IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 September 2025, accepted 29 October 2025,
date of publication 10 November 2025, date of current version 17 November 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3631152

RESEARCH ARTICLE

Fast and Accurate Solar Power Generation Forecasting Using Advanced Deep Learning: A Novel Neural Basis Expansion Analysis Framework

NURUL JANNAH[1], (Student Member, IEEE), TEDDY SURYA GUNAWAN[1], (Senior Member, IEEE), SITI HAJAR YUSOFF[1], (Member, IEEE), MOHD SHAHRIN ABU HANIFAH[1], (Member, IEEE), SITI NADIAH MOHD SAPIHIE[2], AND HAMZAH ADLAN[2]

[1]Electrical and Computer Engineering Department, Faculty of Engineering, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia
[2]PETRONAS Research Sdn Bhd, Bandar Baru Bangi 43000, Malaysia

Corresponding authors: Siti Hajar Yusoff (sitiyusoff@iium.edu.my) and Teddy Surya Gunawan (tsgunawan@iium.edu.my)

This work was supported by the Petroliam Nasional Berhad (PETRONAS) Research Sdn Bhd under Grant SPP22-124-0124.

ABSTRACT Accurate short-term forecasting of photovoltaic (PV) power generation is essential for maintaining the stability and efficiency of modern power grids. However, conventional statistical and regression-based methods often fail to capture the nonlinear and stochastic patterns of solar generation data, leading to unreliable forecasts and operational inefficiencies. This study addresses these challenges by systematically benchmarking eight deep learning architectures for PV power forecasting, including Long Short-Term Memory (LSTM), Temporal Convolutional Network (TCN), Transformer, Temporal Fusion Transformer (TFT), TiDE, N-HiTS, DLinear, and Neural Basis Expansion Analysis for Time Series (N-BEATS). Real-world 15-minute interval PV data from 2018 to 2019 were used to evaluate the models in terms of forecasting accuracy, parameter efficiency, training time, and memory usage. Results demonstrated that compact variants of the N-BEATS architecture, specifically the optimized G-NBEATS and I-NBEATS models, delivered the highest predictive performance, achieving R^2 scores of 98.7% and 98.5%, respectively. These models substantially reduced computational demands, lowering parameter counts by up to 98% relative to N-HiTS (3.1 M parameters), training times by over 90% compared to Transformer and TFT model ($>$ 2300 s), and peak GPU memory usage by more than 95% relative to Transformer (997.67 MB VRAM). Qualitative assessments further confirmed their ability to accurately capture both diurnal ramp patterns and seasonal transitions. This study introduces a lightweight, deployable forecasting framework that combines high accuracy with low computational cost, offering practical solutions for real-time solar power prediction in resource-constrained environments.

INDEX TERMS Deep learning, green energy, hyperparameter optimization, neural networks, photovoltaic systems, power generation, renewable energy sources, smart grids, solar power generation, time series analysis.

I. INTRODUCTION

The continuous evolution of renewable energy technologies has become essential for individuals, businesses, and industries worldwide. With global electricity consumption

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaodong Liang.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

increasing annually [12], there is a growing need for sustainable energy solutions. Photovoltaic (PV) systems have emerged as a viable option to generate electricity, particularly in remote areas that require low to medium power levels, due to their adaptability and reliance on solar energy as an input source [13]. Significant research efforts are directed toward improving the efficiency and reliability of PV cells and modules, reducing production costs, and making solar energy generation more economically viable [14]. In many parts of the world, solar energy has already become the most cost-effective option for new electricity generation, a trend expected to drive further investment in the sector [31].

PV systems offer a promising solution for low-cost electricity production. However, effective integration of PV plants into the energy market is heavily based on accurate forecasting of solar energy generation. Such forecasts are critical for efficient resource management and ensuring grid stability [26]. Accurate predictions of solar power generation help address the inherent variability of PV output, thus improving grid reliability, maintaining power quality, and facilitating greater adoption of PV systems [25].

Despite its advantages, solar power generation faces challenges due to the intermittent and variable nature of sunlight. Unlike traditional energy sources, PV systems are highly dependent on weather conditions, leading to grid-related issues such as frequency instability, power dispatch challenges, and fluctuations in voltage and current [25]. These challenges underscore the importance of accurately predicting PV power generation, which remains a critical focus in PV system engineering.

To address these challenges, researchers have turned to advanced forecasting techniques. Traditional methods, such as statistical models, often fail to capture the nonlinearity and uncertainty associated with time series solar data [2]. In recent years, deep learning algorithms have shown significant promise in improving the accuracy of PV energy generation forecasts. Models such as Long Short-Term Memory (LSTM) [20], Transformer [33], and hybrid approaches have demonstrated notable success. For instance, LSTM-Conv has proven effective in processing extended time sequences, even without weather data, while Transformer models offer flexibility in forecasting specific future time points and reducing training durations [27], [33]. Additionally, the MF-NBEA framework, which integrates multilevel data fusion with Neural Basis Expansion Analysis (N-BEATS), has been proposed to enhance regional solar power generation predictions [21]. Similarly, hybrid frameworks combining Temporal Convolutional Networks (TCN) with physics-based and data-driven models have been explored [34]. However, hybrid approaches often face limitations due to their complexity and computational demands [16], [19]. Notably, the Temporal Fusion Transformer (TFT) has demonstrated superior accuracy in hourly day-ahead solar power generation forecasts compared to traditional methods like ARIMA, LSTM, MLP, and XGBoost across multiple facilities in Germany and Australia [37].

Building on recent advancements, this study implements the N-BEATS algorithm as a single model to address the limitations associated with forecasting approaches that fall short in their complexity. The goal is to improve forecasting performance by leveraging the unique strengths of the architecture while developing a lightweight and accurate solution for the forecasting of photovoltaic power generation. The key contributions of this study are as follows:

- **Lightweight N-BEATS architecture for efficient forecasting**: A streamlined N-BEATS configuration is proposed, designed to deliver accurate solar power forecasts while reducing model depth and computational complexity. The results demonstrate that a substantially shallower architecture can achieve comparable accuracy to the original model, while offering a faster training process.
- **Comparative analysis of generic and interpretable N-BEATS variants**: Both the generic and interpretable forms of the N-BEATS architecture are systematically examined to clarify the distinct roles of trend and seasonality stacks in forecasting performance. This analysis provides insights into how architectural design influences predictive behavior.
- **Comprehensive evaluation on long-range and season-specific datasets**: The optimized N-BEATS models are trained on a continuous long-range dataset and on season-specific subsets to assess their ability to capture long-term temporal dependencies as well as short-term seasonal variations in one-day-ahead solar power forecasting.

This research is structured as follows: Section I introduces the study, providing background information, defining the problem, and outlining the scope, and objectives. Section II briefly describes the photovoltaic systems. Section III provides related work on the N-BEATS algorithm. Section IV details the methodology, pre-processing tools, computational environments, and Python libraries used. Section V presents the performance evaluation of the proposed refined N-BEATS models for PV power generation forecasting. Section VI provides a discussion of the results and their limitations. Finally, Section VI concludes the study, summarizing key findings and suggesting directions for future research.

II. OVERVIEW OF PHOTOVOLTAIC SYSTEMS

Solar power can be generated through two main technologies: *concentrated solar power (CSP)* and *photovoltaic (PV) power*. CSP, also known as solar thermal power, operates similarly to conventional thermal power plants by using solar energy to produce steam that drives turbines. In contrast, PV systems use semiconductor materials to convert sunlight directly into direct current (DC) electricity through the photovoltaic effect. The DC generated is then converted to

alternating current (AC) using inverters and fed into the power grid [13].

PV systems do not store thermal energy because they generate electricity directly, and large-scale electricity storage remains a challenge. However, CSP systems can store thermal energy, allowing for more stable electricity production and better integration with the grid. Despite this advantage, CSP deployment is limited by its high capital cost and technological complexity, while the decreasing costs and simplicity of PV technology have made it the dominant choice for modern solar power installations [13].

PV systems convert solar irradiance directly into electricity through the photovoltaic effect in semiconductor materials. Their electrical performance is primarily governed by the intensity of solar irradiance (I_s), the active surface area (A), and the operating temperature of the PV cells (T_c). The energy conversion efficiency (η_{ele}) of a PV module can be expressed as [4]:

$$\eta_{ele} = \frac{P_{max}}{I_s \cdot A} \tag{1}$$

where P_{max} is the maximum electrical power output, I_s is the incident solar irradiance (W/m^2), and A is the panel surface area (m^2). Temperature management is therefore critical to maintaining PV system performance and maximizing power output.

III. RELATED WORK

Recent years have seen rapid advances in solar photovoltaic (PV) power generation forecasting, with deep learning models emerging as the most widely adopted and effective approaches. As summarized in Table 1, numerous studies have introduced powerful neural architectures capable of capturing temporal dependencies while incorporating exogenous environmental variables. Traditional sequence models such as Long Short-Term Memory (LSTM) networks [24], [29], [30], [36], Temporal Convolutional Networks (TCN) [23], and Transformer-based models [18], [28], [33], [35], [37], [39], [40], [42] have significantly improved PV forecasting accuracy compared to statistical and shallow-learning baselines.

LSTM, as a variant of Recurrent Neural Network (RNN), addresses the vanishing gradient problem using memory cells and gating mechanisms, enabling them to retain long-term dependencies while discarding irrelevant information [24]. Gensler et al. [29] showed that LSTM-based architectures consistently deliver more reliable PV forecasts than other deep learning models, while Hossain and Mahmood [30] achieved a mean absolute error (MAE) of 0.36, outperforming Extreme Learning Machine (1.38), Generalized Regression Neural Network (0.67), and traditional RNN (0.47) baselines. Luo et al. [36] further enhanced LSTM performance through a physics-constrained design for hourly day-ahead PV forecasting.

Transformer-based models, originally developed for natural language processing [42], have recently demonstrated state-of-the-art results in PV forecasting. Santos et al. [37] reported that the Temporal Fusion Transformer (TFT) achieved an RMSE of 0.276, an MAE of 0.120, and a mean absolute scaled error (MASE) of 0.390, outperforming ARIMA, MLP, LSTM, and XGBoost models. TFT extends the standard Transformer with static covariate encoders, gating mechanisms, variable selection networks, and quantile forecasting [35]. Other Transformer variants have also surpassed GRU and DNN approaches [40], while hybrid CNN-LSTM-Transformer frameworks reduced mean absolute percentage error (MAPE) relative to GRU-CNN, DeepAR, AB-Net, ARIMA, and Prophet [18].

TCN has likewise emerged as a strong contender for time-series tasks. By employing causal and dilated convolutions with residual connections, they efficiently capture long-range dependencies without future data leakage, offering high accuracy with fewer parameters and lower computational cost [23]. Their simplicity and interpretability make them suitable alternatives to recurrent and attention-based models for PV forecasting.

