Proc. of Int. Conf. on Math.Adv. and Appl., 2(1) (2025) 113-121



Proceedings of
International Conference on
Mathematical Advances and Applications

ISBN 978-605-69387-8-8

Enhancing Hourly Photovoltaic Power Forecasting with a Hybrid Statistical-Deep Learning Model

Musa TERKEŞ[1], Alpaslan DEMİRCİ[1] and Erdin GÖKALP[1]

[1]*Department of Electrical Engineering, Faculty of Electrical and Electronics, Yıldız Technical University, İstanbul, Türkiye*

Article Info

Keywords: *XGBoost, LSTM, Ensemble learning, Machine learning, PV forecasting.*
2010 AMS: *xxxxx, xxxxx*
Received: 17.06.2025
Accepted: 20.10.2025
Available online: 22.10.2025

Abstract

In this study, a hybrid ensemble machine learning model integrating statistical learning and time series modeling is proposed for hourly forecasting of power generation from photovoltaic (PV) systems. The model architecture combines the high-accuracy regression capabilities of the Extreme Gradient Boosting (XGBoost) algorithm with the temporal dependency learning capacity of Long Short-Term Memory (LSTM) networks. The final prediction is obtained by dynamically aggregating the outputs of both models using a weighted averaging technique. The modeling process utilizes hourly meteorological variables collected over one year, including solar irradiance, ambient temperature, relative humidity, and cloud cover. In addition, domain-specific feature engineering based on solar physics is applied to enhance the explanatory power of the input variables. Hyperparameter optimization for both XGBoost and LSTM models is conducted using a parametric grid search algorithm and cross-validation techniques. The performance of the model is evaluated using statistical metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R^2). The hybrid model demonstrated a high prediction accuracy with an RMSE of 0.0057 and an R^2 of 0.9994. For comparative analysis, conventional methods, including Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Random Forest (RF), and ARIMA, were also implemented, and the proposed hybrid approach outperformed all these traditional techniques. The results obtained indicate that statistical learning-based hybrid models possess significant potential for smart decision support systems in distributed energy applications and contribute to the more efficient and predictable management of renewable energy resources.

1. Introduction

1.1. Motivation and Challenges in PV Power Forecasting

Photovoltaic (PV) power forecasting is inherently complex due to the stochastic and nonlinear nature of solar energy generation, which is strongly influenced by meteorological parameters such as solar irradiance, temperature, humidity, and cloud cover. Traditional time series models often assume stationarity and linear relationships, which limits their ability to capture the irregular patterns and rapid fluctuations in PV output. Artificial intelligence (AI)-based models, particularly those leveraging deep learning and machine learning techniques, have become prominent due to their ability to learn temporal dependencies and complex nonlinear interactions directly from historical data. However, standalone AI models may still face challenges such as overfitting, lack of interpretability, and limited generalization when trained on noisy or insufficiently representative datasets.

Email addresses and ORCID numbers: musa.terkes@yildiz.edu.tr, 0000-0002-4411-411X (M.Terkes), ademirci@yildiz.edu.tr, 0000-0002-1038-7224 (A.Demirci), gokalp@yildiz.edu.tr, 0000-0002-1518-432X (E.Gökalp)
Cite as: M. Terkes, A. Demirci, E. Gökalp, *Enhancing Hourly Photovoltaic Power Forecasting with a Hybrid Statistical-Deep Learning Model*, Proc. of Int. Conf. on Math. Adv. and Appl., 2(1) (2025), 113-121.

To overcome these limitations, hybrid modeling strategies have gained attention by combining the strengths of different algorithmic structures. For instance, deep recurrent neural networks like long short-term memory (LSTM) are capable of capturing temporal dynamics, while ensemble learning models like extreme gradient boosting (XGBoost) are effective in handling high-dimensional inputs and mitigating overfitting through regularization. When integrated in a weighted or stacked architecture, these hybrid approaches not only enhance predictive accuracy but also offer robustness against data variability and uncertainty. Moreover, the incorporation of domain knowledge into feature engineering, optimization algorithms for hyperparameter tuning, and attention mechanisms for focusing on key input patterns further solidify the technical relevance of hybrid models in PV forecasting tasks.

1.2. Evolution of Forecasting Models: From Conventional to Hybrid Approaches

Early approaches to PV power forecasting predominantly relied on statistical and physical models such as autoregressive integrated moving average (ARIMA), seasonal decomposition methods, or numerical weather prediction (NWP)-based regressions. While effective in capturing linear and periodic behaviors, these methods often struggle with nonlinearity, data noise, and structural changes in PV output influenced by dynamic atmospheric conditions. Their assumptions of stationarity and fixed input-output mappings limit their capacity to adapt to abrupt shifts in solar patterns or rare meteorological events.

With advances in machine learning and deep learning, forecasting paradigms have gradually shifted toward data-driven approaches capable of modeling complex nonlinearities. Models such as support vector machines (SVM), k-nearest neighbors (k-NN), and deep architectures like LSTM, gated recurrent unit (GRU), or convolutional neural network (CNN) have shown superior performance in short-term prediction tasks. However, no single model can capture every aspect of the forecasting challenge. This has led to the emergence of hybrid and ensemble approaches, such as weighted average models, stacking, and boosting, which integrate complementary strengths of individual algorithms. These hybrid frameworks enhance generalization, improve temporal-spatial feature extraction, and mitigate the risk of overfitting, thereby offering a more robust solution to the volatile and multivariate nature of PV power generation.

