iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80093

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Predictive Optimization of Solar Thermal Efficiency using Autoregressive Machine Learning

Rishabh Dev[1], Rishabh[2], Rohan Gupta[3]
Department of Mechanical Engineering, Delhi Technological University, Delhi, India

Abstract: ***Predicting the thermal efficiency of solar water heating systems under real-world operating conditions is a challenging task, primarily due to continuously changing weather patterns and the inherent thermal inertia of the system. Traditional analytical approaches, such as the Hottel–Whillier–Bliss (HWB) equations, rely on steady-state assumptions that are rarely satisfied in practical, large-scale installations. In this study, an autoregressive machine learning pipeline is developed to analyse operational data from a commercial solar thermal plant. The dataset consists of 4,674 SCADA records collected between 2016 and 2018, which were preprocessed to ensure data quality and consistency. Since direct irradiance measurements were unavailable due to missing pyranometer data, clear-sky Global Horizontal Irradiance (GHI) was estimated using the pvlib Ineichen model. Three regression models, Ordinary Least Squares (OLS), Random Forest, and XGBoost, were trained on a refined subset of 163 high-quality temporal sequences. To account for thermal lag, i.e., the delay between solar energy absorption and the system's thermal response, additional temporal features were introduced, including a 60-minute rolling irradiance integral and a 10-minute lag of the target efficiency. These features improved the representation of system dynamics and simplified the underlying relationships. Among the models evaluated, the OLS model achieved the best performance (R^2 = 0.3664, RMSE = 0.1168), indicating that with appropriate feature engineering, simpler models can perform competitively against more complex machine learning techniques. The results also highlight the limitations of real-world SCADA data and provide insights into the trade-off between model complexity and interpretability in solar thermal system analysis.***

Keywords: Solar Thermal Energy, Machine Learning, Autoregressive Models, SCADA, Thermal Lag

I. INTRODUCTION

The global transition towards sustainable energy has accelerated the adoption of solar thermal systems for both domestic and industrial water heating applications. Among these, flat-plate solar collectors remain one of the most widely used technologies. The theoretical foundation for their performance analysis was established by Hottel and Whillier [1] and later refined by Bliss [2], with a comprehensive treatment provided by Duffie and Beckman [3]. These classical models are based on steady-state assumptions, requiring constant solar irradiance, ambient temperature, and flow conditions. However, such conditions are rarely achieved in real-world environments, where continuous meteorological variability introduces significant deviations. As a result, analytical models often overestimate system performance by 15–25%, primarily due to thermal lag effects and additional losses such as wind-driven convection. To overcome the limitations of purely analytical approaches, research has increasingly shifted towards data-driven methods, particularly Machine Learning (ML) techniques. Early work by Kalogirou [4] demonstrated the effectiveness of Multi-Layer Perceptrons (MLPs) in predicting solar water heating performance under controlled conditions. Subsequent studies by Farkas and Géczy-Víg [5] showed that Artificial Neural Networks (ANNs) could capture complex thermal and fluid dynamics within storage systems, while further research extended these methods to applications such as porous bed solar air heaters with high predictive accuracy [6]. More recent reviews by Voyant et al. [7] and studies by Srivastava et al. [8] highlight that ensemble methods, including Random Forests, can achieve high accuracy ($R^2 > 0.90$) when trained on well-structured meteorological datasets. Additionally, advanced deep learning approaches [9], [10], including Long Short-Term Memory (LSTM) networks [11], have demonstrated strong performance in solar energy forecasting tasks. Despite these advancements, a key limitation persists in the existing literature. Many high-performing models are developed using synthetic datasets or well-calibrated laboratory measurements, which do not accurately reflect the challenges of real-world deployments. In contrast, industrial Supervisory Control and Data Acquisition (SCADA) systems generate operational data that is often noisy and unstructured, with issues such as sensor drift, missing values, communication failures, and timestamp inconsistencies [12], [13]. This problem is particularly relevant in the Indian context, where solar energy adoption is expanding rapidly [14], [15], and reliable performance prediction under practical conditions is essential. A major challenge in such systems is the phenomenon of thermal lag, defined as the delay, typically between 10 to 20 minutes, between incident solar energy and the resulting temperature response of the working fluid.