Despite these advances, many state-of-the-art models remain challenging to deploy in real-world PV forecasting systems due to high computational demands, long training times, and sensitivity to noisy or sparse inputs. As shown in Table 1, existing N-BEATS studies have focused primarily on load or consumption forecasting tasks and have not been applied to solar PV generation. Moreover, previous work has not systematically evaluated computational efficiency metrics such as parameter counts, training time, or GPU memory usage. These gaps highlight the need for a lightweight and accurate N-BEATS-based framework specifically designed for real-time solar power forecasting.

A. NEURAL BASIS EXPANSION ANALYSIS FOR TIME SERIES (N-BEATS)

N-BEATS [38] is a deep learning architecture designed specifically for univariate time-series forecasting. Its key strength lies in achieving state-of-the-art performance on benchmark datasets (M3, M4) without handcrafted features or time-series-specific components. As shown in Figure 3, the model consists of fully connected layers organized into stacks and blocks, connected by forward and backward residual links, enabling efficient gradient propagation and expressive function approximation. There are two primary variants: the generic model (G-NBEATS) and the interpretable model (I-NBEATS), which can be combined to further enhance performance.

N-BEATS has demonstrated an 11% improvement over traditional statistical benchmarks and a 3% improvement over the previous M4 competition winner [38]. Recent studies have built on its modular design: Singh et al. (2022), Olivares et al. (2023), and Kasprzyk et al. (2024) proposed various enhanced forms incorporating exogenous variables, hybrid loss functions, and destandardization to improve accuracy, robustness, and interpretability. This growing body

TABLE 1. Comparison of recent deep learning approaches for photovoltaic (PV) power forecasting.

Ref	Author	Approach	Contributions	Limitations
[5]	Singh *et al.* (2024)	Applied N-BEATS for univariate and multivariate power forecasting.	Demonstrated robustness for short-term load forecasting with spike detection using Ontario energy data.	No integration of exogenous variables, and limited interpretability.
[6]	Olivares *et al.* (2023)	Proposed N-BEATSx integrating exogenous variables.	Improved accuracy by ∼20% over baseline N-BEATS and added interpretability via decomposition.	Higher model complexity and potential overfitting with limited data.
[8]	Kasprzyk *et al.* (2025)	Enhanced N-BEATS with hybrid loss and destandardization.	Achieved state-of-the-art mid-term forecasting across 35 European datasets.	Not solar-specific and lacks real-time forecasting adaptability.
[7]	Oreshkin *et al.* (2021)	Original N-BEATS for mid-term electricity load.	Outperformed 10 baseline models in accuracy and bias reduction.	Does not incorporate exogenous variables and generic block structure.
[9]	Shaikh *et al.* (2022)	Interpretable N-BEATS with residential smart grid dataset.	Scaled N-BEATS to multi-user energy consumption data with covariate inputs.	Focused on consumption data, not PV generation. Scalability untested.
[10]	Loh *et al.* (2024)	Numerical–Categorical Radial Basis Function DNN ensemble.	Lightweight and robust to sensor noise. High accuracy for solar plants.	Complex ensemble design and lacks time series decomposition.
[11]	Wang *et al.* (2025)	LightGTS with periodic-aware tokenization and decoding.	Achieved SOTA generalization with minimal parameters across benchmarks.	Not specific to solar forecasting and limited physical interpretability.

of work highlights N-BEATS as a promising foundation for further refinement and domain-specific adaptation in PV forecasting.

To address the limitations of existing models, this study develops and evaluates refined N-BEATS architectures on the publicly available PVODataset [43], which captures realistic solar power generation patterns. The investigation explores both generic and interpretable variants of N-BEATS, with an emphasis on improving computational efficiency while maintaining strong predictive performance across long-range and season-specific datasets. Following this internal exploration, the most effective lightweight configuration is benchmarked against seven representative baseline architectures: LSTM, Transformer, TCN, TFT, TiDE, N-HiTS, and DLinear, selected for their proven performance on temporal data and their architectural diversity. This approach is intended to establish a forecasting framework that balances accuracy, interpretability, and efficiency, making it suitable for practical deployment in renewable energy applications.

IV. METHODOLOGY

This study investigates the feasibility of applying the Neural Basis Expansion Analysis for Time Series (N-BEATS) architecture for photovoltaic (PV) power generation forecasting using the publicly available PVODataset introduced by Yao et al. [43]. The overall methodology encompasses the experimental environment, dataset preparation and partitioning, N-BEATS architecture exploration, hyperparameter tuning, and performance evaluation on both a single long-range forecast day and season-wise test sets.

A. EXPERIMENTAL SETUP

All experiments were conducted using Jupyter Lab as the development environment, with Python 3.11.9 as the primary programming language. The deep learning models were implemented using the Darts time-series forecasting library, which is built on top of PyTorch, and supported by a range of additional scientific libraries such as `pandas` and `NumPy` for data handling, `Matplotlib` for visualization, and Weights & Biases (`wandb`) for experiment tracking and GPU memory monitoring.

Training and evaluation were performed on a workstation equipped with an NVIDIA GeForce RTX 4080 GPU with 16 GB of VRAM, running CUDA 12.4 (driver version 551.23). The PyTorch build also utilized CUDA 12.4, and a GPU. This hardware configuration enabled efficient training of deep learning models and supported the large autoregressive rollouts required for long-horizon solar power forecasting.

B. DATASET

The PVODataset (PVOD) provides high-resolution multi-source PV data collected from 10 PV stations (Station 0 to Station 9) across Hebei Province, China (36.64–39.52° N, 113.64–117.46° E). Each station file contains synchronized local measurement data (LMD) and numerical weather prediction (NWP) variables at 15-minute resolution from 2018-07-01 to 2019-06-13. LMD includes PV power output, global and diffuse irradiance, temperature, pressure, wind speed, and direction. NWP is generated from the WRF-ARW v3.9.1 model at 4 km spatial resolution, 45 vertical levels, and 15-minute temporal resolution, initialized daily using ECMWF global forecasts [43]. PVOD

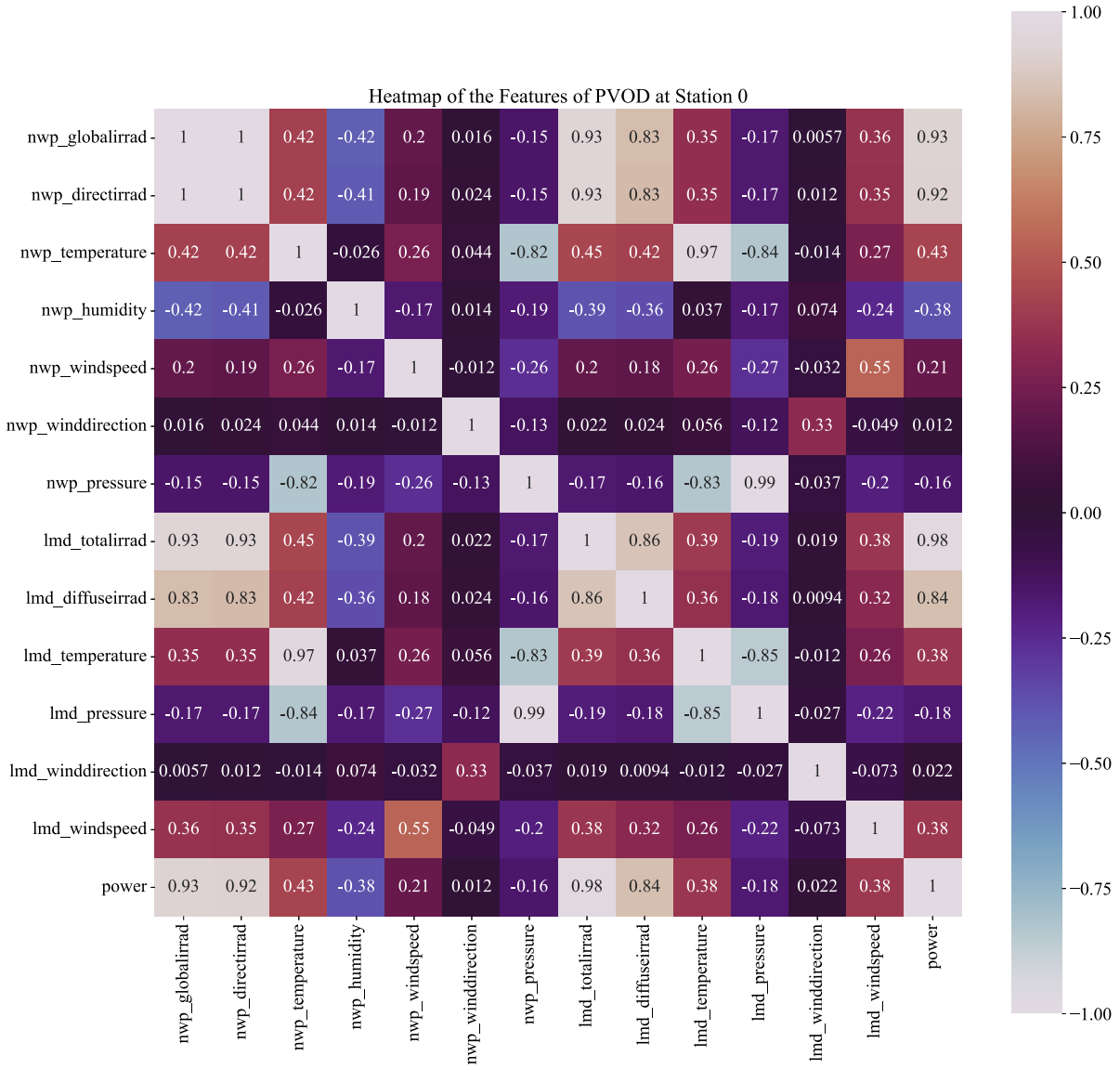
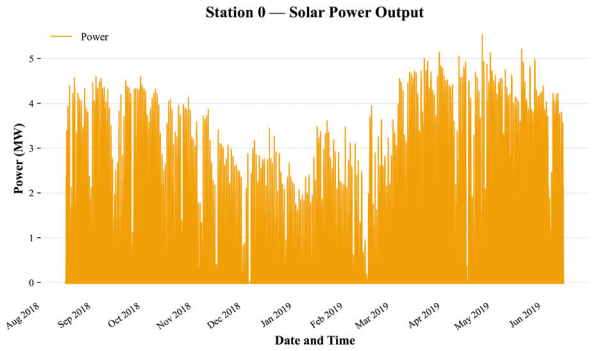


FIGURE 1. Pearson correlation heatmap of PVOD.

