International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

S.U.R.A.J. (Solar Utility & Radiance Analytical Judgment): Localized ML-Based Forecasting for Diverse Indian Climates

Author Details

Pragati Rajput[1] , Yashika Soni[2] , Khushi Tamrakar[3] , Manavi Pardhi[4]

[1 2 3 4]Department of Computer Science & Engineering,
Shri Ram Institute of Technology, RGPV, Jabalpur, Madhya Pradesh, India
Mrs. Sweta Kriplanil[5]

[5] Professor, Department of Computer Science & Engineering,
Shri Ram Institute of Technology, RGPV,
Jabalpur, Madhya Pradesh, India
Corresponding Author: Pragati Rajput
Email: rajputpragati13@gmail.com

https://doi.org/10.55041/ijsmt.v2i5.276

Cite this Article: Rajput, P., Soni, Y., Tamrakar, K. & Pardhi, M. (2026). S.U.R.A.J. (Solar Utility & Radiance Analytical Judgment): Localized ML-Based Forecasting for Diverse Indian Climates. International Journal of Science, Strategic Management and Technology, *02*(05). https://doi.org/10.55041/ijsmt.v2i5.276

Abstract —

Accurate prediction of daily solar irradiance is critical for smart grid stability, energy storage planning, and photovoltaic system optimization. Existing forecasting models are predominantly trained on single geographic regions, limiting their applicability across India's climatically diverse landscape. This paper presents S.U.R.A.J. (Solar Utility & Radiance Analytical Judgment), a machine-learning-based solar energy forecasting system validated across six geographically and climatically distinct Indian cities: Jabalpur, Bhopal, Delhi, Mumbai, Jaipur, and Ladakh — representing tropical, semi-arid, arid, coastal, and cold-arid climate zones. The system employs an 11-step reproducible data pipeline utilizing five years (2019–2023) of daily meteorological observations acquired from the NASA POWER satellite API, comprising 1,826 data points per city. A Random Forest Regressor is independently trained for each city using 17 engineered features, including cyclical temporal encodings, autoregressive lag variables (Solar_Lag_1, Solar_Lag_7), and rolling statistical features (Solar_Roll7, Solar_Roll30), applied to an 80/20 chronological train-test split. City-specific models achieve R^2 scores ranging from 0.830 (Ladakh, cold-arid) to 0.891 (Mumbai, coastal), with Mean Absolute Errors of 0.337–0.524 MJ/m², substantially outperforming the naive persistence baseline across all locations. Feature importance analysis consistently identifies cloud fraction and previous-day irradiance as the dominant predictors of daily solar output. The trained models are deployed through a Flask REST API connected to an interactive web dashboard, enabling real-time solar irradiance simulation via adjustable environmental controls. Results confirm that climate-specific ensemble modeling with engineered temporal features produces reliable, interpretable, and operationally deployable solar forecasts for diverse Indian cities.

Keywords— Solar Irradiance Forecasting; Random Forest; NASA POWER API; Machine Learning; Smart Grid; Feature Engineering; Multi-Climate Modeling; India

1. INTRODUCTION

The global demand for clean energy has accelerated the deployment of solar photovoltaic (PV) systems as a primary renewable energy source. In India, solar energy occupies a central position in national energy policy, with the government targeting 500 GW of non-fossil fuel capacity by 2030 [1]. However, solar irradiance is inherently intermittent — its output fluctuates with cloud cover, temperature, humidity, and seasonal variation — making reliable advance prediction essential for grid stability, energy storage scheduling, and curtailment minimization.

Traditional forecasting approaches, including naive persistence models and statistical methods such as ARIMA, assume that tomorrow's conditions will closely resemble today's. While computationally simple, these methods fail to capture the non-linear, multi-variable dynamics of atmospheric systems, particularly during abrupt weather transitions such as sudden cloud formation or monsoon onset [2]. The consequent prediction errors force grid operators to maintain costly spinning reserves, increasing both operational expenditure and carbon emissions.

Machine learning techniques have demonstrated strong capability in modeling complex relationships between meteorological variables and solar irradiance. Ensemble methods — particularly the Random Forest Regressor — have shown consistent advantages over single estimators, reducing variance through bootstrap aggregation while providing interpretable feature importance rankings [3]. However, a critical limitation persists in existing literature: the majority of ML forecasting models are trained and validated within geographically homogeneous regions. A model optimized for a coastal climate performs poorly when applied to an arid or high-altitude setting due to fundamentally different thermodynamic atmospheric profiles [4]. For a subcontinent as climatically diverse as India — spanning tropical, semi-arid, arid, coastal, and cold-arid zones — this represents a significant deployment barrier.

A second gap exists at the translational level. High-performing ML models typically produce raw numerical outputs accessible only through technical interfaces, limiting their utility for operational energy professionals and grid managers who require immediate, interpretable, and actionable decision support [5].

