Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

Prediction of Solar Power Generation and Ground Area Estimation Using KNN Regression

Dr. Preeti. S. Voditel[1] , Prof. Aparna. M. Gurjar[2], Mr. Priyadarshan V. Gahukar[3]
[1]Department of Computer Science and Emerging Technologies, Earlier, Shri Ramdeobaba College of Engineering and Management Nagpur, India[(a)]
[2,3]Department of Computer Applications, Ramdeobaba University, Earlier, Shri Ramdeobaba College of Engineering and Management Nagpur, India
voditelps2@rknec.edu[1], gurjara@rknec.edu[2], gahukarpv@rknec.edu[3]

ARTICLE INFO

Received: 26 Dec 2024

Revised: 14 Feb 2025

Accepted: 22 Feb 2025

ABSTRACT

This paper focuses on making use of historical weather data to predict the potential power generation through the application of the k-Nearest Neighbours (k-NN) algorithm and subsequent estimation of the precise area needed to accommodate the solar panels. Effect of variables such as ambient temperature, solar irradiation, solar panel temperature, latitude and longitude, topology, and cloud coverage is studied on the solar power generation capacity. The model is trained to forecast the power output of solar panels based on three environmental variables of ambient temperature, solar irradiation and solar panel temperature. The k-NN algorithm, known for its simplicity and efficiency, is employed to capture the patterns inherent in historical datasets. This methodology allows for the accurate prediction of solar power generation under diverse weather conditions. The incorporation of temperature data, both ambient and solar panel-specific, enhances the precision of the model. The research outcomes not only contribute to the advancement of solar energy forecasting but also help optimize the energy grid management using the k-NN methodology.

Keywords: solar power, k-nearest neighbours, KNN, machine learning, weather data, power prediction, renewable energy

I. INTRODUCTION

It was pledged at the 28th Conference of the Parties to the UN Framework Convention on Climate Change COP28 to end the era of fossil fuel gradually and committed itself to tripling renewables capacity as well as doubling energy efficiency by the year 2030 [1]. India, as a participating nation in COP28, has embarked upon the mission of making the earth greener, cleaner and healthier through the target of achieving net zero carbon emission by the year 2070. This involves shifting its heavy reliance from coal to a more sustainable and renewable energy source like solar power. Almost half of India lies in the tropical region and therefore enjoys abundant sunlight throughout the year. As per Government statistics, on an average, most regions receive 4-7 kWh per sqm per day of solar energy [2].

Solar energy offers several advantages like inexhaustibility, low operating costs, and minimal greenhouse gas emissions and can therefore contribute a big chunk towards the renewable energy generation capacity of the nation. However, assessment of the feasibility of deploying solar power plants in certain areas remains a challenge as the solar energy is dependent on environmental factors which are constantly fluctuating across regions and seasons. In [3], the authors quote that due to the changes in the meteorological factors, the solar radiation varies and hence needs to be accurately forecasted for ensuring electricity grid stability.

The instantaneous power output of the solar panels depends on the solar radiation value. This value depends on many environmental factors among which cloud cover over a region considerably affects the power output. In [4] the authors forecast the solar radiation values, taking cloud motion in consideration, over a certain time period using image processing and Long-Short Term Memory (LSTM). In [12] the authors discuss the challenges faced by the power grids due to integration of photovoltaic based power generation systems in those grids. To prevent volatility in the system it is imperative that short term forecasts of power outputs of renewable generation systems should be undertaken. They propose Traditional Encoder Single Deep Learning (TESDL) method for forecasting weather using deep learning methods.

Therefore, before the actual deployment of any project begins, forecasting of many parameters needs to be undertaken based on data. Artificial Intelligence based algorithms are able to mine the historical data and help predict parameters of interest.

Deciding whether to deploy a solar power plant in a specific area is a crucial decision. This paper seeks to address the challenge of assessing the viability of a solar power plant by developing a prediction model. It helps predict the potential solar power which can be generated using the K-Nearest Neighbors (k-NN) algorithm and also allows the estimation of the precise area needed to accommodate solar panels for the predicted power generation.

By leveraging historical weather data, we train the model to forecast solar panel output. Key environmental variables considered include ambient temperature, solar irradiation, and solar panel temperature. The k-NN algorithm identifies patterns in historical datasets, that enables accurate predictions under diverse weather conditions. Incorporating both ambient and solar panel-specific temperature data enhances model precision.