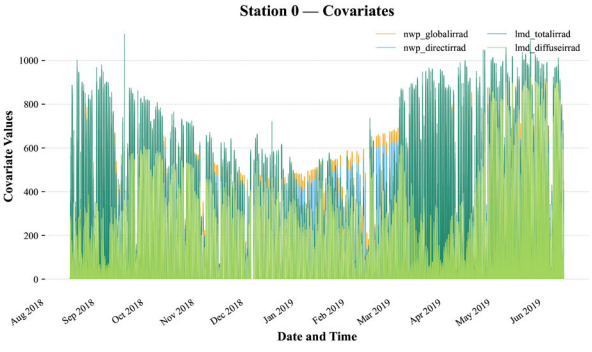
provides 271,968 records with full metadata (capacity, tilt, panel type, inverter, and geographic coordinates) for all 10 stations.

Station 0 was selected for this study because it contains a complete and consistent time series (28,896 samples) without large gaps, ensuring stable model training. Although Yao et al. [43] noted that combining multiple stations typically improves forecast accuracy due to strong inter-station correlations, this study deliberately uses a single-station setup to avoid the geographical smoothing effect that can mask true model behavior. This is because the primary focus is on exploring algorithmic performance rather than spatial ensemble modeling. By employing only Station 0, a controlled evaluation of architectural modifications and hyperparameter tuning is achieved, eliminating potential confounding effects from inter-station correlations that could artificially reduce variability [43].

A Pearson correlation analysis was performed between all available meteorological variables and PV power output. Following the recommendations of [17], a Pearson correlation analysis was conducted as shown in Figure 1, and four covariates with correlation coefficients greater than 0.7 were identified and retained as predictive exogenous features: NWP global irradiance (`nwp_globalirrad`), NWP direct irradiance (`nwp_directirrad`), LMD global irradiance (`lmd_totalirrad`), and LMD diffuse irradiance (`lmd_diffuseirrad`). Data preprocessing steps included aligning timestamps to local time, removing duplicates and extreme outliers, dropping incomplete early-year columns, and normalizing both target and



(a) Power output



(b) Covariates

FIGURE 2. **Power output and covariates data in station 0 PVOD.**

covariates using MinMax scaling, with inverse transformation applied during evaluation.

Two experimental setups were used. For the long-range forecast experiment, data from 2018-08-15 to 2019-04-14 were used for training, while 2019-04-15 to 2019-06-13 served as the testing period. The testing focused on a single representative day (15 April 2019) to evaluate generalization for long-term rollouts. For the seasonal forecast experiment, seasonal subsets were constructed based on climatological ranges: Spring (90:10 split) from 2019-03-01 to 2019-05-30, Summer (80:20 split) from 2018-08-16 to 2018-08-31 and 2019-06-01 to 2019-06-13, Autumn (80:20 split) from 2018-09-01 to 2018-11-30, and Winter (80:20 split) from 2018-12-01 to 2019-02-28. The Spring subset used a 90:10 ratio because it had the shortest data range among the four seasons, and allocating more data to training was necessary to ensure model stability. Each subset was split chronologically while preserving temporal order.

C. N-BEATS ARCHITECTURE EXPLORATION

The N-BEATS model was implemented using the Darts time-series library [1], leveraging PyTorch Lightning for GPU acceleration and Weights & Biases (`wandb`) for experiment tracking. The generic N-BEATS architecture was selected as the base, given its proven capacity to model complex temporal patterns without explicit seasonality decomposition [38].

Following Oreshkin et al. [38], each N-BEATS block receives an input window x_ℓ and produces two outputs: a backcast $\hat{x}_\ell$ and a forecast $\hat{y}_\ell$. Internally, the block first passes x_ℓ through a fully connected (FC) stack to generate backward (θ^b_ℓ) and forward (θ^f_ℓ) expansion coefficients:

$$\begin{aligned} h_{\ell,1} &= \mathrm{FC}_{\ell,1}(x_\ell), \quad h_{\ell,2} = \mathrm{FC}_{\ell,2}(h_{\ell,1}), \\ h_{\ell,3} &= \mathrm{FC}_{\ell,3}(h_{\ell,2}), \quad h_{\ell,4} = \mathrm{FC}_{\ell,4}(h_{\ell,3}), \\ \theta^b_\ell &= W^b_\ell h_{\ell,4}, \quad \theta^f_\ell = W^f_\ell h_{\ell,4}. \end{aligned} \tag{2}$$

These coefficients are then projected onto two sets of basis vectors $\{v^b_i\}$ and $\{v^f_i\}$ to produce the block outputs:

$$\hat{x}_\ell = \sum_i \theta^b_{\ell,i} v^b_i, \qquad \hat{y}_\ell = \sum_i \theta^f_{\ell,i} v^f_i. \tag{3}$$

Blocks are organized into stacks using a doubly residual stacking (DRESS) scheme, where the backcast residual of each block becomes the input to the next block, and all partial forecasts are aggregated:

$$x_{\ell+1} = x_\ell - \hat{x}_\ell, \qquad \hat{y} = \sum_\ell \hat{y}_\ell. \tag{4}$$

In the interpretable configuration, the basis functions v^f_i are constrained to represent either low-degree polynomial trend components,

$$\hat{y}^{\text{trend}}_{s,\ell} = \sum_{i=0}^{p} \theta^f_{s,\ell,i} t^i, \tag{5}$$

or Fourier seasonality components,

$$\hat{y}^{\text{seas}}_{s,\ell} = \sum_{i=0}^{\lfloor H/2 \rfloor - 1} \theta^f_{s,\ell,i} \cos(2\pi i t) + \theta^f_{s,\ell,i+\lfloor H/2 \rfloor} \sin(2\pi i t), \tag{6}$$

allowing the model to separately capture trend and seasonality patterns.

Architecture exploration involved systematically modifying the number of stacks and blocks, the number of fully connected layers per block, the hidden layer width, and the type of basis functions (polynomial degree for the interpretable variant). Several activation functions were compared, including ReLU, PReLU, GELU, RReLU, Tanh, Softplus, SELU, and LeakyReLU. Monte Carlo Dropout ($p = 0.05$) layers were inserted between fully connected layers to improve generalization for G-NBEATS, while standard Dropout ($p = 0.0$) was used for I-NBEATS. The four selected covariates were incorporated as aligned past covariates. GPU memory allocation, model size, parameter counts, training duration, and validation performance were logged for each configuration to support efficiency–accuracy trade-off analysis. The final configuration is summarized in Table 2.

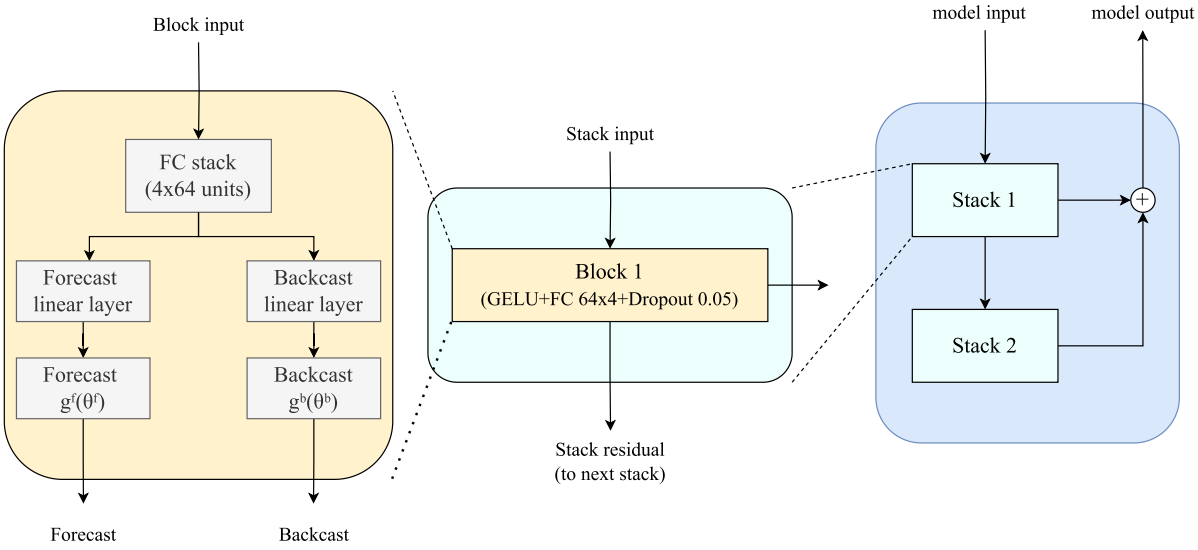


FIGURE 3. Light and fast architecture of G-NBEATS.

TABLE 2. Model architecture (structural hyperparameters).

Parameter	Values	Final value
Stacks	1, 2, 3, 4, 30	2
Blocks per stack	1, 2, 3	1
FC layers per block	3, 4	4
Hidden size	64, 128, 256, 512	64
Activation	PReLU, RReLU, ELU, Softplus, Tanh, SELU, LeakyReLU, Sigmoid, GELU, ReLU	RReLU, GELU
Trend polynomial degree	1, 2, 3, 4	2

The solar power forecasting problem used in this study is formulated as a multivariate time series regression task. Let $\mathbf{x}_t = [P_t, W_{1,t}, W_{2,t}, W_{3,t}, W_{4,t}]$ denote the input vector at time t, where P_t is the historical solar power and $W_{i,t}$ are the corresponding selected NWP and LMD features from PVOD [43]. Given a lookback window of L, the input sequence is

$$\mathbf{X}_{t-L+1:t} = \{\mathbf{x}_{t-L+1}, \ldots, \mathbf{x}_t\}, \tag{7}$$

The model aims to predict the next H future power values:

$$\hat{P}_{t+1:t+H} = f_\theta(\mathbf{X}_{t-L+1:t}), \tag{8}$$

where $f_\theta(\cdot)$ denotes the forecasting function parameterized by θ.

The forecasting task can also be formally expressed as a stochastic optimization problem. Given the training data distribution $\mathcal{D}$, the model aims to minimize the expected forecasting error:

$$\min_\theta \ \mathbb{E}_{(X_t,P_t)\sim\mathcal{D}}\big[(P_t - f_\theta(X_t))^2\big], \tag{9}$$

where f_θ denotes the nonlinear mapping parameterized by θ, representing the N-BEATS forecasting function. This objective corresponds to learning an operator that captures the complex nonlinear transformation between historical power output and relevant meteorological features.

In addition, the forecasting model complements the physical energy-balance relationship of photovoltaic generation, expressed as:

$$P = \eta_{\text{ele}} A I_s, \tag{10}$$

where η_{ele} is the electrical conversion efficiency, A is the active surface area, and I_s is the incident solar irradiance. The N-BEATS model implicitly learns the nonlinear mapping that governs the variability of η_{ele} with respect to temperature, irradiance, and time-dependent effects, thereby providing a data-driven yet physically consistent forecasting framework.

