Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal, 2025, 19(11), 206–220
https://doi.org/10.12913/22998624/209204
ISSN 2299-8624, License CC-BY 4.0

Received: 2025.07.01
Accepted: 2025.09.15
Published: 2025.10.01

Integrating meteorological data for next-day photovoltaic energy prediction using XGBoost

Piotr Kraska[1], Krzysztof Hanzel[1*]

[1] Department of Telecommunications and Teleinformatics, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
* Corresponding author's e-mail: krzysztof.hanzel@polsl.pl

ABSTRACT

This study focuses on the methodology for developing a model to forecast electricity production from photovoltaic (PV) panels through the analysis and processing of meteorological and historical data. An important aspect is the application of the XGBoost algorithm, with a detailed discussion of its selection and the process of hyperparameter tuning. The study also presents an approach to integrating various data sources, including retrieving meteorological data from the Open-Meteo API and combining it with data on PV energy production. The resulting model demonstrates high predictive performance, confirming the effectiveness of XGBoost in capturing complex, non-linear relationships even with limited or noisy training data.

Keywords: photovoltaic power forecasting, xgboost, machine learning, energy prediction models, meteorological data integration.

INTRODUCTION

Solar energy, as one of the main sources of renewable energy, plays a crucial role in the current energy transition. Electricity production in Europe increased from 91,096 MW in 2014 to 288,644 MW in 2023. Globally, production from solar panels grew from 192,602 MW in 2014 to as much as 1,294,481 MW in 2023 (1). Although the topic remains controversial, public attitudes are generally positive, even in regions with currently low levels of energy production (2). Despite the rapid development of this technology, many challenges remain unresolved, particularly related to energy storage and utilization during maximum power generation by photovoltaic panels, the low efficiency of solar energy production and usage, especially in countries with limited sunny days (3) as well as self-consumption of energy, especially under conditions of low transmission network efficiency (4).

Currently, the two most popular methods for dealing with both excess and shortage of production from solar panels are energy storage systems (energy banks) and Net-Metering. An energy bank is a large-sized battery connected to the installation, which stores the excess energy produced when generation exceeds demand. In case of a shortage, the energy bank releases energy to the household or building(s) of the company/enterprise. Although this solution shows great potential, the economic feasibility of purchasing and installing an energy bank is often limited, as investment costs may outweigh the benefits derived from storing and using the accumulated energy. Additionally, energy banks could pose a fire hazard due to the use of lithium-ion energy storage, which would be virtually impossible to extinguish in the event of failure (5).

The second option, currently the most popular, is sending the generated energy to the grid (Net-metering) or dynamic billing for production (Net-billing). This method allows for "storing" energy in the grid. The system owner can send excess energy to the grid and later retrieve it during periods of lower production, after deducting the

appropriate fees, or purchase energy considering previously earned discounts. While Net-metering is a beneficial solution, the high density of photovoltaic panels in a given area can risk overloading the grid. To minimize this issue and enable further development of photovoltaic technologies, grid modernization or increased self-consumption of energy becomes necessary, while simultaneous network planning and optimization (6).

The purpose of the research is to verify the effectiveness of the XGBoost and LSTM prediction algorithms for forecasting energy production from photovoltaic (PV) installations, within a specified range of installation capacities, based on the weather conditions and specific characteristics of local PV instalations. The study particularly focuses on evaluating a modified version of the XGBoost algorithm in comparison to its standard implementation and another selected predictive model in this case LSTM. This new approach is for solving these problems in the wide optimization of energy management. Predictive modeling plays a key role in optimizing energy management by enabling precise estimation of electricity output based on weather data and site-specific PV parameters. Better forecasting allows users to align energy consumption with production peaks, utilize energy storage more effectively, and minimize energy losses. In the long term, the use of accurate predictive models contributes to reducing operational costs, lowering grid dependency, and increasing energy self-sufficiency in households or small-scale PV farms.

METHODOLOGY

This chapter presents a detailed description of the methodological approach that was applied to develop and implement predictive models for photovoltaic installations.

The importance of solar radiation and atmospheric conditions on PV power production

The photovoltaic effect can be defined as the generation of potential when radiation ionizes the area inside or near the built-in potential barrier of a semiconductor. It is characterized by a self-generated electromotive force and a current that can deliver energy to a receiver; the primary source of energy comes from ionizing radiation (7,8).

The intensity of solar radiation, i.e., the power of sunlight per unit area (W/m²), is the main factor influencing energy production in photovoltaic systems. The solar radiation intensity consists of three components:

- Global horizontal irradiance (GHI) – the total amount of solar radiation that can be received on a horizontal surface, including both direct and diffuse radiation;
- Direct normal irradiance (DNI) – the amount of radiation received in the form of direct rays that fall perpendicularly to the surface and are not disturbed by clouds or other atmospheric factors;
- Diffuse horizontal irradiance (DHI) – the portion of solar energy that reaches the horizontal surface after being scattered in the atmosphere (9).

To maximize solar radiation utilization, it is crucial to position the installation relative to the sun. The total radiation intensity is the sum of direct, diffuse, and reflected radiation from the ground. This can be described by Equation 1 (10–12).

$$G_{\mathrm{TOT}} = G_{\mathrm{DNI}} \cdot \cos\theta + G_{\mathrm{DIFF}} \cdot \left(\frac{1+\cos\beta}{2}\right) + + G_{\mathrm{REFL}} \cdot \left(\frac{1-\cos\beta}{2}\right) \quad (1)$$

where: β is the tilt angle of the photovoltaic panels, and θ is the angle of incidence of the sunlight. The angle of incidence can be calculated using Equation 2 (13–15).

$$\cos\theta = \cos z \cdot \cos\beta + + \sin z \cdot \sin\beta \cdot \sin(\varphi_s - \gamma) \quad (2)$$

where: z is the solar zenith angle (the angle between the vertical line and the direction of solar radiation), and φ_s is the solar azimuth angle (the angle between the southern direction and the projection of solar radiation onto the surface of the photovoltaic panel). The parameter α represents the solar altitude angle, and γ is the azimuth angle of the surface (15). A visualization of the most important angles can be found in Figure 1.

