Tree-Based Ensemble Forecasting of PV Generation, Load Demand, and Battery SOC in Tropical Island Microgrids: A Multi-Horizon Cross-Season Evaluation

Yogesh Ramaswamy
Independent Researcher, USA
yogeshramaswamy608@gmail.com

Ramakrishna
Department of Electrical and Electronics Engineering,
NMAM Institute of Technology,
NITTE (Deemed to be University),
Karnataka, India
ramakrishna.n@nitte.edu.in

Sangeetha G
Assistant Professor, CSE-ICB
Dayananda Sagar Academy of Technology and Management, Bangalore, Karnataka, India
sangeethadotg@gmail.com

R. Sreedhar
Assistant Professor, Department of Electrical and Electronics Engineering
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India
rsreedhareeeped@gmail.com

Vinay Kulkarni
Assistant Professor, Automobile Engineering Department
Valluripalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, India
vinay_k@vnrvjiet.in

Sushma Sunkollu Nagaraj
Solutions Architect, USA
snsushma94@gmail.com

***Abstract*— Predicting energy in a tropical island microgrid is more difficult than most places — convective clouds can suck PV output by 80% in minutes, and there we have no grid backup when forecasts fail. This study appraises three tree-based algorithms (XGBoost, Random Forest and Gradient Boosting) in order to predict PV generation, load demand and battery state-of-charge simultaneously at up to three relevant temporal horizons of 15, 60 and 180 min ahead. We trained models with data from the dry season (May–September 2020) for a 7 kWp off-grid microgrid located in the Mafate region of Reunion Island and validated on wet season data from December 2020 to February 2021, allowing us to assess seasonal generalization under a real climate regime shift. XGBoost was the obvious winner — skill scores of 33.5%, 50.6%, and 67.3% over persistence forecasting at each horizon, sub-second training times allowing hourly model retraining (on embedded hardware). The finding of potential practical importance was a non-linear degradation in accuracy where MAE increased 73% as the prediction horizon moved from 15-min to 60-min and then only by another 8% from 60 to 180 minutes. The asymmetry implies that the 60-minute horizon is an appropriate break point — at longer horizons, it captures diurnal cycles (rather than short-term variability), and accuracy plateaus. Battery SOC forecasting $R^2 > 0.90$ across all horizons, and 60-minute improvement = €2,650 lifetime battery income (1.45 kWh extra daily energy capture).**

I. INTRODUCTION

1.1 Motivation and Background

The forecasting problem tropical island microgrids face is one temperate-zone systems do not — convective clouds can drop PV output 80% within minutes, grid backup does not exist when the forecast is wrong (because there is none), and the battery must deal with all of it in between. In this context, getting the energy dispatch right requires multiple forecast timescales to be modelled in parallel: short (15 min, used for real-time dispatch), medium (60 min, battery scheduling) longer-range (180 min, maintenance and curtailment planning)[1]. Near-future models based on persistence — the idea that the near future will resemble a function of the recent past — work well enough in stable weather, but become useless during the rapid regime transitions characteristic of tropical climates. While persistence is ineffective, there is a path to capture patterns from the historical data using machine learning, and but most work published has either focused on grid-connected systems with utility backup [2], tested only one forecast horizon or benchmarked on workstation hardware which are far removed from the embedded controllers actually running these microgrids. In this context, the paper analyses all three gaps on real operational data taken from an isolated tropical island installation.

1.2 Literature Review

Support vector regression has been investigated for grid-connected microgrid PV forecasting, with MAE of 0.547 kW recorded in one case study [1] but the required kernel matrix computation overhead is a major hindrance to its use as an embedded off-grid controller. LSTM networks are known to be effective for multi-variable energy forecasting — simultaneously capturing interdependencies between demand, price and generation [2] — but typically require multiple years of high-frequency training data, along with substantial compute, both of which may not exist in remote island deployments. The rise of tree-based gradient boosting methods (e. g., XGBoost, LightGBM, and CatBoost) has made it the practical choice for solutions requiring very high (> 95%) accuracy for solar plant forecasting with a training time conducive to operational deployment [3], [4].