Each N-BEATS block consists of a fully connected stack that generates two outputs: a *forecast* and a *backcast*. Given the block input $\mathbf{z}_b$, its operations can be expressed as:

$$\begin{aligned} \mathbf{h}_b &= \phi(\mathbf{W}_b\mathbf{z}_b + \mathbf{b}_b), \\ \hat{\mathbf{y}}_b &= \mathbf{G}_b(f_b(\mathbf{X}_b)), \quad \tilde{\mathbf{y}}_b = \mathbf{H}_b(f_b(\mathbf{X}_b)), \\ \mathbf{z}_{b+1} &= \mathbf{z}_b - \tilde{\mathbf{y}}_b, \end{aligned} \tag{11}$$

where $\phi(\cdot)$ denotes the nonlinear activation function (e.g., GELU or RReLU), and $\mathbf{G}_b(\cdot)$ and $\mathbf{H}_b(\cdot)$ represent the forecast and backcast generator functions, respectively. The residual connection $\mathbf{z}_{b+1} = \mathbf{z}_b - \tilde{\mathbf{y}}_b$ allows the network to iteratively refine the signal representation across blocks.

TABLE 3. Training and regularization hyperparameters.

Hyperparameter	Values	Final value
Batch size	32, 64	64
Dropout	0.0, 0.05, 0.1	0.0, 0.05

The final forecast output of the model is computed as the sum of all block-level forecasts:

$$\hat{\mathbf{y}} = \sum_{b=1}^{B} \hat{\mathbf{y}}_b. \tag{12}$$

The model parameters θ are optimized by minimizing the Mean Squared Error (MSE) between the actual and predicted solar power values:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left(P_{t+i} - \hat{P}_{t+i} \right)^2. \tag{13}$$

D. HYPERPARAMETER TUNING AND EVALUATION

The model was trained using an input chunk length of 288 (equivalent to 3 days of 15-minute data) and an output chunk length of 3 (equivalent to 45 minutes predicted per iteration). Other tuned hyperparameters included batch size (32–64), number of epochs set to 50, and Adam optimizer with Mean Squared Error (MSE) loss.

Because the model produces only 3-step predictions at a time, it was configured to perform autoregressive rollouts during forecasting: each newly predicted chunk was appended back into the input sequence to generate the next chunk, until the desired horizon was reached. This is consistent with the Darts N-BEATS implementation, where `output_chunk_length` defines the internal prediction stride, while the user-specified n in `predict(n=...)` determines the overall forecast horizon. This setup enables the model to learn from short-step outputs while still producing multi-hour or multi-day forecasts.

A manual testing search strategy was used to tune the hyperparameters. The best models were retrained with the full training data and evaluated on (i) long-range forecasting of 15 April 2019 to assess generalization, and (ii) each seasonal test set to quantify robustness across climatic regimes. Performance metrics included RMSE, MAE, MSE, and R^2, all computed on inverse-scaled output to ensure interpretability in MW units.

E. PERFORMANCE METRICS

RMSE, MAE, and MSE were used as performance metrics in this study due to their established effectiveness in evaluating forecasting models, as demonstrated in previous research [15], [18], [19], [29]. RMSE provides a comprehensive error measurement throughout the forecast period, effectively penalizing significant errors in square order, making it suitable for assessing overall precision [3]. The MAE metric, widely used in regression problems, calculates the average absolute difference between predicted and actual values, serving as a reliable measure of consistent forecast errors [18], [22], [36]. Meanwhile, MSE is commonly applied to evaluate the performance of a model and is calculated as described in equation (16) [32], [44]. Together, these metrics ensure a robust evaluation framework that addresses both overall accuracy and significant deviation detection. The equation of RMSE (14), MAE (15), MSE (16), and R^2 (17) is shown below.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{p}_i - p_i)^2} \tag{14}$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |\hat{p}_i - p_i| \tag{15}$$

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (\hat{p}_i - p_i)^2 \tag{16}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (\hat{p}_i - p_i)^2}{\sum_{i=1}^{N} (p_i - \bar{p})^2} \tag{17}$$

In addition to forecast accuracy, three efficiency metrics were recorded to evaluate computational performance: parameter count, training time, and maximum GPU memory usage (VRAM). The parameter count was obtained in PyTorch by summing the number of elements across all **trainable** tensors returned by `model.parameters()`[1] [41].

Training time was measured as wall-clock seconds using `time.time()` before and after the training call; for GPU runs, `torch.cuda.synchronize()` was invoked immediately before recording the end timestamp to ensure all kernels had finished executing.[2]

Peak VRAM usage was captured with PyTorch's CUDA memory profiler by resetting the peak counters before training and reading the maxima afterward (reporting both peak allocated and peak reserved where relevant).[3] All byte values were converted to megabytes (MB) by dividing by 1024^2. Together, these measurements provide a transparent, implementation view of the trade-offs between computational cost and predictive accuracy.

V. RESULTS

The model training and evaluation followed a two-stage process. In the first stage, the N-BEATS family of models was trained and refined to develop compact architectures with improved efficiency. A series of targeted ablation studies was conducted to examine the effects of activation functions, architectural depth, and trend polynomial degree

[1] Implemented as $\sum$ of `p.numel()` for all `p` with `p.requires_grad`

[2] Pattern: `st = time.time(); model.fit(...); torch.cuda.synchronize(); et = time.time();`

[3] `torch.cuda.reset_peak_memory_stats()` before training; after training: `torch.cuda.max_memory_allocated()` and `torch.cuda.max_memory_reserved()`. Instantaneous usage was also read via `torch.cuda.mem_get_info()`.

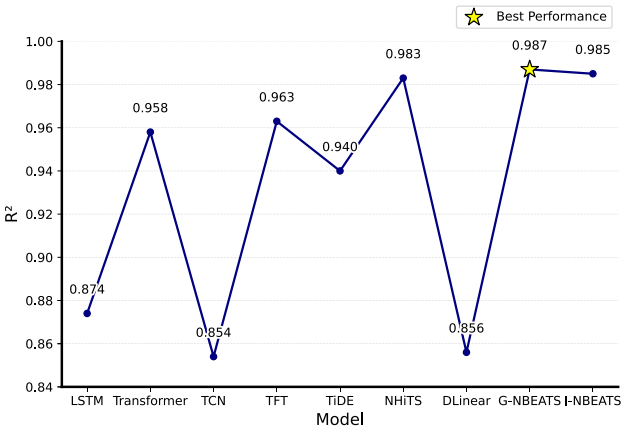


FIGURE 4. Performance comparison graph of modified N-BEATS and baseline models under default configuration.

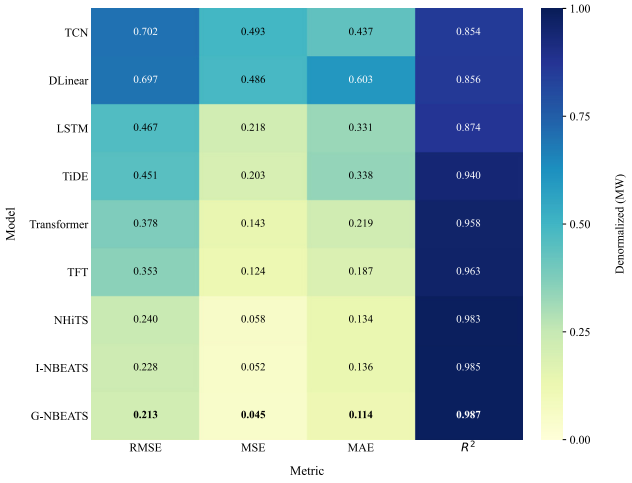


FIGURE 5. Performance comparison heatmap of modified N-BEATS and baseline models under default configuration.

on forecasting performance. In the second stage, the optimized N-BEATS variants were benchmarked against seven widely used baseline architectures, namely LSTM, TCN, Transformer, TFT, TiDE, N-HiTS, and DLinear, under identical experimental settings. Performance was assessed using four accuracy metrics (R^2, RMSE, MSE, and MAE) together with three efficiency metrics (parameter count, training time, and peak memory usage). This structured approach enabled a comprehensive comparison between the refined N-BEATS models and established baselines, providing clear insights into the trade-offs between model complexity, computational cost, and forecasting accuracy.

A. COMPARATIVE PERFORMANCE OF DEFAULT MODELS

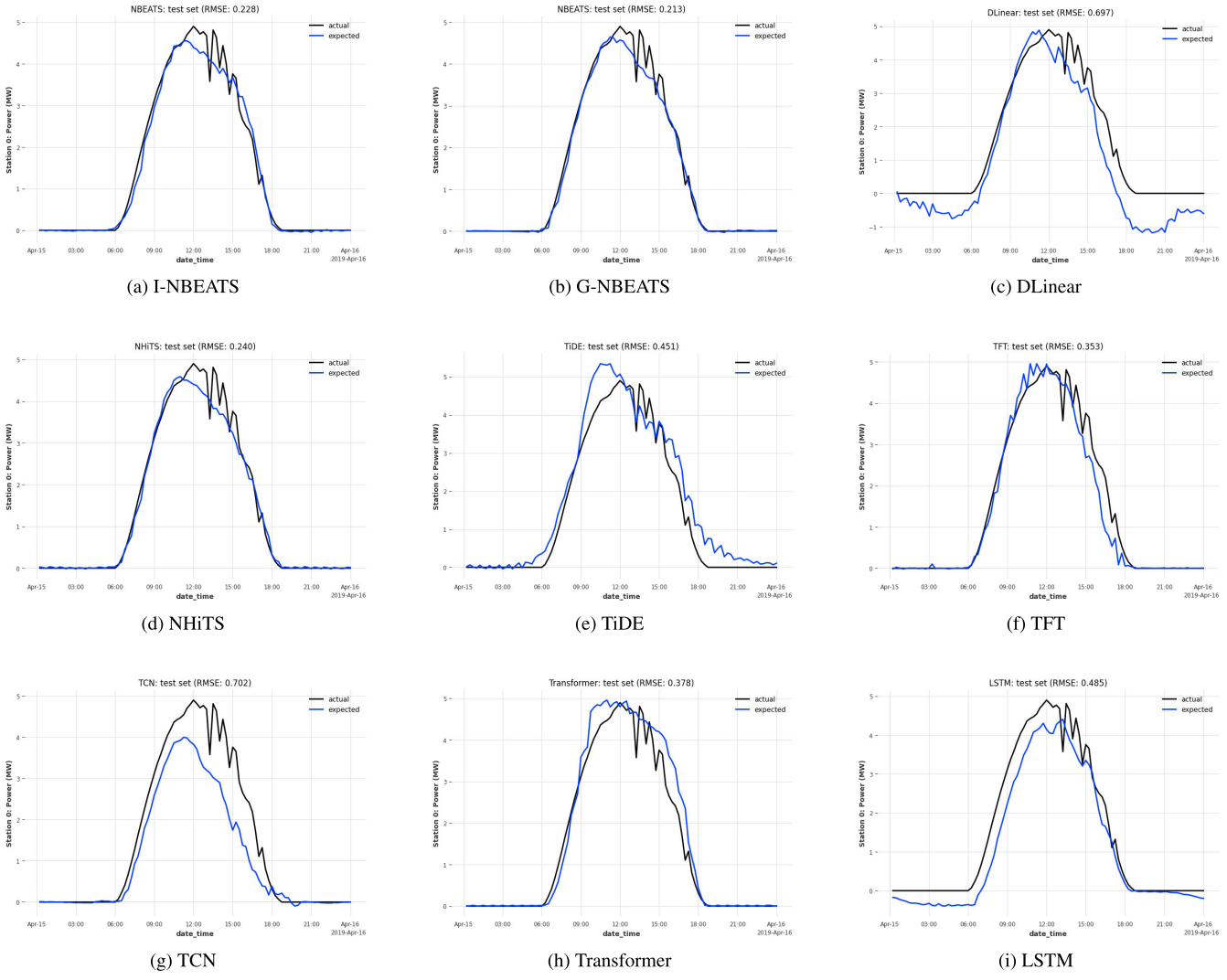
As shown in Figure 5, G-NBEATS achieved the highest overall precision among all models, reaching an RMSE of 0.042 (0.213 MW), which is approximately 11% lower than the second-best model, N-HiTS (0.047). This superior accuracy indicates that G-NBEATS effectively captured the temporal dynamics of photovoltaic generation. I-NBEATS also showed strong performance (RMSE 0.044), outperforming larger architectures such as Transformer and TFT despite having fewer than 220K parameters. In contrast, TCN and DLinear were the weakest performers, with RMSE values of 0.137 and 0.136, respectively, which are more than three times higher than G-NBEATS. This reflects their limited ability to model the complex nonlinear relationships inherent in the data. While LSTM achieved a moderately low RMSE (0.102), it struggled to model peak-hour variability, as evidenced by underestimation during high-generation periods. Performance comparison in R^2 can be seen in Figure 4.

