# Towards an Agentic and Explainable Runtime Architecture for PV Forecast Delivery

Leila Sakli
*Aix-Marseille Université*
*CNRS, LIS UMR 7020*
Marseille, France
leila.sakli@lis-lab.fr, 0000-0001-7321-9968

Seifeddine Ben Elghali
*Aix-Marseille Université*
*CNRS, LIS UMR 7020*
Marseille, France
seifeddine.benelghali@lis-lab.fr, 0000-0002-4557-2825

***Abstract*—Most Photovoltaic (PV) forecasting studies optimize prediction accuracy, but production systems also need safeguards and explanations that operators can inspect. Rather than proposing a new forecasting model, this paper introduces an agentic wrapper for day-ahead PV forecast delivery. The wrapper orchestrates a baseline predictor, applies runtime integrity checks, and corrects or flags outputs when basic physical and operational constraints are violated (e.g., nighttime consistency and capacity bounds). Each forecast is accompanied by structured explanation artifacts, such as triggered checks and reason codes, designed for auditability and integration with operational pipelines. Experiments on real PV plant data over a 24-hour horizon show that the proposed architecture matches the baseline accuracy while improving physical consistency and traceable behavior at runtime.**

***Index Terms*—Photovoltaic power forecasting, Hybrid ensemble forecasting, Agentic workflow, Model-agnostic explainability, Runtime validation, Auditability.**

## I. Introduction

In operational settings, PV forecasting pipelines face persistent data-quality and plausibility issues. Measurements may be noisy, partially missing due to sensor or communication faults, or physically inconsistent [1], [2]. Typical examples include non-zero production during nighttime or values exceeding plant nominal capacity [2]. Such conditions can degrade purely data-driven predictors and motivate the integration of physical constraints and sanity checks into the forecasting process [3]. Hybrid frameworks that mix physical components (e.g., solar-geometry and irradiance-related quantities) with learning-based models, often combined through ensembles and post-processing, have been shown to improve robustness and physical consistency across varying weather regimes [4]–[7].

Despite these advances, most studies still frame PV forecasting primarily as a prediction task optimized for error metrics [7]. In practice, forecasts are consumed inside software ecosystems, monitoring platforms, energy management systems, and decision-support tools, where reliability, traceability, and interpretability matter alongside numerical performance [8]–[10]. This motivates shifting the focus from a single value to a *forecast-delivery process* that includes runtime qualification and operator-facing evidence [10]–[13].

Accordingly, we introduce an agentic and explainable runtime wrapper architecture for day-ahead PV forecast delivery. The wrapper orchestrates an existing baseline predictor, qualifies its outputs through runtime integrity checks against basic physical and operational constraints (e.g., nighttime consistency and capacity bounds), and emits structured, machine-readable evidence (triggered checks, reason codes, and correction events) alongside the published forecast. Here, *agentic* refers to a deterministic, constraint-aware forecast-delivery loop that produces actionable artifacts (checks, reason codes, and correction events) alongside the forecast, rather than autonomous control or online learning. Experiments on real PV plant data in a 24-hour day-ahead setting show that the proposed architecture preserves baseline accuracy while improving physical consistency and traceable runtime behavior.

The remainder of this paper is organized as follows. Section II reviews related work. Section III presents the baseline forecasting pipeline. Section IV introduces the proposed agentic runtime wrapper and its explainability artifacts. Section V details the deployment architecture. Section VI reports the experimental evaluation. Section VII discusses limitations and future work, and Section VIII concludes.

## II. Background on Hybrid Forecasting and Agentic Systems

### A. Hybrid and ensemble PV forecasting with physical consistency

Day-ahead PV forecasting is commonly addressed through data-driven, physics-based, and hybrid approaches [4], [14]–[16]. Data-driven models can deliver strong accuracy when large volumes of high-quality data are available [4], yet their reliability may degrade in operational settings affected by sensor faults, missing values, and weather situations that are under-represented during training [10]. Physics-based models partly mitigate these issues by grounding PV production in known physical drivers (e.g., solar position, irradiance-related quantities, and plant characteristics) [17]. Their outputs are generally physically plausible and interpretable, but they can miss complex nonlinearities and time-dependent patterns observed in field measurements [5]. Hybrid methods aim to reconcile these trade-offs by injecting physical structure into learning-based predictors—through physics-informed features, simplified process components, or explicit constraints—thereby combining physical coherence with empirical flexibility. Prior work reports that hybrid models improve generalization across heterogeneous weather regimes and remain more robust under degraded data quality [5], [10], [14].

