Photovoltaic Power Forecasting using LSTM and Transformers

Anil Bediyasar (2110931)
Electronics & Communication Engineering
NITK, Surathkal
Karnataka, India

***Abstract*—The electricity demand in the world is always increasing. However, the traditional source of fossil fuel is limited and they leave significant carbon footprint. These factors, along with the technological advancements, have driven the increasing usage of distributed renewable energy resources. The rapid adoption of renewable energy sources has made accurate photovoltaic (PV) power forecasting crucial for ensuring grid stability and optimizing energy management systems. This project explores advanced machine learning techniques, including Long Short-Term Memory (LSTM) networks and Transformer models, to predict PV power generation effectively. By leveraging time-series weather data and historical power output, the models capture temporal patterns and dependencies, enabling precise short-term and long-term forecasts.**

Our approach combines the sequential learning strength of LSTMs with the attention mechanism of Transformers to enhance predictive accuracy, even in scenarios of high variability due to weather fluctuations. Experimental results demonstrate improved forecasting performance compared to traditional statistical methods and standalone deep learning models. This study highlights the potential of integrating cutting-edge AI techniques to advance renewable energy forecasting, offering a sustainable pathway for energy resource management and grid reliability.

***Index Terms*—Solar photovoltaic power plant, Forecasting, time-series, Stationarity, ARIMA, LSTM, Transformers**

I. Introduction

With the increasing global reliance on renewable energy sources, accurately forecasting photovoltaic (PV) power generation has become a critical requirement for efficient grid management and resource optimization. PV power generation is highly dependent on weather conditions, which introduces significant variability and complexity into the forecasting process. Developing robust prediction models to address these challenges is essential for enhancing the reliability of renewable energy systems.

This project initially explores traditional time-series forecasting techniques using the AutoRegressive Integrated Moving Average (ARIMA) model to predict PV power generation. ARIMA, a widely used statistical method, effectively handles linear trends and seasonality in data. However, its limitations in capturing non-linear patterns and long-term dependencies necessitated the exploration of advanced machine learning approaches.

Subsequently, the study transitions to implementing Long Short-Term Memory (LSTM) networks and Transformer models, leveraging their ability to model complex temporal patterns and relationships within time-series data. LSTM, a variant of recurrent neural networks (RNNs), excels in learning sequential dependencies, while Transformers utilize self-attention mechanisms to capture global contextual information, enabling enhanced predictive accuracy.

This report outlines the comparative analysis of these techniques, emphasizing the significant improvements achieved by integrating deep learning models over traditional methods. By combining state-of-the-art AI methods, this work contributes to advancing PV power forecasting, offering insights for more sustainable energy management and grid optimization.

II. Preparing Models

A. ARIMA Model

The AutoRegressive Integrated Moving Average (ARIMA) model is a popular statistical method for time-series forecasting. It is defined by three parameters: p (autoregressive order), d (degree of differencing), and q (moving average order). The ARIMA model combines these components to model and forecast stationary time-series data.

1) Components of ARIMA:

a) Autoregressive (AR) Component: The AR component expresses the time series as a linear function of its previous values:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \epsilon_t$$

where:

- X_t: Value at time t
- ϕ_i: AR coefficients
- p: Order of autoregression
- ϵ_t: White noise (error term)

b) Differencing (I - Integration): Differencing is applied to make a non-stationary time series stationary by removing trends. The differenced series is:

$$Y_t = X_t - X_{t-1}$$

For higher orders of differencing (d), this process is repeated d times.

c) Moving Average (MA) Component: The MA component models the dependency between a data point and past forecast errors:

$$X_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}$$

where:

- θ_i: MA coefficients
- q: Order of moving average

2) ARIMA Model Combination: The ARIMA model is a combination of the AR, I, and MA components:

$$\Phi_p(B) \nabla^d X_t = \Theta_q(B) \epsilon_t$$

where:

- $\Phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$: AR polynomial
- $\Theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q$: MA polynomial
- B: Backshift operator ($B^k X_t = X_{t-k}$)
- $\nabla^d X_t$: Differenced series

3) Steps in ARIMA Model Building:

1) **Stationarity Check:** Use techniques like the Augmented Dickey-Fuller test.
2) **Parameter Estimation:** Determine p, d, and q using autocorrelation function (ACF) and partial autocorrelation function (PACF).
3) **Model Fitting:** Fit the ARIMA equation to the time series.
4) **Validation:** Analyze residuals for randomness and ensure no patterns remain.