1.3. Hybrid AI Architectures and Recent Deep Ensemble Advances for Nonlinear PV Forecasting

In one study, grey relational analysis (GRA) was employed for factor selection, while the firefly algorithm (FA) was utilized to enhance model generalizability and stability [1]. Furthermore, a stacked denoising autoencoder (SDA) was combined with an extreme learning machine (ELM) to improve forecasting accuracy. Similarly, another advanced framework integrated the iTransformer, long short-term memory (LSTM), cross-attention, and capsule networks (CapsNet), where the iTransformer encoded the structure of the target variable, LSTM captured the temporal dependencies of the input features, and the fused representations were passed through CapsNet to yield seasonally adaptive forecasts [2].

On the other hand, in line with these hybrid strategies, a combination of bidirectional GRU (BiGRU) and XGBoost was proposed to extract time-series features and select meteorological variables. In this study, the weights of both models were optimized using the Gaining Sharing Knowledge (GSK) algorithm [3]. Likewise, to better capture the nonlinear dynamics of PV power, convolutional neural networks (CNN) were fused with GRU networks, enabling the joint extraction of spatial and temporal patterns [4]. Moreover, in ultra-short-term forecasting tasks, volatility and nonlinearity were addressed through wavelet decomposition, while a dual attention mechanism highlighted key time points and features. Bidirectional LSTM further enhanced temporal learning by considering both forward and backward dependencies [5]. In addition, due to the stochastic nature of PV generation, a deep neural network model based on quantile regression and kernel density estimation (QRKDNN) was developed. This approach improved not only point forecasts but also interval and probabilistic predictions [6]. Meteorological features were selected using the Pearson correlation coefficient, and similar-day clustering was performed via Gaussian mixture models (GMM). Forecasting was then conducted using a CNN–BiGRU–Attention hybrid model.

In another study, a radial basis function (RBF) network trained with the whale optimization algorithm (WOA) was embedded in an autoencoder framework to effectively manage uncertainty in forecasting [7]. Furthermore, a simple yet efficient hybrid of support vector regression (SVR) and artificial neural networks (ANN) was enhanced using several metaheuristic algorithms, including MOA, SSO, PSO, GSO, and NNA [8]. In a distinct study, CNN was used to extract high-level features from input data, which were then utilized by XGBoost to improve forecasting accuracy in time series [9].

In further research, to further improve model robustness, fuzzy C-means (FCM) clustering was used to distinguish between high- and low-variability samples during preprocessing. Within this framework, a stacking ensemble consisting of LSTM and LightGBM effectively captured the PV power mechanism [10]. Additionally, spatial and temporal features were learned via CNN and LSTM layers, respectively, and integrated within an autoencoder network, resulting in RMSE and MAE improvements of approximately 5–25% [11]. While deeper CNN–LSTM models increased accuracy under variable weather conditions, especially during rainy and cloudy periods, they also led to higher computational complexity [12].

An alternative investigation revealed that, to further enhance model performance, Transformer architectures were incorporated into CNN–LSTM hybrids. Here, spatial features were extracted with 1D-CNN, temporal dependencies with LSTM, and global relationships were modeled using Transformer blocks [13]. Moreover, enriching model inputs with meteorological variables such as irradiance, temperature, wind speed, and humidity proved essential. In an alternative two-stage model, LSTM was used to forecast direct irradiance, followed by XGBoost for PV power prediction [14]. However, in cases where sensor infrastructure is lacking, forecasting can be performed based solely on historical production data. In such scenarios, CNN extracted representative patterns affected by weather, while LSTM generated separate forecasts for sunny and cloudy days [15]. To maintain model stability during volatile periods, attention mechanisms were introduced and hyperparameters were fine-tuned using Bayesian optimization with a tree-structured Parzen estimator (TPE) [16]. Moreover, including facility power consumption along with meteorological variables improved the model's ability to respond to both external and internal factors, while analyses using 'look-back' and 'look-forward' windows helped determine the optimal historical input length and forecast horizon [17].

Beyond these, hybrid models combining LSTM with recurrent neural networks (RNN) were developed to capture periodic fluctuations in PV generation. In this setup, time correlation adjustment (TCA) was applied based on patterns from previous years and days [18]. For improved robustness, a separate model was proposed to forecast plane-of-array (POA) irradiance and ambient temperature, with optimal ANN weights obtained through backpropagation and L2 regularization [19].

In further research, in day-ahead forecasting, a BiLSTM–GBDT hybrid model was implemented using the teacher forcing mechanism. BiLSTM captured complex temporal patterns, while GBDT served as the boosting framework, with Adam optimizer used to enhance model training [20]. Additionally, a novel stacking ensemble approach used random forest, XGBoost, and AdaBoost as base learners, with the stack extra trees regressor (ETR) as the meta-learner. Notably, this method achieved competitive performance without the need for hyperparameter tuning [21]. Finally, recent advances have introduced quantum-classical hybrid architectures. In particular, hybrid quantum LSTM (HQLSTM) significantly outperformed conventional LSTM by up to 40%, while HQSeq2Seq models demonstrated robust forecasting capabilities across arbitrary time windows, even without weather data [22].

1.4. Paper Contributions and Research Objectives

The primary objectives of this study are as follows:

- To systematically evaluate the forecasting performance of various machine learning and statistical models on hourly PV power generation data.
- To develop a hybrid weighted average ensemble model that combines LSTM and XGBoost to enhance forecasting accuracy and stability.
- To perform comprehensive hyperparameter optimization for all models to ensure fair performance comparisons.
- To provide practical insights for researchers and practitioners regarding model selection and performance under varying environmental conditions.

