***“A Novel Approach to develop Energy Prediction Model for Large Industries using Artificial Intelligence”***

**A Project Report**



**Bachelor of Engineering in *Computer Science Engineering***

***Submitted by: Coordinated by:***

**Kshitij Kotasthane (0827CS181105)**

**Medha Kasture (0827CS181116)**

**Manas Satpute (0827CS181111)**

**Mr. Praveen Bhanodia**

**AITR, Indore**



***Acropolis Institute of Technology & Research, Indore***

**Jan-June 21**

Table of Contents

# CHAPTER 1 . INTRODUCTION

* 1. Overview
  2. Background and Motivation
  3. Objectives
  4. Challenges of the Project
  5. Team Organization

# CHAPTER 2 . REVIEW OF LITERATURE

* 1. Preliminary Investigation
     1. Time Series Forecasting
  2. ARIMA Model
  3. Prophet Model

|  |  |  |
| --- | --- | --- |
| **CHAPTER 3 .** | **PROPOSED SYSTEM** |  |
| 3.1 | The Proposal |  |
| 3.2 | Benefits of the Proposed System |  |
| 3.3 | Project Diagrams |  |
| 3.4 | Database Structure  Deployment Requirements  3.5.1 Hardware 3.5.2 Software |  |
| **CHAPTER 4 .** | **IMPLEMENTATION** |  |
| 4.1 | Technique  4.1.1 Working  4.1.2 Data Visualization |  |
| 4.2 | Sub-Optimal Solar Panels |  |
| 4.3 | Forecasting  4.3.1 ARIMA Model  4.3.2 Prophet Model |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# CHAPTER 5 . CONCLUSION

* 1. Conclusion
  2. Limitations of the Work
  3. Suggestions for Future Work

# BIBLIOGRAPHY

# SOURCE CODE

Introduction



The global shift towards renewable energy sources (RES) has driven the development of photovoltaic (PV) panels. The cost of producing electricity from PV panels have dropped significantly, while simultaneously increasing the conversion efficiency. More specifically, the levelized cost of large scale PV panels has decreased by 73% in the last decade. The decreased cost and increased efficiency have made PV panels a great competitive alternative as a RES in many countries.

It is likely, as more countries decide to invest more and more in RES, that the use of solar PV panels will continue to increase. This will increase the need for suitable means for forecasting solar PV energy output. While the demand for accurate and efficient forecasts of PV panels is evident, the solution is far from trivial. There are many complications that the current research within the field is handling. One evident nuisance is the inherited variation of weather, which makes accurate weather forecasting challenging.

Parallel to the increased demand of PV power forecasting solutions, the means for forecasting with the help of machine learning (ML) techniques have in recent years gained in popularity to traditional time series predictive models. Although ML techniques are now new, the improved computation capacity and higher availability of quality data have made the techniques useful for forecasting. This poses for an interesting area of research when forecasting the solar power output.

* 1. **Overview**

The project is based on analyzing the dataset for a given period of a solar power plant and then predicting the power generation in the coming days.

A Machine Learning model is made in such a way that it predicts the power generated by a single solar panel. Along with counting the prediction , it is also able to detect a faulty panel or if it needs maintenance.

* 1. **Background and Motivation**

Solar energy is an abundant and renewable for of energy which is boon to the ever increasing energy requirements of today’s world. This energy can be tapped through Solar Power Plant and then can be used for various domestic or industrial needs. Due to limited source of fossil fuels and carbon footprint related to power extraction processes from them, efficient use of solar energy is need of time.

Today, in the world of solar power, forecasting the production of solar energy for short periods (day, several days, week) does not have the well-established and tested technology and is often associated with large errors, which can be 60-65%.

* 1. **Objectives**

1. To forecast Solar power for days.
2. To help maintain the PV panels more efficiently.
3. Save the time, money and effort.
   1. **Challenges of the Project**

Unlike conventional coal or gas based power plants , solar power plants output are available during day time only and highly variable depending upon the availability of sunlight.

Power generated has to be consumed instantly in the absence of a power storage.

* 1. **Team Organization**
* **Medha Kasture :**

Along with doing preliminary investigation and understanding the limitations of the current system, I studied about the topic and its scope and surveyed various research papers related to the object detection and the technology that is to be used.

Documentation is also a part of the work done by me in this project.

* **Kshitij Kotasthane :**

I investigated and found the right technology and studied in deep about it. For the implementation of the project , I collected the object data and trained the model for it. Implementation logic for the project objective and coding of internal functionalities is also done by me.

* **Manas Satpute :**

I worked on the dataset which included analyzing the key attributes.

Cleaning of the dataset was also done by me which involved removing redundant data and the data that was not required in the Machine Learning model.Some part of the documentation is also done by me.

