

Optimizing Forecasting of Rooftop Solar PV Energy Generation for Peak-Load Management

Pramod Kumar Gouda¹, Potnuru Charishma², Badana. Bharadwaj³, Kalivarapu Lokesh⁴ and Nowpada. Jitendra⁵

¹⁻⁵Aditya Institute of Technology and Management, Tekkali, Electrical and Electronic Engineering Sri-kakulam, Andhra Pradesh 532201, India

Email: potnurucharishma2003@gmail.com, {pkgouda18@gmail.com, jnowpada143@gmail.com, kalivarapulokesh@gmail.com, bharadwaj70@gmail.com}

Abstract— To ensure a stable and sustainable power grid, it is imperative to implement sophisticated management strategies that effectively address the inherent variability and unpredictability of renewable energy generation. This research focuses on developing an advanced energy management system for buildings with photovoltaic (PV) systems, integrating continuous energy monitoring and forecasting intelligence. Through the examination of past data, meteorological trends, and current energy consumption patterns, the system forecasts future energy needs, identifies peak load periods, and optimizes energy consumption to maximize on-site solar generation. The next-day forecasting approach for commercial photovoltaic systems is explored through the integration of weather prediction models and technical specifications of PV plants to predict power output, incorporating battery storage and optimal control algorithms to address over-generation and reduce reliance on conventional power sources. Performance is validated using RMSE and MAE metrics. The study also examines energy consumption prediction in buildings with a combination of statistical analysis, time-series modelling, and machine learning methods applied to a five-year dataset (2020–2024), utilizing ARMA, ARIMA, and SARIMA models to generate prediction of Solar Irradiance. Additionally, it highlights the role of PV systems in office buildings for peak shaving, improving economic efficiency and grid stability. For residential applications, a novel residential energy management system (REMS) is proposed to manage power fluctuations, leveraging Python for effective control and optimization.

I. INTRODUCTION

The increasing adoption of photovoltaic (PV) systems in buildings presents a significant opportunity to enhance energy efficiency and grid independence, but effective management of this distributed generation requires sophisticated strategies. Accurate peak load forecasting is crucial for optimizing energy consumption, minimizing grid reliance, and ensuring grid stability. This challenge is addressed by developing intelligent energy management systems that integrate real-time monitoring with advanced machine learning to predict future energy needs, identify potential peak load periods, and maximize on-site solar utilization [1], [2], [3]. Efficient energy utilization is critical for addressing energy shortages and environmental concerns [4]. Energy monitoring systems maximize PV benefits by providing insights into generation and consumption, enabling timely detection of inefficiencies, and optimizing energy distribution, supporting renewable energy integration in smart grids, and contributing to global energy efficiency and carbon emission reduction efforts [5].

Variability in solar energy generation due to environmental factors necessitates advanced forecasting techniques that integrate real-time monitoring and predictive analytics. By leveraging machine learning-based time series models, such as ARMA, ARIMA, and SARIMA, energy demand trends can be effectively analyzed to improve system efficiency [6], [7], [8]. Additionally, the integration of automated control strategies and optimization techniques enables better energy distribution, ensuring that excess solar power is efficiently utilized or stored, further enhancing the resilience of smart buildings [9], [10]. Furthermore, forecasting in Python is conducted using statistical methods, including equation-based and state-space models, to ensure accurate energy demand predictions [11], [12]. These models leverage time-series analysis techniques to capture underlying trends, variations, and dependencies in energy consumption data, enhancing the reliability of peak load forecasting [12], [13]. For example, existing PV systems face challenges with poor real-time monitoring, expensive wired communication, and limited data analysis, resulting in energy inefficiencies [14]. Lavanya et al.'s study showed PV energy output exceeded heat pump consumption (6% contribution) but couldn't cover total en-ergy demand [15]. Markovics and Mayer compared 24 machine learning models for PV forecasting, using two-year da-ta, finding kernel ridge and MLP best (44.6% skill improvement) [16]. Optimized inputs reduced RMSE by 13.1%, and tuning improved over linear regression by 13.9%, demonstrating machine learning's effectiveness in improving energy forecasting accuracy even with limited data availability.