The predicted outcomes will contribute not only to solar energy forecasting but also in optimizing energy grid management. Decision-makers can use it to assess the viability of solar power plant deployments in specific locations, promoting informed and sustainable energy planning.

II. LITERATURE REVIEW

We now proceed to analyze the k-NN regressor as a reliable predictor. The k-NN regressor is a versatile algorithm commonly employed for regression tasks due to its simplicity and effectiveness. Unlike traditional parametric models, k-NN regressors do not assume any specific underlying distribution of the data. Instead, they rely on the principle that similar data points should have similar target values. Thus, 'feature similarity' is used to predict the values of new data points [5].

The k-NN regressor operates by computing the distances between the input feature vectors representing these factors and the historical data points stored in memory. The "k" closest historical data points (Nearest Neighbors) are selected, and their target values are used to predict the output. These distances can be computed in variety of ways. It can be done either by finding the average of k-nearest neighbour target, or calculating weighted average of k-closest neighbours based on inverse of their distances [5]. Selecting the value of k^{th} nearest neighbours is also challenging. Sometimes plain heuristic value is used. At other times the value is calculated using some optimization tool [6][9].

The k-NN approach is particularly useful for regression tasks where the underlying relationship between input features and target values may be nonlinear or complex. Additionally regression using k-NN can adapt to changing data patterns and does not require model retraining when new data becomes available thus making it suitable for dynamic environments.

In summary, the k-NN regressor is an effective tool for various regression tasks, leveraging the similarity of input features to make predictions. Its simplicity, flexibility, and ability to handle nonlinearity make it a valuable choice in diverse domains ranging from real estate to healthcare and beyond.

k-NN is also a suitable choice for solar power prediction using weather data due to several advantages. Firstly, its simplicity makes it straightforward to implement and understand. k-NN is non-parametric, meaning it doesn't assume any specific data distribution, thus making it flexible for various scenarios. Moreover, it exhibits adaptability, which is crucial for solar power prediction as it can effectively adjust to changing weather conditions.

It takes into consideration the local context by taking into account nearby data points. k-NN aligns well with the localized impact of weather on solar energy production. This localized approach allows for more accurate predictions based on the specific weather conditions in the vicinity of the solar power generation site. We now discuss various parameters which affect the k-NN regressor.

A. ***Distance Metric Selection***

In the architecture of a k-NN regressor, the choice of distance metric holds paramount importance. The distance metric determines the notion of similarity or dissimilarity between data points in the feature space. Euclidean distance, Manhattan distance, and Minkowski distance [5] are generally the distance metrics used. The selection of an appropriate distance metric hinges on the data characteristics and the specific requirements of the regression task at hand. A suitable distance metric should effectively capture the underlying relationships between data points, facilitating accurate predictions. Mostly Euclidean distance is used.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

where,

d = Euclidean distance

where (p, q) are two points in Euclidean n-space

B. ***DTW (Dynamic Time Warping)***

When dealing with time series data like solar irradiance, which exhibits a daily parabolic trend with potential weather fluctuations, choosing the right distance metric for KNN becomes crucial. Dynamic Time Warping (DTW) emerges as a compelling option due to its ability to handle time-dependent variations. In [11], the authors apply the DTW technique to the time series solar winds speed prediction model. The solar irradiance data also exhibits time series properties and hence can be evaluated using DTW technique. Unlike Euclidean distance, which focuses solely on the straight-line distance between points, DTW can "warp" the time axis to find the best alignment between two sequences. This makes it particularly adept at capturing the underlying parabolic trend in solar irradiance data, even if there are slight time shifts due to cloud cover or other weather events. Additionally, DTW's robustness to noise proves beneficial when dealing with outliers or fluctuations caused by weather variations, potentially leading to more accurate KNN predictions compared to Euclidean distance. This paper however uses the default “Minkowski” distance available in the “sklearn” library. It is a distance measure which is a generalization of the Euclidean and the Manhattan distance. This metric can be used to measure the similarity or dissimilarity between two series which exhibit time series properties.

C. ***Hyperparameter***

At the heart of the k-NN regression architecture lies a crucial hyperparameter: 'k'. This hyperparameter is used in deciding the number of nearest neighbors to consider when making predictions. The choice of 'k' significantly influences the model's flexibility and generalization ability. Its predictive performance is mainly decided by the value of parameter k [7]. Smaller values of 'k' tend to result in more flexible models, potentially capturing fine-grained data patterns. However, such models may be susceptible to noise and outliers [8]. Conversely, larger values of 'k' lead to smoother predictions but may overlook local variations in the data. Selecting an optimal value for 'k' is a critical aspect of building an effective k-NN regressor.