Review of Literature



In this chapter, general articles related to energy forecasting are demonstrated. Significant findings for time series and machine learning techniques for solar forecasting are also presented .

* 1. **Preliminary Investigation**
     1. **Time series forecasting**

Time series is a series of data points indexed (or listed or graphed) in time order. Therefore, the data is organized by relatively deterministic timestamps, and may, compared to random sample data, contain additional information that we can extract.

Time series forecasting finds wide application in data analytics. These are only some of the conceivable predictions of future trends that might be useful:

* 1. **ARIMA (Auto Regressive Integrated Moving Average)**

ARIMA is a model which is used for predicting future trends on a time series data. It is a model that forms a regression analysis.

· **AR (Autoregression):** Model that shows a changing variable that regresses on its own lagged/prior values.

· **I (Integrated):** Differencing of raw observations to allow for the time series to become stationary.

· **MA (Moving average):** Dependency between an observation and a residual error from a moving average model.

* 1. **Prophet**

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

Prophet is [open source software](https://code.facebook.com/projects/) released by Facebook’s [Core Data Science team](https://research.fb.com/category/data-science/). It is available for download on [CRAN](https://cran.r-project.org/package=prophet) and [PyPI](https://pypi.python.org/pypi/prophet/).

**Advantages of prophet:**

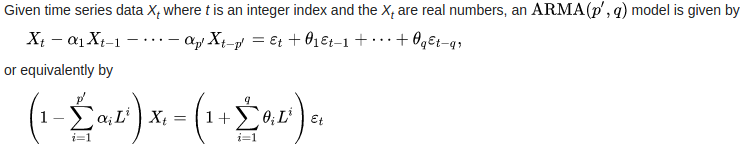
**Tunable Forecast:**

The Prophet procedure includes many possibilities for users to tweak and adjust forecasts. You can use human-interpretable parameters to improve your forecast by adding your domain knowledge.

**Fully automatic**

Get a reasonable forecast on messy data with no manual effort. Prophet is robust to outliers, missing data, and dramatic changes in your time series

**Easy to use**

Prophet doesn’t require much prior knowledge or experience of forecasting time series data since it automatically finds seasonal trends beneath the data and offers a set of ‘easy to understand’ parameters. Hence, it allows non statisticians to start using it and get reasonably good results that are often equal or sometimes even better than the ones produced by the experts.

More about Prophet at - <https://facebook.github.io/prophet/>

**Conclusion**

This chapter reviews the literature surveys that have been done during the research work. The related work that has been proposed by many researchers has been discussed. The research papers related to object detection and recognition of objects from 1985 to 2015 have been shown which discussed different methods and algorithms to identify objects.

Proposed System



* 1. **The Proposal**

The proposal is an application that uses Machine Learning Techniques to forecast the amount of solar power generation for a particular date in the near future. Our software uses a dataset of the energy generated in the previous days and trains a model for the same.

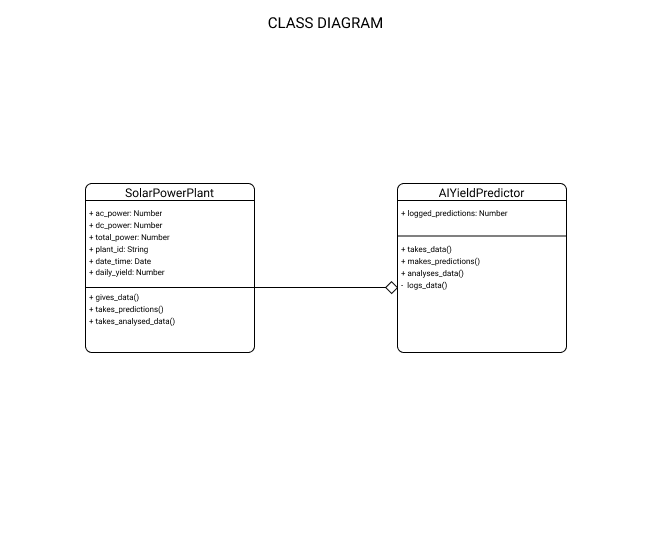
It can also monitor the performance of each solar panel to identify the need of cleaning and maintenance for a faulty or sub-optimally performing Solar panel. The proposal is for a small / large Solar power plant which can use the information to the fullest.