II. METHODOLOGY

This study develops a PV-based peak load forecasting system methodology for buildings, integrating real-time monitoring, predictive modelling, and optimization to improve energy management and grid dependency.

Data Acquisition Data collection is a critical aspect of this study, ensuring robust forecasting and system optimization. Various sources contribute to the dataset:

Weather Data Meteorological variables, including temperature (in °C), solar irradiance (in W/m²), humidity (in %), and wind speed (in m/s), are retrieved from weather monitoring stations and online databases across India.

Historical Energy Usage: Past energy consumption patterns are collected from building energy management systems to establish baseline trends and facilitate predictive analysis. **Grid Load Data:** Data from utility providers help in understanding peak de-mand patterns, ensuring that the forecasting system aligns with actual grid requirements.

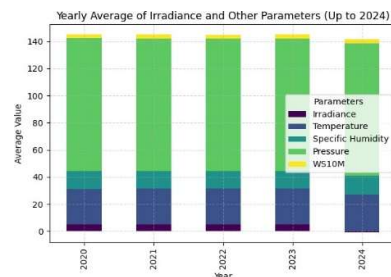


Figure1: Corresponding solar parameters per Annam

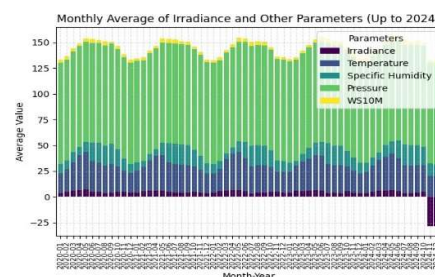


Figure2: Corresponding solar parameters per month

TABLE I: LOCATION DATA (INDIA)

Coordinate Type	Minimum Value	Maximum Value
Latitude	8°4' North	37°6' North
Longitude	68°7' East	97°25' East



Figure 3: Location (India)

III. DATA PRE-PROCESSING

Data preprocessing is required to improve data quality and prepare it for analysis. Cleaning and imputation are used to handle missing or invalid values, time-series smoothing techniques like exponential smoothing and moving averages are applied to decrease noise, and the dataset is divided into training (80%) and testing (20%) sets. To increase model accuracy, feature engineering is carried out by determining time indicators (hour, day, month), lag values, rolling statistics, and seasonal/trend-based indicators. Heatmap visualizations are used to find connections in order to comprehend feature relationships. Additionally, the Partial Autocorrelation Function (PACF) assists in determining the direct correlation at particular lags, while the Autocorrelation Function (ACF) analyses the autocorrelation structure of the time-series data to discover relevant lags. Selecting suitable time-series forecasting models such Autoregressive Moving Average (ARMA), which models the current value as a linear combination of past values and past forecast errors, and Seasonal Autoregressive Integrated Moving Average (SARIMA), an extension of ARIMA explicitly accounting for the seasonality present in the time-series data by including seasonal AR, MA, and differencing terms, depends critically on the insights from ACF and PACF plots. These comprehensive preprocessing steps and model considerations ensure a robust and informative approach for subsequent forecasting and analysis.

