Optimizing Photovoltaic Power Forecasting Through Machine Learning Algorithms

Senye Zhang
Electrical and Computer Engineering
Lakehead University
Thunder Bay, Canada
ORCID:0009-0005-0121-2771

***Abstract*—This paper focuses on improving real-time photovoltaic (PV) power forecasting to support more efficient grid dispatch. By utilizing local weather data, such as solar irradiance, temperature, and historical PV power output, we implemented a hybrid machine learning approach that integrates Long Short-Term Memory (LSTM), XGBoost, and Random Forest models. The predictions from these base models were combined using a stacking ensemble method, where a linear regression meta-learner was employed to generate the final forecast. The results demonstrate that this ensemble approach significantly reduces prediction errors compared to individual models, enhancing the accuracy and reliability of PV power forecasts. This research contributes to the optimization of grid management and addresses challenges associated with integrating renewable energy sources into the power grid.**

***Index Terms*—photovoltaic power forecasting, machine learning, solar energy prediction, renewable energy, power grid scheduling optimization, LSTM, XGBoost, random forest, energy management, stochastic optimization, time series forecasting**

I. Introduction

The rapid deployment of photovoltaic (PV) systems worldwide has significantly transformed the global energy landscape. As PV capacity continues to expand, traditional electricity demand patterns, once shaped by factors such as cooling loads during midday, are shifting. The large influx of solar power has introduced new challenges for power grids, including midday power surges and negative load conditions in some regions. Accurate PV power forecasting is crucial to maintaining grid stability and optimizing dispatch strategies. However, existing forecasting methods often struggle to account for the variability of PV generation under changing weather conditions. This paper addresses these challenges by leveraging machine learning algorithms to predict real-time PV power output using local meteorological data, with the goal of enhancing grid stability and informing more effective dispatch strategies in the face of growing renewable energy penetration.

II. Literature Review

Machine learning has become a crucial tool in photovoltaic (PV) power forecasting, providing significant improvements in accuracy compared to traditional methods. Unlike statistical models such as ARMA and ARIMA, which often struggle to capture nonlinear patterns, machine learning techniques like Random Forest, XGBoost, and Long Short-Term Memory (LSTM) networks excel at modeling complex relationships between weather variables and PV output. LSTM is particularly effective for time-series forecasting, capturing temporal dependencies, but it requires large datasets and computational resources, making it prone to overfitting when data is limited. XGBoost, known for its efficiency and ability to model nonlinear interactions, is highly scalable, yet it also risks overfitting with small datasets and requires careful hyperparameter tuning. Random Forest is robust against overfitting and easy to implement but may not handle highly complex relationships as well as boosting methods like XGBoost. A growing trend is the development of hybrid models that combine these algorithms to leverage their individual strengths while mitigating their weaknesses. However, despite the advantages machine learning offers in PV forecasting, these methods still rely on high-quality data and intensive tuning, which can limit their scalability and generalizability in certain applications.

III. Methodology

In this study, we adopted a hybrid modeling approach that combines Long Short-Term Memory (LSTM), XGBoost, and Random Forest to predict the target variable. First, we generated a synthetic dataset with feature matrix X and target variable y, which was then split into training and testing sets. We trained three base models: an LSTM model using the ‘keras‘ library to capture temporal dependencies in time-series data, an XGBoost model using the ‘xgboost‘ package to handle non-temporal structured data, and a Random Forest model using the ‘randomForest‘ package to capture complex nonlinear relationships. After training, each base model produced predictions for both the training and testing sets. These predictions were then used as features to train a meta-learner, with a simple linear regression model acting as the meta-learner. This stacking method allowed us to combine the strengths of each base model and improve the final forecast accuracy. The performance of the stacked model was evaluated using Mean Squared Error (MSE) on both the training and testing sets. The approach takes advantage of LSTM's ability to model temporal dependencies, XGBoost and Random Forest's strength in handling complex nonlinear relationships, and the stacking technique's ability to improve overall robustness and accuracy by combining multiple model predictions. This

Identify applicable funding agency here. If none, delete this.

hybrid methodology ensures that both time-dependent and non-time-dependent relationships are well captured, leading to better predictive performance.

A. Data Collection and Preprocessing

The dataset used in this study contains multiple weather and environmental features along with the target variable, which is the power output (power). It consists of 15 columns, where the first column represents the timestamp, and the last column (power) is the target variable we aim to predict. The middle 13 columns serve as the input features and include various meteorological variables such as global irradiance, direct irradiance, temperature, humidity, wind speed, wind direction, and atmospheric pressure, derived from both numerical weather prediction (NWP) data and local meteorological data (LMD). For the purpose of model development, 80% of the data is used for training the models, while the remaining 20% is reserved for validation to evaluate the model's performance. The preprocessing steps involve handling missing data and reshaping the input features to ensure compatibility with machine learning models, especially time-series models like LSTM that require a specific input format.