The main contributions to this paper are listed below:

- A rigorous comparative analysis of seven prominent forecasting models (XGBoost, LSTM, SVR, k-NN, Random Forest, SARIMA, and a novel LSTM-XGBoost ensemble) using real-world PV generation data at an hourly resolution.
- Implementation of a weighted average ensemble method that demonstrates the significant performance improvement achieved by combining deep learning and tree-based models.
- Comprehensive hyperparameter optimization and detailed evaluation using error metrics.
- A valuable reference framework for future studies and real-world PV forecasting applications, particularly in the contexts of grid integration, renewable energy planning, and market operations.

2. Forecasting Framework and Modeling Approach

2.1. Data Description

The PV production dataset consists of timestamped measurements recorded at hourly intervals over the course of one year. Each record includes the actual PV power output as well as a variety of environmental and astronomical features commonly influencing solar energy generation. The input features include:

- **Meteorological variables:** These include sunshine duration, humidity, precipitation status, cloud cover indicators (e.g., overcast, cloudy), wind speed (at surface and at 10 meters), air temperature at 2 meters, and ambient temperature. Sunshine duration, humidity, precipitation status, and cloud cover data were obtained from the Turkish State Meteorological Service, while surface wind speed and ambient temperature were sourced from NASA's Surface Solar Energy Dataset (NASA POWER). Wind speed at 10 meters and air temperature at 2 meters were retrieved from the Photovoltaic Geographical Information System (PVGIS).
- **Solar irradiance components:** These include global solar irradiance, beam (direct) irradiance on the inclined plane, diffuse irradiance on the inclined plane, reflected irradiance on the inclined plane, PV incident solar irradiance, and extraterrestrial irradiance—all measured in kW/m². The beam, diffuse, and reflected irradiance values on the inclined plane were obtained from PVGIS, whereas global solar irradiance, PV incident solar irradiance, and extraterrestrial irradiance were collected from NASA POWER.
- **Astronomical and positional parameters:** These include solar altitude, sun height, solar azimuth, angle of incidence, declination, and hour angle. Solar altitude, solar azimuth, angle of incidence, declination, and hour angle were retrieved from Sustainable by Design (Susdesign), Seattle, Washington, while sun height data was acquired from PVGIS.
- **Target variable:** The target variable is the actual PV power output (in kW) of a 1 kW photovoltaic system. The PV power generation data was obtained from PVGIS, and a supervised learning approach was adopted for model development.

The dataset was obtained in `.xlsx` format, with timestamps converted into datetime objects and set as the index to facilitate time series operations.

2.2. Data Preprocessing

To ensure consistency and improve model convergence, the following preprocessing steps were applied:

- **Handling of missing data:** Missing values were implicitly addressed prior to modeling. The selected time interval does not contain any missing entries.
- **Normalization and standardization:**
  - For neural network models such as LSTM, all input features were scaled to the range [0, 1] using Min-Max normalization.
  - For traditional machine learning models (e.g., SVM, k-NN, RF), the features were standardized using StandardScaler, transforming the data to have zero mean and unit variance.

To preserve the temporal structure of the data, a chronological split was applied:

- 80% of the data was used for training.
- 20% of the data was reserved for testing.
- For models requiring time-aware cross-validation (e.g., SVM, k-NN, RF), TimeSeriesSplit with 5 folds was employed.

2.3. Forecasting Models

2.3.1. Seasonal ARIMA

The seasonal autoregressive integrated moving average (SARIMA) model is a statistical method that explicitly captures both trend and seasonal components of time series data. Given that PV power generation inherently exhibits daily and yearly recurring seasonal cycles, SARIMA is well-suited for modeling such patterns. By directly representing these seasonal structures mathematically, the model provides consistent and interpretable forecasts, particularly for long-term horizons (e.g., full-year projections of 8,760 hours). SARIMA is defined by the parameters (p, d, q) and (P, D, Q, s), which account for both short-term autocorrelation and long-term seasonality. The model also incorporates lagged effects of error terms, allowing it to assess the influence of past observations on current PV output. This capability enables a statistical interpretation of seasonal effects. However, SARIMA's strength lies primarily in its ability to model linear relationships, and it may struggle to adapt to sudden weather changes or unexpected fluctuations. Despite this limitation, its interpretability and efficiency with limited data make it a reliable choice for classical statistical modeling. In this study, based on the observed hourly seasonality, the SARIMA model was configured with (p=1, d=1, q=1) and seasonal order (P=1, D=0, Q=2, s=24). Forecasting was conducted both on the test dataset and for a full-year horizon of 8,760 hours.

2.3.2. Random Forest (RF)

The RF model is a bagging-based ensemble method in which multiple decision trees operate collectively based on majority voting. In the context of PV power forecasting, the influence of various input variables on power output is complex and multidimensional. RF effectively captures these interactions while mitigating overfitting, thus achieving high prediction accuracy. Each decision tree in the ensemble is trained on a different subset of the data and a random selection of features, which enhances the diversity of the model and contributes to more stable and balanced predictions. Model performance is optimized through the tuning of hyperparameters such as `n_estimators`, `max_depth`, and `min_samples_split`. In systems like PV production that are directly influenced by environmental conditions, RF provides the ability to extract feature importance scores, identifying which meteorological variables are most influential. This interpretability allows RF to serve not only as a predictive tool but also as an analytical framework. Furthermore, the model is well-equipped to handle non-linear relationships and is robust against the effects of outliers.

