architecture, each $\mathbf{x}_\tau$ is fed to the LSTM. The forward and the backward hidden states of the last recurrent layer are concatenated and passed through a feed-forward forecasting head, which outputs the 24-dimensional prediction vector $\hat{\mathbf{y}}_t$. This design follows recent practice in sequential models for energy-related time series, where recurrent encoders are used to summarize multi-scale temporal dependencies before a multi-step decoding stage (Hewamalage et al., 2021; Qu et al., 2021; Theocharides et al., 2021).

4.3.2 Training setup and optimization

In the baseline model, training is carried out by minimizing a Smooth L1 (Huber-type) loss between the predicted day-ahead profile $\hat{y}_t \in R^H$ and the true target $y_t \in R^H$. For each forecast origin t and horizon index $h = 1, \ldots, H$, the residual is

$$r_{t,h} = \hat{y}_{t,h} - y_{t,h}.$$

The per-component Smooth L1 loss $\ell(r_{t,h})$ is a hybrid between mean squared error and mean absolute error: behaves quadratically for small residuals and linearly for large residuals, combining the accuracy of mean squared error with the robustness of mean absolute error to occasional spikes and outliers in PV production. In the PyTorch implementation `SmoothL1Loss`, this behaviour is controlled by a threshold parameter $\beta > 0$; a common scalar form is

$$\ell(r) = \begin{cases} \dfrac{1}{2\beta}\, r^2, & |r| < \beta, \\ |r| - \dfrac{\beta}{2}, & |r| \geq \beta, \end{cases} \tag{4.1}$$

so that the loss is smooth at the origin and less sensitive to occasional large errors than a purely quadratic objective (V7 Labs, 2022; DigitalOcean Community, 2022). For a mini-batch B of training examples, the objective minimized at each update step is the average loss

$$\mathcal{L}(\theta) = \frac{1}{|B|\, H} \sum_{(t) \in B} \sum_{h=1}^{H} \ell\big(r_{t,h}(\theta)\big), \tag{4.2}$$

where θ denotes all trainable parameters of the forecasting network.

Model parameters are updated by mini-batch gradient-based optimization. At each step, the gradients $\nabla_\theta \mathcal{L}(\theta)$ are computed by backpropagation through the recurrent encoder and the forecasting head. To limit the impact of rare but very large gradients, the ℓ_2-norm of the gradient vector is clipped to a fixed maximum value before applying the update (gradient clipping), which improves numerical stability in deep architectures (Versloot, 2021; DigitalOcean Community, 2022).

The optimizer used in all experiments is AdamW, as provided by the `torch.optim.AdamW` implementation in PyTorch. AdamW is a variant of the Adam algorithm with decoupled weight decay: it maintains adaptive learning rates for each parameter, based on running averages of first and second moments of the gradients, while applying weight decay as a separate step rather than as part of the gradient

(PyTorch Contributors, 2024; Hugging Face, 2020). This design allows one to use standard Adam-style adaptivity together with a simple and interpretable ℓ_2-type regularization on the weights. In the baseline configuration, a single global learning rate and weight-decay coefficient are chosen; training proceeds for one epoch, that is, one complete pass over all mini-batches of the training set.

4.4 Text-conditioned forecasting

4.4.1 Long-context transformer as text encoder

In the text-conditioned variant of the model, the numerical baseline described in Section 4.3 is augmented with a pretrained transformer encoder that maps a structured textual prompt to a dense representation. The goal is not to generate text, but to provide the numerical forecaster with an additional embedding that summarises physically meaningful prior information and a small set of analogous past situations, i.e. few-shot exemplars.

In our implementation the standard BERT-type encoder was the first candidate, since it has become a widely used baseline for transfer learning in natural language processing (Devlin et al., 2019). However, BERT is constrained to input sequences of at most 512 tokens and it turned out being insufficient for the prompt, which includes (i) a block of general instructions, (ii) several 24-hour context–target exemplars, and (iii) a detailed description of the current query window. This motivates the use of a long-context architecture such as Longformer instead of BERT.

Longformer is a transformer variant specifically designed for long documents: it replaces the standard quadratic self-attention pattern with a combination of local sliding-window attention and a small set of global attention tokens, so that the overall attention complexity scales linearly with the sequence length (Beltagy et al., 2020b). Intuitively, each token attends only to a sliding window, while a few global tokens can attend to and be attended by all positions; this yields a sparse attention matrix that still allows information to propagate across the full sequence.This sparse pattern allows thousands of tokens to be processed efficiently on commodity hardware while retaining the ability to model long-range dependencies (Beltagy et al., 2020a). The different attention patterns used in Longformer are illustrated in Figure 4.1, adapted from (Beltagy et al., 2020b).

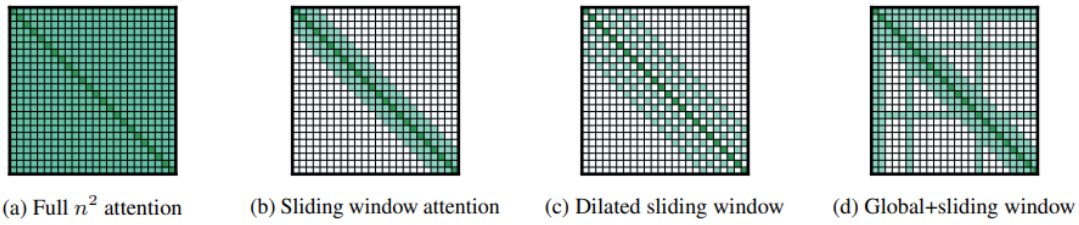


Figure 4.1: Attention patterns in Longformer compared to standard full self-attention: (a) full dense self-attention with quadratic complexity, (b) local sliding-window attention, (c) dilated sliding-window attention and (d) combined global plus sliding-window attention. Adapted from Beltagy et al. (2020b).