Orthogonal to hybridization, ensemble learning operates at the pipeline level by aggregating multiple standalone forecasters to reduce variance and model-specific biases [5]–[7]. In PV forecasting, ensembles often combine physics-based and machine-learning predictors (e.g., recurrent networks or tree-based regressors), but the defining characteristic is the output-level aggregation of independently trained models rather than an internal fusion within a single model. To further support operational reliability, constrained aggregation and post-processing layers have been proposed to enforce physical consistency, preventing implausible forecasts such as non-zero nighttime production or power values exceeding nominal capacity. Empirical studies on industrial PV datasets indicate that such physically constrained ensembles can improve plausibility and robustness without sacrificing predictive accuracy [5].

### B. Explainability in operational forecasting systems

When forecasts are integrated into operational software stacks, explanations must support runtime validation, auditability, and downstream trust [9], [18]. Many studies rely on post-hoc explanation methods to interpret black-box predictors and provide insights such as feature importance or sensitivity patterns [10]–[12]. While useful for offline analysis, these explanations are often weakly connected to operational constraints and to system-level decision processes. This motivates *explainability by design*, where transparency is embedded directly into the forecasting pipeline and architecture rather than added retrospectively [4]. In this view, explanation is treated as an operational artifact produced at runtime, alongside the forecast, to document integrity checks and the conditions under which the output is considered usable.

### C. From forecasting models to agentic systems

A related line of work frames forecasting as a constrained forecast-delivery loop that produces decision-ready artifacts in addition to numerical predictions [5]. From this perspective, forecasting is not only a point estimate but part of an explicit reasoning process that includes validation and explanation under operational constraints. Despite growing interest in such agentic perspectives [13], relatively few works structure PV forecasting pipelines as autonomous and explainable components suitable for operational deployment. Building on established hybrid and ensemble forecasting foundations [5], this work adopts a system-level view that bridges accurate predictors and operationally trustworthy forecasting services.

## III. Baseline Forecasting Pipeline and Mathematical Foundations

This section summarizes the baseline forecasting pipeline used in this work. The pipeline combines a physics-based estimator with two data-driven predictors (Bi-LSTM and XGBoosT), and integrates their outputs through constrained aggregation followed by a context-aware stacking stage [5]. In this paper, physical plausibility enforcement is treated as a runtime qualification step performed by the proposed wrapper.

### A. Problem definition

Let $P(t)$ denote the active power produced by a PV plant at time $t$. In the day-ahead setting, the forecasting task consists in predicting the future power output at horizon $h \in \{1, \dots, 24\}$:

$$\hat{P}(t+h) = f(\mathbf{X}(t)), \tag{1}$$

where $\mathbf{X}(t)$ gathers historical measurements, weather variables, and time-related features (e.g., cyclical encodings).

### B. Physics-based power estimation

A physics-oriented model offers a scientifically substantiated approximation of photovoltaic power generation [17]. The global horizontal irradiance is disaggregated into direct and diffuse elements, subsequently transformed into plane-of-array irradiance through the utilization of solar geometry correlations. Module temperature and DC power are estimated using standard PV performance models, followed by conversion to AC power. We denote the resulting estimate by $\hat{P}_{\mathrm{phys}}(t+h)$. This component improves physical coherence and interpretability, although it may not capture all local nonlinearities and short-term dynamics.

### C. Data-driven predictors

To complement the physical estimator, we use learning-based predictors to model nonlinear interactions and temporal dependencies. We consider: (i) a two-layer recurrent neural network, producing $\hat{P}_{\mathrm{LSTM}}(t+h)$, and (ii) a gradient-boosted decision tree model, producing $\hat{P}_{\mathrm{XGB}}(t+h)$. Both predictors leverage meteorological variables, cyclical time features, and lagged measurements.