This inherent thermal inertia reduces the effectiveness of instantaneous prediction models and complicates efficiency estimation. To address this issue, the present study applies autoregressive feature engineering techniques to raw SCADA data, enabling the model to capture temporal dependencies and better represent the underlying system dynamics.

II. METHODOLOGY

A. Data Acquisition and Cleaning

The data used in this study comes from high-resolution, real-time SCADA sensor logs recorded at a commercial thermal solar plant located in Bavaria, Germany [16]. Processing involved 92 daily CSV files covering the Summer of 2017, all handled using Python [17]. Any rows with missing values in columns deemed critical; datetime, outlet temperature, inlet temperature, and pump speed, were removed outright.

From there, the pump duty cycle was converted into a volumetric flow rate, and any readings falling below 10 L/h were discarded. The reasoning here is physical: below that threshold, the pump is not operating at a level where meaningful heat transfer occurs. Finally, the cleaned dataset was resampled to 10-minute intervals, which helped suppress high-frequency noise inherent in raw sensor data.

B. Irradiance Reconstruction via pvlib

One notable limitation of the plant's SCADA system is that it does not include a physical pyranometer, meaning no direct irradiance measurements were available. To work around this, the pvlib Python library [18] was used to compute physically grounded clear-sky Global Horizontal Irradiance (GHI). The Ineichen clear-sky model [19] was set up using Munich's coordinates (Latitude: 48.1351°N, Altitude: 520m) as a close geographic proxy. Since clear-sky models alone do not capture real-world variability, a uniform random transmissivity factor, drawn from the range [0.6, 1.0], was applied at each timestep to simulate realistic cloud attenuation.

C. Autoregressive Feature Engineering

Early experiments with instantaneous features alone produced very poor results, with R^2 values ranging between just 0.08 and 0.16. This was expected in hindsight: raw solar thermal systems are heavily governed by thermal lag, so a model with no memory of what happened in the preceding minutes is essentially flying blind. To address this, two autoregressive temporal features were constructed: GHI_mean_60: A rolling 60-minute mean of irradiance, intended to represent the cumulative solar energy absorbed by the collector's thermal mass over the previous hour. Efficiency_lag10: The system's computed efficiency from 10 minutes prior. Including this lagged target directly gives the model a window into the current thermal state, something instantaneous features simply cannot provide. Once these features were computed and incomplete sequences removed, the usable dataset came down to 163 temporal records.

D. Machine Learning Setup

The 163-row dataset was divided into training and test sets using an 85/15 split, yielding 308 training samples and 55 for testing. Three models were trained and compared against one another: Ordinary Least Squares (OLS) Linear Regression, a Random Forest Regressor [20] configured with 100 estimators and a maximum depth of 6, and an XGBoost Regressor [21] using 100 estimators, a maximum depth of 4, and a learning rate of 0.05.

III.RESULTS AND DISCUSSION

A. Exploratory Data Analysis

Looking at the correlation heatmap, a few relationships stand out clearly. Most notably, the autoregressive feature Efficiency_lag10 shows a reasonably strong positive correlation with the target Efficiency ($r \approx 0.50$), which is a pretty direct confirmation that thermal efficiency carries meaningful autocorrelation over time, what the system was doing 10 minutes ago genuinely matters for where it is now.