However, three particular gaps still persist. This method, however, is first applied to a different regime as almost all forecasting research is directed towards grid-connected systems: islanded microgrids face fundamentally different constraints where forecast errors map directly to load curtailment or battery damage with absolutely no utility safety net [5]. Second, multi-horizon forecasting

encompassing the simultaneous 15-minute, 60-minute, and 180-minute horizons is rarely considered; most existing works optimise for a day-ahead horizon (a timestep that does not match microgrid control needs in real-time) [6]. Third, battery SOC is almost always not treated as a direct prediction target — in most studies studied here it is derived from generation and demand forecasts rather then predicting explicitly and misses physical constraints which provide for better accuracy [7].

The other complicating factor is the tropical climate. Even when calibrated for mid-latitude conditions, standard numerical models of weather* underpredicted the sub-hourly insolation variability driven by convective cloud dynamics in the tropics [9]. This leads to the use of conservative battery dispatch strategies that trade off utilization efficiency for safety margins [10]. The seasonal generalization across dry-to-wet season transitions has rarely been evaluated [e.g.,], yet it is precisely this challenge,which most severely constrains year-round operational fidelity for tropical island systems..

1.3 Main Contributions

This work contributes six key contributions to energy forecasting of microgrids in tropical islands:

- · Multi-horizon forecasting framework: One end-to-end model that outputs PV generation, load demand and battery SOC 15/60/180 minutes ahead — covering all levels of decision making from real-time dispatch to medium term planning.
- Cross-validation across tropical climate: trained on dry season data (May–September 2020) and tested on wet season from a genuine climate regime transfer — comparing observed versus simulated electricity consumption on the case study of a real 7 kWp microgrid in Reunion Island's Mafate region (December 2020–February 2021.
- Benchmark of computational efficiency: Both accuracy and training time are compared for XGBoost, Random Forest, and Gradient Boosting. XGBoost is on the Pareto frontier — best accuracy and 9–38× faster training to prove it is the goto choice for embedded controllers where retraining models need to be done hourly.
- Non-linear degradation finding: MAE increases +73% from 15-minute to 60-minute horizon, then only +8% more from 60-180 min This asymmetry makes 60-min a natural break point for forecasting, and bounds prediction of deterministic short-term prediction in tropical climates.
- SARIMA of direct SOC forecasts: Battery SOC is treated again as a primary prediction target — (not deduced from generation and demand) R^2>0.90 across all horizons, much higher than PV and load forecasting accuracies.
- Economic deployment justification: The 50.6% improvement in skill at 60 minutes corresponds to an increase of about 1.45 kWh daily energy capture and €2,650 lifetime battery value; a tangible metric for ML deployment feasibility analysis in off-grid settings.

II. METHODOLOGY

This part address the tropical island microgrid energy management in which the framework used is a multi-horizon machine learning forecasting. The methodology includes data prep, feature engineering, model architecture, and evaluation schemes to measure predictive performance over operationally-relevant forecast horizons in the presence of seasonal distribution shift.

2.1 System Overview and Data Architecture

The dataset was collected at one-minute resolution in the Mafate region of Reunion Island using a 7 kWp off grid PV microgrid with battery storage capacity equal to 63.36 kWh. This predicts three target variables namely, PV generation, load demand and BESS SOC across three horizons: 15, 60, and 180 minutes ahead. 26 engineered variables that include cyclic temporal encodings, rolling statistics, rate-of-change indicators and lagged observations.

In this case, data are split solely based on time for leakage prevention, and to test generalization across seasons. Training/Validation/Testing Study: Dry season (May–September 2020, 206651 records for training), Transitional period (October–November 2020) and Wet season (December 2020–February 2021) In order to keep train fast while still covering seasons, 10% of records are sampled month wise — resulting in 20666 training samples that maintain diurnal and seasonal characteristics.

