Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2026) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

30th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2026)

Hybrid Intelligent PV Forecasting with Runtime Validation and Qualification

Leila Sakli[a,*], Seifeddine Ben Elghali[a]

[a] *LIS UMR CNRS 7020, Aix-Marseille Université, Marseille, France*

Abstract

Photovoltaic (PV) forecasting is increasingly used in time-critical energy applications, where heterogeneous input streams, including local measurements, third-party weather forecasts, and contextual signals, may be missing, delayed, or silently misaligned at runtime. In such conditions, predictions may remain numerically plausible while being unreliable for downstream use. This paper presents a runtime validation and qualification layer for hybrid PV forecasting pipelines. The proposed approach wraps a hybrid ensemble predictor combining physics-based and data-driven models through constrained aggregation and context-aware stacking, without modifying its inference core. It externalizes post-stacking plausibility checks as an explicit validation stage and assigns a qualification status (Valid, Corrected, Withheld) to each forecast. In addition, the runtime layer produces auditable traces that record input-quality issues, constraint violations, correction actions, and fallback activations. Experiments on real-world PV data under nominal, degraded, and severe input conditions show that the approach preserves forecasting performance while improving operational transparency. In severe degradation scenarios, the validated pipeline maintains competitive accuracy ($R^2 \geq 0.86$), corrects up to 15.6% of implausible forecasts, exposes fallback activations in 42.3% of inference cycles, and withholds 7.1% of forecasts judged unreliable. These results show that runtime validation and qualification can improve the operational usability of hybrid PV forecasting systems without redesigning the underlying prediction models.

Keywords: PV forecasting; hybrid intelligent systems; runtime validation; forecast qualification; knowledge-based constraints; auditable traces

1. Introduction

Photovoltaic power forecasting is increasingly embedded in time-critical operational loops for grid balancing, storage scheduling, and flexibility services [1, 2, 3]. In real deployments, forecasting pipelines consume heterogeneous inputs, including local PV measurements, third-party weather forecasts, and contextual signals, whose availability and integrity may degrade at runtime [4]. Delays, missing segments, silent cross-source misalignment, and platform outages can occur even when data streams appear nominal. A key failure mode then emerges: forecasts may remain

* Corresponding author.
E-mail address: leila.sakli@lis-lab.fr (Leila Sakli).

1877-0509

numerically plausible while being unsuitable for downstream use because the pipeline provides no explicit qualification of forecast usability under current execution-time conditions [5].

At the model level, hybrid and ensemble approaches have improved robustness by combining complementary predictors and learning context-dependent fusion mechanisms [6, 7, 8]. Constrained aggregation and context-aware stacking [9] stabilize contributions across predictors, but stacking outputs do not inherently guarantee physical plausibility. Some predictors embed internal fallback behavior to preserve continuity under missing inputs [4, 10], but these mechanisms are typically not exposed as first-class outcomes: downstream systems rarely observe when a fallback was activated or whether a forecast should be withheld [11].

From an operational pipeline perspective, there is growing recognition that energy systems, especially those integrating distributed renewable resources, require explicit runtime qualification, interpretability, and auditability to support human oversight and automated decision-making [11, 12]. This work shifts the focus from prediction accuracy to *decision-aware forecasting*, where the objective is not only to produce a forecast but to determine, through constraint-based reasoning, whether that forecast should be used, corrected, or withheld at runtime. This need is particularly acute in contexts where forecasts drive high-stakes actions (e.g., market participation, storage dispatch) and where regulatory or safety requirements demand traceability of automated decisions.

This paper addresses the gap between robust forecasting models and operational decision-making by introducing a runtime validation and qualification layer, formalized as a rule-based decision system, that wraps a hybrid ensemble predictor combining a physics-based model with data-driven learners. We frame this contribution as a knowledge-based decision layer that augments predictive models with explicit reasoning capabilities at inference time. Our contribution is fourfold:

- **Post-stacking validation as an explicit runtime stage:** We externalize physical plausibility filtering, previously embedded within the predictor, into an inspectable validation component that enforces domain constraints and records corrective actions.
- **Forecast qualification interface:** We introduce a lightweight qualification scheme that assigns each forecast a usability status (Valid, Corrected, Withheld) together with execution-time metadata, enabling downstream systems to make informed use of forecast outputs.
- **Auditable runtime traces:** We generate structured trace records for each inference cycle, capturing input status, constraint violations, fallback activations, and correction actions to support post-hoc analysis, debugging, and compliance verification.
- **Operational evaluation:** We evaluate the proposed runtime layer using both standard forecasting metrics (MAE, RMSE, R^2) and operational indicators, including constraint-violation rates, correction frequency, fallback rate, and withholding rate under nominal, degraded, and severe input conditions.