In this work, textual prompts are encoded using the publicly released `allenai/longformer-base-4096` checkpoint, accessed through the Hugging Face `transformers` library (Wolf et al., 2020). For each forecast origin time t,the corresponding prompt is tokenised into a sequence of subword tokens and constrained to a maximum length $L_{\max} = 2048$: shorter sequences are padded, while longer ones are truncated from the tail. This yields a token matrix of shape $L_{\max} \times d_{\text{text}}$, where d_{text} is the hidden size of the pretrained encoder. The resulting sequence is fed to the Longformer encoder, which returns a sequence of contextual token embeddings. To obtain a single fixed-dimensional representation of the prompt, the token-level embeddings are aggregated by masked average pooling, i.e. by averaging only over non-padding positions so that artificial padding tokens do not affect the final vector. This yields a text embedding z_{text} that is later combined with the numerical context representation produced by the LSTM encoder, via a gated fusion mechanism, explained in (Section 4.4.3).

Longformer parameters are not frozen: the encoder is initialised from pretrained weights and updated jointly with the rest of the forecasting network during training. In this sense, the text encoder is fine-tuned on PV-related prompts, while still exploiting the general linguistic representations acquired during pre-training.

4.4.2 Prompt template

In the text-conditioned implementation, the model uses a in-context learning setup: for every supervised example, we have built a textual prompt that contains: a short instruction, k past examples from the same device, and a description of the current forecasting problem. The prompt is then encoded by the Longformer model,

and its representation is fused with the numerical context before producing the 24-hour ahead forecast. This design follows recent work on using textual prompts and few-shot examples to convey task definitions and inductive biases to transformer encoders (Brown et al., 2020; Dong et al., 2024).

The *instruction block* provides a compact verbal description of the forecasting task and of basic physical constraints. It specifies that the model should predict the next 24 hourly AC power values, that night-time hours are expected to be near zero, that the diurnal production curve should be smooth with a single peak around local noon, and that higher irradiance and clearer skies are associated with higher daytime output. These rules are fixed across all samples and are intended to encode domain knowledge about PV production in a way that is accessible to a language model.

The *exemplar block* contains up to $k \in \{1, 2, 3\}$ "few-shot" examples drawn from the same inverter as the current query. For each exemplar, the prompt includes: the time stamp of the forecast origin and short sequences or summaries of selected exogenous drivers, such as cloud cover, and temperature when available. In addition, the exemplar block reports the realised 24-hour target profile that followed that context. In this way, each exemplar describes a past situation in terms of both recent dynamics, context, and the subsequent production trajectory, next 24 hours, closely mirroring the structure of the numerical forecasting task.

Exemplars are selected by scanning backwards in time from the current forecast origin and extracting non-overlapping context–horizon blocks of length $T+H$, where $T = 168$ is the context length and $H = 24$ is the forecast horizon. Only blocks corresponding to the same device as the current query are retained. This procedure yields up to k exemplar situations that are temporally separated, device-consistent, and structurally analogous to the current forecasting problem, while avoiding leakage from future information.

The *current query block* encodes the state of the system at the forecast origin. It reports the time stamp, including the hour of day and month, the device and plant labels, and compact numerical summaries of the last 24 hours before the forecast time. In particular, the query includes the recent history of normalised AC power, a simple representation of radiative conditions and day–night status, through the available irradiance-related features and the day/night indicator, and short histories of cloud cover and temperature, when available. These time series are formatted as short numerical lists with limited precision and length, which improves robustness to prompt length constraints without changing the underlying information content.

The full prompt is obtained by concatenating the instruction block, the exemplar

block and the current query block into a single text string, an example is provided in the appendix B. This string is then tokenised and passed to the Longformer encoder described in Section 4.4.1. The resulting text embedding provides a complementary representation of the task and of a small set of analogous past situations, which is later fused with the numerical context representation in the multimodal forecasting architecture.

4.4.3 Gated fusion

In the text-conditioned model, the numerical representation produced by the LSTM encoder and the text representation of the prompt are combined through a gated fusion mechanism. The goal is to let the model learn, for each forecasting instance, how much weight should be assigned to the information coming from the numerical time series and how much to the information coming from the textual prompt.

Concretely, for each forecast origin t, the LSTM encoder produces a latent vector that summarizes the 168-hour numerical context window, while the Longformer encoder produces a latent vector that summarizes the full prompt described in (Section 4.4.2). These two vectors have the same dimensionality and are passed to a small gating network, which outputs a set of coefficients between 0 and 1. Each coefficient can be interpreted as the relative importance given to the text in a specific latent dimension, with the complementary part assigned to the numerical representation.

In the implementation, the gating coefficients are further constrained to lie within a restricted interval, so that the contribution of each modality is never completely suppressed. In practice, this means that the fused representation always contains at least a minimum fraction of numerical information and a minimum fraction of textual information. This design choice reflects the fact that the numerical time series carry detailed information about the recent dynamics of each inverter, while the prompt provides higher-level prior structure and analogous past situations; both sources are therefore kept active in the final representation.

The fused latent vector is then passed to the same feed-forward forecasting head used in the baseline model, described in Section 4.3.1, which produces the 24-hour ahead prediction. Training remains fully end-to-end: the Smooth L1 loss and the AdamW optimizer, described in Section 4.3.2, are applied to the final forecasts, and gradients are propagated through the LSTM encoder, the Longformer encoder, and the fusion gate. As a result, the model not only learns how to encode numerical and textual inputs, but also how to calibrate their relative influence for the PV forecasting task.