### D. Constrained ensemble aggregation

The physics-based and data-driven forecasts are combined through constrained ridge aggregation:

$$\hat{P}_{\mathrm{ridge}}(t+h) = \sum_{m \in \{\mathrm{phys}, \mathrm{LSTM}, \mathrm{XGB}\}} w_m \, \hat{P}_m(t+h), \tag{2}$$

where weights $w_m$ are learned with regularization to promote numerical stability and to avoid degenerate solutions.

### E. Context-aware stacking

To improve robustness across operating conditions, a stacking layer refines the aggregated prediction using contextual features:

$$\hat{P}_{\mathrm{stack}}(t+h) = g\Big(\hat{P}_{\mathrm{ridge}}(t+h), \, \mathbf{C}(t)\Big), \tag{3}$$

where $\mathbf{C}(t)$ includes interaction features related to time-of-day, seasonality, and cloud-related indicators.

### F. Core forecast output

The baseline pipeline outputs a *core* day-ahead forecast after context-aware stacking:

$$\hat{P}_{\text{core}}(t+h) \triangleq \hat{P}_{\text{stack}}(t+h), \tag{4}$$

where $\hat{P}(\cdot)$ denotes a predicted quantity, the superscript "core" is an output label (not an exponent), and the subscript "stack" refers to the stacking stage of the pipeline. The symbol $\triangleq$ denotes a definition: the core forecast is defined as the stacking output. This core forecast is used as input to the runtime wrapper introduced in Section IV. In this paper, physical plausibility enforcement is treated as a runtime qualification step performed by the wrapper, in order to expose integrity checks and correction events as auditable artifacts.

### G. Role of the baseline forecasting pipeline

The hybrid ensemble pipeline described above is reused in this work as the predictive core. It provides the *core* day-ahead forecast $\hat{P}_{\text{core}}(t+h)$ through physical modeling, data-driven inference, constrained aggregation, and context-aware stacking, without altering the underlying models. This paper focuses on the system-level contribution introduced in Section IV: a deterministic runtime wrapper that qualifies $\hat{P}_{\text{core}}(t+h)$ against operational constraints, produces the published forecast $\hat{P}_{\text{pub}}(t+h)$, and emits structured explainability artifacts (e.g., triggered checks and reason codes) to support auditability in operational deployments.

## IV. Agentic Reasoning as a Runtime Wrapper

This section presents the proposed agentic reasoning methodology implemented as a runtime wrapper around the baseline forecasting pipeline. The goal is not to modify the underlying predictive models, but to make forecast delivery explicit and auditable by organizing it into a small set of reasoning stages that support validation, explainability, and operational use. The wrapper is deterministic and designed for industrial deployment: each output is produced by a fixed sequence of steps and is accompanied by machine-readable evidence.

The forecast-delivery loop is illustrated in Fig.1 and follows five stages: *Perception*, *Inference*, *Validation*, *Explanation*, and *Action*. Importantly, this is not introduced as a conceptual abstraction: each stage corresponds to standard operations commonly found in practical forecasting systems (data checks, model inference, constraint enforcement, and structured reporting).

### A. Scope and motivation

In operational PV systems, forecasts are not consumed as isolated numerical values but as inputs to monitoring, planning, and energy management workflows [9], [19]. In this context, stakeholders need to assess whether an output is physically plausible, understand what checks were applied, and obtain traceable information that can be audited downstream [11]. The proposed agentic wrapper addresses these requirements by transforming the forecasting pipeline into an explicit forecast-delivery process with runtime qualification and built-in explanations. Here, *agentic* refers to the autonomous execution of a lightweight reasoning loop under physical and operational constraints, and to the emission of actionable artifacts alongside the forecast.