In addition to the positioning of photovoltaic panels and solar radiation, weather phenomena play a crucial role in the efficiency of electricity production. Key weather factors include (9):

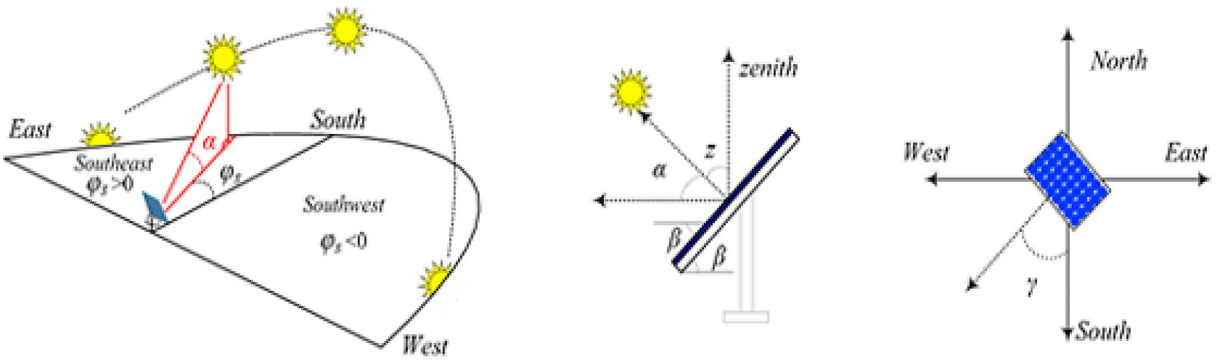

Figure 1. Visualization of characteristic angles (15)

- Cloud cover: Clouds can block solar radiation, causing fluctuations in energy production.
- Humidity: High air humidity can lead to increased cloud cover and scattering of radiation in the atmosphere, affecting the amount of solar energy reaching the photovoltaic system.
- Precipitation and dust: Atmospheric precipitation such as snow or rain is often associated with cloud cover, which blocks or scatters sunlight. Snow and dust can also cover the PV panels, reducing the area available for electricity generation.
- Temperature: The efficiency of a photovoltaic system typically decreases as temperature rises. As temperature increases, the efficiency of the cells in PV panels drops, leading to a reduction in overall energy production. However, it is important to note that higher temperatures are often linked to increased solar radiation levels, which paradoxically may increase the available energy, although this does not fully compensate for the decreased efficiency of the cells.

Time series

Time series are datasets organized at successive time intervals, used for the analysis and modeling of phenomena that change over time (16). The data in such series can be collected at regular or irregular intervals. In the case of irregular intervals, resampling methods such as aggregation or interpolation can be used. The primary goal of time series analysis is to identify patterns in the data and forecast future values based on historical data.

Another aspect when working with time series is the transformation of values. Transformations are intended to improve the quality of the analysis and adjust the data to the requirements of predictive models. One commonly used approach is logarithmic transformation, which helps stabilize the variance of the data and reduce the impact of large outliers. It should be noted that this method should only be used for positive values due to the properties of the logarithmic function, which is not defined for $x < 0$ in the set of real numbers. The second approach is the Box-Cox transformation, defined by the formula (3).

$$y_t^{\lambda} = \begin{cases} \dfrac{y_t^{\lambda} - 1}{\lambda} & \text{for } \lambda \neq 0 \\ \log y_t & \text{for } \lambda = 0 \end{cases} \tag{3}$$

where: y_t is the value of the time series and λ is the transformation parameter. The value of λ is chosen in such a way as to minimize deviations from normality in the transformed time series. The third approach is power transformation, described by the formula (4).

$$\omega_t = y_t^p \tag{4}$$

where: y_t is the value of the time series and p is the parameter determining the degree of transformation. In most cases, transformations are not necessary, but if this method is applied, it is important to invert the transformation in order to obtain prediction results in the original scale. Another method of time series processing is smoothing. It involves smoothing fluctuations in the time series to eliminate noise (outlying observations) and highlight major trends or patterns. There are two methods of data smoothing:

- Moving average: This method calculates the average value within a window that includes

neighboring points. The moving average is computed using formula (5).

$$MA_t = \frac{y_t + y_{t-1} + y_{t-2} + \cdots + y_{t-n}}{n} \quad (5)$$

- Exponential smoothing: This type of smoothing uses the formula (6).

$$ES_t = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \\ + + \alpha(1-\alpha)^2 y_{t-2} + \cdots \quad (6)$$

One of the methods of processing time series is called decomposition. The decomposition method is an analytical technique that allows for the breakdown of data into individual components with different characteristics, such as trend, seasonality, and residuals.

Decomposition

Decomposition can be divided into:

- additive decomposition. In this approach, the values of the time series y_t are represented as the sum of components (7) (17).

$$y_t = S_t + T_t + R_t \quad (7)$$

where: S_t is the seasonal component, T_t is the trend/cycle component, and R_t is the residual component, which represents random fluctuations or noise,

- multiplicative decomposition. This decomposition is useful when seasonal fluctuations or variability around the trend are proportional to the level of the values in the series (16). The multiplicative decomposition is described by the formula (8) (17).

$$y_t = S_t \cdot T_t \cdot R_t \quad (8)$$

In addition to time series processing, the importance of feature engineering should be noted. Feature engineering refers to creating new variables based on the original data. For time series, the most common features are those based on time-related parameters such as hours, months, or seasons. Furthermore, features related to lag (lag features), aggregate statistics, or autocorrelation can be extracted. Lag features are created based on previous values of the time series. Other features include aggregate statistics calculated over specific time windows, such as the mean, median, maximum and minimum values, or standard deviation. These statistics provide valuable information about local trends or variability over time. Additionally, feature engineering involves analyzing and utilizing autocorrelation, which measures how values of the time series are related at different time lags. High autocorrelation at specific lags may indicate the presence of repeating patterns in the data, which are important for forecasting (9).

All the aforementioned time series processing and analysis methods help improve the quality of data, which, when combined with predictive methods, allow for more accurate forecasting of future values.

Prediction methods

There are various approaches to forecasting, ranging from simple methods, which are easy to implement and interpret, to more advanced algorithms that require more data and computational power but offer higher forecasting accuracy.

Simple methods

The first methods for time series prediction are simple methods. Some of these include:

- mean: the forecast is the average of the time series values calculated on training data,
- naive: the forecast is the most recent observed value in the series,
- naive with seasonality adjustment: the forecast is equal to the last value from the same season,
- drift: the forecast is calculated as the sum of the most recent value and the average change between consecutive observations.

Simple forecasting methods are mainly useful for very short-term forecasts, as their effectiveness decreases as the forecast horizon extends.

Linear regression

This method assumes a linear relationship between the predicted variable y and other series. It is described by the formula (9) (18).

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \\ + \cdots + \beta_k x_{k,t} + \epsilon_t \quad (9)$$

ETS models (exponential smoothing state space models)

This method is based on exponential smoothing, taking seasonality and trends into account. The following ETS models can be presented:

- simple exponential smoothing;
- holt's method – an extension of simple exponential smoothing, this technique allows for forecasting the next period based on actual data and previous predictions, and also reduces the amount of data needed for analysis (19);
- holt-winters method – also known as triple exponential smoothing. This method accounts for seasonal fluctuations in time series. An important element of this method is the gamma parameter, which allows smoothing the seasonal component (20);
- holt-winters with trend damping – this method includes an additional parameter that gradually reduces the influence of the trend, which, over time, becomes smoothed and eventually turns into a horizontal line. This approach makes the forecasts more realistic and stable over the long term (21).