CHAPTER 5

Results

This chapter reports the performance of the numerical baseline model and of the text-conditioned variants described in Chapter 4. All models are evaluated on the inverter-level normalised AC power target $P_{\mathrm{AC\_RN}}$, using the error and association metrics introduced in Section 4.2.1.

Case	**MAE**	**MdAE**	**MAPE**	**MSE**	**RMSE**	**Pearson**	R^2	**SMAPE**
Baseline	0.0800	0.0258	43.0825	0.0261	0.1615	0.8831	0.7721	0.1230
$k = 1$	0.0793	0.0260	38.7337	**0.0258**	**0.1606**	**0.8858**	**0.7744**	0.1218
$k = 2$	**0.0780**	**0.0245**	44.1719	0.0263	0.1622	0.8849	0.7698	**0.1188**
$k = 3$	0.0799	0.0248	**35.5055**	0.0273	0.1653	0.8812	0.7612	0.1212

Table 5.1: Final results for all day. Best results are highlighted.

Case	**MAE**	**MdAE**	**MAPE**	**MSE**	**RMSE**	**Pearson**	R^2	**SMAPE**
Baseline	0.1101	0.0588	3.0551	0.0343	0.1853	0.7973	0.6051	0.1098
$k = 1$	0.1086	0.0557	**2.9590**	**0.0340**	**0.1844**	0.8017	**0.6088**	0.1082
$k = 2$	**0.1060**	**0.0525**	3.1563	0.0342	0.1848	**0.8030**	0.6072	**0.1047**
$k = 3$	0.1090	0.0540	3.2851	0.0355	0.1885	0.7937	0.5916	0.1072

Table 5.2: Final results for daylight hours. Best results are highlighted.

For each metric, results are reported both over the full 24-hour forecast horizon and, separately, restricted to daylight hours only. The latter case is obtained by applying the `is_day` mask defined in Section 4.1.3, so that nighttime hours with structurally near-zero production are excluded from the computation of daytime metrics.

Four configurations are compared: the numerical *baseline*, which uses only the LSTM-based encoder with plant/device embeddings and cyclical time features; the text-conditioned models with $k = 1, 2, 3$, which differ in the number of past context–target exemplars from the same inverter included in the textual prompt provided to the Longformer encoder.

Tables 5.1 and 5.2 summarise the final test performance for the four cases. For each metric, the best value across the configurations is highlighted in bold.

Over the full 24-hour horizon (Table 5.1), all text-conditioned configurations achieve performance comparable to the numerical baseline, with small but systematic improvements for $k = 1$ and $k = 2$. In particular, the $k = 1$ model attains the lowest squared-error metrics and the highest linear association, while $k = 2$ achieves the lowest absolute and symmetric percentage errors; no further gains are observed for $k = 3$.

When the evaluation is restricted to daylight hours only (Table 5.2), the same qualitative pattern is observed. The text-conditioned variants remain at least competitive with the baseline, and the configurations with $k = 1$ and $k = 2$ provide modest reductions in absolute errors and slight increases in correlation and R^2, whereas the $k = 3$ model does not yield additional improvements.

To complement the aggregate analysis, Table 5.3 reports daylight hours performance of the $k = 2$ configuration for each plant.

Plant	$\mathrm{MAE}_{\mathrm{day}}$	R^2_{day}	N_{day}
Modica	0.042	0.9528	38 475
Garpuglia	0.049	0.9277	25 740
Ortona	0.074	0.7438	85 560
Bisceglie	0.081	0.8382	42 915
Molino Casillo	0.114	0.6292	115 628
Celso	0.118	0.5026	144 666
San Cristoforo	0.148	0.2821	121 899

Table 5.3: Daytime performance by plant for the $k = 2$ text-conditioned configuration. Metrics are computed on the normalized AC power target over daylight hours only. N_{day} denotes the number of hourly daylight observations in the test set for each plant.

The daytime MAE ranges from approximately 0.04 to 0.15, i.e. a spread of a factor of about three between the best and worst cases, with a broadly consistent pattern across R^2_{day}.

CHAPTER 6

Discussion

This chapter interprets the results reported in Chapter 5. The focus is on the relative behaviour of the numerical LSTM baseline and of the text-conditioned variants, rather than on the absolute value of individual metrics.

6.1 Overall effect of text conditioning

At aggregate level, the introduction of a text encoder conditioned on a structured prompt yields small but systematic gains over the purely numerical baseline. Across both evaluation regimes, full 24-hour horizon and daylight hours only, the three text-conditioned configurations are at least competitive with the baseline, and in several metrics they achieve slightly better scores.

For the full 24-hour horizon, the configuration with $k = 1$ exemplar attains the lowest squared-error metrics, MSE and RMSE, and the highest Pearson correlation and R^2, while the $k = 2$ model provides the best MAE and SMAPE. When the evaluation is restricted to daylight hours, the same pattern persists: $k = 1$ and $k = 2$ jointly dominate the baseline on most error and association measures, with differences that are numerically modest but consistent. This split between metrics is interpretable. Squared-error and correlation measures are more sensitive to occasional large deviations and to the overall shape of the daily profile, which appear to be slightly better captured when $k = 1$. In contrast, MAE and SMAPE emphasise typical absolute and relative errors, for which the configuration with $k = 2$ achieves a marginally more balanced performance over the distribution of conditions. Since MAPE is strongly affected by hours with very low or zero production, we primarily rely on MAE and SMAPE for ranking the few-shot configurations, which points to $k = 2$ as the most robust choice among the tested text-conditioned models. In particular, the relatively high MAPE observed for $k = 2$ and for the baseline (Figure 6.1) is largely driven by small but non-zero predictions during nighttime hours in which the target is exactly zero: in this regime even a tiny absolute error is translated into a very large percentage error.