2. Related Work

2.1. PV Forecasting Models

PV power forecasting has evolved from statistical methods to hybrid ML approaches [1, 6, 7], with ensemble methods dominating recent benchmarks [8, 13]. Physics-based models such as pvlib [14] provide interpretable forecasts grounded in solar geometry but degrade when meteorological inputs are missing [4, 10]. Data-driven models (XGBoost, LSTM, Transformers [16]) capture nonlinear temporal dependencies [13, 15] but may face challenges when applied to conditions different from training data [15]. Hybrid approaches combining both paradigms through constrained aggregation and context-aware stacking [9] stabilize predictor contributions, but do not expose internal mechanisms such as fallback activations or constraint corrections as first-class operational outcomes [10].

2.2. Trustworthiness in Energy Systems

The integration of renewable energy sources into grid operations has heightened the need for trustworthy forecasting systems. Key dimensions include reliability under input degradation [2, 13], interpretability of predictions [11], auditability of automated decisions [12, 19], and robustness to missing or corrupted inputs [10]. Recent work on intelligent agent orchestration explores autonomous management of distributed energy resources [3], while interpretable anomaly detection frameworks highlight the value of attaching explanatory metadata to model outputs [11].

2.3. Post-Prediction Validation and Qualification

Post-prediction validation enforces domain constraints on model outputs to ensure physical plausibility. In PV forecasting, common constraints include capacity bounds, day/night consistency, and outlier control. While widely recognized as essential [9, 10], constraint enforcement is typically embedded as an implicit post-processing step rather than an inspectable stage exposed to downstream consumers.

In the broader ML engineering literature, data validation for production pipelines has received significant attention. Breck et al. [19] proposed a schema-based data validation system deployed at scale in continuous ML pipelines, which detects anomalies in input data before model training. Similarly, recent end-to-end frameworks for energy forecasting [12] advocate formal decision gates and traceable artifacts at each pipeline phase, but focus primarily on data preparation and model training rather than on runtime qualification of predictions after inference. Task-oriented forecasting [2] aligns forecast generation with downstream objectives but addresses model-level optimization rather than runtime traceability.

Our work differs from these approaches in two respects. First, existing approaches do not explicitly support runtime decision-making at inference time: they validate input data before training or optimize model-level objectives, but do not reason about whether an individual prediction should be consumed, corrected, or withheld after it has been produced. Second, we attach explicit usability metadata to each individual forecast through a rule-based reasoning mechanism, enabling downstream systems to reason about forecast reliability on a per-prediction basis.

3. Baseline Predictor

The runtime validation layer proposed in this paper wraps an existing hybrid ensemble predictor [9], which we refer to as the *inference core*. This section reviews the baseline architecture to clarify what is produced by the predictor itself before any runtime qualification and traceability are attached.

3.1. Predictor Architecture

The inference core relies on three complementary models [9]: (i) a *physics-based model* (pvlib [14]) that computes expected PV power from solar geometry and system specifications, providing domain-grounded forecasts whose accuracy depends on meteorological input quality; (ii) a *gradient-boosting regressor* (XGBoost) that captures nonlinear relationships between tabular weather, temporal, and historical power features; and (iii) a *bidirectional LSTM* that models sequential PV-weather dynamics and diurnal/seasonal patterns through cyclic temporal encodings.

3.2. Ensemble Inference: Aggregation and Stacking

These predictors are combined through a two-level ensemble mechanism.

Constrained aggregation. At a first level, base outputs are fused through constrained Ridge regression to obtain a stable aggregated prediction.

Context-aware stacking. At a second level, an XGBoost meta-learner performs context-aware stacking to combine base predictions based on contextual features (e.g., temporal and meteorological covariates). This stage adapts the fusion behavior across operating regimes and input conditions.