Although LSTM had the smallest number of parameters (3.3K), its training time (699 s) was substantially longer than both G-NBEATS and I-NBEATS $\approx$ 182, indicating that small model size does not necessarily guarantee faster training. Meanwhile, the qualitative prediction plots in Figure 6 further reinforce these findings. G-NBEATS and N-HiTS closely tracked the observed power curve, capturing the sharp morning ramp and afternoon decay. By contrast, DLinear displayed a negative drift during nighttime and underestimated the evening shoulder, while TCN and LSTM under-predicted the afternoon peak. Transformer and TFT achieved competitive accuracy but required extreme training times (2300–2366 s) and memory usage (>900 MB), making them less practical for operational deployment.

B. SEASONAL GENERALISATION

The performance of the models varied substantially between seasons, as shown in Table 4. G-NBEATS consistently ranked among the best models in spring, autumn, and winter, achieving the lowest RMSE in autumn (0.054, 0.249 MW), which was more than 30% lower than TCN (0.103). In spring, G-NBEATS (0.081) outperformed Transformer (0.086) and N-HiTS (0.082), indicating its ability to capture transitional patterns when irradiance variability is high. During winter, N-HiTS delivered the best results (0.031, 0.111 MW), about 28% lower than Transformer (0.094), while G-NBEATS and I-NBEATS followed closely (0.043 and 0.040). Summer was the only season in which Transformer excelled (0.047), surpassing G-NBEATS (0.055) by roughly 15%, which may be attributed to the stability of irradiance and the attention mechanism's strength in modeling steady temporal dependencies.

These seasonal patterns suggest that architecture choice influences robustness under varying data regimes. G-NBEATS's strong performance in transitional seasons, such as autumn highlights its capacity to adapt to abrupt shifts in generation caused by cloud cover or shorter daylight cycles. N-HiTS's superior winter accuracy suggests it handles the higher volatility and sharper ramp rates typical of this season. Conversely, the poor seasonal robustness of TCN and DLinear, whose RMSE values were high in every season



(a) I-NBEATS (b) G-NBEATS (c) DLinear

(d) NHiTS (e) TiDE (f) TFT

(g) TCN (h) Transformer (i) LSTM

FIGURE 6. One-day-ahead prediction outputs of G-NBEATS, I-NBEATS, and baseline models on 15 April 2019.

(e.g., 0.123 spring; 0.153 autumn), implies that convolutional or purely linear approaches struggle to generalise beyond their training context. Overall, these results underscore the importance of model flexibility in handling seasonal variability in solar forecasting tasks.

C. COMPUTATIONAL EFFICIENCY

Although accuracy distinguished the models, their computational efficiency profiles differed even more markedly. G-NBEATS and I-NBEATS offered the best balance between accuracy and resource demands. Both used fewer than 220K parameters, completed training in under 190 seconds, and consumed less than 50 MB of memory. In stark contrast, TFT and Transformer required more than 2300 seconds to train and nearly 1 GB of memory each, representing a tenfold increase in compute burden despite only modest accuracy gains. N-HiTS also showed high accuracy but incurred heavy resource demands, requiring over 3.1M parameters and 144 MB VRAM. The training time

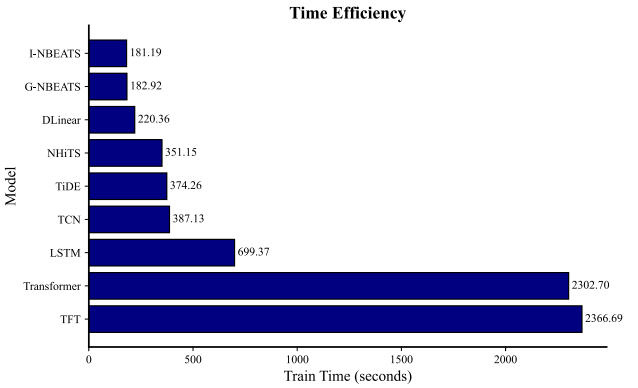


FIGURE 7. Training time of G-NBEATS, I-NBEATS, and baseline models.

differences are visualised in Figure 7, highlighting the steep efficiency penalty paid by Transformer-like architectures.

These findings demonstrate that model efficiency does not scale linearly with size. Although LSTM had the

TABLE 4. Performance of deep learning models across seasonal datasets. RMSE, MSE, MAE, and R^2 are reported in normalized units and megawatts (MW). The table also reports parameter count, train time, and VRAM peak.

Season	Model	RMSE/MSE/MAE/R^2		Params	Train time (s)	VRAM Peak (MB)
		Normalized	MW			
Spring	**G-NBEATS**	0.081/0.007/0.049/0.924	0.449/0.201/0.270/0.924	219 K	55.91	27.59
	I-NBEATS	0.097/0.009/0.061/0.891	0.537/0.288/0.337/0.891	210 K	58.61	44.03
	LSTM	0.110/0.012/0.076/0.860	0.608/0.370/0.417/0.860	3.3 K	215.73	48.29
	Transformer	0.086/0.007/0.043/0.915	0.474/0.225/0.240/0.915	549 K	706.31	997.67
	TCN	0.123/0.015/0.085/0.826	0.679/0.462/0.468/0.826	400	115.49	7.97
	TFT	0.097/0.010/0.055/0.891	0.538/0.289/0.304/0.891	18.2 K	718.36	118.47
	TiDE	0.098/0.010/0.066/0.889	0.541/0.293/0.362/0.889	416 K	114.05	57.90
	NHiTS	0.082/0.007/0.048/0.923	0.453/0.205/0.266/0.923	3.1 M	128.14	143.99
	DLinear	0.438/0.192/0.351/−1.213	2.420/5.858/1.938/−1.213	8.6 K	35.91	20.71
Summer	G-NBEATS	0.055/0.003/0.032/0.978	0.249/0.062/0.148/0.978	219 K	16.71	27.59
	I-NBEATS	0.077/0.006/0.046/0.957	0.350/0.123/0.209/0.957	210 K	22.49	44.03
	LSTM	0.053/0.003/0.036/0.980	0.242/0.058/0.164/0.980	3.3 K	53.63	48.29
	Transformer	0.047/0.002/0.031/0.984	0.214/0.046/0.140/0.984	549 K	181.97	997.67
	TCN	0.058/0.003/0.046/0.975	0.266/0.071/0.210/0.975	400	17.45	7.97
	TFT	0.149/0.022/0.089/0.839	0.679/0.461/0.407/0.839	18.2 K	185.24	116.75
	TiDE	0.189/0.036/0.131/0.741	0.861/0.741/0.598/0.741	416 K	15.99	57.68
	NHiTS	0.055/0.003/0.038/0.978	0.252/0.064/0.175/0.978	3.1 M	49.99	144.04
	DLinear	0.200/0.040/0.157/0.710	0.910/0.827/0.717/0.710	8.6 K	10.01	18.98
Autumn	**G-NBEATS**	0.054/0.003/0.029/0.788	0.249/0.062/0.132/0.788	219 K	53.63	27.59
	I-NBEATS	0.083/0.007/0.046/0.509	0.378/0.143/0.209/0.509	210 K	55.90	44.03
	LSTM	0.082/0.007/0.052/0.519	0.374/0.140/0.236/0.519	3.3 K	200.54	48.29
	Transformer	0.172/0.030/0.100/−1.132	0.788/0.621/0.456/−1.132	549 K	672.39	997.67
	TCN	0.103/0.011/0.054/0.231	0.473/0.224/0.247/0.231	400	112.64	7.97
	TFT	0.108/0.012/0.052/0.158	0.495/0.245/0.240/0.158	18.2 K	690.18	116.75
	TiDE	0.434/0.189/0.279/−12.605	1.990/3.960/1.279/−12.605	416 K	108.25	57.68
	NHiTS	0.124/0.015/0.071/−0.102	0.567/0.321/0.327/−0.102	3.1 M	122.23	144.04
	DLinear	0.153/0.023/0.095/−0.687	0.701/0.491/0.433/−0.687	8.6 K	63.09	18.98
Winter	G-NBEATS	0.043/0.002/0.027/0.966	0.156/0.024/0.095/0.966	219 K	26.02	27.60
	I-NBEATS	0.040/0.002/0.022/0.971	0.144/0.021/0.080/0.971	210 K	56.48	44.03
	LSTM	0.122/0.015/0.095/0.734	0.438/0.191/0.341/0.734	3.3 K	196.54	48.29
	Transformer	0.094/0.009/0.056/0.840	0.339/0.115/0.199/0.840	549 K	665.05	997.67
	TCN	0.043/0.002/0.023/0.967	0.154/0.024/0.083/0.967	400	109.71	7.97
	TFT	0.144/0.021/0.078/0.630	0.516/0.266/0.279/0.630	18.2 K	681.61	116.75
	TiDE	0.177/0.031/0.105/0.438	0.635/0.404/0.377/0.438	416 K	106.87	57.68
	NHiTS	0.031/0.001/0.018/0.983	0.111/0.012/0.065/0.983	3.1 M	123.19	144.04
	DLinear	0.085/0.007/0.078/0.871	0.305/0.093/0.281/0.871	8.6 K	60.76	18.98

fewest parameters (3.3K), its training time (699 s) was longer than G-NBEATS, illustrating that smaller models can be inefficient when sequential dependencies are long. Conversely, G-NBEATS achieved both low RMSE and fast training due to its fully connected block-based structure, which allows parallel processing and reduces gradient path length. This makes G-NBEATS and I-NBEATS more viable for deployment in resource-constrained settings, whereas Transformer and TFT are best suited for offline or high-performance environments where accuracy outweighs efficiency constraints.