D. ***Training Process***

Unlike traditional model training processes, k-NN regression does not involve explicit model parameter optimization. During the training phase, certain function mapping the input to the output is not learnt; rather it simply memorizes the training data. It exhibits the characteristics of lazy learning [9]. This entails storing the feature vectors and corresponding target values (labels) of the training set in memory. As such, the training process in k-NN regression primarily revolves around data preprocessing and storage. Since k-NN regression does not learn explicit model parameters, the training phase is relatively lightweight and computationally inexpensive compared to many other machine learning algorithms. Steps involved in Training process is given in flow chart of Fig-1.

A brief explanation of the steps follows:

- Selecting the Value of “k” –

The training process begins with the selection of the value of K, which determines the nearest neighbors to be considered when making predictions. This hyperparameter is typically chosen based on domain knowledge, cross-validation, or other optimization techniques

- Considering All Points in an n-Dimensional Space –
  Each data point in the training dataset is represented as a vector in an n-dimensional space, where "n" is the number of features or dimensions in the dataset.
- Calculating the Distance of New Point from All Points –
  For each new data point or sample in the dataset, the algorithm calculates its distance from all other points in the dataset. Euclidean distance, Manhattan distance, or Minkowski distance can be typically used to compute this distance.
- Sorting the Distance of All Points -
  After calculating the distances of the new point from all other points in the dataset, the distances are sorted in ascending order
- Selecting the K Nearest Neighbors –
  Once the distances are sorted, the algorithm selects the K nearest neighbors to the new point based on their distances. These neighbors are the data points with the smallest distances from the new point.
- Estimating the Value of the Test Point –
  After selecting the K nearest neighbors, the algorithm estimates the value of the test point (or the new data point) by computing the weighted average of the target values (or labels) of its nearest neighbors. The weights assigned to each neighbor can be based on various factors such as distance or similarity
- Evaluating the Error of the Test Point –
  Finally, the algorithm evaluates whether the error of the test point satisfies certain criteria. Depending on the application and specific requirements, this evaluation may involve comparing the predicted value with the actual target value and determining if the error falls within an acceptable range.
- Iterating or Finishing the Process –
  Depending on the training strategy and convergence criteria, the training process may involve iterating through the dataset multiple times or until certain convergence criteria are met. Once the training process is complete, the K-NN regressor is ready to make predictions on new, unseen data.

E. ***Prediction Process***

The prediction process in a K-NN regressor involves several key steps. Given a new input sample for which a prediction is sought, the algorithm computes the distances between this sample and all the training samples in the dataset. The chosen distance metric governs the computation of these distances. Subsequently, the algorithm identifies the 'K' nearest neighbors based on these distances. For regression tasks, the predicted value for the new input sample is typically determined as the average (or weighted average) of the target values of its 'K' nearest neighbors. This process is influenced by the notion that similar data points should exhibit similar target values.

F. ***Memory Usage and Computational Complexity***

A notable characteristic of k-NN regression is its memory-intensive nature. The algorithm necessitates storing the entire training dataset in memory, which can pose scalability challenges for large datasets. Furthermore, the prediction time complexity for each new sample is relatively high. It involves computing distances between the new sample and all training samples, resulting in a time complexity of O (n * d), where 'n' represents the number of training samples and 'd' denotes the dimensionality of the feature space. As such, while K-NN regression offers simplicity and ease of implementation, its computational demands warrant careful consideration, particularly for large-scale datasets