2.3.3. K-Nearest Neighbors (k-NN)

The k-NN regression algorithm is a pattern-based method that makes predictions by referencing the neighboring data points of a given observation. PV generation data often exhibit recurring patterns depending on the time of day, solar angle, and cloud cover. By leveraging these similarities, k-NN refers to historical instances with comparable conditions to estimate the expected power output. The model identifies the k closest neighbors, as defined by a chosen distance metric (e.g., Euclidean or Manhattan), and predicts the output by averaging the target values of these neighbors. The choice of distance metric directly influences the model's sensitivity. Moreover, applying a weighted approach—where closer neighbors contribute more heavily—can further improve predictive accuracy. Thanks to its simplicity, k-NN is highly interpretable and, when key parameters such as k and distance type are optimally tuned, it can serve as a strong baseline model, especially in scenarios with relatively stable environmental conditions.

2.3.4. Support Vector Regression (SVR)

Support Vector Regression (SVR) is a powerful regression technique capable of modeling complex and nonlinear relationships by projecting the data into a high-dimensional feature space. In photovoltaic (PV) power generation data, numerous meteorological parameters are often interrelated in a nonlinear fashion. SVR captures these relationships using kernel functions, allowing it to tolerate deviations within a specified error margin. Thanks to this property, SVR is insensitive to minor errors while effectively addressing significant deviations. The model employs an ε-insensitive loss function and constructs the regression function solely based on support vectors. This sparse representation reduces computational complexity and mitigates the risk of overfitting. When hyperparameters such as C, ε, and γ are carefully tuned, SVR can produce robust predictions even under short-term fluctuations in PV generation. Additionally, SVR performs well with small datasets, making it advantageous in regions with limited sunshine or a low number of clear-sky days.

2.3.5. LSTM + XGBoost Weighted Average Ensemble

XGBoost is an optimized and regularized implementation of the gradient boosting algorithm. One of its most significant advantages in PV power forecasting is its ability to deliver highly accurate results in high-dimensional datasets while remaining robust against noise. The model trains a sequence of decision trees to iteratively minimize prediction errors, enabling it to effectively capture complex relationships between variables such as cloud cover, solar angle, and temperature. Moreover, through its built-in feature importance metrics, XGBoost allows for the identification of variables that contribute most significantly to PV power generation.

As a tree-based ensemble method, it incorporates regularization techniques (L1 and L2) to prevent overfitting. When its hyperparameters, such as `learning_rate`, `max_depth`, `n_estimators`, and `subsample`, are finely tuned through optimization, XGBoost demonstrates strong generalization capabilities. Additionally, its built-in mechanisms for handling missing values minimize information loss due to incomplete meteorological data. Consequently, XGBoost is a highly suitable and stable model for problems that are sensitive to environmental variability, such as PV power generation.

LSTM is a type of RNN architecture specifically designed to retain information over long temporal intervals. Due to the intrinsic nature of PV

time series data, exhibiting daily, seasonal, and short-term fluctuations, LSTM models are well-suited for capturing patterns in historical solar irradiance and temperature data to predict future outputs. LSTM's internal mechanisms—namely the forget, input, and output gates—enable the model to learn temporal context effectively while maintaining resilience to noisy inputs.

Technically, LSTM layers transform sequential input data into vectorized representations stored in memory cells, allowing them to model the influence of past meteorological conditions, such as cloudiness, on current power production unlike conventional feedforward neural networks. Furthermore, the model can be regularized using techniques such as dropout, and optimized using algorithms like Adam to avoid overfitting. A notable strength of LSTM lies in its capacity to capture nonlinear relationships, enabling it to model the interactive effects of irradiance, temperature, and angular parameters. Therefore, LSTM is particularly effective in systems inherently tied to physical processes, such as solar energy production.

The proposed LSTM-based architecture for PV power prediction consists of two stacked LSTM layers with intermediate dropout regularization, followed by a dense output layer. The model receives a multivariate time-series input with shape (T, F), where T is the time window and F is the number of input features. The model architecture is as follows:

- Layer 1: LSTM with U_1 units and `return_sequences=True`
- Layer 2: Dropout with dropout rate D_1
- Layer 3: LSTM with U_2 units and `return_sequences=False`
- Layer 4: Dropout with dropout rate D_2
- Layer 5: Dense output layer with a single unit (regression target)

The model is trained using the Adam optimizer, where the learning rate is selected from a predefined set $\{0.001, 0.005, 0.01\}$. The loss function used is MSE, which is standard for continuous regression tasks. The forecasting approach considers both the dataset size and the input window configuration.

In the ensemble models, the strengths of multiple algorithms are combined to produce more accurate and balanced predictions. The ensemble method used in this study is based on a weighted average approach that integrates the temporal pattern recognition capabilities of LSTM with the capacity of XGBoost to model both linear and nonlinear interactions among features. This hybrid structure is particularly effective for PV power forecasting, where the data is influenced by both historical patterns and external environmental conditions. By reducing the diversity of prediction errors, the ensemble yields more robust forecasts. The final prediction is generated using dynamic weights assigned to the outputs of LSTM and XGBoost, based on their individual performance on the test dataset. These weights are determined by metrics such as prediction accuracy or error rate, allowing the model to adaptively emphasize the more reliable estimator at any given time. Technically, this strategy optimizes the bias-variance trade-off and helps mitigate extreme prediction deviations. In scenarios where one model underperforms due to specific weather anomalies (e.g., sudden cloud cover or wind fluctuations), the other model's output can dominate, thereby enhancing forecast stability. This adaptive behavior makes the ensemble a more reliable and resilient solution for PV power prediction.

Specifically, in the implementation, the LSTM model's input was reshaped into a 3D format $(\text{samples}, \text{time steps} = 1, \text{features})$. The relative mean values of the XGBoost and LSTM predictions were used to compute the dynamic weights. By integrating both tree-based and sequence-based models, the ensemble approach ensures prediction robustness across varying conditions.