3.3. Baseline Output and Motivation for a Runtime Wrapper

At runtime, the inference core produces a point forecast over a horizon H, denoted $\hat{\mathbf{y}}(t) = (\hat{y}_{t+1}, \ldots, \hat{y}_{t+H})$. In the original architecture, physical plausibility was enforced through post-prediction filtering, including statistical bounding, capacity clipping, and daylight enforcement. In operational settings, downstream consumers benefit from ad-

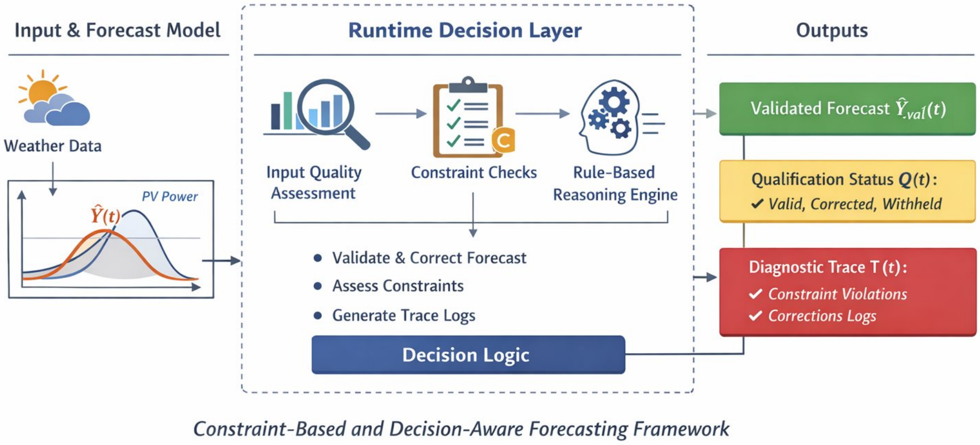

Fig. 1. Constraint-based and decision-aware forecasting framework: the runtime decision layer wraps the forecast model, producing a validated forecast, a qualification status, and a diagnostic trace for each inference cycle.

ditional first-class signals beyond a point forecast, including explicit usability qualification and auditable traces of execution-time conditions and triggered safeguards. These requirements motivate the runtime layer introduced next.

4. Runtime Validation Layer

This section presents the core contribution: a runtime validation and qualification layer that wraps the baseline predictor described in Section 3. The proposed layer externalizes plausibility checks as an explicit runtime checkpoint, attaches usability metadata, and generates auditable traces without redesigning the underlying inference core.

4.1. Architecture Overview

Formally, the runtime layer implements a constraint-based decision function:

$$\mathcal{F} : (\hat{\mathbf{y}}(t),\ q(t),\ C) \longrightarrow (\hat{\mathbf{y}}^{\mathrm{val}}(t),\ Q(t),\ \mathcal{T}(t)), \tag{1}$$

where $\hat{\mathbf{y}}(t)$ is the baseline forecast, $q(t)$ is the input quality score, C is the set of domain constraints, $\hat{\mathbf{y}}^{\mathrm{val}}(t)$ is the validated forecast, $Q(t) \in \{\textsc{Valid}, \textsc{Corrected}, \textsc{Withheld}\}$ is the qualification status, and $\mathcal{T}(t)$ is the diagnostic trace. This formulation makes explicit that the runtime layer is a rule-based reasoning system that produces decision-oriented outputs, not merely a post-processing filter.

The runtime layer executes a fixed five-stage loop at each inference cycle: input collection, input assessment, inference, validation, and qualification/delivery (Fig. 1). By separating inference from constraint-based reasoning, the proposed wrapper preserves the baseline predictor unchanged while making plausibility enforcement explicit, attaching a usability status, and producing an auditable trace for downstream consumers. The forecasting engine is treated as a black box: the wrapper only requires access to the predicted trajectory and timestamps. When available, the wrapper additionally consumes an optional runtime flag indicating that the inference core activated a fallback mode; no retraining or architectural modifications are required.

The five stages operate as follows: (i) *input collection* gathers PV measurements, weather forecasts, and contextual signals; (ii) *input assessment* evaluates execution-time quality (missingness, staleness, cross-source alignment); (iii) *inference* invokes the unchanged baseline predictor; (iv) *validation* enforces physical plausibility constraints and emits correction records; (v) *qualification and delivery* assigns a usability indicator (Valid, Corrected, or Withheld) and returns the qualified forecast with its trace record.

