

# Solar Energy Prediction and Forecasting

Rishi Sankhe

*Received August 12, 2023*

*Accepted September 15, 2023*

*Electronic access October 31, 2023*

The growing shift towards sustainability sees solar energy at its center, incentivizing world energy leaders to exploit the energy source's benefits. However, harnessing the potential of solar power comes with a few challenges. Predicting and forecasting solar energy output plays a quintessential role in overcoming these challenges, including determining the optimum locations for solar production and managing energy demand requirements. Thus, this paper explores the extent to which we can improve the performance of a prediction model and forecast the future power output of solar panels. To achieve this aim, a highly curated dataset from Kaggle was procured. This dataset comprised of weather conditions, solar power output and even solar irradiance data. To further improve the quality of the data being used for the research, the data was processed to fill in any absent data points and identify anomalies. Using this data the paper focuses on two models - Prediction and Forecasting. Previous research work has successfully established prediction algorithms for solar power; however, they have not attained maximum accuracy, and none have centered forecasting future output. For prediction and forecasting, multiple models such as Tree-based models, AdaBoost, and Metalearner have been tested to derive the greatest accuracy for the desired task. The key findings of the research indicate two things. First, that the accuracy of the prediction models cannot be increased by a lot after reaching a particular value (R2 Score of 0.682) unless there are changes in the extensiveness of the data available. Second, that forecasting models can be successfully created and trained to yield numerical values of the power output up to a day in advance.

**Keywords:** Forecasting, Hyperparameters, Model, Prediction, Solar Energy

## Introduction

Within the global energy landscape, society is witnessing a transformative shift towards sustainability, driven primarily by increasing awareness of the vital need to battle climate change and reduce greenhouse gas emissions. As a byproduct of the emergent need to find sustainable alternatives to energy, the global demand for renewable energy sources, especially solar energy, has experienced an extraordinary surge in recent years. As per recent reports by the International Renewable Energy Agency (IREA), 'Solar PV's installed power capacity is poised to surpass that of coal by 2027, becoming the largest in the world'<sup>1</sup>.

The escalating adoption of solar energy technologies is evident across various residential, commercial, and industrial sectors as countries strive to achieve energy security, reduce reliance on fossil fuels, and meet ambitious renewable energy targets. According to another release by the IREA, 'Cumulative solar PV capacity almost triples in our forecast, growing by almost 1 500 GW over the period, exceeding natural gas by 2026 and coal by 2027'<sup>1</sup>. Additionally, governments worldwide support the deployment of solar power in many ways, including offering financial incentives, tax credits, and policy support to accelerate the transition to renewable energy sources. While the rapid expansion of solar power brings the immense promise of sus-

tainability for the future, it also poses a great challenge to global sustainable energy leaders, particularly in effectively harnessing and managing solar energy resources. Power output generation is highly volatile and unpredictable due to unprecedented weather fluctuations, seasonal changes, and daily fluctuations, underscoring the significance of creating accurate prediction and forecasting models. Some more commonly used factors significantly impacting solar energy production are ambient temperature, humidity, wind speed, cloud ceiling, and pressure.

Reliable solar energy machine learning prediction models play a vital role in planning and optimizing the integration of solar power into society's existing energy grids. The accurate prediction of solar power output comes with multiple benefits. Grid operators can anticipate fluctuations and manage energy supply-demand imbalances, thus ensuring a stable and resilient power system. Furthermore, solar energy forecasting continues to become essential as renewable energy penetration rises. Accurate forecasts enable grid operators to anticipate changes in solar power generation, which helps them make informed decisions regarding backup power sources. By leveraging precise forecasting models, energy stakeholders can also create strategies for deploying additional renewable energy sources, such as wind power, to create a balanced and reliable renewable energy mix. Thus, the key research questions that this research paper aims to address are:

- 
- 1 To what extent can the accuracy of prediction models be maximized and
  - 2 Is it possible to create a model that can numerically forecast values for power output from solar plants.

By exploring these questions, this paper will try to create models that can optimize energy generation, improve grid management, and enhance energy efficiency.

