**People's Democratic Republic of Algeria**
**Ministry of Higher Education and Scientific Research**

**University M'Hamed BOUGARA – Boumerdes**

**Institute of Electrical and Electronic Engineering**

**Department of Electronics**

Final Year Project Report Presented in Partial Fulfilment of the Requirements for the Degree of

# MASTER

In **Electronics**

Option: **Control Engineering**

Title:

# PV Power Forecasting using Machine Learning techniques

Presented by:

- **CHERCHARI Abdelmalek**
- **BOUROUIS Ahmed**

Supervisor:

**Prof. Aissa KHELDOUN**

Registration Number: 2020/2021

# ABSTRACT

Due to the overwhelming challenge of catching up with the increasing demand of energy and the pressing need to greenify the energy sector to face the sensitive topics of climate changes and global warming, the importance of renewable energy sources experienced an impressive augment that is expected to continue. Hence Solar photovoltaic plants are widely integrated into most countries worldwide. either via grid-connection or stand-alone networks, as a result, forecasting the output power of solar systems, this constitutes the main challenge towards ensuring large-scale and seamless integration of photovoltaic systems to improve the accuracy of energy yield forecasts. However Photovoltaic (PV) power generation is prone to fluctuations and it is affected by different weather conditions. In this case, accurate forecasting provides the grid operators and power system designers with significant information to manage the power of demand and supply. This project aims to analyze and compare various machine learning based forecasting methods in terms of characteristics and performance. This comparative study of the models is done through error analysis. The accuracy is evaluated using historical weather data. In addition, this dissertation investigates the assessment of these models based on some well-known metrics. The obtained results show that some forecasting models for PV systems are more effective than others.

# *DEDICATION*

*This dissertation is dedicated to.*

*My supportive parents, my loving mother and father, my Insightful sister, my aunts, uncles and cousins who were a source of inspiration and motivation. through good times and bad. Thank you for all the unconditional love, guidance, and support that you have given me. You made me the person I am today. Thank you for everything.*

*To my friends who helped me in every way possible, Mohammed Seddik Abachi, Khiar Nadir, Merzkani Mouloud, Bousaa Mehdi, Ibrahim kherachi, Raib Ismail, Abdelghani unfortunately I can't mention all of you but that doesn't change the fact that I am grateful, thank you for your being by my side. Thank you for believing in me when I didn't believe in myself. I count myself lucky to have met you.*

***Cherchari Abdelmalek***

*I dedicate this dissertation to*

*The sake of Allah, my Creator and my Master*

*My great and beloved parents, my supportive sister and brother, Zeineb and Abderahmen, without all their encouragements, sacrifices and blessings throughout all my life stations, none of my success would be possible*

*From the bottom of my heart, I would like to express my profound gratitude for my wonderful friends who were always there for me: Sarah Cherigui, Raouf Bougherara, Hamza Belmadani, Toufik Kadous, Hammam Habib, Yasser Sadek, Yazid guarmat, Karim Omari, Oussama Belhadef, Chahid Azizi. I am blessed and fortunate to have you in my life.*

***Ahmed Bourouis***

# *ACKNOWLEDGMENT*

*In the name of Allah, the Most Beneficent and the Most Merciful, we thank Allah for all His blessing and strength that He gave us in completing this modest thesis.*

*We would like to express our deepest and sincere gratitude to our project Supervisor Professor Aissa Kheldoun for his continuous assistance, guidance and patience.*

*Our warm gratefulness goes to our internship supervisors Ms Amine Derriche and Ms Mohamed Kahlouche for their continuous guidance during the three months training at the service of Knowledge and Development of Renewable Energy KDRE at SONELGAZ.*

*We also would like to express our appreciation to Professor Abdelmalek Kouadri and Mister Lahcene Rouani for their generous help which was essential for the completion of this work.*

***Ahmed Bourouis – Abdelmalek Cherchari***

# Table of contents

# List of figures

# List of Tables:

# List of Acronyms:

| | |
|---|---|
| AC | Alternative Current |
| AI | Artificial Intelligence |
| ANN | Artificial Neural Network |
| CdTe | Cadmium Telluride |
| CIGS | Copper Indium Gallium Selenide |
| DC | Direct Current |
| DT | Decision Tree |
| EL | Ensemble Learning |
| EVS | Explained Variance Score |
| GD | Gradient Decent |
| GS | Grid Search |
| IRENA | International Renewable Energy Agency |
| LR | Linear Regression |
| LT | Low Tension |
| LSTM | Long Short-Term Memory |
| MAE | Mean Absolute Error |
| MedAE | Median Absolute Error |
| ML | Machine Learning |
| MPPT | Maximum Power Point Tracking |
| MSE | Mean Squared Error |
| NOAA | National Oceanic and Atmospheric Administration |
| NWP | Numerical Weather Prediction |
| PIAT | Pole In-Salah Adrar Timimoune |
| POA | Plane of Array |
| PR | Polynomial Regression |
| PV | Photovoltaic |
| QA | Quality Assessment |

| | |
|---|---|
| ReLU | Rectified Linear Unit |
| RIN | Reseau Interconnecte Nord |
| RIS | Reseau Interconnecte Sud |
| RNN | Recurrent Neural Network |
| RS | Random Search |
| SVM | Support Vector Machine |
| VR | Voting Regressor |

# General introduction

Global warming along with the alarming depletion of fossil fuel over the past decades have encouraged people to work hard and focus their efforts toward mitigating exhaust CO2 emissions and ensure a clean energy future, which led to a fast development of the renewable power generation techniques that rely on renewable energy sources e.g., solar, wind, hydropower, and geothermal energy, such energy sources have not only been acknowledged as novel solutions to the issues listed above but also reflect the future of energy developments. Compared to conventional energy sources, Solar energy emerges as one of the most promising source for generating power for residential, commercial, and industrial applications, especially considering the fact that the cost of PV modules is subject to a constant decrease when compared to increasing costs of energy generation from fossil fuels and other polluting energy sources, hence it becomes more practical to use renewable energy resources such as solar energy, that can convert solar irradiance into electric energy through the Photovoltaic Effect. Energy generated by PV system is directly proportional to geographical and weather conditions such as temperature, solar intensity, site-specific conditions etc. However, this variability of PV output power brings serious challenges to the operation of the power grid.

The integration of solar PV plants into power grids has received much attention due to its ability in generating electric power. Solar plants are widely used in smart grids. The implementation of large-scale grid connected solar PV plants has shown significant issues to the power networks such as system stability, reliability, electric power balance. Solar PV power forecasting has emerged as a brilliant way to address these issues. Indeed, the prediction of the amount of energy generated by renewable energy sources is imperative in terms of satisfying the energy demand and supply planning to avoid critical conditions in these systems. PV power generation forecasting can help in reducing the impact of system output uncertainty on the grid. Therefore, assists in making the system more reliable and maintaining the power quality. Hence to make use of PV power more efficiently, forecasting becomes vital.

To promote the absorption of PV power generation, multiple technologies and techniques have been developed and applied, including power flow optimization, demand response, energy storage sizing, designing, and simulating the solar PV plants by tweaking the number of PV modules, and inverter capacity, and controller types…etc. At present, PV power forecasting is one of the most economical and feasible solutions.

The goal of the study is to use several machine learning techniques to generate models that can forecast the irradiance and energy generation, since solar energy is subject to fluctuations and it is weather dependent, solar photovoltaic power forecasting is vital to ensure optimum planning and modelling of the solar photovoltaic plants, hence precise and accurate forecasting could help the grid operators and power system designers with significant information to design an optimal solar photovoltaic plant as well as operate dispatch operations. This study is based on the PV power generation data obtained from three different locations and climates for approximately one-year period observed in 5-minute interval. The data measurement locations were Cocoa in Florida, Eugene in Oregon and Golden in Colorado. We obtained the weather data from the National Renewable Energy Laboratory under the Systems Integration Subprogram, which is funded and monitored by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy.

This thesis is structured as follows. In the first chapter, we provide a brief introduction to PV energy system and potential of Photovoltaic energy as well as the importance of PV forecasting. The second chapter discusses machine learning, we will cover topics like fundamentals of machine learning, theoretical information on the models used in this study, evaluation metrics and proposed approach. The third chapter provides information on the different steps we took to process and model the data in the optimal form possible. Finally, in the last chapter we summarize the results and compare them between all models.

# CHAPTER 1:

# Review of Photovoltaic Systems

## 1.1 Introduction:

We understand the word photovoltaics as the direct conversion of sunlight into electric energy. The basic component of every photovoltaic plant is the solar cell, this consists in most cases of silicon, a semiconductor that is also used for diodes, transistors and computer chips. With the introduction of foreign atoms, a p-n junction is generated in the cell that "generates" an electrical field in the crystal. If light falls on the solar cell, then charge carriers are dissolved out of the crystal bindings and moved by the electrical field to the outer contacts. The result at the contacts of the solar cells is the creation of a voltage of approximately 0.5 V. The released current varies depending on radiation and cell area, and lies between 0 and 10 A. In order to achieve a usable voltage in the region of 20–50 V, many cells are switched together in series in a solar module. Besides this, the solar cells in the modules are mechanically protected and sealed against environmental influences (e.g., entrance of moisture).

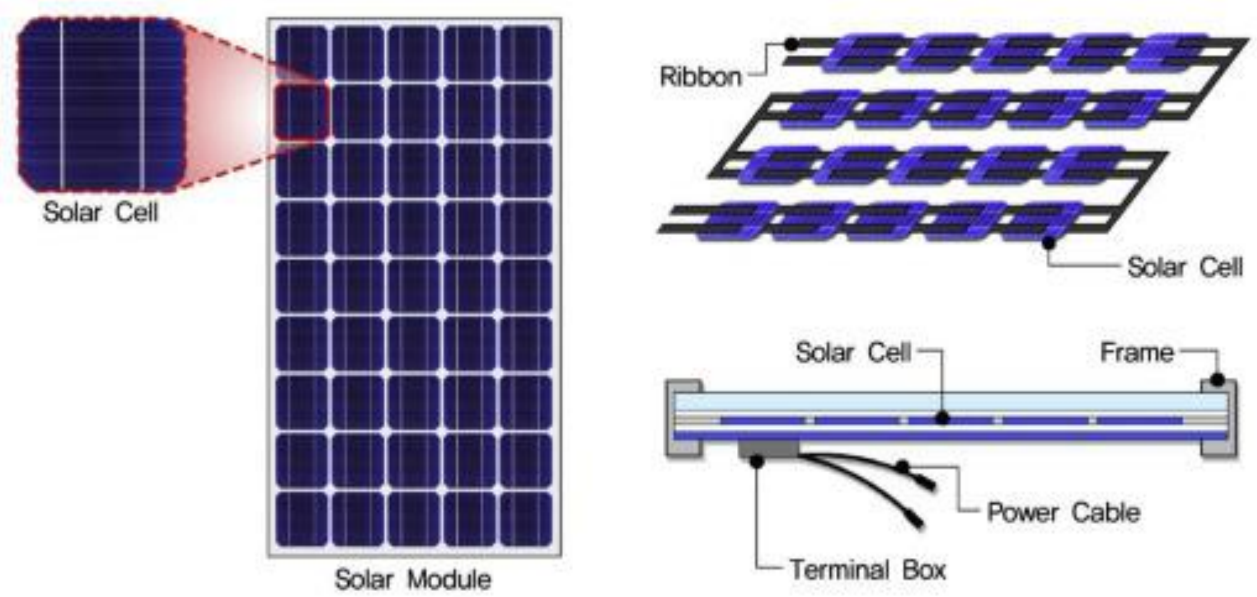


**Figure 1.1:** The solar cell and solar module as basic components of photovoltaics [1].

The typical photovoltaic plant consists of:

- **Mounting system:** basically, this refers to the mounting apparatus which fixes the solar array to the ground or rooftop. Typically constructed from steel or aluminum, these apparatuses mechanically fix the solar panels in place with a high level of precision. Some ground mounted racking systems also incorporate tracking systems which use motors and sensors to track the Sun through the sky, increasing the amount of energy generated at a higher equipment and maintenance cost.
- **Solar panel:** A solar panel consists of many solar cells with semiconductor properties encapsulated within a material to protect it from the environment. These properties enable the cell to capture light, or more specifically, the photons from the sun and convert their energy into useful electricity through a process called the “photovoltaic effect”.
- **Charge controller:** it regulates the voltage and current generated by a solar array, it also takes some of the electricity from the DC current generated by a solar array and use it so that it can properly charge the battery or bank of batteries. Since the power generated by solar panels varies with light exposure. The charge controller also protects the batteries from being overcharged and end up damaged. Controllers can be for stand- alone systems or grid tied battery backup systems.
- **Batteries:** Due to the fluctuation of power generation of photovoltaic panels, Photovoltaic systems are often connected to electricity storage units such as batteries. The purpose of installing batteries is not only to store excess electricity but also to ensure a more stable supply of power at night or under cloudy weather conditions.
- **Inverter:** An inverter is an electrical device which accepts electrical current in the form of direct current (DC) and converts it to alternating current (AC). This conversion is necessary to operate most electric devices or interface with the electrical grid. Inverters are important for almost all solar energy systems and are typically the most expensive component after the solar panels themselves. Most inverters have conversion efficiencies of 90% or higher and contain important safety features including ground fault circuit interruption and anti-islanding. These shut down the PV system when there is a loss of grid power.
- **DC-DC converter:** A DC-DC converter takes DC power at the input and transforms it to another form of DC power at the output (i.e., different voltage levels). In grid-connected

PV systems, the main purpose of this stage is to control the PV panel connected to the input side so that it operates at the MPP of the PV string. That power is processed through the DC-DC converter and delivered to the DC-link, which is also the input to the PV inverter.

- **Transformer:** A transformer is used when as a step-up/down action for the output AC voltage from the inverter is required to meet the desired level for the given operation.

Utilities may use more advanced systems for generating substantial quantities of electricity such as [2]:

- Single axis or double axis tilting systems.
- Automatic cooling and cleaning systems.
- Fuel cell, battery or other type of power storage systems.
- Transmission lines.

## 1.2 PV energy:

Global power generation from solar PV increased by 22% in 2019, to 720 TWh. With this increase, the solar PV share in global electricity generation is reached almost 3%. In 2020 global renewable energy capacity transcended earlier estimates and all previous records despite the economic slowdown that resulted from the COVID-19 pandemic. According to data released by the International Renewable Energy Agency (IRENA) the world added more than 260 gigawatts (GW) of renewable energy capacity that year, exceeding expansion in 2019 by close to 50 per cent [3].

### 1.2.1 In The World (Globally):

The PV market experienced an upward trend growth during the last decade, then rocketed starting from the year 2010 as we can see in the following figure and this despite the economic difficulties the market has had to face. PV is quickly becoming a major source of electricity in the world.