The $k = 3$ variant does not yield further systematic improvements and in some metrics slightly underperforms the simpler few-shot setups, suggesting diminishing returns when too many exemplars are injected into the prompt.
A plausible interpretation is that, as k increases, the additional examples become progressively more heterogeneous, so that the textual channel starts to convey noisy or weakly relevant situations rather than sharpening the conditioning on the current forecast, which can offset potential gains from providing more context.

To complement the aggregate metrics, Figure 6.1 shows the hourly trajectories predicted by all models for a representative sunny day. In this example, the three text-conditioned configurations follow the observed diurnal profile more closely than the numerical baseline: they reproduce the morning ramp-up and evening ramp-down with smaller timing errors, while the baseline tends to anticipate the increase at dawn, underpredict around the late afternoon. Moreover, both the baseline and the ground-truth series display a non-negligible onset of production roughly one time step earlier than the few-shot models, and return to near-zero output one time step earlier in the evening. This systematic shift suggests that part of the disagreement may be related to temporal alignment issues, for instance, the hourly resolution of the series or a residual time-zone mismatch between the numerical inputs and the conventions implicitly learned by the text encoder. In principle, this aspect could be further investigated by increasing the temporal resolution of the data and explicitly checking the consistency of time stamps and time zones across all input sources.
These discrepancies at the edges of the diurnal cycle are consistent with the higher absolute and relative errors observed for the baseline in the aggregate statistics, and illustrate how few-shot text conditioning mainly refines the shape of the daily curve rather than introducing large changes in overall production levels.

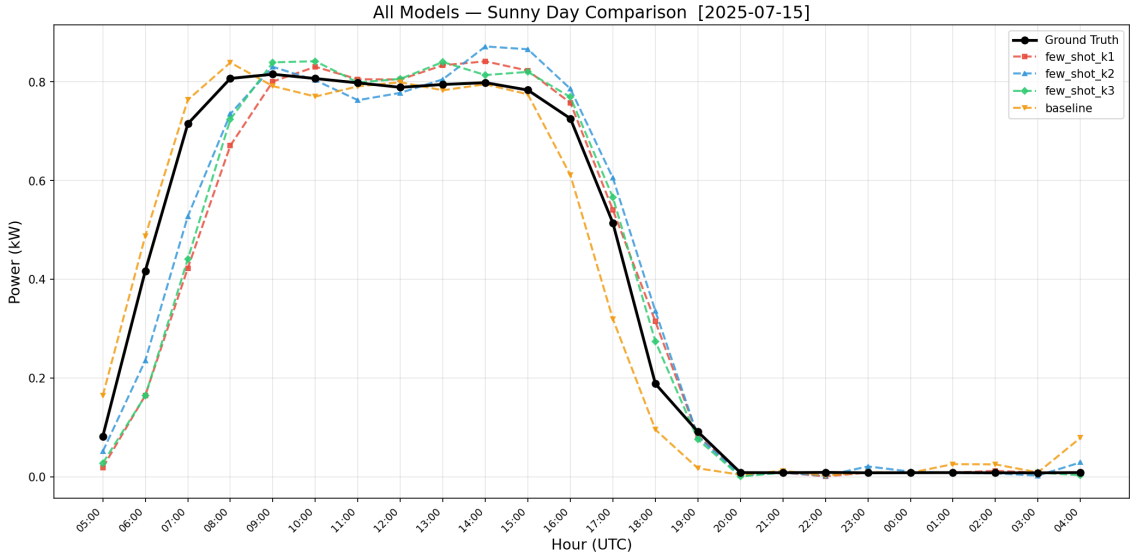


Figure 6.1: Example of hourly profiles for a representative sunny day, comparing the ground truth normalised AC power with the numerical baseline and the three text-conditioned configurations.

Taken together, these results directly address the research question of whether few-shot text conditioning can improve upon a purely numerical LSTM baseline. Across both the full and daytime horizons, all text-conditioned configurations perform at least as well as the baseline and often slightly better, with no evidence of degradation in any of the core metrics. The gains are not large in magnitude, which is consistent with a setting in which the baseline already exploits most of the predictive content of the meteorological and irradiance inputs, and with the limited size of the available dataset. The absence of monotonic improvement as k increases also indicates that, in this setup, the text encoder is not primarily limited by a lack of exemplars, but rather by the amount of information that can be usefully integrated with the numerical representation: beyond one or two well-chosen past examples, additional context appears to offer diminishing returns. Nonetheless, the comparison indicates that augmenting numerical features with a small amount of structured textual context can provide a modest but systematic benefit over a well-tuned numerical-only model.

From a modelling standpoint, the fact that different metrics attain their best values at different k highlights that the few-shot parameter is a genuinely sensitive hyperparameter. In practice, k should be tuned according to the aspects of performance that are most relevant for the application, e.g. absolute versus relative errors, and, potentially, to the characteristics of each individual plant. In an operational set-

ting, k could be selected on a validation set separately for each site, optimising it with respect to the metrics and error patterns that are most critical for that specific installation.

6.2 Daytime and full-day performance

One motivation for reporting metrics separately over the full 24-hour horizon and over daylight hours is to verify that the observed gains are not driven primarily by trivial nighttime behaviour. If the text-conditioned models were improving the forecast mainly by enforcing near-zero output during the night, we would expect a clear advantage on 24-hour metrics but a much weaker effect when restricting the evaluation to hours with non-negligible production.

