Anomaly Detection and Forecasting of Solar Power Plant Performance across Deep Learning Models

Purva Singh
MEng – Systems and Technology, W Booth School of Engineering Practice and Technology,
Hamilton, Ontario
singhp36@mcmaster.ca

Chris Xavier Mathias
MEng – Systems and Technology, W Booth School of Engineering Practice and Technology,
Hamilton, Ontario
mathiac@mcmaster.ca

Vipin Chandran Muthirikkaparambil
MEng – Systems and Technology, W Booth School of Engineering Practice and Technology,
Hamilton, Ontario
muthiriv@mcmaster.ca

Suraj Ramesh
MEng – Systems and Technology, W Booth School of Engineering Practice and Technology,
Hamilton, Ontario
ramess18@mcmaster.ca

***Abstract*—This project presents a comprehensive deep learning-based framework for short-term forecasting and anomaly detection in solar power plants. Leveraging time-series data from two Indian solar facilities, the study utilizes advanced neural architectures including Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), Feedforward Neural Network (FFNN), and a hybrid CNN-LSTM model to forecast AC power output based on historical operational and weather data. Each model is evaluated in terms of its ability to capture temporal trends and forecast power generation under varying environmental conditions. Additionally, an unsupervised autoencoder-based anomaly detection system is developed to identify abnormal inverter behavior, with further localization performed using Dynamic Time Warping (DTW). The combined forecasting and anomaly detection framework supports intelligent monitoring, enabling timely maintenance and enhancing the reliability of energy generation.**

Keywords*—Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), Convolutional Neural Network (CNN), auto-encoder, Dynamic Time Warping (DTW)

I. Introduction

The increasing global emphasis on renewable energy has positioned solar power as a critical contributor to sustainable electricity generation. However, the inherent variability of solar energy, influenced by dynamic environmental factors such as solar irradiation, temperature, and panel efficiency, poses significant challenges to accurate performance forecasting and anomaly detection. In large-scale photovoltaic (PV) installations, reliable predictive systems are essential for optimizing power output, ensuring operational efficiency, and minimizing downtime due to undetected faults or inefficiencies. This study addresses these challenges by utilizing a data-driven solution that combines deep learning-based forecasting models with unsupervised anomaly detection techniques. Using high-resolution inverter-level and plant-level sensor data from two solar power plants in India, the project investigates the performance of Feedforward Neural Networks (FFNN), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, and hybrid CNN-LSTM architectures in modeling temporal patterns in AC power output. In parallel, an autoencoder-based anomaly detection system is designed to identify deviations from normal inverter behavior, with Dynamic Time Warping (DTW) applied to localize the specific time intervals of anomalies. By integrating these approaches, the framework offers both proactive forecasting and reactive anomaly identification, contributing to smarter, data-centric energy management in solar PV systems.

II. Literature Review

Solar photovoltaic (PV) power generation is inherently dependent on various environmental parameters, making its accurate forecasting a complex but crucial task. Recent advancements in deep learning have enabled the development of more robust and intelligent forecasting frameworks, which can model non-linear and time-dependent patterns more effectively than traditional statistical approaches. Miraftabzadeh et al. [1] proposed an unsupervised deep autoencoder model for detecting anomalies in photovoltaic (PV) power time-series data using only power output, addressing challenges posed by insufficient sensor information. Unlike traditional multivariate or supervised models, their approach relies solely on reconstruction error, making it highly applicable to low-instrumented PV systems. The model demonstrates robust anomaly detection capabilities without requiring weather or irradiance data, making it cost-effective and scalable. Sharma et al. [2] propose a novel long-term PV power forecasting method using an LSTM model optimized with Nadam, applied to a 250-kW solar installation in Bhopal, India. The model significantly outperformed traditional time-series methods (ARIMA, SARIMA) with accuracy gains of ~30–47%, and surpassed LSTM variants using other optimizers (e.g., RMSprop, Adam) by margins up to ~58%. It demonstrates that careful multiplier selection in LSTM training (specifically Nadam) can boost long-horizon forecast performance for large-scale PV power systems. Iheanetu (2022) [3] delivers a comprehensive, systematic review of data-driven solar PV power forecasting methods, critically analyzing artificial intelligence, machine learning, hybrid/ensemble models, and statistical techniques across varying time horizons. The study meticulously classifies approaches, ANNs, SVMs, deep learning, hybrids, highlighting their strengths, limitations, and performance trade-offs under different conditions. It underscores that hybrid models consistently outperform standalone methods, and emphasizes the need for standardized metrics, benchmark datasets, and thorough evaluation to reliably gauge forecasting accuracy. Lim et al. (2022) [4] introduce a parallel CNN-LSTM hybrid for PV power forecasting at a Busan, Korea plant, where the CNN classifies weather patterns and the LSTM learns power generation sequences accordingly. The model achieved 4.58% MAPE on sunny days and 7.06%