4.2. Validation Stage

The validation stage implements constraint-based reasoning through knowledge-based physical plausibility checks on the baseline predictor output. For each inference cycle at forecast origin t and for each horizon step $h \in \{1, \dots, H\}$, it maps the baseline forecast $\hat{y}_{t+h}$ to a corrected value $\hat{y}^{\mathrm{val}}_{t+h}$ through three rule-based checks.

Statistical outlier bounding. We bound unusually large predictions using a μ+3σ cap estimated per site from historical baseline predictions. Let $\mu_{\hat{y}}$ and $\sigma_{\hat{y}}$ denote the mean and standard deviation of historical $\hat{y}$:

$$\hat{y}^{\mathrm{bnd}}_{t+h} = \min(\hat{y}_{t+h},\ \mu_{\hat{y}} + 3\sigma_{\hat{y}}). \tag{2}$$

Capacity cap. We then cap the bounded prediction using a fixed safety factor s_{cap} applied to the nominal installed capacity:

$$\hat{y}^{\mathrm{cap}}_{t+h} = \min(\hat{y}^{\mathrm{bnd}}_{t+h},\ s_{\mathrm{cap}} \cdot P_{\mathrm{DC0}}), \tag{3}$$

where P_{DC0} is the nominal installed (DC) capacity and we set $s_{\mathrm{cap}} = 1.5$.

Day/night enforcement. Finally, we enforce daytime consistency using the solar elevation angle elev(t) computed from solar geometry [14]:

$$\hat{y}^{\mathrm{val}}_{t+h} = \begin{cases} 0, & \mathrm{elev}(t+h) \leq 0, \\ \hat{y}^{\mathrm{cap}}_{t+h}, & \text{otherwise.} \end{cases} \tag{4}$$

Correction records. Whenever any constraint is triggered, the validation stage emits a correction record containing the constraint type, original value, corrected value, and timestamp. These records are included in the trace and enable post-hoc analysis of validation behavior. The contribution does not lie in the individual constraints themselves, which are well-established in PV engineering, but in their integration into a unified reasoning layer that produces decision-oriented outputs and explicit operational semantics.

4.3. Qualification Stage

The qualification stage assigns a usability indicator based on input quality and validation outcomes. We use three labels: VALID (usable), CORRECTED (usable with caution), and WITHHELD (abstain).

Correction magnitude. For a multi-step horizon $\hat{\mathbf{y}}(t)$ and its validated counterpart $\hat{\mathbf{y}}^{\mathrm{val}}(t)$, we quantify the maximum correction as:

$$\Delta(t) = \max_{1 \leq h \leq H} |\hat{y}_{t+h} - \hat{y}^{\mathrm{val}}_{t+h}|. \tag{5}$$

Input quality score. We define an operational input quality score $q(t) \in [0, 1]$ that aggregates three runtime checks: missingness, staleness, and temporal misalignment. Let $m(t)$, $s(t)$, and $a(t)$ be normalized penalties in $[0, 1]$, where 0 means "no issue" and 1 means "severe issue". The overall score is:

$$q(t) = 1 - (w_m\, m(t) + w_s\, s(t) + w_a\, a(t)), \tag{6}$$

Table 1. Runtime layer hyperparameters.

Parameter	**Value**	**Description**
s_{cap}	1.5	Capacity safety factor
τ_q	0.4	Withholding threshold on $q(t)$
τ_Δ	0.10 p.u.	Correction magnitude threshold
w_m	0.5	Missingness weight
w_s	0.3	Staleness weight
w_a	0.2	Misalignment weight

with $w_m + w_s + w_a = 1$ and $w_m, w_s, w_a \geq 0$.

Missingness penalty $m(t)$ is the fraction of required inputs missing at time t. Staleness penalty $s(t)$ is computed from the age of the most recent available observation relative to the expected update rate, clipped to $[0, 1]$. Misalignment penalty $a(t)$ captures timestamp inconsistencies between PV measurements and exogenous inputs, also clipped to $[0, 1]$.