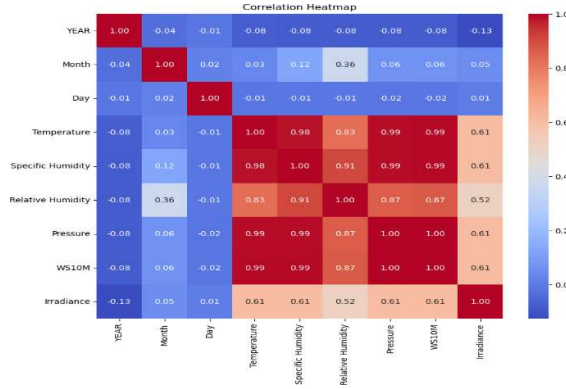


Figure 4: Heatmap of Solar Parameters

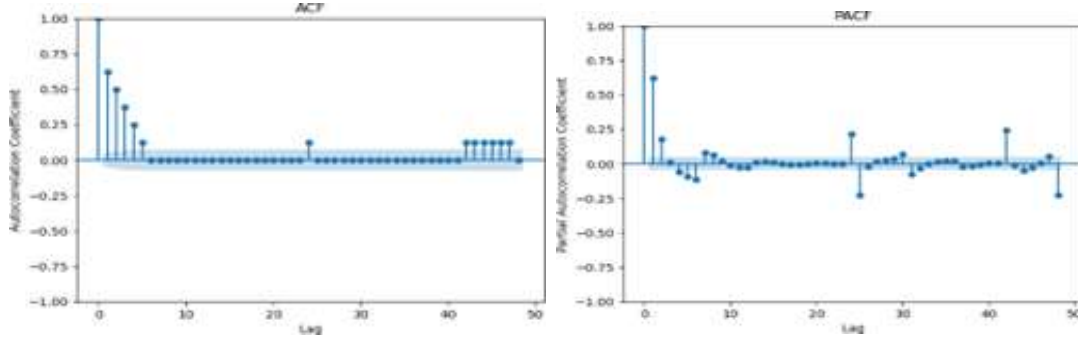


Figure 5: Autocorrelation function and Partial Autocorrelation function

A. Forecasting Techniques

To predict energy demand and PV generation accurately, advanced time-series models are utilized, implemented in Python.

B. Autoregressive Moving Average (ARMA)

ARMA is employed for stationary time-series forecasting, incorporating both autoregressive (AR) and moving average (MA) components to capture dependencies in the dataset. Used when the dataset exhibits no trend or seasonality after preprocessing.

C. Model Representation

$$(1 - \phi_1 L)y_t = c + (1 + \theta_1 L)\varepsilon_t \quad (1)$$

D. Autoregressive Integrated Moving Average (ARIMA)

ARIMA is applied to non-stationary data, incorporating a different component to stabilize the time series before applying ARMA modelling. This model is useful when trends exist in historical data that impact future energy demand.

E. Model representation

$$(1 - \phi_1 L)(1 - L)y_t = c + (1 + \theta_1 L)\varepsilon_t \quad (2)$$

F. Seasonal ARIMA (SARIMA)

Extends ARIMA by including seasonal patterns, critical for applications where periodic fluctuations in energy demand occur (e.g., higher demand during summer due to air conditioning use).

General notation: SARIMA models are represented as SARIMA (p, d, q). Represent non-seasonal ARIMA components. Where: (p, d, q) are the non-seasonal AR, I, and MA orders.

$$(1 - \phi_1 L)(1 - L)y_t = c + (1 + \theta_1 L)\varepsilon_t \quad (3)$$

G. Implementation Platforms

To ensure comprehensive analysis, forecasting models are developed using two different computational environments. Python Implementation: Libraries Used: Python offers an extensive range of libraries, including Scikit-learn for machine learning, State-models for statistical forecasting and prediction models. The methodology of this research is built around three central functions. To begin with, data preprocessing and feature engineering are carried out in order to improve data quality and relevance. Second, time-series forecasting models such as ARMA, ARIMA, and SARIMA are applied to forecast energy consumption. Third, the precision and efficiency of these models are evaluated using proper error metrics in a rigorous as-assessment.

H. Energy Management Optimization:

The predicted energy demand and PV generation results are used to optimize energy consumption of Irradi-ance and improve grid interaction. Grid Interaction Strategy by Implementing demand response techniques to sell surplus energy back to the grid or balance load based on peak pricing models.

I. Performance Evaluation

The accuracy and reliability of the forecasting models are assessed using the following evaluation metrics:

J. Root Mean Square Error (RMSE):

Measures the average deviation between predicted and actual values, penalizing larger error.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (f_i - o_i)^2} \quad (4)$$

Where,
N=f/o,
f=Forecast values,
o=Observation values

K. Mean Absolute Error (MAE):

Evaluates absolute differences between predictions and observed values, providing a straightforward measure of accuracy.

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{N} \quad (5)$$

Where,
yi=Prediction values,
xi=Observation values,
N=Total number of data points
R-squared (R²) Score:

Determines how well the predictive model explains the variance in energy demand and generation data.

$$R^2=1-RSS/TSS \quad \text{-----}(6)$$

Where,

R^2 =Coefficient of determination,

RSS= sum of squares of residuals

TSS=Total number of Squares.