on cloudy days, outperforming standalone CNN or LSTM models in precision. Their approach effectively combines spatial features (weather conditions) with temporal dependencies, demonstrating improved reliability for solar plant operations under varying conditions. Giorgino [5] introduces the R package dtw, a comprehensive implementation of Dynamic Time Warping (DTW), enabling flexible time-series alignment under a variety of continuity constraints, windowing schemes, and local distance metrics. The package supports global and partial alignments, step patterns (e.g., Sakoe–Chiba, Itakura), and multivariate inputs, all accessible via a unified dtw () interface. It also features robust visualization tools (e.g., warping path plots and three-way alignment diagrams) to aid interpretation of alignments. Widely cited across domains, dtw has become a standard for both computing and graphically exploring optimal series alignments in research and applied time-series analysis. Khouili et al. [6] conduct a rigorous systematic literature review (SLR) of 26 deep learning–based solar and PV power forecasting studies, selected from an initial pool of 155 (via Web of Science). They find LSTM models used in about 32.7% of cases and CNNs in about 28.8%, with Wavelet Transform and Pearson correlation common for feature engineering, and ambient temperature, pressure, and humidity most frequently utilized as inputs. The review highlights key gaps such as limited interpretability, generalization issues, and the need for multi-source data fusion and recommends future research focus on robust, interpretable architectures and richer input integration to improve forecasting accuracy. Barhmi et al. [7] deliver an expansive review of solar forecasting methods, highlighting the central role of artificial intelligence, particularly neural networks, alongside regression, ensemble, and physics-based approaches for predictions across horizons from minutes to a full day. The study emphasizes integrations of satellite imagery, weather modeling, and historical sensor data to enhance predictive accuracy. It also underscores the importance of standardized datasets, benchmark methodologies, and unified error metrics to ensure fair model evaluations and meaningful comparisons.

III. Methodology

A. Data Collection

The dataset used for this project was sourced from Kaggle and comprises real-world solar power generation and sensor data collected over a period of 34 days from two different solar power plants located in India. Each plant contributes two separate datasets: one detailing power generation and the other capturing environmental sensor readings. These datasets are named as follows:

- Plant_1_Generation_Data.csv
- Plant_1_Weather_Sensor_Data.csv
- Plant_2_Generation_Data.csv
- Plant_2_Weather_Sensor_Data.csv

The generation data is recorded at the inverter level, where each inverter is responsible for converting the direct current (DC) output from multiple strings of solar panels into alternating current (AC). Inverters form the core operational units within a solar plant, making this data highly granular and valuable for performance evaluation and forecasting. In contrast, the sensor data is collected at the plant level from a strategically placed array of sensors. These sensors monitor environmental factors such as temperature and solar irradiation, which directly influence the efficiency of solar energy conversion. Each file includes timestamped observations, allowing for synchronized analysis between generation metrics and environmental conditions. The granularity of this data enables detailed modeling and insight generation for both anomaly detection and performance forecasting.

1) Variable Description

The variables in the dataset are grouped into two categories: Generation Data Variables and Sensor Data Variables. The Generation Data Variables for both Plant 1 and Plant 2 are as follows:

- DATE_TIME: The timestamp at which the reading was recorded. This is used to align data from different sources.
- PLANT_ID: A unique identifier for solar plant. Plant 1 is assigned ID 4135001, while Plant 2 has ID 4136001.
- SOURCE_KEY: A unique alphanumeric key identifying each inverter. Each inverter has multiple lines of solar panels and serves as a source of generation data.
- DC_POWER: The amount of power generated in direct current by the inverter at a given timestamp, measured in watts (W).
- AC_POWER: The amount of power output from the inverter in alternating current, after conversion from DC, measured in watts (W).
- DAILY_YIELD: The amount of energy generated by a particular inverter since the beginning of the day, measured in kilowatt-hours (kWh).
- TOTAL_YIELD: The cumulative amount of energy generated by the inverter over its lifetime, measured in kilowatt-hours (kWh).

Sensor Data Variables for both Plant 1 and Plant 2 are as follows:

- DATE_TIME: The timestamp at which the reading was recorded.
- PLANT_ID: A unique identifier for solar plant, matching the generation data.
- SOURCE_KEY: A unique key identifying the sensor unit. For each plant, there is one weather sensor unit.
- AMBIENT_TEMPERATURE: The temperature of the air surrounding the solar panel array at the time of measurement, measured in degree Celsius (°C).
- MODULE_TEMPERATURE: The temperature of the solar panel module itself, also measured in degree Celsius (°C). This value is critical for evaluating the thermal effects on panel efficiency.
- IRRADIATION: The solar irradiance value, representing the amount of solar power received per unit area at the plant, measured in watts per square meter (W/m^2).

B. *Data Pre-Processing*

1) *Forecasting*

a) *Data Cleaning:*

This process began by importing CSV files corresponding to the generation and weather data from two solar power plants. Upon loading the datasets, the column names were standardized to eliminate formatting issues such as leading or trailing spaces and inconsistent casing. Ensuring uniform naming conventions was critical for the subsequent merging and manipulation of datasets. Additionally, the DATE_TIME column was converted into a recognized datetime object using pandas datetime functions. This conversion enabled the sorting of records in chronological order and facilitated time-based operations such as interpolation and resampling. The datasets were then sorted based on the DATE_TIME field to preserve temporal integrity—an essential requirement for deep learning models that rely on time-sequenced inputs. To provide a coherent basis for downstream analysis, generation data and corresponding weather data were merged using an inner join on the DATE_TIME attribute. This operation ensured that only the records with both power readings and environmental conditions were retained. Each plant's data was merged separately to maintain clarity and preserve source-specific patterns, resulting in df_plt1_merged and df_plt2_merged. The PLANT_NAME column was also inspected, and where null values were found, a static assignment such as "Plant_2" was made based on the source file to ensure completeness. This cleaning step eliminated structural inconsistencies in column values and established a well-defined dataset for further preprocessing.