ARIMA model (autoregressive integrated moving average)

The ARIMA method is used for the analysis and forecasting of time series. Its foundation lies in the assumption that the value of the time series at a given point in time depends on its previous values, as well as the earlier forecast errors. The ARIMA model consists of three parameters: (p, d, q) (21,22):

Autoregressive (AR(p)) part

Refers to autoregression, i.e., the use of previous values of the time series to predict the current value. This part is described by formula (10) (21). It is worth noting that for higher degrees of non-stationarity, differencing of higher orders may be needed.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \\ + \cdots + \phi_p y_{t-p} + \epsilon_t \tag{10}$$

where: ϕ are the autoregressive coefficients, p is the number of lags, $\in_t$ is the random error.

Integrated (I(d)) part

Involves differencing the time series to make it stationary (eliminating trends and non-stationarity in the mean). The first-order differencing is described by formula (11) (21).

$$y'_t = y_t - y_{t-1} \tag{11}$$

Moving average (MA(q)) part

Uses earlier forecast errors to predict the current value. The model utilizes the relationship between the observation and the residual error obtained from the moving average model applied to the lagged observations. The moving average component describes the error as a combination of earlier errors. The parameter q etermines the number of past errors to be considered in the model. The moving average calculation is described by formula (12) (21).

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \\ + \cdots + \phi_p y_{t-p} + \epsilon_t \tag{12}$$

where: θ are the moving average coefficients, q is the number of error lags.

The ARIMA model often encounters difficulties in handling nonlinearities and higher-order seasonal variations. In such cases, it is recommended to use the Seasonal ARIMA (SARIMA) model. However, due to its seven parameters, the SARIMA model (p,d, q) (P, D,Q) is quite complex and inefficient for large datasets. An alternative approach to reduce the complexity of such models, resulting from their dynamic nature, is the use of the sliding window technique with a fixed size (23).

The ARIMA model is relatively easy to implement, but it may not be sufficiently effective in capturing the complex relationships between solar radiation intensity, weather conditions, and PV energy production.

XGBoost model (extreme gradient boosting algorithm)

The XGBoost algorithm uses gradient boosting of decision trees, iteratively building models (in this case, decision trees) to improve results based on the residuals of previous models (22,23). Each iteration adds a new tree to minimize the loss function, so each successive model learns to correct the errors of its predecessors. The XGBoost algorithm can be applied to both classification and regression problems. The most important operations in the algorithm are the computation of the loss function and feature importance.

Formula (13) describes the computation of the loss function (24–26).

$$L(t) \approx \sum_{i=1}^{n} \left[\begin{array}{c} l\left(y_i, y_i^{(t-1)}\right) \\ +g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) + \omega(f_t) \end{array} \right] \quad (13)$$

where: $L(t)$ – the loss function in the t-th iteration of the model, $l\left(y_i, y_i^{(t-1)}\right)$: The loss function for the i-th sample, based on the actual value y_i and the predicted value in the previous iteration $y_i^{(t-1)}$, g_i – the first-order gradient of the loss function with respect to the predicted value in the previous iteration. It represents the direction and magnitude of the change needed in predictions to reduce the loss function, $f_t(x_i)$ – the predicted value by the current decision tree f_t for the i-th sample x_i, h_i – the second-order gradient. Including this term improves the stability of optimization and accounts for the curvature of the loss function, $\omega(f_t)$ – the regularization parameter. It reduces the complexity of the model, preventing overfitting (25).

The regularization parameter is calculated based on formula (14) (24–26):

$$\omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} \omega_j^2 \quad (14)$$

where: T – the number of leaves in the decision tree, ω_j: the value assigned to leaf j, γ – the regularization parameter penalizing tree complexity (the number of leaves), λ – the regularization parameter penalizing the weights of the leaves.

Formula (15) describes the calculation of feature importance in the XGBoost algorithm (25,26).

$$\text{gain} = \frac{1}{2} \left[\begin{array}{c} \frac{\sum_{I_L} g_i^2}{\sum_{I_L} (h_i + \lambda)} + \\ + \frac{\sum_{I_R} g_i^2}{\sum_{I_R} (h_i + \lambda)} + \frac{\sum_{I} g_i^2}{\sum_{I} (h_i + \lambda)} \end{array} \right] \quad (15)$$

where: $I = I_L \cup I_R$, I_L and I_R represent the number of samples in the left and right nodes of the decision tree, λ and γ are the regularization parameters. The higher the feature's strength, the higher its importance for the objective and the more crucial the feature becomes (21).

The XGBoost algorithm is currently one of the most popular choices for forecasting time series values. Based on the literature review it can be said that the main advantages of this algorithm are (23–25): effectiveness in handling nonlinear and multidimensional data, flexibility in handling missing data thanks to the "sparsity-aware" mechanism, scalability and optimization as well as high prediction accuracy. An additional advantage of the XGBoost algorithm is its ability to handle time series with low frequency or insufficient data, where solutions such as deep learning may not be applicable.

Deep networks with LSTM layer (long short-term memory)

Deep neural networks with LSTM layers are increasingly being used for time series problems. LSTM is an advanced type of recurrent neural network (RNN) layer designed to store information over long periods of time. "The LSTM memory approach works well for time series fitting because it can handle both long-term and short-term dependencies in the data" (25). An important advantage is the absence of the gradient explosion problem, as well as the lack of the need for manual feature selection.

In Figure 2, the LSTM layer is visualized. x_t represents the value at time t in the sequential data, and C_t and h_t correspond to the cell state and hidden state of the sequential unit at time t respectively.

The learning process of each LSTM neural unit consists of three main phases: forgetting, selective retention, and output (25). In the forgetting phase, the first gate σ (from left to right) at time t_{fl} determines which information from the previous cell state C_{t-1} should be discarded. In the selective retention phase, the second gate σ at time t_l decides which information from the current input should be updated. Then, the *tanh* function generates candidate values for the new cell state at time t_{cl}. A part of the cell state C_{t-1} is forgotten, creating its modified version. The retained candidate values are then combined with the remaining contents of C_{t-1}. The result of these operations is the new cell state at time t, obtained by combining the candidate and forgotten values. In the final phase, known as the "output" phase, the third gate σ at time t computes the output value O_t, which determines which data should be output. The cell state at time t is transformed using the *tanh* function to

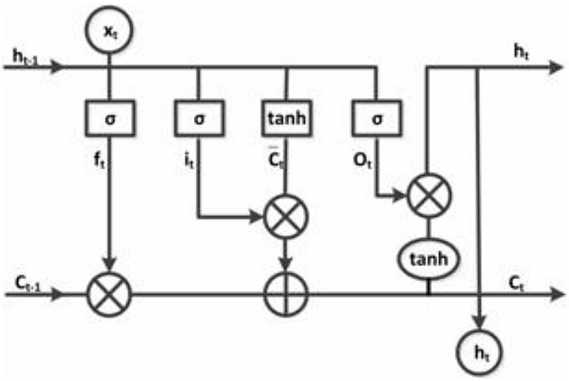

Figure 2. Example LSTM layer (25)

obtain the hidden layer h_t, which represents the result of the network (25).