L. Autocorrelation Function (ACF)

The Autocorrelation Function (ACF) is an important diagnostic tool in time series forecasting, which tests model fit by verifying residuals are white noise. Large lags beyond confidence limits signal model misspecification, and parameters (p, d, q) need to be adjusted. A well-patterned ACF indicates the model detects temporal structure, making reliable forecasts. It identifies best ARMA, ARIMA, and SARIMA parameters, seasonality, and noise.

Prior to model finalization, residual ACF is examined to ensure that errors are white noise-like— meaning a good fit. Persistent autocorrelations indicate underfitting, while a clean ACF ensures that the model has successfully picked up temporal patterns, making accurate forecasts.

The diagram further specifies that within specific time-series forecasting models are being employed: Figure (6):

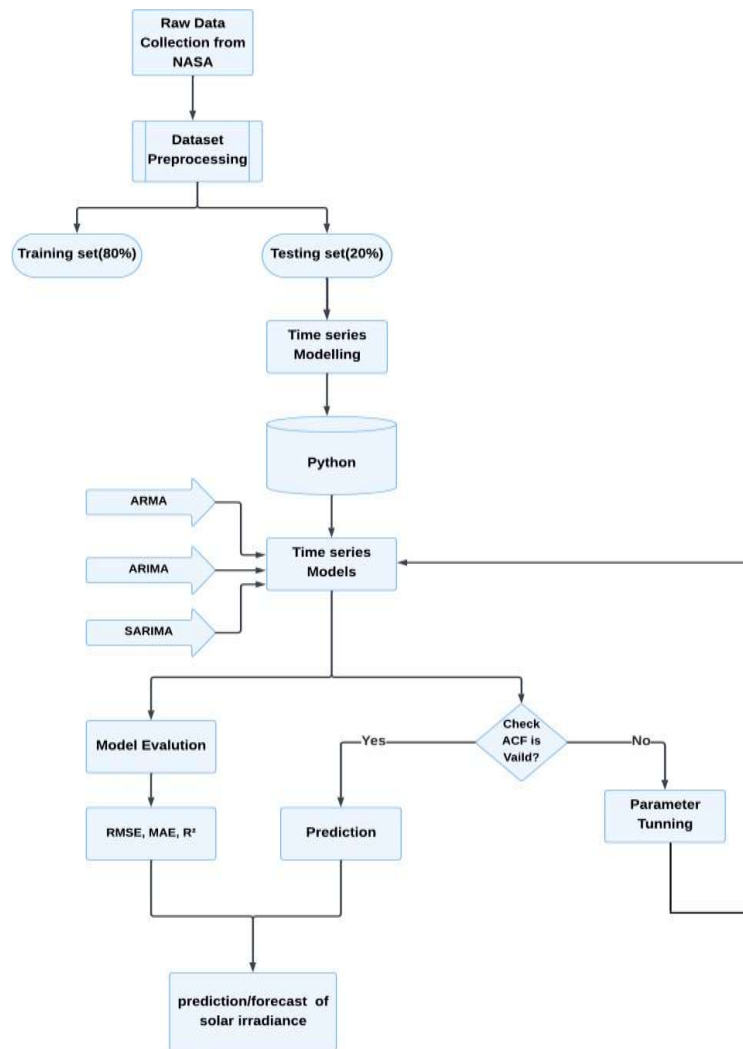


Figure 6: Workflow of the Proposed Forecasting

IV. RESULTS AND DISCUSSION

The proposed monitoring and forecasting system leverages the versatility in Python, employing independent technique to produce reliable peak load predictions.