The results do not support this scenario. The relative ordering of the four configurations is broadly stable across the two evaluation regimes: the few-shot models that improve RMSE, correlation and absolute errors over the full horizon also provide modest gains when the metrics are computed only on daylight hours. This suggests that the additional textual context contributes to refining the shape of the daytime production curve rather than just correcting residual artifacts in periods where the physical signal is structurally close to zero. The example in Figure 6.1 illustrates this behaviour: while some configurations, notably the baseline and $k = 2$, maintain small but non-zero values during the night, the main differences across models arise around sunrise and sunset, where the timing and steepness of the ramp are most critical for operational use.

This interpretation is consistent with the intended role of the prompt, which combines simple physical priors, such as the single-peaked diurnal profile and near-zero generation at night, with few-shot examples of full daily profiles. Together with the day/night mask used at evaluation time, this setup helps separate any improvements in trivial nighttime conditions from the impact of text conditioning on operationally relevant daylight hours.

6.3 Plant-level performance

The two plants with the lowest daytime errors are *Modica* and *Garpuglia*, with $\mathrm{MAE}_{\mathrm{day}}$ of 0.042 and 0.049 and R^2_{day} above 0.92. For these sites, the model is able to track both the level and the shape of the daytime production curve with relatively small typical deviations. In the intermediate range, *Ortona* and *Bisceglie* show moderately larger MAE, around 0.07–0.08, with R^2_{day} between 0.74 and 0.84.

Ortona achieves a slightly lower MAE than *Bisceglie* but worse R^2_{day}, suggesting that its errors are more concentrated in a few periods with larger discrepancies, while *Bisceglie* exhibits a more homogeneous, though slightly higher, typical error.

Figure 6.2 visualises these differences by plotting daytime MAE per plant, ordered from lowest to highest error.
The chart highlights the clear separation between the best-performing sites, i.e. Modica and Garpuglia, the intermediate group, i.e. Ortona and Bisceglie, and the three plants with substantially higher errors.

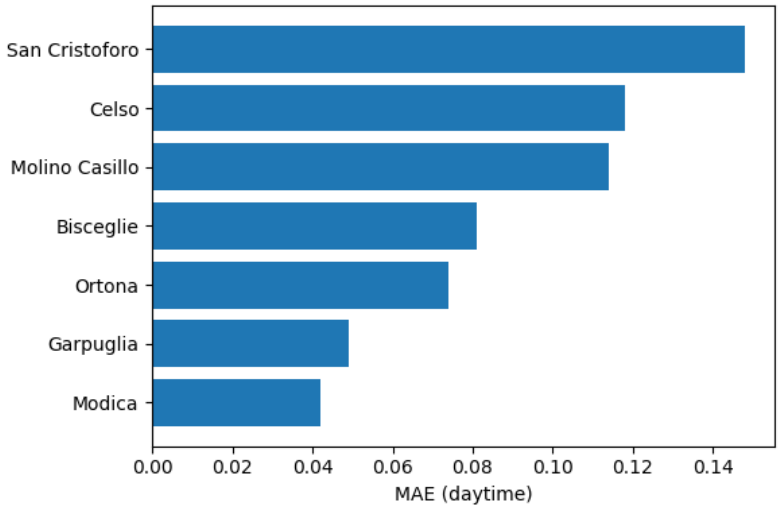


Figure 6.2: Daytime MAE by plant for the $k = 2$ configuration, sorted from lowest to highest error.

The largest daytime errors are observed for *Molino Casillo*, *Celso* and *San Cristoforo*, with MAE_{day} between 0.11 and 0.15 and R^2_{day} in the 0.28–0.63 range. For these plants, residual variability remains relatively high, indicating that the current model struggles to fully capture plant-specific behaviour or local conditions. This could reflect a combination of unmodelled operational factors, e.g. curtailment, maintenance, site-specific shading, and remaining imperfections in the meteorological inputs, but a detailed diagnostic analysis is beyond the scope of this thesis.

The sample sizes N_{day} also vary substantially across plants, from about 25,000 to 145,000 hours observations, yet there is no clear monotonic relationship between the amount of available data and the achieved accuracy. For instance, *Modica* and *Garpuglia* achieve the lowest errors with intermediate sample sizes, while *Celso* and *San Cristoforo* accumulate the largest number of daytime observations but still show relatively high MAE. This suggests that, in this setting, plant-level differences in

data quality and physical complexity play a more important role than sample size.

Overall, plant-level results confirm the aggregate picture: the $k = 2$ configuration delivers reasonably consistent performance across sites, with better tracking of the diurnal profile in some plants than in others. Using $\mathrm{MAE}_{\mathrm{day}}$ as the primary ranking criterion and R^2_{day} as a complementary measure of explained variability provides a compact yet informative characterisation of these differences.

Conclusion

The methodological novelty of this thesis, which focuses on Solar Power Forecasting, lies in analyzing whether in-context learning could improve the efficacy of the LSTM forecaster, by experimenting different levels of few shot prompting. Within this framework, the empirical analysis compared the hybrid model and the purely numerical LSTM under a common data preparation, training, and validation protocol.

The experimental tests show that the proposed text-conditioned architecture is able to match, and in several cases modestly improve upon, the performance of the LSTM baseline when evaluated both over the full 24-hour horizon and, separately, over daylight hours only. The few-shot component embedded in the prompt has a measurable effect: increasing the number of past examples generally strengthens the conditioning signal at first, but the gains quickly saturate and may even deteriorate as additional examples mostly inject redundant variability and noise. The analysis also highlights heterogeneity across plants and inverters, suggesting that the benefit of text conditioning is context-dependent and tends to be larger where the diurnal pattern is more regular and the numerical inputs are noisier or less informative.

It is important to emphasise that the primary objective of this work was not to obtain the single best-performing forecast in absolute terms, but rather to perform a controlled comparison between alternative modelling strategies drawn from the recent literature. From this perspective, the contribution is methodological: we designed a jointed LSTM-Transformer architecture where natural language inputs and forecasting numerical data are combined with a gated fusion mechanism.