4) ARIMA Example: For an ARIMA(1,1,1) model:

$$(1 - \phi_1 B)(1 - B) X_t = (1 + \theta_1 B) \epsilon_t$$

Expanding this:

$$X_t - \phi_1 X_{t-1} - X_{t-1} + \phi_1 X_{t-2} = \epsilon_t + \theta_1 \epsilon_{t-1}$$

This equation combines differencing, autoregression, and moving average components to predict X_t.

B. LSTM

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) capable of learning long-term dependencies in sequential data. They are particularly effective for time-series prediction tasks like photovoltaic (PV) power forecasting because they can model the temporal patterns and relationships in the data.

Key Components of the LSTM

1) Sequential Data Handling:: LSTMs process data in a sequence, taking one time step at a time, while remembering information from previous time steps. This is crucial for PV forecasting, as the model can learn the impact of past weather conditions and power outputs on future predictions.

2) Cell State and Gates:: LSTMs process data in a sequence, taking one time step at a time, while remembering information from previous time steps. This is crucial for PV forecasting, as the model can learn the impact of past weather conditions and power outputs on future predictions.

- **Cell State**: Acts as a memory that carries information across time steps. It can add or remove information using gates.
- **Input Gate**: Decides what new information should be stored in the cell state.
- **Forget Gate**: Determines what information should be removed from the cell state.
- **Output Gate**: Controls how much information from the cell state should be passed to the next layer or as the output.

3) Mathematical Operations:: At each time step t, the LSTM performs the following computations:

Forget Gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

where:

- f_t is the forget gate output,
- h_{t-1} is the previous hidden state,
- x_t is the input at time t,
- W_f, b_f are learnable parameters.

Input Gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

where:

- i_t determines what information to add,
- $\tilde{C}_t$ is the candidate cell state.

*Cell State Update

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Output Gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

where:

- h_t is the hidden state passed to the next time step or used for predictions.

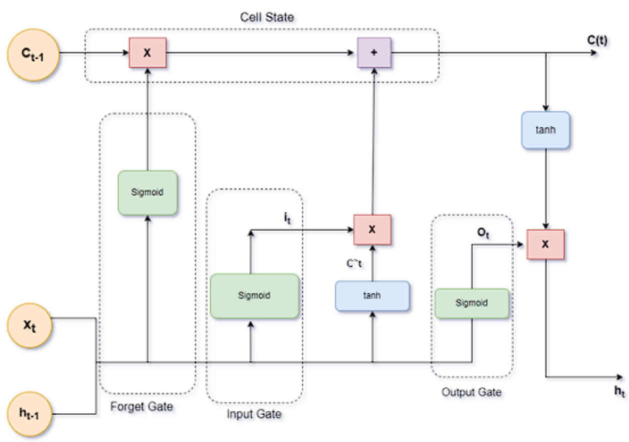

Fig. 1. LSTM architecture

C. Transformer Model

The Transformer model, originally designed for natural language processing tasks, has proven effective in time-series forecasting due to its ability to capture long-term dependencies and global patterns in data. Unlike recurrent models such as LSTM, Transformers rely on self-attention mechanisms, enabling them to process sequences in parallel and focus on the most relevant features for prediction.

1) Key Components of the Transformer:

a) Input Embedding: The input time-series data is first converted into a suitable format using embeddings. For numerical time-series data, this involves embedding the input features and possibly adding positional encodings to retain information about the order of the sequence. The positional encoding is defined as:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

where:

- i: dimension index within the embedding vector
- d: Represents the total dimensionality of the embedding vector

b) Self-Attention Mechanism: The self-attention mechanism allows the model to weigh the importance of different time steps in the sequence when making predictions. The attention mechanism is calculated as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

c) Encoder-Decoder Architecture: The encoder processes the input sequence and generates context-aware representations. It consists of multiple layers, each with a multi-head self-attention mechanism and a feed-forward network. The decoder takes the encoded sequence and predicts the next time step. It also uses a masked self-attention mechanism to ensure predictions are based only on past and present data.

d) Multi-Head Attention: Instead of relying on a single attention mechanism, the Transformer employs multiple attention heads to capture different aspects of the sequence:

$$MultiHead(Q, K, V) = Concat\left(head_1, head_2, ..., head_h\right)W^O$$

e) Feed-Forward Network: After the attention mechanism, a feed-forward network is applied to introduce non-linearity:

$$FFN(x) = ReLU\left(xW_1 + b_1\right)W_2 + b_2$$

f) Layer Normalization and Residual Connections: Layer normalization and residual connections are used to stabilize training and improve convergence.