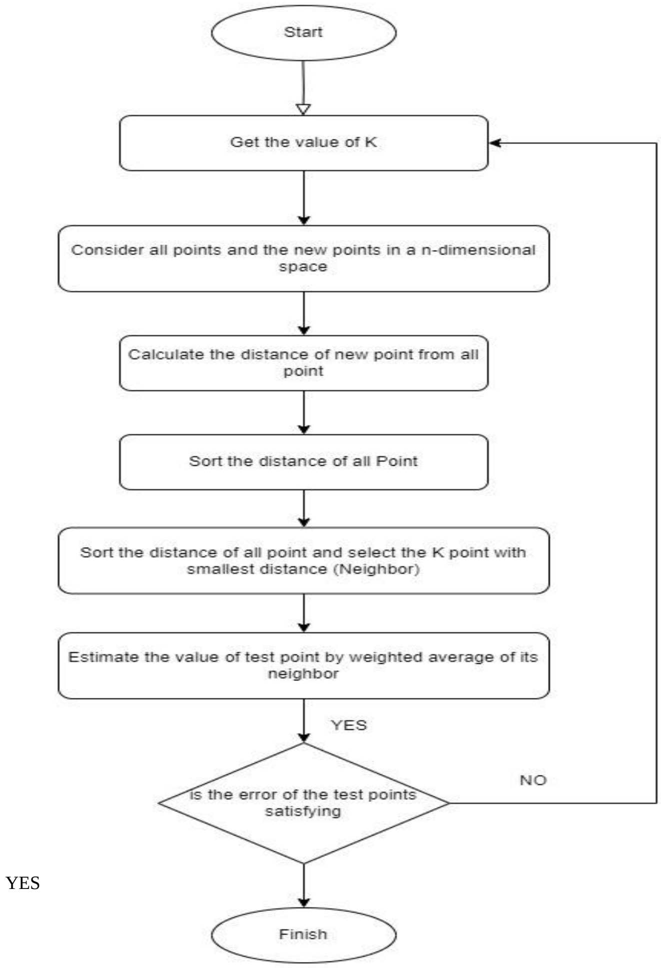

Fig-1

G. ***Model Limitations and Comparison***

Despite its simplicity and intuitive nature, k-NN regression is not without limitations. One notable limitation is its sensitivity to the choice of distance metric. Distance based algorithms like k-NN perform differently with homogeneous and heterogeneous data. In case of heterogeneous data, the Euclidean distance method is slightly less effective [10]. The performance of the algorithm can vary significantly depending on the selected distance metric, emphasizing the importance of careful hyperparameter tuning. Additionally, k-NN regression is susceptible to the curse of dimensionality, whereby the predictive performance deteriorates as the number of features (dimensions) increases. This phenomenon can lead to sparse data and increased computational complexity, posing challenges for high-dimensional datasets. Due to these limitations k-NN regression needs to be adapted carefully in practical scenarios.

However, the flexibility of KNN in capturing non-linear relationships makes it particularly advantageous in scenarios where the relationship between variables deviates significantly from linearity.

Other techniques like Linear Regression offers simplicity and interpretability, its rigid linearity assumption may limit its effectiveness in accurately modeling the parabolic behavior of solar irradiation. Another one like Random Forest Regression can handle non-linear relationships and high-dimensional data well, but might struggle to capture the intricate nuances of the parabolic pattern, especially if the relationship is highly non-linear.

In summary, for solar power prediction where solar irradiation follows a parabolic pattern, KNN emerges as the preferred choice over Linear Regression or Random Forest Regression. By leveraging the similarities among neighboring data points, KNN can better capture the complexities of solar irradiation data, potentially leading to more precise forecasts of power generation.

III. METHODOLOGY

A. Pre-requities

This project, requires diverse methodologies and tools like supervised learning, Python programming, navigation of Kaggle notebooks, and utilization of the scikit-learn library.
Scikit-learn is a powerful open-source library designed to facilitate machine learning in Python. We use Pandas which is an open-source Python library, for data manipulation, analysis, and CSV file handling. NumPy, a fundamental library in Python is needed for numerical computing. Matplotlib, is a Python library for data visualization and graphical representations of data.

B. DataSet used for Training

The dataset (history data of year 2020) utilized in this paper comprises solar power generation and environmental sensor data collected from two solar power plants situated in India over a 34-day period. It consists of paired files, each containing power generation and sensor readings datasets. The power generation dataset (Dataset 1) includes critical fields such as SOURCE_KEY, DC_POWER, DATE_TIME, AC_POWER, DAILY_YIELD, and TOTAL_YIELD, recorded at 15-minute intervals totaling over 70,000 rows. This dataset offers insights into key metrics of solar power generation, including inverter-level power output and cumulative yield. The environmental sensor data (Dataset 2) encompasses fields such as DATE_TIME, AMBIENT_TEMPERATURE, MODULE_TEMPERATURE, and IRRADIATION, also recorded at 15-minute intervals comprising approximately 4,600 rows. The SOURCE_KEY field identifies data from 15 different solar panels, providing information on the specific source of each observation. Together, these datasets provide information on environmental conditions impacting solar power generation efficiency, including ambient and module temperatures, as well as solar irradiation.