As highlighted in the problem statement, the growing penetration of photovoltaic generation increases the economic relevance of accurate short-term forecasts. Improvements in forecasting performance, such as those explored in this work, may therefore contribute to reducing forecast uncertainty, imbalance costs, and balancing needs in electricity systems. Similar considerations apply, however, to many technical activities which receive the help of AI tools.

Several extensions of this work are left for future research. In this thesis the focus was on day-ahead point forecasts over a single 24-hour horizon; applying the same framework to longer or multi-day horizons, or to settings where different lead times are modelled jointly, would help to clarify how far the benefits of text conditioning

can scale in time. On the modelling side, the comparison was limited to one LSTM baseline and a single family of transformer architectures. A broader benchmark against state-of-the-art numerical forecasters for time series, would provide a sharper assessment of the incremental value brought by the textual channel. Finally, the prompts used here encoded only simple physical priors and a small number of few-shot examples. Future work could experiment with richer forms of domain knowledge and quantify to what extent these additional signals translate into measurable gains in forecast quality.

Appendix A

Dataset variable dictionary

Table A.1: Complete list of dataset variables grouped by category.

Variable	Description	Unit	Source
Identifiers and target variables			
UTC	Timestamp of observation (UTC).	–	Monitoring system
P_AC_RN	Normalized inverter-level AC power (normalized by nominal peak power W_p).	–	Monitoring system
device	Inverter identifier (sub-plant ID).	–	Monitoring system
plant	Plant identifier (city name).	–	Metadata
Irradiance and radiative features			
pvlib	Modelled irradiance computed via `pvlib` based on solar geometry and plant configuration (plant-level; invariant across inverters).	W/m^2	pvlib pipeline
SRAD	Irradiance feature provided in the dataset.	W/m^2	External/dataset
shortwave_rad	Global shortwave radiation.	W/m^2	Open-Meteo
shortw_rad_inst	Instantaneous shortwave radiation.	W/m^2	Open-Meteo
dir_norm_irr	Direct normal irradiance (DNI).	W/m^2	Open-Meteo
dir_norm_irr_inst	Instantaneous DNI.	W/m^2	Open-Meteo
diffuse_rad	Diffuse radiation component.	W/m^2	Open-Meteo
diffuse_rad_inst	Instantaneous diffuse radiation.	W/m^2	Open-Meteo
direct_rad	Direct radiation component.	W/m^2	Open-Meteo

Continued on next page.

Table A.1: Complete list of dataset variables grouped by category (continued).

Variable	Description	Unit	Source
direct_rad_inst	Instantaneous direct radiation.	W/m^2	Open-Meteo
terr_rad	Terrestrial radiation.	W/m^2	Open-Meteo
terr_rad_inst	Instantaneous terrestrial radiation.	W/m^2	Open-Meteo
Atmospheric state variables			
temperature_2m	Air temperature at 2 m above ground.	°C	Open-Meteo
apparent_temp	Apparent temperature.	°C	Open-Meteo
dewpoint_2m	Dew point temperature at 2 m.	°C	Open-Meteo
relativehumid_2m	Relative humidity at 2 m.	%	Open-Meteo
vapor_p_deficit	Vapour pressure deficit (provider definition).	–	Open-Meteo
pressure_msl	Mean sea level pressure.	hPa	Open-Meteo
surface_pressure	Surface pressure.	hPa	Open-Meteo
Cloud cover variables			
cloudcover	Total cloud cover.	%	Open-Meteo
cloudcover_low	Low-level cloud cover.	%	Open-Meteo
cloudcover_mid	Mid-level cloud cover.	%	Open-Meteo
cloudcover_high	High-level cloud cover.	%	Open-Meteo
Wind variables			
windspeed_10m	Wind speed at 10 m.	m/s	Open-Meteo
winddir_10m	Wind direction at 10 m.	deg	Open-Meteo
windgusts_10m	Wind gust speed at 10 m.	m/s	Open-Meteo
windspeed_100m	Wind speed at 100 m.	m/s	Open-Meteo
winddir_100m	Wind direction at 100 m.	deg	Open-Meteo
Precipitation and surface variables			
precipitation	Total precipitation.	mm	Open-Meteo
rain	Rainfall.	mm	Open-Meteo
snowfall	Snowfall (provider definition).	–	Open-Meteo
et0_fao_evapotr	Reference evapotranspiration (FAO-56 ET0).	mm	Open-Meteo
is_day	Daylight indicator.	0/1	Open-Meteo

Continued on next page.

Table A.1: Complete list of dataset variables grouped by category (continued).

Variable	**Description**	**Unit**	**Source**
weathercode	Weather condition code.	–	Open-Meteo
Soil-related variables			
soil_m_0_7cm	Soil moisture, 0–7 cm layer (provider definition).	–	Open-Meteo
soil_m_7_28cm	Soil moisture, 7–28 cm layer (provider definition).	–	Open-Meteo
soil_m_28_100cm	Soil moisture, 28–100 cm layer (provider definition).	–	Open-Meteo
soil_m_100_255cm	Soil moisture, 100–255 cm layer (provider definition).	–	Open-Meteo
soil_t_0_to_7cm	Soil temperature, 0–7 cm layer.	°C	Open-Meteo
soil_t_7_28cm	Soil temperature, 7–28 cm layer.	°C	Open-Meteo
soil_t_28_100cm	Soil temperature, 28–100 cm layer.	°C	Open-Meteo
soil_t_100_255cm	Soil temperature, 100–255 cm layer.	°C	Open-Meteo
Quality flags			
fault	Binary fault indicator for anomalous observations.	0/1	Monitoring system