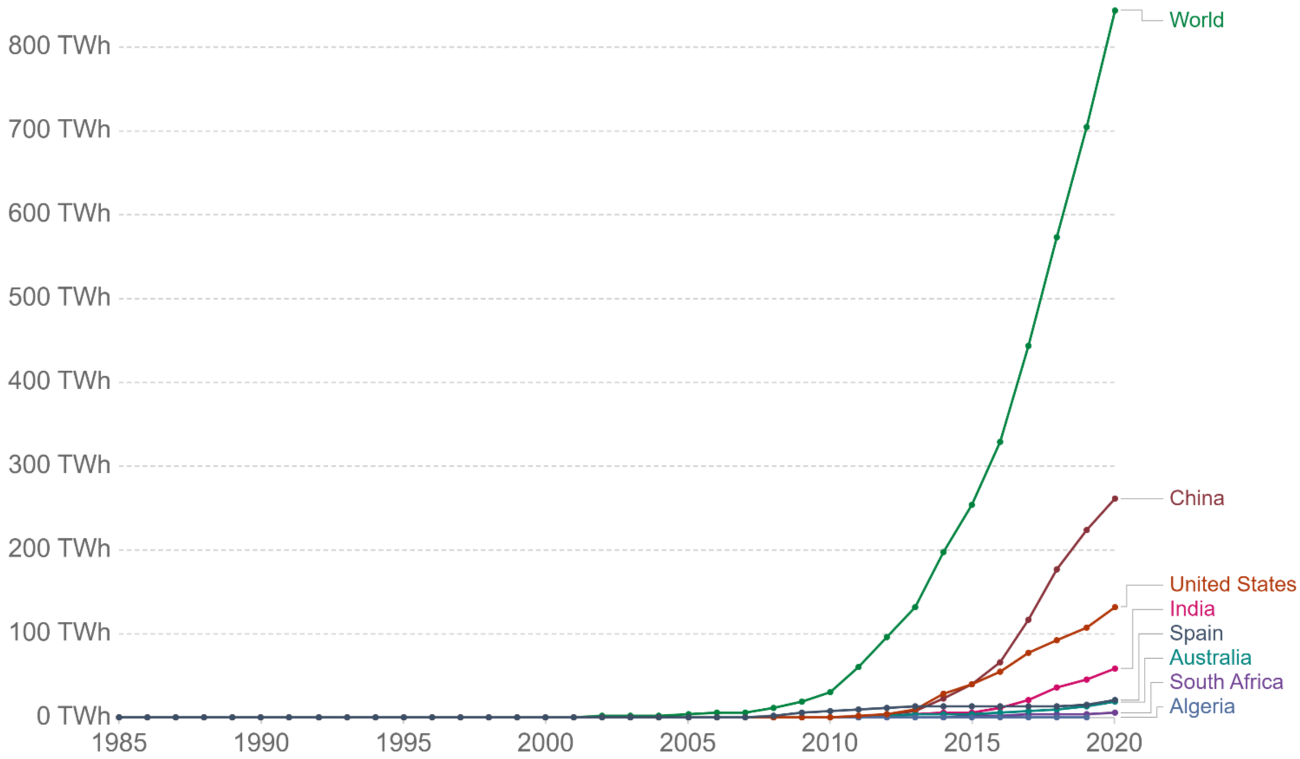


**Figure 1.2:** Solar power generation measured in terawatt-hours per year from 1985 to 2020 [4].

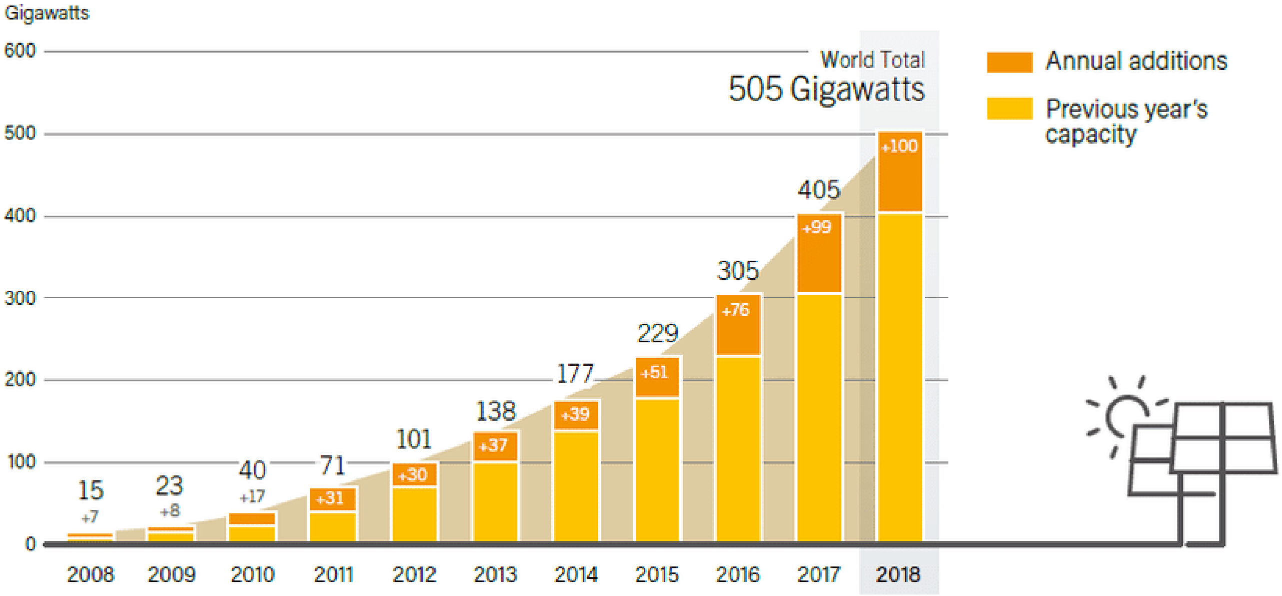


**Figure 1.3**: Solar PV Global Capacity and Annual Additions, 2008-2018 [5].

This strong development of PV Energy can be explained, on the one hand, by the raise of public awareness toward the dangers of global warming and climate change, and on the other hand, thanks to the impressive drop in the price of PV modules which was continuous during the period from 2010 up to 2020 (see below figure). This impressive decrease is mainly due to advances in research and also to module manufacturing.

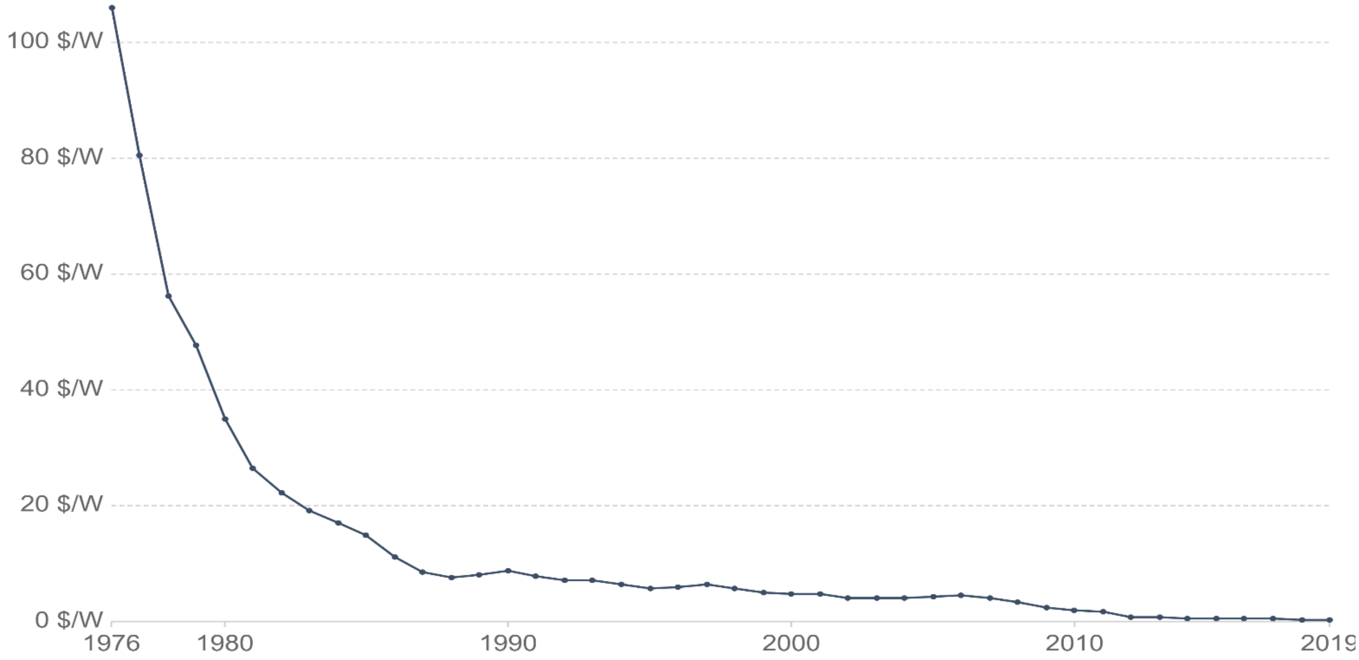


**Figure 1.4:** Global Average Price of Solar PV Modules, Measured in 2019 US dollar per Watt from 1976 to 2019 [6].

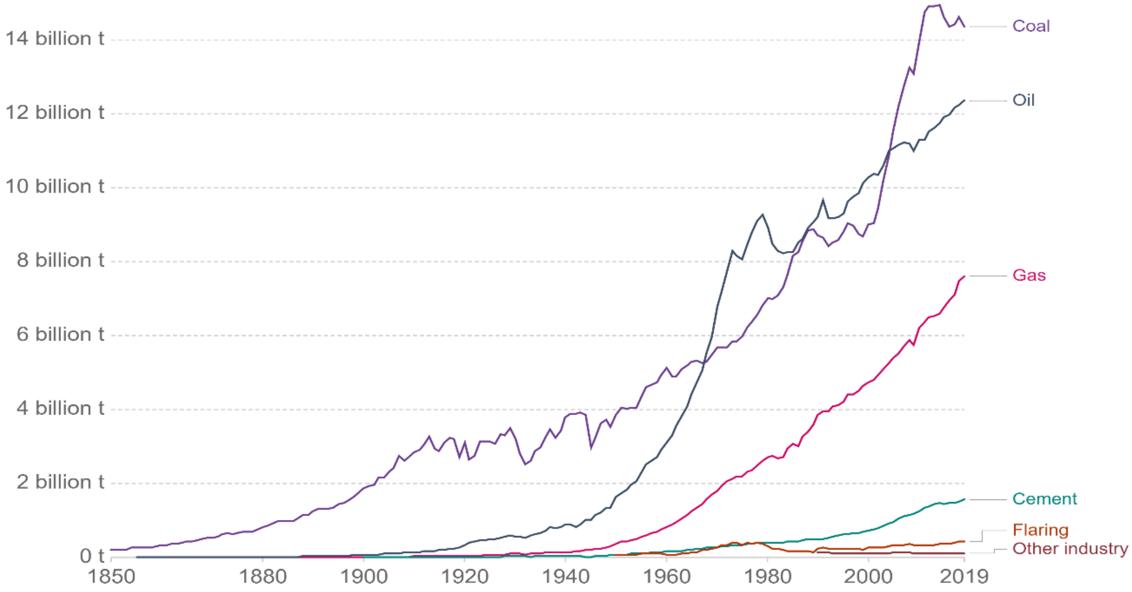


**Figure 1.5:** $CO_2$ Emissions by Fuel in the World 1850-2019 [7].

This drop in the price of modules makes PV today competitive (or even advantageous) with respect to other sources of electricity in countries where the grid is not yet well developed and / or having a high annual sunshine rate, which was unimaginable a few years ago.

### 1.2.2 In North Africa:

The global high level of solar irradiation intensity region mainly concentrated in the 10°north latitude to 35°north latitudes, and the annual solar irradiation intensity is between 1800kWh/m$^2$ to 2600kWh/m$^2$. Hence, the resource of solar energy is rich in North Africa, and the potential is quite large to build solar power generation base in the most of North Africa region countries, such as Morocco Tunisia, Algeria, Egypt [8]. In recent years, North African economy flourished and is undergoing a steady growth, hence Electricity consumption increased (as we can see in the figure below).