b) *Dealing with Missing Values:*

Sensor data collected from industrial installations often contain missing values due to transmission failures, hardware malfunctions, or maintenance downtime. In this study, a structured approach was adopted to handle missing values based on the nature of each feature. For time-dependent continuous variables such as DC_POWER and AC_POWER, missing entries were filled using time-based interpolation. The interpolation was conducted within each SOURCE_KEY group, representing individual inverters. This group-wise imputation ensured that interpolated values respected the operational profile of each inverter and did not introduce artificial correlations between unrelated units. The use of temporal interpolation preserved the continuity of the power curves, which is important for forecasting models that rely on trend detection. Cumulative metrics such as TOTAL_YIELD was treated differently. Since yield represents an accumulating total, a forward fill strategy was used to propagate the last valid observation forward within the same inverter group. This method, maintained consistency in yield progression and avoided artificially resetting the metric due to temporary data gaps. In the case of weather data, key environmental variables such as AMBIENT_TEMPERATURE, MODULE_TEMPERATURE, and IRRADIATION showed considerable missingness. Rows with missing entries in these columns were dropped after evaluating the proportion of missing data. This choice was justified by the critical importance of these variables in performance forecasting, and the removal of a small fraction of data was deemed acceptable to preserve model reliability.

c) *Exploratory Data Analysis*

Correlation Analysis: The merged dataset was analyzed to assess the linear correlation between generation and weather variables. It is not surprising that DC power and AC power are highly correlated, which is a good sign as it likely means the inverters in the plant are working properly to convert DC to AC. In our analysis we will use AC_POWER as the target variable. There is also a strong relationship between the hour of the day and the daily yield, which makes sense because the daily yield naturally increases as the day progresses. For the weather sensor data, we see a strong correlation between irradiation and AC power, as well as between module temperature and AC power.

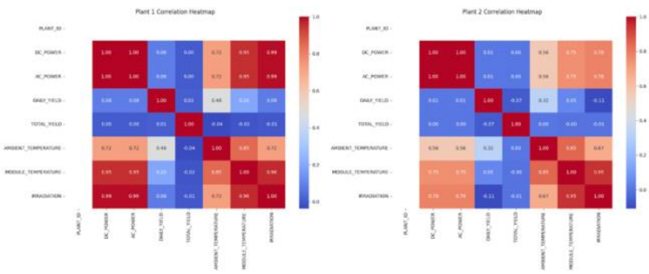

Fig. 1. Corelation heatmap of generational and weather variables

Distribution Analysis: Histogram plots were created to examine the distribution of key numerical features including DC_POWER, AC_POWER, and temperature readings. Most variables displayed right-skewed distributions, particularly power metrics, indicating periods of zero or minimal generation during early morning or night hours. Understanding these distributions is important when selecting normalization techniques and designing model input pipelines.

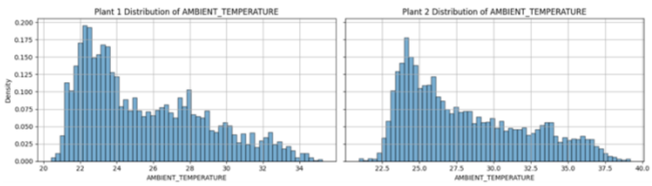

Fig. 2. Distribution Analysis of Weather Parameters (AMBIENT_TEMPERATURE)

Weather vs Generation Analysis: Scatter plots and time-series overlays were used to study the interaction between weather conditions and power generation. It was observed, as seen in Fig.3. that spikes in solar irradiation closely aligned with increases in AC_POWER, further supporting the causal influence of environmental parameters on plant performance. These patterns highlighted the significance of including weather features in the forecasting models.

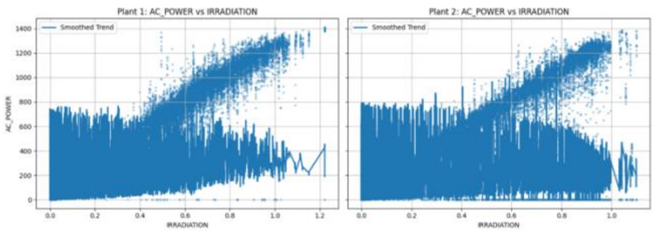

Fig. 3. Smoothened Scatter Plot of AC_POWER vs IRRADIATION

As seen in Fig.4. AC output rises modestly through 20–30 °C, then declines slowly above 32 °C due to reduced module efficiency.

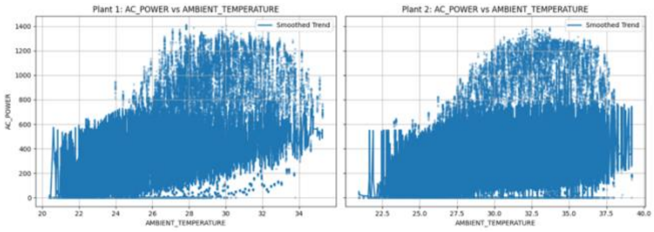

Fig. 4. Smoothened Scatter Plot of AC_POWER vs AMBIENT TEMPERATURE

d) Feature Engineering:

Feature engineering was performed to transform raw inverter-level and weather data into structured, informative inputs for solar power forecasting. Data from Plant 1 was aggregated at 15-minute intervals across all inverters, producing plant-level indicators such as total AC_POWER and DC_POWER, and statistical summaries of IRRADIATION, MODULE_TEMPERATURE, and AMBIENT_TEMPERATURE (mean, standard deviation, minimum, and maximum where applicable). The resulting dataset was reindexed to maintain consistent 15-minute intervals, and missing values were handled using time-based interpolation along with forward and backward filling to ensure continuity. To incorporate temporal patterns, the fractional hour of each timestamp was encoded using sine and cosine transformations, creating cyclic features hour_sin and hour_cos that help the model learn daily trends. [3] Based on domain knowledge and literature guidance, especially Kelachukwu's study on solar forecasting, the final selected features included AC_POWER, DC_POWER, solar irradiation and temperature statistics, and the cyclical time features. Notably, AC_POWER was also used as an input due to its autoregressive nature, as recent power values carry predictive information about future outputs. This comprehensive engineering step provided temporally-aware, scale-consistent inputs optimized for deep learning-based forecasting.