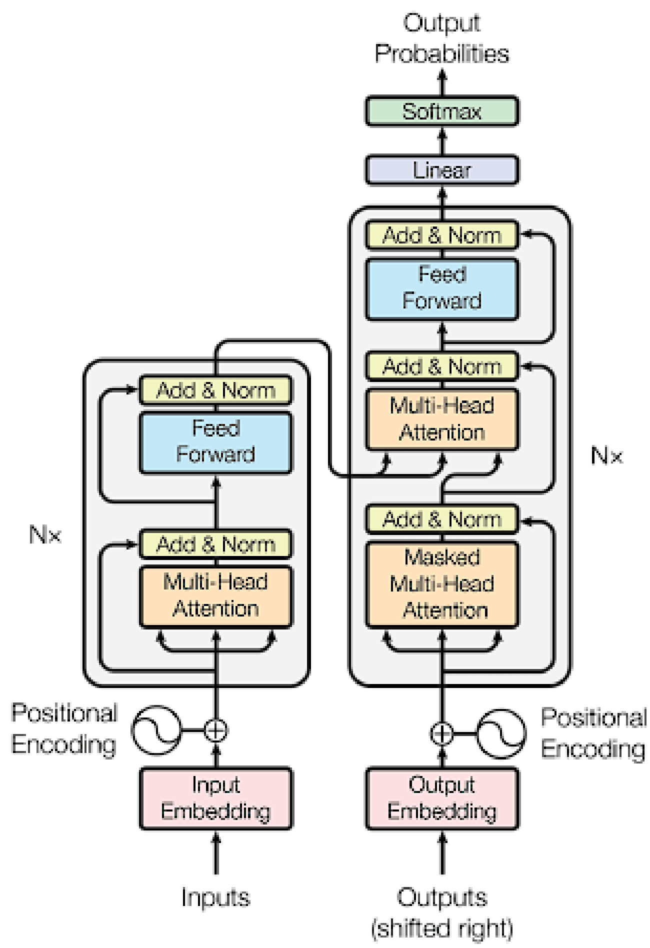

Fig. 2. Transformer architecture illustration.

III. Data Preparation

The Dataset used is a combination of two datasets, first one is the power generation data (MW) over a year 2006 from 1st January 2006 to 31st December 2006 at a particular location.

The power data is collected at a time interval of 30 minutes for entire day.The other data is weather data obtained from NREL website for same location as well as same year.

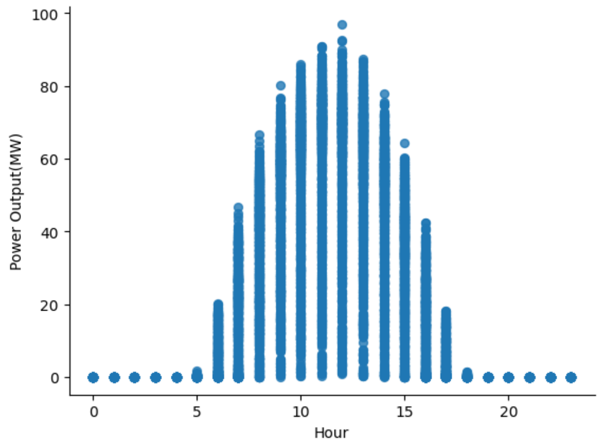

Fig. 3. HOUR vs Power Output Scatter Plot(MW)

