Methodology

The methodology adopted in this project follows a structured data science workflow tailored to the forecasting of renewable energy production. It integrates classical time-series modeling with modern machine learning techniques to ensure accuracy, robustness, and interpretability. The approach is divided into seven key phases: data acquisition, preprocessing, exploratory data analysis, feature engineering, model development, evaluation, and deployment.

Step 1: Data Acquisition

The dataset used in this study is sourced from the National Renewable Energy Laboratory (NREL), which provides reliable and comprehensive meteorological and solar irradiance data. NREL's database includes ground-based measurements and satellite-derived estimations of global horizontal irradiance (GHI), direct normal irradiance (DNI), and diffuse horizontal irradiance (DHI), which are key variables in predicting solar energy generation. These datasets also contain important meteorological attributes like temperature, wind speed, relative humidity, and cloud coverage, collected from various monitoring stations across the U.S.

The data spans multiple years (e.g., 2000–2026) and is publicly available in CSV format. Each file corresponds to a specific station and year. The inclusion of multiple meteorological features provides a strong foundation for developing a robust and predictive model that accounts for diverse climate conditions and temporal patterns.

Step 2: Data Preprocessing

Given the raw format and size of the datasets, a comprehensive data preprocessing stage is necessary to ensure consistency and quality:

- Merging Files: All annual files are concatenated into a single unified dataset.
- Date Parsing: Timestamp columns are standardized to Python datetime formats.
- Handling Missing Values: Missing entries are imputed using interpolation, forward fill, and statistical imputation where needed. Rows with a high percentage of null values are discarded.
- Unit Normalization: Temperature and irradiance units are validated to ensure consistency across files.
- Outlier Detection: Visual inspections using boxplots and statistical thresholds (e.g., IQR) are applied to detect and filter extreme outliers that could bias model training.

Step 3: Exploratory Data Analysis (EDA)

EDA is a crucial step to understand the underlying trends, periodicities, and anomalies in the data. The following techniques are employed:

- Time-Series Visualization: Line plots display variations in GHI, temperature, and wind speed across months and years.
- Correlation Analysis: Heatmaps and scatter plots reveal correlations between meteorological features and energy output.

- Statistical Summaries: Descriptive statistics help identify the distribution of values and data range.
- Seasonality Checks: Time decomposition is applied to identify seasonal, trend, and residual components in the solar irradiance data.

Step 4: Feature Engineering

To enhance model performance, raw features are transformed into informative predictors:

- Temporal Features: Year, month, day, day of week, and hour extracted from datetime
- Cyclical Encoding: Sine and cosine transformation of time features (e.g., day-of-year, month) to preserve seasonal cycles
- Lag Features: Past values (1-day, 7-day, 30-day lags) of GHI, temperature, and other key variables
- Rolling Statistics: Moving averages and rolling standard deviations to smooth volatility and capture trends
- Weather Conditions: Inclusion of categorical weather descriptions if available (e.g., “clear”, “cloudy”)

These engineered features significantly improve the predictive capability of time-series models and enable them to learn both short-term fluctuations and long-term patterns.

Step 5: Model Development

A variety of forecasting models are implemented and compared:

- ARIMA (Autoregressive Integrated Moving Average): This serves as a baseline linear model capturing autoregressive and moving average components in stationary time-series data.
- LSTM (Long Short-Term Memory): A recurrent neural network model capable of learning temporal dependencies. LSTM layers are stacked with dropout regularization and trained using sequences of historical values.
- XGBoost (Extreme Gradient Boosting): A tree-based ensemble model that efficiently handles structured data and supports feature importance analysis. Lagged and encoded features serve as inputs.
- Prophet: Developed by Facebook, Prophet automates time-series decomposition and is highly effective for forecasting in the presence of trend shifts and multiple seasonalities.

Each model undergoes training and validation using a train-test split (e.g., 80%-20%) or time-series cross-validation. Hyperparameters are tuned using techniques such as grid search, Bayesian optimization, and early stopping.

Step 6: Model Evaluation

Models are evaluated using a combination of quantitative and visual methods:

- Root Mean Square Error (RMSE): Sensitive to large errors and commonly used for regression tasks
- Mean Absolute Error (MAE): Measures average magnitude of errors, regardless of direction
- R-squared (R^2): Explains the proportion of variance in the dependent variable accounted for by the model
- Residual Analysis: Residual plots and error distributions to assess the model's consistency across time

Evaluation also includes a comparison across models using visual forecast overlays, enabling interpretation of prediction quality.

Step 7: Visualization, Interpretation, and Deployment

Post-evaluation, models are interpreted and communicated effectively through visual tools:

- Forecast Plots: Show actual vs. predicted values for test sets
- Decomposition Charts: Prophet's trend, seasonality, and holiday effects visualization
- Feature Importance Plots: Especially for XGBoost, identifying which features influence predictions the most
- Heatmaps and Time Plots: Displaying patterns in meteorological variables and their impact on production

The final outputs (code, models, datasets, and documentation) are version-controlled using GitHub and backed up on OneDrive. A ReadMe file guides users through installation, dataset format, dependencies, and execution. All visuals and interpretations are summarized in the final project report and PowerPoint presentation for academic submission.

This comprehensive methodology ensures the credibility, transparency, and reproducibility of the forecasting system developed in this project. It aligns with academic best practices and industry standards for energy analytics and machine learning.