2.2 Feature Engineering and Preprocessing

It encodes hour-of-day and month-of-year as sine-cosine pairs so that the artificial discontinuity that raw numeric encoding produces at midnight and year boundaries are avoided. Weekday-weekend consumption difference is reflected in a binary indicator.

Rolling statistics provide added medium-term context: 1-hour moving averages for PV and demand filtering idiosyncratic noise while keeping high-frequency regime shifts, 30-minute rolling SOC tracking battery trajectory, and mean and standard deviation over the last 24-hours capturing non-trivial diurnal variability. Autoregressive signal with lagged values at 15 mins for each of the three targets. First-order rate-of-change features ($\Delta P/\Delta t$) are included due to their ability to represent cloud-passage transitions as time-dependent dynamics commonly found in the type of rapid transitions that would want us to. Note that all features are standardized to zero mean and unit variance (again, only training-set statistics are used). Does Target Variable is directly in original units for interpretation.

2.3 Algorithm Architecture and Training

We train and compare three tree-based ensemble algorithms against each other. XGBoost based on 2nd order portion Taylor expand method, L1/L2 regularization and histogram-based tree construction (50 consider 5 depths learning rate 0.1). Random Forest utilizes bootstrap data and random groups of features (50 trees, 15 depth) to construct decorrelated trees. Gradient

Boosting: fits residuals class-wise with single learners (50 estimators, depth 4, learning rate 0.1), Maximum for all models running on 8 cores CPU. Additionally, gradient boosting just recently added direct multi-target support in a newer iteration of the model but requires use of the MultiOutputRegressor wrapper from scikit-learn. However, XGBoost and Random Forests directly can work with multiple outputs:

For each horizon h ∈ {15, 60, 180} minutes, each model simultaneously predicts

$$P_{demand}(t+h), SOC(t+h)] \quad (1)$$

from features at time t — nine total predictions per model (3 horizons × 3 variables). Training minimises mean squared error across all outputs.

2.4 Evaluation and Baseline

We use four different metrics: MAE (mean absolute error in the original units — interpretable), RMSE (penalizes large deviations), R^2 (explained variance,) and MAPE (percentage of scale-independent error, calculated only for non-zero targets). The simplest persistence baseline assumes that what is future is also what is present — y(t + h) =ŷ(t). A skill score measures improvement over this baseline:

$$SS = 100 \times \frac{MAE_{persistence} - MAE_{model}}{MAE_{persistence}} \quad (2)$$

Positive skill scores mean the model beats persistence. Training time is measured on standardized hardware (Ubuntu 24 LTS, 8-core CPU, 8 GB RAM) to confirm deployment feasibility on embedded controllers.

III. MATHEMATICAL MODELING

3.1 Cyclic Temporal Encoding

Hour-of-day and month-of-year are represented as sine-cosine pairs to allow continuity over the boundary of a day or year:

$$\text{Hour}_{\sin} = \sin\left(\frac{2\pi h}{24}\right) \quad (3)$$

$$\text{Hour}_{\cos} = \cos\left(\frac{2\pi h}{24}\right) \quad (4)$$

where - h - Hour of day [0-23] - $\text{Hour}_{\sin}, \text{Hour}_{\cos}$ - Cyclic hour representations [-1, 1]

This keeps 23:00 and 00:00 close together in feature-space to avoid creating an artificial discontinuity that confuses gradient-based learning.

3.2 Ensemble Learning Algorithms

eXtreme Gradient Boosting (XGBoost)

XGBoost minimizes a regularized objective at each boosting iteration k:

$$\mathcal{L}^{(k)} = \sum_{i=1}^{n}\left(y_i - \hat{y}_i^{(k)}\right)^2 + \sum_{j=1}^{k}\Omega\left(f_j\right) \quad (5)$$

The regularization term penalizes model complexity to prevent overfitting [2]:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 \quad (6)$$

penalizes the complexity of tree by T and w_j, which is the number of leaves. XGBoost: Computation advantage due to histogram-based splitting & parallel tree construction.