D. N-BEATS ABLATIONS

1) EFFECT OF ACTIVATION FUNCTIONS

The choice of activation function had a clear impact on performance, as shown in Table 5 and Table 6. For G-NBEATS, GELU achieved the lowest RMSE (0.046), about 20% lower than the default ReLU (0.078), while also reducing MAE and boosting R^2 to 0.984. For I-NBEATS, RReLU achieved the best accuracy (0.044) with $R^2 = 0.985$, improving upon the untuned baseline by roughly 25%. In contrast, smooth activations like Softplus and Sigmoid produced higher errors, suggesting that they may oversmooth temporal dynamics and reduce responsiveness to sharp ramps in PV output. These results suggest that smooth rectifiers such as GELU and stochastic variants like RReLU offer more expressive nonlinear transformations suited to volatile solar patterns.

Training-time patterns also reveal interesting trade-offs. While ReLU and PReLU achieved relatively high accuracy, they had longer training times (175–182 s) than GELU (170 s) and RReLU (181 s), suggesting that GELU offers a better balance between accuracy and computational cost. This highlights how activation choice influences both convergence speed and representational power. Architectures using GELU and RReLU showed lower gradient variance, enabling faster convergence and more stable training, whereas ReLU-based models exhibited slower convergence and greater overfitting risk. These findings underscore the critical role of activation selection

TABLE 5. Activation-function ablation for G-NBEATS. Metrics are RMSE/MSE/MAE/R^2 in normalized units and MW; lower is better for errors, higher is better for R^2.

Activation Function	RMSE/MSE/MAE/R^2		Params	Train time (s)	VRAM Peak (MB)
	Normalized	MW			
PReLU	0.045/0.002/0.028/0.984	0.230/0.053/0.146/0.984	219 K	177.12	27.38
RReLU	0.050/0.003/0.028/0.980	0.257/0.066/0.144/0.980	219 K	175.01	27.63
ELU	0.070/0.005/0.037/0.961	0.361/0.130/0.192/0.961	219 K	172.86	27.38
Softplus	0.051/0.003/0.031/0.979	0.264/0.070/0.158/0.979	219 K	170.78	27.38
Tanh	0.068/0.005/0.042/0.964	0.351/0.123/0.216/0.964	219 K	172.19	27.13
SELU	0.071/0.005/0.033/0.961	0.362/0.131/0.170/0.961	219 K	172.92	27.38
LeakyReLU	0.064/0.004/0.037/0.969	0.326/0.106/0.187/0.969	219 K	174.27	27.38
Sigmoid	0.060/0.004/0.034/0.972	0.308/0.095/0.176/0.972	219 K	173.91	27.13
GELU	**0.046/0.002/0.026/0.984**	**0.235/0.055/0.131/0.984**	**219 K**	**170.46**	**27.38**
ReLU	0.078/0.0060/0.040/0.953	0.400/0.160/0.205/0.953	219 K	175.88	27.13

TABLE 6. Activation-function ablation for I-NBEATS. Metrics are RMSE/MSE/MAE/R^2 in normalized units and MW. Bold indicates per-column best (min RMSE/MSE/MAE, max R^2, min Train time/VRAM; ties bolded).

Activation Function	RMSE/MSE/MAE/R^2		Params	Train time (s)	VRAM Peak (MB)
	Normalized	MW			
PReLU	0.055/**0.003**/0.036/0.977	0.281/0.079/0.182/0.977	210 K	182.48	43.78
RReLU	**0.044/0.002/0.027/0.985**	**0.228/0.052**/0.136/**0.985**	210 K	181.19	44.03
ELU	0.056/0.003/0.033/0.976	0.285/0.081/0.167/0.976	210 K	178.93	43.78
Softplus	0.054/0.003/0.034/0.978	0.276/0.076/0.175/0.978	210 K	181.92	43.78
Tanh	0.057/0.003/0.035/0.975	0.290/0.084/0.177/0.975	210 K	174.39	**43.53**
SELU	0.091/0.008/0.053/0.936	0.465/0.216/0.272/0.936	210 K	178.87	43.78
LeakyReLU	0.060/0.004/0.032/0.972	0.306/0.094/0.166/0.972	210 K	180.23	43.78
Sigmoid	0.059/0.004/0.034/0.973	0.303/0.092/0.175/0.973	210 K	176.41	**43.53**
GELU	0.051/0.003/0.029/0.980	0.262/0.055/**0.131**/0.980	210 K	**173.75**	43.79
ReLU	0.070/0.005/0.037/0.962	0.357/0.127/0.189/0.962	210 K	179.72	**43.53**

in optimising N-BEATS architectures for operational forecasting.

2) EFFECT OF ARCHITECTURAL DEPTH

The architectural ablation in Table 7 shows that increasing depth by adding stacks raised parameter counts and training times dramatically, but produced no meaningful accuracy gains. Moving from two stacks to thirty increased parameters from 219K to 17.3M and training time from 181 s to 2446 s, yet only reduced RMSE from 0.065 to 0.059 (a marginal 9% improvement). This suggests diminishing returns beyond two stacks, where extra layers primarily add computational burden without boosting generalisation. Deeper models also risk overfitting on limited seasonal datasets, which can harm out-of-sample accuracy.

Interestingly, shallow configurations such as 2 stacks × 1 block × 4 FC layers (219K params) delivered accuracy competitive with very deep designs, showing that N-BEATS does not rely on depth to achieve strong performance. Instead, its block structure and trend-seasonality decomposition allow shallow networks to represent complex temporal dynamics effectively. Excessive depth increased gradient path length, slowing convergence and demanding more VRAM (685 MB at 30 stacks). These results indicate that moderate depth strikes the best balance between accuracy and deployability, enabling faster training and lower memory use while maintaining forecasting precision.

3) EFFECT OF TREND POLYNOMIAL DEGREE

As shown in Table 8, a polynomial degree of two achieved the lowest error values (RMSE 0.044), outperforming degree 1 (0.089) and degree 3 (0.067). Higher degrees ≥ 4 worsened performance, raising both RMSE and training time, likely due to overfitting. This suggests that a low-order polynomial basis is sufficient to capture the smooth diurnal shape of the power curve without introducing unnecessary complexity. The results also align with the theoretical premise of N-BEATS's interpretable block, where overparameterisation in the trend basis can lead to noise-fitting.

Moreover, the degree-2 configuration maintained low parameter counts (210K) and training time (181 s), making it the most efficient choice. Degree-1 models were simpler but failed to represent curvature in the daily profile, while degree-3 and degree-4 models added complexity without accuracy gains. These findings confirm that model expressiveness is best improved through architectural refinement, such as activation choice and stack design, rather than by increasing polynomial order, which inflates hypothesis space and raises overfitting risk on finite data.

VI. DISCUSSION

The ablation results in Table 9 and Table 10 show that compact G-NBEATS and I-NBEATS configurations outperform their untuned baselines while using far fewer computational resources. The compact G-NBEATS variant

TABLE 7. Ablation of N-BEATS architectural settings (Stack/Block/FC configuration). Metrics are RMSE/MSE/MAE/R^2 in normalized units and MW; lower is better for errors, higher is better for R^2.

Architecture				RMSE/MSE/MAE/R^2		Params	Train time (s)	VRAM Peak (MB)
Stack	Block	Fc Layer	Layer Width	Normalized	MW			
1	1	4	256	0.075/0.006/0.037/0.956	0.387/0.149/0.191/0.956	567K	262.65	39.28
1	2	4	256	0.054/0.003/0.030/0.977	0.279/0.078/0.152/0.977	1.1M	306.26	62.98
1	2	4	64	0.064/0.004/0.038/0.969	0.326/0.106/0.195/0.969	219K	324.08	26.29
2	**1**	**4**	**64**	**0.065/0.004/0.035/0.967**	**0.334/0.112/0.178/0.967**	**219K**	**181.01**	**27.59**
2	2	3	128	0.072/0.005/0.042/0.960	0.367/0.134/0.213/0.960	900K	405.33	53.04
2	2	4	256	0.096/0.009/0.061/0.929	0.490/0.240/0.314/0.929	2.3M	455.75	106.74
3	3	4	256	0.064/0.004/0.031/0.968	0.327/0.107/0.161/0.968	5.2M	833.67	218.22
4	2	3	256	0.067/0.004/0.044/0.965	0.341/0.116/0.227/0.965	4.1M	698.71	175.79
4	3	4	512	0.092/0.008/0.071/0.935	0.469/0.220/0.363/0.935	18.5M	1295.05	742.76
30	1	4	256	0.059/0.003/0.035/0.973	0.301/0.090/0.180/0.973	17.3M	2446.05	685.15

TABLE 8. Performance of I-NBEATS with varying polynomial degrees. Metrics shown are RMSE/MSE/MAE/R^2 in normalized units and megawatts (MW).

Polynomial Degree	RMSE/MSE/MAE/R^2		Params	Train time (s)	VRAM Peak (MB)
	Normalized	MW			
1	0.089/0.008/0.047/0.939	0.455/0.207/0.239/0.939	210 K	177.27	44.01
2	**0.044/0.002/0.027/0.985**	**0.228/0.052/0.136/0.985**	**210 K**	**181.19**	**44.03**
3	0.067/0.004/0.038/0.966	0.341/0.116/0.196/0.966	210 K	176.09	44.05
4	0.067/0.004/0.045/0.965	0.341/0.117/0.229/0.965	210 K	178.14	44.06

TABLE 9. Benchmark of G-NBEATS variants. Metrics are RMSE/MSE/MAE/R^2 in normalized units and MW. Bold indicates per-column best (min error metrics, max R^2; min Params, Train time, VRAM).

Model	RMSE/MSE/MAE/R^2		Params	Train time (s)	VRAM Peak (MB)
	Normalized	MW			
G-NBEATS without tuning	0.059/0.003/0.035/0.973	0.301/0.090/0.180/0.973	17.3 M	2446.05	685.15
G-NBEATS + fewer layer	0.065/0.004/0.035/0.967	0.334/0.112/0.178/0.967	219 K	181.01	27.59
G-NBEATS + fewer layer + GELU	0.046/0.002/0.026/0.984	0.235/0.055/0.131/0.984	219 K	170.46	27.38
G-NBEATS + fewer layer + GELU + Dropout 0.05	**0.042/0.002/0.022/0.987**	**0.213/0.045/0.114/0.987**	**219 K**	182.92	31.14

TABLE 10. Benchmark of I-NBEATS variants. Metrics are RMSE/MSE/MAE/R^2 in normalized units and MW. Bold indicates per-column best (min error metrics, max R^2; min Params, Train time, VRAM).