Decision rules. Given the input quality score $q(t)$ (Eq. 6) and correction magnitude $\Delta(t)$ (Eq. 5), the qualification status $Q(t)$ is determined by the following priority-ordered rules:

$$Q(t) = \begin{cases} \textsc{Withheld}, & \text{if } q(t) < \tau_q, \\ \textsc{Corrected}, & \text{if } \Delta(t) > \tau_\Delta \text{ or fallback activated}, \\ \textsc{Valid}, & \text{otherwise}. \end{cases} \tag{7}$$

This three-level qualification scheme follows the principle of multi-level monitoring with escalating severity, as established in drift detection approaches where distinct In-Control, Warning, and Drift thresholds govern the transition between operational regimes based on observed statistics [18]. In our setting, the three levels correspond to increasingly severe runtime conditions rather than concept drift, but the underlying logic is analogous: a statistically grounded escalation from nominal operation to cautious use to abstention.

The thresholds τ_q and τ_Δ are treated as site-specific policy hyperparameters rather than learned parameters. Their values are derived from the validation set as follows. The withholding threshold $\tau_q = 0.4$ is set at the point where the input quality score indicates that more than 60% of the quality budget is consumed; below this level, the forecasting error on the validation set exceeds twice the nominal MAE, making the forecast operationally unreliable. The correction threshold $\tau_\Delta = 0.10$ p.u. of installed capacity aligns with standard PV monitoring practice, where deviations exceeding 10% of nominal capacity are considered significant for operational decision-making [17]. The input quality weights ($w_m = 0.5, w_s = 0.3, w_a = 0.2$) reflect the relative operational impact of each degradation type: missingness has the strongest effect on forecast reliability, as it removes information entirely, whereas staleness and misalignment degrade information quality more gradually. These weights are consistent with data quality hierarchies reported in quality-aware energy forecasting frameworks [17]. Table 1 summarizes all hyperparameters. By design, all thresholds are transparent, auditable, and adjustable by the site operator without retraining the forecasting model. Crucially, the goal of these parameters is not to optimize predictive performance but to encode operational decision policies, ensuring interpretability, auditability, and controllability. Learning thresholds from data would improve adaptivity but reduce transparency and auditability, which are primary design objectives in operational energy systems; this trade-off is left for future investigation. In practice, we observed stable qualification behavior across a reasonable range of threshold values (e.g., $\tau_q \in [0.3, 0.5]$, $\tau_\Delta \in [0.05, 0.15]$), suggesting that the system is not critically sensitive to the precise threshold values within this range. A formal sensitivity analysis is left for future work, as it requires multi-site validation data to draw statistically meaningful conclusions.

Operational explainability. Rather than explaining the forecaster's internal mechanisms, the wrapper explains why a prediction can or cannot be used reliably at runtime. Each prediction is accompanied by a machine-readable record including $q(t)$, the detected degradation type(s), the constraints evaluated, and the qualification decision. This enables

Table 2. Forecasting performance: baseline $\hat{\mathbf{y}}$ vs. validated $\hat{\mathbf{y}}^{\text{val}}$ (day-ahead, hourly; p.u.).

	Baseline ($\hat{\mathbf{y}}$)			**Runtime layer** ($\hat{\mathbf{y}}^{\text{val}}$)		
Scenario	MAE	RMSE	R^2	MAE	RMSE	R^2
Nominal	0.14	0.23	0.91	0.14	0.23	0.91
Degraded	0.20	0.30	0.88	0.19	0.29	0.88
Severe	0.25	0.35	0.86	0.24	0.34	0.86

operators to trace whether an abnormal output originates from input issues, constraint violations, or a conservative withholding decision, and supports post-hoc auditing and compliance reporting [12, 19].

5. Experimental Evaluation

5.1. Sites and Data Sources

Experiments are conducted on a real-world PV dataset collected from a PV installation over 18 months (January 2023–June 2024). The forecasting task is day-ahead PV power prediction at hourly resolution (24-step horizon). The dataset integrates heterogeneous evidence streams: PV measurements (site power output in kW, aggregated from 15-min to hourly resolution), weather forecasts (hourly irradiance, temperature, humidity, and wind speed from two independent third-party providers, denoted Provider A and Provider B), and contextual signals (time-of-day and calendar features, and short-term recent patterns derived from available observations).

The dataset is split into training (70%), validation (15%), and test (15%) sets. To evaluate robustness under execution-time input degradation, the test set is evaluated under three scenarios:

- **Nominal:** all inputs available with typical noise levels.
- **Degraded:** moderate degradation (20% missingness, 10% delayed weather updates, occasional cross-source misalignment).
- **Severe:** strong degradation (40% missingness, 25% delayed weather updates, frequent cross-source misalignment).