### B. Wrapper inputs and outputs

At each issue time $t$, the wrapper consumes (i) plant measurements and context (historical PV power, weather variables, timestamps) and (ii) the baseline predictor output $\hat{P}_{\text{core}}(t+h)$ for $h \in \{1, \ldots, 24\}$. It produces (a) the published forecast $\hat{P}_{\text{pub}}(t+h)$ and (b) a structured explanation payload that reports which integrity checks were triggered and why.

### C. Perception

The perception stage acquires and preprocesses the information required for day-ahead forecasting. Inputs include historical PV power, weather variables, and temporal context. Time-related variables are encoded using cyclical representations to preserve diurnal and seasonal periodicities. This stage performs basic data-quality checks (missing values, timestamp consistency, and identification of nighttime periods) and outputs a validated feature set for inference.

### D. Inference

During inference, the baseline forecasting pipeline (Section III) is executed to produce a preliminary forecast. The baseline combines a physics-based estimator with data-driven predictors through constrained aggregation and context-aware refinement. The output of this stage is the core forecast $\hat{P}_{\text{core}}(t+h)$, which is not yet qualified against operational constraints.

### E. Validation

The validation stage evaluates the core forecast against basic physical and operational constraints and applies deterministic corrections when needed. We denote by $\mathcal{P}(\cdot)$ the physical plausibility operator implementing these checks, and the published forecast is obtained as

$$\hat{P}_{\text{pub}}(t+h) = \mathcal{P}\left(\hat{P}_{\text{core}}(t+h)\right). \tag{5}$$

At runtime, the wrapper logs (i) whether a correction was applied and (ii) which check(s) were activated, exposed as machine-readable reason codes. The main codes used in this work are:

- `NIGHT_ZERO`: nighttime timestamp; enforce $\hat{P}_{\text{pub}}(t+h) = 0$.
- `CAP_BOUND`: capacity violation; clip to the admissible range (in p.u.: $[0, 1]$).
- `NON_NEG`: negative value; enforce non-negativity by clipping to zero.
- `ANOMALY`: implausible output flagged by simple sanity rules (e.g., isolated abrupt jumps/outliers); apply a conservative correction or record the event for downstream inspection.

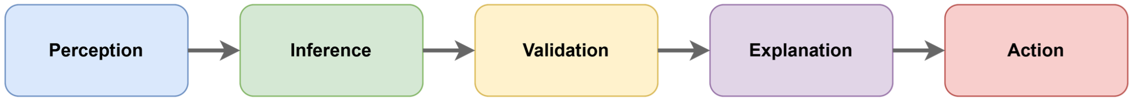


Fig. 1. Agentic reasoning workflow structured as a sequence of explicit reasoning stages.

Each published point $\hat{P}_{\text{pub}}(t+h)$ is delivered with a payload containing the triggered code(s) and the correction event, including $\Delta(t+h) = \hat{P}_{\text{pub}}(t+h) - \hat{P}_{\text{core}}(t+h)$.

### *F. Explanation*

Explainability is achieved *by design* and is *model-agnostic*. Rather than interpreting internal representations of learning-based predictors, the wrapper derives explanations from observable pipeline events (orchestration traces, triggered integrity checks, and applied corrections) and emits structured artifacts (check identifiers, reason codes, and indicators) to support auditability and operational interpretability.

### *G. Action*

The action stage delivers the published forecast together with its associated explanation payload to downstream systems such as monitoring dashboards and energy management platforms. This stage does not perform autonomous control; it ensures that forecast outputs are accompanied by the contextual information needed for informed human or system-level decisions.

## V. System Architecture and Deployment

This section describes the software architecture supporting the proposed agentic forecasting wrapper. The objective is to operationalize the baseline forecasting pipeline (Section III) together with the runtime reasoning loop (Section IV) within a modular architecture suitable for industrial deployment. Our contribution focuses on architectural organization and integration interfaces rather than algorithmic innovation.

### *A. Architectural overview*

The proposed system follows a service-oriented architecture in which forecasting is delivered as a dedicated microservice. The microservice encapsulates the full forecast-delivery loop: perception, inference, validation, explanation, and action, and exposes well-defined interfaces to upstream data sources and downstream operational applications. This modular organization promotes maintainability, fault isolation, and reproducible behavior in production settings.