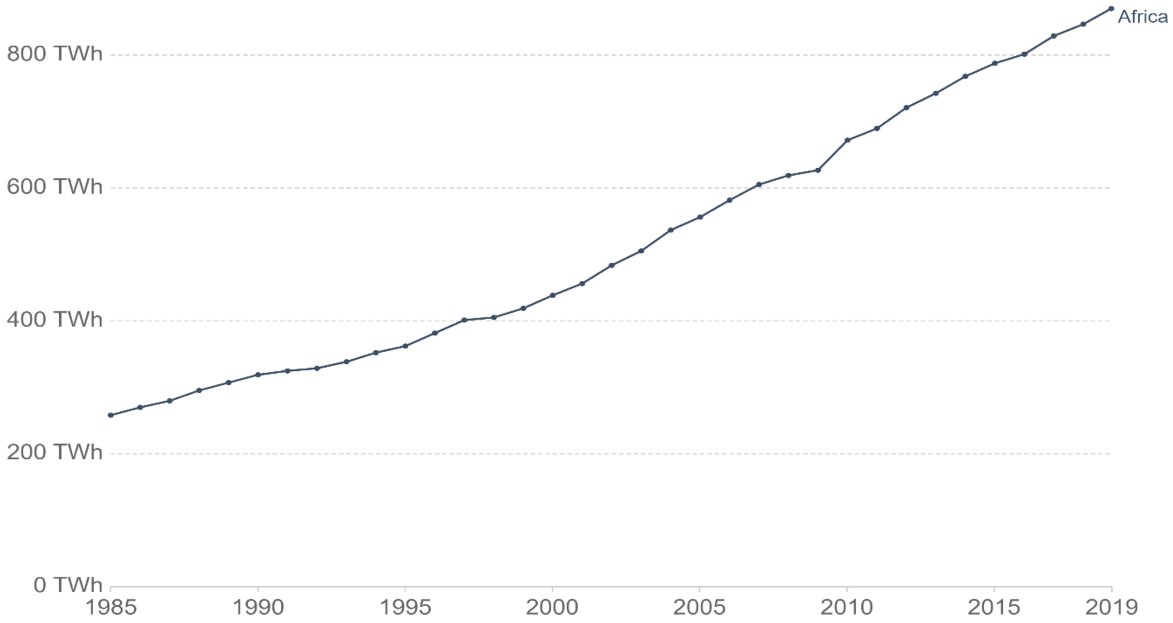


**Figure 1.6:** Electricity Consumption in Africa Measured in Terawatt-hours (TWh) 1985-2019 [9].

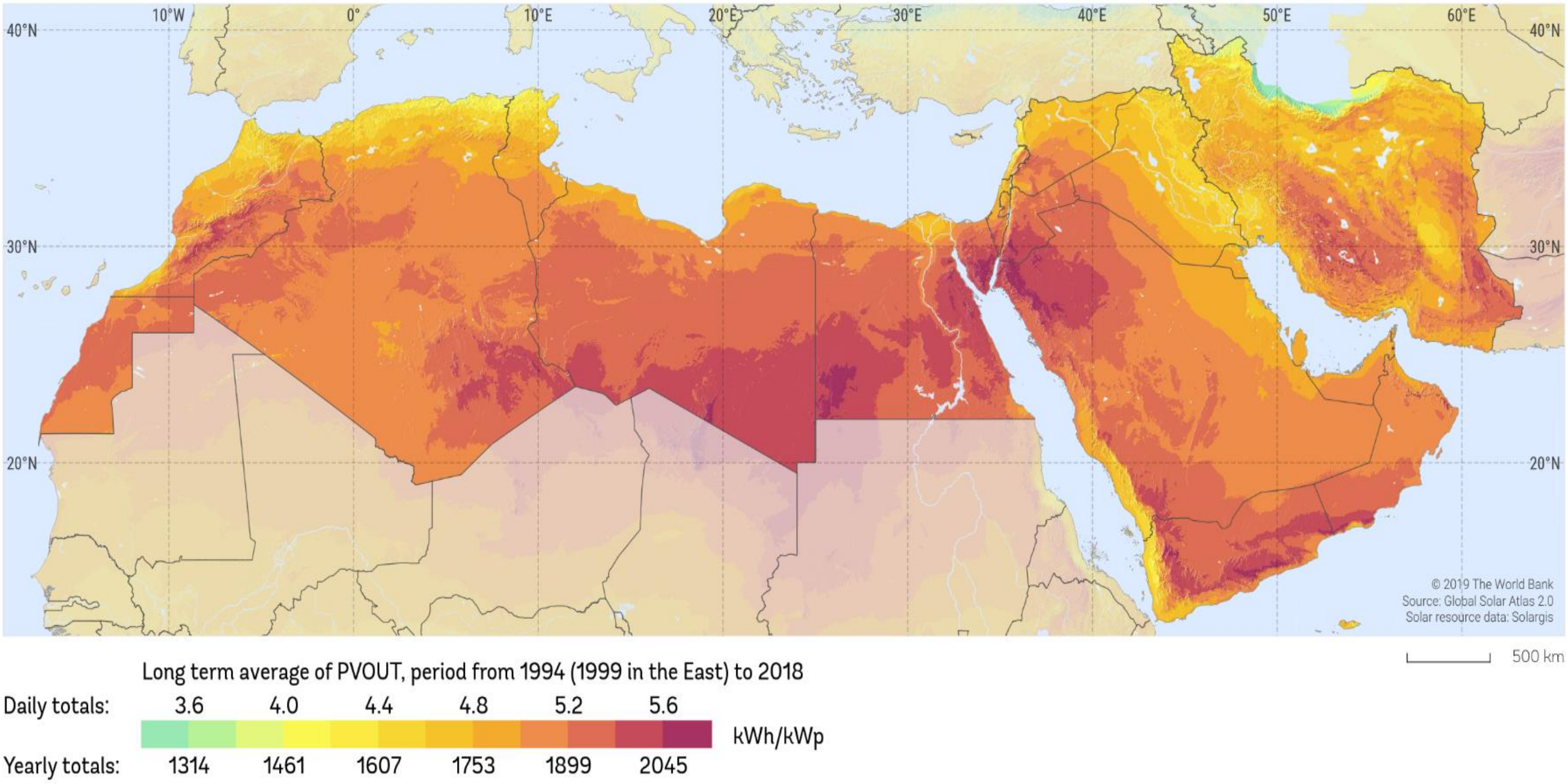

10°W
0°
10°E
20°E
30°E
40°E
50°E
60°E
40°N
30°N
20°N
© 2019 The World Bank
Source: Global Solar Atlas 2.0
Solar resource data: Solargis
500 km
Long term average of PVOUT, period from 1994 (1999 in the East) to 2018
Daily totals: 3.6 4.0 4.4 4.8 5.2 5.6
kWh/kWp
Yearly totals: 1314 1461 1607 1753 1899 2045

**Figure 1.7:** PV Potential in Middle-East-and-North-Africa [10].

Thank to this great potential it is promising to invest in PV solar energy in the region. In fact, from the below graph we can notice that north African countries started using solar PV energy as a source of electricity generation in 2009, then started seriously investing in it starting from the year 2013.

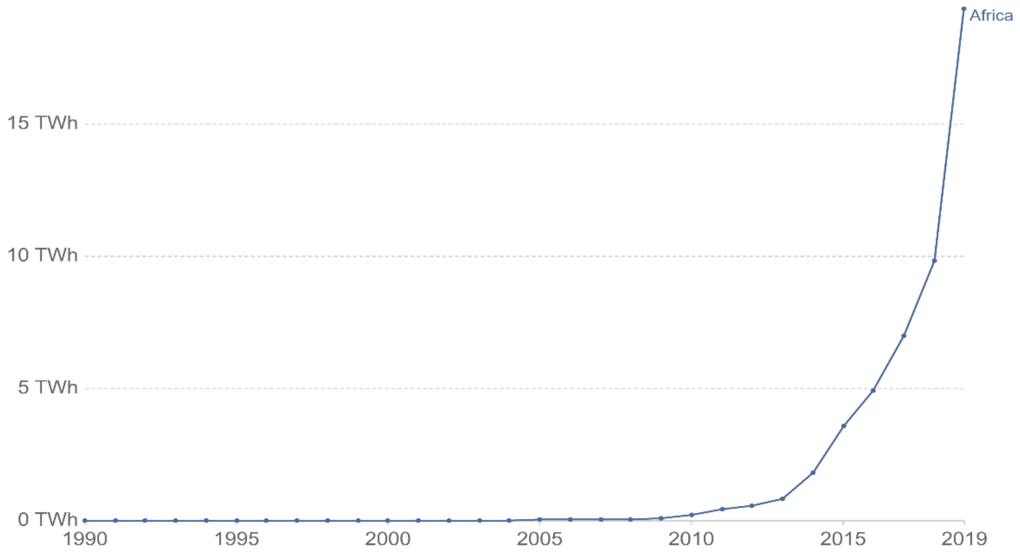


**Figure 1.8:** Renewable Solar PV Electricity Generation Africa 1990-2018 [4].

### 1.2.3 In Algeria:

The Renewable Energy and Energy Efficiency Development Plan launched with its first version in 2011 and updated in 2015, has put greater focus on the deployment of large-scale solar photovoltaic installations. Algeria has a plan to deploy 22 GW of renewable energy power generation capacity by 2030, including 13.6 GW of PV [11]. The strategic choice is motivated by the huge potential in solar energy. Such actions can be easily understood since Algeria just like many other countries experience a steady increase in its electricity consumption (see figure 1.9), the country decided to lunch several projects in order to harvest the huge potential of PV solar energy. As we can clearly see in figure 1.10, thanks to that development plan the amount of Photovoltaic energy Algeria generates started to increase from 2016 and is still increasing.

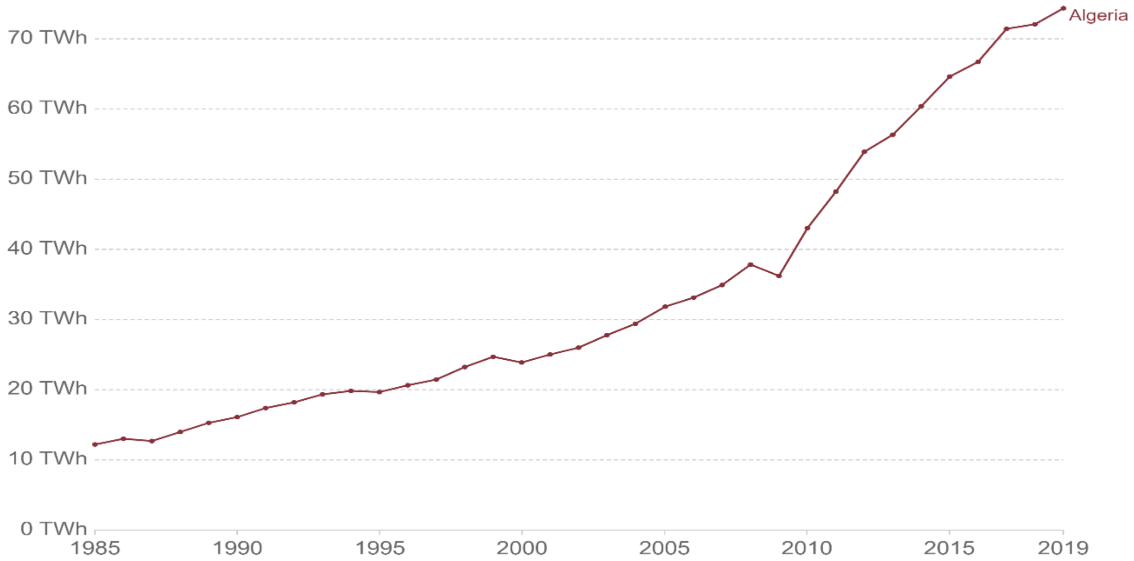


**Figure 1.9:** Electricity Consumption in Algeria Measured in Terawatt-hours (TWh) 1985-2019 [4].

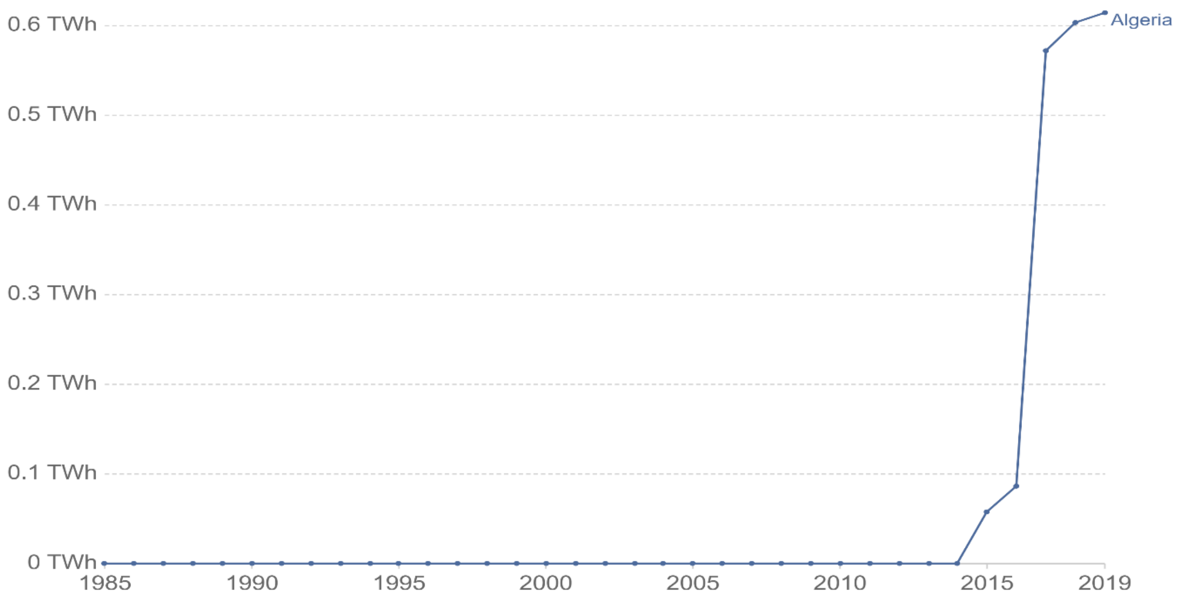


**Figure 1.10:** Solar PV Electricity Generation Algeria Measured in Terawatt-hours (TWh) 1985-2019 [4].

Today, Algeria has realized 23 PV power plants, with a total capacity of 344.1 MWp

- 1 PV station in Ghardaia with a capacity of 1.1 MWp to test available technologies and develop optimal PV architectures.
- 12 PV stations with a capacity of 265 MWp Grid-connected to the north interconnected network (RIN - Réseau Interconnecté Nord).
- 7 PV stations with a capacity of 53 MWp connected to PIAT (Pole In-Salah Adrar Timimoune).
- 3 PV station connected to the south isolated network (RIS – reseau Isolé Sud).