A. Graphical Analysis of Time series Graphs

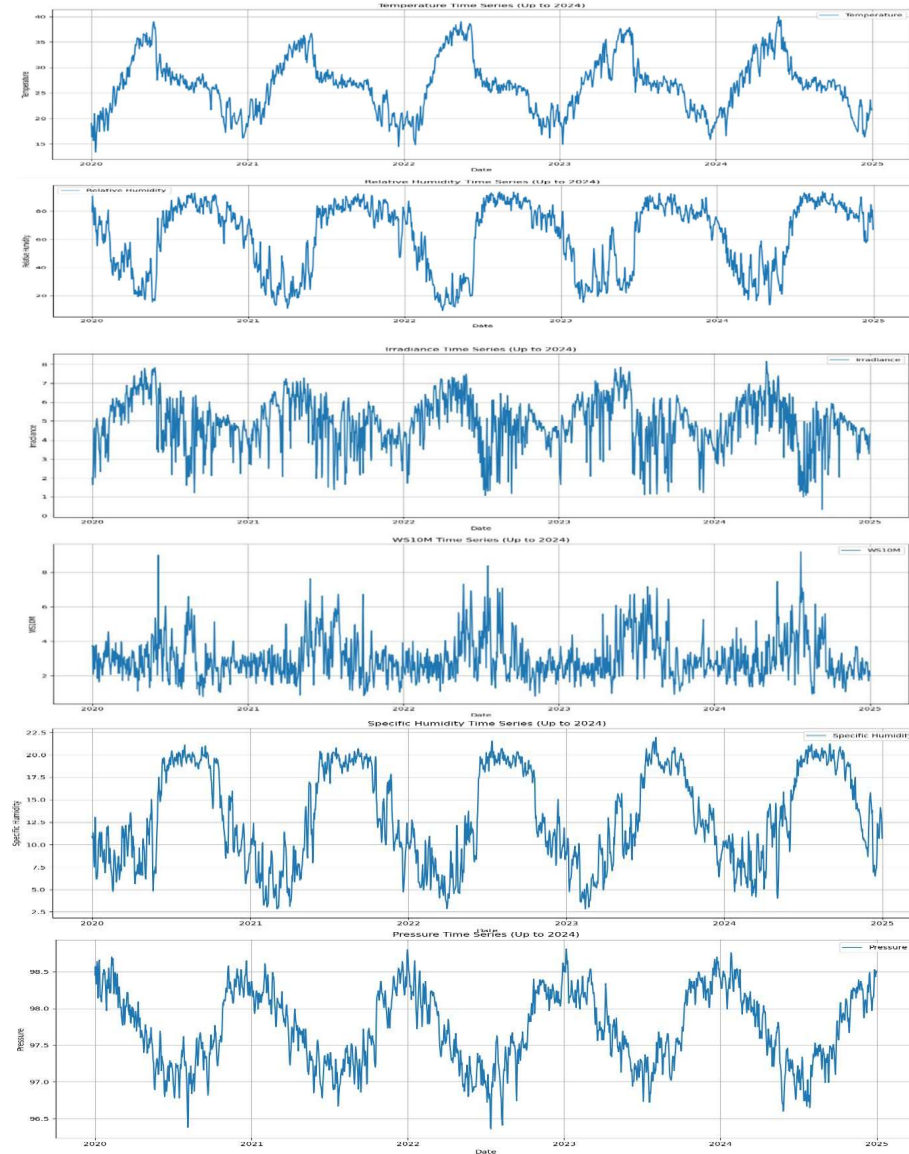


Figure7: Time series Graphs of solar parameters

The time-series forecasting graphs generated for all parameters as per the dataset provide insights into the accuracy and reliability of each model in predicting peak load demand. This figure2 presents clean and consistent time series trends for six key weather parameters, all maintaining values above zero. The uniform structure highlights strong data continuity, with identifiable placeholders (e.g., -1000) that can be easily managed during preprocessing to enhance model performance. To ensure the quality of our forecasting models, we applied thorough preprocessing steps to clean the dataset. All invalid entries, including placeholders like -1000, as well as any NaN or null values, were effectively identified and removed. This helped us retain only meaningful data, improving both the accuracy and reliability of the time series analysis moving forward.

B. ARMA Method

This study uses the ARMA method to forecast power generation from real-time data (2020–2024), considering key parameters to capture dependencies and external factors.