Appendix B

Prompt Used for In-Context Forecasting

Listing B.1 reports the few shot prompting, in this example $k = 1$, used in the day-ahead photovoltaic forecasting experiments.

```
prompt = """
You are a solar PV day-ahead forecaster.
Task: predict the next 24 hourly AC power values.

Rules:
- Night hours should be near 0.
- Forecast should follow a smooth diurnal curve (single peak
    around solar noon).
- More irradiance -> higher power; more cloudcover -> lower
   peak / flatter curve.

## FEW-SHOT EXAMPLES (past situations)
time_utc=2025-06-18 07:00
ctx_power_last24=[0.5, 0.2, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
   0.0, 0.0, 0.0, 0.0, 0.2, 0.5, 0.7, 0.8, 0.8, 0.8, 0.8,
   0.8, 0.8, 0.8, 0.8, 0.6]
ctx_irr_last24=[1.2, -0.0, -0.6, -0.6, -0.7, -0.7, -0.7,
   -0.7, -0.7, -0.7, -0.7, -0.7, -0.1, 1.0, 1.9, 2.2, 2.3,
   2.4, 2.3, 2.3, 2.3, 2.3, 2.2, 1.7]
ctx_cloud_last24=[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
   -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -0]
ctx_temp_last24=[3.0, 2.8, 2.6, 2.5, 2.5, 2.4, 2.3, 2.1,
   2.1, 1.9, 1.9, 1.9, 1.9, 2.3, 2.7, 2.9, 3.1, 3.2, 3.3,
   3.4, 3.4, 3.4, 3.4, 3.4]

## CURRENT QUERY (forecast origin)
```

```
time_utc=2025-06-26 16:00 | hour=16 | month=6
device=Id179209 | plant=Bisceglie
is_day=[1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
   0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
   1.0, 1.0, 1.0]
ctx_power_last24=[0.5, 0.2, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
   0.0, 0.0, 0.0, 0.0, 0.2, 0.5, 0.7, 0.8, 0.8, 0.8, 0.8,
   0.8, 0.8, 0.8, 0.8, 0.6]
ctx_irr_last24=[1.2, -0.0, -0.6, -0.6, -0.7, -0.7, -0.7,
   -0.7, -0.7, -0.7, -0.7, -0.7, -0.1, 1.0, 1.9, 2.2, 2.3,
   2.4, 2.3, 2.3, 2.3, 2.3, 2.2, 1.7]
ctx_cloud_last24=[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
   -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -0]
ctx_temp_last24=[3.0, 2.8, 2.6, 2.5, 2.5, 2.4, 2.3, 2.1,
   2.1, 1.9, 1.9, 1.9, 1.9, 2.3, 2.7, 2.9, 3.1, 3.2, 3.3,
   3.4, 3.4, 3.4, 3.4, 3.4]
"""
```

Listing B.1: Python example

Bibliography

Ahmed, R., Sreeram, V., and Mishra, Y. (2020). A review and evaluation of the state-of-the-art in pv solar power forecasting. *Renewable and Sustainable Energy Reviews*, 124:109792.

Ansari, A. F. et al. (2024). Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

Beltagy, I., Peters, M. E., and Cohan, A. (2020a). allenai/longformer-base-4096. `https://huggingface.co/allenai/longformer-base-4096`. Hugging Face model card, accessed 13 Feb 2026.

Beltagy, I., Peters, M. E., and Cohan, A. (2020b). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

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

Bommasani, R., Hudson, D. A., Adeli, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Bouyeddou, B., Harrou, F., Taghezouit, B., Sun, Y., and Hadj Arab, A. (2022). Improved semi-supervised data-mining-based schemes for fault detection in a grid-connected photovoltaic system. *Energies*, 15(21):7978.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Cao, D., Jia, F., Arik, S. O., Pfister, T., Zheng, Y., Ye, W., and Liu, Y. (2023). Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*.

Das, A., Kong, W., Sen, R., and Zhou, Y. (2024). A decoder-only foundation model for time-series forecasting.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

DigitalOcean Community (2022). PyTorch loss functions. `https://www.digitalocean.com/community/tutorials/pytorch-loss-functions`. Accessed: 13 February 2026.

Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Chang, B., et al. (2024). A survey on in-context learning. pages 1107–1128.

ENEA (2026). ENEA – Agenzia nazionale per le nuove tecnologie, l'energia e lo sviluppo economico sostenibile. Accessed: 2026-03-03.

Garg, S., Tsipras, D., Liang, P. S., and Valiant, G. (2022). What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35:30583–30598.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press, Cambridge, MA. Online version available at `https://www.deeplearningbook.org/`.

Gruver, N., Finzi, M., Qiu, S., and Wilson, A. G. (2023). Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36.

Gui, Y., Nainar, K., Bendtsen, J. D., Diewald, N., Iov, F., Yang, Y., Blaabjerg, F., Xue, Y., Liu, J., Hong, T., and Stoustrup, J. (2024). Voltage support with pv inverters in low-voltage distribution networks: An overview. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 12(2):1503–1522.

Gurioli, A. (2024). Lecture slides on attention mechanisms. Lecture slides, University of Bologna. Available at `https://virtuale.unibo.it/mod/resource/view.php?id=2197404` (accessed 1 March 2026).

Harrou, F., Kini, K. R., Madakyaru, M., and Sun, Y. (2025). Sensor fault detection and diagnosis in photovoltaic systems using hellinger distance and individual conditional expectation analysis. *Solar Energy*, 298:113633.

Hewamalage, H., Ackermann, K., and Bergmeir, C. (2023). Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, 37(2):788–832.