## Literature Review

In recent research, several works have attempted at delving deeper into the practice of employing machine learning techniques for predicting short-term solar output in industrial scale solar plants. One example of such, Sun, Yuchi, et al. (2019), comprehensively explored these techniques, including support vector regression model, artificial neural networks, and hybrid or stacked models, shedding light on their strengths and drawbacks<sup>2</sup>. Similarly, R. Gupta et al. (2017) offered an extensive and detailed overview of data analytics methods for predicting solar power generation. Even this work focused its discussion on statistical models, machine learning algorithms, and artificial neural networks while also examining data sources like meteorological data and satellite imagery to check for its impact on increasing the accuracy of models that are trained<sup>3</sup>. Both the papers that have been mentioned above, made use of common machine learning techniques such as hyperparameter tuning to get the best values for the hyperparameters that are used in the creation of the machine learning models. Seeing that the tuning process has increased the performance of the models outlined considerably, this paper will ensure that all models are tuned before being fitted to the dataset. Additionally, another common trend in both these papers is the usage of artificial neural networks as a model to predict the power output. While artificial neural networks do non-linear modeling and can-do real-time predictions, because of the complexity as well as drawbacks of artificial neural networks such as their tendency to overfit to the dataset, these will not be implemented and instead the paper will implement more gradient boosting based and tree-based models that try to minimize errors by ensuring that the fitting process is performed accurately. Furthermore, using satellite imagery is a very novel idea that contributed towards the improvement of the model accuracy. However, due to the inability to find the satellite imaging data for the locations that have been included within the dataset that will be utilized through this paper, the models have been created without the additional information that could have been greatly beneficial.

Another paper such as A. S. Mohan et al.<sup>4</sup> conducted research reviews that portrayed similar ideas to the papers that have been outlined above.

Furthermore, E. Subramaniam et al. (2023) outlined a machine learning-based approach for predicting solar power gener-

ation with a remarkable 99% AUC metric. The paper's methodology encompassed data collection, preparation, feature selection, model selection, training, evaluation, and deployment all of which helped them effectively implement the model and thus maximize accuracy<sup>5</sup>. These studies have collectively contributed towards the increasingly evolving landscape of solar energy prediction. Unlike most other research papers, E. Subramaniam et al. (2023) utilizes the 99% AUC metric that is very uncommon. This metric is unique since it focusses on the detection of very rare events or the positive class events that only occur in a few situations. While this is a fresh and unique lens of analyzing the machine learning models, the models in this work will not use the 99% AUC metric as the performance needs to be evaluated on a more general level while not focusing on only the positive class cases.

Through the study of the various papers that have been detailed above, the work done through this paper will be enhanced as the newfound information will enable a more focused and streamlined work process. By understanding the best models that usually work, as well as the kind of data features that have the most important role in predicting the power output of solar panels, the selection of models and the public dataset will be done carefully and smartly to try and improve the accuracy of models. Furthermore, certain machine learning techniques such as cross-validation and hyperparameter tuning that have improved the quality of the existing papers will be implemented in this paper. On the other hand, the drawbacks of the models detailed earlier on will be carefully considered to ensure that these drawbacks are worked on and improved upon through this paper.

## Methods

This section will explore the different methods that will be used in the creation, selections as well as improvement of the machine learning models. As mentioned at the end of section I.I, the methodology will factor in previously existing efforts and methods that have been employed with the aim of achieving the best results possible and will attempt to improve upon the limitations of the previous efforts.

To obtain the positive results as shown in section II as the product of this research, the first step was to search for a dataset that consists of multiple predictive features of solar energy power output as well as enough datapoints for each model to be trained on. Thus, the chosen dataset has been carefully selected from Kaggle to ensure that the models that have been trained are as accurate as possible. The dataset, 'Pasion et al.', comprises the power output produced by solar panels from 12 different locations. Along with the generated power output, other essential data have also been recorded within the dataset. It records the multiple factors that affect the power outputted, such as latitude, wind speed, cloud cover, humidity, visibility, pressure, and other

---

factors. The initial dataset was filtered only to include hours between 10:00 and 15:45 to avoid including data which consisted of times when no power was generated, as it would significantly skew the result of prediction and forecasting. This restriction also made the data more accurate as, within this period, the sun was very high in the sky, which reduced the scope of any artificial objects obstructing the insolation of the sun that was falling directly on top of the solar panels. This dataset was used for both prediction model creation as well as forecasting. All the models were trained using the pandas and numpy libraries on the local jupyter notebook on the MAC OS GPU.