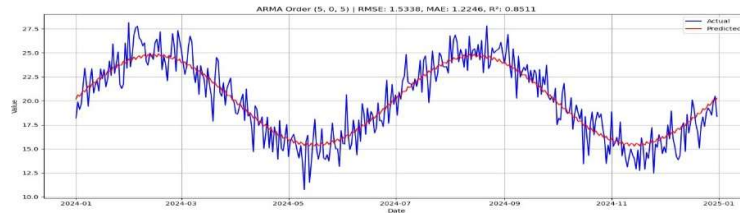


Figure8: ARMA Graph of Irradiance

The ARMA model graph demonstrates a strong correlation between predicted and actual values, with minimal deviation. The predicted trend closely follows the actual data, effectively capturing short-term fluctuations. Since ARMA is best suited for stationary data without significant trends or seasonality, it performs exceptionally well in datasets where energy consumption remains stable. Minimal prediction errors are observed during peak demand periods, highlighting ARMA's efficiency in short-term load forecasting.

C. ARIMA Method

This study uses the ARMA method to forecast power generation from real-time data (2020–2024), considering key parameters to capture dependencies and external factors.

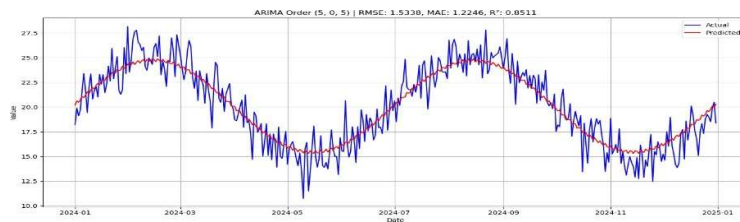


Figure 9: ARIMA Graph of Irradiance

The ARIMA model graph reveals a slight lag in peak load prediction due to trend incorporation, leading to occasional overestimation or underestimation. While effective for non-stationary data, its deviations from actual values are more pronounced than ARMA, particularly in short-term fluctuations. Higher error margins during sudden demand shifts indicate ARIMA's reduced precision for short-term forecasting.

D. SARIMA Method

This study uses the ARMA method to forecast power generation from real-time data (2020–2024), considering key parameters to capture dependencies and external factors.

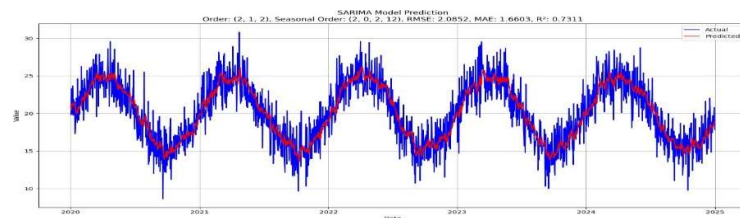


Figure10: SARIMA Graph of Irradiance

The SARIMA model captures seasonal patterns, but its predictions show instability and larger deviations, especially during abrupt demand shifts. Higher RMSE and MAE values confirm its limitations in short-term energy forecasting. With minimal seasonal influence in the dataset, SARIMA adds unnecessary complexity, leading to lower accuracy than ARMA and ARIMA.

Now, within the PYTHON environment, time-series forecasting was primarily conducted utilizing the Econometrics Toolbox, enabling the implementation of ARMA, ARIMA and SARIMA models for accurate peak load prediction.

E. Visualization and Evaluation of Irradiance Forecast Accuracy

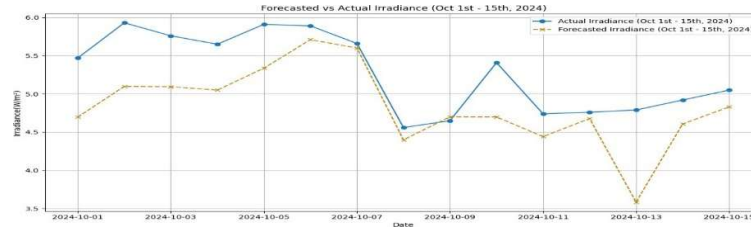


Figure11: Solar Irradiance Forecast

Fig. 11 presents a comparative visualization of actual versus forecasted solar irradiance over the period from October 1st to October 15th, 2024. The plot demonstrates the performance of the forecasting model in capturing temporal variations in irradiance. The actual irradiance data exhibits typical fluctuations influenced by atmospheric and environmental conditions, while the forecasted values aim to replicate this dynamic behavior. Notably, the pre-dicted trend closely follows the actual data during peak irradiance days, indicating strong shortterm predictive capability. This graphical comparison reinforces quantitative evaluation by highlighting the model's effectiveness in accurately forecasting solar irradiance, thereby supporting its suitability for energy planning in photovoltaic-integrated systems.