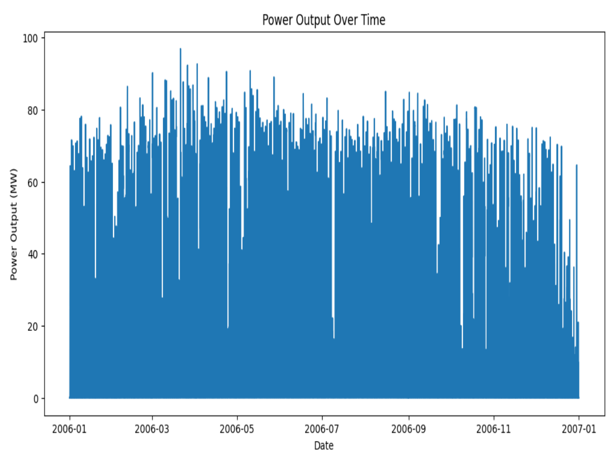

Fig. 4. Power Output over time

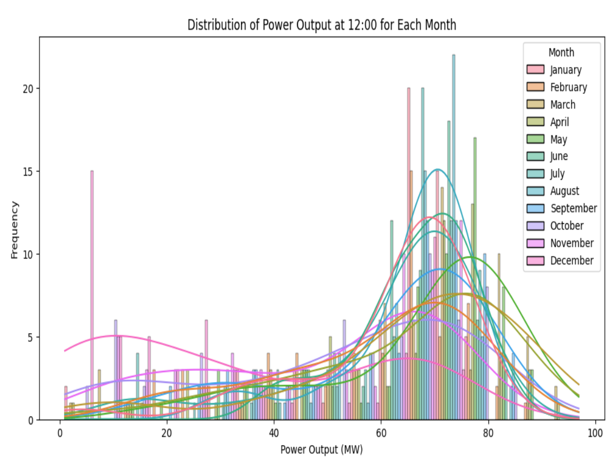

Fig. 5. Distribution of Power at 12:00pm for each day w.r.t month

IV. Analysis

A. ARIMA

The dataset was tested for stationarity using statistical methods, such as the Augmented Dickey-Fuller (ADF) test. A p-value was obtained from the test, and if the p-value was less than the significance level (e.g., 0.05), the null hypothesis of non-stationarity was rejected, indicating that the dataset is stationary.

The ARIMA model performs well in capturing the major peaks and valleys of the actual power output, indicating that it can model the overall structure and seasonality of the data. During periods of rapid increase and decrease (e.g., the sharp peaks), the model aligns closely with the actual data. However, there are slight deviations in some areas, particularly at the transition points. The model shows difficulty in handling the smaller fluctuations (noise) seen in the data.

In conclusion we can say that ARIMA model is able perform well in modeling the stationary trends and seasonality of the data and can capture the larger, significant changes in power output efficiently. But it fails to model the smaller, higher-frequency variations in the data.

TABLE I
Error Metrics for ARIMA Model

Error	Value
MAE	3.4549
RMSE	5.6088

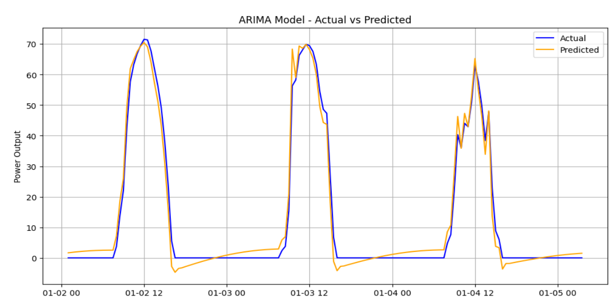

Fig. 6. ARIMA model predicted output vs Actual Data

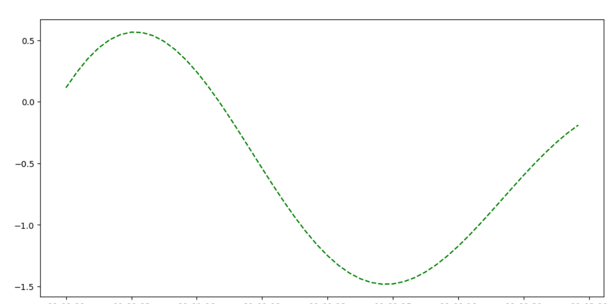

Fig. 7. ARIMA model predicted output for next 2 days

B. Long Short-Term Memory (LSTM)

LSTMs excel in learning patterns over time, handling both short-term and long-term dependencies effectively. LSTMs can identify both short-term fluctuations and long-term trends in your time-series data (e.g., power output patterns). Unlike traditional models like ARIMA, LSTMs can handle complex nonlinear relationships. LSTMs are robust to non-stationary time-series data without requiring explicit differencing or transformations.