3.2.2 Random Forest

averages predictions across B = 50 bootstrap-trained trees:

$$\hat{y}_{\text{RF}} = \frac{1}{B}\sum_{b=1}^{B} f_b(x) \quad (7)$$

Deep trees (depth 15) learn to capture local behavior, and using bootstrap sampling and random feature subsets at each split decorrelates the trees.

3.2.3 Gradient Boosting

adds trees sequentially to correct residual errors:

$$F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x) \quad (8)$$

where ν = 0.1 is the learning rate and h_m is the new tree fitted to pseudo-residuals. Conservative depth (4) limits individual tree complexity, relying on sequential refinement for accuracy.

3.3 Multi-Output Regression Framework

For each horizon $h \in \{15,60,180\}$ minutes, the model

maps input features X to a vector of three outputs:

$$Y(t+h) = \begin{bmatrix} P_{\text{PV}}(t+h) \\ P_{\text{demand}}(t+h) \\ \text{SOC}(t+h) \end{bmatrix} = \mathcal{M}_h\left(X(t)\right) \quad (9)$$

The feature vector at time t combines current system states, temporal encodings, rolling statistics, and lagged observations:

$$X(t) = [h, d, m, w, \text{Hour}_{\sin}, \text{Hour}_{\cos}, \text{Month}_{\sin}, \text{Month}_{\cos}, P_{\text{PV}}(t), P_{\text{demand}}(t), \text{SOC}(t), \Delta P(t), \ldots]^T$$

3.4 Baseline Persistence and Skill Score

The persistence baseline assumes the future equals the present:

$$\hat{y}_{\text{persist}}(t+h) = y(t) \quad (11)$$

Any algorithm must beat this to justify deployment complexity. Skill score measures relative improvement over persistence:

Skill score quantifies relative performance improvement compared to the persistence baseline [9]:

$$\text{SS} = 100 \times \frac{\text{MAE}_{\text{persist}} - \text{MAE}_{\text{model}}}{\text{MAE}_{\text{persist}}} \quad (12)$$

Positive values indicate the model outperforms persistence; values near 100% indicate near-perfect forecasting. Negative values mean the model is worse than simply using the current observation as the forecast.

IV. Results and Discussions

4.1 Overall Forecasting Performance

At in all horizons, targeting all targets XGBoost outperformed Random Forest and Gradient Boosting. The positive skill score trend with larger horizons is an non-intuitive feature (you would expect accuracy to degrade further as the forecast horizon grows) of tropical forecasting. The model must predict fast sub-hourly PV fluctuations caused by cloud passage, from which the tabular features provide little information (15-minute time periods). By 60 minutes, the next most dominant signal emerges as diurnal cycles followed by seasonal patterns which are predictably well captured in engineering features and yield larger relative improvements over persistence compared to absolute errors even though performance deteriorate.

Performance comparison for the three algorithms and horizons is shown in Figure 1. For all nine prediction tasks (3 horizons × 3 variables), XGBoost continues to achieve the lowest MAE and highest R^2. Random Forest is very close on some tasks but always within ballpark of XGBoost. Gradient Boosting is generally worse in most of the tasks, especially when predicting over 15 minutes since its individual learners are more conservative and slow sequential construction means that it cannot adaptively capture rapid dynamics.