B. Machine Learning Models

- Base Models
  We utilized three distinct machine learning models as base learners:
  1. **LSTM**: LSTM is a neural network architecture designed to handle time-series data by capturing temporal dependencies. Its prediction can be formulated as:

$$\hat{y}_{LSTM} = f_{LSTM}(X)$$

  where $f_{LSTM}(X)$ represents the LSTM model that takes the input feature matrix X (13 selected features) and outputs the prediction $\hat{y}_{LSTM}$.
  2. **XGBoost**: XGBoost is a powerful gradient-boosted decision tree model that efficiently handles non-linear relationships in the data. Its prediction is defined as:

$$\hat{y}_{XGB} = f_{XGB}(X)$$

  where $f_{XGB}(X)$ denotes the XGBoost model's output for the input feature matrix X.
  3. **Random Forest**: Random Forest is an ensemble learning method based on decision trees, which models complex, non-linear relationships between input features. The prediction from the Random Forest model is expressed as:

$$\hat{y}_{RF} = f_{RF}(X)$$

  where $f_{RF}(X)$ is the Random Forest model's prediction.
- Stacking Model
  After obtaining predictions from each base model, we applied a stacking method to combine these predictions. We used a linear regression model as the meta-learner to generate the final prediction based on the outputs of the base models. The stacking model can be formulated as:

$$\hat{y} = g(\hat{y}LSTM, \hat{y}XGB, \hat{y}_{RF})$$

  where $g(\cdot)$ represents the meta-learner, which combines the predictions from LSTM, XGBoost, and Random Forest to produce the final output $\hat{y}$.
- Linear Combination in the Stacking Model
  The meta-learner in this study is a linear regression model. Thus, the final prediction is a weighted combination of the predictions from the base models:

$$\hat{y} = w_1\hat{y}LSTM + w_2\hat{y}XGB + w_3\hat{y}RF$$

  where w_1, w_2, w_3 are the learned weights that determine the contribution of each base model to the final prediction. $\hat{y}LSTM, \hat{y}XGB, \hat{y}RF$ are the respective predictions from the LSTM, XGBoost, and Random Forest models.

C. Model Evaluation Metrics

The performance of the stacked model was evaluated using Mean Squared Error (MSE) on both the training and testing datasets. The MSE is calculated as:

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

where y_i is the actual value and $\hat{y}_i$ is the predicted value from the model. This hybrid methodology leverages LSTM's ability to model temporal patterns and the strength of XGBoost and Random Forest in handling complex, non-linear relationships. By combining these models through stacking, the final model benefits from the strengths of each base learner, resulting in improved predictive performance and robustness.

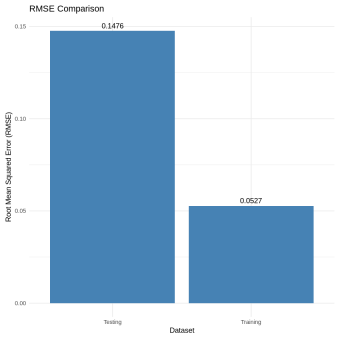

Fig. 1. RMSE Comparison

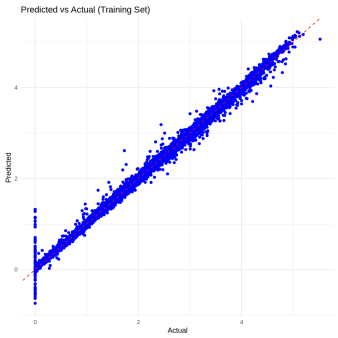

Fig. 2. Training Set

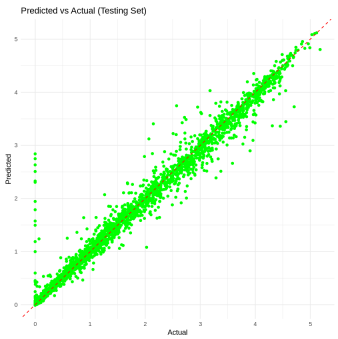

Fig. 3. Testing Set

TABLE I
TABLE TYPE STYLES

Table Head	Table Column Head		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a]Sample of a Table footnote.

IV. RESULTS AND DISCUSSION

A. *Forecasting Accuracy Comparison*

B. *Impact of Weather Variables on Prediction*

C. *Implications for Power Grid Scheduling Optimization*

V. CONCLUSION

VI. FUTURE WORK

ACKNOWLEDGMENT

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.