### *B. Data ingestion and management*

Operational data (historical PV power, weather variables, and temporal context) are ingested through a dedicated ingestion component that performs basic preprocessing and consistency checks. Data can be persisted in a storage layer (e.g., database or data lake) for traceability and offline analysis, while the forecasting microservice retrieves the required inputs at each issue time $t$. Separating ingestion from forecasting improves robustness to heterogeneous data sources and simplifies maintenance.

### *C. Forecasting wrapper as a microservice*

The agentic wrapper is deployed as an independent microservice responsible for executing the reasoning loop described in Section IV. It consumes input features and produces (i) the published day-ahead forecast $\hat{P}_{\text{pub}}(t+h)$ and (ii) a structured explanation payload (e.g., triggered integrity checks and reason codes). The service exposes a minimal API for integration, enabling asynchronous consumption by monitoring dashboards, energy management platforms, or decision-support tools.

### *D. Observability and traceability*

To support operational trust, the architecture includes observability mechanisms such as logging, monitoring, and trace storage. At runtime, the wrapper records integrity-check activations and correction events, enabling auditability of forecast delivery without requiring access to internal model parameters. These artifacts are designed to be machine-readable for downstream systems and interpretable by operational stakeholders.

### *E. Integration with operational systems*

Published forecasts and their associated explanation artifacts are delivered to downstream operational systems through decoupled interfaces. This design allows operational users to consume forecasts enriched with validation evidence, supporting informed decision-making while keeping predictive models and internal logic encapsulated within the forecasting service.

## VI. Experimental Evaluation

This section evaluates the proposed explainable runtime wrapper for day-ahead PV forecast delivery. The evaluation has two objectives: (i) verify that adding the wrapper preserves the predictive performance of the baseline pipeline under nominal conditions, and (ii) quantify operational benefits in terms of physical consistency and traceable execution-time behavior enabled by validation and explanation artifacts.

### *A. Experimental setup and dataset*

We consider a day-ahead setting with horizon $h \in \{1, \ldots, 24\}$ at hourly resolution on real PV plant data. The dataset consists of historical PV power measurements and corresponding meteorological variables. Preprocessing follows the perception stage described in Section IV, including timestamp alignment, missing-value handling, and cyclical encoding of time-related features. The baseline forecasting pipeline (Section III) is used without modification to the predictive models, and the proposed wrapper (Section IV) is applied to the resulting core forecasts. Training and evaluation follow a chronological split to reflect realistic forecasting conditions and avoid temporal leakage; all reported results are computed on the test set.

### B. Evaluation metrics

Forecasting accuracy is assessed using RMSE, MAPE together with sMAPE to account for scale effects and near-zero targets. All errors are reported in per-unit (p.u.) after normalizing power by the plant nominal capacity.

To quantify operational integrity, we report runtime indicators produced by the wrapper. For clarity, $\hat{P}_{\text{core}}(t+h)$ and $\hat{P}_{\text{pub}}(t+h)$ denote the core forecast and the published forecast after runtime validation, respectively. Let $C_{t+h}$ denote the set of basic operational constraints (e.g., nighttime consistency, non-negativity, and capacity bounds), and let $N$ be the total number of evaluated forecast points on the test set (i.e., the number of valid $(t, h)$ pairs). The violation rate (Viol%) is defined as

$$v_{t,h} = \begin{cases} 1, & \hat{P}_{\text{pub}}(t+h) \text{ violates } C_{t+h}, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

$$\text{Viol\%} = \frac{100}{N} \sum_{t,h} v_{t,h} \tag{7}$$

$$c_{t,h} = \begin{cases} 1, & \hat{P}_{\text{pub}}(t+h) \neq \hat{P}_{\text{core}}(t+h), \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

$$\text{Corr\%} = \frac{100}{N} \sum_{t,h} c_{t,h} \tag{9}$$

$$\Delta(t+h) = \hat{P}_{\text{pub}}(t+h) - \hat{P}_{\text{core}}(t+h) \tag{10}$$

$$\overline{|\Delta|} = \frac{1}{N} \sum_{t,h} |\Delta(t+h)| \tag{11}$$

Finally, we summarize the most frequent triggered checks (reason codes) to provide evidence of explainability-by-design at runtime and to characterize typical integrity issues encountered in operational conditions.