### Exploratory Data Analysis and Feature Engineering

Before the models were created, the data provided by the dataset was effectively analyzed as a part of the research elaborately explained in this paper. The first step for analyzing the data was a simple proofread to develop a general understanding of the correlation between the power output and other factors. This proofreading helped develop a preliminary understanding of what models can be used and what correlations need to be explored in greater detail. The following are the different statistical graphs employed to understand the data in greater depth:

- A. **Confusion Matrices** - Because the database comprises unlinked features, the model employed confusion matrices for data exploration. While confusion matrices are traditionally used to determine the performance of a prediction model, for this particular use case, they were used to understand label relationships. As mentioned previously, the dataset consists of unlinked features, and thus to truly understand the dependencies or relationships between various features that could effectively be integrated within model creation, multiple confusion matrices were needed to understand the relationship between a particular factor and the power output<sup>6</sup>.
- B. **Scatterplots** – Scatterplots were used to map the correspondence of each variable or factor that affected the power output, which would help later in the process of feature engineering. Primarily, scatterplots helped visualize the pattern and relationship between two variables. However, they were also useful for visualizing data distribution, identifying outliers and detecting heteroscedasticity<sup>6</sup>.
- C. **Heatmaps** - Heatmaps were uniquely used for data exploration. While they function very similarly to scatter plots, they are much more effective and easier to interpret as they color code large and complex data from a 2D matrix, simplifying identifying any patterns and trends within the data<sup>6</sup>.

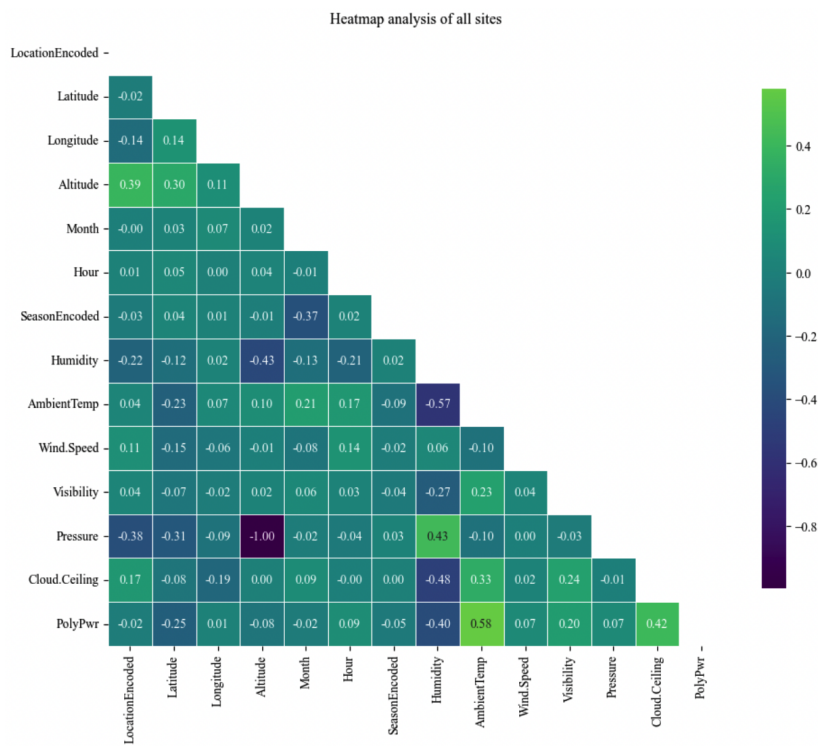
Upon examining these preliminary trend identification diagrams, certain features were engineered to increase the prediction ac-

curacy of the models. Feature engineering helped in creating various categorical variables that could be used for our model. It would help us identify more patterns within the data itself. The first step of engineering features was to perform one-hot encoding of the 'location' and 'season' variables within the data. One-hot encoding converts categorical variables such as 'location' and 'season' into numerical values that can be further used to train the models. The next feature engineering step is to create cyclical features using the month and hour data available. After creating the cyclical features, another correlation analysis was conducted between the newly created features and their actual values, resulting in a perfect correlation, thus allowing the Month and Hour features to be dropped from the columns that need to be used to train the model. Once all feature engineering was done, another heatmap was employed to perform a new correlation analysis between the features and the various sites. *Note:* For the process of feature engineering, the paper refers greatly to work done by an MIT researcher - published on medium.com<sup>7</sup> who had written an extension on Christil Pasion et al. (2020). The code used for feature engineering was referred to for the completion of this paper.