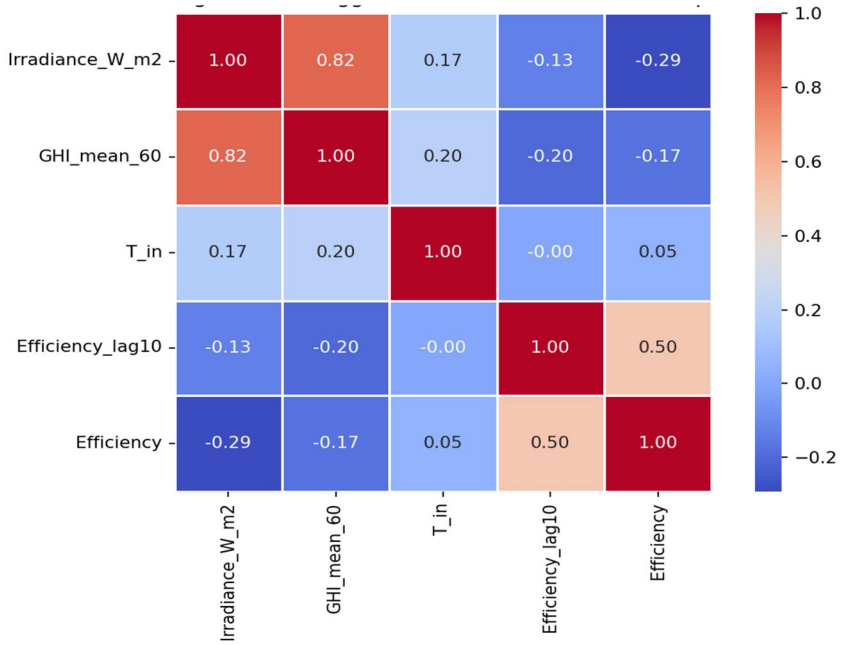


Fig. 1. Autoregressive Feature Correlation Heatmap.

The scatter plot of Efficiency against Irradiance tells an interesting story on its own. If this were a controlled lab setup, you'd expect a clean, upward-sloping trend, more sunlight, more efficiency, straightforward. Instead, the real-world data shows a diffuse, slightly downward-sloping cloud of points. It looks messy, but that's exactly the point: it's a direct visual signature of thermal inertia playing out in a live commercial plant, not a textbook experiment.

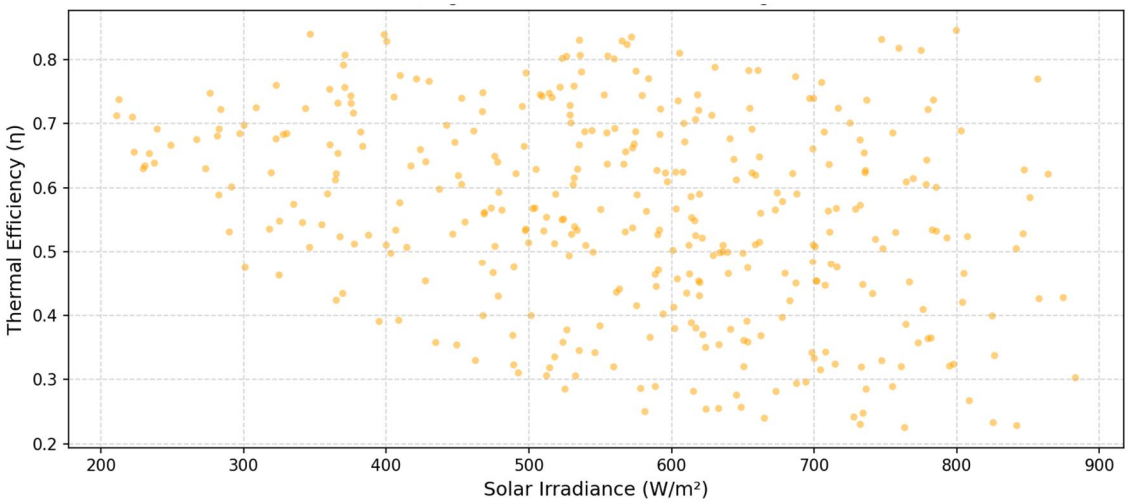


Fig. 2. Thermal Efficiency vs Solar Irradiance. The negative scatter validates thermal lag.