* 1. **Benefits of the Proposed System**

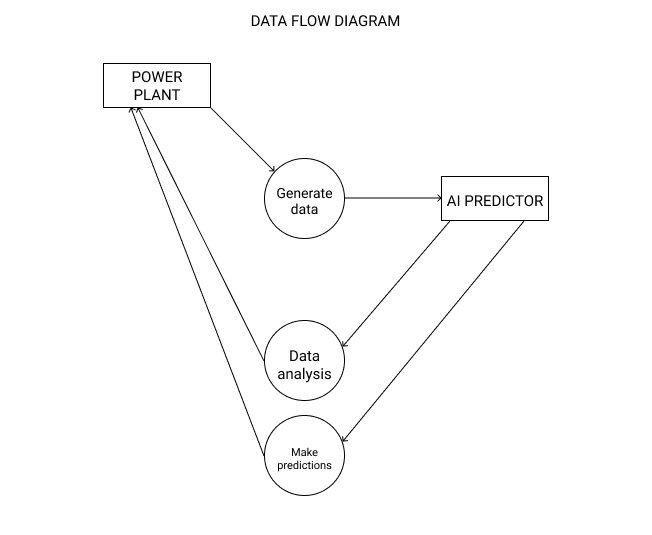
The current system had a lot of challenges that are overcome by this system:

* + - **Statistical analysis :** The complete data is used to train a Machine Learning model which can forecast the power generation.
    - **Real-Time Observation :** Unlike most, our proposal will keep updating the dataset with real time entities and enhance itself over time.
    - **Man Power :** Observing everything would be as easy as sitting on a computer, hence reducing the need for more workers.
    - **Economic :** The proposed system is economic as it will increase the energy production by identifying a sub-optimally performing solar panel.
  1. **Project Diagrams**

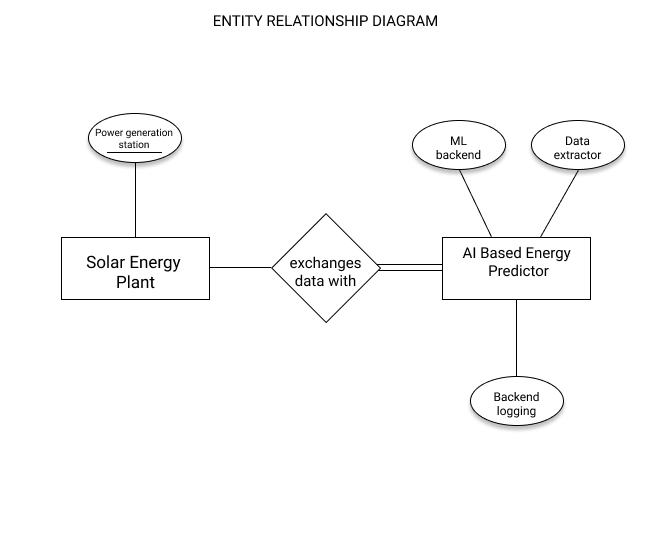
# Figure 3-1 : Activity Diagram



# Figure 3.2 : Class Diagram



# Figure 3.3 : Data Flow Diagram



# Figure 3.4 : Entity-Relationship Diagram

# 

# Figure 3.5 : Use Case Diagram

* 1. **Database Structure**

The dataset should have the following Structure -

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Function |
| Date and Time Stamp | smalldatetime | Provides the Date and time of the observation. (Which is at an interval of 15mins.) |
| Plant ID | Short Integer | Points out the Power Plant where the reading came from. |
| Source ID | Text | Identification number of the solar panel which took the reading. |
| AC / DC Power | Decimal | The amount of power than was converted from the solar energy |
| Total Yield | Decimal | The sum of both AC and DC powers |

**Table 1 : Database Structure**

* 1. **Deployment Requirements**

There are various requirements (hardware and software) to successfully deploy the system. These are mentioned below :

* + 1. **Hardware**
       - 32-bit, x86 Processing system
       - Windows 7 or later operating system
       - High processing computer system without good internet connection.
    2. **Software**
       - Google Colab
       - Python and its supported libraries

Implementation



For the problem of predicting the future energy generation, the system is designed in such a way so as to mimic the pattern previously generated by the plant.

**4.1 Technique**

**4.1.1 Working**

#### When a photon hits the surface of the photovoltaic cell, its energy is transferred to the electrons present on the silicon cell. These electrons are "excited" and begin to flow in the circuit producing electric current. A solar panel produces Direct Current energy(DC power).

#### Then, it's up to the inverter to convert it into alternating current to transport it and use it in our distribution networks. In fact, domestic and industrial buildings are designed for the transport and use of alternating current.

#### Every solar power plant consists at least of two basic components:

1. Modules- that convert sunlight into electricity
2. One or more inverters- devices that convert direct current into alternating current.

**4.1.2 Data visualization of daily yield and AC power**

The daily yield and AC power generated are two of the main attributes of our model. Their graphical representation gives us an insight to the general pattern made by them and will help in making decisions for the model.It also helps us to recognize the pattern- when most of the power is generated in a day.