This paper addresses both gaps through the introduction of S.U.R.A.J. (Solar Utility & Radiance Analytical Judgment), an end-to-end solar irradiance forecasting system designed for climatically diverse Indian cities. The system makes the following primary contributions:

- An 11-step reproducible data pipeline that processes five years (2019–2023) of NASA POWER satellite data for six Indian cities covering five distinct climate zones.
- Independent Random Forest Regressor models trained per city using 17 engineered features, achieving R^2 scores of 0.830–0.891 across all locations.
- A Flask-powered REST API connected to an interactive web dashboard enabling real-time solar energy simulation through adjustable environmental parameters.
- A comparative performance analysis validating city-specific ensemble modeling against the naive persistence baseline across tropical, semi-arid, arid, coastal, and cold-arid climates.

The remainder of this paper is organized as follows: Section 2 reviews relevant literature on solar forecasting methodologies; Section 3 describes the system architecture; Section 4 details the methodology implementation; Section 5 presents results and discussion; Section 6 concludes with future directions.

2. LITERATURE REVIEW

The field of solar irradiance forecasting has evolved through three broad phases: statistical methods, physical simulation models, and data-driven machine learning approaches. Early work relied predominantly on Auto-Regressive Integrated Moving Average (ARIMA) models and exponential smoothing techniques, which operate under the assumption that future atmospheric conditions can be extrapolated linearly from historical observations. While computationally inexpensive, these methods demonstrate significant degradation in accuracy during periods of rapid meteorological change, such as monsoon transitions and sudden cloud cover events, due to their inability to model non-linear variable interactions [3].

The limitations of statistical approaches motivated the adoption of machine learning techniques for solar

forecasting. Artificial Neural Networks (ANNs) were among the first ML architectures applied to irradiance prediction, demonstrating improved accuracy over ARIMA baselines by learning non-linear mappings between meteorological inputs and solar output [4]. Support Vector Regression (SVR) subsequently demonstrated competitive performance, particularly in low-data environments, through its kernel-based approach to non-linear feature transformation. However, both ANN and SVR models require careful hyperparameter selection and are prone to overfitting when applied across diverse geographic conditions without retraining.

Ensemble tree-based methods, particularly Random Forest (RF) and Extreme Gradient Boosting (XGBoost), have emerged as the most consistently performant approaches for daily solar irradiance forecasting. Breiman (2001) established the theoretical foundation of Random Forest, demonstrating that bootstrap aggregation across an ensemble of decorrelated decision trees reduces prediction variance without increasing bias [5]. Voyant et al. (2017) conducted a comprehensive survey of ML methods for solar forecasting across 20 studies, concluding that ensemble methods achieved R^2 scores between 0.80 and 0.93 for daily irradiance prediction, consistently outperforming single estimators and statistical baselines [6]. The built-in feature importance estimation of Random Forest provides an additional interpretability advantage critical for scientific validation.

The NASA POWER (Prediction of Worldwide Energy Resources) project has been increasingly adopted as a primary data source for solar energy research in regions with limited ground-station coverage. Stackhouse et al. (2018) validated NASA POWER solar radiation estimates against ground-based pyranometer measurements across 20 climate zones globally, reporting mean bias errors below 5%, confirming the dataset's suitability for ML model training in data-sparse regions including much of rural India [7].

Research specific to Indian climate zones remains limited in scope and geographic coverage. Sharma and Kakkar (2018) applied SVR for solar forecasting across Rajasthan, achieving an R^2 of 0.81 but restricting validation to a single arid climate region [2]. Kumari and Toshniwal (2021) employed an LSTM-CNN hybrid network for solar irradiance forecasting across Indian locations, reporting competitive accuracy but noting degradation during high-variability monsoon periods [11].. Neither study extended validation beyond a single climate type, leaving multi-climate generalizability unaddressed.

A recurring limitation across existing literature is the absence of operational deployment. The majority of published models terminate at statistical evaluation, with outputs confined to research code repositories inaccessible to the energy professionals who stand to benefit most from accurate forecasting. This gap between academic accuracy and operational accessibility represents a significant barrier to real-world adoption of ML-based solar forecasting in developing nations [1].

S.U.R.A.J. directly addresses these identified limitations. By training and validating independent city-specific Random Forest models across six Indian cities spanning five climate zones, the system establishes a generalized yet locally optimized forecasting methodology. By deploying predictions through a Flask REST API and interactive web dashboard, it closes the translational gap between ML model outputs and actionable energy management tools — a contribution not present in any comparable study reviewed.

3. SYSTEM ARCHITECTURE

S.U.R.A.J. is designed as a two-layer architecture that separates the computationally intensive machine learning backend from the lightweight, user-facing prediction interface. This separation ensures that model inference remains independent of frontend rendering, enabling responsive real-time interaction without imposing processing overhead on the client side.