To ensure reproducibility, degradations are injected independently per input source using i.i.d. Bernoulli masks for missingness, stale-value replacement for weather delays, and timestamp shifting for cross-source misalignment. All injected events are logged and reflected in the runtime evidence indicators.

5.2. Metrics

We use standard forecasting metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and coefficient of determination (R^2).

In addition, we quantify the operational behavior of the runtime layer. For each inference cycle k, the system produces a baseline forecast $\hat{\mathbf{y}}_k$, a validated forecast $\hat{\mathbf{y}}_k^{\text{val}}$, a qualification status $Q_k \in \{\textsc{Valid}, \textsc{Corrected}, \textsc{Withheld}\}$, and trace records.

The *violation rate* r_{viol} is the percentage of baseline forecasts violating at least one constraint before the validation stage. The *correction frequency* r_{corr} is the percentage of cycles where the validation stage modifies the baseline forecast. The *fallback rate* r_{fb} is the percentage of cycles where at least one base predictor activates an internal fallback mechanism. The *withholding rate* r_{with} is the percentage of cycles assigned the Withheld qualification.

5.3. Results

The runtime layer maintains competitive forecasting accuracy across all scenarios, with $R^2 \geq 0.86$ even under severe input degradation (Table 2). This indicates graceful degradation relative to the baseline predictor, while adding explicit validation and qualification capabilities.

Table 3. Operational metrics across input scenarios.

Scenario	Violation Rate	Correction Freq.	Fallback Rate	Withholding Rate
Nominal	3.2%	3.1%	8.4%	0.5%
Degraded	8.7%	8.5%	22.6%	2.8%
Severe	15.6%	15.2%	42.3%	7.1%

Operational metrics further characterize how the runtime layer responds to degraded inputs (Table 3). The violation rate increases from 3.2% in the nominal scenario to 15.6% in the severe scenario, indicating that the baseline predictor produces more physically implausible outputs as input conditions deteriorate [4]. The correction frequency closely follows this trend, showing that the validation stage effectively enforces the imposed domain constraints [4, 17]. The fallback rate rises from 8.4% to 42.3%, making explicit the activation of internal fallback behavior that would otherwise remain hidden from downstream systems. The withholding rate increases from 0.5% to 7.1%, suggesting that the qualification stage appropriately identifies forecasts that should not be consumed under highly degraded runtime conditions [2].

5.4. Trace Analysis

Trace records enable post-hoc analysis of runtime behavior. In the severe scenario, they reveal that fallback activations increase under degraded inputs, making explicit when the forecasting core reverts to a safe estimator. They also indicate that capacity violations are the most frequent (52%), followed by outlier violations (35%) and day/night violations (13%), suggesting that degraded inputs primarily affect amplitude consistency rather than temporal plausibility. In addition, 78% of corrections remain minor (below 10% of forecast value), whereas 22% are more substantial, supporting the use of the Corrected qualification as a meaningful operational signal. The withholding rate is strongly correlated with missing data rate ($r = 0.87$) and weather forecast staleness ($r = 0.79$), providing evidence that the input quality assessment mechanism captures relevant degradation patterns [4]. Specifically, capacity and outlier violations (87% of all violations) are predominantly associated with degraded and severe input conditions, where missing or stale weather inputs cause the inference core to produce amplitude-inconsistent forecasts, while day/night violations remain rare and scenario-independent, consistent with the misalignment penalty capturing spatial discrepancies between third-party weather provider coverage areas and the actual installation location.

Importantly, the qualification status correlates with forecast error: forecasts labeled Withheld correspond to significantly higher error regimes (MAE > 2× nominal), confirming that the rule-based decision function $\mathcal{F}$ (Eq. 1) identifies unreliable predictions rather than withholding arbitrarily. Similarly, Corrected forecasts exhibit intermediate error levels between Valid and Withheld; under severe degradation, the per-status MAE confirms this ordering quantitatively: $\mathrm{MAE}_{\mathrm{valid}} = 0.282$ p.u., $\mathrm{MAE}_{\mathrm{corrected}} = 0.273$ p.u., and $\mathrm{MAE}_{\mathrm{withheld}} = 0.337$ p.u., validating the three-level qualification as an operationally meaningful and quantitatively grounded decision-oriented output. These trace-based observations illustrate the value of auditable runtime records for understanding, debugging, and governing forecasting behavior in operational settings [12].