2) Anomaly Detection

a) Data Cleaning:

Consolidation of Generation Files: All generation CSV files were first uploaded and parsed. Each file corresponds to a different inverter, and the relevant fields, DATE_TIME, SOURCE_KEY, and AC_POWER, were extracted. The SOURCE_KEY field, which identifies the inverter, was renamed to a more consistent label, inverter. These datasets were then merged into a single dataframe for further processing.

Time Slot Indexing: To capture the daily pattern of each inverter, each timestamp was converted into two indices, namely, Date, which is the calendar date of the record and Slot, which is a 15-minute time interval index, ranging from 0 to 95, representing the 96 slots in a day (24 hours x 4 intervals per hour). Using these indices, a complete MultiIndex was created for all combinations of dates, inverters and time slots.

Matrix Reshaping: The merged data was reshaped into a 2D matrix X with dimensions (m, 96), where m is the number of valid (date, inverter) pairs. Each row represents one inverter on one day, and each column corresponds to a 15-minute time slot. Missing readings naturally appeared as NaN values in this matrix due to gaps in the original data.

b) Dealing with Missing Values (Data Imputation):

Once the data was consolidated, it was reshaped into a structured matrix format where each row represented the AC power profile of one inverter on one day, and each column represented a 15-minute interval slot within that day. This resulted in 96 columns per row (24 hours × 4 intervals per hour). Due to gaps in the data collection process, missing values (NaNs) were present in the resulting matrix. Each inverter-day (i.e., a row) was assessed based on the number and pattern of missing values. The imputation strategy followed a balanced approach wherein if an inverter-day contained more than 10 consecutive missing time slots, the entire row was dropped. These profiles were considered too incomplete to yield reliable insights, and, if the missing values were fewer and scattered (i.e., less than or equal to 10 consecutive missing slots), they were filled using linear interpolation, which estimates the missing values based on trends from surrounding data points. For any remaining leading or trailing gaps, forward-fill and backward-fill techniques were applied to ensure completeness. This two-tiered strategy allowed the retention of a majority of the data while maintaining its integrity and suitability for model training.

c) Exploratory Data Analysis (EDA):

Prior to transformation, the merged data underwent a brief exploratory analysis to understand its overall structure and data quality. Basic statistics such as the number of unique inverters, total rows, and the percentage of missing values in each column were computed. The daily inverter profiles were visualized after imputation to inspect the typical power output patterns across different time slots.

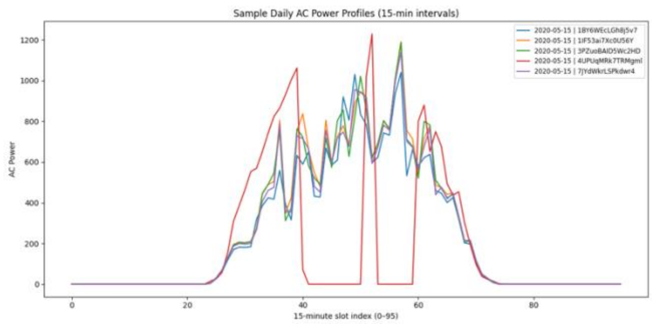

Fig. 5. Inverter profiles of AC Power Output across different time slots.

These plots as seen in Fig.5. confirmed that the inverter profile shown in red clearly indicates the presence of an anomaly in an inverter compared to other inverters. The same inverter is detected by the model as well. Most inverter profiles followed a bell-shaped generation curve, peaking during midday. This visual and statistical inspection helped validate the effectiveness of the cleaning and imputation process and confirmed that the data was ready for transformation and modeling.

d) Data Transformation

To prepare the data for input into deep learning models, the cleaned and imputed matrix was normalized. Each row, representing one inverter's daily power profile, was independently scaled using Min-Max normalization. This method scaled the values in each profile to a [0, 1] range, preserving the shape and variability of the profile while ensuring consistency across all samples. Normalization is essential for improving model performance, especially when using activation functions such as sigmoid, which operate optimally within a fixed range. This transformation enabled the learning algorithms to focus on the profile patterns rather than the absolute power values. This improved generalization and convergence during training. The final output of the preprocessing pipeline was a complete and normalized dataset that captured the daily AC power generation behavior of each inverter, ready to be used for forecasting and anomaly detection.

C. Model Prediction

1) Compartive Study for Forecasting Analysis

a) Feed Forward Neural Network (FFNN)

The FFNNs are not inherently designed to handle sequences, so, the function *_flattenWindowForFFNN* transforms the dataset by sliding a fixed-size window of 96 timesteps (equivalent to 24 hours with 15-minute intervals) across the feature matrix. For each iteration, the previous 96-time steps of input features are flattened into a single vector to form the model input, while the corresponding target value at the current timestep becomes the output. This process enables the FFNN to utilize historical temporal context despite lacking sequential memory capabilities. Following the transformation, the dataset is split into training and testing sets (80:20), and both the inputs and targets are scaled using MinMax normalization to accelerate convergence during training. A sequential FFNN model is defined with an input layer matching the flattened feature vector size, followed by multiple dense layers with ReLU activations and dropout layers to prevent overfitting. The model is compiled using the Adam optimizer with mean squared error as the loss function and mean absolute error as an evaluation metric. Training is carried out for up to 50 epochs with early stopping to prevent overfitting, using 10% of the training data as a validation set.

b) Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) was implemented to capture local temporal dependencies in the solar power time-series data. Unlike Feedforward Neural Networks, CNNs can recognize spatial patterns over time when input is structured in three dimensions. To enable this, the function _windowForCNNLSTM was used to slide a 96-timestep window over the feature set, preserving the shape (samples, time_steps, features) instead of flattening it. This allowed the model to process each sequence of 24 hours (with 15-minute intervals) as a time series with multiple features at each step. The input and target sets were then split into training and testing sets with an 80:20 ratio, and MinMax normalization was applied to both features and targets. Importantly, reshaping was done after scaling to maintain the appropriate 3D structure required by CNN layers. The CNN model architecture consisted of two 1D convolutional layers with ReLU activations to detect feature patterns over time, followed by a max pooling layer to reduce dimensionality and focus on dominant features. The resulting representation was flattened and passed through dense layers, with the final output layer producing a single prediction value. Based on prior observation of the AC power patterns, such as sharp daytime spikes and near-zero values during the night, the model employed the Huber loss function with a delta of 1500.0. This choice provided robustness to outliers and helped the model better capture both smooth and volatile power trends. The model was compiled with the Adam optimizer and trained using early stopping and a validation split, aiming to improve generalization while preventing overfitting. This CNN-based approach leveraged temporal convolution to better understand the shape and transition of solar power generation curves.

c) Long Short Term Memory (LSTM)

Long Short-Term Memory (LSTM) network was employed to model temporal dependencies in solar power generation. The same windowing function used for the CNN model was applied here, structuring the input data as a three-dimensional array with the shape (samples, time_steps, features). This format allows LSTM layers to learn from sequential patterns over a 24-hour period (96-time steps). The input and target data were split into training and testing sets, followed by MinMax scaling to normalize the features and targets. The scaled inputs were reshaped back to their original 3D structure, maintaining compatibility with LSTM architecture. The LSTM model architecture consisted of two stacked LSTM layers with 64 and 32 units, respectively. The first LSTM layer returned sequences to feed into the second, enabling deeper temporal learning. This was followed by two dense layers, the last of which produced a single output value representing the predicted AC power. [2] Drawing inspiration from a paper which concluded that the Nadam optimizer provided superior performance on time-series tasks, the team chose to implement Nadam with a learning rate of 0.005. The model was compiled with mean squared error as the loss function and mean absolute error as a performance metric, then trained with early stopping to prevent overfitting. This LSTM-based architecture was designed to effectively capture the long-term temporal trends and non-linearities present in solar power generation.

d) Hybrid Model:

A hybrid CNN-LSTM model was constructed to leverage the strengths of both convolutional and recurrent neural networks for solar power forecasting. The input shape remained three-dimensional, structured as (samples, time_steps, features), which enabled the model to learn both spatial patterns across features and temporal dependencies across time. The architecture began with two Conv1D layers, which acted as feature extractors, identifying short-term patterns and trends from the input sequences. A dropout layer was introduced after the convolutional layers to reduce overfitting by randomly deactivating neurons during training. Following the convolutional component, the model included two stacked LSTM layers, which learned the longer-term temporal dependencies inherent in solar power generation. These were followed by a dense layer with ReLU activation and an output layer predicting a single AC power value. Consistent with earlier findings and insights from prior architectures, the model used the Nadam optimizer and Huber loss function to improve robustness and convergence, especially given the high variance and spiky behavior in the power data. The model was trained using a 10% validation split and early stopping to avoid overfitting.

2) Auto-encoder based Anomaly Detection model

The core of the anomaly detection component is a deep autoencoder architecture designed to reconstruct daily AC power generation profiles. Each input sample to the model corresponds to a 96-dimensional vector representing the normalized AC power output of a specific inverter across 96 time slots (15-minute intervals) in a day. The autoencoder consists of two main parts, one being the encoder, which compresses the 96-dimensional input into a lowerdimensional latent representation. It consists of a series of fully connected (dense) layers with decreasing units: 512, 256, 128, and 64. Each layer uses the ReLU activation function, followed by a dropout layer with a 20 percent dropout rate to prevent overfitting, and the second one is a decoder that reconstructs the original input from the latent representation. The decoder mirrors the encoder in structure, with dense layers of increasing size: 64, 128, 256, and 512. The final output layer uses a sigmoid activation function to produce values in the same range as the normalized input, ensuring consistent reconstruction. The model is compiled using the Mean Squared Error (MSE) loss function, which measures the average squared difference between the input and its reconstruction. The optimizer used is Adam, with a learning rate of 0.001. The objective of training is to minimize reconstruction error for normal inverter behavior, allowing the model to later detect deviations as anomalies.

Training Configuration and Prediction: The model was trained on the entire normalized dataset using a validation split of 20 percent. Training was performed over 50 epochs with a batch size of 32. During training, both training and validation loss were monitored to assess model convergence and generalization. The model showed stable learning dynamics with a consistent decrease in loss, indicating effective learning of normal inverter behavior.

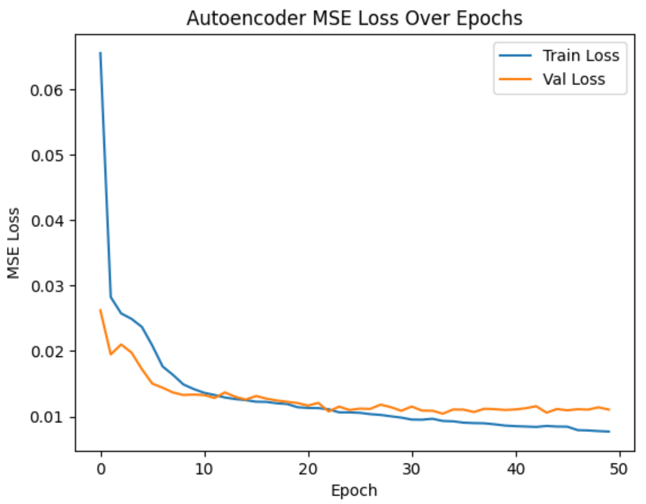

Fig. 6. Trends observed for Mean Squared Error loss across various epochs

After training, the model was used to reconstruct the input profiles. The reconstruction error for each sample was calculated using Mean Squared Error.