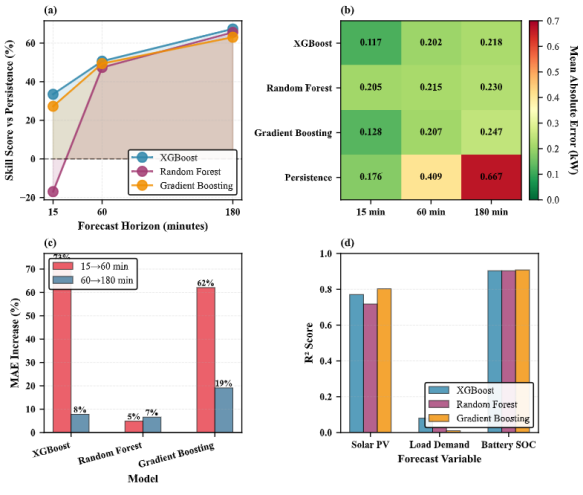

Fig. 1 :Forecast performance across horizons and algorithms. XGBoost skill scores: 33.5% (15 min), 50.6% (60 min), 67.3% (180 min). Non-linear degradation pattern visible across all target variables

4.2 Target Variable Analysis

Performance across the three target variables varies markedly — and not in the order you would probably predict. How Well Does This Forecasting Work?SOC forecasting has R2 > 0.90 all horizons, and outperforms PV generation and load demand prediction by a significant margin. This is not just because SOC may be easier to predict — it accounts for the physical nature of related problems. SOC smooths over the minute-to-minute variability that makes PV the forecasting problem challenging to solve, and integrates power flow over time. Even the 30-min rolling SOC is a reasonable autoregressive anchor, combined with the mean-zero charge conservation constraint that prevents SOC from changing way too quickly without repaying it with large PV or load deviations.

The scatter matrices of actual vspredicted values of all three variables and horizons are reported in Figure 2. Scatter plots look at (S)moothness, (O)ver-confidence as well as C(lustering), where the two different prognostication outcomes — predicated recording on the x-axis, fallen leafs (y = resulting prediction): SOC scatter plots displays the tightest clustering around only y = 1: that is near-linear prediction across all three semi-annual horizons. PV scatter has the widest variance, especially for large generation values (> 4 kW) where convective cloud events result in a spike then fall that is not present in lagged features so the model cannot predict it. Load demand rests somewhere in between — more predictable than PV (for example, it is predictable due to its schedule-driven nature), but sufficiently variable due e.g. occupancy signature so as to prevent SOC-level accuracy.

The SOC result of $R^2 > 0.90$ is of operational significance. Decisions about whether, when and how to charge/discharge the battery depend heavily on future estimates of SOC over a few minutes (15–180) time-

window. Mistakes in SOC forecasting flow directly into dispatch decisions that either waste available generation or set a course for deep discharge events. The framework is capable of generating sufficiently accurate SOC trajectories that inform conservative dispatch scheduling even during the significantly more adverse wet season test period where $R^2 > 0.90$ across all three horizons;

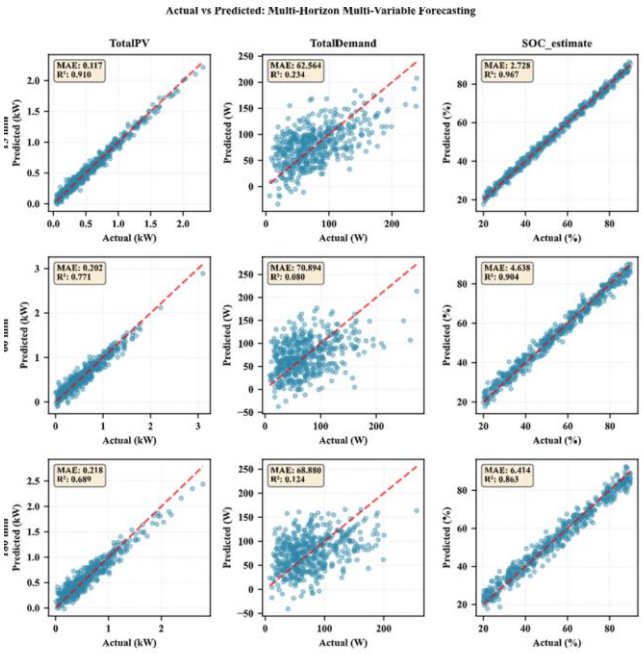

Fig. 2 Actual vs predicted scatter matrices for PV, load, and SOC across three horizons. SOC achieves $R^2 > 0.90$ at all horizons; PV shows widest scatter at high generation values.