### Prediction Model Selection

For this study, the Jupyter notebook was employed to compare various modelling algorithms to determine which one is the best fit for predicting the power output. The notebook itself only provides the user with a Python kernel, and thus to program different models, other libraries first need to be installed on the GPU and then imported within the code. Using these libraries, the models are created. The scikit learn library was especially important for the accuracy of the model. From the scikit library, practically everything ranging from the correct module for splitting the dataset into train and test, the various metric tests and, most importantly, the model regressor, were all imported. In this research, five different algorithms were compared. Out of these five, four are base models or more popularly known as "base learner" models. The fifth one, however, was a model created using a stacking regressor imported from the stacked ensemble build, which combined all the base models to become a "meta learner" that combines the function of the others. The five models work as follows:

- A. **XGBoost** - This model, more commonly known as extreme gradient boosting, works by iteratively adding trees, calculating gradients, and repeatedly adjusting the models' predictions until the stopping criterion is reached. All the iteratively added trees combine to make the final ensemble of trees of the XGBoost model, which is then used to make predictions on the test data<sup>8</sup>.
- B. **Random Forests** - The Random Forest regression machine combines the regression patterns and predictions of multi-



**Fig. 1** CORRELATION ANALYSIS HEATMAP EXCLUDING ENCODED FEATURES FOR ALL DATA SITES

ple decision trees to make accurate and robust predictions for regression tasks. Each decision tree within the Forest contributes towards the final predicted output. The aggregation of the outputs of each tree helps increase the reliability of the output that is given. The concept of randomness originates within the fact that the data and selected features that are fed into the regressor are selected randomly from the train data set, thus introducing diversity to the prediction. This also helps reduce overfitting to the train data, which can reduce the accuracy of the final model<sup>9</sup>.

**C. AdaBoost** - AdaBoost or Adaptive Boosting is a learning algorithm used for classification purposes. This boosting algorithm essentially works towards creating a strong classifier by combining multiple weak classifiers. Weak classifiers do not work the best but tend to perform better than random guessing. These classifiers are weighted as per their importance to determine which classifier can vote with more importance, thus affecting the performance of the model<sup>10</sup>.

**D. Support Vector Machine** - The support vector machine works by mapping the data available within the train data set onto a high-dimensional feature space to allow the data to be categorized, especially when the data are not linearly separable. The distinction between the mapped

categories is then found as a separator which results in the modification of the data to allow the separators to be drawn as a hyperplane. The newly modified data is what is used to predict and categorize to which group the test data should belong<sup>11</sup>.

**E. Stacking Regressor (Meta Learner)** - In machine learning, stacking is a way of assembling regression models to allow the final model to have two layers of estimators. The first layer will comprise all the baseline models that have been used to predict the output on the dataset. The second layer will be the meta learner or regressor that takes the predictions of each baseline model as a separate input which is then further used to predict. The stacking regressor tends to have the greatest accuracy since it combines multiple regressors, thus enabling better training of the model<sup>12</sup>.

### Prediction Model Development

For this research, the accuracy of each distinct model was assessed using the entire dataset - slightly modified due to the previously engineered features - wherein the hyperparameters of each model were tuned using a randomized search cross-validation process. This process is used in place of a grid search cross-validation wherein it is significantly harder to exhaustively explore all the parameters available to a model exhaustively. In

---

the randomized CV search, one usually makes you of k-fold cross-validation where the dataset is distributed into k subsets of approximately equal size. To make the training reproducible and increase the accuracy, the value of k or cv was set to 4, implying that the same subset was tested four different times. To restrict the time taken for hyperparameter tuning, the maximum number of iterations or combinations of various hyperparameters was limited to 1000, implying that a total of 4,000 fits would occur. Before conducting hyperparameter tuning, initiating, and defining the range of values or options that each parameter could assume was important. Thus, lists were created defining these various possibilities.

### Forecasting Model Selection

Almost all the models selected for the purpose of forecasting were like the ones that were selected in the prediction process as well. Forecasting models can be defined as any model created to use a particular set of data and make future predictions of any value that has been outlined. As mentioned earlier, forecasting models were created in the domain of solar energy to make informed decisions in the maintenance of energy supply and determine whether a particular location is worth further developing upon. The following are the forecasting models that were selected:

- A. **Support Vector Machine** - Like the prediction model, even the forecasting model tries to locate a hyperplane that best fits the dataset while trying to maximize the margin between the different classes. In the case of time series forecasting, the model attempts to find a hyperplane that can minimize the loss function while considering a specified margin. It then learns the relationships between the past and the current data. Finally, the prediction is based on the data point's position relative to the hyperplane<sup>11</sup>.
- B. **Random Forest, Gradient Boosting, XGBoost**<sup>8, 9</sup> All these models are collectively known as ensemble models, and they work in a very similar manner to each other and hence have been clubbed together. They function by combining base models or, in this case, decision trees to increase the forecast's accuracy. Usually, for time series forecasting, the data can be modified from a sequential to a tabular format for time series forecasting. Within the tabular format, each row represents features derived from previous observations. The ensemble or model uses historical data to identify patterns and then uses them to predict the next value in the sequence.
- C. **Long Short-Term Memory Layer** – The long short-term memory layer is a type of recurrent neural network specifically designed for sequence prediction tasks. It functions by taking a sequence of previous input data from within

the dataset and tries to identify temporal data patterns. The model makes use of modules that can record both short-term as well as long-term dependencies. Each module processes a unique time step within the sequence while updating its internally stored memory and making predictions that can help produce a numerical output<sup>13</sup>.

