Hybrid Deep Learning Framework for Solar Power Forecasting.

EE4750
Data Analytics in Power Systems

Presented by

- S. Vaishnavi - 210666H
- V. Nilesh - 210426A

Problem Statement

With the growing need for clean and renewable energy, solar power has become an important source of electricity. However, solar energy is not always consistent—it changes based on sunlight, temperature, humidity, clouds, and dust. This makes it hard to manage energy storage and grid stability

Why Use Machine Learning?

Traditional models use complex formulas and depend on specific locations. Instead, machine learning can learn patterns from data, handle complex changes, and give fast and accurate predictions.

Objective

To build a ML model using CNN and XGBoost that can accurately predict solar power output based on input features, for both time series and non-time series datasets.

Dataset Overview, Structure

1. Time Series Dataset
    - Each row has a timestamp (like hourly or daily).
    - Data is in order, showing how features change over time.
    - Helps the model learn from the past to predict the future.

2. Non-Time Series (Tabular) Dataset
    - Each row is independent, no specific time attached.
    - Used when we only care about the current conditions.
    - Works well for regression models.

Target Output

- Power Output (kW) — the amount of electricity generated.

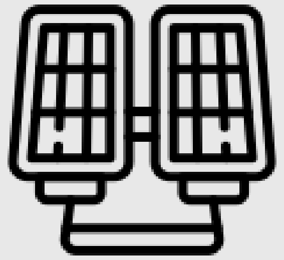
Model Architecture Overview

We propose a hybrid deep learning model that combines the strengths of both Convolutional Neural Networks (CNN) and XGBoost for accurate solar power prediction.

- CNN is used for automatic feature extraction from input data, capturing complex spatial or temporal patterns.
- The extracted features are then passed through a flattening layer to convert them into a 1D feature vector.
- These features are fed into XGBoost, which acts as a regression head to predict the final solar power output.

This structure works efficiently for both:

- Time Series Data (with sequential input)
- Tabular Data (reshaped into 2D matrices for CNN input)



Hybrid Model Mechanism

[Input Data]

↓

[Reshape (if needed)]

↓

[CNN Layers]

↓

[Flatten Layer]

↓

[XGBoost Regressor]

↓

[Predicted Power Output (kW)]

Prediction of photovoltaic power generation based on a hybrid model

Xiaohua Zhang[1], Yuping Wu[1], Yu Wang[2]*, Zhirui Lv[1], Bin Huang[1], Jingzhong Yuan[3], Jingyu Yang[1], Xinsheng Ma[4], Changyuan Li[5] and Lianchao Zhang[2]

[1]State Grid Jibei Electric Power Company Limited, Beijing, China, [2]Beijing KeDong Electric Power Control System Co., Ltd., Beijing, China, [3]State Grid Jibei Electric Power Company Limited Economic Research Institute, Beijing, China, [4]State Grid Jibei Electric Power Research Institute, Beijing, China, [5]State Grid Qinhuangdao Power Supply Company, Qinhuangdao, China

In order to fully exploit the relationship between temporal features in photovoltaic power generation data and improve the prediction accuracy of photovoltaic power generation, a photovoltaic power generation forecasting method is proposed based on a hybrid model of the convolutional neural network (CNN) and extreme gradient boost (XGBoost). Taking the historical data of China's photovoltaic power plants as a sample, the high-dimensional mapping relationship of photovoltaic power generation variables is extracted based on the convolutional layer and pooling layer of the CNN network to construct a high-dimensional time-series feature vector, which is an input for the XGBoost. A photovoltaic power generation prediction model is established based on CNN-XGBoost by training CNN and XGBoost parameters. Since it is difficult for a single model to achieve optimal prediction accuracy under different weather conditions, the *k*-means clustering algorithm is used to group the power datasets and train independent models to improve prediction accuracy. Through the actual data verification of photovoltaic power plants, the proposed photovoltaic power generation prediction model can accurately predict the power, which shows high prediction accuracy and generalization ability compared with other methods.

https://www.frontiersin.org/journals/energy-research/articles/10.3389/fenrg.2024.1411461/full

CNN for Feature Extraction on Tabular Data

Task : Prediction

Technical Summary:
Each input row is reshaped into a 2D matrix and passed through CNN layers. CNN learns useful patterns between features and creates a meaningful feature vector. This vector is then given to XGBoost to predict the final power output.

[Input Row (1D Vector)]

↓

[Reshape to 2D Matrix]

↓

[CNN Layers (Conv1D/Conv2D + Activation)]

↓

[Flattened Feature Vector]

↓

[XGBoost Regressor]

↓

[Predicted Power Output (kW)]

CNN for Time Series Modeling : Forecasting

[Time Series Window (Time Steps × Features)]

↓

[2D Input Matrix]

↓

[CNN Layers (Temporal + Feature Pattern Extraction)]

↓

[Flattened Feature Vector]

↓

[Regression Layer / XGBoost]

↓

[Predicted Power Output (kW)]

We feed a fixed-size time window (e.g., 10 time steps × 8 features) into a CNN as a 2D input.

CNN captures both temporal and spatial (feature-wise) dependencies in the data.

This produces a feature map that highlights important patterns.

After flattening, the output is passed to a regression head (e.g., XGBoost or Dense Layer) to Forecast the next power output value.

Research Gaps ,Proposed Improvements

1.Tabular [deeper CNN exploration]

Paper ↣ CNN + XGBoost (row-wise only)

↧

Our Gap ↣ Limited CNN exploration in tabular data

↧

Improvement ↣ Add GAP + Dense layers ↣ XGBoost

2.Time Series [proper temporal modeling]

Paper ↣ Treated as row-by-row (no temporal modeling)

↧

Our Gap ↣ Missed temporal dependencies

↧

Improvement ↣ Sliding window ↣ CNN + XGBoost + Explore GRU / RNN / LSTM

Clustering Strategy for Dataset

Paper’s approach:
K-means clustering was used because groups (weather conditions) were known.

Our situation: Dataset will be given ➻ clusters not known beforehand.

Plan

1. If dataset is homogeneous ➻ use single CNN–XGBoost model.
2. If dataset is heterogeneous ➻
    - Apply clustering (Elbow).
    - Train a separate CNN–XGBoost model per cluster.
    - During inference ➻ new data is assigned to the correct cluster and predicted using its model.

Tools Flow

Python

↓

NumPy ,Pandas

↓

TensorFlow or PyTorch (for CNN)

↓

XGBoost (for Regression)

↓

MAE, RMSE (for Evaluation)

Workflow Steps

1. Data Collection
   - Gather solar data with environmental features.
2. Data Preprocessing
   - Handle missing values, normalization, reshaping.
3. CNN Feature Extraction.
   - Reshape input, apply CNN to extract patterns.
4. Model Training
   - Use XGBoost on extracted features to train regression model.
5. Model Evaluation
   - Measure prediction performance using MAE , RMSE.

Towards a Sustainable Future with AI + Solar Energy !

Thank you!