Model	RMSE/MSE/MAE/R^2		Params	Train time (s)	VRAM Peak (MB)
	Normalized	MW			
I-NBEATS without tuning	0.061/0.004/0.034/0.971	0.314/0.099/0.174/0.971	3.6 M	311.42	80.12
I-NBEATS + fewer layer	0.070/0.005/0.037/0.962	0.357/0.127/0.189/0.962	**210 K**	**179.72**	**43.53**
I-NBEATS + fewer layer + RReLU	**0.044/0.002/0.027/0.985**	**0.228/0.052/0.136/0.985**	**210 K**	181.19	44.03

reduced the number of parameters by approximately ninety-nine percent, shortened training time by more than ninety percent, and reduced memory usage by over ninety-five percent, while improving accuracy. The RReLU-based I-NBEATS configuration reduced parameters by more than ninety-four percent, cut training time by forty-two percent, and halved memory consumption, while also lowering RMSE compared to the original 3.6 million-parameter baseline.

These results highlight three main factors behind the accuracy improvements. First, the choice of activation function strongly influenced performance. Second, a moderate depth of two stacks was more effective than very deep configurations, which were expensive and delivered minimal gains. Third, a low-order trend polynomial, particularly degree two, provided the best accuracy while avoiding overfitting.

Qualitative patterns in Figure 6 are consistent with these findings. G-NBEATS and N-HiTS accurately followed both the sharp morning ramp and post-noon decay, while TCN and LSTM lagged during the afternoon, and DLinear showed negative drift at night. These results confirm that compact N-BEATS variants offer superior accuracy at a fraction of the computational cost, making them well-suited for operational photovoltaic power forecasting systems. Transformer can match or exceed its performance in very

stable summer conditions, but its heavy resource demands limit its practicality.

From a theoretical standpoint, the proposed N-BEATS formulation aligns with the physical basis of photovoltaic generation. By minimizing the expected squared error (13), the model learns a nonlinear function f_θ that approximates the irradiance–power conversion dynamics described by (9). This hybrid perspective ensures that the proposed deep learning model is not purely empirical but grounded in an interpretable energy-generation process. Hence, the architecture maintains both mathematical rigor and physical relevance, addressing the limitations noted in previous machine-learning-only approaches.

VII. CONCLUSION AND FUTURE WORK

This study demonstrates the substantial potential of deep learning architectures for short-term photovoltaic (PV) power forecasting, with particular emphasis on enhancing accuracy while reducing computational overhead. Through a comprehensive benchmarking of eight representative models, G-NBEATS and I-NBEATS emerged as the most effective architectures, outperforming widely adopted baselines such as Transformer, TFT, and LSTM. Systematic ablation experiments revealed that model efficiency and generalization can be significantly improved through targeted architectural refinements, including shallow two-stack designs, modern activation functions (GELU and RReLU), and low-order trend polynomials. These compact variants achieved markedly higher predictive performance, reaching R^2 scores of 98.7% (G-NBEATS) and 98.5% (I-NBEATS), while simultaneously reducing parameter counts by up to 98% (compared to the heaviest baseline, NHiTS with 3.1M parameters), training times by over 90% (compared to Transformer and TFT exceeding 2300 s), and memory usage by more than 95% (compared to Transformer with 997.67 MB VRAM). This offers a favourable balance between predictive accuracy and computational efficiency.

The findings highlight that model depth and size alone do not guarantee forecasting accuracy. Rather, the careful configuration of architectural elements can yield lightweight models capable of capturing the complex, nonlinear patterns inherent in solar generation data. Qualitative analyses further showed that compact N-BEATS models accurately reproduced both diurnal ramps and seasonal transitions, while conventional architectures often underestimated peaks or exhibited drift. This work thus establishes a methodological framework for building deployment-ready PV forecasting models that combine high precision with operational practicality, making them particularly suitable for resource-constrained environments.

Despite these promising results, several limitations warrant attention. First, the study was based on a single geographical site and used a single dataset, which may constrain the generalizability of the models to regions with different climatic conditions or irradiance patterns. Second, the feature set was deliberately restricted to core weather and generation variables. Incorporating exogenous covariates such as cloud imagery, satellite data, or other features in numerical weather predictions could further improve performance. Third, the experiments focused on point forecasts, without explicitly modeling uncertainty, which are critical for operational risk-aware grid management. Finally, explainability analyses of model decisions were limited, leaving the internal attribution of seasonal and trend components largely unexamined.

Future work should address these gaps by extending the evaluation of compact N-BEATS architectures to multi-site and cross-climate datasets to test their robustness under diverse conditions. Incorporating richer meteorological covariates and ensemble weather forecasts can enhance short-term responsiveness, while adding uncertainty quantification modules would enable risk-informed decision making. Furthermore, integrating explainable AI techniques, such as SHAP-based feature attribution or temporal saliency maps, could improve the interpretability of model outputs and foster trust in operational deployment. These directions will build on the foundations established here, advancing towards accurate, efficient, and explainable deep learning solutions for large-scale renewable energy integration.

REFERENCES

[1] J. Herzen, F. Lässig, S. G. Piazzetta, T. Neuer, L. Tafti, G. Raille, T. Van Pottelbergh, M. Pasieka, A. Skrodzki, N. Huguenin, M. Dumonal, J. Koscisz, D. Bader, F. Gusset, M. Benheddi, C. Williamson, M. Kosinski, M. Petrik, and G. Grosch, "Darts: User-friendly modern machine learning for time series," *J. Mach. Learn. Res.*, vol. 23, no. 124, pp. 1–6, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1177.html

[2] Y. Dou, S. Tan, and D. Xie, "Comparison of machine learning and statistical methods in the field of renewable energy power generation forecasting: A mini review," *Frontiers Energy Res.*, vol. 11, Jul. 2023, Art. no. 1218603, doi: 10.3389/fenrg.2023.1218603.

[3] M. Abdelsattar, M. A. Ismeil, M. M. A. A. Zayed, A. Abdelmoety, and A. Emad-Eldeen, "Assessing machine learning approaches for photovoltaic energy prediction in sustainable energy systems," *IEEE Access*, vol. 12, pp. 107599–107615, 2024, doi: 10.1109/ACCESS.2024.3437191.

[4] K. B. Prakash, M. K. Pasupathi, S. Chinnasamy, S. Saravanakumar, M. Palaniappan, A. Alasiri, and M. Chandrasekaran, "Energy and exergy enhancement study on PV systems with phase change material," *Sustainability*, vol. 15, no. 4, p. 3627, Feb. 2023, doi: 10.3390/su15043627.

[5] N. P. Singh and M. N. Alam, "Short-term forecasting in smart grid environment using N-BEATS," in *Research Square Platform LLC*, Durham, NC, USA, Mar. 2024, doi: 10.21203/rs.3.rs-4116626/v1.

[6] K. G. Olivares, C. Challu, G. Marcjasz, R. Weron, and A. Dubrawski, "Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx," *Int. J. Forecasting*, vol. 39, no. 2, pp. 884–900, Apr. 2023, doi: 10.1016/j.ijforecast.2022.03.001.

[7] B. N. Oreshkin, G. Dudek, P. Pełka, and E. Turkina, "N-BEATS neural network for mid-term electricity load forecasting," *Appl. Energy*, vol. 293, Jul. 2021, Art. no. 116918, doi: 10.1016/j.apenergy.2021.116918.

[8] M. Kasprzyk, P. Pełka, B. N. Oreshkin, and G. Dudek, "Enhanced N-BEATS for mid-term electricity demand forecasting," *Appl. Soft Comput.*, vol. 182, Oct. 2025, Art. no. 113575, doi: 10.1016/j.asoc.2025.113575.

[9] A. K. Shaikh, A. Nazir, I. Khan, and A. S. Shah, "Short term energy consumption forecasting using neural basis expansion analysis for interpretable time series," *Sci. Rep.*, vol. 12, no. 1, p. 22562, Dec. 2022, doi: 10.1038/s41598-022-26499-y.

[10] C.-H. Loh, Y.-C. Chen, C.-T. Su, and H.-Y. Su, "Establishing lightweight and robust prediction models for solar power forecasting using numerical–categorical radial basis function deep neural networks," *Appl. Sci.*, vol. 14, no. 22, p. 10625, Nov. 2024, doi: 10.3390/app142210625.

[11] Y. Wang, Y. Qiu, P. Chen, Y. Shu, Z. Rao, L. Pan, B. Yang, and C. Guo, "LightGTS: A lightweight general time series forecasting model," 2025, *arXiv:2506.06005*.

[12] H. Ritchie, P. Rosado, and M. Roser, "Energy production and consumption," *Our World Data*, 2020. [Online]. Available: https://ourworldindata.org/energy-production-consumption

[13] K. N. Nwaigwe, P. Mutabilwa, and E. Dintwa, "An overview of solar power (PV systems) integration into electricity grids," *Mater. Sci. Energy Technol.*, vol. 2, no. 3, pp. 629–633, Dec. 2019, doi: 10.1016/j.mset.2019.07.002.

[14] I. M. Peters, C. D. Rodriguez Gallegos, S. E. Sofia, and T. Buonassisi, "The value of efficiency in photovoltaics," *Joule*, vol. 3, no. 11, pp. 2732–2747, Nov. 2019, doi: 10.1016/j.joule.2019.07.028.

[15] M. N. Akhter, S. Mekhilef, H. Mokhlis, Z. M. Almohaimeed, M. A. Muhammad, A. S. M. Khairuddin, R. Akram, and M. M. Hussain, "An hour-ahead PV power forecasting method based on an RNN-LSTM model for three different PV plants," *Energies*, vol. 15, no. 6, p. 2243, Mar. 2022, doi: 10.3390/en15062243.

[16] M. N. Akhter, S. Mekhilef, H. Mokhlis, and N. Mohamed Shah, "Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques," *IET Renew. Power Gener.*, vol. 13, no. 7, pp. 1009–1023, May 2019, doi: 10.1049/iet-rpg.2018.5649.

[17] H. Akoglu, "User's guide to correlation coefficients," *Turkish J. Emergency Med.*, vol. 18, no. 3, pp. 91–93, Sep. 2018, doi: 10.1016/j.tjem.2018.08.001.

[18] E. M. Al-Ali, Y. Hajji, Y. Said, M. Hleili, A. M. Alanzi, A. H. Laatar, and M. Atri, "Solar energy production forecasting based on a hybrid CNN-LSTM-transformer model," *Mathematics*, vol. 11, no. 3, p. 676, Jan. 2023, doi: 10.3390/math11030676.