### Forecasting Model Development

For creating a forecasting model, the dataset had to be initially filtered to only represent data points from one singular location, which was called 'Hill Weber'. This location was chosen after exploring the data in detail and identifying that there were almost no data points of the important time stamps missing for this location compared to the others. This was done to increase the accuracy of the model, as including data from all 12 locations would result in great variations and errors in the performance of the forecasting model as not only the locational data but also the weather conditions would drastically vary. Thus, in the development of the forecasting model, this was the first step that was taken. Since the forecasting model aimed to predict the total power output for the next day, the model was run six different times by filtering the time column for the six main time stamps that the power output for each day that was recorded. These time stamps were 1000, 1100, 1200, 1300, 1400 and 1500. After running the model for each of the different time stamps, the forecasted value for the power output was recorded and then tallied to calculate the total output for the next day of power generation. Further, for the development of the model itself, multiple measures had to be taken into consideration. For example, one key measure that was considered normalization of data using the min-max scaler function, which essentially converts the historical data of the 'y' variable - the variable that needs to be forecasted, which in this case is the power output - into a numerical value between 0 and 1. This ensures that different features within the neural networks and other algorithms do not dominate the learning process due to their differing scales. After the forecasting predictions, to scale back the values, a function called the inverse transform was also implemented.

### Metrics Chosen for Model Evaluation

Even after creating a prediction and forecasting model to determine how well a model can perform, a variety of metrics were chosen to test the performance of each model. Other than simply testing the performance of any model, metrics are also required for hyperparameter tuning, which has been outlined before. By having metrics to show the model's performance, the perfect hyperparameter configuration can be determined by checking how the model is performing for various combinations. By doing so, it simply allows the user to maximise the accuracy of the

model in question. By checking the performance, the user can also determine whether the model is overfitting or underfitting the dataset, which is a crucial element in increasing the accuracy of any model. Lastly, they also help in making comparisons between the multiple models tested. The various metrics chosen were as follows:

- A. **Mean Absolute Error** - The mean absolute error of any model is calculated by taking the summation of the absolute difference between the actual and calculated value of each predicted output over the entire dataset and then dividing the sum by the total number of data points within the dataset<sup>14</sup>

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

MAE = mean absolute error

$y_i$  = prediction

$x_i$  = true value

n = total number of data points

- B. **Root Mean Squared Error** - Root Mean Squared Error or RMSE is a measure of how far away do the model's predictions fall from the measured true values. This distance is usually measured using Euclidean distance. To compute RMSE, the residual or the distance of a prediction from the real measured value for each prediction must be calculated. Then the norm of each residual must be calculated, and the mean of these residuals should be taken and square rooted.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x}_i)^2}{N}}$$

RMSE = root mean-square deviation

i = variable i

N = number of non-missing data points

$x_i$  = actual observations time series

$\bar{x}_i$  = estimated time series

- C. **R2 Score** - R2 or r-square is a commonly used metric to determine the performance of a model, which works by taking the sum of the squares of the residuals and comparing it with the total sum of squares. This total sum calculates the perpendicular distance between each data point and the trend line that passes through<sup>15</sup>. The formula for calculating the same is:

$$R^2 = 1 - \frac{RSS}{TSS}$$

$R^2$  = coefficient of determination

RSS = sum of squares of residuals

TSS = total sum of squares

Note: All formulae and variable interpretations are taken from Google's search engine and no source.

These metrics for analysis, especially RMSE, were chosen based on their ability to give emphasis to outliers as well as quantify the magnitude of errors and even factor the sensitivity that errors have. On the other hand, the R2 score was primarily to understand the degree of variance between the datapoints included in the dataset. Another metric that could have been initially considered was the p-value. However, it was ultimately excluded from the analysis of models as it did not directly quantify the accuracy of each model but rather attempted at explaining whether certain features are more likely than the other in explaining the target variable.

## RESULTS

This section presents the performances of both the prediction and forecasting models. The forecasting model evaluation also provides the total value forecast for the next day by each different model. Different metrics such as the R2 score, mean absolute error and root mean squared error are used to evaluate the model performances.

### Prediction Model Evaluation

By comparing the results of each model with the values for the metrics provided in Table 1 above, we can deduce that the Stacking regressor - which was a combination of all the baseline models - provided the most accurate result with an R2 Score of 0.682. Similarly, Table 2 on the right-hand side shows the results of the models created in O. Abiodun, (2022)<sup>7</sup> an extension of Chrstil Pasion et al. (2020)<sup>16</sup>. Upon closer inspection it is noted that like the stacking regressor created through this paper, the meta learner model for the paper which has the same function as the stacking regressor has an R2 score of 0.681. Notably, despite employing various models that function and perform regression and train to the dataset in unique manners, the performance of the final stacked models remained almost constant. However, some improvement was still seen in the stacked model created through this research.

### Forecasting Model Evaluation

The table above presents all the values that each model forecasted for each time stamp that was outlined. At the bottom of the table, the total value has also been tallied to portray the total power output that the solar plant of horizontal solar panels would produce in the 'Hill Weber' location which was selected carefully by exploring the data and checking to see which location had the most number as well as informative datapoints. Along with the tallied power output, the RMSE or the root mean

Model Name	Mean Absolute Error	Root Mean Squared Error	R2 Score
Support Vector Machine	2.763	4.079	0.672
Random Forest	2.775	4.094	0.670
XGBoost	2.745	4.064	0.675
AdaBoost	2.802	4.102	0.668
Stacking Regressor	2.556	4.017	0.682

TABLE 1. PREDICTIONS MODEL CREATED WITHIN THIS RESEARCH

Model Name	Mean Absolute Error	Root Mean Squared Error	R2 Score
K Nearest Neighbours	2.971	4.403	0.618
Random Forest	2.787	4.095	0.670
Light Gradient Boosting Machine	2.723	4.054	0.677
Deep Neural Network	2.709	4.142	0.662
Meta Learner	2.670	4.024	0.681

TABLE 2. PREDICTION MODELS CREATED DURING PREVIOUS RESEARCH

**Table 3** FORECASTING MODELS ALONG WITH VALUES FORECASTED FOR EACH TIME STAMP AND RMSE FOR LOCATION HILL WEBER

Time Stamp	Forecasted Values and Error of Each Model									
	Gradient Boosting		Random Forest		Support Vector Machine		Extreme Gradient Boosting		Long Short Term Memory Layer	
	Value	RMSE	Value	RMSE	Value	RMSE	Value	RMSE	Value	RMSE
1000	10.59	5.56	7.81	3.36	6.42	3.16	8.00	3.26	8.60	3.59
1100	12.98	6.40	11.22	3.99	10.79	3.99	13.68	4.21	14.84	4.18
1200	14.79	7.17	14.88	4.69	15.47	4.74	14.02	4.83	14.09	5.04
1300	15.35	7.16	12.39	5.16	12.68	5.15	10.94	5.43	10.22	5.96
1400	14.66	7.25	14.31	5.27	13.53	5.22	14.27	5.72	12.54	5.87
1500	13.00	6.78	12.22	4.83	12.71	4.68	12.10	4.93	10.42	5.34
Total / Average	81.37	6.72	72.83	4.55	71.60	4.49	73.01	4.73	70.71	4.99

squared error for each time, stamp has also been included since the forecasting for each time stamp was done individually. The RMSE of the models has been averaged at the bottom of the table alongside the tallied output. As seen from the table above, the model that would be the best model for forecasting purposes is the Support Vector Machine since its average RMSE is 4.49 - the lowest - after which comes the Random Forest forecasting model with a 4.55 RMSE.

## Discussion

Using our results, we can identify that the goal of creating prediction and forecasting models that can be used to analyze a dataset of power output from horizontal solar panels has been successfully achieved. To begin with, by comparing Table 1 and Table 2, we can see how the stacked model created through this research performed better than the meta-learner model created within the paper written by the MIT researcher – Abiodun Olaye. This shows that these prediction models - especially ones that use meta-learners- can be more accurate if the correct models are combined. The values obtained in the tables before can also be compared with Christil Pasion et al. (2020)<sup>16</sup> wherein this dataset originated. Within that paper, the models were trained using AutoML available on H2O.ai and the best R2 score using cross-validation turned out to be 0.687, which is 0.05 better than

the stacking regressor model created during this research paper. This difference in accuracy may stem from a variety of reasons. The first was that the GPU processing might be better than that of MACOS, which resulted in better accuracy. Furthermore, the models used for that paper were different, such as Deep Learning and a Generalized Linear Model, thus having a greater impact. Notably, however, despite trying multiple different models, the prediction accuracy could only be maximized until a certain point. This point, being close to an r-squared value of 0.7, implies that the prediction models tested through this research can most definitely be employed by solar plant managers in their quest to plan and optimize the integration of solar power within their grids. Especially in a setting as unpredictable and volatile as the solar plant, weather conditions are crucial in determining how much power will be produced on a particular day. Having an r-square correlation value that high means that the models are performing much better than one would expect. This performance can be credited to the various weather conditions that were included in the dataset. Despite this, the dataset can still be made more comprehensive with the massive number yet to be recorded. It is important to note that using correlation as a factor is essential to not only understanding the performance of models but also in understanding the importance of certain features over the other in impacting the target variable that the paper is trying to predict, namely the power output of the solar panels.

---

This paper derives its novelty from the forecasting models that it employs. Most papers, as outlined within section I.I., simply focus on creating machine learning models that can fit themselves as per the data provided within a particular dataset and then run prediction models. However, this paper's forecasting models aim to provide the user with a numerical output for the power for the next day. As mentioned previously, creating such forecasting models is becoming ever so vital in the field of sustainability to maximize the potential of solar energy as much as possible, especially due to the increasing shift towards sustainable energy sources. While the forecasting model created above is not the best and can be improved upon, it lays the foundation for future development. By providing a numerical output for each time stamp with relatively high accuracy and a root mean squared error of as low as 4.49, especially in the support vector machine model case, the model can already be used to forecast power output in large solar plants and determine the plan of action for the near future.

Despite trying to make the models as accurate as possible, one drawback was that the prediction models only train and test for hyperparameters till a certain value. To perform a more exhaustive search of the best parameter combination, a more in-depth analysis of the dataset needs to be conducted to determine the limit to yield a value for the hyperparameter that would maximize the accuracy of the model. Moreover, to improve the accuracy of the model, making the dataset even more detailed and extensive would be beneficial. This could be done by taking data points at more frequent time intervals and for a longer duration than a year and three months. By doing so, the model would have more data to fit on and thus try and improve its accuracy. Another potential improvement relies on the GPU on which the model fitting and hyperparameter tuning are done. While conducting the research, the same process of hyperparameter tuning was performed on both google collab and the computer's local terminal, and despite what one would expect, due to varying processing strengths, the hyperparameter tuning that was performed was slightly different, which resulted in the accuracy and the R2 score of the GPU processed model being greater than the same model that was tuned on google collab instead.

To improve the forecasting models, various methods can be implemented. Firstly, because the forecasting models have not been created with a dataset filtered for a season, but for a particular location, the variation in the power output is greater than expected, thus resulting in a considerable amount of error in the forecasted value. The dataset should ideally be filtered as per season to reduce the error. However, it has not already been done because doing so would have resulted in there being too few data points for the model to train itself on. This might have increased the accuracy of the model. However, it would not have been a true reflection of the model's capabilities as with fewer data points, there is a smaller scope for error. Secondly,

the current model simply predicts the power output expected at a particular timestamp of the next day rather than the entire expected power output. To make the model predict the entire predicted output instead of only at a single timestamp, an additional piece of code would be required, wherein the model would have to be trained as per the different timestamps, and a cumulative prediction would be performed at the end of the runtime. Lastly, because of the complex nature of forecasting any value, the models currently in use are simple, so the root mean squared error in some of the time forecasts has gone as high as 7.258.

## Conclusion

Previous research papers that have been elaborated upon in section I.I encompass many methods to create prediction models trained per various datasets. However, none provide a conclusive model for forecasting solar output and a numerical value for the expected power output. The research work being discussed through this paper digresses from the works mentioned earlier for that very reason. It tests numerous forecasting models, provides the values yielded, and determines which model is the most accurate thus exploring an avenue of integrating machine learning in solar energy and sustainability that few or none have attempted earlier.

Thus, to conclude, we must answer both the research questions that had originally been declared. The first is to check to what extent the prediction models have been improved and the accuracy maximized. The second to check whether models that can use time series to forecast future power output data have been created.