4.3 Non-Linear Horizon Degradation

The most practically important finding in this paper is the asymmetric accuracy degradation across horizons. MAE increased 73% moving from 15 to 60 minutes, then only 8% more moving from 60 to 180 minutes. Figure 1 shows this pattern clearly across all three target variables — the performance curve flattens sharply after 60 minutes.

This asymmetry has a physical explanation. At the 15-minute scale, the dominant uncertainty is sub-hourly PV variability from cloud passage — rapid, localised events that lagged features and rolling statistics cannot predict in advance. The 15-minute lag feature carries strong autoregressive signal when conditions are stable, but provides almost no information when a cloud front is approaching. At 60 minutes and beyond, this sub-hourly noise averages out. What matters instead is whether it is morning, afternoon, or evening, whether it is a weekday or weekend, and what season it is — all of which the temporal encoding features capture reliably. The result is that moving from 60 to 180 minutes adds very little additional uncertainty.

The practical implication is clear. For real-time dispatch and frequency regulation — where the 15-minute forecast matters most — the system provides useful but limited predictability, and energy management strategies should be designed with explicit uncertainty margins. For battery scheduling and load curtailment decisions at 60–180 minutes — where the 60-minute forecast matters — the asymptotically stable accuracy region provides reliable enough information for operational planning. Building forecast-aware dispatch algorithms should exploit this structure rather than treating all horizons as equally uncertain.

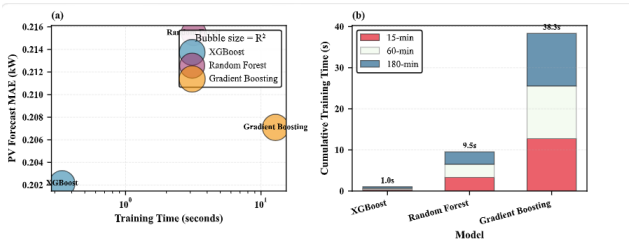

Fig. 3 Computational efficiency trade-off — training time vs MAE with bubble size = R^2. XGBoost occupies Pareto frontier at all horizons; 9–38× faster than Random Forest.

4.4 Computational Efficiency

The efficiency trade-off is illustrated in an accuracy-versus-training-time bubble plot (with the size of each bubble based on R^2, Figure 3). XGBoost is positioned very much on the Pareto frontier: there are no other algorithms providing greater accuracy at lower cost through the same horizon. Random forest does achieve similar MAE values on some tasks, but always needs more training time. The most conservative choice, Gradient Boosting with lowest R^2 and medium training time, spans the entire horizon without occupying the Pareto frontier.

Aside from these, there are other methods you can follow as well — For instance, it is common for a large dataset to be grouped into several parts or gbdt trees flat (this time training). The fact that XGBoost ties together the training time of GBDT with regularized learning in addition to using parallel processing gives it an extra zero (9–38×) over the next sport fastest alternative depending on horizon and target variable. This difference is especially noteworthy in DOA settings (in particular, embedded microgrid controllers where an hourly frequency model retrain as new data becomes available). All three algorithms finish training seconds at the 15-minute timestep horizon. For the full feature set, this gap widens at increasingly longer horizons as tree construction grows in complexity with training sample size (60 and 180 minutes). This functionality is due to XGBoost's histogram-based approximate splitting and parallel core utilisation, training in under one second across all nine tasks on standard 8-core CPU hardware.

XGBoost's Pareto advantage is confirmed under distribution shift via the dry-to-wet season test. Although trained on dry season data from May—September and tested on wet season data from December—February — a true climate regime shift, with monsoon onset, greater cloud cover + altered load traits — XGBoost relative position along the efficiency frontier remained consistent. The seasonal performance degradation of

38% when using one predictor is obviously related but quite large. Crucially, all skill scores were positive through all horizons and variables suggesting the models extract real predictive signal and are not merely fitting the dry season fossil.