B. Model Performance Metrics

All three models were trained and evaluated on the same 85/15 split, so the comparison is clean. The results are summarised in TABLE I:

Model	RMSE	MAE	R^2
Linear Regression	0.1168	0.1016	0.3664
Random Forest	0.1366	0.1199	0.1334
XGBoost	0.1360	0.1211	0.1417

Table 1: Model Performance Metrics

Of the three, Linear Regression came out on top with an R^2 of 0.3664, edging out both XGBoost and Random Forest on the held-out test set.

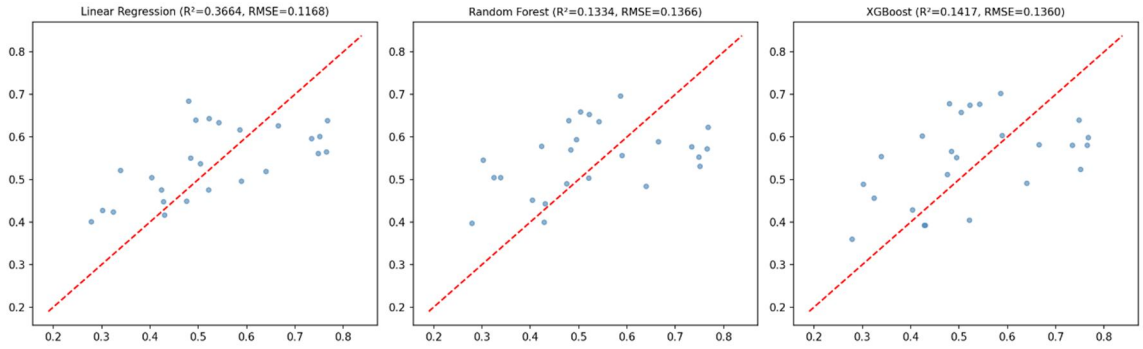


Fig. 3. Predicted vs Actual Thermal Efficiency for all three models. Linear Regression achieves the tightest clustering.

C. Why Linear Regression Outperforms Ensembles

Arguably the most counterintuitive finding here is that the simplest model wins. OLS Linear Regression, with no hyperparameters and no ensemble machinery, achieves the highest R^2 of the three. The explanation lies in what the autoregressive features actually do to the problem's structure. Without them, the relationship between instantaneous irradiance and efficiency is genuinely non-linear; thermal lag scrambles any clean mapping. But once Efficiency_lag10 is in the picture, the model is essentially extrapolating a slowly drifting thermal process one step forward, which turns out to be a much more linear task. On a dataset of this size, the added complexity of Random Forest and XGBoost stops being an advantage. Rather than capturing real signal, the ensemble methods end up fitting noise in the training data, and that hurts them on the test set.

D. The R^2 Paradox: Empirical vs. Synthetic Data

An R^2 of 0.3664 might look underwhelming at first glance, especially against benchmarks commonly seen in the literature where models hit $R^2 > 0.95$. But that comparison is not quite fair. Those high scores are typically achieved on simulation data, where the model is essentially learning to reverse-engineer a known mathematical function, a much easier task by construction. What we have here is fundamentally different. The target variable is derived from independent, live sensor readings. There is no closed-form equation connecting the inputs to efficiency; the relationship is shaped by real-world physics, thermal lag, convective losses, sensor noise, and the absence of dedicated pyranometer hardware. In that context, capturing roughly 37% of the total variance is a meaningful result. It reflects genuine predictive skill on a physically complex system, not a numerical artifact of a clean simulation environment.

IV. CONCLUSION

In this study, a machine learning pipeline was developed to predict the thermal efficiency of a commercial solar water heating plant using raw SCADA telemetry. The results show that even noisy, unfiltered SCADA data can be effectively cleaned and paired with clear-sky irradiance estimates from pvlib to create a reliable training dataset. A major takeaway from the data was the heavy impact of thermal lag, which makes it very difficult to predict a commercial system's efficiency at any exact moment.