Formulas 16–21 represent the operations as follows:

- input gate:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{16}$$

- forget gate:

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \tag{17}$$

- output gate:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{18}$$

- candidate for the new cell state:

$$\widetilde{C_t} = \tanh(W_s[h_{t-1}, x_t] + b_s) \tag{19}$$

- cell state at time *t*:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{20}$$

- hidden layer output at time *t*:

$$h_t = o_t * \tanh(C_t) \tag{21}$$

Data preprocessing

The dataset used in this work originates from private databases provided by third parties. The measurements from photovoltaic panels come from four locations with installations of the following capacities: 26 kWp, 25.55 kWp, and two installations of 49.5 kWp each. Measurements for the 26 kWp and 25.55 kWp installations were taken from October 15, 2023, to October 15, 2024. Measurements for the 49.5 kWp installations were taken from May 8, 2023, to May 8, 2024. Weather data was retrieved using the Open-Meteo API. Table 1 describes all the features used in the model training process.

Prior to model training, a comprehensive data preprocessing pipeline was implemented to ensure data quality and consistency. As illustrated in Figure 3, the process began with the removal of erroneous PV measurements and nighttime weather data, which are not relevant for solar energy prediction. Subsequently, PV and weather datasets were merged and resampled to a uniform 15-minute resolution, then further aggregated to hourly intervals to reduce short-term variability. Feature engineering techniques were applied to enrich the dataset with additional variables, such as season indicators and solar altitude. Finally, both input and output data were normalized – the inputs to a [0, 1] range, and the outputs to [0, 1] kW based on the rated power of each installation – to facilitate model convergence and comparability across systems.

Based on the input data, a heatmap was created showing the correlation between the input parameters and the "value_kW", which represents the historical hourly measurements from the photovoltaic installations. Figure 4 illustrates the relationships between these parameters.

Experiments

Traditional predictive modeling methods can be prone to overfitting and may exhibit low forecasting accuracy when dealing with complex, multidimensional datasets. Therefore, the prediction model for PV power was based on a long short-term memory (LSTM) neural network, which is well-suited for capturing long-term dependencies in the data (24).

During the experiments, several LSTM network configurations were tested in order to identify the optimal structure for the prediction task. The number of LSTM layers (num_layers) varied from 1 to 5, while the number of hidden units per layer (hidden_size) ranged from 16 to 256. Each model received an input sequence of time-dependent features, with a sequence length (sequence_length) between 32 and 72 time steps, depending on the configuration. The network output was a single value (output_size = 1), representing the predicted PV power. The number of input features (input_size) depended on the dataset and remained constant for a given experiment. Training was conducted for 10 to 250 epochs (num_epochs), with learning rates (learning_rate) explored in the range from 0.0001 to 0.1. These values were manually adjusted through iterative

Table 1. Input features

Feature Name	Description
Temperature_2m	Temperature at 2 meters (°C)
Relative_humidity_2m	Relative humidity at 2 meters (%)
Dew_point_2m	Dew point temperature (°C)
Apparent_temperature	Apparent temperature (°C)
Precipitation	Precipitation amount (mm)
Rain	Rainfall in the given hour (mm)
Wind_speed_10m	Wind speed at 10 meters (km/h)
Wind_gust_10m	Maximum wind gusts at 10 meters (km/h)
Shortwave_radiation	Instantaneous shortwave solar radiation (W/m²)
Diffuse_radiation	Instantaneous diffuse solar radiation (W/m²)
Global_tilted_irradiance	Instantaneous global radiation on a tilted surface (W/m²)
UV_index	Ultraviolet (UV) radiation index
Altitude	Sun altitude at the given moment
Is_Spring, Is_Summer, Is_Fall, Is_Winter	Binary values (0/1) indicating the current season (spring, summer, fall, winter)
Weather_code	Numerical code representing current weather conditions
Cloud_coverage	Cloud cover percentage (%)
Surface_pressure	Atmospheric pressure (hPa)
Date and Time	Used for extracting time-based features
Historical PV Data	Measurement data from photovoltaic panels

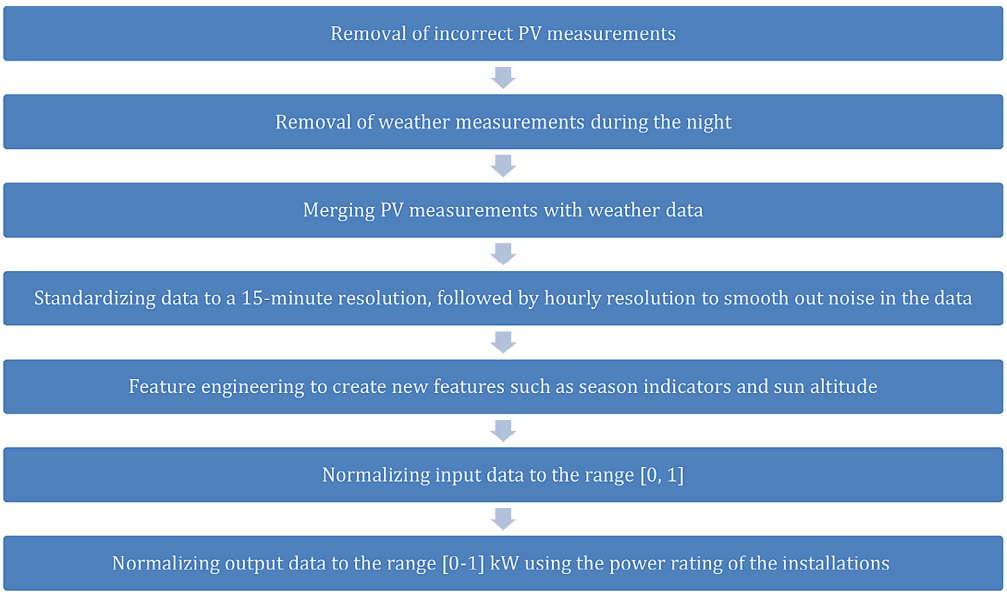

Figure 3. Steps of data preprocessing

experimentation in order to balance convergence speed and model stability.

The final model architecture, which yielded the best validation performance, used two LSTM layers with 64 hidden units each, followed by a feed-forward block composed of three dense layers with decreasing size and ReLU activations. Each dense layer included dropout regularization with rate of 0.3 to prevent overfitting. The full list of tested hyperparameters and their value ranges is presented in Table 2.

Additionally, XGBRegressor and XGBRFRegressor models were chosen for the experiments due to their ability to handle complex and

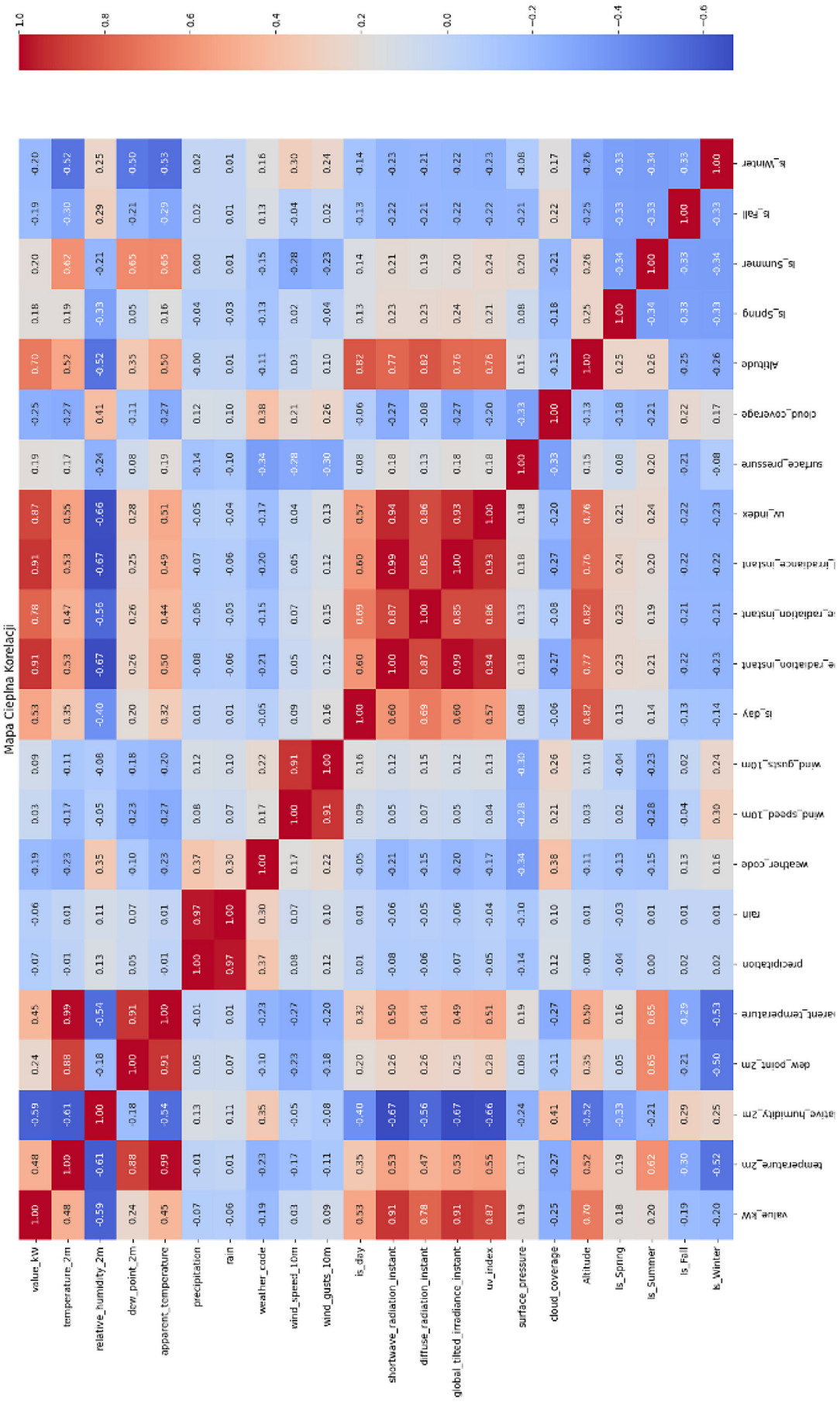

Figure 4. Relationship between input parameters

nonlinear relationships within the data (27). Figure 5 illustrates the process of analyzing the selected machine learning models.

The hyperparameter space potentially containing the most optimal parameters was searched using the BayesSearchCV function, which works by creating a probabilistic model that maps input hyperparameters to the output objective function (28). In the search process, the user defines the hyperparameter space, which is the range of values that can be tested for a given model. This space may include both discrete variables (e.g., number of trees in a random forest model) and continuous variables (e.g., regularization coefficient in regression).

BayesSearchCV starts the optimization with several random samples, allowing the construction of an initial probabilistic model that describes the relationship between the hyperparameters and the objective function value (e.g., model accuracy). In subsequent iterations, the algorithm selects new hyperparameter sets, guided by the exploration principle (trying new areas of the space) and the exploitation principle (focusing on the best values found so far).

For the hyperparameters selected by BayesSearchCV, the model was analyzed using test data. The following metrics were used to evaluate the model:

Mean absolute error (MAE) – This measures the average absolute error between the actual values y_i and the predicted values $\hat{y}_i$. The MAE has the same unit as the data. The formula for MAE is given in Equation 22 (25):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{22}$$

Mean squared error (MSE) – This measures the average of the squared errors, meaning that larger errors have a greater impact on the result. The formula for MSE is given in Equation 23 (18):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{23}$$

Root mean squared error (RMSE) – This metric is derived from MSE, but the result is expressed in the same units as the original data *y*. RMSE is particularly useful as it highlights large errors more than MAE, meaning that larger deviations from actual values are penalized more. This allows for better evaluation of models that should minimize large prediction errors. The formula for RMSE is given in Equation 24 (25):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{24}$$

Coefficient of determination (R^2) – A statistical measure used to determine how much of the variability in the dependent variable *y* is explained by the independent variables in the regression model. Its value is given by R^2. When $R^2 = 1$, it indicates the model's predictive power. When $R^2 = 0$, the model does no better than predicting the mean value of the data $\bar{y}$. For $R^2 < 0$, the model performs worse than simply predicting the mean. The formula for R^2 is given in Equation 25 (25):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{25}$$

If the quality metrics were better than in the previous trial, the model was saved, and the search space for hyperparameters was further refined. The process continued until the model consistently yielded a low R^2 or, despite achieving a satisfactory R^2, failed to improve RMSE, MAE, and MSE. Finally, the model with the lowest RMSE and MAE values for the test data was selected. Table 3 presents the hyperparameter space considered during the tuning of the XGBRegressor