2.4. Hyperparameter Optimization

- **XGBoost Model:** GridSearchCV was used to optimize the hyperparameters: `n_estimators`, `learning_rate`, `max_depth`, `subsample`, and `colsample_bytree`. The objective function was set to `reg:squarederror`.
- **LSTM Model:** Hyperparameters for the LSTM model, such as units, dropout rates, and learning rate, were optimized using Keras Tuner with RandomSearch. The loss function is based on minimizing the mean squared error. Here, `units` refers to the number of neurons in the LSTM layer, `dropout` is used to prevent overfitting, and the `learning_rate` term defines the learning rate for the Adam optimizer.
- **SVR Model:** The kernel types `rbf` and `linear`, along with regularization parameters such as `C`, `epsilon`, and `gamma`, were optimized using GridSearchCV.
- **k-NN Model:** The hyperparameter space for the k-NN model includes neighbors count, weight strategy, distance metric (Manhattan, Euclidean), and tree algorithms, all of which were optimized through GridSearchCV.
- **RF Model:** The hyperparameters `n_estimators`, `max_depth`, `min_samples_split`, `max_features`, and `bootstrap` were optimized using GridSearchCV.
- **SARIMA Model:** To determine the stationarity of the time series, the Augmented Dickey-Fuller (ADF) test was applied. If the p-value of the test was less than 0.05, the series was considered stationary, and `d=0` was selected; otherwise, `d=1` was chosen. A grid search for the parameters `p` and `q` in the range (0, 1, 2) was performed, and the SARIMA model with the lowest Akaike Information Criterion (AIC) value was selected. The seasonal parameters were fixed at $(P = 1, D = d, Q = 1, s = 24)$. Additionally, the `auto_arima()` function, which automatically determines the `p`, `d`, `q`, and seasonal parameters, was used for model selection, with `stepwise=True` to narrow the iterative search and reduce computation time.

All models' performance was evaluated on the test dataset. As negative predictions are physically meaningless, the final predictions were filtered using the `np.clip()` function with a lower bound of 0. Furthermore, error analyses were performed using MAPE, RMSE, and seasonal day samples. Also, Table 1 presents the hyperparameters specific to each model considered in the study.

2.5. Evaluation Metrics

In this study, the forecasting performance of the models was evaluated using four widely accepted statistical metrics: root mean squared error (RMSE), R^2, mean absolute error (MAE), and mean absolute percentage error (MAPE). The R^2 score, also known as the coefficient of determination, quantifies how well the independent variables explain the variance in the dependent variable. A score of 1.0 indicates a perfect prediction, while negative values suggest that the model performs worse than a simple mean-based baseline. This metric is sensitive to the dataset's structure and offers insight into how effectively the model generalizes unseen data. When the residuals (prediction

Table 1: AI architecture and hyperparameter grid search parameters

AI architecture	Hyperparameter grid search	AI architecture	Hyperparameter grid search
XGBoost	**n_estimators:** [50, 100]	LSTM	**units1:** {32, 64, 96, 128}
	learning_rate: [0.01, 0.1]		**dropout1:** {0.1, 0.2, 0.3, 0.4, 0.5}
	max_depth: [3, 5]		**units2:** {32, 64, 96, 128}
	subsample: [0.8, 1.0]		**dropout2:** {0.1, 0.2, 0.3, 0.4, 0.5}
	colsample_bytree: [0.8, 1.0]		**learning_rate:** {0.001, 0.005, 0.01}
RF	**n_estimators:** [50, 100, 150]	k-NN	**n_neighbors:** [3, 5, 7, 9, 11, 15]
	max_depth: [None, 10, 20]		
	min_samples_split: [2, 5, 10]		**weights:** ['uniform', 'distance']
	min_samples_leaf: [1, 2, 4]		
	max_features: ['sqrt', 'log2']		**p:** [1, 2], # Manhattan (1), Euclidean (2)
	bootstrap: [True, False]		
SVR	**C:** [0.1, 1, 10, 100]		**algorithm:** ['auto', 'ball_tree', 'kd_tree']
	epsilon: [0.01, 0.05, 0.1, 0.2]		
	kernel: ['rbf', 'linear']		**leaf_size:** [20, 30, 40]
	gamma: ['scale', 'auto']		

errors) average out to zero, the R^2 becomes equivalent to the ratio of explained variance to total variance. The mathematical representation involves $\hat{y}_i$ as the predicted output for the i-th instance, y_i as the actual output, the mean target $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, and the squared error term $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \varepsilon_i^2$. MAE, or Mean Absolute Error, measures the average magnitude of errors between predicted and observed values, without considering their direction. As a form of L1-norm loss, it is also referred to as mean bias error (MBE) and provides a straightforward interpretation of prediction accuracy. Unlike metrics that square the error, MAE treats all deviations equally, which makes it particularly useful for scenarios where outliers are less of a concern. Lower MAE values, ideally approaching zero, indicate higher predictive reliability. Similarly, MAPE assesses error magnitude in relative terms and is not affected by the scaling of the target variable. To avoid undefined values when actual targets are zero, a small constant ε is typically added. This metric is advantageous when comparing model accuracy across datasets with different scales, and lower percentage values reflect better performance. MSE quantifies the average of the squared differences between predicted and actual values and is classified as an L2-norm loss. By squaring errors, it disproportionately penalizes larger deviations, making it particularly sensitive to outliers. This characteristic is useful in optimization tasks such as Ordinary Least Squares (OLS), where emphasizing large errors helps guide the learning process. The values for MSE range from zero to infinity, with values closer to zero suggesting better predictive accuracy. The RMSE (Root Mean Squared Error) is simply the square root of MSE, and while it shares the same units as the target variable, it offers a more interpretable measure of error magnitude. Lower RMSE values are preferable and indicate that the model is producing predictions that closely align with the actual values. Overall, Table 2 contains the equations for the relevant performance metrics.