Firstly, through this research, it is observed that creating a model that can train itself according to previously accumulated data from a solar plant can only reach a certain accuracy, after which increasing the accuracy will become increasingly hard unless additional parameters or factors which affect the power output that is produced are introduced within the dataset. Despite trying various regressors and feeding them into the stacking regressor as a base regressor, the accuracy only increased to a certain amount. This finding is also observable in the results section where the accuracies of not only this, but Christil Pasion et al. (2020)<sup>16</sup> and<sup>7</sup> were found to be varying by a margin of only 0.005. This variance was a surprising revelation as the models tested by each paper were drastically different and function in different ways. The reason behind this variation has already been discussed in section IV and has been pinned on the processing power of the GPU of the computer that the code has been written on as well as different levels of extensiveness in the tuning of the parameters. However, since the variation is only 0.005 it can be considered almost negligible in the large scale of things. Thus, this is conclusive evidence of the fact that after a certain point, increasing the accuracy of a prediction

---

model further becomes an increasingly challenging task until and unless new weather features are introduced into the dataset, and the dataset is made even more extensive with more data points for the models to explore.

Secondly, we can conclude that models to forecast future power output have been successfully created. By testing multiple different forecasting models, even the best model has been determined. These models use time series forecasting and are only basic in nature for now. However, upon further development, these forecasting models have the potential to be applied in the commercial landscape to aid solar plant managers in making decisions about the near future.

Additionally, this paper also ensured that in the creation of both, the prediction and forecasting models, issues such as overfitting and underfitting which were visible in papers mentioned in section I.I were clarified. Furthermore, even drawbacks such as selection bias in the case of prediction models was minimized as before any model was created in this paper, all the other research papers were carefully examined to see which models had not been tested before and which of the previously tested models yielded the best results. After doing this careful analysis, both the variety of models were combined and tested through this paper. As mentioned earlier, this paper gains its novelty through its unique method of forecasting a numerical power output. It predicts the power output for different timestamps and tallies it up – a method that has not been implemented before. However, one key drawback of the models that were created in this paper was that for the process of hyperparameter tuning, the perfect combination possibilities were limited and restricted by keeping an upper limit for some of the numerical parameters, which could have limited the chance of achieving a higher accuracy than the one that was obtained.

To conclude, despite the importance of accurate prediction and forecasting, the complexities associated with solar energy generation make this task inherently challenging. The non-linear and dynamic nature of solar power generation and the influence of external factors such as cloud cover, shading, and atmospheric conditions necessitate advanced modelling techniques to achieve accurate predictions. Thus, this research paper addresses the critical need for enhancing solar energy prediction and forecasting models. By exploring a range of prediction and forecasting approaches, the study identifies the most effective methods to improve the accuracy of solar power output estimation. Ultimately, the findings through this research aim to contribute significantly to advancing renewable energy technologies, promoting sustainable energy practices, and facilitating a seamless transition towards a cleaner and greener future.

## References

1 *Renewables - Energy System*, <https://www.iea.org/energy-system/renewables>.

- 2 Y. Sun, V. Venugopal and A. R. Brandt, *Solar Energy*, 2019, **188**, 730–741.
- 3 A. Gupta, K. Gupta and S. Saroha, *Materials Today: Proceedings*, 2021, **47**, 2420–2425.
- 4 S. K. Aggarwal and L. M. Saini, *Energy*, 2014, **78**, 247–256.
- 5 S. E.
- 6 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and Duchesnay, *The Journal of Machine Learning Research*, 2011, **12**, 2825–2830.
- 7 A. Olaoye, (PDF) *Predicting solar power output using machine learning techniques*.
- 8 *How LightGBM algorithm works—ArcGIS Pro | Documentation*, <https://pro.arcgis.com/en/pro-app/latest/tool-reference/geoai/how-lightgbm-works.htm>.
- 9 L. Breiman, *Machine Learning*, 2001, **45**, 5–32.
- 10 N. Verma, *AdaBoost Algorithm Explained in Less Than 5 Minutes*, 2022, <https://medium.com/@techynilesh/adaboost-algorithm-explained-in-less-than-5-minutes-77cdf9323bfc>.
- 11 *IBM Documentation*, 2021, <https://www.ibm.com/docs/en/spss-modeler/saas?topic=models-how-svm-works>.
- 12 *Stacking in Machine Learning*, 2019, <https://www.geeksforgeeks.org/stacking-in-machine-learning/>.
- 13 J. Brownlee, *Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras - MachineLearningMastery.com*, <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>.
- 14 *What is Mean Absolute Error | Deepchecks*, <https://deepchecks.com/glossary/mean-absolute-error/>.
- 15 *Academic Skills Kit*, <https://www.ncl.ac.uk/academic-skills-kit/>.
- 16 C. Pasion, T. Wagner, C. Koschnick, S. Schuldt, J. Williams and K. Hallinan, *Energies*, 2020, **13**, 2570.