# Solar Energy Production

**Author**: Dallin Moore
**Project**: Project 3 - DATA 5610

## Overview

The project explores the feasibility of using LSTMs to forecast inverter AC power output from historical weather and power data. Success is evaluated based on reducing the MAE below a naive persistence model baseline. The model was able to outperform the baseline suggesting feasibility of a model to be used in production.

## Approach

A time-series forecasting model was built using an LSTM architecture trained on inverter-level data. The dataset consists of weather and power generation readings for two solar plants over a 34-day period.

### Data Preparation

The dataset used in this study was sourced from [Kaggle](#) and includes weather and power generation data for two solar power plants in India, recorded over a 34-day period. The data consists of four CSV files: two for plant-level and inverter-level power generation, and two for weather data corresponding to each plant. However, because the relationships were substantially different between Plant 1 and Plant 2, only data from Plant 1 was used.

To prepare the data for modeling with an LSTM:

- **Inverter-specific segmentation:** Each inverter was treated as an independent time series. The corresponding weather data was merged based on matching timestamps
- **Target and features:** The target variable selected for prediction was `AC_POWER`, though `DC_POWER` could also be a viable alternative. The features used for model training were:
    - `AMBIENT_TEMPERATURE`
    - `MODULE_TEMPERATURE`
    - `IRRADIATION`
- **Sequencing:** The data was reshaped into sequences of length **48**, corresponding to **12 hours of historical data** given 15-minute sampling interval.
- **Normalization:** Feature values were scaled between 0 and 1 using MinMax normalization, applied separately for each inverter to preserve the relative patterns within individual time series.
- **Splitting:** The dataset was split into training, validation, and test sets in a time-aware fashion, maintaining chronological order to prevent information leakage.
    - **70%** of the data was used for training,
    - **15%** for validation,
    - and the remaining **15%** for testing.

# Model Architecture

The model was designed as a sequential LSTM network to capture temporal dependencies in the inverter-level time series data.

The model was compiled using the **Mean Squared Error (MSE)** loss function, the **RMSPROP** optimizer, and **Mean Absolute Error (MAE)** as an evaluation metric. The output layer uses a linear activation function to ensure continuous-valued predictions.

| Layer (type) | Output Shape | Param # | Activation |
|---|---|---|---|
| InputLayer | (None, 48, 3) | 0 | |
| LSTM | (None, 48, 64) | 17,408 | tanh |
| Dropout | (None, 48, 64) | 0 | |
| LSTM | (None, 32) | 2,112 | tanh |
| Dense | (None, 1) | 17 | Linear |

## Model Tuning

The model was fine-tuned to reduce the Mean Absolute Error (MAE) below the naive persistence baseline of 58.7. Hyperparameter tuning focused on optimizing the LSTM unit size, dropout rate, learning rate, and sequence length. While initially having a sequence length of 96 seemed optimal (96 time steps accounts for 24 hours), cutting the sequence in half immediately resulted in better performance. Likewise epochs were increased from 20 to 30 to enhance model performance.

# Results

The final LSTM model achieved a **test set Mean Absolute Error (MAE) of 55.8**, outperforming the naive baseline of 58.7, which simply predicted that the AC power at time $t$ would be the same as at time $t–1$. This indicates that the model was able to learn meaningful temporal patterns from the input features (irradiation, ambient temperature, and module temperature) to make more accurate predictions than a basic persistence strategy.

# Conclusion

This project demonstrated the feasibility of using a deep learning-based sequence model (LSTM) to forecast solar energy production at the inverter level. After pre-processing the data into time series sequences and training an LSTM network, the model achieved better accuracy than a naive baseline, validating the approach. While the improvement in MAE was relatively modest, it is a meaningful step toward deploying forecasting tools in production environments.