6. Discussion

6.1. Implications for Operational Deployment

The constraint-based decision layer addresses several key challenges in deploying PV forecasting systems in time-critical operational contexts. The decision function $\mathcal{F}$ (Eq. 1) produces decision-oriented outputs that enable downstream systems to make informed choices about forecast consumption, reducing the risk of acting on unreliable predictions [2]. Operational metrics such as violation rate, fallback rate, and withholding rate provide transparent evidence of how the rule-based reasoning responds to input degradation, supporting human oversight [2]. Trace records support post-hoc analysis, debugging, and compliance verification, which are increasingly important in regulated energy markets [12]. Because the decision layer wraps the baseline predictor without modifying its internal architecture, operational systems can adopt this reasoning capability incrementally without redesigning their forecasting models [10]. While simpler strategies such as silent clipping or systematic discarding of forecasts under missing data are already embedded in the baseline architecture (Section 3.3), the runtime layer makes these interventions explicit, qualified, and traceable, a distinction that is operationally significant when downstream systems must decide whether to act on, adjust, or abstain from consuming a forecast.

6.2. Limitations

The proposed approach has several limitations. Qualification rules depend on configurable thresholds (τ_q, τ_Δ, and the input quality weights), whose values are derived from validation-set behavior and PV engineering conventions rather than from a formal optimization procedure. This is a deliberate design choice: these thresholds encode operational policies rather than model parameters, and their transparency is a feature rather than a limitation. Nevertheless, a systematic sensitivity analysis would strengthen confidence in the reported operational metrics and does not affect the qualitative behavior of the system. Trace serialization and storage may accumulate overhead in high-frequency forecasting scenarios; however, for the hourly day-ahead setting evaluated here, the runtime layer adds negligible latency, as trace records consist of lightweight rule evaluations with no iterative optimization. In deployments requiring higher temporal resolution, selective logging of Corrected and Withheld events can reduce storage overhead while preserving auditability.The runtime layer does not learn from trace data or update qualification rules over time; future work could investigate adaptive threshold calibration inspired by self-adaptive windowing mechanisms [18], while preserving the auditability guarantees of the current design. The proposed approach is complementary to uncertainty estimation methods (e.g., conformal prediction, probabilistic forecasting), which quantify prediction dispersion but do not provide explicit decision policies at inference time; combining both paradigms is a promising direction. Finally, the current evaluation is conducted on a single PV installation (18 months, hourly day-ahead forecasting). While the constraint-based architecture is designed to be site-agnostic, relying on normalized capacity units (p.u.) and domain-invariant physical constraints, empirical confirmation of this property across multiple installations with varying capacities, climates, and degradation profiles is essential to confirm threshold transferability and operational robustness [1, 10]. Future work will evaluate the framework on geographically diverse PV fleets to quantify cross-site generalization.

6.3. Generalization Potential

The runtime layer is designed to be predictor-agnostic and could be adapted to wind power forecasting [4, 7], load forecasting [2, 3], and industrial process monitoring. Adaptation would require domain-specific constraint definitions and input quality mechanisms, but the core architecture based on validation, qualification, and traceability remains applicable.

7. Conclusion and Future Work

This paper addresses a critical gap in operational PV forecasting: the absence of explicit, reasoned qualification signals when predictions are deployed in time-critical settings. We introduced a constraint-based decision layer, formalized as a decision function $\mathcal{F}$ that maps forecasts and runtime evidence to decision-oriented outputs: a validated forecast, a qualification status (Valid/Corrected/Withheld), and an auditable diagnostic trace for every inference cycle, without requiring model redesign or retraining.

Experimental evaluation on real-world PV data showed that the decision layer maintains competitive forecasting accuracy ($R^2 \geq 0.86$ under severe input degradation) while detecting and correcting 15.6% of constraint violations, exposing 42.3% fallback activations, and withholding 7.1% of forecasts judged unreliable. Crucially, the qualification status correlates with forecast error, confirming that the rule-based reasoning identifies genuinely unreliable predictions.