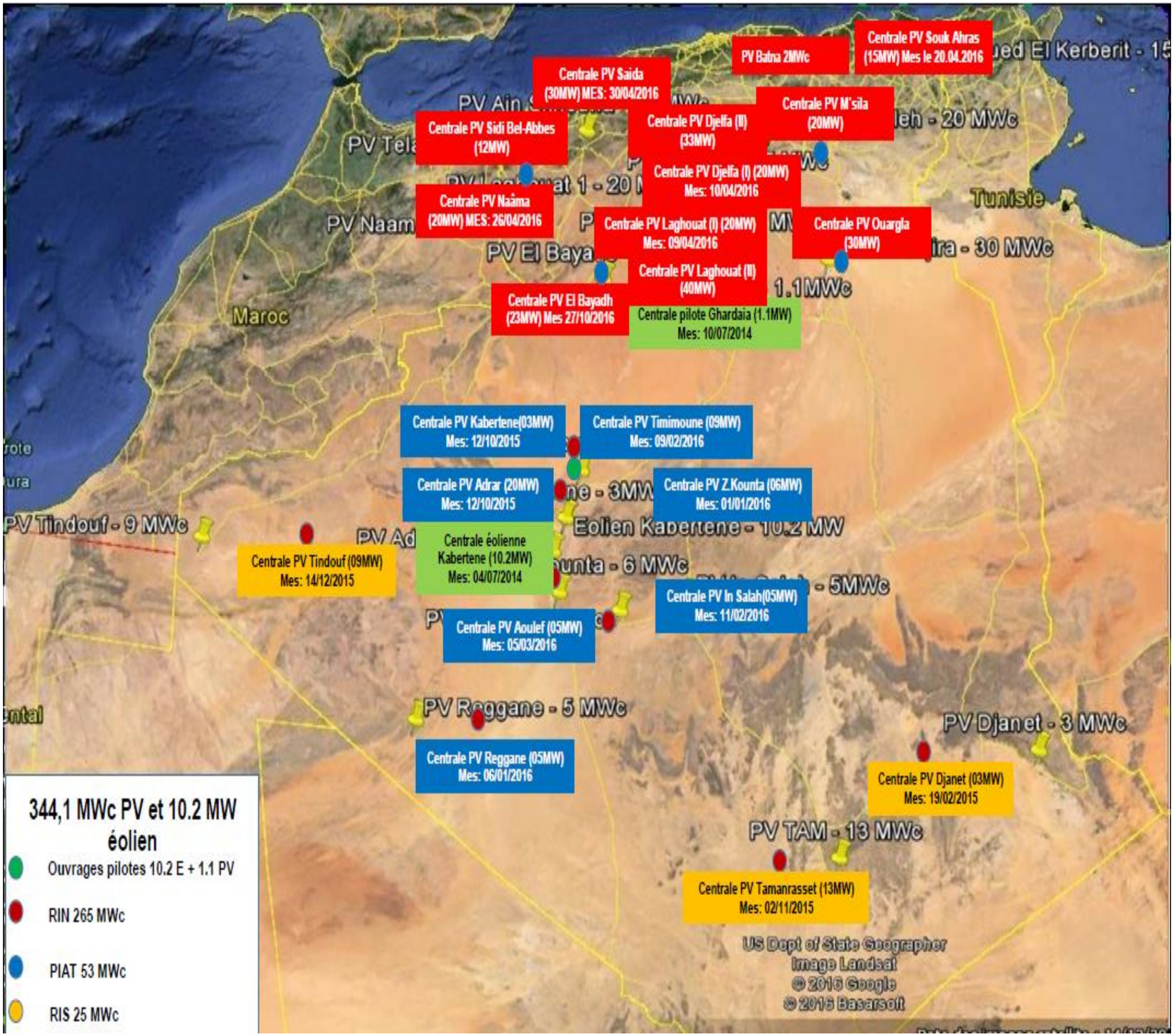

Centrale PV Souk Ahras (15MW) Mes le 20.04.2016
PV Batna 2MWc
Oued El Kerberit - 15
Centrale PV Saida (30MW) MES: 30/04/2016
Centrale PV M'sila (20MW)
Centrale PV Sidi Bel-Abbes (12MW)
Centrale PV Djelfa (II) (33MW)
PV Ain
PV Tela
Centrale PV Djelfa (I) (20MW) Mes: 10/04/2016
Centrale PV Naâma (20MW) MES: 26/04/2016
PV Naam
Tunisie
Centrale PV Laghouat (I) (20MW) Mes: 09/04/2016
Centrale PV Ouargla (30MW)
PV El Baya
Centrale PV Laghouat (II) (40MW)
1.1MWc
Centrale PV El Bayadh (23MW) Mes 27/10/2016
Centrale pilote Ghardaia (1.1MW) Mes: 10/07/2014
Maroc
Centrale PV Kabertene(03MW) Mes: 12/10/2015
Centrale PV Timimoune (09MW) Mes: 09/02/2016
Centrale PV Adrar (20MW) Mes: 12/10/2015
Centrale PV Z.Kounta (06MW) Mes: 01/01/2016
Eolien Kabertene - 10.2 MW
PV Tindouf - 9 MWc
Centrale PV Tindouf (09MW) Mes: 14/12/2015
PV Ad
Centrale éolienne Kabertene (10.2MW) Mes: 04/07/2014
unta - 6 MWc
Centrale PV In Salah(05MW) Mes: 11/02/2016
- 5MWc
Centrale PV Aoulef (05MW) Mes: 05/03/2016
PV Reggane - 5 MWc
Centrale PV Reggane (05MW) Mes: 06/01/2016
PV Djanet - 3 MWc
Centrale PV Djanet (03MW) Mes: 19/02/2015
PV TAM - 13 MWc
Centrale PV Tamanrasset (13MW) Mes: 02/11/2015
344,1 MWc PV et 10.2 MW éolien
🟢 Ouvrages pilotes 10.2 E + 1.1 PV
🔴 RIN 265 MWc
🔵 PIAT 53 MWc
🟡 RIS 25 MWc
US Dept of State Geographer
Image Landsat
© 2016 Google
© 2016 Basarsoft

**Figure 1.11:** Photovoltaic Power Potential in Algeria.

## 1.3 The different types of PV systems:

Photovoltaic power systems are broadly classified into standalone systems and grid-connected system. Both systems have their strengths and weaknesses making it really comes down to preferences and the objectives of the system, grid connected systems seem to become increasingly popular, the penetration of grid-connected PV systems will increase drastically in the near future. However, because the grid was not particularly designed for large-scale distributed generation, engineers may be concerned about the implications of variable solar generation on the power quality, its impact on the low-tension (LT) distribution grid, and the safety of its workforce.

### 1.3.1 Grid-connected PV systems:

A grid connected PV system is one where the photovoltaic panels or array are connected to the public grid through a power inverter unit allowing them to operate in parallel with the electric utility grid. This allows power to be shared between the two sides depending on the situation. It is not uncommon for the photovoltaic system to provide more power than the power required by the load. When this happens, the excess power will be fed into the grid. And vice versa.

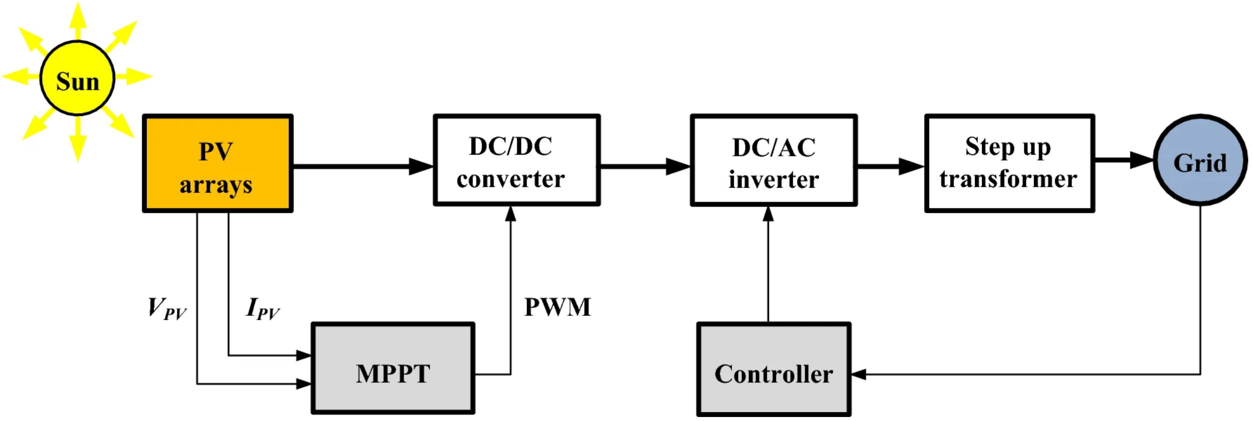


**Figure 1.12:** block diagram of basic grid connected PV system [12].

### 1.3.2 Standalone PV systems:

Standalone solar systems are self-contained fixed or portable solar PV systems that are not connected to any local utility or mains electrical grid as they are generally used in remote and rural areas. This generally means that the electrical appliances are a long way from the nearest fixed electrical supply, or were the cost of extending a power line from the local grid may be very expensive. Standalone PV systems are often connected to storage banks such as batteries in order to store the excess power generated. The batteries will then be used to keep the energy consummation and production as close as possible.

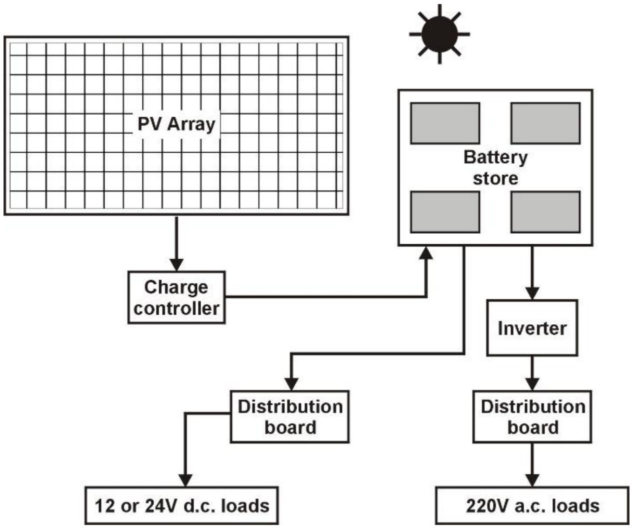


**Figure 1.13:** Block Diagram of Standalone PV System [13].

## 1.4 PV technologies and recent innovations:

There are different types of photovoltaics, some developed long ago, and others that are relatively new. Descriptions below provide a brief overview of a few well-developed PV materials.

### 1.4.1 Silicon Solar Cells:

About 90% of current solar PV deployment is based on crystalline silicon solar cells, a technology that has been commercialized for decades and is still improving. This efficient, reliable technology could achieve the needed large-scale deployment without major technological advances however it is hard to make it cheaper. Silicon solar cells can be subdivided further into:

- **Monocrystalline silicon:**

Monocrystalline silicon solar cells are probably the oldest type of solar cells. They are made from pure silicon crystal, which has continuous lattice and almost no defects. Its properties provide for high efficiency of light conversion (typical ~15%; with recent developments this efficiency can be boosted up to 22-24%). However, this type of photovoltaics has a high cost. The monocrystalline silicon cells have a typical black or iridescent blue color. The monocrystalline silicon cells are believed to be very durable and last over 25 years. However, their efficiency will gradually decrease (about 0.5% per year), so replacement of operating modules might be needed sooner. The main disadvantages of the monocrystalline silicon panels are high initial cost and mechanical vulnerability (brittle) [14].

- **Polycrystalline silicon:**

Polycrystalline cells are made by assembling multiple grains and plates of silicon crystals into thin wafers. Smaller pieces of silicon are easier and cheaper to produce, so the manufacturing cost of this type of PV is less than that of monocrystalline silicon cells. The polycrystalline cells are slightly less efficient (~12%). These cells can be recognized by their mosaic-like appearance. Polycrystalline cells are also very durable and may have a service life of more than 25 years. The disadvantages of this type of PV technology are mechanical brittleness and not very high efficiency of conversion [14].

### 1.4.2 Amorphous silicon (Thin-film):

Thin film photovoltaic cells are produced by depositing silicon film onto substrate glass. In this process, less silicon is used for manufacturing compared to mono- or polycrystalline cells, but this economy comes at the expense of conversion efficiency. Thin-film PV have efficiency of ~6%. One way to improve the cell efficiency is to create a layered structure of several cells. The main advantage of the thin-film PV technology is that it can be made flexible and come in different shapes and therefore can be used in many applications. The amorphous silicon is also less prone to overheating, which usually decreases the solar cell performance. Amorphous silicon is most developed among the thin-film PV [14].

### 1.4.3 Recent innovations:

Cadmium Telluride, CdTe (thin-film) is a promising technology, its main advantages are that they capture shorter wavelengths of light than silicon cells can do, it also has an efficiency of 16%. Another interesting technology is Copper Indium Gallium Selenide (CIGS) it has a decent efficiency of ~20%. At this moment, the CIGS are the most efficient among the thin-film PV technologies. While, lab results confirmed high promise of this kind of photovoltaics, the mass production of CIGS proved to be a problem.

## 1.5 From cells to modules to arrays:

A PV module is created by connecting many cells in a circuit. This assembly of cells is done in different ways depending on the technologies and can cause additional losses. The module is usually framed in some glass and aluminum. If several PV modules are electrically connected and mounted on one single supporting structure, in other words when they are mechanically linked, they form a solar panel. A photovoltaic array is the complete power-generating unit, consisting of any number of PV modules and panels that are mechanically independent. Apart from the mechanical connection, panels also need to be electrically linked through cables. Multiple panels that are electrically connected are called a string. The panels in a string are usually connected in series, to obtain a desired output voltage. When two or more strings are connected in parallel, they again from an array. By connecting multiple strings in parallel, the output current of the PV array is increased.

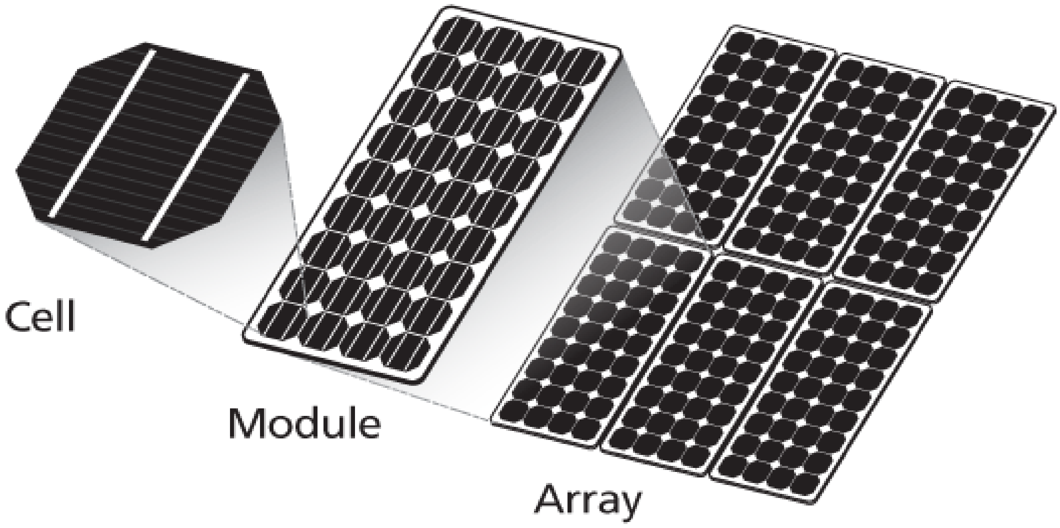


**Figure 1.14:** Photovoltaic cell, module and array [15].

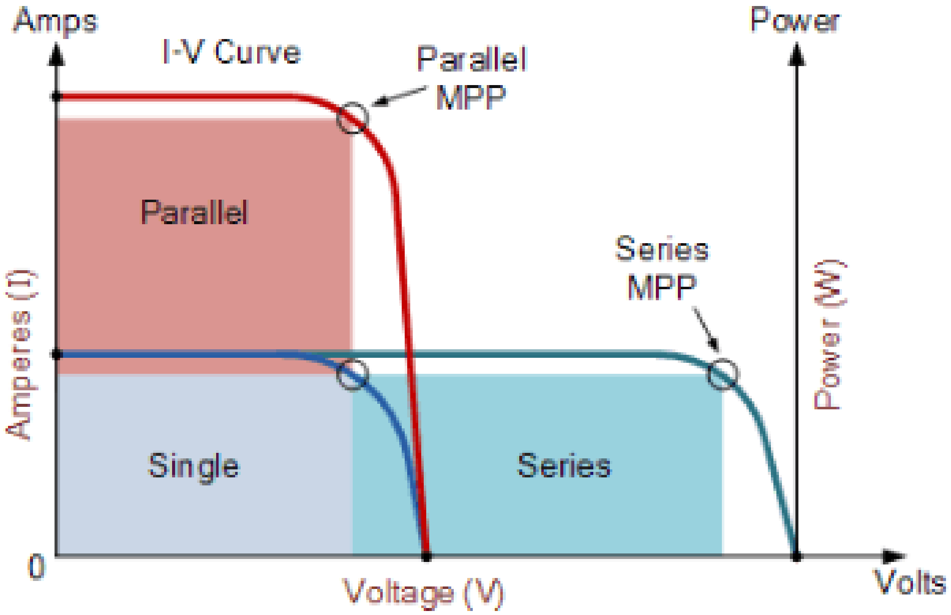


**Figure 1.15:** Charachteristing of PV Cells According to Wiring [16].

## 1.6 Challenges of integrating PV energy to the grid:

Solar-Grid integration is the technology that allows large scale solar power produced from PV system to penetrate the already existing power grid. This technology requires careful considerations and attentions in areas of solar component manufacturing, installations and operation. Owing to the unbalanced, random, and volatile characteristics of the solar resources, the PV generation itself is not dispatchable. Thus, the grid-connection requirements of PV systems vary based on the PV capacity, grid-connection mode, and the target grid. From the perspective of the power grid, connecting PV systems to the power grid can be complex and difficult. In other words, the existing PV grid-connection technologies do not seem to be grid friendly. Fortunately, several promising approaches and techniques are being developed to facilitate and stabilize the integration of PV energy to the grid, we note that PV power is reaching higher and higher penetration level in the smart grid. An important feature of the smart grid is its high ability to integrate renewable energy generation. Development of protection technologies of inverters will have a great impact on the safe and stable operation of the power grid, and finally to ensure secure and economic integration of PVs into the smart grid, accurate PV power forecasting has become a critical element of energy management systems. Accurate forecasting can help improve the quality of the electric power quality delivered to the electrical network end, and thus reduce the ancillary costs associated with general volatility. Since PV power output is directly related to solar irradiance at the ground level, solar irradiance prediction is also equally important to energy management in the smart grid. Moreover, solar prediction with multiple look-ahead times is significant in that it addresses the needs of different operation and control activities, including grid regulation, power scheduling, and unit commitment in both the distribution and transmission grids [17].