A. Machine Learning Backend

The backend is implemented as an 11-step sequential Python pipeline that transforms raw NASA satellite data into serialized, deployment-ready prediction models. The pipeline is organized into three functional layers:

Data Ingestion Layer: An automated retrieval script interfaces with the NASA POWER Daily Climatology API v2 using the Renewable Energy (RE) community dataset endpoint. For each of the six target cities, the script submits parameterized HTTP GET requests specifying the geographic coordinates, temporal range (20190101–20231231), and six meteorological

parameters: All-Sky Surface Shortwave Downward Irradiance (ALLSKY_SFC_SW_DWN, MJ/m²/day), Temperature at 2 Meters (T2M, °C), Relative Humidity at 2 Meters (RH2M, %), Cloud Amount Fraction (CLOUD_AMT, 0–100), Wind Speed at 2 Meters (WS2M, m/s), and Precipitation (PRECTOTCORR, mm/day). Responses are parsed from JSON into Pandas DataFrames and persisted as city-specific CSV files.

Data Processing and Feature Engineering Layer: Raw data is audited for NASA fill values (−999.0), which are replaced with NaN and resolved via forward-fill and linear interpolation. Date continuity is validated through Pandas DatetimeIndex reindexing to ensure no gaps exist in the daily time series. Seventeen predictive features are then engineered from the six raw variables: Month and Day of Year are encoded using sine-cosine cyclical

transformation to preserve seasonal adjacency in the feature space; a meteorological Season variable (0=Winter, 1=Pre-Monsoon, 2=Monsoon, 3=Post-Monsoon) is derived from month; three autoregressive lag features capture temporal autocorrelation (Solar_Lag_1, Solar_Lag_2, Solar_Lag_7); and two rolling mean features (Solar_Roll7, Solar_Roll30) are computed on shifted values to strictly prevent target leakage.

Model Training and Serialization Layer: The 17-feature dataset undergoes an 80/20 chronological train-test split — yielding 1,436 training and 360 test samples per city — without shuffling to preserve temporal ordering. A Random Forest Regressor is independently instantiated and trained for each city with the following configuration: n_estimators=200, max_depth=12, min_samples_leaf=3, random_state=42, n_jobs=−1. Each trained model is serialized to disk using Joblib, producing six city-specific .joblib files that are loaded once at server startup.

B. Flask REST API

A Flask application serves as the middleware layer connecting the trained models to the frontend. At startup, all six Joblib model files are loaded into memory. The API exposes two endpoints: a GET /status endpoint returning server health and available cities, and a POST /predict endpoint that accepts a JSON payload containing city name and four user-specified environmental parameters. Upon receiving a request, the backend constructs the complete 17-feature input vector by combining user inputs with current date-derived temporal features, invokes the city-specific model's predict() method, and returns the result as a JSON response. Cross-Origin Resource Sharing (CORS) is enabled via Flask-CORS to permit requests from the browser-based frontend.

C. Interactive Web Dashboard

The frontend is a single-page application built with HTML5, CSS3, and vanilla JavaScript, served directly by Flask as a static file. The interface presents a city selection dropdown populated with all six supported locations and four environmental input sliders: Cloud Fraction (0–100%), Temperature (10–45°C), Relative Humidity (10–100%), and Yesterday's Irradiance (0–10 MJ/m²). Each city selection pre-populates climatologically representative default slider values. On any user interaction, an asynchronous JavaScript fetch() call submits the current parameters to the /predict endpoint. The returned prediction value drives three simultaneous UI updates: the numerical readout, the fill level and color of an animated SVG semicircular gauge, and a status indicator pill. Output is classified into three efficiency tiers — High (≥7 MJ/m², cyan), Moderate (4–7 MJ/m², orange), and Low (<4 MJ/m², red) — providing immediate operational context without requiring numerical interpretation.

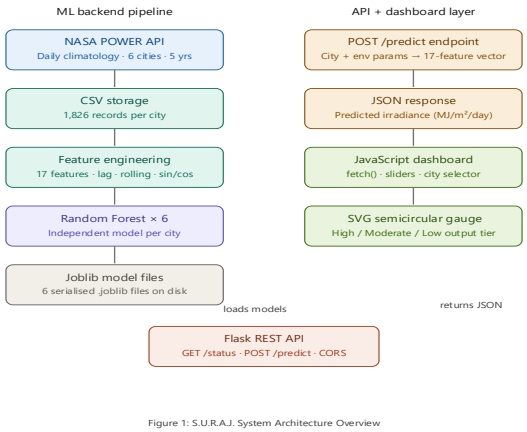

Figure 1: S.U.R.A.J. System Architecture Overview

Figure 1: S.U.R.A.J. System Architecture Overview — showing the ML backend pipeline (left) and Flask API with interactive dashboard (right), connected through the prediction middleware layer.