Table 2. Hyperparameter space for the LSTM network

Hyperparameter	Description	Current Value	Range
input_size	Number of input features	data.shape[1] - 1	Depends on the dataset
hidden_size	Number of neurons in the hidden layer	64	16–256
output_size	Number of output neurons	1	Always 1
num_layers	Number of LSTM layers	2	1–5
learning_rate	Learning rate	0.001	0.0001 - 0.1
num_epochs	Number of training epochs	50	10–250
sequence_length	Length of the input sequence (time window)	72	32–72

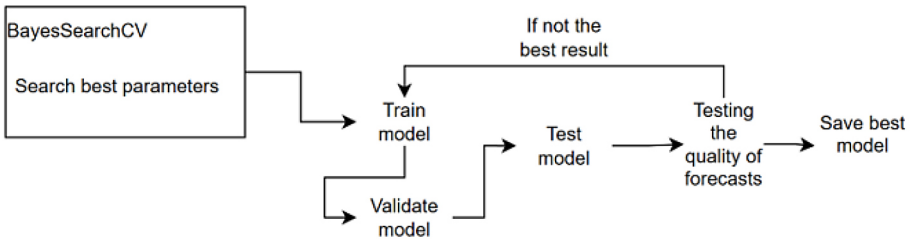

Figure 5. Analysis process of selected machine learning models

model, along with the best values obtained as a result of the optimization process.

For each hyperparameter, the search range and the optimal value are provided. The number of trees (n_estimators) was tested within the range from 50 to 2000, with the best result achieved at 1000. The maximum tree depth (max_depth) was set to 10 to maximize model complexity while avoiding overfitting. The learning rate (learning_rate), which affects the convergence speed, was optimally set to 0.0222. Other parameters controlling tree pruning (gamma, min_child_weight) and regularization (reg_alpha, reg_lambda) were also adjusted to maximize prediction accuracy. Parameters such as the loss function (objective), evaluation metric (eval_metric), tree construction method (tree_method), and processing device (device) were kept fixed.

RESULTS AND DISCUSSION

This chapter presents a comparative evaluation of the models discussed in the previous section. The analysis focuses on quantitative metrics to determine which algorithm offers the highest accuracy in short-term PV power forecasting. It is important to emphasize that the final performance assessment was conducted exclusively on a single, designated test installation (49.5 kWp) – the only dataset not used during the training phase. Each of the three models – XGBRegressor, XGBRFRegressor, and LSTM – was evaluated independently on this test dataset, allowing for an objective comparison of their predictive capabilities. Moreover, the results reported in this section represent values obtained solely for the test installation and are not aggregated across all four installations. The remaining three PV systems were used strictly for training purposes. This distinction ensures that the reported performance reflects the models' ability to generalize unseen data.

Table 4 presents the best metric values achieved by the models, highlighting the performance of XGBRegressor and XGBRFRegressor in particular compared to LSTM.

Table 3. Hyperparameter space and best results for XGBRegressor

Hyperparameter	Description	Parameter space (Range)	Best for XGBRegressor
n_estimators	Number of boosting rounds	50–2000	1000
max_depth	Maximum depth of each tree	2–20	10
learning_rate	Step size shrinkage to prevent overfitting	0.005–0.05	0.0222
subsample	Fraction of data used per tree	0.3–0.9	0.9
gamma	Minimum loss reduction for further splitting	0.1–0.8	0.1
min_child_weight	Minimum sum of instance weight in a child	1–8	6
reg_alpha	L1 regularization (Lasso)	0.1–1.5	0.5
reg_lambda	L2 regularization (Ridge)	0.5–10	0.1
colsample_bytree	Fraction of features for each tree	0.5–0.9	0.714
colsample_bylevel	Fraction of features per tree level	0.5–0.9	0.658
objective	Loss function	Fixed: reg:squarederror	reg:squarederror
eval_metric	Evaluation metric	Fixed: rmse	rmse
tree_method	Tree construction algorithm	Fixed: hist	hist
device	Processing device	Fixed: cpu	cpu

In terms of the MSE metric, the lowest value was achieved by the XGBRegressor (32.77), while both XGBRFRegressor and LSTM performed significantly worse, with results 181.32 and 147.98, respectively. A similar trend is observed in RMSE, where XGBRegressor reached 5.72, clearly outperforming LSTM (6.27) and especially XGBRFRegressor (13.46). When considering the MAE, the XGBRegressor again achieved the best result (3.79), while the LSTM and XGBRFRegressor showed nearly identical and substantially higher errors (12.16 and 12.11, respectively). Similar results were observed across the datasets from all three training installations. However, since these datasets were used during the training process, the final outcome presented in Table 4 refers exclusively to the data from a single test installation.

The R^2 score was the highest for the XGBRegressor (0.79), indicating strong model performance and goodness of fit. The XGBRFRegressor reached a moderate value of 0.54, suggesting a less precise model in compare to XGBRegressor. In contrast, the LSTM yielded a negative R^2 of -0.27, indicating that its predictions were worse than those of a simple baseline model. This could be due to overfitting, not enough samples in training data, or inadequate hyperparameter tuning in the context of this specific task.

The main analysis of the best model's performance was based on a plot comparing the actual values (x-axis) with the predicted values (y-axis). Figure 6 shows the comparison of predictions versus actual values for the test data. The test data comes from a photovoltaic installation with a capacity of 49.5 kWp.

In our work, we were able to demonstrate that XBG alone performs even better, unlike XGBRF, which, as our research has shown, does not perform significantly better than LSTM. In [25], the use of a hybrid model combining XGBoost and LSTM was shown to improve the performance of a conventional LSTM-based predictor. In the study, the XGBoost model was employed to calculate feature importance. Based on the results of this analysis, features were either retained or removed prior to training the LSTM model. The reported outcomes (e.g. in (26)) demonstrated that this hybrid approach yielded better results compared to the standalone LSTM model. XGBoost was used independently to predict PV power generation. The findings indicated that the XGBoost model performed favorably in comparison to the Random Forest algorithm, highlighting its potential as a strong standalone predictor in PV power forecasting tasks. In (29), the implementation of LSTM and XGBoost models was associated with a significant reduction in operational costs and CO_2 emissions, compared to scenarios in which these models were not utilized. Also in (30), a hybrid model integrating XGBoost and LSTM was proposed. The experimental results demonstrated that this configuration outperformed the GRU model, another recurrent neural network variant. In (31), four hybrid models were compared: GRF-LSTM-XGBoost, GRF-XGBoost, LSTM-XGBoost, and GRF-LSTM. The results showed that all configurations achieved R^2 values around 0.9 or higher, accompanied by low RMSE values, indicating high predictive accuracy. In recent literature, the integration of XGBoost and LSTM – whether employed individually or as part of hybrid architectures – has gained substantial attention. The combination leverages the strengths of both methods: XGBoost offers efficient feature selection and high performance in structured data prediction, while LSTM excels in capturing temporal dependencies within sequential data. However, our work demonstrated that using XGBoost alone, especially for weather data, yields noticeably better results than using LSTM or XGBFR alone.