Some notable challenges associated with Solar-Grid integration include problems of voltage stability, frequency stability, and overall power quality. According to Belcher et al, a distributed system is considered large-scale when loading on the system is greater than 10 MW. Systems under this limit do not qualify for power integration and usually have many power quality issues. However, large-scale systems also experience power quality problems. Power generation plants that use the conventional method to spin a turbine benefit from having complete control over generation, Photovoltaic generation does not have the luxury of producing power on demand. Power quality issues range from voltage and frequency to other areas such as harmonics. The harmonics problem comes mainly from power inverters used in converting renewably generated DC voltage into AC. Harmonics are created by certain loads who introduce frequencies that are multiples of 50 or 60 Hz and can cause equipment to not operate as intended. The inherent non-dispatchable characteristics of PV systems (i.e. generation of electrical energy that cannot be turned on or off in order to meet societies fluctuating electricity needs) allow voltage generation fluctuations that have not previously been present in the grid. In order to combat these voltage issues, storage solutions along with other instantaneous power producing solutions are on the forefront of current PV research and development. Alongside the intermittency of PV generation itself, there are also grid-connected voltage quality issues that must be considered. Power plants must be able to ride-through various voltage levels sags in order to operate with-out outages. This requires that PV plants should be adaptable to voltage sags just as conventional power plants [18].

## 1.7 The importance of PV forecasting:

The forecasting of power generated by variable energy resources such as wind and solar has been the focus of academic and industrial research and development for as long as significant amounts of these renewable energy resources have been connected to the electric grid. Numerical Weather Prediction (NWP) models became more sophisticated in assessing cloud interactions with aerosols;

infrared satellite imagery allowed discovery of pre-sunrise cloud formations; advanced data processing methods such as deep machine learning became increasingly accessible; probabilistic forecasts began replacing deterministic ones; and, in balancing areas with high PV penetration, solar forecasts are now used operationally. Because solar power generation depends mostly on incident irradiance, the cost-efficient integration of significant amounts of solar electricity in the grid ultimately depends on the ability to forecast accurately solar irradiance. However, knowledge of the future level of irradiance is not by itself adequate for the calculation of solar power output. Knowledge of the attributes of the interconnected systems (such as DC and AC nameplate capacities, orientation, PV module and inverter properties, etc.) is also necessary. At the same time, efficient operation of the grid requires the accurately projected contribution of solar generation to be presented in a manner that allows almost error-free, optimally-timed decision making by the operators and/or the automated systems they use during Unit Commitment and Economic Dispatch operations. In summary, from a load balancing perspective, the reliable and economically optimal operation of an electric grid with high penetration of solar [19] (especially distributed solar) generation depends on:

- Accurate forecasting of the solar irradiance and its evolution in time over the area of interest, with temporal resolutions that range from 5 minutes to hourly for time horizons between 0 and 72 hours, with 1-6 hour and day-ahead horizons being of particular importance;
- Accurate forecasting of solar power output (and its evolution in time, including variability) over the area of interest, including an estimate of the forecast's uncertainty;
- Effective integration of the projected solar power output information with the systems used to manage and operate the network and other generation sources.

## 1.8 Conclusion:

In this chapter we reviewed the fundamentals of photovoltaic systems, in other words we covered the energy potential of Photovoltaic power namely in the world, north Africa and finally Algeria, we then examined the science behind solar cells, the main components of a basic PV system, the different technologies used in photovoltaics manufacturing, the different types of PV systems and their main components, we also explained the connection and wiring required to go from a single solar PV cell up to the PV array, We have also straightened out the matter of integrating and connecting solar systems to the grid and all the challenges that implies we then closed the chapter by highlighting the benefits and importance of forecasting of PV to ensure an efficient and stable operation in grid connected systems.

# CHAPTER 2:

# Machine Learning models

When most people hear “Machine Learning” they picture a robot: a dependable butler or a deadly Terminator, depending on who you ask. But Machine Learning is not just a futuristic fantasy; it’s already here and we are dealing with it every day when we use our smartphones or check our emails or even watch a movie on Netflix.

The first ML application that really became mainstream, improving the lives of hundreds of millions of people, took over the world back in the 1990s: the spam filter, which is a Machine Learning program that is used to detect unsolicited and unwanted email and prevent those messages from getting to a user's inbox. It was followed by hundreds of ML applications that now quietly power hundreds of products and features that we use regularly, from better recommendations on YouTube to voice search with Siri or Google’s Alexa down to cancer diagnosis and stock market trading, Machine learning is framing the modern world.

The nearly limitless quantity of available data, affordable data storage, and the growth of less expensive and more powerful processing has propelled the growth of machine learning. Today, ML has advanced to the point where is has the ability to transform every major sector in coming years, and just as electricity revolutionized lives 100 years ago, AI and ML are changing our lives completely today, we are on the point where we cannot live without AI and ML

In contrast, concerns are rising in the scientific community about the negative impact AI and ML will have on jobs, people’s privacy and how it can be a threat to humanity if we reach the singularity point where AI become more intelligent than humans and independent on human-will. Tesla and SpaceX CEO Elon Musk has repeatedly said that he thinks AI poses a threat to humanity and it is more dangerous than nuclear weapons.

So, where does Machine Learning start and where does it end? What exactly does it mean for a machine to learn something? What are the most known techniques to train a ML model? how a model is evaluated and how it is tuned?

## 2.1 Fundamentals of Machine Learning:

The first one who used the term machine learning was the American well-known scientist Arthur Samuel in 1959, when he defined machine learning as:

“The field of study that gives computers the ability to learn without being explicitly programmed”

- Arthur Samuel, 1959

And a more engineering-oriented definition is:

“A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.”

- Tom Mitchell, 1997

We can take the example of the spam filter to clarify this definition, where:

- The task T is classifying emails as spam or not spam
- The experience E is the already labeled emails as spam or not spam
- And the performance measure P is the number (or fraction) of emails correctly classified

Machine learning is about building models from data, it is a branch of Artificial Intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

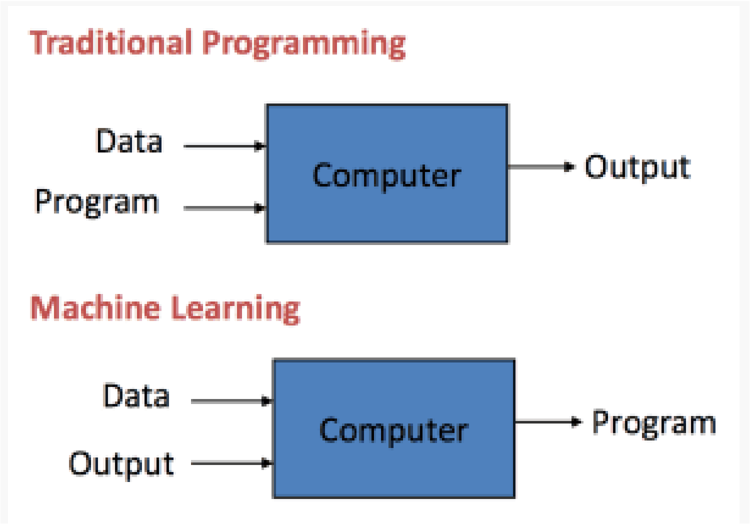


**Figure 2.1:** difference between traditional programming and machine learning.

learning is the process of converting experience into expertise or knowledge. The input to a learning algorithm is training data, representing experience, and the output is some expertise, which usually takes the form of another computer program that can perform some tasks; some concrete examples of these tasks are:

- Analyzing images of products on a production line to automatically classify them
- Detecting tumors in brain scans
- Automatically flagging offensive comments in social media
- Summarizing long documents automatically
- Creating a chatbot or a personal assistant
- Recommending systems
- Detecting credit card fraud
- Forecasting PV energy to help greening the grid.

## 2.2 Types of Machine Learning:

There are so many different types of Machine Learning systems that it is useful to classify them in broad categories, based on the following criteria:

- Whether or not they are trained with human supervision (supervised, unsupervised, semi-supervised, and Reinforcement Learning)
- Whether or not they can learn incrementally on the fly (online versus batch learning)
- Whether they work by simply comparing new data points to known data points, or instead by detecting patterns in the training data and building a predictive model, much like scientists do (instance-based versus model-based learning).

### 2.2.1 Supervised / unsupervised Learning:

When the machine is supervised while it is “learning”, the training type is called supervised learning. It means that we provide the machine with a ton of information about a case and also provide it with the case outcome. The outcome is called the labelled data while the rest of the information is used as input features.

in case of unsupervised learning, there is no help from the user for the computer to learn. In the lack of labelled training sets, the machine identifies patterns and similarities in the data that is not so obvious to the human eye and then cluster the outcome.

While in semi-supervised learning, a small amount of data is labeled. Computers only need to find features through labeled data and then classify other data accordingly. This method can make predictions more accurate and is the most commonly used method. If there are 100 photos of dogs and cats, 10 of them are labeled (weather it’s a dog or cat), through the characteristics of these 10 photos, the machine identifies and classifies the remaining photos. Because there is already a basis for identification, the predicted results are usually more accurate than un-supervised learning.

For reinforcement learning, the algorithm or the agent learns continually from its environment by interacting with it. There is no labeled data but the agent gets a positive or a negative reward based on its action.

### 2.2.2 batch / online Learning:

In batch learning the machine learning model is trained using the entire dataset that is available at a certain point in time. Once we have a model that performs well on the test set, the model is shipped for production and thus learning ends. This process is also called offline learning.

Whereas in online learning the model is trained and launched into production, but it keeps learning as new data comes in, it is great for systems that receive data as a continuous flow (e.g., stock prices) and need to adapt to change rapidly and autonomously.

### 2.2.3 instance-based / model-based learning:

One more way to categorize Machine Learning systems is by how they generalize, model-based learning algorithms use the training data to create a model that has parameters learned from the training data, after the model is built, the training data can be discard.

Whereas instance-based learning algorithms use the entire dataset as the model, for example, the algorithm looks at the close neighborhood of the input example in the space of feature vectors and outputs the label that it saw the often in this close neighborhood, thus the data cannot be discard.

Most supervised learning algorithms are model-based. In model-based, you can generalize your rules in the form of a model which can be stored unlike instance-based where generalization happens for each scoring instance individually as when seen, this makes model-based faster in scoring for new instance and have smaller size.

Now after seeing the fundamentals and different types of machine learning, we can dive in into some more complex concepts and talk about the different machine learning algorithms and the various comparative performance analysis that we used for the task of PV forecasting.

## 2.3 PV forecasting techniques:

We are probably living in the most defining period of human history. The period when computing moved from large mainframes to PCs to cloud. But what makes it defining is not what has happened, but what is coming our way in years to come.

What makes this period exciting and enthralling is the democratization of the various tools and techniques, which followed the boost in computing. This enabled any person, anywhere on the planet, with a PC and an internet connection to get access to technologies that others spent decades developing and to work with it and develop it too.

This what enabled us to use the most advance techniques in machine learning and apply it on the task of PV forecasting, in this part of chapter two we will go gradually from the most basic techniques like linear regression to more complicated ones like neural networks and ensemble learning.

### 2.3.1 Linear regression:

Linear regression is one of the simplest, most well-known and well understood algorithms in statistics and machine learning. It is an attractive model because the representation is so simple.

The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric.

The linear equation assigns one scale factor to each input value or column, called a coefficient and represented by the capital Greek letter Beta (B). One additional coefficient is also added (C), giving the line an additional degree of freedom (e.g., moving up and down on a two-dimensional plot) and is often called the intercept or the bias coefficient [20].

For example, in a simple regression problem (a single x and a single y), the form of the model would be:

$$y = C + B*x \quad \ldots\ldots. \ (2.1)$$

In higher dimensions when we have more than one input (x), the above equation will become:

$$\begin{matrix} y0 \\ y1 \\ .. \\ yn \end{matrix} = \begin{matrix} c0 \\ c1 \\ .. \\ cn \end{matrix} + \begin{bmatrix} b00 & \cdots & bn0 \\ \vdots & \ddots & \vdots \\ b0m & \cdots & bnm \end{bmatrix} * \begin{matrix} x0 \\ x1 \\ .. \\ xn \end{matrix} \quad \ldots\ldots. \ (2.2)$$

It is common to talk about the complexity of a regression model like linear regression. This refers to the number of coefficients used in the model.

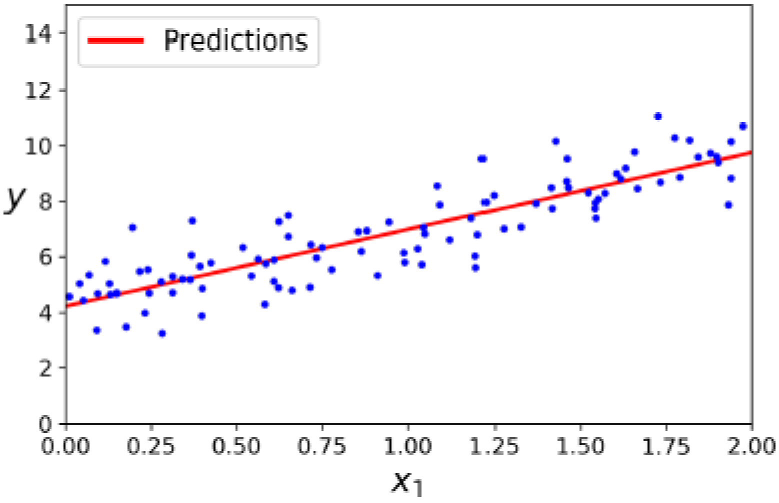


**Figure 2.2:** Linear regression model prediction

Now that we understand the representation used for a linear regression model, let’s see how this representation can learn from data, there are two famous ways to do that:

1- Ordinary Least Squares:

This procedure seeks to minimize the sum of the squared residuals. This means that given a regression line through the data we calculate the distance from each data point to the regression line, square it, and sum all of the squared errors together. This is the quantity that ordinary least squares seek to minimize.