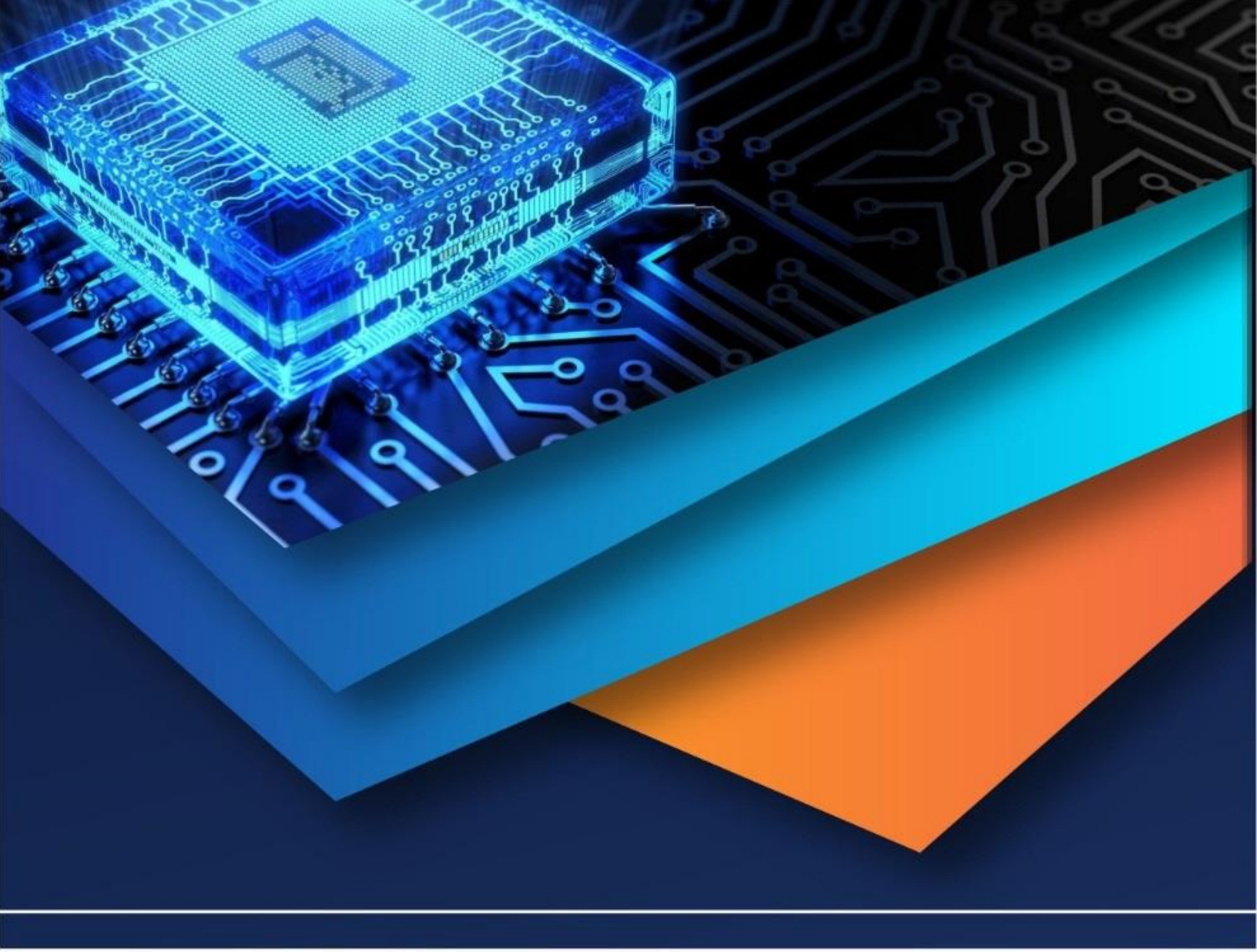
To get around this issue, we introduced autoregressive features into the model, specifically a 60-minute rolling average of irradiance and a 10-minute lag of the efficiency itself. This strategy successfully handled the system's thermal inertia, bumping the baseline R^2 score from a low 0.08 up to 0.3664. Interestingly, adding these time-based features smoothed out the data's non-linear behavior so well that a simple Linear Regression model actually ended up performing better than much heavier ensemble algorithms like Random Forest and XGBoost.