### C. Results

Table I reports day-ahead (24 h) accuracy and runtime integrity metrics in per-unit (p.u.). The physical model yields physically consistent behavior by construction, with a zero violation rate (Viol%=0) under the considered constraints. For the ensemble core predictor, the runtime wrapper preserves predictive performance (RMSE/MAE/MAPE/sMAPE change marginally) while enforcing strict physical consistency at delivery time. In particular, the wrapper eliminates nighttime leakage by resetting non-zero nighttime predictions to zero and applying capacity bounds, which drives the post-validation violation rate to a very low level. Corrections remain sparse (Corr%) and small on average ($\overline{|\Delta|}$), indicating that the wrapper acts as a lightweight, deterministic qualification layer rather than a new predictor. The most frequently triggered rule is `NIGHT_ZERO`, consistent with basic operational constraints.

### D. Qualitative behavior on a representative period

Fig. 2 illustrates a representative test period. The green, orange, and violet curves correspond to the physical model, Bi-LSTM, and XGBoost base predictors, respectively, while the red dashed curve shows the published forecast $\hat{P}_{\text{pub}}$ obtained after runtime validation of the ensemble core forecast $\hat{P}_{\text{core}}$. The figure highlights how the validation checkpoint produces localized deterministic corrections, primarily by resetting spurious nighttime production to zero and bounding peaks, while leaving the daytime profile largely unchanged. This qualitative behavior matches the quantitative evidence Table I: a low correction rate (Corr%) and a small average correction magnitude ($\overline{|\Delta|}$) coexist with strong gains in physical plausibility at delivery time.

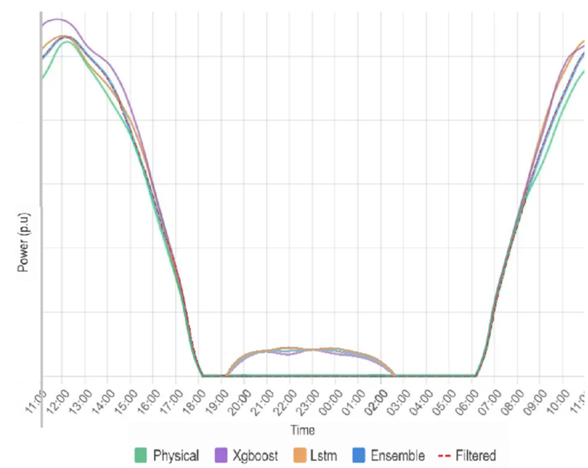


Fig. 2. Representative day-ahead PV power forecasts (p.u.) over a test period. Measured power is compared with the core forecast $\hat{P}^{\text{core}}$ and the published forecast $\hat{P}^{\text{pub}}$ after runtime validation.

### E. Impact of runtime validation and explanation artifacts

Beyond numerical accuracy, the wrapper provides operational value by making integrity enforcement explicit and auditable. Each triggered check (e.g., `NIGHT_ZERO` and capacity bounds) generates a structured correction record that specifies the rule identifier, the original value, the corrected value, and the timestamp/horizon. These artifacts externalize execution-time behavior that is typically implicit in forecasting pipelines, enabling downstream systems to (i) gate forecast usage based on the usability status, (ii) diagnose degraded input situations, and (iii) perform post-hoc analyses of when and why corrections occurred. Importantly, this transparency is achieved without access to internal model parameters and without retraining the baseline predictor.