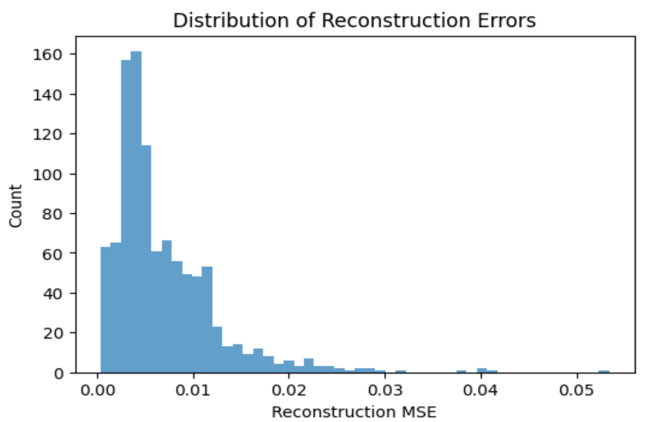

Fig. 7. Distribution of Reconstruction Errors.

These errors were then analyzed to determine a threshold for anomaly detection. The threshold was defined as the mean plus three standard deviations ($\mu + 3\sigma$) of the reconstruction error distribution. The reconstruction error analysis revealed that most inverter-day profiles were accurately reconstructed, with errors concentrated between 0.005 and 0.020. A threshold of approximately 0.024, set at the mean plus three standard deviations, identified only the most significant deviations. Out of 1,012 samples, 17 (around 1.7 percent) exceeded this threshold and were flagged as anomalies. These profiles are considered strong candidates for fault diagnosis or further investigation.

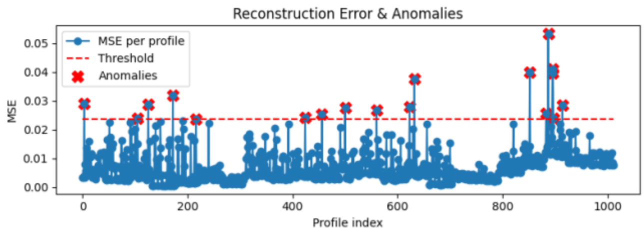

Fig. 8. Trends observed in reconstruction error and anomalies detected.

The result obtained was a threshold of 0.02350. This statistical approach ensures that only the most significant deviations are flagged. Profiles with reconstruction errors exceeding this threshold were classified as anomalies. The model was able to detect 18 anomalies out of 1012 samples. These may correspond to underperforming inverters, sensor faults, or irregular operational patterns. The integration of this model allows for automated, data-driven identification of unusual inverter performance without the need for manual rule-setting, enabling scalable and real-time monitoring in large solar installations.

a) Localization of Anomalies using Dynamic Time Warping (DTW)

After identifying anomalous inverter-day profiles using the autoencoder reconstruction error, the next objective was to localize where within each profile the abnormal behavior occurred. [5] To achieve this, the project utilized Dynamic Time Warping (DTW), a time-series alignment technique that measures similarity between two sequences that may vary in time or speed. Each inverter-day profile contains 96 time slots (15-minute intervals). These were divided into 24 non-overlapping segments of 4 time slots each, corresponding to 1-hour intervals. For every anomalous profile flagged by the autoencoder, the original and reconstructed sequences were compared segment-wise using DTW. DTW calculates the distance between corresponding segments of the original and reconstructed profiles. A local threshold was then determined for each profile based on the 95th percentile of its segment-wise DTW errors. Segments exceeding this local threshold were flagged as locally anomalous. This process enables precise identification of time intervals within the day where the reconstruction and behavior diverge significantly from the expected pattern. For instance, in one flagged profile, the DTW-based bar chart clearly highlighted certain one-hour blocks where the anomaly was most prominent, facilitating targeted fault analysis.

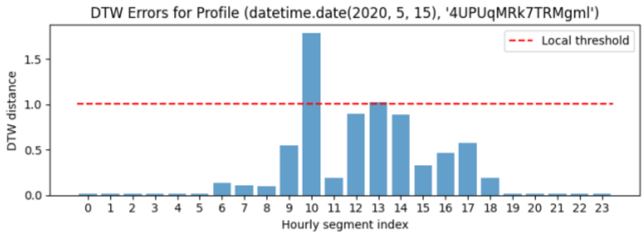

Fig. 9. DTW distance across various hours in a day along with local threshold.

The first visualization displays the DTW distances for each one-hour segment in an anomalous inverter-day profile. Each bar in the chart corresponds to the DTW error for a specific segment. A red horizontal line represents the local threshold (95th percentile), and segments that exceed this value are considered anomalous. This visual format helps to clearly identify which specific time windows within a day contributed to the anomaly.

IV. Results

A. Forecasting

1) Feed Forward Neural Network

To evaluate the performance of the Feedforward Neural Network (FFNN) model, standard regression metrics including Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were computed on the test set. Predictions from the trained model were first obtained on the scaled test input and then inverse-transformed to their original scale using the fitted target scaler. The resulting RMSE was approximately 9154.23 kW, and the MAE was 1370.95 kW, indicating that while the model performs reasonably well in general, there are significant errors, especially during periods of rapid change in solar output. A time-series plot comparing actual and forecasted AC_POWER values was generated for visual inspection. The model was able to learn and replicate the overall diurnal patterns in solar power production, such as the typical rise and fall across each day. However, it struggled to accurately predict sharp peaks and sudden drops, which contributed to the higher RMSE. This behavior reflects the FFNN's limitations in modeling volatile transitions in solar power output, despite its ability to capture smooth trends. These results emphasize the importance of exploring advanced temporal models or hybrid architectures for improved forecasting accuracy.