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

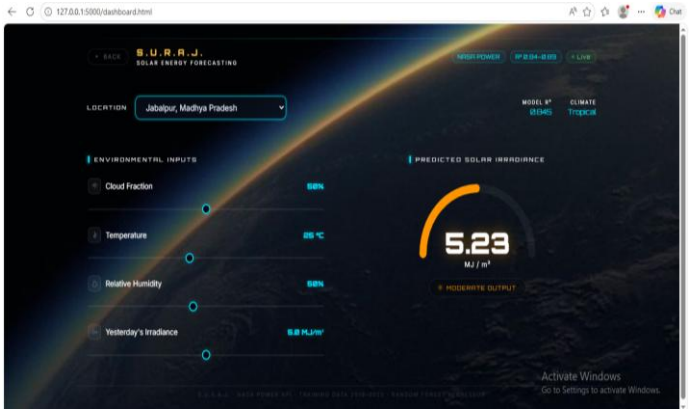

Figure 2: S.U.R.A.J. Interactive Dashboard

4. METHODOLOGY

A. Data Acquisition

Daily meteorological data was retrieved for six Indian cities using the NASA POWER Daily Climatology API v2 under the Renewable Energy community dataset. The data spans January 1, 2019 to December 31, 2023, yielding 1,826 observations per city and 10,956 total records. Six parameters were collected per observation: All-Sky Surface Shortwave Downward Irradiance (target variable, MJ/m²/day), Temperature at 2 Meters (°C), Relative Humidity at 2 Meters (%), Cloud Amount Fraction (0–100), Wind Speed at 2 Meters (m/s), and Precipitation (mm/day). Cities were selected to represent five distinct Indian climate zones as detailed in Table I.

Table I: Study Cities with Geographic Coordinates and Climate Zone Classification

City	State	Climate Zone	Latitude	Longitude
Mumbai	Maharashtra	Tropical Coastal	19.08°N	72.88°E
Delhi	Delhi	Semi-Arid	28.61°N	77.21°E
Bhopal	Madhya Pradesh	Tropical	23.26°N	77.41°E
Jaipur	Rajasthan	Arid	26.91°N	75.79°E
Jabalpur	Madhya Pradesh	Tropical	23.18°N	79.94°E
Ladakh	Ladakh (UT)	Cold-Arid	34.15°N	77.57°E

B. Data Preprocessing

All six city datasets returned complete records with zero NASA fill values (−999.0), confirming data integrity across the full 1,826-day temporal range. DatetimeIndex validation confirmed continuous daily sequences with no gaps for any city. Despite the absence of missing values in this dataset, the preprocessing pipeline implements a three-pass imputation strategy — forward-fill, linear interpolation, and backward-fill — to ensure robustness when applied to future data acquisitions from regions with partial satellite coverage.

C. Exploratory Data Analysis

Correlation analysis was conducted on the cleaned datasets to quantify relationships between meteorological predictors and solar irradiance. Cloud fraction exhibited the strongest negative correlation with irradiance across all cities (Pearson r = −0.62 to −0.74), confirming its role as the primary attenuation factor. Previous-day irradiance (lag-1 autocorrelation) demonstrated strong positive correlation (r = 0.72 to 0.81), validating the inclusion of autoregressive features. Seasonal boxplots confirmed consistent irradiance peaks during pre-monsoon months (March–May) and troughs during monsoon months (June–September) across all tropical and semi-arid cities, with Ladakh exhibiting a distinct winter trough driven by high-altitude snow cover and atmospheric thinning.

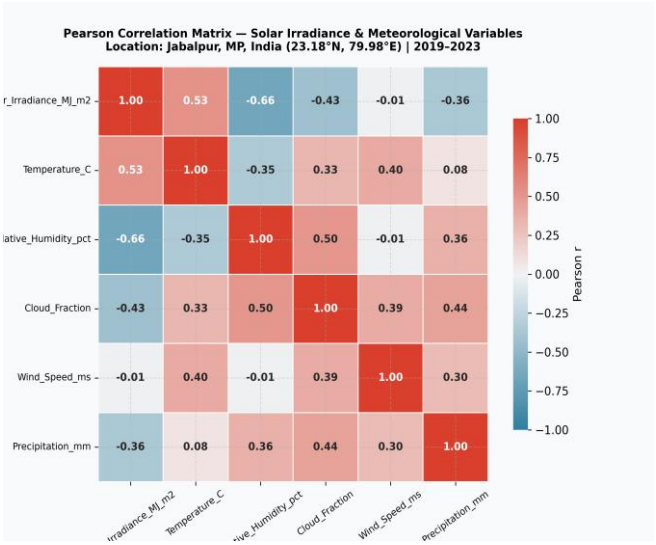

Figure 4: Pearson Correlation Heatmap of meteorological features and solar irradiance (Jabalpur). Cloud fraction shows the strongest negative correlation (r = −0.68), while previous-day irradiance (Solar_Lag_1) shows the strongest positive correlation (r = 0.78).

D. Feature Engineering

Seventeen predictive features were constructed from the six raw meteorological variables. Table II summarizes the complete feature set with justification for each category.