TABLE I
DAY-AHEAD (24H) FORECASTING PERFORMANCE AND RUNTIME INTEGRITY ON THE TEST SET

| Model | RMSE | MAE | MAPE(%) | sMAPE(%) | Viol% | Corr% | $\overline{\|\Delta\|}$ | Top checks |
|---|---|---|---|---|---|---|---|---|
| Physical (pvlib) | 0.029 | 0.016 | 26.858 | 35.533 | 0.0 | – | – | – |
| XGBoosT | 0.132 | 0.076 | 82.878 | 136.806 | 11.2 | – | – | – |
| Bi-LSTM | 0.308 | 0.176 | 382.020 | 104.700 | 12.8 | – | – | – |
| Ensemble (core predictor) $\hat{P}_{\mathrm{core}}$ | 0.065 | 0.038 | 37.342 | 61.050 | 6.1 | – | – | – |
| Core + runtime wrapper (ours) $\hat{P}_{\mathrm{pub}}$ | 0.066 | 0.039 | 37.900 | 62.200 | 0.4 | 5.2 | 0.007 | NIGHT_ZERO |

## VII. CONCLUSION

This paper introduced an agentic and explainable runtime wrapper architecture for day-ahead PV forecast delivery. Rather than modifying the underlying predictive models, the contribution focuses on operationalizing forecasting as a deterministic, constraint-aware forecast-delivery loop: a baseline predictor produces a core forecast, the wrapper qualifies it against basic physical and operational constraints, and machine-readable evidence is emitted together with the published output.

Experimental results in a 24-hour day-ahead setting indicate that the wrapper preserves the baseline predictive performance while improving the physical consistency and traceability of delivered forecasts. Importantly, the generated artifacts (triggered checks, reason codes, and correction events) provide process-level, model-agnostic explanations that can be directly consumed by operational software stacks, supporting auditability and operator trust during degraded data conditions.

Several limitations should be noted. First, the qualification logic is deterministic and rule-based; while this design improves transparency and predictable behavior, it does not adapt its validation policy to changing operating regimes. Second, the provided explainability is primarily operational and does not aim to interpret internal model representations (e.g., feature attributions), which may be useful for offline analysis. Finally, the evaluation is restricted to a day-ahead horizon and a single-plant setting.

Future work will extend the evaluation to multiple PV plants and longer horizons (e.g., intra-day and multi-day), to assess robustness across heterogeneous climates and operating regimes. We will stress-test the wrapper under degraded-input scenarios (prolonged missingness, timestamp misalignment, sensor faults) and attach explicit data-quality indicators to the evidence payload. Finally, we will investigate uncertainty-aware qualification, where validation thresholds and correction policies are informed by predictive uncertainty to better separate model uncertainty from upstream data issues.

## ACKNOWLEDGMENT

This work has been carried out within the framework of the European Union's Horizon Europe research and innovation programme under Grant Agreements No. 101096946 (FlexCHESS) and No. 101096836 (MASTERPIECE)

## REFERENCES

[1] Y. Guo, Q. Han, T. Li, H. Fu, M. Liang, and S. Zhang, "Robust photovoltaic power forecasting model under complex meteorological conditions," *Mathematics*, vol. 13, p. 1783, 2025, doi: 10.3390/math13111783.

[2] W. Liu, C. Ren, and Y. Xu, "PV generation forecasting with missing data: A super-resolution perception approach," *IEEE Trans. Sustain. Energy*, 2020, doi: 10.1109/TSTE.2020.3029731.

[3] L. Yang, Z. Huang, X. Mo, and T. Luo, "Enhanced GAIN-Based Missing Data Imputation for a Wind Energy Farm SCADA System," *Electronics*, vol. 14, no. 8, p. 1590, 2025, doi: 10.3390/electronics14081590.

[4] N. Rahimi, S. Park, W. Choi, B. Oh, S. Kim, Y.-h. Cho, S. Ahn, C. Chong, D. Kim, C. Jin, and D. Lee, "A comprehensive review on ensemble solar power forecasting algorithms," *J. Electr. Eng. Technol.*, published Jan. 12, 2023.

[5] L. Sakli and S. Ben El Ghali, "PV forecasting with constrained ensemble learning and context-aware stacking," in *Proc. 37th Int. Conf. on Tools with Artificial Intelligence (ICTAI)*, Athens, Greece, Nov. 3-5, 2025, doi: 10.1109/ICTAI66417.2025.00136.