[19] H. Alharkan, S. Habib, and M. Islam, "Solar power prediction using dual stream CNN-LSTM architecture," *Sensors*, vol. 23, no. 2, p. 945, Jan. 2023, doi: 10.3390/s23020945.

[20] T. M. Al-Jaafreh, A. Al-Odienat, and Y. A. Altaharwah, "The solar energy forecasting using LSTM deep learning technique," in *Proc. Int. Conf. Emerg. Trends Comput. Eng. Appl. (ETCEA)*, Nov. 2022, pp. 1–6, doi: 10.1109/ETCEA57049.2022.10009717.

[21] S. Almaghrabi, M. Rana, M. Hamilton, and M. Saiedur Rahaman, "Multivariate solar power time series forecasting using multilevel data fusion and deep neural networks," *Inf. Fusion*, vol. 104, Apr. 2024, Art. no. 102180, doi: 10.1016/j.inffus.2023.102180.

[22] H. I. Aouidad and A. Bouhelal, "Machine learning-based short-term solar power forecasting: A comparison between regression and classification approaches using extensive Australian dataset," *Sustain. Energy Res.*, vol. 11, no. 1, p. 28, Aug. 2024, doi: 10.1186/s40807-024-00115-1.

[23] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.

[24] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Multi-sequence LSTM-RNN deep learning and metaheuristics for electric load forecasting," *Energies*, vol. 13, no. 2, p. 391, Jan. 2020, doi: 10.3390/en13020391.

[25] U. K. Das, K. S. Tey, M. Seyedmahmoudian, S. Mekhilef, M. Y. I. Idris, W. Van Deventer, B. Horan, and A. Stojcevski, "Forecasting of photovoltaic power generation and model optimization: A review," *Renew. Sustain. Energy Rev.*, vol. 81, pp. 912–928, Jan. 2018, doi: 10.1016/j.rser.2017.08.017.

[26] L. Fara, A. Diaconu, D. Craciunescu, and S. Fara, "Forecasting of energy production for photovoltaic systems based on ARIMA and ANN advanced models," *Int. J. Photoenergy*, vol. 2021, no. 1, 2021, Art. no. 6777488, doi: 10.1155/2021/6777488.

[27] G. Fungtammasan and I. Koprinska, "Convolutional and LSTM neural networks for solar power forecasting," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2023, pp. 1–7, doi: 10.1109/IJCNN54540.2023.10191813.

[28] G. A. Galindo Padilha, J. Ko, J. J. Jung, and P. S. G. de Mattos Neto, "Transformer-based hybrid forecasting model for multivariate renewable energy," *Appl. Sci.*, vol. 12, no. 21, p. 10985, Oct. 2022, doi: 10.3390/app122110985.

[29] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting—An approach using AutoEncoder and LSTM neural networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Budapest, Hungary, Oct. 2016, pp. 002858–002865, doi: 10.1109/SMC.2016.7844673.

[30] M. S. Hossain and H. Mahmood, "Short-term photovoltaic power forecasting using an LSTM neural network," in *Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT)*, Washington, DC, USA, Feb. 2020, pp. 1–5, doi: 10.1109/ISGT45199.2020.9087786.

[31] International Energy Agency (IEA). (2023). *Renewables 2023—Executive Summary*. Paris, France. [Online]. Available: https://www.iea.org/reports/renewables-2023/executive-summary

[32] K. J. Iheanetu, "Solar photovoltaic power forecasting: A review," *Sustainability*, vol. 14, no. 24, p. 17005, Dec. 2022, doi: 10.3390/su142417005.

[33] N. Kim, H. Lee, J. Lee, and B. Lee, "Transformer based prediction method for solar power generation data," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2021, pp. 7–9, doi: 10.1109/ICTC52510.2021.9620897.

[34] Y. Li, L. Song, S. Zhang, L. Kraus, T. Adcox, R. Willardson, A. Komandur, and N. Lu, "A TCN-based hybrid forecasting framework for hours-ahead utility-scale PV forecasting," *IEEE Trans. Smart Grid*, vol. 14, no. 5, pp. 4073–4085, Sep. 2023, doi: 10.1109/TSG.2023.3236992.

[35] B. Lim, S. Ö. Arik, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *Int. J. Forecasting*, vol. 37, no. 4, pp. 1748–1764, Oct. 2021, doi: 10.1016/j.ijforecast.2021.03.012.

[36] X. Luo, D. Zhang, and X. Zhu, "Deep learning based forecasting of photovoltaic power generation by incorporating domain knowledge," *Energy*, vol. 225, Jun. 2021, Art. no. 120240, doi: 10.1016/j.energy.2021.120240.

[37] M. López Santos, X. García-Santiago, F. Echevarría Camarero, G. Blázquez Gil, and P. Carrasco Ortega, "Application of temporal fusion transformer for day-ahead PV power forecasting," *Energies*, vol. 15, no. 14, p. 5232, Jul. 2022, doi: 10.3390/en15145232.

[38] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," 2019, *arXiv:1905.10437*.

[39] Q.-T. Phan, Y.-K. Wu, and Q.-D. Phan, "Application of a new transformer-based model and XGBoost to improve one-day-ahead solar power forecasts," in *Proc. IEEE/IAS 59th Ind. Commercial Power Syst. Tech. Conf. (I&CPS)*, New Delhi, India, May 2023, pp. 1–7, doi: 10.1109/ICPS57144.2023.10142107.

[40] F. Tian, X. Fan, R. Wang, H. Qin, and Y. Fan, "A power forecasting method for ultra-short-term photovoltaic power generation using transformer model," *Math. Problems Eng.*, vol. 2022, Oct. 2022, Art. no. 9421400, doi: 10.1155/2022/9421400.

[41] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.

[43] T. Yao, J. Wang, H. Wu, P. Zhang, S. Li, Y. Wang, X. Chi, and M. Shi, "A photovoltaic power output dataset: Multi-source photovoltaic power output dataset with Python toolkit," *Sol. Energy*, vol. 230, pp. 122–130, Dec. 2021, doi: 10.1016/j.solener.2021.09.050.

[44] J. Zhang, A. Florita, B.-M. Hodge, S. Lu, H. F. Hamann, V. Banunarayanan, and A. M. Brockway, "A suite of metrics for assessing the performance of solar power forecasting," *Sol. Energy*, vol. 111, pp. 157–175, Jan. 2015, doi: 10.1016/j.solener.2014.10.016.

NURUL JANNAH (Student Member, IEEE) received the Bachelor of Engineering (S.T.) degree in telecommunication engineering from Telkom University, Indonesia, in 2022, and the M.Sc. (Eng) degree in computer and information engineering from the International Islamic University Malaysia (IIUM), in 2025. From 2023 to 2024, she was a Graduate Research Assistant with the EMS Research Group, International Islamic University Malaysia (IIUM). From 2021 to 2022, she conducted research in machine learning and computer vision with the Image Processing and Vision Laboratory, Telkom University, Indonesia. Her research interests include computer vision, solar energy, machine learning, and time-series forecasting.

TEDDY SURYA GUNAWAN (Senior Member, IEEE) received the B.Eng. degree (cum laude) in electrical engineering from the Institut Teknologi Bandung (ITB), Indonesia, in 1998, the M.Eng. degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 2001, and the Ph.D. degree from the School of Electrical Engineering and Telecommunications, The University of New South Wales, Australia, in 2007. He was the Head of Department with the Department of Electrical and Computer Engineering, from 2015-2016, was a Visiting Research Fellow with UNSW, form 2010 to 2021, and an Adjunct Professor with Telkom University, from 2022 to 2023. He has been a Professor, since 2019. His research interests include speech and audio processing, biomedical signal processing and instrumentation, image and video processing, and parallel computing. He was awarded the Best Researcher Award from IIUM, in 2018. He was the Chairperson of the IEEE Instrumentation and Measurement Society–Malaysia Section, in 2013, 2014, 2020, and 2021, and the Head of Programme Accreditation, and the Quality Assurance for Faculty of Engineering, International Islamic University Malaysia. He has been a Chartered Engineer (IET, U.K.), since 2016 and Insinyur Profesional Utama (PII, Indonesia) since 2021, a registered ASEAN Engineer, since 2018, and an ASEAN Chartered Professional Engineer, since 2020.

SITI HAJAR YUSOFF (Member, IEEE) is a former student of Kolej Yayasan UEM (KYUEM), Lembah Beringin. She received the M.Eng. degree (Hons.) and Ph.D. degree in electrical and electronic engineering from the University of Nottingham, U.K., in 2009 and 2014, respectively.

She is currently an Associate Professor (Ts. Dr.) with the Department of Electrical and Computer Engineering, Kulliyyah of Engineering, International Islamic University Malaysia (IIUM), Gombak. In addition, she was the Deputy Director with the International Affairs Office, IIUM. Her areas of specialization include power electronics, nonlinear control systems, and intelligent energy systems. Her research interests include energy management systems, power converters, renewable energy storage, microgrid integration, and the application of artificial intelligence in smart grid, and sustainable energy technologies.

Dr. Yusoff has received several recognitions for her research contributions and academic excellence, including 2nd runner-up in the New Energy category at the CHESS Innovation Challenge 2024 organized by PETRONAS Sdn Bhd, and the Best Paper Award at the International Conference on Engineering Technology (ICET), in 2023.

MOHD SHAHRIN ABU HANIFAH (Member, IEEE) received the Master of Engineering and Doctor of Engineering degrees from Tokai University, Kanagawa, Japan, in 2012 and 2016, respectively. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, International Islamic University Malaysia, Gombak. He is teaching programming for engineers and engineering electromagnetics courses. His research interests include power distribution network reconfiguration, service restoration optimization using a multi-objectives algorithm, electric vehicle charging station integration in the distribution networks, and application of distributed generation and renewable energy in microgrid.

SITI NADIAH MOHD SAPIHIE received the M.Sc. degree in advanced chemical engineering from the University of Birmingham. Her studies concentrated on electrochemistry, fuel cell & hydrogen technology with the University of Birmingham. She holds the position of an Executive (Renewable Energy) within the Group Research & Technology Division. She entered the Oil & Gas sector, in 2018, and currently a Researcher with the Renewable Energy Program, Petronas. Her primary focus revolves around integrated energy management systems, encompassing solar, wind and energy storage systems. Throughout her research journey, she has been actively engaged in the development of predictive tools employing machine learning/artificial intelligence technology to address various business challenges within Petronas. Notably, these technological solutions have been effectively implemented in Petronas refineries, LNG assets, and petrochemical plants.

HAMZAH ADLAN received the bachelor's degree in mechanical engineering from Toyama University, in Japan, and the Master of Science degree in mechanical from the University Putra Malaysia. He holds the position of Principal Scientist with PETRONAS Research Sdn. Bhd. He entered the Oil & Gas sector, in 2008, following a background in automotive engine development. He has been a Professional Engineer, since 2010, and has been involved in engineering and research for the past 25 years. His research interests include energy, mobility, electrification, and motorsports.