Table 2: Performance Evaluation

Performance Metrics	Equation	Performance Metrics	Equation
RMSE$(y,\hat{y})$	$\sqrt{\frac{\sum_{i=0}^{n_{\text{samples}}-1}(y_i-\hat{y}_i)^2}{n_{\text{samples}}}}$	MAPE$(y,\hat{y})$	$\frac{1}{n_{\text{samples}}}\sum_{i=0}^{n_{\text{samples}}-1}\frac{\lvert y_i-\hat{y}_i\rvert}{\max(\varepsilon,\lvert y_i\rvert)}$
MAE$(y,\hat{y})$	$\frac{1}{n_{\text{samples}}}\sum_{i=0}^{n_{\text{samples}}-1}\lvert y_i-\hat{y}_i\rvert$	$R^2(y,\hat{y})$	$1-\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}$

3. Results and Discussion

3.1. Model Selection and Hyperparameter Tuning Results

Table 3 presents the results of the obtained hyperparameter tuning.

In the XGBoost model, parameters such as `colsample_bytree`, `subsample`, and `max_depth` are critical for achieving high accuracy. These parameters were optimized to maintain the model's learning ability while reducing the risk of overfitting. In particular, setting the `learning_rate` to 0.1 allowed small updates at each step, enhancing the overall model stability. Choosing `n_estimators` as 100 ensured that the model reached sufficient diversity.

In the LSTM model, which models temporal dependencies, the number of layers and neuron density were carefully determined. The two-layer model, with each LSTM cell consisting of 128 units, improved the model's ability to learn complex temporal patterns. Dropout rates (0.1 and 0.2) were specifically applied to prevent overfitting. The `learning_rate` was optimized to 0.01 to allow the model to converge quickly and stably.

In the SVR model, a linear kernel was chosen to reduce computational cost and enhance interpretability. The `C` value was set to 10 to make the model less flexible but more stable. The `epsilon` value (0.01) increased sensitivity to bring predictions closer to the target value. The `gamma` parameter was set to `scale`, allowing for automatic selection of an appropriate value based on data distribution.

In the k-NN model, hyperparameter optimization focused on the number of neighbors (`n_neighbors`) and the distance metric. The `n_neighbors` value was set to 15, creating a model that was resistant to noise with smoother decision boundaries. Additionally, using `p=1`

with the Manhattan distance metric improved accuracy. The `weights='distance'` setting increased the influence of nearby neighbors, enabling more accurate representation of local patterns.

In the RF model, hyperparameters such as the `max_depth` of decision trees (`max_depth=10`) and minimum sampling conditions (`min. samples leaf=1`, `min_samples_split=5`) were selected to balance the model's learning power and generalization ability. The `bootstrap = False` preference resulted in deterministic outcomes without sampling. The `max_features='sqrt'` setting ensured that each tree was more diverse, reducing the model's variance. With `n_estimators=100`, the model produced stable results by learning in different subspaces.

In the SARIMA model, the stationarity of the series was first examined using the Augmented Dickey-Fuller (ADF) test, and if necessary, the first difference ($d = 1$) was applied. After hyperparameter optimization through both grid-based search and the `auto_arima` function, it was determined that the $(1,1,1)(1,0,2,24)$ parameters yielded the best performance. The SARIMA model was configured to account for daily seasonality in hourly resolution data, and the predictions were evaluated on the test dataset. The poor MAPE performance of the model, particularly due to zero values during nighttime hours, showed that classical error metrics were skewed. However, due to its success in capturing seasonal patterns and its strong generalization ability, the SARIMA model provides a solid statistical approach for PV production time series.

3.2. Model Performance Comparison

Table 4 presents the performance metrics obtained. In this study, the performance of different machine learning and statistical models developed for PV power production forecasting was evaluated. In order to ensure a more reliable interpretation of the MAPE, a filtering threshold was applied. Specifically, hours during which the solar irradiance was below 0.1 kW/m² were excluded from the MAPE calculation. This threshold aims to eliminate periods where actual PV production is nearly zero, which can disproportionately inflate percentage-based error metrics despite small absolute prediction deviations.

The best performance was achieved with the ensemble model, which combined the LSTM and XGBoost models using a weighted average approach. With MAE: 0.0034, RMSE: 0.0057, and a very high R² value of 0.9995, the ensemble model produced the closest predictions to the actual data. After filtering out low-irradiance periods (below 0.1 kW/m²), the ensemble model achieved a significantly improved filtered MAPE of 2.82%, reflecting its high reliability during meaningful production hours. Ultimately, the ensemble approach improves the overall prediction performance by compensating for the weaknesses of individual models. When combined, LSTM's ability to capture temporal dependencies and XGBoost's strong regression capacity make short-term fluctuations more effectively captured.