Table 4. Comparison of quality metrics for XGBRegressor, XGBRFRegressor, and LSTM for test installation (49.5 kWp)

Metric	XGBRegressor	XGBRFRegressor	LSTM
MSE	32.77	181.32	147.98
RMSE	5.72	13.46	6.27
MAE	3.79	12.11	12.16
R^2	0.79	0.54	-0.27

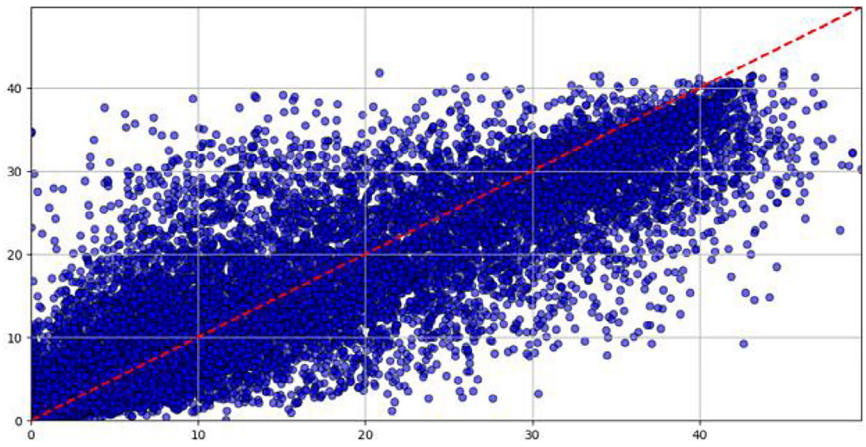

Figure 6. Comparison of forecasts with actual data. The red line symbolizes the same values for both axes (ideal scenario)

CONCLUSIONS

This paper presented the problem of short-term photovoltaic power forecasting using predictive models based on machine learning and deep learning techniques. The raw data underwent a comprehensive preparation process, including cleaning, preprocessing, normalization, and standardization. The models were trained and tested using real-world datasets collected from four PV installations located in Poland.

Our research has demonstrated that, among the tested approaches (XGB XGBRF, LSTM), the XGBRegressor model based on gradient boosting yielded the most accurate results (obtained RMSE was 5.72 where the next value for the LSTM was 6.27). This confirms its effectiveness in capturing complex relationships in the data and highlights its applicability in PV power forecasting tasks. In contrast, the XGBRFRegressor model, which utilizes random forests instead of gradient boosting, did not perform as well, being even worse than the previously mentioned LSTM. This finding is particularly noteworthy, as it suggests that not all tree-based ensemble methods are equally effective for this forecasting problem. The observed underperformance of the LSTM-based neural network model can largely be attributed to the limited size of the available training dataset. The one-year observation period likely constrained the model's ability to learn seasonal and long-term temporal patterns. Furthermore, our results suggest that XGBoost-based models can serve as a strong baseline in PV forecasting tasks, particularly when data volume is limited. This indicates high added value, especially in the context of production prediction for new installations and wherever access to historical data is not possible or training in such a large time window is unprofitable (e.g. free predictive models on inverter manufacturers' websites).

Another noteworthy added value of this work is the identification of the relationship between weather input parameters, which can be useful in selecting features relevant to energy production prediction. Although such studies exist, the results may be linked to the data source and a specific latitude, so creating them independently for specific data is always a good starting point for meteorological data analysis.

The insights gained through our study also point to the potential of hybrid solutions. Future work may focus on integrating XGBRegressor with neural networks incorporating LSTM layers to better capture temporal dependencies while preserving strong feature learning capabilities. Based on our findings, extending the dataset – both in terms of time span and measurement resolution – is expected to significantly improve model performance, especially for deep learning architectures – which – given the ongoing cooperation and continuous data collection from the installations will be possible in the future. Moreover, integrating probabilistic forecasting techniques may help quantify uncertainty, which is particularly relevant for grid operators and energy planning systems. These directions offer promising opportunities for enhancing the reliability and practical value of PV power prediction models.

Acknowledgements

This work was supported by Polish Ministry of Science and Higher Education by Young Researchers funds of Department of Telecommunications and Teleinformatics Silesian University of Technology.