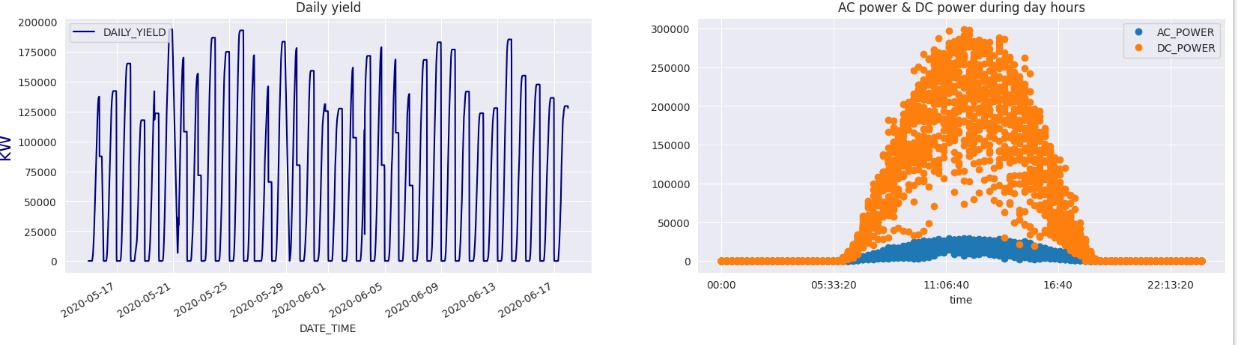


Figure 4.1 (a) Shows the power generation plotted against each day for 10 days. (b) The AC and DC power generated during a day.

**4.1.3 Data visualization of daily yield and AC power for 34 days**

This helps us to give a bigger picture and gives us a somewhat consistent pattern as is shown in Fig. 4.2.

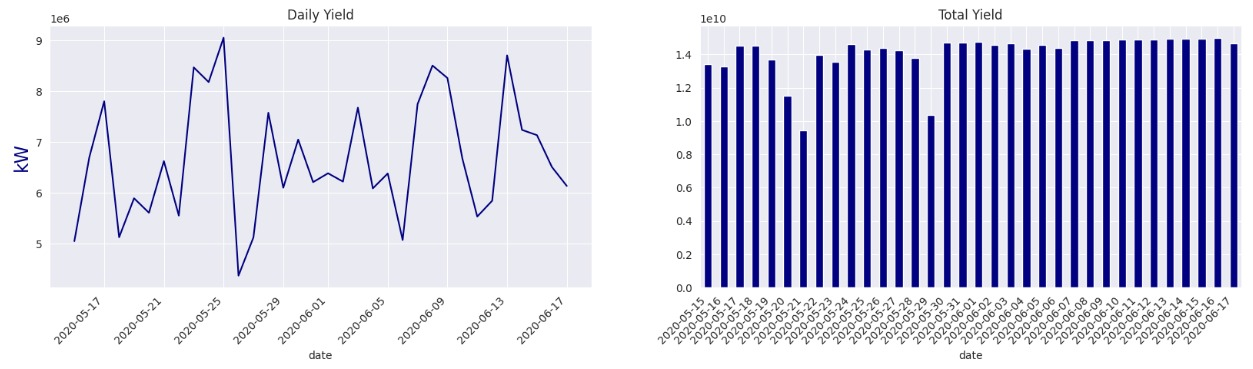


Figure 4.2 The plots for the (a) Daily yield and (b) Total yield

**4.2 To identify the sub-optimally performing equipment**

We know how all inverters works during day hours, cause we know from the quick exploratory data analysis that something went wrong with dc power generation

We can try to identify inefficient equipment by plotting how the inverters work during the day and then compare their efficiencies. **We will only focus on the DC POWER which is converted to AC POWER for domestic use.**