Future work will focus on (i) validating the decision layer across multiple sites and operating regimes, (ii) developing adaptive threshold calibration mechanisms inspired by self-adaptive windowing approaches [18], and (iii) evaluating the actionability of runtime traces for operators under realistic incident scenarios.

By bridging robust forecasting models and operational decision-making, this work contributes to the broader goal of deploying reliable, interpretable, and auditable knowledge-based AI systems in time-critical energy applications.

Acknowledgements

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] K. Barhmi, C. Heynen, S. Golroodbari, W. van Sark, A review of solar forecasting techniques and the role of artificial intelligence, Solar 4 (1) (2024) 99–135.

[2] X. Huang, T. Zhao, B. Huang, Z. Zhang, Y. Meng, Advancing energy system optimization via data-centric task-oriented forecasting: An application in PV-battery operation, Applied Energy 378 (2024) 124753.

[3] K. Kravari, M. Roussi, K. Ladomenou, A. Thysiadou, M. Chalaris, Intelligent agent orchestration for climate-resilient energy systems: Autonomous management of renewable integration and distributed resources, E3S Web of Conferences 638 (2025) 02009.

[4] L. Yang, Z. Huang, X. Mo, T. Luo, Enhanced GAIN-based missing data imputation for a wind energy farm SCADA system, Electronics 14 (8) (2025) 1590.

[5] U. Hijjawi, S. Lakshminarayana, T. Xu, G.P.M. Fierro, M. Rahman, A review of automated solar photovoltaic defect detection systems: Approaches, challenges, and future orientations, Solar Energy 266 (2023) 112186.

[6] J. Gaboitaolelwe, A.M. Zungeru, A. Yahya, C.K. Lebekwe, D.N. Vinod, A.O. Salau, Machine learning based solar photovoltaic power forecasting: A review and comparison, IEEE Access 11 (2023) 40820.

[7] W. Castillo-Rojas, J. Salinas, Forecasting models applied in solar photovoltaic and wind energy: A systematic mapping study, IEEE Access (2024) 1–1.

[8] N. Rahimi, S. Park, W. Choi, B. Oh, S. Kim, Y.-h. Cho, S. Ahn, C. Chong, D. Kim, C. Jin, D. Lee, A comprehensive review on ensemble solar power forecasting algorithms, J. Electr. Eng. Technol. (2023).

[9] L. Sakli, 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, 2025.

[10] S. Meisenbacher, B. Heidrich, T. Martin, R. Mikut, 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.

[11] V. Zaccaria, C. Masiero, D. Dandolo, G.A. Susto, Enabling efficient and flexible interpretability of data-driven anomaly detection in industrial processes with AcME-AD, in: Proc. 10 th Int. Conf. on Control, Decision and Information Technologies (CoDIT), 2024, p. 1375.

[12] E. Gabaldón, R. Barricarte, J. Garrido, An end-to-end data and machine learning pipeline for energy forecasting: A systematic approach integrating MLOps and domain expertise, Information 16 (9) (2025) 805.

[13] M. Gupta, A. Arya, U. Varshney, J. Mittal, A. Tomar, A review of PV power forecasting using machine learning techniques, Prog. Eng. Sci. 2 (1) (2025) 100058.

[14] W.F. Holmgren, C.W. Hansen, M.A. Mikofski, pvlib python: A python package for modeling solar energy systems, J. Open Source Softw. 3 (29) (2018) 884.

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

[16] M. Tortora, F. Conte, G. Natrella, P. Soda, MATNet: Multilevel fusion and self-attention transformer-based model for multivariate multi-step day-ahead PV generation forecasting, arXiv preprint arXiv:2306.10356 (2023).

[17] A. Sundararajan, M. Olama, M. Ferrari, B. Ollis, G. Liu, A data quality-aware framework to reliably forecast photovoltaic generation and consumer load for an improved resilience of microgrids, in: Proc. IEEE 13th Int. Symp. PEDG, 2022, pp. 1–6.

[18] I. Khamassi, M. Sayed-Mouchaweh, M. Hammami, K. Ghédira, Self-adaptive windowing approach for handling complex concept drift, Cognitive Computation 7 (6) (2015) 772–790.

[19] E. Breck, M. Zinkevich, N. Polyzotis, S. Whang, S. Roy, Data validation for machine learning, in: Proc. SysML Conference, 2019.