This approach treats the data as a matrix and uses linear algebra operations to estimate the optimal values for the coefficients. It means that all of the data must be available and you must have enough memory to fit the data and perform matrix operations.

2- Gradient decent:

When there are one or more inputs you can use a process of optimizing the values of the coefficients by iteratively minimizing the error of the model on your training data.

This operation is called Gradient decent and works by starting with random values for each coefficient. The sum of the squared errors is calculated for each pair of input and output values. A learning rate is used as a scale factor and the coefficients are updated in the direction towards minimizing the error. The process is repeated until a minimum sum squared error is achieved or no further improvement is possible [23].

The second method is more practical when we have large datasets.

But What if our data is more complex than a straight line? Surprisingly, we can use a linear model to fit nonlinear data. A simple way to do this is to add powers of each feature as new features, then train a linear model on this extended set of features. This technique is called Polynomial Regression.

### 2.3.2 Polynomial Regression:

If the data distribution is complex as we can see in Figure XX, straight lines might not capture patterns of the data. Therefore, unlike the linear equation, polynomial equations depicted in the equation below, is more suitable to capture the data distribution.

$$Y = \theta o + \theta_1 X + \theta_2 X^2 + \ldots + \theta_{\mathrm{m}} X^{\mathrm{m}} \quad \ldots\ldots (2.3)$$

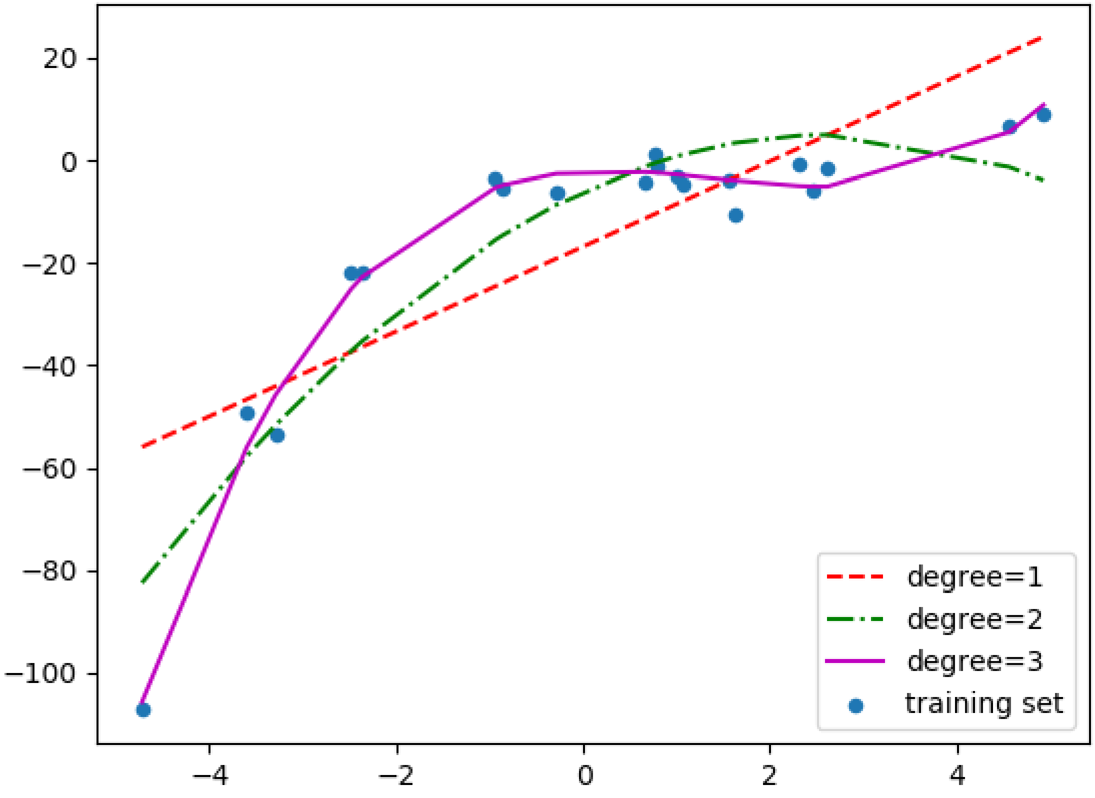


**Figure 2.3:** polynomial regression model predictions with order 1, 2 and 3

The order of the polynomial model is kept as low as possible. Some transformations can be used to keep the model to be of the first order. If this is not satisfactory, then the second-order polynomial is tried. Arbitrary fitting of higher-order polynomials can be a serious abuse of regression analysis. A model which is consistent with the knowledge of data and its environment should be taken into account. It is always possible for a polynomial of order (n - 1) to pass through n points so that a polynomial of sufficiently high degree can always be found that provides a "good" fit to the data. Such models neither enhance the understanding of the unknown function nor be a good predictor [21].

By taking the model's order 20, we can see how it fits the training set very well, and even capture the noise in the data, but fail to generate to new instances. This is an example of over-fitting.

To prevent over-fitting, we can add more training samples so that the algorithm doesn't learn the noise in the system and can become more generalized, or choosing lower order.

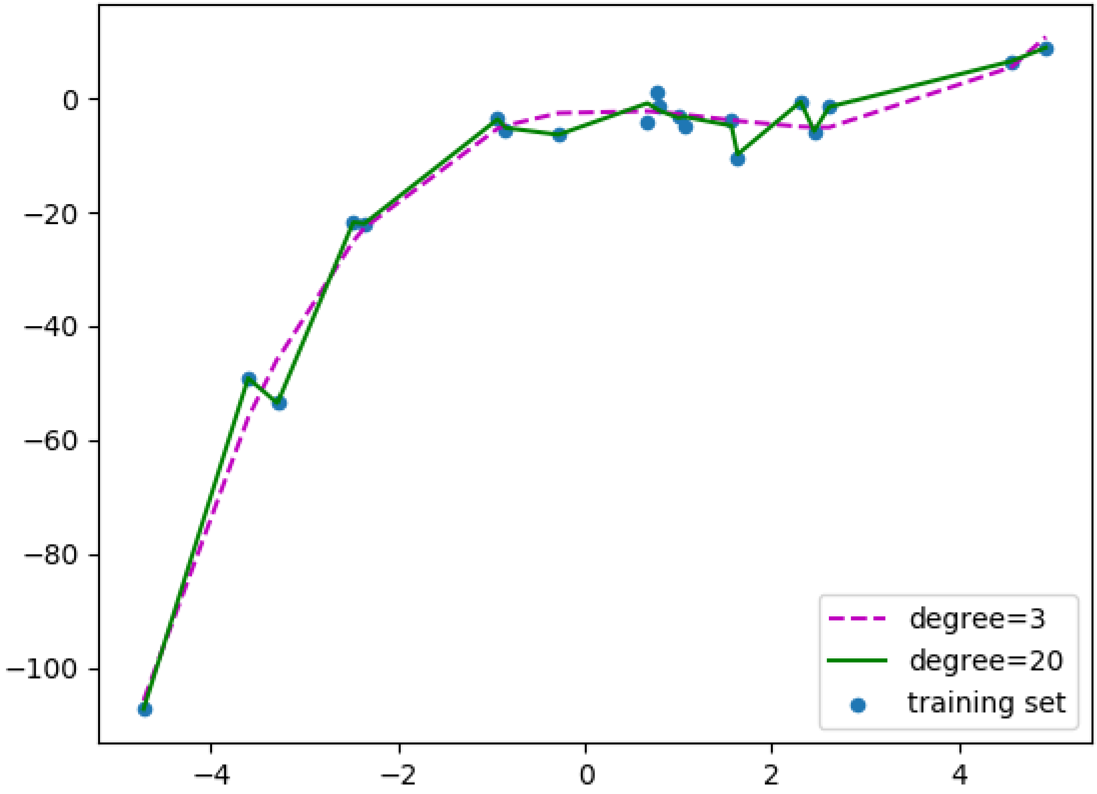


**Figure 2.4:** polynomial regression model over-fitting the training set.

Comparing to linear model, the advantages of using polynomial regression are:

- Polynomial provides the best approximation of the relationship between the dependent and independent variable.
- A Broad range of function can be fit under it.
- Polynomial basically fits a wide range of curvature.

But it has some disadvantages which are:

- The presence of outliers in the data can seriously affect the results of the nonlinear analysis which means polynomial regression models are too sensitive to the outliers.
- In addition, there are unfortunately fewer model validation tools for the detection of outliers in nonlinear regression than there are for linear regression.

### 2.3.3 Support Vector Machine (SVM):

A Support Vector Machine (SVM) is a powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification, regression, and even outlier detection. It is one of the most popular models in Machine Learning. SVM's train time complexity is O($n^2$). If the dataset is very large SVM may take a while to train, that's why SVMs are particularly well suited for small- or medium-sized datasets.

To understand how an SVM regressor work, we need to understand first the working principle of an SVM classifier.

The linear SVM classifier model predicts the class of a new instance x by simply computing the decision function

$$\boldsymbol{w}^{\intercal}\boldsymbol{x} + b = w1\,x1 + \cdots + wn\,xn + b \quad \ldots\ldots (2.4)$$

If the result is positive, the predicted class ŷ is the positive class (1), and otherwise it is the negative class (0); as shown in the following equation;

$$y =$$

$$0\ if\ \boldsymbol{w}^{\intercal}\boldsymbol{x} + b < 0,$$

$$1\ if\ \boldsymbol{w}^{\intercal}\boldsymbol{x} + b \geq 0 \quad \ldots\ldots (2.5)$$

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier [22]. The figure below shows the decision function for a linearly separable example, with three samples on the margin boundaries, called "support vectors":

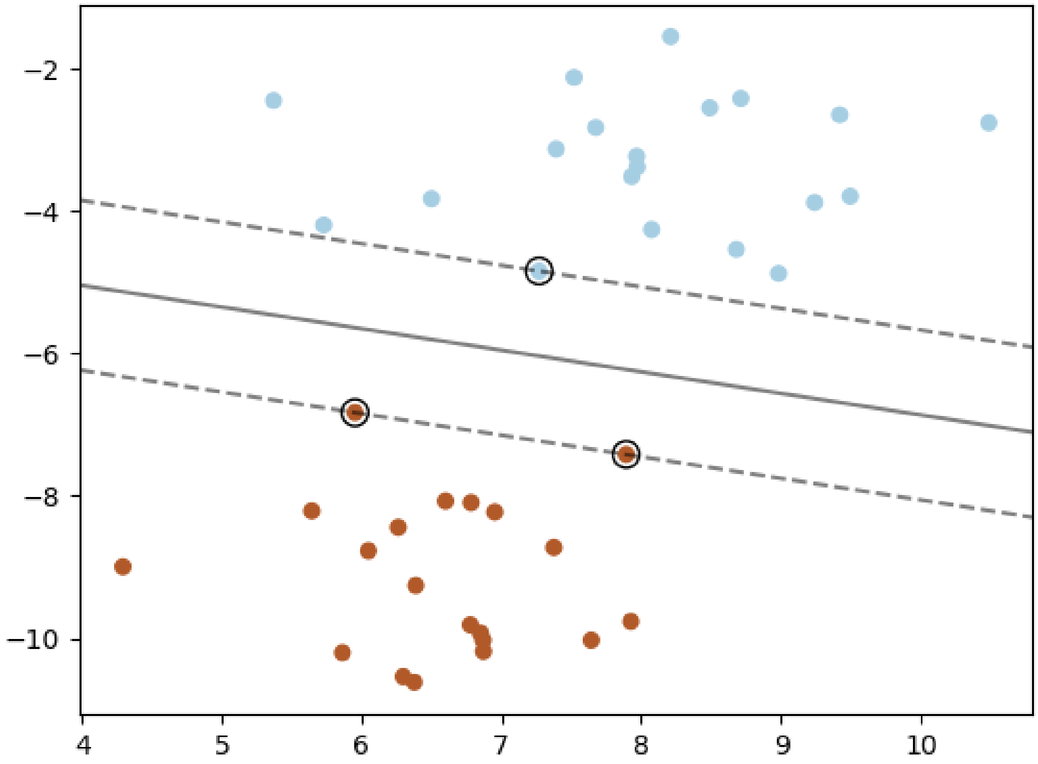


**Figure 2.5:** SVM’s decision function for a linearly separable problem

In general, when the problem isn’t linearly separable, the support vectors are the samples within the margin boundaries.

The method of Support Vector Classification can be extended to solve regression problems. This method is called Support Vector Regression.

The model produced by support vector classification (as described above) depends only on a subset of the training data, because the cost function for building the model does not care about training points that lie beyond the margin. Analogously, the model produced by Support Vector Regression depends only on a subset of the training data, because the cost function ignores samples whose prediction is close to their target [23].

The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

### 2.3.4 Decision Tree:

Tree models are among the most popular models in supervised machine learning, they are expressive and easy to understand.

A decision tree is arriving at an estimate by asking a series of questions to the data, each question narrowing our possible values until the model get confident enough to make a single prediction. The order of the question as well as their content are being determined by the model. In addition, the questions asked are all in a True/False form, the most common tree structure can be defined as follow:

“A feature tree is a tree such that each internal node (the nodes that are not leaves) is labelled with a feature, and each edge emanating from an internal node is labelled with a literal. The set of literals at a node is called a split. Each leaf of the tree represents a logical expression, which is the conjunction of literals encountered on the path from the root of the tree to the leaf. The extension of that conjunction (the set of instances covered by it) is called the instance space segment associated with the leaf” [24].

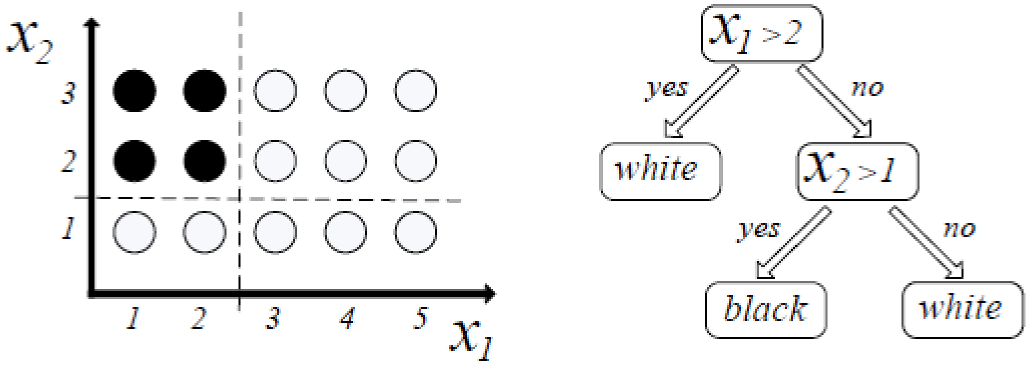


**Figure 2.6:** example data of two classes with black and white discs and a corresponding decision tree for classification on the right [25]

Tree models are not limited to classification but can be employed to solve almost any machine learning task, including ranking and probability estimation, regression and clustering.