SVR achieved a reasonable error rate with a filtered MAPE of 6.19%, while the RF model followed with 12.53%. These values, calculated after excluding low-irradiance hours (below 0.1 kW/m²), offer a more accurate representation of the models' performance during actual PV generation periods. Both models also exhibited strong R² values (0.9985 and 0.9958, respectively). While SVR is sensitive to data scaling, it yields stable predictions when the input data follows regular patterns. The Random Forest model, on the other hand, demonstrates strong robustness against overfitting in high-dimensional datasets, making it more tolerant to small-sample variability. The k-NN model showed relatively higher error rates (RMSE: 0.0419, filtered MAPE: 29.69%), suggesting its limited effectiveness in capturing the complex and high-frequency fluctuations typical of PV production data. Its reliance on local pattern similarity becomes a drawback under volatile or irregular time series conditions.

The SARIMA model, representing a statistical approach, showed lower performance compared to the other models. With MAE: 0.0591 and RMSE: 0.1211, its R² value was calculated to be only 0.7635. Although SARIMA captures seasonality and linear trends, its filtered MAPE was excessively high, as it performed poorly during low-output periods and was not reported due to error inflation. This suggests that while SARIMA can model periodic structures, it lacks the flexibility to handle nonlinear dynamics and abrupt changes commonly observed in PV production. In contrast, modern machine learning-based methods demonstrated greater accuracy and robustness for such complex forecasting tasks. SARIMA's deterministic nature limited its responsiveness to external influences, such as sudden weather variations, resulting in elevated prediction errors.

Table 3: Optimized hyperparameter settings for AI-based forecasting models

AI architecture	Optimized Hyperparameters	AI architecture	Optimized Hyperparameters
XGBoost	**n_estimators:** 100	LSTM	**units1:** 128
	learning_rate: 0.1		**dropout1:** 0.1
	max_depth: 5		**units2:** 128
	subsample: 0.8		**dropout2:** 0.2
	colsample_bytree: 0.8		**learning_rate:** 0.01
RF	**n_estimators:** 100	k-NN	**n_neighbors:** 15
	max_depth: 10		**weights:** distance
	min_samples_split: 5		**p:** Manhattan (1)
	min_samples_leaf: 1		**algorithm:** auto
	max_features: sqrt		**leaf_size:** 20
	bootstrap: False		
SVR	C: 10		
	epsilon: 0.01		
	kernel: linear		
	gamma: scale		

Table 4: Performance comparison among the AI models

AI Model	RMSE	MAE	R^2	Filtered MAPE
SARIMA	0.1211	0.0591	0.7635	N/A
k-NN	0.0419	0.0201	0.9717	29.69%
RF	0.0162	0.0079	0.9958	12.53%
SVR	0.0096	0.0074	0.9985	6.19%
LSTM+XGBoost Ensemble	0.0057	0.0034	0.9995	2.82%

4. Conclusions

This study comprehensively evaluated and compared the performance of various AI and statistical models for PV power forecasting using real-world meteorological and production data. The analyzed models included XGBoost, LSTM, RF, SVR, k-NN, and the SARIMA model. To leverage the strengths of deep learning and gradient boosting techniques, a weighted average ensemble model combining LSTM and XGBoost was proposed. The results demonstrated that nonlinear ML models, particularly XGBoost and LSTM, achieved superior accuracy in capturing the temporal and nonlinear dynamics of PV production. The ensemble model outperformed individual models in terms of error metrics such as RMSE, MAE, and MAPE, showing enhanced robustness and consistency across different seasons. From a practical perspective, the proposed ensemble approach presents a promising direction for integrating PV forecasting into energy management systems, microgrid operations, and electricity market bidding strategies. The study also emphasized the importance of model selection and hyperparameter tuning in improving forecasting reliability. Future research may focus on the following directions:

- Exploring hybrid architectures that incorporate CNN or Transformer-based models for spatio-temporal pattern learning.
- Extending the forecasting horizon to day-ahead or week-ahead predictions and analyzing uncertainty using probabilistic forecasting techniques.
- Integrating model explainability methods (e.g., SHAP values, attention mechanisms) to improve interpretability of black-box models.
- Applying the proposed ensemble model in real-time forecasting scenarios and testing its scalability in larger PV farms or at the grid level.

Article Information

Acknowledgements: The authors would like to express their sincere thanks to the editor and the anonymous reviewers for their helpful comments and suggestions.

Author's Contributions: All authors contributed equally to the writing of this paper. All authors read and approved the final manuscript. (or The article has a single author. The author has read and approved the final manuscript.)

Conflict of Interest Disclosure: No potential conflict of interest was declared by the author.

Copyright Statement: Authors own the copyright of their work published in the journal and their work is published under the CC BY-NC 4.0 license.

Supporting/Supporting Organizations: No grants were received from any public, private or non-profit organizations for this research.

Ethical Approval and Participant Consent: It is declared that during the preparation process of this study, scientific and ethical principles were followed and all the studies benefited from are stated in the bibliography.

Plagiarism Statement: This article was scanned by the plagiarism program. No plagiarism detected.

Availability of Data and Materials: Not applicable.