C. DataSet used for Prediction

The dataset utilized for prediction in this paper consists of weather recordings from a city in Rajasthan (Jodhpur), spanning the entirety of the year 2022. The dataset provides a comprehensive record of various weather parameters such as temperature, humidity, precipitation, wind speed, and solar radiation, among others, allowing for detailed analysis and prediction of weather patterns over the specified timeframe. The hourly resolution of the data enables a granular examination of time related changes in weather conditions, offering valuable insights for forecasting and predictive modeling purposes.

D. Pre-processing the datasets

Dataset preprocessing was performed on two primary datasets: "Power_Generation_Data" and "Weather_Sensor_Data." These datasets, which respectively encapsulate information regarding power generation and weather-related sensor readings, were subjected to preprocessing steps for ensuring data integrity and compatibility.
There was the presence of negative values in fields such as irradiation and temperature, which typically occur when data is missing. To handle these negative values, a method involving the computation of the mean of upper and lower data ranges was employed. By replacing negative values with these representative measures, the datasets were rendered more consistent and suitable for analysis.
Furthermore, to facilitate integration and alignment of the two datasets, a datetime model was constructed. This model enabled the effective merging of temporal data from both datasets, thereby enhancing data organization and facilitating analyses across different timeframes.
Additionally, columns were pruned by identifying and subsequently dropping the irrelevant or redundant columns. Through these preprocessing steps, the "Power_Generation_Data" and "Weather_Sensor_Data" datasets were refined.
Similar preprocessing steps were also applied to the Rajasthan state dataset to ensure consistency. Preprocessing was primarily focused on the "Power_Generation_Data" and "Weather_Sensor_Data" datasets to align them with the hourly resolution of the Rajasthan prediction data. Aggregating data from these datasets to match the hourly resolution of the prediction dataset was essential for making the data across different timeframes consistent. This process involved computing the mean or average of relevant parameters over each hourly interval, ensuring consistency and compatibility for subsequent analysis.

E. Traning Prediction model:

Before training the predictive model, a preliminary analysis revealed a direct proportional relationship between irradiation and power generation, indicating a potential correlation between these variables. Fig-2 (a) and (b) show the direct positive correlation between irradiation and DC power generation respectively for history data (year 2020). Using the scikit-learn (sklearn) library, a k-NN model was selected for training. By utilizing the capabilities of the sklearn library and implementing the k-NN algorithm, the predictive model was trained.

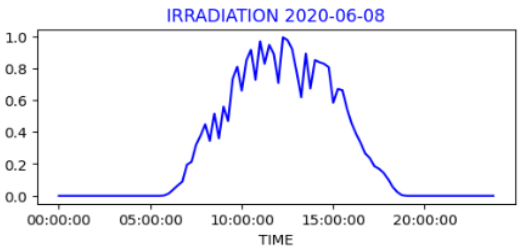

Fig-2(a)

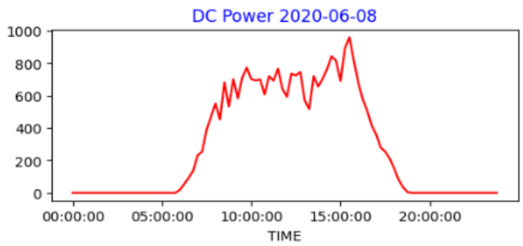

Fig-2(b)

The training process typically began with dataset collection. Historical weather data, including variables such as temperature and solar irradiance, along with corresponding solar power output, were identified from the dataset. After that pre-processing and feature extraction was performed to extract relevant features, and normalize them for analysis. From the available training data 80% was reserved for training and 20% was used for testing the prediction and performing error calculation. During the model training phase, the number of neighbours (value of k=5) was chosen. The new data point's category was assigned based on majority voting among the selected neighbours from the historical dataset.

F. Applying trained model on Rajasthan's Dataset:

After training the model, predicting the potential power generation for a chosen city in Rajasthan (Jodhpur) over the course of a year was performed. This application allows us to estimate the total amount of power that could be generated. Additionally, the project incorporates calculations for daily mean energy generation and determines the plant's capacity based on specific geographic parameters, contributing to informed decision-making regarding solar energy infrastructure and utilization.