F. Graphical Comparison and Best Model Selection

A comprehensive evaluation of the Actual vs. Predicted graphs, along with key performance metrics, indicates that the ARMA model provides the most accurate forecasts for energy demand. It closely aligns with the actual data while yielding the lowest RMSE and MAE values. The ARIMA model demonstrates moderate performance but exhibits minor overfitting or underfitting tendencies due to trend adjustments. In contrast, the SARIMA model proves to be the least effective, as seasonality does not significantly influence energy demand in this dataset, making it less suitable for this application.

G. Forecasting Model Performance

To assess the accuracy and reliability of the proposed forecasting models, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) were employed as performance evaluation metrics. RMSE quantifies the average magnitude of prediction errors while penalizing larger deviations, whereas MAE provides a straightforward measure of absolute forecast error. Lower RMSE and MAE values indicate a more precise model, ensuring accurate peak load forecasting. The comparative performance of the ARMA, ARIMA, and SARIMA models is summarized in Table 2.

TABLE II: FORECASTING ACCURACY METRICS

Forecasting Model	RMSE	MAE	R-squared
ARMA	1.5338	1.2246	0.8511
ARIMA	1.5338	1.2246	0.8511
SARIMA	2.0852	1.6603	0.7311

Among the evaluated models, ARMA and ARIMA demonstrated the highest accuracy, both yielding the lowest RMSE (1.5338), MAE (1.2246), and a strong R-squared value (0.8511). These results indicate that the simpler models without seasonal components effectively captured the underlying patterns in the data. SARIMA, which accounts for seasonality, showed reduced performance with a higher RMSE (2.0852), MAE (1.6603), and a lower R-squared value (0.7311), suggesting that seasonal adjustments offered little benefit in this context. Overall, ARMA and ARIMA out-performed SARIMA by approximately 36% in RMSE, 35.6% in MAE, and 14.1% in R-squared, confirming their superiority for this forecasting task.

V. CONCLUSION

This research established a strong forecasting model to maximize rooftop photovoltaic (PV) energy production for peak load management in buildings. Through the combination of real-time monitoring and traditional time series models—ARMA, ARIMA, and SARIMA—the research successfully predicted energy demand with five years of weather and load data. Among the models, ARMA and ARIMA performed better than SARIMA,

providing good pre-dictive performance with high R^2 values (0.8511) and low RMSE and MAE, particularly under scenarios with little sea-sonal variation. Despite these models providing simplicity, transparency, and low computational expense, there are some weaknesses. Shortcomings include dependence on linear assumptions, data stationarity sensitivity, and lower ca-pacity to extract nonlinear or extremely dynamic patterns. SARIMA, for instance, brought unnecessary complexity to a dataset with poor seasonality, resulting in reduced accuracy. Despite such limitations, the models still provided suc-cessful results—very high forecasting accuracy and stable R^2 values. This proves that with appropriate preprocessing, parameter optimization, and model selection, classic time series models can still provide accurate and actionable pre-dictions for smart grid applications. The system facilitates better energy efficiency, grid stability, and well-informed energy management decisions in commercial and residential buildings.