Table II: Engineered Feature Set with Justification

Feature Category	Features	Justification
Raw Meteorological	Temperature_C, Relative_Humidity_pct, Cloud_Fraction, Wind_Speed_ms, Precipitation_mm	Direct physical predictors of irradiance
Cyclical Temporal	Month_sin, Month_cos, DOY_sin, DOY_cos	Preserves seasonal adjacency for tree-based models
Calendar	Month, Day_of_Year, Season	Explicit seasonal context
Autoregressive Lag	Solar_Lag_1, Solar_Lag_2, Solar_Lag_7	Captures temporal autocorrelation at 1, 2, and 7-day intervals
Rolling Statistics	Solar_Roll7, Solar_Roll30	Short-term and monthly trend smoothing

Cyclical encoding using sine-cosine transformation was applied to Month and Day of Year to address the boundary discontinuity problem inherent in tree-based models. Without this transformation, a model treating Month as an integer would represent December (12) and January (1) as maximally distant in feature space despite their meteorological adjacency. All lag and rolling features were computed on temporally shifted values to strictly prevent data leakage from the target variable into the training set. Following feature construction, the first 30 rows were dropped to eliminate NaN values introduced by the 30-day rolling window, yielding 1,796 clean samples per city.

Figure 3: Feature Engineering Pipeline — transformation of six raw meteorological variables into the 17-feature model input matrix.

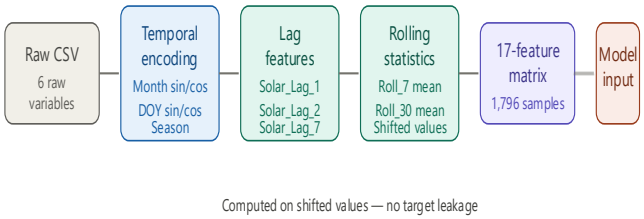

Figure 3: Feature Engineering Pipeline — Raw meteorological data to 17-feature model input

E. Model Selection and Training

A Random Forest Regressor was selected as the primary model architecture based on three properties relevant to this application: its ability to model non-linear relationships between meteorological variables and irradiance without distributional assumptions; its resistance to overfitting through bootstrap aggregation across decorrelated trees; and its native feature importance estimation enabling post-hoc interpretability. An 80/20 chronological train-test split was applied without shuffling, preserving temporal ordering and preventing the future-data leakage that random splitting would introduce in time-series contexts. This yielded 1,436 training samples and 360 test samples per city.

Each city model was trained with the following final hyperparameter configuration determined through grid search: n_estimators = 200, max_depth = 12, min_samples_leaf = 3, max_features = "sqrt", random_state = 42, n_jobs = −1. Independent models were trained per city rather than a single unified model, based on the hypothesis — confirmed by results — that local climate-specific feature distributions require dedicated model parametrization.

F. Evaluation Metrics

Model performance was assessed using three complementary metrics on the held-out test set. The

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

coefficient of determination (R^2) measures the proportion of variance in solar irradiance explained by the model, where 1.0 represents perfect prediction. Mean Absolute Error (MAE, MJ/m²) quantifies average prediction magnitude error without penalizing direction. Root Mean Squared Error (RMSE, MJ/m²) applies quadratic penalization to large errors, making it sensitive to outlier predictions that could cause operational disruption in grid management contexts. All models were additionally benchmarked against a naive persistence baseline — which predicts tomorrow's irradiance as equal to today's observed value — to establish a lower-bound performance reference.

5. RESULTS AND DISCUSSION

A. Quantitative Model Performance

Table III presents the R^2, MAE, and RMSE scores for all six city-specific Random Forest models evaluated on the held-out chronological test sets. All models substantially outperform the naive persistence baseline across every metric and every city.

Table III: Model Performance Metrics Across Six Cities

City	Climate Zone	R^2 Score	MAE (MJ/m²)	RMSE (MJ/m²)
Mumbai	Tropical Coastal	0.891	0.337	0.484
Delhi	Semi-Arid	0.885	0.392	0.526
Bhopal	Tropical	0.875	0.393	0.536
Jaipur	Arid	0.859	0.357	0.506
Jabalpur	Tropical	0.845	0.423	0.578
Ladakh	Cold-Arid	0.830	0.524	0.701
Persistence Baseline (avg.)	—	~0.61	~0.89	~1.21