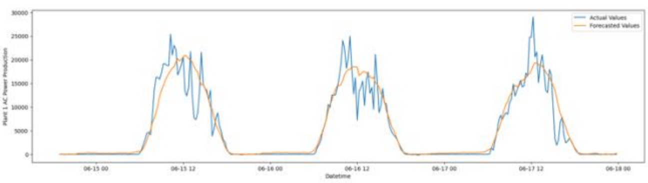

Fig. 10. FFNN Forecast – Actual and Forecasted value trends.

2) Convolutional Neural Network

To evaluate the performance of the Convolutional Neural Network (CNN) model, predictions were made on the test dataset, followed by inverse transformation of the scaled output to return to the original power scale. The model achieved a Root Mean Squared Error (RMSE) of 2905.11 kW and a Mean Absolute Error (MAE) of 1446.18 kW. These results suggest that while the CNN model produced slightly higher absolute errors on average compared to the FFNN, it was significantly more accurate in reducing large deviations, as reflected in the lower RMSE. This improvement is likely due to CNN's ability to detect and model short-term local dependencies in the input sequences, which are important in capturing subtle fluctuations in solar power generation. The visual comparison of actual versus forecasted values further demonstrates the model's effectiveness in reproducing the overall shape and cycle of daily solar power output. The CNN model effectively follows the rise and fall of generation across days and is more stable than FFNN in modeling daily patterns. However, it tends to underestimate sharp peaks and smooth ramp-ups, leading to residual errors during high-output periods. This limitation contributes to a slightly higher MAE, showing that while CNN is excellent at learning localized temporal features, it may benefit from enhancements such as hybrid architectures or attention mechanisms to better model longer-term trends and extreme fluctuations.

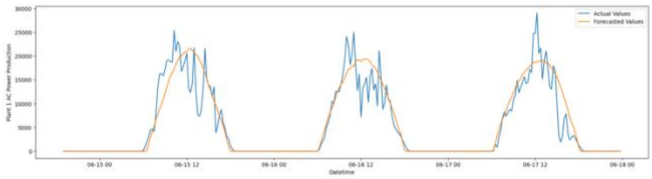

Fig. 11. CNN Forecast – Actual vs Forecasted value trends.

3) Long Short Term Memory

The Long Short-Term Memory (LSTM) model produced strong forecasting performance, yielding a Root Mean Squared Error (RMSE) of 2206.16 kW and a Mean Absolute Error (MAE) of 1182.46 kW on the test dataset. These results indicate that the LSTM outperformed both the FFNN and CNN models in terms of both overall error magnitude and the model's ability to follow true power output values. The lower RMSE, in particular, suggests improved handling of large deviations and peak values, something the previous models struggled with. This is largely due to the LSTM's ability to retain and learn long-term temporal patterns in sequential data, making it well-suited for time-series forecasting. The plotted predictions further demonstrate that the LSTM model tracks the daily cyclical nature of solar power output more accurately, including during ramp-up and ramp-down phases. It shows fewer underestimations at high-output intervals and maintains better alignment with the true values over time. The use of stacked LSTM layers allowed the model to build a hierarchical understanding of patterns across different temporal scales. Overall, the LSTM model demonstrated superior temporal awareness and adaptability, making it a highly effective architecture for forecasting solar power generation in this application.

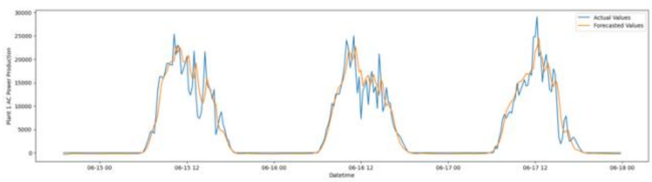

Fig. 12. LSTM Forecast – Actual vs Forecasted value trends

4) Hybrid Model

The hybrid CNN-LSTM model was evaluated on the test dataset, producing a Root Mean Squared Error (RMSE) of

2905.11 kW and a Mean Absolute Error (MAE) of 1446.18 kW. Predictions were inverse-transformed and clipped to eliminate negative values, ensuring physical validity since power generation cannot be negative. The hybrid model integrates the strengths of both CNN and LSTM architectures, CNN layers capture local temporal patterns and fluctuations in feature sequences, while LSTM layers model longer-term dependencies. However, the observed RMSE indicates that although the model produces stable and smooth forecasts, it tends to underperform during extreme variations or sharp spikes in power generation. The plotted results show that the CNN-LSTM model effectively captures general daily cycles and smooth transitions, maintaining reasonable accuracy throughout the day. The MAE suggests that the model performs well during typical power production periods, especially during daytime hours. However, the elevated RMSE highlights the model's difficulty in handling rare but significant deviations, likely caused by its smoothing behavior over sudden peak values. This trade-off between stability and sensitivity to extremes suggests that while the hybrid model provides robust baseline forecasting, additional refinement, such as attention mechanisms or ensemble methods may be required to better handle high-variance conditions in solar power output.