Moreover, we can also estimate the size of the solar power plant required to generate that specific amount of power using equation-1. By applying the formula of equation-1 we can determine the optimal plant capacity necessary to meet energy demands effectively.

$$Area = \frac{Energy}{Daily\ Solar\ Insolation \times Efficiency}$$

Equation-1

Efficiency in equation-1 refers to the ratio of usable electrical power output from the solar panel to the total solar energy input from sunlight. It's essentially a measure of how effectively the solar panel converts sunlight into electricity. Efficiency may differ depending on the type of solar panel used, approximately ranging from 15% to 20%. The paper assumes a value of 10% to get an idea on the upper bound area requirement.

IV. RESULTS

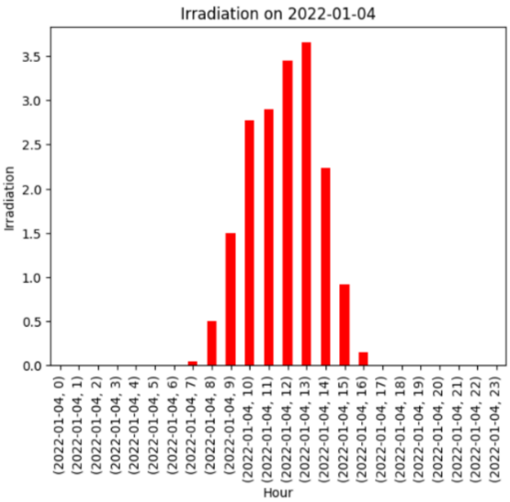

Fig-3 (a)

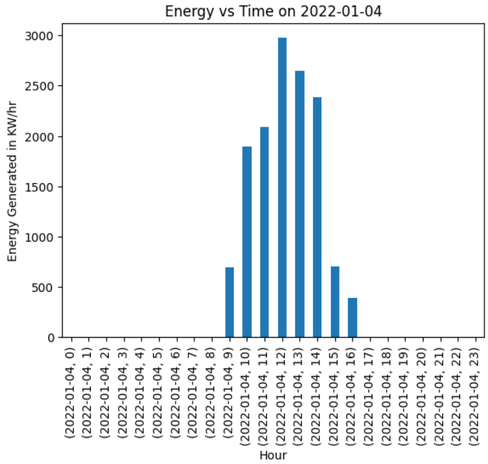

Fig-3 (b)

V. ANALYSIS AND DISCUSSION

Fig-3(a) and (b) show the plot of irradiation and energy respectively on one single day from the dataset used for predicting the potential power generation for Jodhpur. It is observed from the graphs that the irradiation and hence the energy generated by the solar panel peaks around noon.

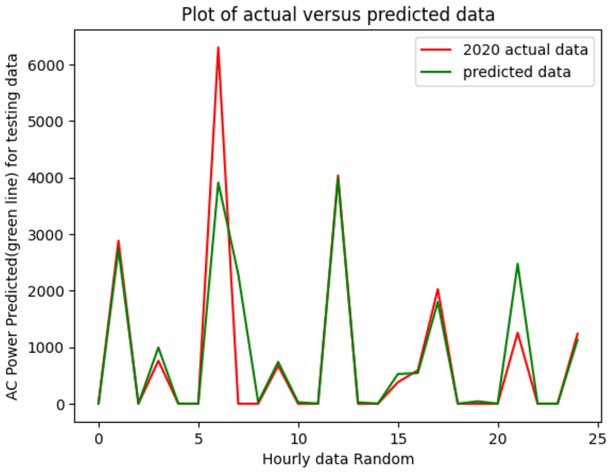

Fig-4

Fig-4 shows the AC power predicted for any random day based on the 2022 solar data comprising parameters like Ambient_Temperature, Module_Temperature, and Irradiation of 20 sensors installed in Jodhpur city of Rajasthan. It is clear from the figure that the predicted power generation (green line) based on k-NN, closely follows the actual power generated (red line) as per history data of year 2020 for the same set of sensors. The correlation coefficient for the actual and predicted values was around 60% which shows a fairly good fit for a non-parametric model like k-NN which is based on local neighborhood.

Based on the predictions available for one year, the average energy generated in kilowatt hour (kwh) per day is calculated by summing all the predictions and then dividing by 365 (days in a year). The value obtained is 19945 kWh per day.