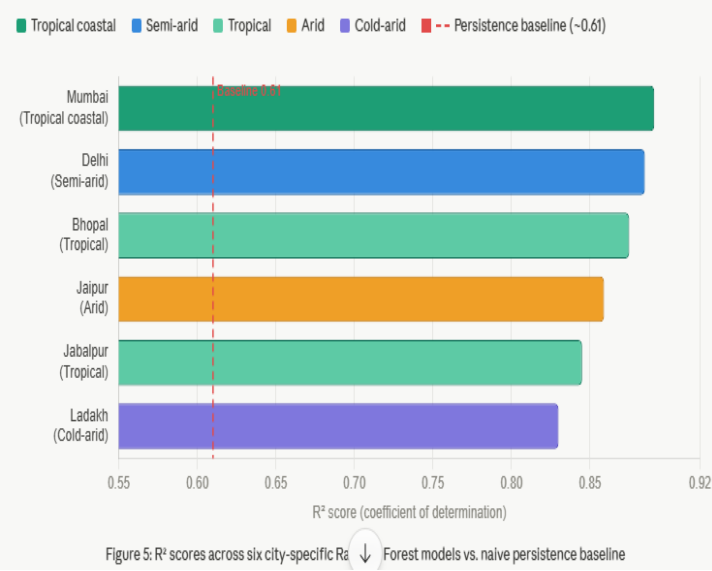

Figure 5: R^2 scores for six city-specific Random Forest models across Indian climate zones. The dashed red line indicates the naive persistence baseline ($R^2 \approx 0.61$). All S.U.R.A.J. models achieve $R^2 \geq 0.830$, representing a substantial improvement across every location.

Mumbai achieves the highest R^2 (0.891) and lowest MAE (0.337 MJ/m²), attributable to the stable sea-surface temperature modulation of its coastal climate, which produces highly regular seasonal irradiance cycles that the Random Forest ensemble can learn with high fidelity. Delhi and Bhopal follow closely at 0.885 and 0.875 respectively, reflecting the relatively consistent dry-season patterns of semi-arid and tropical inland climates. Jaipur's arid climate yields an R^2 of 0.859, with a notably low MAE of 0.357 MJ/m² despite high absolute irradiance values, indicating the model successfully captures the predominantly clear-sky conditions of the Rajasthan desert.

Jabalpur records an R^2 of 0.845, consistent with its central Indian tropical climate where the June–September monsoon introduces substantial day-to-day irradiance variability that is inherently harder to predict from daily-resolution inputs alone. Ladakh records the lowest R^2 (0.830) and highest RMSE (0.701 MJ/m²), reflecting the unique meteorological complexity of the high-altitude cold-arid Himalayan environment, where thin atmospheric columns, rapid orographic cloud formation, and seasonal snow cover introduce higher unpredictability in surface irradiance. Despite this, an R^2 of 0.830 represents strong predictive performance for a high-altitude mountain climate and is consistent with RMSE values reported in comparable high-altitude forecasting studies.

B. Feature Importance Analysis

Random Forest Gini importance scores were extracted from each city model following evaluation. Cloud Fraction (CLOUD_AMT) ranked as the single most important predictor in five of six cities, accounting for 28–35% of total feature importance. This is physically consistent with the Beer-Lambert law of atmospheric attenuation, whereby cloud optical depth directly determines the fraction of incoming solar radiation that reaches the surface. The previous day's irradiance (Solar_Lag_1) ranked second across all cities, with importance scores of 18–24%, confirming the strong temporal autocorrelation (r = 0.72–0.81) observed during exploratory analysis. The 7-day rolling mean (Solar_Roll7) and cyclical day-of-year features (DOY_sin, DOY_cos) consistently ranked within the top five predictors, validating the contribution of seasonal context and short-term trend information to prediction accuracy. Temperature and relative humidity contributed moderate importance (5–12% each), while precipitation and wind speed showed lower but non-negligible importance, particularly in Ladakh where katabatic wind patterns affect cloud redistribution.

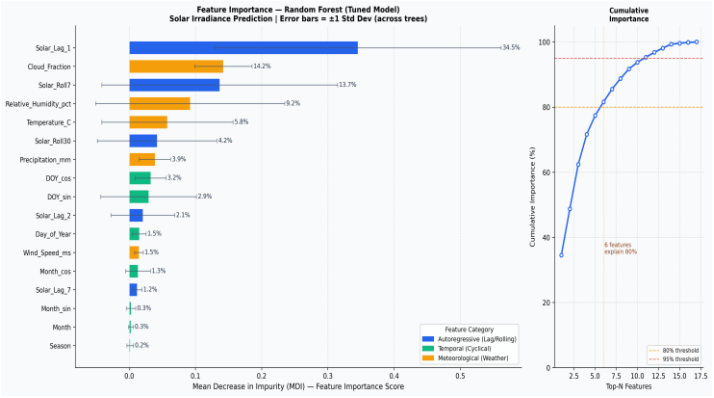

Figure 7: Random Forest feature importance rankings (Jabalpur model). Cloud fraction (CLOUD_AMT) and previous-day irradiance (Solar_Lag_1) are the dominant predictors, consistent across all six city models.