References

[1] S. Pu, Z. Li, H. Wan, and Y. Chen, "A hybrid prediction model for photovoltaic power generation based on information entropy," IET Gener. Transm. & Distrib., vol. 15, no. 3, pp. 436–455, 2021, doi: https://doi.org/10.1049/gtd2.12032.
[2] G. Wu, Y. Wang, Q. Zhou, and Z. Zhang, "Enhanced Photovoltaic Power Forecasting: An iTransformer and LSTM-Based Model Integrating Temporal and Covariate Interactions," Dec. 2024, doi: 10.48550/arXiv.2412.02302.
[3] G. Xiong, J. Zhang, X. Fu, J. Chen, and A. W. Mohamed, "Seasonal short-term photovoltaic power prediction based on GSK–BiGRU–XGboost considering correlation of meteorological factors," J. Big Data, vol. 11, no. 1, p. 164, 2024, doi: 10.1186/s40537-024-01037-x.
[4] A. Hussain, Z. A. Khan, T. Hussain, F. U. M. Ullah, S. Rho, and S. W. Baik, "A Hybrid Deep Learning-Based Network for Photovoltaic Power Forecasting," Complexity, vol. 2022, no. 1, p. 7040601, 2022, doi: https://doi.org/10.1155/2022/7040601.
[5] M. Liu, X. Wang, and Z. Zhong, "Ultra-Short-Term Photovoltaic Power Prediction Based on BiLSTM with Wavelet Decomposition and Dual Attention Mechanism," Electronics, vol. 14, no. 2, 2025, doi: 10.3390/electronics14020306.
[6] W. Guo, L. Xu, T. Wang, D. Zhao, and X. Tang, "Photovoltaic Power Prediction Based on Hybrid Deep Learning Networks and Meteorological Data," Sensors, vol. 24, no. 5, 2024, doi: 10.3390/s24051593.
[7] Q. Zhang, N. Tang, J. Lu, W. Wang, L. Wu, and W. Kuang, "A hybrid RBF neural network based model for day-ahead prediction of photovoltaic plant power output," Front. Energy Res., vol. Volume 11, 2024, doi: 10.3389/fenrg.2023.1338195.
[8] M. Alrashidi and S. Rahman, "Short-term photovoltaic power production forecasting based on novel hybrid data-driven models," J. Big Data, vol. 10, no. 1, p. 26, 2023, doi: 10.1186/s40537-023-00706-7.
[9] X. Zhang et al., "Prediction of photovoltaic power generation based on a hybrid model," Front. Energy Res., vol. Volume 12, 2024, doi: 10.3389/fenrg.2024.1411461.
[10] F. Chen, J. Ding, Q. Zhang, J. Wu, F. Lei, and Y. Liu, "A PV Power Forecasting Based on Mechanism Model-Driven and Stacking Model Fusion," J. Electr. Eng. Technol., vol. 19, no. 8, pp. 4683–4697, 2024, doi: 10.1007/s42835-024-01906-8.
[11] M. S. Ibrahim, S. M. Gharghory, and H. A. Kamal, "A hybrid model of CNN and LSTM autoencoder-based short-term PV power generation forecasting," Electr. Eng., vol. 106, no. 4, pp. 4239–4255, 2024, doi: 10.1007/s00202-023-02220-8.

[12] M. Tovar, M. Robles, and F. Rashid, "PV Power Prediction, Using CNN-LSTM Hybrid Neural Network Model. Case of Study: Temixco-Morelos, México," Energies, vol. 13, no. 24, 2020, doi: 10.3390/en13246512.
[13] D. Salman, C. Direkoglu, M. Kusaf, and M. Fahrioglu, "Hybrid deep learning models for time series forecasting of solar power," Neural Comput. Appl., vol. 36, no. 16, pp. 9095–9112, 2024, doi: 10.1007/s00521-024-09558-5.
[14] D. Kossoko, B. Audace, A. R. Gilles, and A. B. Macaire, "LSTM and XGBoost Models for 24-hour Ahead Forecast of PV Power from Direct Irradiation," vol. 5, no. 2, pp. 229–241, 2024, doi: 10.22044/rera.2023.12880.1209.
[15] S.-C. Lim, J.-H. Huh, S.-H. Hong, C.-Y. Park, and J.-C. Kim, "Solar Power Forecasting Using CNN-LSTM Hybrid Model," Energies, vol. 15, no. 21, 2022, doi: 10.3390/en15218233.
[16] N. Zhou, B. Shang, M. Xu, L. Peng, and G. Feng, "Enhancing photovoltaic power prediction using a CNN-LSTM-attention hybrid model with Bayesian hyperparameter optimization," Glob. Energy Interconnect., vol. 7, no. 5, pp. 667–681, 2024, doi: https://doi.org/10.1016/j.gloei.2024.10.005.
[17] A. Agga, A. Abbou, M. Labbadi, Y. El Houm, and I. H. Ou Ali, "CNN-LSTM: An efficient hybrid deep learning architecture for predicting short-term photovoltaic power production," Electr. Power Syst. Res., vol. 208, p. 107908, Jul. 2022, doi: 10.1016/j.epsr.2022.107908.
[18] F. Wang, Z. Xuan, Z. Zhen, K. Li, T. Wang, and M. Shi, "A day-ahead PV power forecasting method based on LSTM-RNN model and time correlation modification under partial daily pattern prediction framework," Energy Convers. Manag., vol. 212, p. 112766, May 2020, doi: 10.1016/j.enconman.2020.112766.
[19] R. Chen, S. Gao, Y. Zhao, D. Li, and S. Lin, "A hybrid model based on the photovoltaic conversion model and artificial neural network model for short-term photovoltaic power forecasting," Front. Energy Res., vol. Volume 12, 2024, doi: 10.3389/fenrg.2024.1446422.
[20] S. Wang and J. Ma, "A novel GBDT-BiLSTM hybrid model on improving day-ahead photovoltaic prediction," Sci. Rep., vol. 13, no. 1, p. 15113, Sep. 2023, doi: 10.1038/s41598-023-42153-7.
[21] A. Abdellatif et al., "Forecasting Photovoltaic Power Generation with a Stacking Ensemble Model," Sustainability, vol. 14, no. 17, 2022, doi: 10.3390/su141711083.
[22] A. Sagingalieva et al., "Photovoltaic power forecasting using quantum machine learning," pp. 1–12, 2023, [Online]. Available: http://arxiv.org/abs/2312.16379.