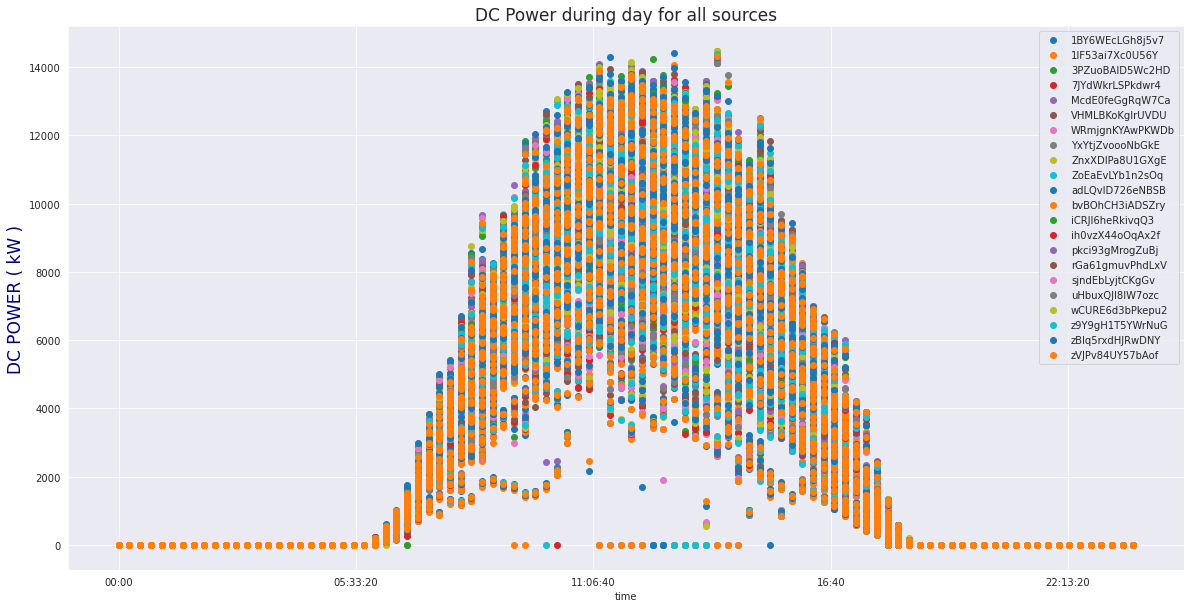
****

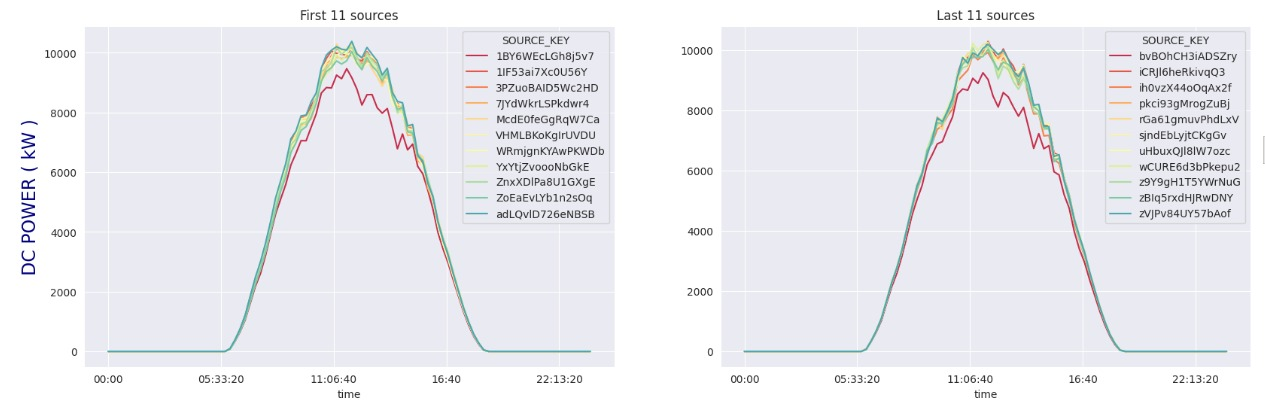
Figure 4.3 The DC power generated by all sources in the plant

Here we are interested in the solar power that is generated by the inverters which get an input from the solar module. Although this plot shows what might be wrong with the equipment, there are still two things that we can conclude from this:

* Not all of the energy that should be converted from DC POWER from solar modules to AC POWER is actually converted and there is a loss of energy in between.
* **Inverters (from solar power to electrical energy) are inefficient and these are the suboptimally performing equipment we are looking for.**
* Only about 9-10% of the overall power is converted. Still this does not mean that all inverters are faulty and contribute to losses.

We compare the DC power generated by different panels and we can see clearly what is going wrong with the power plant.

#### From Fig. 4.4 it is clear that inverters 1BY6WEcLGh8j5v7 & bvBOhCH3iADSZry are underperforming compared to other inverters, maybe these inverters require maintenance or require to be replaced. But before going into depth about underperforming inverters, let's look at which are the common problems for the entire plant, so let's see DC power generation during day hours for all 34 days.



#### Figure 4.4 The performances of the 22 energy sources used in the power plant. The curves in red highlight the sub-optimally performing equipment that our model identifies.

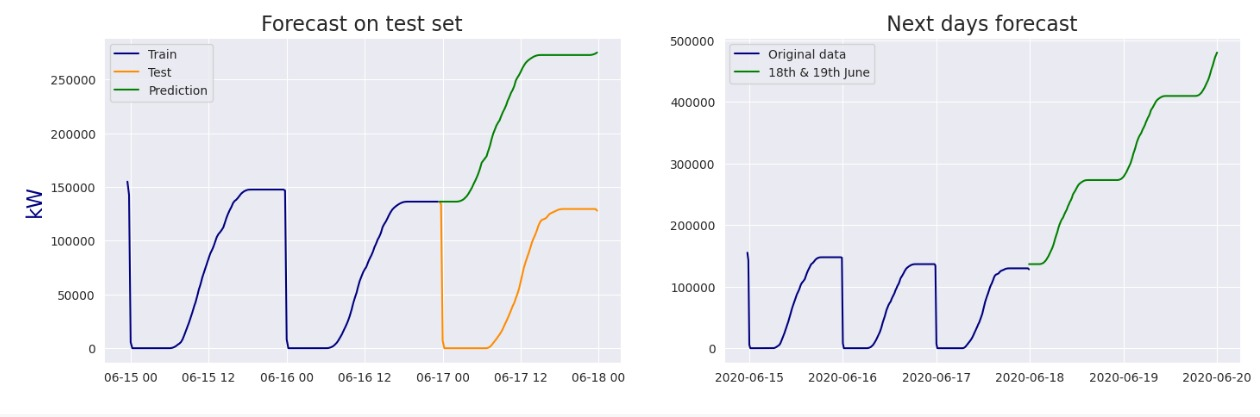
**4.3 Forecasting**

We want to predict the yield generated by plant\_1 for the next two days.

**4.3.1 ARIMA model**

The first model that we implemented was the ARIMA model. It’s a model used in statistics and econometrics to measure events that happen over a period of time. The model is used to understand past data or predict future data in a series. It’s used when a metric is recorded in regular intervals, from fractions of a second to daily, weekly or monthly periods.

This model fits all the parameters for our project as our project is also based on making predictions from analyzing past data.



#### Figure 4.5 Forecast of power generation obtained by ARIMA(SARIMAX) model on the (a)Test set and (b) Next day forecast

As is apparent in Fig. 4.5, even though the pattern is similar, it would not be able to predict the generated energy.

**4.3.2 Prophet**

The next model that we implemented was the Prophet model. Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Fig 4.6 shows the predictions made by the **Prophet model.**

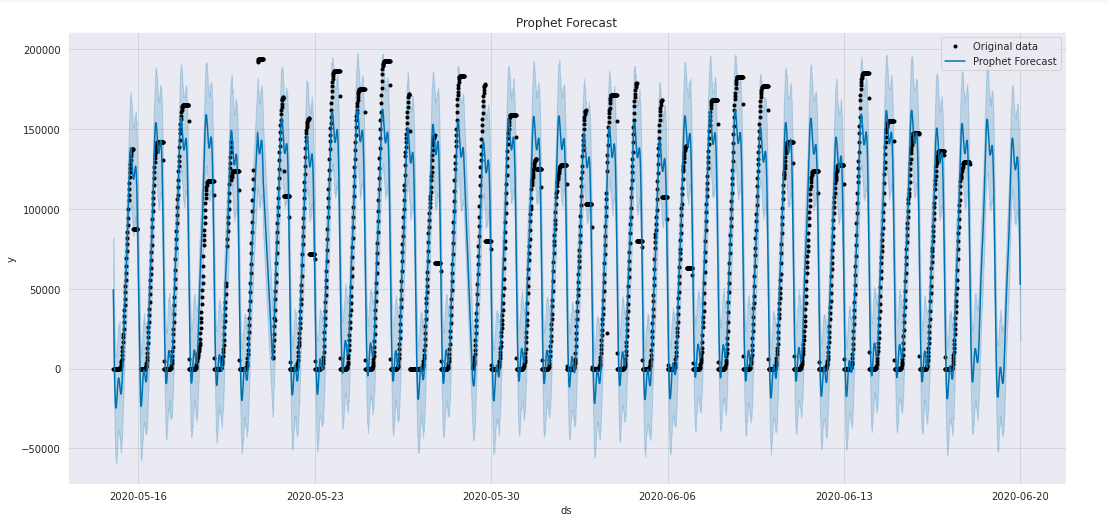


Fig 4.6 The prediction on the input data by **Prophet** (shown by the continuous curve in blue) and the original data (shown by the discrete black points). The prediction resembles the actual data closely.

Conclusion



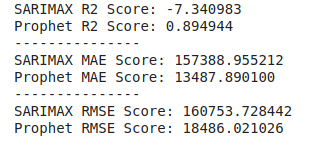
* 1. **Conclusion**

In this work, we have created an application which takes the solar weather data and uses it to forecast solar energy for days in the future. Also it is able to identify sub-optimally performing solar panels which needed cleaning or maintenance.

In section using data analysis techniques we were able to identify the equipment.

We find that employing time series models is a complex procedure due to the non-stationary energy time series. In contrast, machine learning techniques were more straightforward to implement. In particular, we find that the Prophet performs best on average across all the data. Below is a comparison of the **SARIMAX** against the **Prophet** model based upon three metrics, namely their,

1. R2 score, the proportion of the variance
2. MAE score, the average magnitude of the error
3. RMSE (root mean squared error)



We find that with a *higher R2 score (variation proportion)* and *lower MAE and RMSE (error measures)* the ***Prophet model performs much better than the SARIMAX*** to predict solar energy generation rates.

* 1. **Limitations of the Work**
     + An issue with the weather data is that the produced forecast is not for the exact coordinates of the PV installation. This is not ideal, however, it is likely that the impact is limited as the weather usually is similar in nearby areas.
     + Error Metrics - As observations in the morning and afternoon generally have lower output on average, many of the absolute prediction errors during those times are lower compared to the peak hours, resulting in a boosted RMSE as an average is computed.
     + Methodology - Regarding methodology, there are aspects to consider to improve the results and there is still a lot of scope for the improvement of the models.
  2. **Recommendations for Future Work**
     + One way to Enhance the project is to implement different models for different seasons. Splitting the year into a winter and a summer season, and then training the models independently would likely improve the results.
     + Increase the number of Platforms supported - It would be more convenient for the User if there was a mobile application for the same.
     + Instead of using the dataset we can directly deploy sensors to get the information and create a network of the whole power plant which can be observed from a single device.

**Bibliography**

1. IRENA. Renewable power generation costs in 2017. Technical report, International Renewable Energy Agency, Abu Dhabi, January 2018
2. Jose R. Andrade and Ricardo J. Bessa. Improving renewable energy forecasting with a grid of numerical weather predictions. IEEE Transactions on Sustainable Energy, 8(4):1571–1580, October 2017.
3. Rich H. Inman, Hugo T.C. Pedro, and Carlos F.M. Coimbra. Solar forecasting methods for renewable energy integration. Progress in Energy and Combustion Science, 39(6):535 – 576, 2013.
4. J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F.J. Martinez-de Pison, and F. Antonanzas-Torres. Review of photovoltaic power forecasting. Solar Energy, 136:78–111, October 2016.
5. V Kostylev, A Pavlovski, et al. Solar power forecasting performance–towards industry standards. In 1st international workshop on the integration of solar power into power systems, Aarhus, Denmark, 2011.
6. Tao Hong, Pierre Pinson, Shu Fan, Hamidreza Zareipour, Alberto Troccoli, and Rob J. Hyndman. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. International Journal of Forecasting, 32(3):896 – 913, 2016.
7. Gordon Reikard. Predicting solar radiation at high resolutions: A comparison of time series forecasts. Solar Energy, 83(3):342 – 349, 2009.
8. Peder Bacher, Henrik Madsen, and Henrik Aalborg Nielsen. Online short-term solar power forecasting. Solar Energy, 83(10):1772 – 1783, 2009.
9. Hugo T.C. Pedro and Carlos F.M. Coimbra. Assessment of forecasting techniques for solar power production with no exogenous inputs. Solar Energy, 86(7):2017 – 2028, 2012.
10. Federica Davò, Stefano Alessandrini, Simone Sperati, Luca Delle Monache, Davide Airoldi, and Maria T. Vespucci.

**Source Code**



* 1. **Daily Yield as AC/DC Power**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('darkgrid')

import warnings

import datetime as dt

import matplotlib.dates as mdates

from google.colab import files

import io

warnings.filterwarnings('ignore')

uploaded = files.upload()

gen\_1=pd.read\_csv(io.BytesIO(uploaded['Plant\_1\_Generation\_Data.csv']))

gen\_1.drop('PLANT\_ID',1,inplace=True)

sens\_1= pd.read\_csv(io.BytesIO(uploaded['Plant\_1\_Weather\_Sensor\_Data.csv']))

sens\_1.drop('PLANT\_ID',1,inplace=True)

#format datetime

gen\_1['DATE\_TIME']= pd.to\_datetime(gen\_1['DATE\_TIME'],format='%d-%m-%Y %H:%M')

sens\_1['DATE\_TIME']= pd.to\_datetime(sens\_1['DATE\_TIME'],format='%Y-%m-%d %H:%M:%S')

df\_gen=gen\_1.groupby('DATE\_TIME').sum().reset\_index()

df\_gen['time']=df\_gen['DATE\_TIME'].dt.time

fig,ax = plt.subplots(ncols=2,nrows=1,dpi=100,figsize=(20,5))

# daily yield plot

df\_gen.plot(x='DATE\_TIME',y='DAILY\_YIELD',color='navy',ax=ax[0])