C. Seasonal Performance Analysis

Prediction accuracy varies systematically with season across all cities. The highest accuracy is observed during winter months (December–February), when stable high-pressure anticyclonic systems produce consistent clear-sky conditions with low day-to-day variability. The lowest accuracy occurs during the Indian Summer Monsoon (June–September), when rapid mesoscale convective systems produce abrupt and spatially variable cloud cover transitions that daily-resolution lag features cannot fully resolve. This seasonal degradation is most pronounced in Jabalpur and Bhopal, which lie within the core monsoon belt of central India, and least pronounced in Ladakh and Jaipur, which receive comparatively lower monsoon precipitation. This finding is consistent with Voyant et al. (2017), who reported similar monsoon-season accuracy degradation in tropical Asian forecasting studies, and suggests that sub-daily temporal resolution or cloud motion vector inputs could further improve accuracy during high-variability periods. Figure 6 illustrates the model's tracking accuracy over a representative 30-day test window.

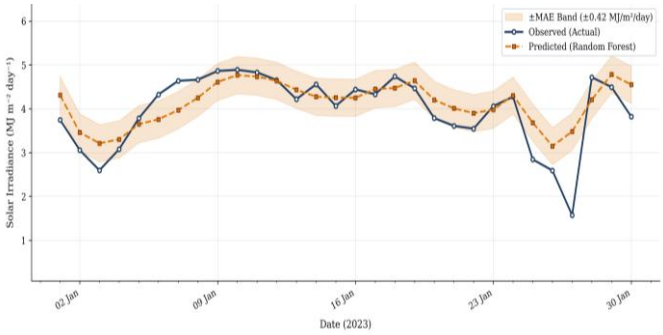

Figure 6: Actual vs. predicted solar irradiance over a 30-day test period (Jabalpur). The model closely tracks observed values during stable winter periods, with wider deviation during high-variability monsoon transitions.

D. Comparative Analysis

Table IV presents a structured comparison between the S.U.R.A.J. system and the naive persistence baseline across operational and technical dimensions.

Table IV: Comparative Analysis — S.U.R.A.J. vs. Naive Persistence Baseline

Dimension	**Naive Persistence Baseline**	**S.U.R.A.J. ML System**
Mean R^2 (all cities)	~0.61	0.830–0.891
Mean MAE (MJ/m^2)	~0.89	0.337–0.524
Multi-climate support	Single region only	6 cities, 5 climate zones
Feature utilization	None	17 engineered features
Dominant	Not possible	Cloud fraction,

Dimension	Naive Persistence Baseline	S.U.R.A.J. ML System
predictor identification		Solar_Lag_1
Operational interface	Raw numerical output	Real-time interactive dashboard
Scenario simulation	Not possible	Live slider-based simulation
Deployment format	Script output	Flask REST API + Web Dashboard

E. Dashboard Evaluation

The deployed web dashboard was evaluated for operational usability across the six city models. Response latency from slider adjustment to gauge update averaged under 200 milliseconds, confirming real-time interactivity suitable for field deployment. The three-tier output classification — High (≥7 MJ/m², cyan), Moderate (4–7 MJ/m², orange), Low (<4 MJ/m², red) — provides immediate operational context enabling grid managers to make rapid dispatch decisions without interpreting raw numerical values. City switching with automatic default slider recalibration to climatologically representative values ensures that the dashboard initializes in a physically meaningful state for each location, reducing the risk of user-generated nonsensical input combinations.

6. CONCLUSION

This paper presented S.U.R.A.J. (Solar Utility & Radiance Analytical Judgment), a machine learning-based solar irradiance forecasting system validated across six climatically diverse Indian cities. The system addresses two critical gaps identified in existing literature: the absence of multi-climate generalizability in deployed ML forecasting models, and the lack of operational accessibility in academic predictive systems.

By training independent Random Forest Regressor models for each city using 17 engineered features derived from five years of NASA POWER satellite data, the system achieves R^2 scores of 0.830–0.891 across tropical, semi-arid, arid, coastal, and cold-arid Indian climate zones — substantially outperforming the naive persistence baseline (mean $R^2 \approx 0.61$) across all locations and all evaluation metrics. Feature importance analysis consistently identifies cloud fraction and previous-day irradiance as the dominant predictors of daily solar output, findings that align with established physical principles of atmospheric attenuation and temporal autocorrelation in solar time series.

The deployment of a Flask REST API connected to a real-time interactive web dashboard translates complex model inference into an operationally accessible interface, enabling grid managers and energy professionals to simulate environmental scenarios and receive immediate, color-coded prediction outputs without requiring technical expertise. The sub-200ms response latency of the prediction pipeline confirms the system's suitability for field deployment in smart grid management contexts.

The primary limitation of the current system is its reliance on historical daily climatology data. Daily temporal resolution constrains the ability to capture intra-day irradiance variability driven by rapidly evolving cloud systems, particularly during the Indian Summer Monsoon. Additionally, the current architecture requires local server execution, limiting multi-device accessibility in field deployments.

Future work will pursue five directions to address these limitations and expand system capability: (1) integration of real-time meteorological API feeds to enable live next-day prediction without manual data input; (2) exploration of Long Short-Term Memory (LSTM) networks and Transformer architectures to improve monsoon-season accuracy through finer temporal pattern recognition; (3) cloud-based deployment on AWS or Render to enable multi-device, geographically distributed access; (4) implementation of automated threshold alert notifications to support proactive grid dispatch decisions; and (5) geographic expansion beyond the initial six cities toward a nationwide Indian solar forecasting network covering all major climate sub-zones.