REFERENCES

- [1] H.-K. Ringkjøb, P. M. Haugan, and I. M. Solbrekke, “A review of modelling tools for energy and electricity sys-tems with large shares of variable renewables,” *Renewable and Sustainable Energy Reviews*, vol. 96, pp. 440–459, Nov. 2018, doi: 10.1016/j.rser.2018.08.002.
- [2] D. A. Fadare, “Modelling of solar energy potential in Nigeria using an artificial neural network model,” *Applied Energy*, vol. 86, no. 9, pp. 1410–1422, Sep. 2009, doi: 10.1016/j.apenergy.2008.12.005.
- [3] M. L. Ferrari, “Advanced control approach for hybrid systems based on solid oxide fuel cells,” *Applied Ener-gy*, vol. 145, pp. 364–373, May 2015, doi: 10.1016/j.apenergy.2015.02.059.
- [4] A. Lertwiputh, C. Watcharopas, and P. Wattuya, “Forecasting of photovoltaic power using deep learning,” in *Proceedings of the 2024 6th Asia Conference on Machine Learning and Computing*, Bangkok Thailand: ACM, Jul. 2024, pp. 1–6. doi: 10.1145/3690771.3690774.
- [5] M. Habib, A. A. Ladjici, E. Bollin, and M. Schmidt, “One-day ahead predictive management of building hy-brid power system improving energy cost and batteries lifetime,” *IET Renewable Power Gen*, vol. 13, no. 3, pp. 482–490, Feb. 2019, doi: 10.1049/iet-rpg.2018.5454.
- [6] K. Goswami and A. B. Kandali, “Electricity Demand Prediction using Data Driven Forecasting Scheme: ARI-MA and SARIMA for Real-Time Load Data of Assam,” in *2020 International Conference on Computational Performance Evaluation (ComPE)*, Shillong, India: IEEE, Jul. 2020, pp. 570–574. doi: 10.1109/ComPE49325.2020.9200031.
- [7] A. Mystakidis, P. Koukaras, N. Tsalikidis, D. Ioannidis, and C. Tjortjis, “Energy Forecasting: A Comprehensive Review of Techniques and Technologies,” *Energies*, vol. 17, no. 7, p. 1662, Mar. 2024, doi: 10.3390/en17071662.
- [8] C. Tarmanini, N. Sarma, C. Gezezin, and O. Ozgonenel, “Short term load forecasting based on ARIMA and ANN approaches,” *Energy Reports*, vol. 9, pp. 550–557, May 2023, doi: 10.1016/j.egyr.2023.01.060.
- [9] E. Sarmas, E. Spiliotis, E. Stamatopoulos, V. Marinakis, and H. Doukas, “Short-term photovoltaic power fore-casting using meta-learning and numerical weather prediction independent Long Short-Term Memory mod-els,” *Renewable Energy*, vol. 216, p. 118997, Nov. 2023, doi: 10.1016/j.renene.2023.118997.
- [10] M. A. Alemu and M. Zegeye. Getie, “Experimental study on the performance of household electric cooking stoves: Locally made versus imported technologies,” *Energy Reports*, vol. 11, pp. 535–543, Jun. 2024, doi: 10.1016/j.egyr.2023.11.045.
- [11] E. Mocanu et al., “On-line Building Energy Optimization using Deep Reinforcement Learning,” 2017, arXiv. doi: 10.48550/ARXIV.1707.05878.
- [12] H. Kalantar-Neyestanaki and M. Soltani, “Uncertainty-Cognizant Model Predictive Control for Energy Man-agement of Residential Buildings with PVT and Thermal Energy Storage,” 2022, arXiv. doi: 10.48550/ARXIV.2201.08909.
- [13] J. Barrales-Ruiz and P. Neudörfer, “The oil price (Ir)relevance for global CO2 emissions,” *Energy Reports*, vol. 11, pp. 3016–3021, Jun. 2024, doi: 10.1016/j.egyr.2024.02.044.
- [14] L. A et al., “Smart energy monitoring and power quality performance based evaluation of 100-kW grid tied PV system,” *Heliyon*, vol. 9, no. 6, p. e17274, Jun. 2023, doi: 10.1016/j.heliyon.2023.e17274.
- [15] D. Markovics and M. J. Mayer, “Comparison of machine learning methods for photovoltaic power forecasting based on numerical weather prediction,” *Renewable and Sustainable Energy Reviews*, vol. 161, p. 112364, Jun. 2022, doi: 10.1016/j.rser.2022.112364.
- [16] N. C. Batista, R. Melício, J. C. O. Matias, and J. P. S. Catalão, “Photovoltaic and wind energy systems moni-toring and building/home energy management using ZigBee devices within a smart grid,” *Energy*, vol. 49, pp. 306–315, Jan. 2013, doi: 10.1016/j.energy.2012.11.002.