In conclusion we can say that LSTMs often yield improved performance, especially when the data exhibits irregular, non-seasonal patterns or external variables influence the output. The ability to include additional features (e.g., weather conditions, time of day) enhances the predictive power of the model.

TABLE II
ERROR METRICS FOR UNIVARIATE MODEL

Error	Value
MAE	2.4269
RMSE	5.4922

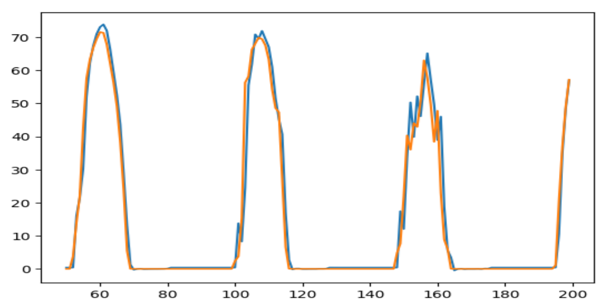

Fig. 8. Univariate LSTM model predicted output vs Actual Data

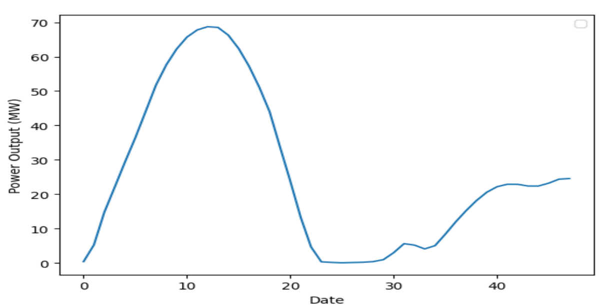

Fig. 9. Univariate LSTM model predicted output for next 1 day

TABLE III
ERROR METRICS FOR MULTIVARIATE MODEL

Error	Value
MAE	2.9670
RMSE	5.4223

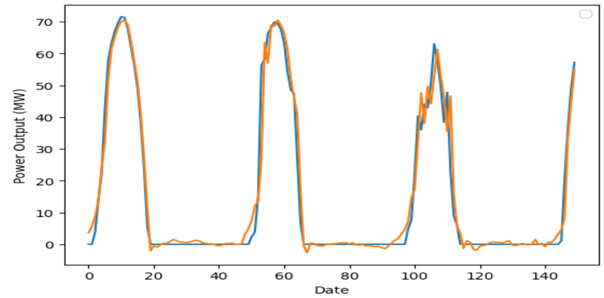

Fig. 10. Mutlivariate LSTM model predicted output vs Actual Data

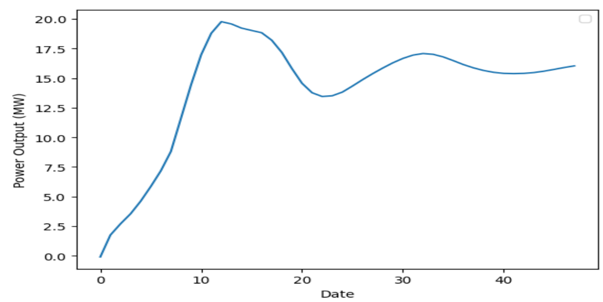

Fig. 11. Multivariate LSTM model predicted output for next 1 day

C. Transformer Model

The Transformer model closely follows the overall trends of the actual data. It accurately captures the peaks and valleys, indicating good performance in modeling periodic and long-term patterns. The predictions align well with the sharp peaks and transitions. Compared to traditional models like ARIMA, Transformers seem better at handling abrupt changes and retaining higher precision near the transition points. However, some peaks are slightly under-predicted or over-predicted, which suggests minor inaccuracies in scaling. The model effectively handles smaller fluctuations in the data and aligns well with the noise level, which could be due to its attention mechanism that captures long- and short-term dependencies. In conclusion we can say that transformer model appears to handle abrupt changes and noise more effectively, making it a superior choice for non-linear and complex time-series data. Its ability to capture long-term dependencies ensures accurate trend prediction, but minor calibration could further refine its performance. Further tuning of hyperparameters such as learning rate, number of layers, or attention heads could enhance accuracy.

TABLE IV
ERROR METRICS FOR TRANSFORMER MODEL FOR 2 ENCODER STACK

Error	Value
MAE	3.3418
RMSE	6.0678

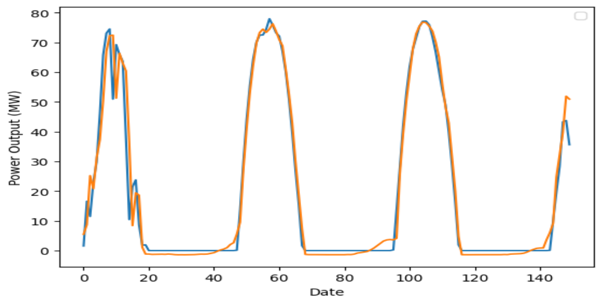

Fig. 12. Transformer with stack 2 predicted output vs Actual Data

TABLE V
ERROR METRICS FOR TRANSFORMER MODEL FOR 6 ENCODER STACK

Error	Value
MAE	2.8899
RMSE	5.7463

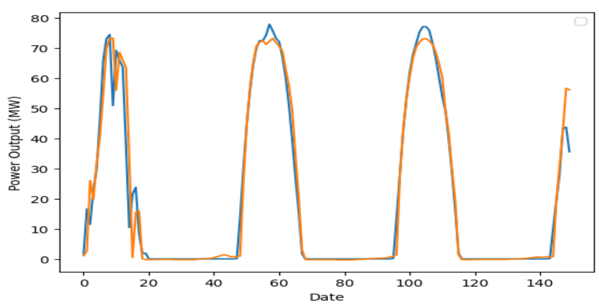

Fig. 13. Transformer with stack 6 predicted output vs Actual Data

V. CONCLUSION AND FUTURE SCOPE

Conclusion: In this project, we successfully implemented ARIMA, LSTM, and Transformer models to forecast photovoltaic (PV) power output. The ARIMA model proved effective for linear time-series data, while the deep learning models, particularly LSTM and Transformers, excelled in capturing complex non-linear patterns and temporal dependencies. Among these, the Transformer model exhibited superior performance due to its ability to process long-range dependencies and perform parallel computations efficiently. This study highlights the importance of accurate forecasting for ensuring effective grid management, load balancing, and optimal energy storage utilization, which are critical for integrating renewable energy sources into modern power systems.

Future Scope: The project offers significant opportunities for further development. Hyperparameter optimization of LSTM and Transformer models, using techniques such as grid search or Bayesian optimization, could enhance prediction accuracy. Additionally, the adoption of advanced architectures like Informers, specifically designed for time-series forecasting, holds great promise for improving efficiency and handling long-term dependencies. Integrating richer weather data, such as solar irradiance and wind speed, could improve the robustness of the models.

ACKNOWLEDGMENT

I would like to express our sincere gratitude to all those who have contributed to this research project. First and foremost, I would like to thank our supervisor, Prof. Debashisha Jena for their invaluable guidance, support, and encouragement throughout the duration of this work. Lastly, I would like to acknowledge the participants who generously contributed their time and expertise to this study.

REFERENCES

[1] Jimin Kim, Josue Obregon, Hoonseok Park and Jae-Yoon Jung, "Renewable and Sustainable Energy Reviews" Department of Industrial and Management Systems Engineering, Kyung Hee University, 1732 Deogyeong-daero, Giheung-gu, Yongin-si, Gyeonggi-do, 17104, Republic of Korea.

[2] Atique Sharif, Noureen Subrina, Roy Viswajit, Subhuraj Vinitha, Bayne Stephen, Macfie Joshua, "Forecasting of total daily solar energy generation using ARIMA: A case study," unpublished.

[3] V. P. Singh, V. Vijay, M. S. Bhatt, and D. K. Chaturvedi, "Generalized neural network methodology for short term solar power forecasting," in 2013 13th International Conference on Environment and Electrical Engineering (EEEIC), Nov 2013, pp. 58–62.