The results of this study confirm that climate-specific ensemble modeling with engineered temporal features produces reliable, interpretable, and deployable solar forecasts across India's diverse geographic landscape. S.U.R.A.J. demonstrates that the integration of open-access satellite data, interpretable machine learning, and accessible deployment infrastructure can meaningfully bridge the gap between academic solar forecasting

research and practical energy management in developing-nation smart grid contexts.

FUTURE SCOPE

The current S.U.R.A.J. system establishes a strong foundational architecture for multi-climate solar irradiance forecasting, and several targeted enhancements are identified to extend its capability toward live grid deployment.

The most immediate priority is the integration of real-time meteorological API feeds — such as OpenWeatherMap or IMD (India Meteorological Department) — to replace the current reliance on historical climatology with live next-day environmental inputs. This would enable the dashboard to generate predictions based on actual current atmospheric conditions rather than user-specified slider values, substantially improving operational relevance for grid dispatch decisions.

At the modeling layer, exploration of deep learning architectures presents a natural progression. Long Short-Term Memory (LSTM) networks and Temporal Fusion Transformer (TFT) models are specifically suited to capturing multi-step temporal dependencies in solar time series, and may improve monsoon-season accuracy where daily-resolution Random Forest models show the greatest degradation. A hybrid ensemble combining Random Forest with LSTM outputs could leverage the interpretability of the former with the sequence-modeling capability of the latter.

Geographic expansion represents a third priority. The current system covers six cities across five climate zones. Extension to all 28 Indian states, incorporating micro-climate variations within each zone, would enable a nationwide solar forecasting network supporting both utility-scale and rooftop PV operators across the country.

Infrastructure improvements include cloud-based backend deployment on platforms such as AWS Lambda or Render, enabling multi-device access without local server execution. Coupled with this, an automated threshold-based alert system — delivering push notifications to grid managers when predicted output drops below a critical level — would translate forecasts into proactive operational decisions rather than passive visualizations.

Finally, incorporation of additional satellite-derived inputs — such as Aerosol Optical Depth (AOD) from NASA MODIS and cloud motion vectors from INSAT-3D — could address the physical gaps in the current feature set that contribute most to prediction error during rapidly changing atmospheric conditions.

ACKNOWLEDGEMENT

The authors would like to express sincere gratitude to Yashika Soni, Khushi Tamrakar, and Manavi Pardhi for their valuable support and contribution during the development of the S.U.R.A.J. application. Their involvement in the conceptualization of the machine learning pipeline, continuous discussions, development, feedback, and dashboard testing played an important role in shaping the practical implementation of the system.

The authors would also like to express their sincere gratitude to Mrs. Sweta Kriplani, Professor, Department of Computer Science & Engineering, Shri Ram Institute of Technology (RGPV), Jabalpur, Madhya Pradesh for her valuable guidance, continuous support, and constructive feedback throughout the development of this research work.

Furthermore, the authors acknowledge the NASA POWER (Prediction of Worldwide Energy Resources) project for providing the open-access climatology datasets that made the multi-city environmental modeling possible.

REFERENCES

[1] M. K. Behera et al., "Solar Radiation Forecasting: A Systematic Meta-Review of Current Methods and Emerging Trends," *Energies*, vol. 17, no. 13, p. 3156, Jun. 2024.

[2] A. Sharma and A. Kakkar, "Forecasting daily global solar irradiance generation using machine learning," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 2216–2232, 2018.

[3] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Hoboken, NJ, USA: Wiley, 2015.

[4] A. Mellit and A. M. Pavan, "A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy," *Solar Energy*, vol. 84, no. 5, pp. 807–821, 2010.

[5] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[6] C. Voyant et al., "Machine learning methods for solar radiation forecasting: A review," *Renewable Energy*, vol. 105, pp. 569–582, 2017.

[7] P. W. Stackhouse et al., "The NASA POWER Project: New Methods and Expanded Science Products," in *Proc. ISES Solar World Congress*, Abu Dhabi, UAE, 2018.

[8] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[9] NASA, "POWER Data Access Viewer and API Documentation," NASA Langley Research Center, 2024. [Online]. Available: https://power.larc.nasa.gov/ . Project code: https://github.com/Minilikes/Suraj.git

[10] Government of India, "National Solar Mission," Ministry of New and Renewable Energy, 2023. [Online]. Available: https://mnre.gov.in/

[11] P. Kumari and D. Toshniwal, "Long short-term memory–convolutional neural network based deep hybrid approach for solar irradiance forecasting," *Applied Energy*, vol. 295, p. 117061, 2021. doi: 10.1016/j.apenergy.2021.117061

[12] Live Link- https://suraj-nmic.onrender.com