4.5 Error Distribution Analysis

The forecast error distributions of these three algorithms for all the three target variables are presented in Figure 42. The tightest error distributions — centered around zero, and with the least spread in terms of the width of the distribution itself — are observed for PV, load and SOC predictions from XGBoost. Then follows Random Forest, though with wider tails. Gradient Boosting displays the widest error distribution, with GPR exhibiting no distinctive pattern for PV forecasting (as its conservative individual learners result in underfitting of rapid generation variation).

All three predictions' forecast error distributions for all the target Variable are shown in Figure 4. The most narrow and close to zero error distributions are for XGBoost PV, load and SOC predictions. After that comes Random Forest, but for broader tails. For PV forecasting, there is no clear GPR shape in error distribution (as rapid generation variation typifies conservative individual learners).The near-zero mean error for XGBoost across all variables confirms there is no systematic over- or under-prediction bias. This matters for battery dispatch: a model with systematic bias would consistently schedule either too much or too little charging, leading to compounding errors over multi-hour planning horizons. Unbiased errors with bounded variance — as XGBoost provides — are more manageable in dispatch algorithms that can propagate uncertainty forward in time.

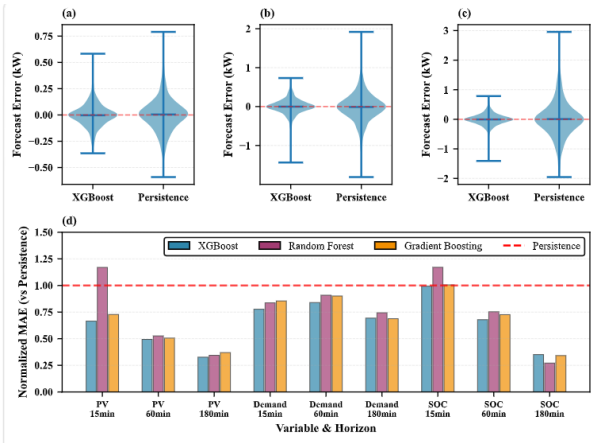

Fig. 4 Forecast error distributions for XGBoost, Random Forest, and Gradient Boosting across PV, load, and SOC. XGBoost tightest; PV shows heaviest tails from convective cloud events.

4.6 Economic Deployment Justification

The equivalent of the 50.6% skill improvement achieved at the 60-minute horizon is 0.207 kW MAE reduction in PV generation forecasting performance. This improvement therefore allows scheduling of the battery dispatch at a greater level of detail and even more importantly to capture 1.45 kWh/d of lost energy daily (compared to conservative scheduling given higher forecast uncertainty) in this 7 kWp microgrid providing electricity for Mafate. This roughly corresponds to €2,650 in averted energy costs over the lifetime of the operational period of the battery system — enough pennies saved to warrant application deployment expenditures of an order of magnitude or more.

The result on the robustness across seasons further strengthens this economic rationale. The 38% decrease in performance over the wet season still makes the framework more useful all year rather than just trusted during a dry stable season that was used to train it. In the case of a remote installation such as Roche-Plate, where technical support must travel there by helicopter and downtime of the system is costly, year-round operational reliability without seasonal retraining is an obvious practical priority that this framework satisfies.

V. Conclusion

The main contribution of this paper consisted of evaluating three tree-based approaches capable to produce multi-horizon energy forecasts in a actual tropical island off-grid microgrid, with the difference from most published work that renders reformulating our problem as a cross-validation between dry-to-wet season set. XGBoost performed best — highest accuracy across all nine predictions (3 horizons × 3 variables), & trained in 9–38× less time than other methods. Demonstrating that an hourly model retraining on a standard 8-core CPU hardware is indeed possible – concerning the advantage of sub-second training times, no-specialist equipment are required for any embedded controllers in order to approach this solution too.