# AC & DC power plot

df\_gen.set\_index('time').drop('DATE\_TIME',1)[['AC\_POWER','DC\_POWER']].plot(style='o',ax=ax[1])

ax[0].set\_title('Daily yield',)

ax[1].set\_title('AC power & DC power during day hours')

ax[0].set\_ylabel('kW',color='navy',fontsize=17)

plt.show()

daily\_gen=df\_gen.copy()

daily\_gen['date']=daily\_gen['DATE\_TIME'].dt.date

daily\_gen=daily\_gen.groupby('date').sum()

fig,ax= plt.subplots(ncols=2,dpi=100,figsize=(20,5))

daily\_gen['DAILY\_YIELD'].plot(ax=ax[0],color='navy')

daily\_gen['TOTAL\_YIELD'].plot(kind='bar',ax=ax[1],color='navy')

fig.autofmt\_xdate(rotation=45)

ax[0].set\_title('Daily Yield')

ax[1].set\_title('Total Yield')

ax[0].set\_ylabel('kW',color='navy',fontsize=17)

plt.show()

* 1. **Faulty or Sub-Optimally Performing Panels**

losses=gen\_1.copy()

losses['day']=losses['DATE\_TIME'].dt.date

losses=losses.groupby('day').sum()

losses['losses']=losses['AC\_POWER']/losses['DC\_POWER']\*100

losses['losses'].plot(style='o--',figsize=(17,5),label='Real Power')

plt.title('% of DC power converted in AC power',size=17)

plt.ylabel('DC power converted (%)',fontsize=14,color='red')

plt.axhline(losses['losses'].mean(),linestyle='--',color='gray',label='mean')

plt.legend()

plt.show()

dc\_gen=gen\_1.copy()

dc\_gen['time']=dc\_gen['DATE\_TIME'].dt.time

dc\_gen=dc\_gen.groupby(['time','SOURCE\_KEY'])['DC\_POWER'].mean().unstack()

cmap = sns.color\_palette("Spectral", n\_colors=12)

fig,ax=plt.subplots(ncols=2,nrows=1,dpi=100,figsize=(20,6))

dc\_gen.iloc[:,0:11].plot(ax=ax[0],color=cmap)

dc\_gen.iloc[:,11:22].plot(ax=ax[1],color=cmap)

ax[0].set\_title('First 11 sources')

ax[0].set\_ylabel('DC POWER ( kW )',fontsize=17,color='navy')

ax[1].set\_title('Last 11 sources')

plt.show()

* 1. **Forecast**

from pandas.tseries.offsets import DateOffset

! pip install pmdarima

from pmdarima.arima import auto\_arima

from statsmodels.tsa.stattools import adfuller

pred\_gen=gen\_1.copy()

pred\_gen=pred\_gen.groupby('DATE\_TIME').sum()

pred\_gen=pred\_gen['DAILY\_YIELD'][-288:].reset\_index()

pred\_gen.set\_index('DATE\_TIME',inplace=True)

pred\_gen.head()

train=pred\_gen[:192]

test=pred\_gen[-96:]

plt.figure(figsize=(15,5))

plt.plot(train,label='Train',color='navy')

plt.plot(test,label='Test',color='darkorange')

plt.title('Last 4 days of daily yield',fontsize=17)

plt.legend()

plt.show()

* 1. **ARIMA Model**

arima\_model = auto\_arima(train,

start\_p=0,d=1,start\_q=0,

max\_p=4,max\_d=4,max\_q=4,

start\_P=0,D=1,start\_Q=0,

max\_P=1,max\_D=1,max\_Q=1,m=96,

seasonal=True,

error\_action='warn',trace=True,

supress\_warning=True,stepwise=True,

random\_state=20,n\_fits=1)

future\_dates = [test.index[-1] + DateOffset(minutes=x) for x in range(0,2910,15) ]

prediction=pd.DataFrame(arima\_model.predict(n\_periods=96),index=test.index)

prediction.columns=['predicted\_yield']

fig,ax= plt.subplots(ncols=2,nrows=1,dpi=100,figsize=(17,5))

ax[0].plot(train,label='Train',color='navy')

ax[0].plot(test,label='Test',color='darkorange')

ax[0].plot(prediction,label='Prediction',color='green')

ax[0].legend()

ax[0].set\_title('Forecast on test set',size=17)

ax[0].set\_ylabel('kW',color='navy',fontsize=17)

f\_prediction=pd.DataFrame(arima\_model.predict(n\_periods=194),index=future\_dates)

f\_prediction.columns=['predicted\_yield']

ax[1].plot(pred\_gen,label='Original data',color='navy')

ax[1].plot(f\_prediction,label='18th & 19th June',color='green')

ax[1].legend()

ax[1].set\_title('Next days forecast',size=17)

plt.show()