Hewamalage, H., Bergmeir, C., and Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hoffmann, J., Borgeaud, S., Mensch, A., et al. (2022). Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Holmgren, W. F., Hansen, C. W., and Mikofski, M. A. (2018). pvlib python: a python package for modeling solar energy systems. *Journal of Open Source Software*, 3(29):884.

Hong, T., Wang, P., and White, L. (2015). Weather station selection for electric load forecasting. *International Journal of Forecasting*, 31(2):286–295.

Hugging Face (2020). Adamw optimizer documentation. `https://huggingface.co/transformers/main_classes/optimizer_schedules.html#adamw`. Accessed: 13 February 2026.

Husein, M., Gago, E., Hasan, B., and Pegalajar, M. (2024). Towards energy efficiency: A comprehensive review of deep learning-based photovoltaic power forecasting strategies. *Heliyon*, 10(13).

IEA PVPS Task 16 (2020). Regional solar power forecasting 2020. Technical report, International Energy Agency Photovoltaic Power Systems Programme (IEA PVPS). Accessed: 2026-01-31.

Iheanetu, K. J. (2022). Solar photovoltaic power forecasting: A review. *Sustainability*, 14(24):17005.

International Energy Agency (2023a). Renewables 2023: Analysis and forecast to 2028. Technical report, International Energy Agency, Paris. Accessed: 2026-01-31.

International Energy Agency (2023b). Trends in photovoltaic applications 2023. Accessed: 2026-01-24.

International Renewable Energy Agency (IRENA) (2019). Future of solar photovoltaic: Deployment, investment, technology, grid integration and socio-economic aspects. Technical report, International Renewable Energy Agency, Abu Dhabi. Accessed 2026-02-28.

Jin, M. et al. (2023). Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.

Jordan, D. C., Kurtz, S. R., VanSant, K., and Newmiller, J. (2016). Compendium of photovoltaic degradation rates. *Progress in Photovoltaics: Research and Applications*, 24(7):978–989.

Kaplan, J., McCandlish, S., Henighan, T., et al. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Kong, W., Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., and Zhang, Y. (2019). Short-term residential load forecasting based on lstm recurrent neural network. *IEEE Transactions on Smart Grid*, 10(1):841–851.

Lazzaretti, A. E. et al. (2020). A monitoring system for online fault detection and classification in photovoltaic plants. *Sensors*, 20(17):4688.

Lee, H., Kim, D., Cho, H., Song, G., and Yoon, J. (2026). Evaluation of data imputation models for building-integrated photovoltaic systems with practical performance and reproducibility. *Solar Energy*, 308:114428.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., et al. (2022). Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857.

Lim, B., Arık, S. Ö., Loeff, N., and Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764.

Lin, T., Wang, Y., Liu, X., and Qiu, X. (2022). A survey of transformers. *AI open*, 3:111–132.

Nowotarski, J. and Weron, R. (2018). Recent advances in electricity price forecasting: A review of probabilistic forecasting. *Renewable and Sustainable Energy Reviews*, 81:1548–1568.

Open-Meteo (2026a). Weather forecast api documentation. Online documentation.

Open-Meteo (2026b). Weather forecast api documentation. `https://open-meteo.com/en/docs`. Accessed: 2026-01-31.

Piantadosi, G., Dutto, S., Galli, A., De Vito, S., Sansone, C., and Di Francia, G. (2024). Photovoltaic power forecasting: A Transformer based framework. *Energy and AI*, 18:100444.

Pierro, M., Cornaro, C., Moser, D., Betti, A., Morschella, M., Collino, E., Dario, R., Van der Meer, D., Widen, J., Visser, L., et al. (2020). Regional solar power forecasting 2020. Technical report.

PyTorch Contributors (2024). torch.optim.adamw. `https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html`. Accessed: 13 February 2026.

Qu, J., Qian, Z., and Pei, Y. (2021). Day-ahead hourly photovoltaic power forecasting using attention-based cnn-lstm neural network embedded with multiple relevant and target variables prediction pattern. *Energy*, 232:120996.

Rasul, K. et al. (2023). Lag-llama: Towards foundation models for time series forecasting. *arXiv preprint arXiv:2310.08278*.

Sabir, D., Hafeez, K., Batool, S., Akbar, G., Khan, L., Hafeez, G., and Ullah, Z. (2024). Prediction of solar pv power using deep learning with correlation-based signal synthesis. *IEEE Access*, 12:40736–40751.

scikit-learn developers (2025). `sklearn.metrics.r2_score` documentation. scikit-learn documentation. Accessed 2026-02-12.

Solar Resource Solutions, L. (2015). Best practices handbook for the collection and use of solar resource data for solar energy applications. Technical report.

Theocharides, S., Theristis, M., Makrides, G., Kynigos, M., Spanias, C., and Georghiou, G. E. (2021). Comparative analysis of machine learning models for day-ahead photovoltaic power production forecasting. *Energies*, 14(4):1081.

V7 Labs (2022). The essential guide to PyTorch loss functions. `https://www.v7labs.com/blog/pytorch-loss-functions`. Accessed: 13 February 2026.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.

Versloot, C. (2021). How to use PyTorch loss functions. `https://github.com/christianversloot/machine-learning-articles/blob/main/how-to-use-pytorch-loss-functions.md`. Accessed: 13 February 2026.

Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. (2023). Transformers learn in-context by gradient descent. pages 35151–35174.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. (2022). Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Wolff, B., Lorenz, E., and Kramer, O. (2016). Statistical learning for short-term photovoltaic power predictions. pages 31–45.

World Meteorological Organization (2024). *Guide to Instruments and Methods of Observation (WMO-No. 8): 2024 edition*. World Meteorological Organization.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115.