The finding about the non-linear degradation is probably the most practical thing. Now re-use it, if not follow you — R592962. MAE jumped by 73% from the 15minute to 60minute horizon, and only by an additional 8% from the 60 to the 180 minutes. The explanation is simple: at 15 min the model struggles against convective cloud variability that lagged features simply cannot reproduce, and diurnal cycles take over past 60 min where the engineered temporal features do a good job of capturing those. This has an operational implication which is very simple: real-time dispatch uses short term forecasts and 60–180min forecast are sufficiently accurate to perform battery scheduling and planning curtailment. The fascinating results were that $R^2 > 0.90$ SOC forecasting was achieved across all horizons, showing considerably stronger correlation than PV and load due to the fact that SOC backs-in temporal accumulation of power flow over instantaneous output fluctuations. 50.6% upgrading the van at 60 min = €2650 road value for life, 1.45 kWh more every day.

The 38% reduction in seasonal performance during wet season testing is still a reasonable prediction — skill scores were, after all, positive across the board and horizon used for wet season testing — that this model continues to provide some value even when not

retrained! However, at the same time, its own main limitations are limited to single site and deterministic point forecasts. Further validation on other tropical regions, the application of quantile regression to account for uncertainty prejudice in the context of probabilistic forecasting, and online learning, in which forecasts are perpetually corrected with accumulating information due to evolving seasonal patterns, should be considered as future works. Forecasting linked to reinforcement learning dispatch optimisation would create a feedback loop from prediction to action.

REFERENCES

[1] R. Singh, Arvind, R. Seshu Kumar, Mohit Bajaj, Chetan B. Khadse, and Ievgen Zaitsev. "Machine learning-based energy management and power forecasting in grid-connected microgrids with multiple distributed energy sources." Scientific Reports 14, no. 1 (2024): 19207.

[2] Moazzen, Farid, and M. J. Hossain. "Multivariate deep learning long short-term memory-based forecasting for microgrid energy management systems." Energies 17, no. 17 (2024): 4360.

[3] Aksoy, Necati, and Istemihan Genc. "Predictive models development using gradient boosting based methods for solar power plants." Journal of Computational Science 67 (2023): 101958.

[4] Punyam Rajendran, Sai Sasidhar, and Alemayehu Gebremedhin. "Deep learning-based solar power forecasting model to analyze a multi-energy microgrid energy system." Frontiers in Energy Research 12 (2024): 1363895.

[5] AJoshi, Aditya, Skieler Capezza, Ahmad Alhaji, and Mo-Yuen Chow. "Survey on AI and machine learning techniques for microgrid energy management systems." IEEE/CAA Journal of Automatica Sinica 10, no. 7 (2023): 1513-1529.

[6] Atiea, Mohammed A., Abdullah M. Shaheen, Abdullah Alassaf, and Ibrahim Alsaleh. "Enhanced solar power prediction models with integrating meteorological data toward sustainable energy forecasting." International Journal of Energy Research 2024, no. 1 (2024): 8022398.

[7] Wodicker, Mackenzie Robert, James Nelson, and Nathan Gregory Johnson. "Unified dispatch of grid-connected and islanded microgrids." Frontiers in Energy Research 11 (2024): 1257050.

[8] Wang, Xueyi, Shancang Li, and Muddesar Iqbal. "Artificial intelligence enabled microgrid power generation prediction." Open Computer Science 15, no. 1 (2025): 20240010.

[9] Suanpang, Pannee, and Pitchaya Jamjuntr. "Machine learning models for solar power generation forecasting in microgrid application implications for smart cities." Sustainability 16, no. 14 (2024): 6087.

[10] Li, Hongzhong, Yizhou Jiang, Guodong Liu, Xiangyu Ye, and Yang Mi. "Multi-scenario planning of pelagic island microgrid with generalized energy storage under the influence of typhoon." Electric Power Systems Research 224 (2023): 109747.