V. ACKNOWLEDGMENT

The authors wish to acknowledge the Department of Mechanical Engineering, Delhi Technological University, for providing the necessary guidance and resources to conduct this research, and the open-source community for publishing the SCADA dataset.

REFERENCES

[1] H. C. Hottel and A. Whillier, "Evaluation of flat-plate solar collector performance," Trans. Conf. Use of Solar Energy, vol. 2, pp. 74–104, 1955.
[2] R. W. Bliss, "The derivations of several 'plate efficiency factors' useful in the design of flat-plate solar heat collectors," Solar Energy, vol. 3, no. 4, pp. 55–64, 1959.
[3] J. A. Duffie and W. A. Beckman, Solar Engineering of Thermal Processes, 4th ed. John Wiley & Sons, 2013.
[4] S. A. Kalogirou, "Applications of artificial neural networks in energy systems: A review," Energy Conversion and Management, vol. 40, no. 10, pp. 1073–1087, 1999.
[5] I. Farkas and P. Géczy-Víg, "Neural network modelling of flat-plate solar collectors," Computers and Electronics in Agriculture, vol. 40, pp. 87–102, 2003.
[6] H. K. Ghritlahre and R. K. Prasad, "Prediction of thermal performance of unidirectional flow porous bed solar air heater with optimal training function using Artificial Neural Network," Energy Procedia, vol. 109, pp. 369–376, 2017.
[7] C. Voyant et al., "Machine learning methods for solar radiation forecasting: A review," Renewable Energy, vol. 105, pp. 569–582, 2017.
[8] R. Srivastava, A. N. Tiwari, and V. K. Giri, "Solar radiation forecasting using MARS, CART, M5, and Random Forest model: A case study for India," Heliyon, vol. 5, no. 10, e02692, 2019.
[9] C. Fan, F. Xiao, and Y. Zhao, "A short-term building cooling load prediction method using deep learning algorithms," Applied Energy, vol. 195, pp. 222–233, 2017.
[10] X. Qing and Y. Niu, "Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM," Energy, vol. 148, pp. 461–468, 2018.
[11] P. Kumari and D. Toshniwal, "Deep learning models for solar irradiance forecasting: A comprehensive review," Journal of Cleaner Production, vol. 318, 128566, 2021.
[12] K. Leahy et al., "Issues with Data Quality for Wind Turbine Condition Monitoring and Reliability Analyses," Energies, vol. 12, no. 2, 201, 2019.
[13] J. Zhang et al., "A suite of metrics for assessing the performance of solar power forecasting," Solar Energy, vol. 111, pp. 157–175, 2015.
[14] R. Kumar and M. A. Rosen, "A critical review of photovoltaic-thermal solar collectors for air heating," Applied Energy, vol. 88, no. 11, pp. 3603–3614, 2011.
[15] A. Shukla, D. Buddhi, and R. L. Sawhney, "Solar water heaters with phase change material thermal energy storage: A review," Renewable and Sustainable Energy Reviews, vol. 13, no. 8, pp. 2119–2125, 2009.
[16] stritti, "Realtime Thermal Solar Plant Dataset," GitHub, 2018. [Online]. Available: github.com/stritti/thermal-solar-plant-dataset
[17] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[18] W. F. Holmgren, C. W. Hansen, and M. A. Mikofski, "pvlib python: A python package for simulating solar energy systems," Journal of Open Source Software, vol. 3, no. 29, p. 884, 2018.
[19] P. Ineichen and R. Perez, "A new airmass independent formulation for the Linke turbidity coefficient," Solar Energy, vol. 73, no. 3, pp. 151–157, 2002.
[20] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[21] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," Proc. 22nd ACM SIGKDD, pp. 785–794, 2016.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY