Journal of Green Energy and Transition to Sustainability

The Netherlands Press

*Article*

# PREDICTION OF SOLAR ENERGY GENETRATION FROM THE WEATHER DATA USING MACHINE LEARNING

**Kusuma Kurma**

Department of Information Technology Project Management,
St. Francis College,
180 Remsen St #4305, Brooklyn, NY 11201, United States.

Orchid Id: 0000-0003-1458-883X
E-mail: kkurma@sfu.edu

**Abstract.** The significant growth of the amount of grid energy supplied by renewable sources is one of the main objectives of smart grid efforts. One of the challenges in incorporating renewable energy sources into the system is the intermittent and unpredictable nature of electricity generation. The necessity to relocate generators to meet demand as production fluctuates makes it imperative to forecast future renewable energy output. While building complex prediction models by hand for huge solar farms may be feasible, doing so for distributed power generation in the grid's millions of homes is a difficult undertaking. This research investigates machine learning methods for automatically generating site-specific forecasting models for solar power generation using National Weather Service (NWS) weather predictions in order to address the problem. by comparing several regression techniques to create prediction models, including multilayer perceptron’s and neural networks with long-term memory. combining historical NWS forecasts and sun intensity data from a weather station that has been operational for about a year to assess the accuracy of each model. Our findings demonstrate that predictive models developed for our site employing seven different weather forecasting parameters are more accurate than current forecast-based models.

**Keywords:** Multi-layer perceptron, long short term memory, weather forecast data, Prediction, accuracy, root mean square error

**Journal of Green Energy and Transition to Sustainability** Volume 1 Issue 1
Received Date 05 June 2022
Accepted Date 25 June 2022
Published Date 23 July 2022
Online at https://www.jgets.org/

## 1. Introduction

The Earth's surface receives enough sunlight in little over an hour and a half to provide all of the planet's energy requirements for a whole year. Solar technology uses photovoltaic (PV) panels or mirrors to focus solar radiation and turn it into electrical energy. This energy can be converted into electricity, or it can be chemically or thermally stored [1]. Tthe fundamentals of solar radiation, centralised solar photovoltaic and thermal technologies, grid system integration, and non-hardware components are all covered in the resources and information provided here (expenses) [2]. sun energy (soft charge).

Additionally, you may discover more about solar energy usage and the solar business. Additionally, you may learn more about solar energy and the cutting-edge research and development that the U.S. Department of Energy's Office of Solar Technology carries out in these fields. Light released by the sun is known as solar radiation or electromagnetic radiation. The amount of solar energy that reaches every location on the Earth's surface fluctuates, even though every place on Earth receives some sunshine each year [3]. Solar energy technology is able to catch this radiation and transform it into usable energy.

Two is sometimes preferable to one. One illustration is the fusion of solar power with storage technologies. The cause is that solar energy is not always available when it is most required. Summer afternoons and nights see the highest levels of electricity use since solar power generation is at its lowest during these times [4]. As a result, people who work during the daytime return home during these periods and begin consuming power to cool their houses, cook, and run appliances [5].

Even when the sun isn't shining, storage enables solar energy to contribute to the supply of electricity. Additionally, it can assist in addressing irregularities in the grid's transmission of solar energy [6]. The quantity of sunshine reaching the photovoltaic (PV) panels, or a centralised solar thermal power system has changed, which is the cause of these variations (CSP). Season, time of day, clouds, dust, haze, or obstructions like shadows, rain, snow, and dirt can all have an impact on the amount of solar electricity produced. The energy storage system may be separate from the solar system or co-located with it, but in any case, it can aid in more efficient solar integration into the overall energy system [7].

Because solar output is closely correlated with solar intensity, we are motivated to anticipate it. If future solar intensity modelling is done accurately by supplying current weather data for a given location, the solar energy production of that area may be projected in the near future. computed with greater precision [8]. One of the obstacles to the widespread integration of renewable energy into the national grid is being addressed by moving toward more precise projections of solar power output. The emphasis is on conventional (fossil fuel-based) power generation because of the very variable nature of renewable energy production. The majority of grid utilities are currently compelled to alter their conventional generating rates in response to almost instantaneous renewable power (ie wind gusts on another day). When renewable power supply is expected to fluctuate inversely with customer demand, grid sites face a major difficulty. It is expensive for websites to operate during these "sloping" times, where the usual creation process abruptly rises or decreases [9].

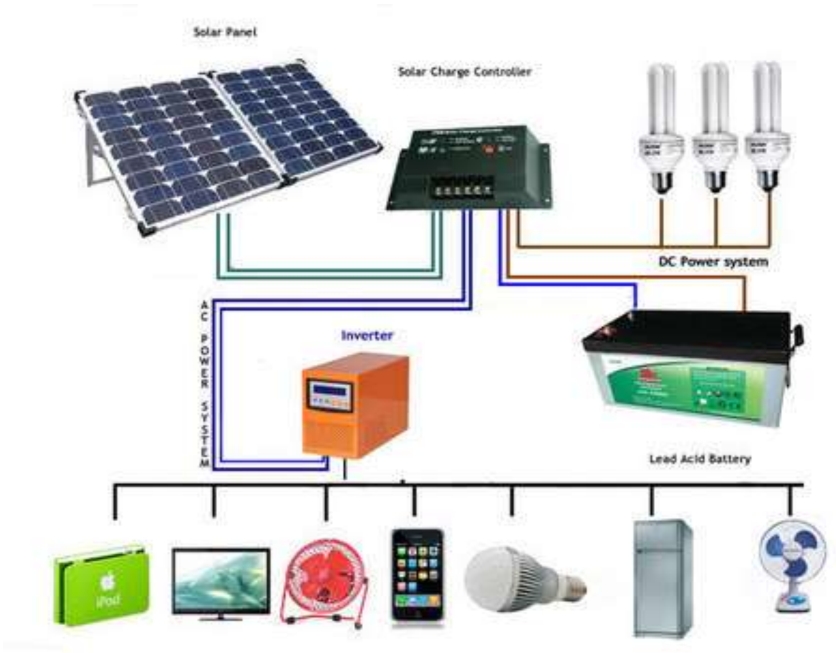


Figure 1:Solar grid [10].

## 2. MLPREGRESSOR

The partial derivatives of the loss function in relation to the model parameters are computed each time to update the parameters, which causes the MLPRegressor to go through iterative learning. In order to minimise model parameters and prevent overfitting, a harmonic component may also be included to the attenuation function [11].

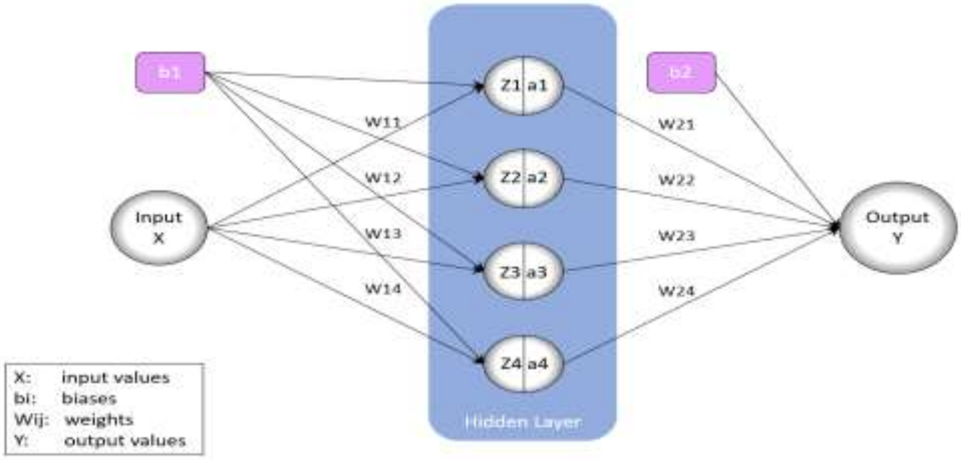


Figure 2:MLP Regressor [12].

### 2.1. Long Short-Term Memory

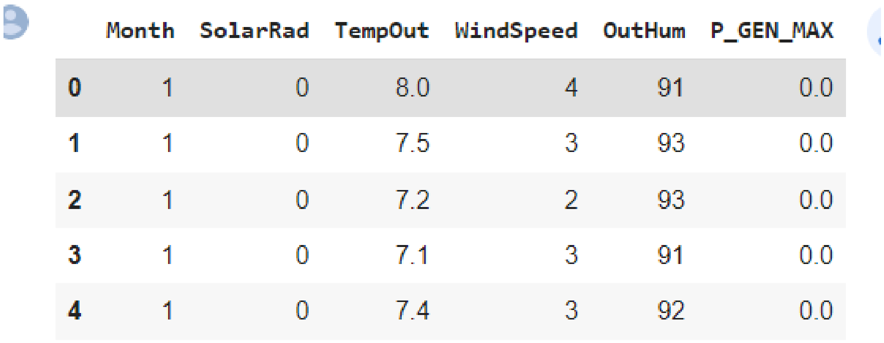
LSTM networks are highly suited for categorization, analysis, and prediction generation based on this data since there might be unpredictably long gaps between critical events in time series. The leakage gradient can be problematic when training a standard RNN; as a result, the LSTM was developed to address this issue [13].

The relative insensitivity of LSTM to gap length makes them superior to RNN, Hidden Markov Model, and other sequence learning algorithms in many situations [14].

Through the employment of a number of "gates," LSTM regulates the flow of data into, through, and out of the network. In a normal LSTM, there are three doors: forget doors, doors, and exits. Each of these gates is a separate neural network, and they may be viewed as filters [15].

## 3. Dataset Preparation

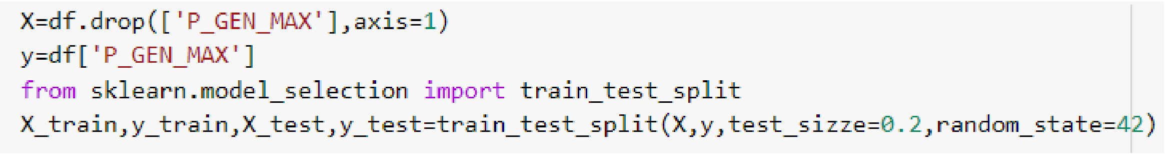
Weather data from that region is taken, including Solar radiation, Temperature output, Wind speed, and Output humidity, and the output data was electricity generated. First null items in the data should search for the feature engineering process. The null **elements** should replace with mean, median, and mode data. If any column or row is categorical, they were converted into numerical items by replacing them with suitable numbers such as yes-no columns. After completing the data cleaning process, values in data should be converted on the scale of 0 to 1 as the neural networks perform well when data is in the range of 0 and 1. Furthermore, Minmaxscalar is employed for this method. This function will convert the data into desired numbers. This feature engineering process plays a crucial role in model performance.



| | Month | SolarRad | TempOut | WindSpeed | OutHum | P_GEN_MAX |
|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 8.0 | 4 | 91 | 0.0 |
| **1** | 1 | 0 | 7.5 | 3 | 93 | 0.0 |
| **2** | 1 | 0 | 7.2 | 2 | 93 | 0.0 |
| **3** | 1 | 0 | 7.1 | 3 | 91 | 0.0 |
| **4** | 1 | 0 | 7.4 | 3 | 92 | 0.0 |

### 3.1. Splitting Data And Labelling

Data is converted into two parts by splitting into training and testing data as required along with the label for data is given. The input feature columns are given with a label 'X' as well as the output column is given with a label 'Y'. Train data should have the maximum percent of data. So, twenty percent of data is taken as test data and rest of the data is taken for training the model.

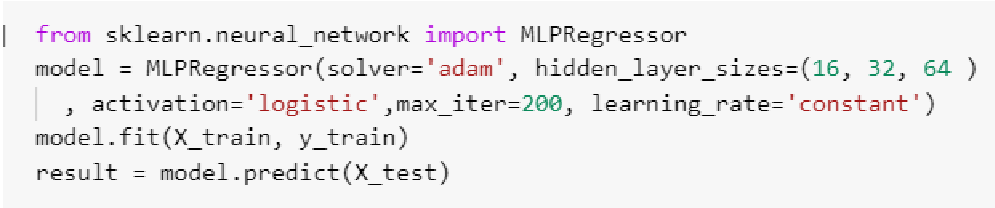


```
X=df.drop(['P_GEN_MAX'],axis=1)
y=df['P_GEN_MAX']
from sklearn.model_selection import train_test_split
X_train,y_train,X_test,y_test=train_test_split(X,y,test_sizze=0.2,random_state=42)
```

### 3.2. Mlp Regressor Model

Large-scale applications are not the focus of this implementation. Scikit-learning, in particular, does not support GPUs. The MLPClassifier class provides frameworks with considerably greater flexibility for creating deep learning architectures, as well as a multi-layer perceptron (MLP) technique for training that uses backpropagation for faster GPU-based implementations. The two arrays that MLP uses for training are array X of size (n samples, n functions), which contains the training data encoded as a floating point feature vector, and array Y of size (n samples), which contains the target values (labels of the class for sample training).

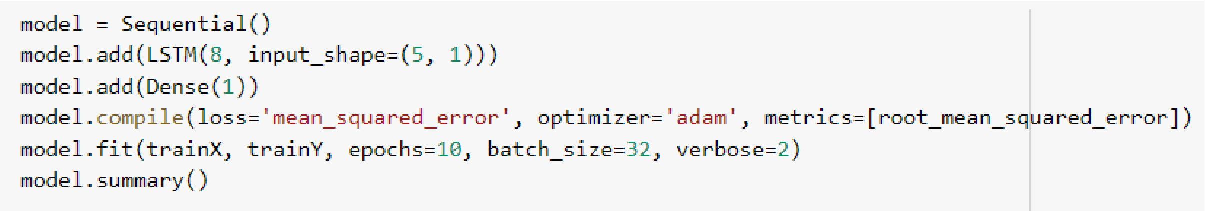
The MLPRegressor class implements a multilayer perceptron (MLP), which may be conceptualised as utilising the recognition function as an activation function, and trains via reverse propagation without an activation function in the output layer. As a result, it returns a collection of continuous values and employs squared error as the loss function. Both the MLPRegressor and the MLPClassifier employ an alpha parameter for the regularisation term (L2 regularisation), which discourages high magnitude weights and hence prevents overfitting. An alpha-valued variable decision function is displayed in the graph below.



```
from sklearn.neural_network import MLPRegressor
model = MLPRegressor(solver='adam', hidden_layer_sizes=(16, 32, 64 )
  , activation='logistic',max_iter=200, learning_rate='constant')
model.fit(X_train, y_train)
result = model.predict(X_test)
```

## 3.3. Long Short-Term Memory

Long-term short-term memory (LSTM) repeating neural networks can learn order dependency in sequence prediction issues. Complex problem areas like machine translation, speech recognition, and others call for this behaviour. A challenging area of deep learning is LSTMs. Keras sequential model is used to build LSTM neural network. The first layer in the sequential model is LSTM, with eight neurons. Moreover, the input shape is (5,1) as the data size is 5. The output layer in the neural network is a dense layer with a single neuron with an activation function of softmax.

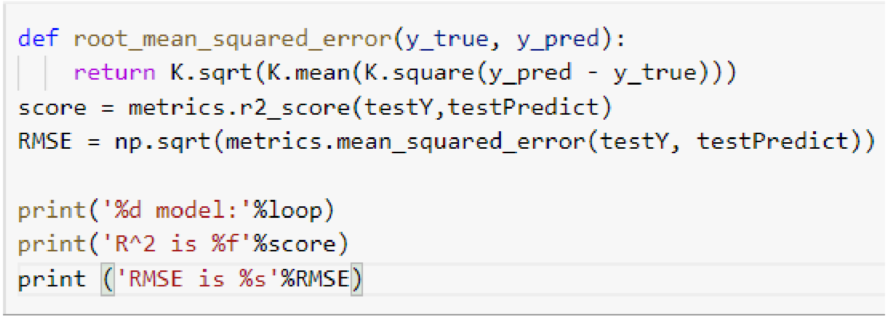


```
model = Sequential()
model.add(LSTM(8, input_shape=(5, 1)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam', metrics=[root_mean_squared_error])
model.fit(trainX, trainY, epochs=10, batch_size=32, verbose=2)
model.summary()
```

After building a Neural network model, it must be compiled using some loss and optimization functions along with metrics that should be calculated for evaluating the model. For this model, the mean squared error is the loss function, which calculates the loss of train and validation data. Furthermore, adam function is used as an optimizer for learning the model. For calculating metrics, the RMSE (root mean squared error) function is created to evaluate model performance. Training data is fitted into the model with a batch size of 32.

## 3.4. Model Evaluation Performance Testing

The discrepancy between values (sample or population values) predicted by a model or estimator is frequently measured using the root mean square deviation (RMSD) or base mean squared error. both quantity and measured values (RMSE). The root-squared deviation, or RMSD, is the difference between the projected value and the actual value. When computations are made outside of the data sample used for estimate, the deviation is referred to as the error rather than the residuals (or prediction error).The amounts of forecasting errors for several data points are combined using the RMSD to provide a single indication of predictive power. Since RMSD depends on size, it should only be used to compare the predictive performance of different models on the same dataset, not across datasets. The data would be perfectly suited with a value of 0 (almost never achieved in practise), and RMSD is always non-negative. Generally speaking, a smaller RMSD is preferable than a greater RMSD. However, because the measurement depends on the scale of the numbers used, comparisons across other data kinds won't be meaningful. The mean squared error's square root is known as RMSD. Larger mistakes have a disproportionate impact on RMSD since each error's influence on RMSD is proportional to the square of its magnitude. As a result, RMSD is very susceptible to unrelated causes.



```
def root_mean_squared_error(y_true, y_pred):
        return K.sqrt(K.mean(K.square(y_pred - y_true)))
score = metrics.r2_score(testY,testPredict)
RMSE = np.sqrt(metrics.mean_squared_error(testY, testPredict))

print('%d model:'%loop)
print('R^2 is %f'%score)
print ('RMSE is %s'%RMSE)
```

### 3.4.1.R2 VALUES

The coefficient of determination, often known as R2 or r2, and pronounced "R squared" in statistics, is the rate of change of the dependent variable that can be anticipated from the independent variable(s).It is a statistic applied to statistical models whose main objective is to make predictions about the future or test hypotheses using other pertinent data. Based on the percentage of overall variance in the findings that is explained by the model, it offers a gauge of how repeatable the model's reported outcomes are. R2 has a number of definitions, only some of which are equal.

One of these groups of cases where r2 is used instead of R2 is simple linear regression. When just one intercept is taken into consideration, r2 is just the square of the sample correlation coefficient (i.e., r) between the actual results and the predicted actual values. When extra regressors are added, R2 is the square of the multiple correlation coefficient. In these two scenarios, the coefficient of determination often falls between 0 and 1.

Depending on the definition employed, there are several situations where the computer definition of R2 can return a negative result. This may occur when the predictions are contrasted with the matching results that weren't obtained utilising this data throughout the model tuning procedure. R2 can be negative even while using the model fitting process, for as when running a linear regression without accounting for an intercept or when using a nonlinear function to fit the data. When negative values are present, the data mean offers a better match than the function's value that has been modified to account for this requirement.

## 4. Results And Discussions

In the evaluation of the model MLP accuracy and loss plots are plotted. By observing the loss graph training loss is continuously decreasing and the training loss is also decreasing. While coming to the accuracy plot it is increasing and the accuracy obtained was 80.91 percent.

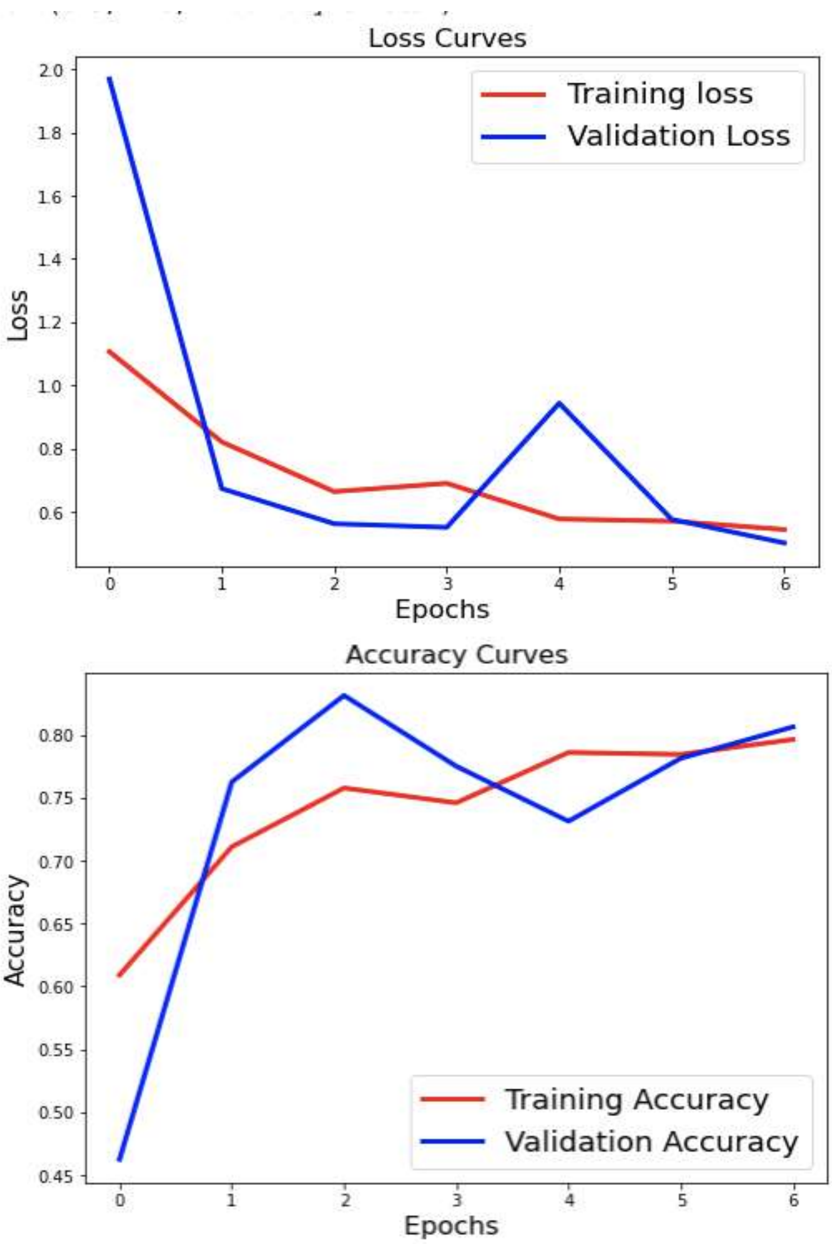


Figure 3:Loss and accuracy plot.

In this graph the true values and predicted values of electricity generation is plotted by using red and brown lines and dots. By observing the graphs for the first 250 samples the difference between true and predicted values are high. While the remaining samples have low error between true values and predicted values this graph shows that the loss between them is decreasing, and the model loss is also decreasing. The final accuracy score for a model obtained was 80.91 percentage for MLP model.

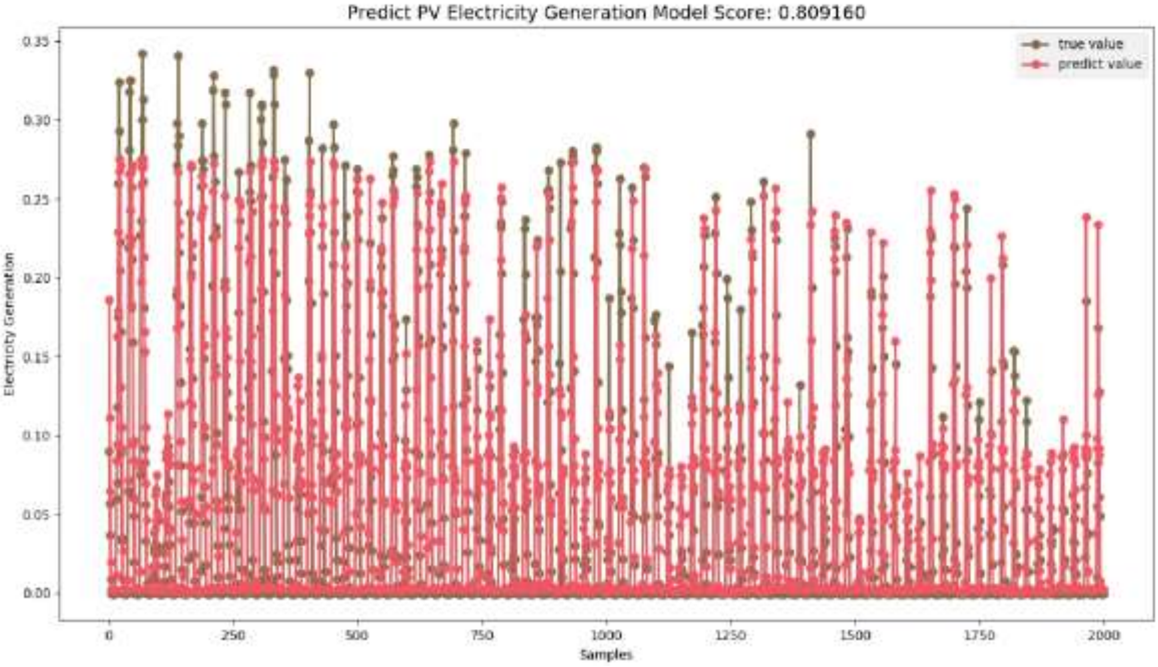


Figure 4:True vs predicted graph for MLP

The plot below shows that accuracy and loss plot for LSTM network. By observing the loss graph training loss is continuously decreasing and the training loss is also decreasing. While coming to the accuracy plot it is increasing and the accuracy obtained was 92.44 percent.

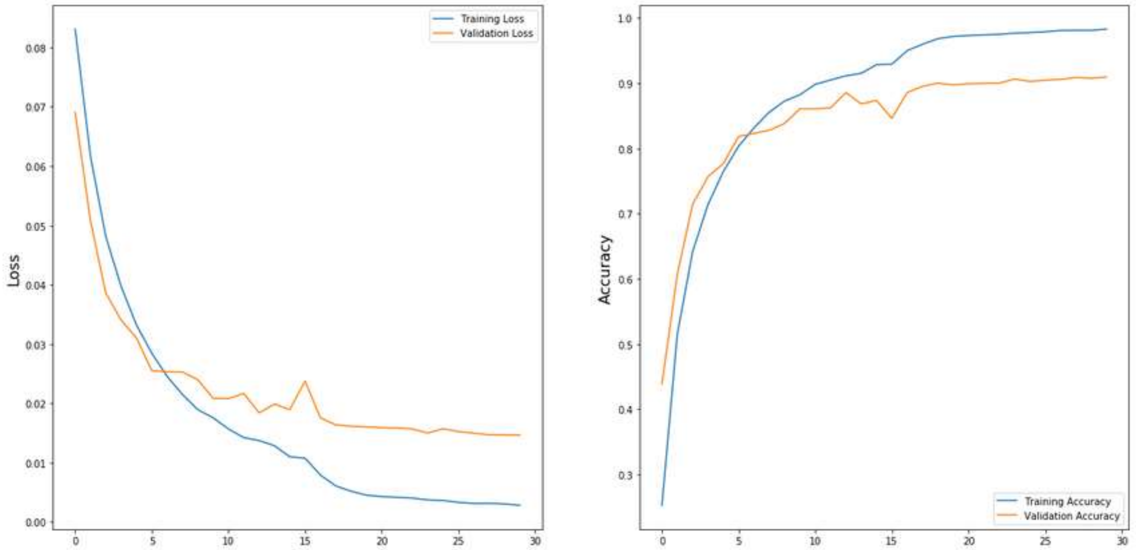


Figure 5:Accuracy and loss plot LSTM

For the LSTM model the line graph shows the true and predicted plots and the difference between them is the error by observing graph there is a less gap between them. In 4-10 hors gap power generation is increasing high and then decreasing the same thing happens on 33-45 hours range. The final accuracy score for a model obtained was 92.44% percentage for LSTM model.

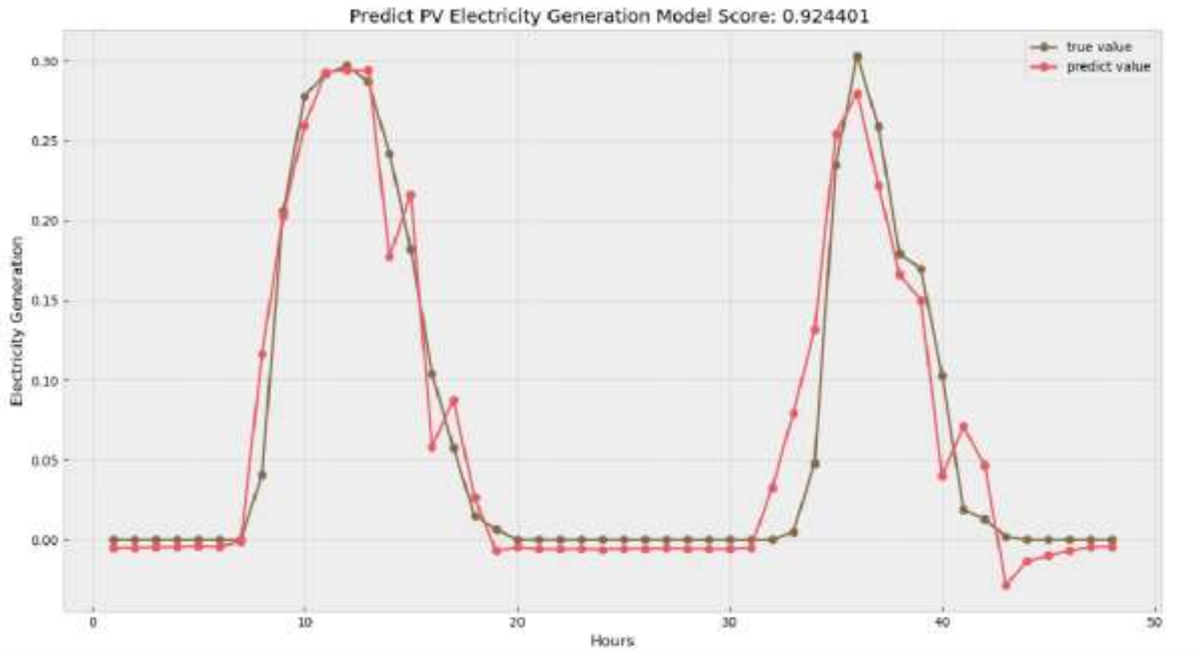


Figure 6:True vs predicted for LSTM

## 5. Conclusion

Based on the weather and other factors, this research advocated predicting the power production of solar panels. Here, the prediction is made using LSTM and MLP modes, and it is shown that solar panel characteristics have attained the best accuracy of almost 92 percent across all models. Top linked traits and the weather have both produced positive results. However, it was also discovered that, in comparison to two other studies, the meteorological aspects did not provide favourable outcomes. Finally, e produced the LSTM model, which had an accuracy of about 92 percent, as the best model. The purpose of our article is to identify useful characteristics for predicting the power output of solar panels. When there are no obstructions in the way of the sunlight as it reflects on the solar panels, energy may be collected properly. However, there are several occasions when environmental factors like weather or air pollution prevent sunlight from reflecting directly on the solar panels. As a result, these environmental factors also affect where to deploy solar panels. Additionally, depending on predictions of power generation, owners of solar panels may determine when to clean their solar panels. Users may be informed, for instance, that future electricity output from solar panels will decrease by the prediction system. due to the possibility of snow, rain, or air pollution factors covering the solar panels. Generally speaking, anticipating the power production of solar panels can have a number of advantages, including modifying installation, monitoring operation, and developing future business plans. To further improve model accuracy, environmental factors like wind, temperature, and others can be included. Consider deleting outliers from the power output data points since irregular solar panel functioning may lead to the potential of outliers; doing so may boost forecast accuracy even more. In the future, deep learning models like LSTM may be used to further improve the accuracy of power output prediction.

## References

[1]. I. Jebli, F. Z. Belouadha, M. I. Kabbaj, and A. Tilioua, "Prediction of solar energy guided by Pearson correlation using machine learning," *Energy*, vol. 224, article 120109, 2021.

[2]. G. Narvaez, L. F. Giraldo, M. Bressan, and A. Pantoja, "Machine learning for site-adaptation and solar radiation forecasting," *Renewable Energy*, vol. 167, pp. 333–342, 2021.

[3]. S. Park, Y. Kim, N. J. Ferrier, S. M. Collis, R. Sankaran, and P. H. Beckman, "Prediction of solar irradiance and photovoltaic solar energy product based on cloud coverage estimation using machine learning methods," *Atmosphere*, vol. 12, no. 3, p. 395, 2021.

[4]. K. Mahmud, S. Azam, A. Karim, S. Zobaed, B. Shanmugam, and D. Mathur, “Machine learning based PV power generation forecasting in Alice Springs,” *IEEE Access*, vol. 9, pp. 46117–46128, 2021.

[5]. A. Nespoli, A. Niccolai, E. Ogliari, G. Perego, E. Collino, and D. Ronzio, “Machine learning techniques for solar irradiation nowcasting: cloud type classification forecast through satellite data and imagery,” *Applied Energy*, vol. 305, article 117834, 2022.

[6]. F. Rodríguez, F. Martín, L. Fontán, and A. Galarza, “Ensemble of machine learning and spatiotemporal parameters to forecast very short term solar irradiation to compute photovoltaic generators output power,” *Energy*, vol. 229, article 120647, 2021.

[7]. H. Musbah, H. H. Aly, and T. A. Little, “Energy management of hybrid energy system sources based on machine learning classification algorithms,” *Electric Power Systems Research*, vol. 199, article 107436, 2021.

[8]. U. Singh, M. Rizwan, M. Alaraj, and I. Alsaidan, “A machine learning-based gradient boosting regression approach for wind power production forecasting: a step towards smart grid environments,” *Energies*, vol. 14, no. 16, article 5196, 2021.

[9]. H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng, “A review of deep learning for renewable energy forecasting,” *Energy Conversion and Management*, vol. 198, article 111799, 2019.

[10]. J. Ferrero Bermejo, J. F. Gomez Fernandez, F. Olivencia Polo, and A. Crespo Márquez, “A review of the use of artificial neural network models for energy and reliability prediction A study of the solar PV hydraulic and wind energy sources,” *Applied Sciences*, vol. 9, no. 9, article 1844, 2019.

[11]. A. Mosavi, M. Salimi, S. Faizollahzadeh Ardabili, T. Rabczuk, S. Shamshirband, and A. R. Varkonyi-Koczy, “State of the art of machine learning models in energy systems, a systematic review,” *Energies*, vol. 12, no. 7, article 1301, 2019.

[12]. “NN - Multi-layer Perceptron Regressor (MLPRegressor) - Michael Fuchs Python,” *NN - Multi-layer Perceptron Regressor (MLPRegressor) - Michael Fuchs Python*, Feb. 10, 2021. https://michael-fuchs-python.netlify.app/2021/02/10/nn-multi-layer-perceptron-regressor-mlpregressor/ (accessed Jul. 16, 2022).

[13]. A. Zendehboudi, M. A. Baseer, and R. Saidur, “Application of support vector machine models for forecasting solar and wind energy resources: a review,” *Journal of Cleaner Production*, vol. 199, pp. 272–285, 2018.

[14]. U. K. Das, K. S. Tey, M. Seyedmahmoudian et al., “Forecasting of photovoltaic power generation and model optimization: a review,” *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 912–928, 2018.

[15]. C. Voyant, G. Notton, S. Kalogirou et al., “Machine learning methods for solar radiation forecasting: a review,” *Renewable Energy*, vol. 105, pp. 569–582, 2017.