We used in our project a decision tree regressor. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches, each representing values for the attribute tested. Leaf node represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

The most important aspect of the decision tree is how it actually learns (how the 'questions' are formed and how the thresholds are set). As a supervised machine learning model, a decision tree learns to map data to outputs in what is called the training phase of model building.

During training, the model is fitted with any historical data that is relevant to the problem domain and the true value we want the model to learn to predict. The model learns any relationships between the data and the target variable.

After the training phase, the decision tree produces a tree calculating the best questions as well as their order to ask, in order to make the most accurate estimates possible. When we want to make a prediction the same data format should be provided to the model in order to make a prediction. The prediction will be an estimate based on the train data that it has been trained on [26].

Decision trees regression normally use mean squared error (MSE) to decide to split a node in two or more sub-nodes. First will pick a value, and split the data into two subsets. For each subset, it will calculate the MSE separately. The tree chooses the value with results in smallest MSE value.

A common problem with decision trees is that they tend to overfit decisions along the way that are made from individual features. This is where the random forest idea comes into play.

### 2.3.5 Random Forest:

A random forest is an ensemble method which makes decisions based on an ensemble of decision trees. The different decision trees are created by creating different training sets for each of them by randomly sampling. Since a decision tree is a non-parametric method, which means it uses a flexible number of parameters, and the number of parameters often grows as it learns from more data, this can be particularly effective way to create a variety of trees where a vote of all the trees in the end can make better decisions as it prevents some form of overfitting. This is why it is common in practice to use random forests instead of a single decision tree. They are powerful algorithms, capable of fitting complex datasets.

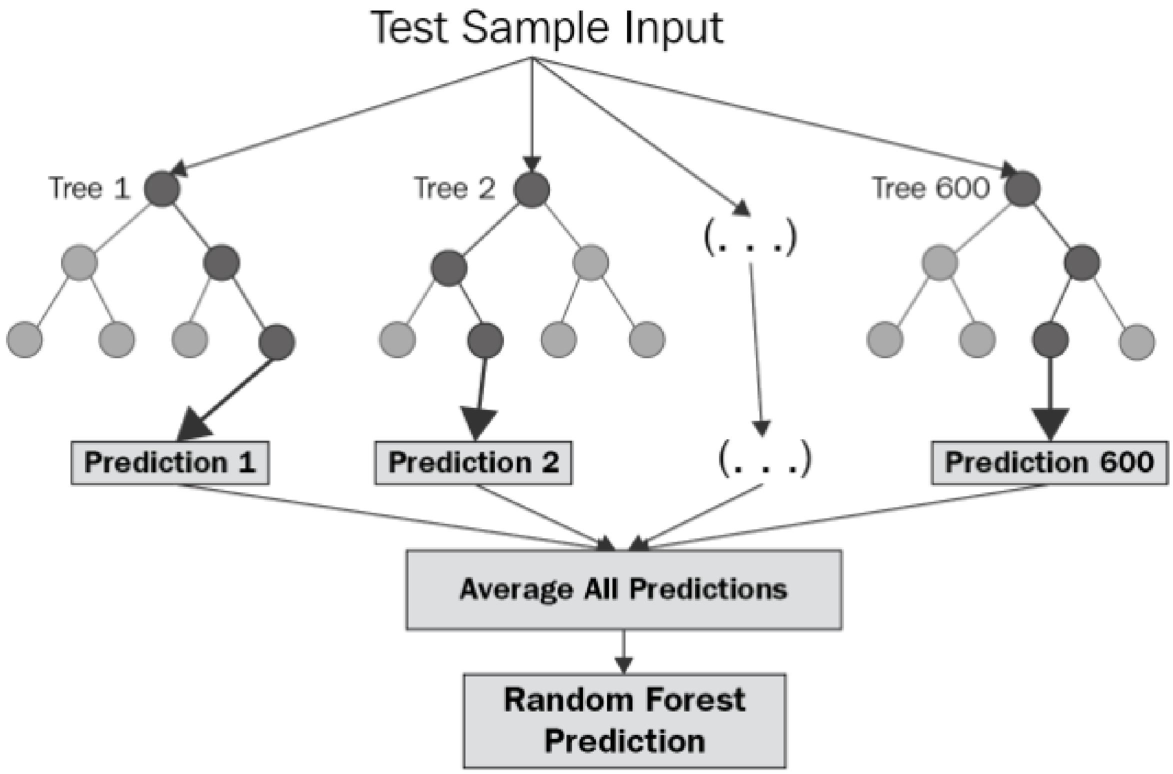


**Figure 2.7:** the structure of random forest [27]

The CART algorithm used in random forest tries to split the training set in a way that minimizes the MSE. The below equation shows the cost function that the algorithm tries to minimize.

$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} \left(\hat{y}_{\text{node}} - y^{(i)}\right)^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

……. (2.6)

Just like for classification tasks, Decision Trees are prone to overfitting when dealing with regression tasks. Without any regularization (i.e., using the default hyperparameters), we get the predictions on the left in Figure 23. These predictions are obviously overfitting the training set very badly. Just setting minimum number of leaf nodes to 10 results in a much more reasonable model, represented on the right in Figure 23.

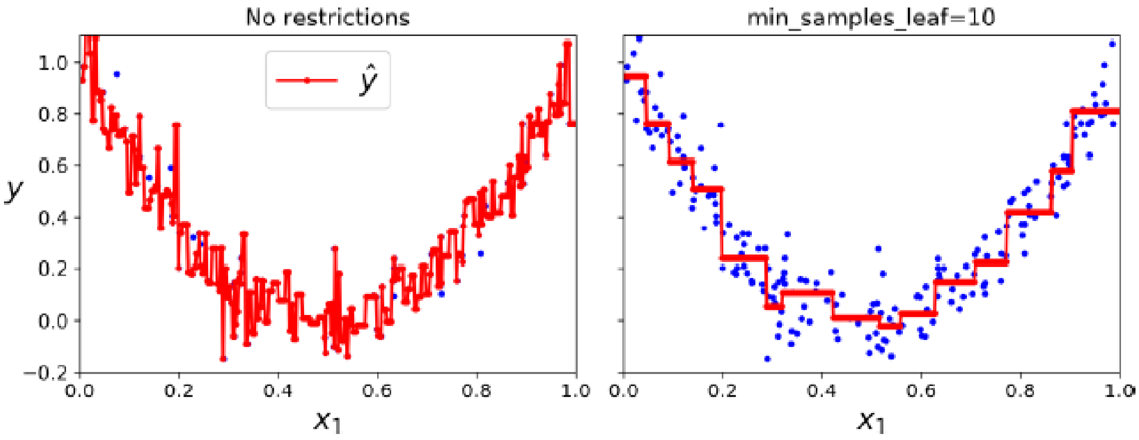


**Figure 2.8:** regularizing a decision tree in random forest regressor

Random forest is the most used technique of ensemble learning, but there are other techniques that we were able to use and gave us promising results.

### 2.3.6 Ensemble Learning:

The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability / robustness over a single estimator.

Two families of ensemble methods are usually distinguished:

- In averaging methods, the driving principle is to build several estimators independently and then to average their predictions. On average, the combined estimator is usually better than any of the single base estimator because its variance is reduced.
- By contrast, in boosting methods, base estimators are built sequentially and one tries to reduce the bias of the combined estimator. The motivation is to combine several weak models to produce a powerful ensemble.

In our case, we used Voting Regressor and AdaBoost Regressor.

#### 2.3.6.1 Voting regressor:

A voting regressor is an ensemble meta-estimator that fits several base regressors, each on the whole dataset, then it averages the individual predictions to form a final prediction. The idea behind the Voting Regressor is to combine conceptually different machine learning regressors and return the average predicted values. Such a regressor can be useful for a set of equally well performing models in order to balance out their individual weaknesses [28].

#### 2.3.6.2 AdaBoost:

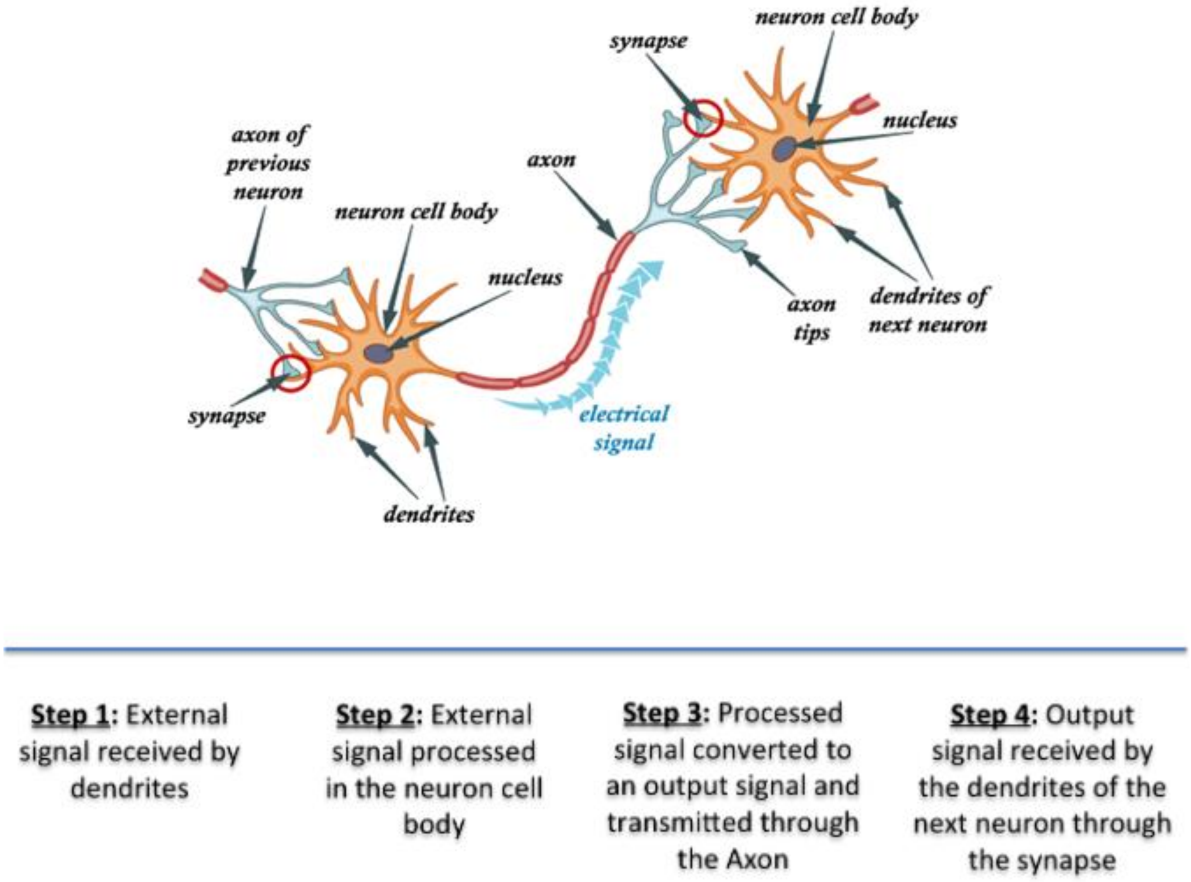
The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction. The data modifications at each so-called boosting iteration consist of applying weights $w1, w2, \dots, wN$ to each of the training samples. Initially, those weights are all set to $wi = 1/N$, so that the first step simply trains a weak learner on the original data. For each successive iteration, the sample weights are individually modified and the learning algorithm is reapplied to the reweighted data. At a given step, those training examples that were incorrectly predicted by the boosted model induced at the previous step have their weights increased, whereas the weights are decreased for those that were predicted correctly. As iterations proceed, examples that are difficult to predict receive ever-increasing influence. Each subsequent weak learner is thereby forced to concentrate on the examples that are missed by the previous ones in the sequence [29].

### 2.3.7 Neural Network:

Countless inventions were inspired by nature, so it seems logical when trying to build an intelligent machine to look at the brain’s architecture for inspiration. This is the key idea that sparked artificial neural networks (ANNs). However, ANNs have gradually become quite different from their biological cousins. Some researchers even argue that we should drop the biological analogy altogether (e.g., by saying “units” rather than “neurons”).

ANNs are at the very core of Deep Learning. They are versatile, powerful, and scalable, making them ideal to tackle large and highly complex Machine Learning tasks, such as classifying billions of images (e.g., Google Images), powering speech recognition services (e.g., Apple’s Siri), recommending the best videos to watch to hundreds of millions of users every day (e.g., YouTube), or forecasting PV energy.

To better understand ANNs, Let’s begin by first understanding how our brain processes information. In our brain, there are billions of cells called neurons, which processes information in the form of electric signals. External information/stimuli are received by the dendrites of the neuron, processed in the neuron cell body, converted to an output and passed through the Axon to the next neuron. The next neuron can choose to either accept it or reject it depending on the strength of the signal.