[6] W. F. Hassen and M. Challouf, "Long short-term renewable energy sources prediction for grid-management systems based on stacking ensemble model," *Energies*, vol. 17, no. 13, p. 3145, 2024, doi: 10.3390/en17133145.

[7] W. Khan, S. Walker, and W. Zeiler, "Improved solar photovoltaic energy generation forecast using deep learning-based ensemble stacking approach," *Energy*, vol. 240, p. 122812, 2021, doi: 10.1016/j.energy.2021.122812.

[8] X. Huang, T. Zhao, B. Huang, Z. Zhang, and Y. Meng, "Advancing energy system optimization via data-centric task-oriented forecasting: An application in PV-battery operation," *Applied Energy*, vol. 378, p. 124753, Oct. 2024, doi: 10.1016/j.apenergy.2024.124753.

[9] K. Kravari, M. Roussi, K. Ladomenou, A. Thysiadou, and M. Chalaris, "Intelligent agent orchestration for climate-resilient energy systems: Autonomous management of renewable integration and distributed resources," *E3S Web of Conferences*, vol. 638, p. 02009, 2025, doi: 10.1051/e3sconf/202563802009.

[10] Z. Wang, H. Liu, S. Wu, N. Liu, X. Liu, Y. Hu, and Y. Fu, "Explainable time-varying directional representations for photovoltaic power generation forecasting," *Journal of Cleaner Production*, vol. 468, p. 143056, 2024, doi: 10.1016/j.jclepro.2024.143056.

[11] M. Kuzlu *et al.*, "Gaining insight into solar photovoltaic power generation forecasting utilizing explainable artificial intelligence tools," *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.3031477.

[12] R. M. Rizk-Allah, L. M. Abouelmagd, A. Darwish, V. Snášel, and A. E. Hassanien, "Explainable AI and optimized solar power generation forecasting model based on environmental conditions," *PLOS ONE*, vol. 19, no. 10, art. no. e0308002, Oct. 2024, doi: 10.1371/journal.pone.0308002.

[13] Z. Yan *et al.*, "Probabilistic PV power forecasting by a multimodal method using GPT-agent," in *Proc. IEEE Int. Conf. Industrial Electronics and Applications (ICIEA)*, 2024, doi: 10.1109/ICIEA61579.2024.10665072.

[14] W. Castillo-Rojas and J. Salinas, "Forecasting models applied in solar photovoltaic and wind energy: A systematic mapping study," *IEEE Access*, pp. 1-1, 2024.

[15] M. Gupta, A. Arya, U. Varshney, J. Mittal, and A. Tomar, "A review of PV power forecasting using machine learning techniques," *Prog. Eng. Sci.*, vol. 2, no. 1, p. 100058, 2025, doi: 10.1016/j.pes.2025.100058.

[16] S. Meisenbacher, B. Heidrich, T. Martin, R. Mikut, and V. Hagenmeyer, "AutoPV: Automated photovoltaic forecasts with limited information

using an ensemble of pre-trained models," in *Proc. ACM Int. Conf.*, 2023, pp. 386-414, doi: 10.1145/3575813.3597348.
[17] W. F. Holmgren, C. W. Hansen, and M. A. Mikofski, "pvlib python: A python package for modeling solar energy systems," *J. Open Source Softw.*, vol. 3, no. 29, p. 884, 2018, doi: 10.21105/joss.00884.
[18] V. Zaccaria, C. Masiero, D. Dandolo, and G. A. Susto, "Enabling efficient and flexible interpretability of data-driven anomaly detection in industrial processes with AcME-AD," in *Proc. 8th Int. Conf. on Control, Decision and Information Technologies (CoDIT)*, Jul. 2024, p. 1375, doi: 10.1109/CoDIT62066.2024.10708099.
[19] M. Husein, E. J. Gago, B. Hasan, and M. C. Pegalajar, "Towards energy efficiency: A comprehensive review of deep learning-based photovoltaic power forecasting strategies," *Heliyon*, vol. 10, no. 13, Jun. 2024, doi: 10.1016/j.heliyon.2024.e33419.