Assuming a mean energy requirement of 19945 kWh per day, an average solar insolation of 6-7 kWh/m^2/day in Rajasthan and a solar panel efficiency of 10%, we can easily calculate the required area for the solar plant using the formula of equation-1. This calculation allows us to determine the precise area needed to accommodate the solar panels and meet the daily energy production target. The calculations show that it is required to set up a solar plant having an area of 28492.86 m^2 (around 7 acres) which will be capable of generating this amount of energy. The area estimation here is on the higher side as the solar panel efficiency is taken as 10%. More efficiency (15%-20%

available with new technology) would translate into comparable power generation capacity with lesser number of panels, which would result into lower area requirement.

$$Area = \frac{19945}{7 \times 0.10} = 28492.86\ m^2$$

VI. CONCLUSION

Our prediction model, built upon historical data and leveraging the K-Nearest Neighbors algorithm, effectively tackles the challenge of predicting solar power generation. By accurately forecasting the energy generation potential of solar power plants in specific regions, our model offers a practical solution for decision-makers. This enhances the efficiency and sustainability of energy planning initiatives, enabling informed assessments of deployment feasibility.

This approach in addition to contributing to the field of renewable energy planning, also enhances the overall efficiency and sustainability of energy infrastructure development. By providing reliable insights into deployment viability, it empowers stakeholders to make informed decisions that align with environmental goals and economic considerations.

This paper underscores the importance of data-driven methodologies in shaping energy policies and investments for moving toward a greener and more sustainable future.

REFERENCES

[1] https://www.un.org/en/climatechange/cop28

[2] https://mnre.gov.in/solar-overview

[3] Naveen Krishnan, K. Ravi Kumar, Chandrapal Singh Inda, How solar radiation forecasting impacts the utilization of solar energy: A critical review, Journal of Cleaner Production, Volume 388, 2023,135860, ISSN 0959-6526,
https://doi.org/10.1016/j.jclepro.2023.135860.

[4] Ardan Hüseyin Eşlik, Emre Akarslan, Fatih Onur Hocaoğlu, Short-term solar radiation forecasting with a novel image processing-based deep learning approach, Renewable Energy, Volume 200, 2022, Pages 1490-1505, ISSN 0960-1481, https://doi.org/10.1016/j.renene.2022.10.063.

[5] Kohli, S., Godwin, G.T., Urolagin, S. (2021). Sales Prediction Using Linear and KNN Regression. In: Patnaik, S., Yang, XS., Sethi, I. (eds) Advances in Machine Learning and Computational Intelligence. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-15-5243-4_29

[6] Martínez, F., Frías, M.P., Pérez, M.D. *et al.* A methodology for applying *k*-nearest neighbor to time series forecasting. *Artif Intell Rev* **52**, 2019–2037 (2019). https://doi.org/10.1007/s10462-017-9593-z

[7] Zhongguo, Yang & Hongqi, Li & Liping, Zhu & Qiang, Liu & Ali, Sikandar. (2017). A case based method to predict optimal k value for k-NN algorithm. Journal of Intelligent & Fuzzy Systems. 33. 1-10. 10.3233/JIFS-161062.

[8] Ömer Faruk Ertuğrul, Mehmet Emin Tağluk, A novel version of k nearest neighbor: Dependent nearest neighbor, Applied Soft Computing, Volume 55, 2017, Pages 480-490, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2017.02.020.

[9] Shichao Zhang, Jiaye Li in KNN Classification With One-Step Computation. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (2020).

[10] Ali, N., Neagu, D. & Trundle, P. Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets. *SNApplied.Sciences.*1, 1559 (2019). https://doi.org/10.1007/s42452-019-1356-9

[11] Samara, Evangelia & Laperre, Brecht & Kieokaew, Rungployphan & Temmer, M. & Verbeke, C. & Rodriguez, Louis & Magdalenic, Jasmina & Poedts, Stefaan. (2022). Dynamic Time Warping as a Means of Assessing Solar Wind Time Series. The Astrophysical Journal. 927. 187. 10.3847/1538-4357/ac4af6.

[12] Rui Chang, Lei Bai and Ching-Hsien Hsu, Solar power generation prediction based on deep Learning, Sustainable Energy Technologies and Assessments, Volume 47, 2021, 101354, ISSN 2213-1388, https://doi.org/10.1016/j.seta.2021.101354.