REFERENCES

1. IRENA – International Renewable Energy Agency [Internet]. 2025 [cited 2025 Jun 26]. Available from: https://www.irena.org/
2. Beithou N, Mansour MA, Abdellatif N, Alsaqoor S, Tarawneh S, Jaber AH, et al. Effect of the residential photovoltaic systems evolution on electricity and thermal energy usage in Jordan. Adv Sci Technol Res J. 2023 Jun 1;17(3):79–87.
3. Pfeifer A, Krajačić G, Ljubas D, Duić N. Increasing the integration of solar photovoltaics in energy mix on the road to low emissions energy system – Economic and environmental implications. Renew Energy. 2019 Dec;143:1310–7.
4. Hanzel K. Analysis of financial losses and methods of shutdowns prevention of photovoltaic installations caused by the power grid failure in Poland. Energies. 2024;17(4):1–18.
5. Hou Y, Vidu R, Stroeve P. Solar energy storage methods. Ind Eng Chem Res. 2011 Aug 3;50(15):8954–64.
6. Glaa R, Jeddi N, Lakhoua N, Amraoui LE. Application of system modeling and simulation of the photovoltaic production. Adv Sci Technol Res J. 2017 Sep 3;11(3):48–55.
7. Rappaport P. The photovoltaic effect and its utilization. Sol Energy. 1959 Dec;3(4):8–18.
8. Copeland AWallace, Black OD, Garrett AB. The Photovoltaic Effect. Chem Rev. 1942 Aug 1;31(1):177–226.
9. Jasiński M, Leonowicz Z, Jasiński J, Martirano L, Gono R. PV Advancements & Challenges: Forecasting Techniques, Real Applications, and Grid Integration for a Sustainable Energy Future. In: 2023 IEEE International Conference on Environment and Electrical Engineering and 2023 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe) [Internet]. Madrid, Spain: IEEE; 2023 [cited 2025 Jun 26]. 1–5. Available from: https://ieeexplore.ieee.org/document/10194796/
10. Chakraborty S, Kumar S, Tripathi B, Saini ML. AI Explainable for Forecasting Crop Production Affected by Weather. In: 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE) [Internet]. Gautam Buddha Nagar, India: IEEE; 2024 [cited 2025 Jun 26]. 1–6. Available from: https://ieeexplore.ieee.org/document/10593021/
11. Sangrody H, Zhou N, Zhang Z. Similarity-Based Models for Day-Ahead Solar PV Generation Forecasting. IEEE Access. 2020;8:104469–78.
12. Munir MA, Khattak A, Imran K, Ulasyar A, Khan A. Solar PV Generation Forecast Model Based on the Most Effective Weather Parameters. In: 2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE) [Internet]. Swat, Pakistan: IEEE; 2019 [cited 2025 Jun 26]. 1–5. Available from: https://ieeexplore.ieee.org/document/8940664/
13. Rao S. Solar engineering of thermal processes. NASA STIRecon Tech Rep A [Internet]. 1980 Jan 1 [cited 2025 Jun 26]; Available from: https://www.academia.edu/1361110/Solar_engineering_of_thermal_processes
14. De Soto W, Klein SA, Beckman WA. Improvement and validation of a model for photovoltaic array performance. Sol Energy. 2006 Jan;80(1):78–88.
15. Alzahrani A, Shamsi P, Dagli C, Ferdowsi M. Solar irradiance forecasting using deep neural networks. Procedia Comput Sci. 2017;114:304–13.
16. Tunnicliffe Wilson G. Time Series Analysis: Forecasting and Control,5th Edition, by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015. Published by John Wiley and Sons Inc., Hoboken, New Jersey, pp. 712. J Time Ser Anal. 2016 Mar 1;37:n/a-n/a.
17. Dudek G. STD: A seasonal-trend-dispersion decomposition of time series. IEEE Trans Knowl Data Eng. 2023 Oct 1;35(10):10339–50.
18. Feng Y, Wang S. A forecast for bicycle rental demand based on random forests and multiple linear regression. In: 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS) [Internet]. Wuhan, China: IEEE; 2017 [cited 2025 Jun 26]. 101–5. Available from: http://ieeexplore.ieee.org/document/7959977/
19. Xia L, Yaomei G, Weiwei S. Forecast to Textile and Garment Exports Based on Holt Model. In: 2010 International Conference of Information Science and Management Engineering [Internet]. Shaanxi, China: IEEE; 2010 [cited 2025 Jun 26]. 274–7. Available from: http://ieeexplore.ieee.org/document/5572530/
20. Santos RS, Pour Yousefian Barfeh D. Time Series Forecasting for Issuance of Alien Employment Permits by Nationality in the Philippines Using Holt-Winters Method. In: 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE) [Internet]. Dubai, United Arab Emirates: IEEE; 2019 [cited 2025 Jun 26]. 240–5. Available from: https://ieeexplore.ieee.org/

document/9004243/

21. Forecasting: Principles and Practice (2nd ed) [Internet]. [cited 2025 Jun 26]. Available from: https://otexts.com/fpp2/
22. Gupta A, Kumar A. Mid Term Daily Load Forecasting using ARIMA, Wavelet-ARIMA and Machine Learning. In: 2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe) [Internet]. Madrid, Spain: IEEE; 2020 [cited 2025 Jun 26]. 1–5. Available from: https://ieeexplore.ieee.org/document/9160563/
23. Sheoran S, Pasari S, Shukla S. Application of Window Sliding ARIMA in Wind Speed and Solar Irradiance Forecasting. In: 2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE) [Internet]. MANGALORE, India: IEEE; 2022 [cited 2025 Jun 26]. 321–5. Available from: https://ieeexplore.ieee.org/document/10054263/
24. Phan QT, Wu YK, Phan QD. Short-term Solar Power Forecasting Using XGBoost with Numerical Weather Prediction. In: 2021 IEEE International Future Energy Electronics Conference (IFEEC) [Internet]. Taipei, Taiwan: IEEE; 2021 [cited 2025 Jun 26]. 1–6. Available from: https://ieeexplore.ieee.org/document/9661874/
25. Chen X, Wu Y, He X. Long Short-Term Memory Network PV Power Prediction Incorporating Extreme Extreme Gradient Boosting Algorithm. In: 2022 12th International Conference on Power and Energy Systems (ICPES) [Internet]. Guangzhou, China: IEEE; 2022 [cited 2025 Jun 26]. p. 821–5. Available from: https://ieeexplore.ieee.org/document/10073236/
26. Hong Y, Yang J, Yang Z, Yan J. PV Power Prediction Based on XGBoost Algorithm. In: 2023 IEEE 5th International Conference on Civil Aviation Safety and Information Technology (ICCASIT) [Internet]. Dali, China: IEEE; 2023 [cited 2025 Jun 26]. p. 930–3. Available from: https://ieeexplore.ieee.org/document/10351690/
27. Kurnala V, Naik SA, Surapaneni DC, Reddy ChB. Hybrid Detection: Enhancing Network & Server Intrusion Detection Using Deep Learning. In: 2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA) [Internet]. Hamburg, Germany: IEEE; 2023 [cited 2025 Jun 26]. p. 248–51. Available from: https://ieeexplore.ieee.org/document/10346699/
28. Mudaser M, Krathu W, Jaiyen S. Automated Ensemble Deep Learning for Chest X-Ray Covid-19 Image Classification Using Multiple Hyperparameter Optimizations. In: 2023 27th International Computer Science and Engineering Conference (ICSEC) [Internet]. Samui Island, Thailand: IEEE; 2023 [cited 2025 Jun 26]. 348–53. Available from: https://ieeexplore.ieee.org/document/10329679/
29. Divakar B, Ali A, Sengupta D, Sebastiao BG, Priyatharsini GS, Babu BH. Big Data Analytics in Renewable Energy Management for Intelligent Smart Grid Operation. In: 2025 13th International Conference on Smart Grid (icSmartGrid) [Internet]. 2025 [cited 2025 Aug 7]. 410–5. Available from: https://ieeexplore.ieee.org/abstract/document/11071786
30. Xue J, Hu X, Chen H, Zhou G. Research on LSTM-XGBoost Integrated Model of Photovoltaic Power Forecasting System. In: 2022 14th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC) [Internet]. Hangzhou, China: IEEE; 2022 [cited 2025 Aug 7]. 22–5. Available from: https://ieeexplore.ieee.org/document/9903235/
31. Zhao T, Huang L. A Short-Term Photovoltaic Power Forecasting Method Based on Meteorological Data Using GRF-LSTM-XGBoost Model. In: 2025 7th Asia Energy and Electrical Engineering Symposium (AEEES) [Internet]. Chengdu, China: IEEE; 2025 [cited 2025 Aug 7]. 1068–73. Available from: https://ieeexplore.ieee.org/document/11019969/