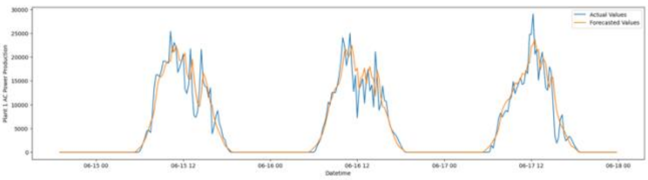

Fig. 13. Hybrid Forecast – Actual vs Forecasted value trends

B. *Anomaly Detection*

The final visualization overlays the original daily power profile and its autoencoder reconstruction. This plot allows for a direct comparison of expected versus actual behavior. Additionally, anomalous segments identified using DTW are highlighted as shaded red regions across the time axis (in 15-minute slots). For the inverter 4UPUqMRk7TRMgml on 2020-05-15, the DTW analysis flagged segments 10 and 13 (10:00–11:00 and 13:00–14:00) as anomalous due to elevated reconstruction errors. In the overlaid time-series plot, these intervals show abrupt drops and spikes in the original profile, while the reconstructed profile remains smooth. This clearly highlights deviations from normal behavior, confirming localized anomalies. This combined view not only demonstrates where the reconstruction fails but also aligns those failures with the exact periods of the day. This is particularly useful for maintenance teams or plant operators to correlate anomalies with real-world events, such as shading, cloud cover, equipment faults, or temporary disconnects.

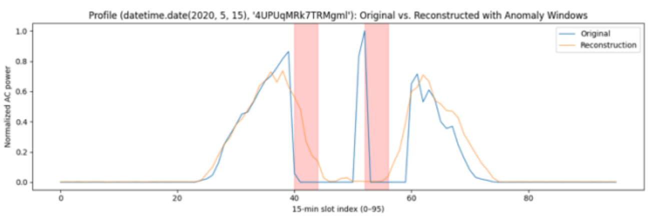

Fig. 14. Original vs Reconstructed AC Power including Anomaly Windows.

V. CONCLUSION

This project presents a comprehensive deep learning framework aimed at forecasting solar power generation and detecting anomalies in photovoltaic plant operations. Through rigorous data preprocessing, time-based feature engineering, and sequence modeling, we have developed and compared the performance of multiple neural architectures, including Feedforward Neural Networks (FFNN), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, and a hybrid CNN-LSTM model. Each of these architectures was evaluated in terms of its ability to capture temporal dependencies and accurately predict AC power output based on historical operational and environmental parameters. The results demonstrate that while FFNN models provide a baseline for forecasting, sequence-based models such as LSTM and CNN-LSTM offer superior performance in capturing the periodic and spiky nature of solar power data. The CNN model effectively learns short-term local patterns, whereas LSTM excels in modeling long-range dependencies. The hybrid CNN-LSTM model combines the strengths of both and delivers stable and smooth predictions, albeit with a tradeoff in underpredicting rare peaks. Moreover, the integration of techniques such as cyclic time encoding and MinMax scaling ensures that the models are well-optimized for training on real-world time-series data. Overall, for this dataset while forecasting, LSTM model tuned with Nadam seems to be the best model as it gives a stable RMSE and MAE value, which means the model handles temporal patterns and peaks well, and also, it handles spiky AC power quite with ease. After that, the CNN model with Huber loss and CNN-LSTM model seems to be on par with each other. They both have almost the same RMSE and MAE value as well, which means the forecasts are stable in nature but still smooth out sharp noon spikes, inflating the RMSE values. The worst performing network is FFNN where it is able to capture the trend but fails at peak, and also has a very high RMSE value due to these spikes. Also, this project not only showcases the effectiveness of deep learning in renewable energy forecasting but also lays the groundwork for future enhancements, including the implementation of anomaly detection modules using autoencoders and DTW for localized diagnostics in inverter-level data.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to our Course Instructor (SEP 769 - Cyber Physical System), Dr. Anwar Mirza and Teaching Assistant, Mr. Amir for their continuous guidance, support, and valuable feedback throughout the duration of this project. Their insights have been instrumental in shaping the direction and depth of our research. We also extend our appreciation to the authors of the referenced research papers, whose contributions to the field of solar power forecasting and deep learning served as a foundation for our work. Finally, we acknowledge the collaborative efforts of our team members, whose dedication and collective enthusiasm made this project both intellectually enriching and technically rewarding.

REFERENCES

[1] S. Miraftabzadeh, M. Jaradat, and A. Mohamed, "Data anomaly detection in photovoltaic power time-series via unsupervised deep learning with insufficient information," *Data-Centric Engineering*, vol. 5, 2025. [Online]. Available:

https://www.sciencedirect.com/science/article/pii/S2352467725001511

[2] R. Sharma, B. Singh, and M. P. Sharma, “A novel long term solar photovoltaic power forecasting approach using LSTM with Nadam optimizer: A case study of India,” *Energy Science & Engineering*, vol. 11, no. 5, pp. 1535–1547, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/ese3.1178

[3] C. Iheanetu, “Solar Photovoltaic Power Forecasting: A Review,” *Sustainability*, vol. 14, no. 24, p. 17005, 2022. [Online]. Available: https://www.mdpi.com/2071-1050/14/24/17005

[4] S. Lim, S. Park, and H. J. Kim, “Solar Power Forecasting Using CNN-LSTM Hybrid Model,” *Energies*, vol. 15, no. 21, p. 8233, 2022. [Online]. Available: https://www.mdpi.com/1996-1073/15/21/8233

[5] T. Giorgino, “Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package,” *Journal of Statistical Software*, vol. 31, no. 7, pp. 1–24, 2009. [Online]. Available: https://www.jstatsoft.org/article/view/v031i07

[6] Y. Khouili, A. Moughit, and A. Mousannif, “Evaluating the impact of deep learning approaches on solar and photovoltaic power forecasting: A systematic review,” *Renewable Energy Focus*, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211467X25000987

[7] M. Barhmi, M. Essaadi, and H. Ghennioui, “A Review of Solar Forecasting Techniques and the Role of Artificial Intelligence,” *Smart Cities*, vol. 4, no. 1, pp. 74–94, 2024. [Online]. Available: https://www.mdpi.com/2673-9941/4/1/5