**Step 1**: External signal received by dendrites

**Step 2**: External signal processed in the neuron cell body

**Step 3**: Processed signal converted to an output signal and transmitted through the Axon

**Step 4**: Output signal received by the dendrites of the next neuron through the synapse

**Figure 2.9:** information processing in the brain

Now, let’s try to understand how an ANN works:

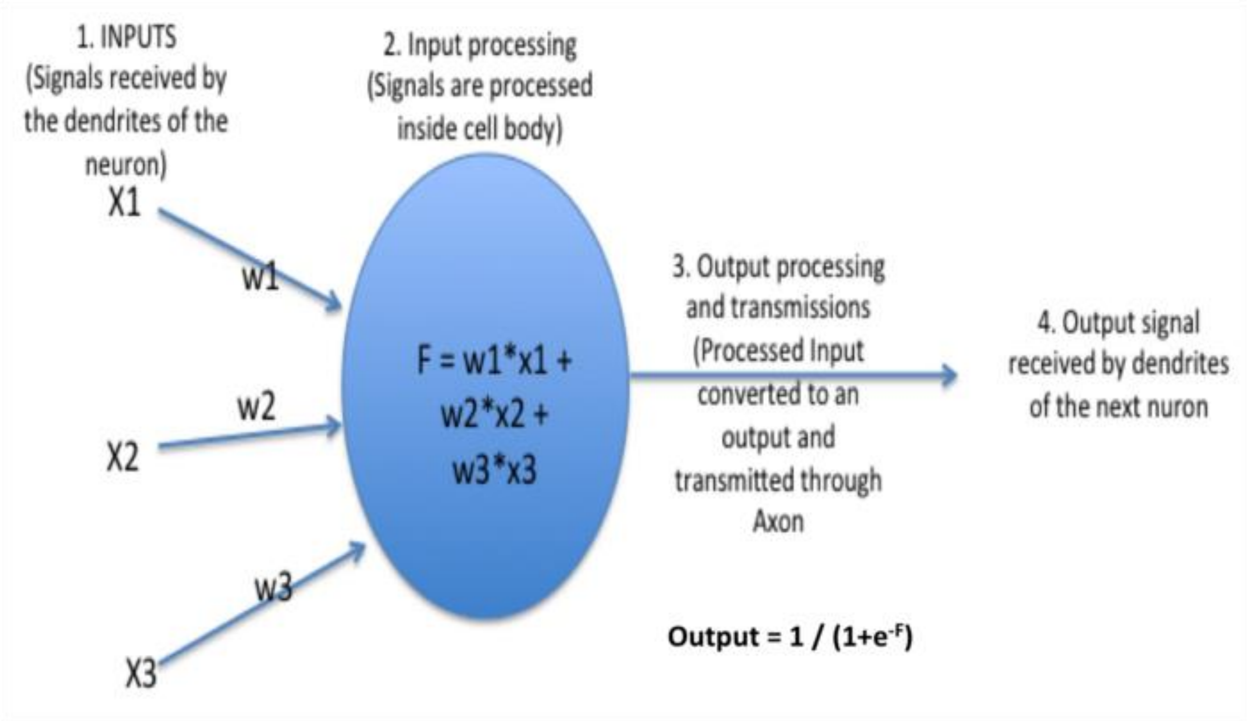
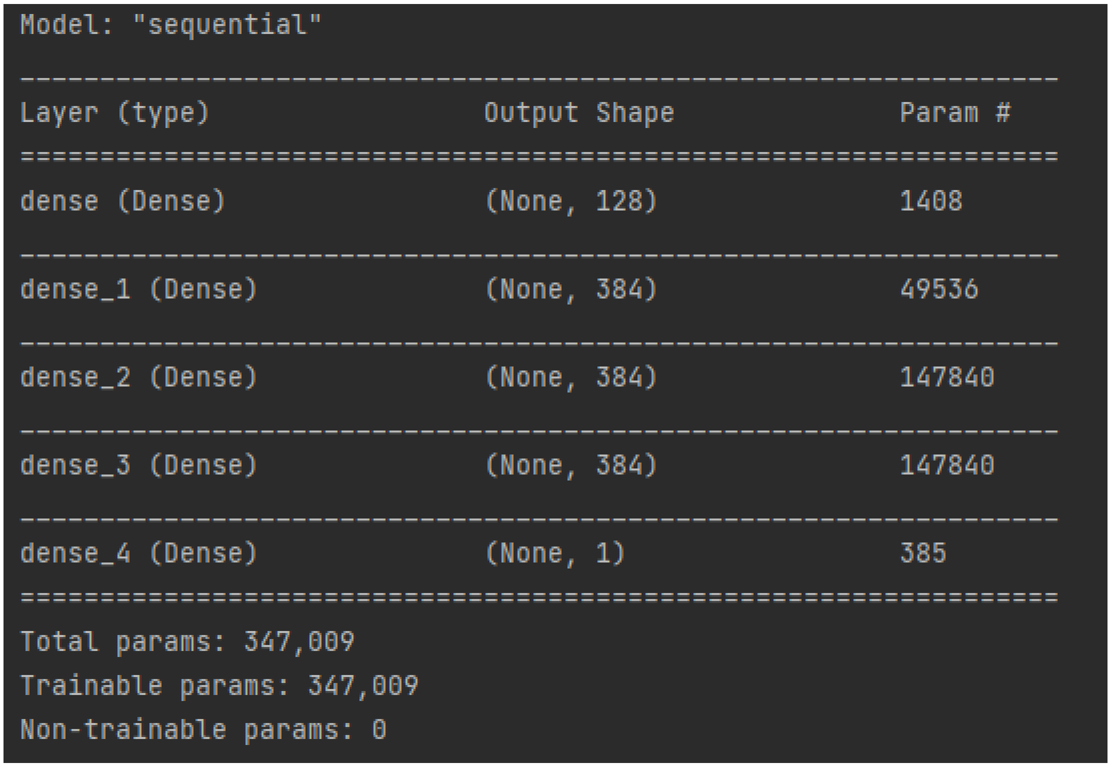

1. INPUTS
(Signals received by the dendrites of the neuron)
2. Input processing
(Signals are processed inside cell body)
X1
w1
X2
w2
X3
w3
F = w1*x1 + w2*x2 + w3*x3
3. Output processing and transmissions
(Processed Input converted to an output and transmitted through Axon
4. Output signal received by dendrites of the next nuron
Output = 1 / (1+e^-F)

**Figure 2.10:** simplest ANN architecture

Here, w1, w2, w3 gives the strength of the input signals

As we can see from the above, an ANN is a very simplistic representation of a how a brain neuron works.

To deal with more complicated tasks, like forecasting PV power, we need more than one perceptron, the architecture that we used is as follow:



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 128)               1408
_________________________________________________________________
dense_1 (Dense)              (None, 384)               49536
_________________________________________________________________
dense_2 (Dense)              (None, 384)               147840
_________________________________________________________________
dense_3 (Dense)              (None, 384)               147840
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 385
=================================================================
Total params: 347,009
Trainable params: 347,009
Non-trainable params: 0
```

**Figure 2.11:** ANN architecture used for PV forecasting

We have an input layer with 128 unit, three hidden layers with 384 unit each, and one output unit to give us the forecasted power. We got a total of 347009 trainable parameter.

In order to capture non-linear relationships between the inputs, we used the ReLU function as an activation function for each unit.

The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

### 2.3.8 LSTM:

Long Short-Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies.

Recurrent Neural Network is a generalization of feedforward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the recurrent network. For making a decision, it considers the current input and the output that it has learned from the previous input.

Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. In other neural networks, all the inputs are independent of each other. But in RNN, all the inputs are related to each other [30].

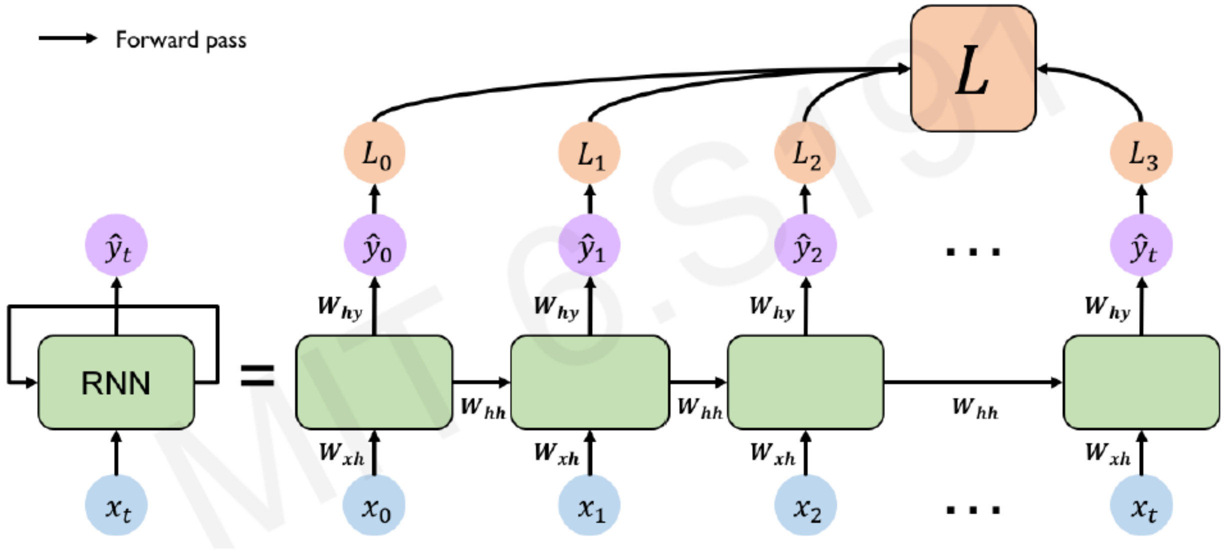


**Figure 2.12:** an unrolled recurrent neural network [31]

First, it takes the X(0) from the sequence of input and then it outputs h(0) which together with X(1) is the input for the next step. So, the h(0) and X(1) is the input for the next step. Similarly, h(1) from the next is the input with X(2) for the next step and so on. This way, it keeps remembering the context while training.
The formula for the current state is:

$$ht = f(h(t-1), xt) \quad \ldots\ldots. (2.7)$$

Applying activation function:

$$ht = tanh(Whh * h(t-1) + Wxh * xt) \quad \ldots\ldots. (2.8)$$

W is weight, h is the single hidden vector, Whh is the weight at previous hidden state, Whx is the weight at current input state, tanh is the Activation fucntion, that implements a Non-linearity that squashes the activations to the range [-1.1].

But the problem with RNN, in addition to the gradient vanishing and exploding problems, is that it cannot process very long sequences if using tanh or relu as an activation function, this is where LSTMs are introduced.

Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The vanishing gradient problem of RNN is resolved here. It trains the model by using back-propagation. In an LSTM network, three gates are present:

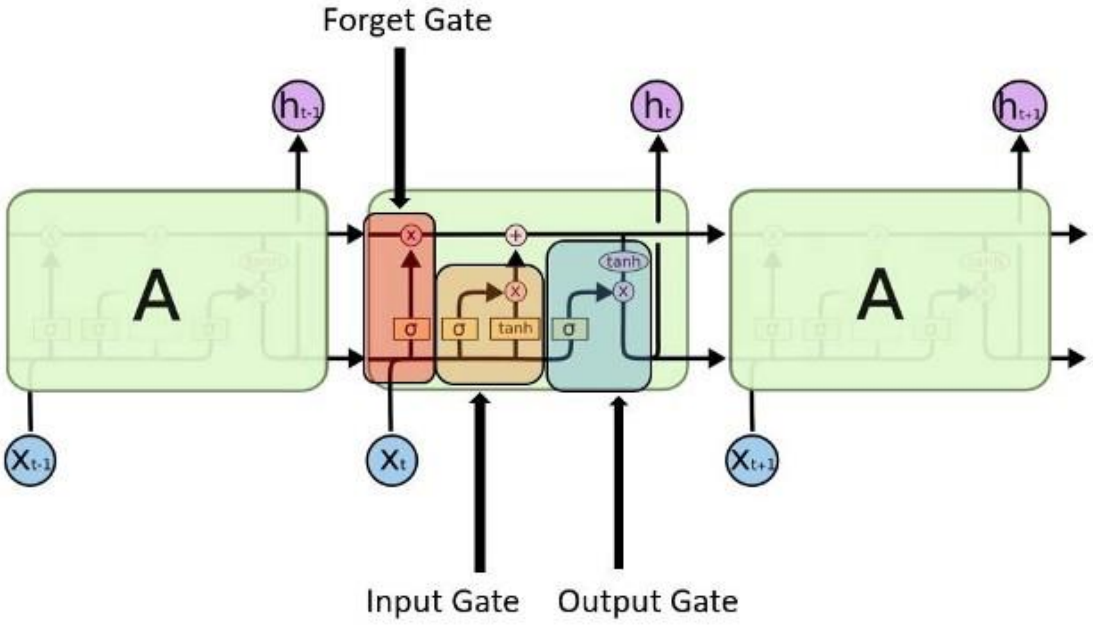


**Figure 2.13:** LSTM unit [31]

1) Input gate: discover which value from input should be used to modify the memory. Sigmoid function decides which values to let through 0,1. and tanh function

gives weightage to the values which are passed deciding their level of importance ranging from-1 to 1.

2) Forget gate: discover what details to be discarded from the block. It is decided by the sigmoid function. it looks at the previous state(ht-1) and the content input (Xt) and outputs a number between 0 (omit this) and 1 (keep this) for each number in the cell state Ct−1.
3) Output gate: the input and the memory of the block is used to decide the output. Sigmoid function decides which values to let through 0,1. and tanh function gives weightage to the values which are passed deciding their level of importance ranging from-1 to 1 and multiplied with output of Sigmoid.

## 2.4 Evaluation metrics:

In this part of chapter 2 we will talk about the evaluation metrics that we used to evaluate model's performance and compare between them to find the most adequate model for the task of PV forecasting.

An evaluation metric quantifies the performance of a predictive model. This typically involves training a model on a dataset, using the model to make predictions on a holdout dataset not used during training, then comparing the predictions to the expected values in the holdout dataset [32].

Evaluation metrics differ on the assumptions they make about the problem or about what is important in the problem. Therefore, an evaluation metric must be chosen so that it best captures what is believed to be important about the model or predictions, which makes choosing model evaluation metrics challenging.

We must know that for regression models, we cannot calculate accuracy like in classification models, The performance of a regression model must be reported as an error in those predictions. This makes sense because when we are predicting a numeric value like the irradiance or PV output power, we don't want to know if the model predicted the value exactly (this might be intractably difficult in practice); instead, we want to know how close the predictions were to the expected values. Error addresses exactly this and summarizes on average how close predictions were to their expected values.

There are five error metrics that are commonly used for evaluating and reporting the performance of a regression model; they are:

- Mean Squared Error (MSE).
- Mean Absolute Error (MAE).
- Median Absolute Error (MedAE).
- Explained Variance Score (EVS).
- R squared Score (R2).

There are many other metrics for regression, although these are the most adequate for our task, to understand why we will take each one in detail.

### 2.4.1 Mean Squared Error:

Mean Squared Error, or MSE for short, is a popular error metric for regression problems. It is also an important loss function for algorithms fit or optimized using the least squares framing of a regression problem. Here "least squares" refers to minimizing the mean squared error between predictions and expected values.

The MSE is calculated as the mean or average of the squared differences between predicted and expected target values in a dataset;

$$MSE = \frac{1}{N} \sum_{j=1}^{N} (yj - \hat{y}\mathrm{j})^2 \quad \ldots\ldots. (2.9)$$

Where $yj$ is the j'th expected value in the dataset and ŷj is the j'th predicted value. The difference between these two values is squared, which has the effect of removing the sign, resulting in a positive error value. We then sum them and take the average by dividing over N which is the number of samples in the testing set.

The squaring also has the effect of inflating or magnifying large errors. That is, the larger the difference between the predicted and expected values, the larger the resulting squared positive error. This has the effect of "punishing" models more for larger errors when MSE is used as a loss function. It also has the effect of "punishing" models by inflating the average error score when used as a metric [32].

An extension of MSE is also commonly used, which is Root Mean Squared Error, or RMSE, the square root of the MSE is calculated, which means that the units of the RMSE are the same as the original units of the target value that is being predicted. In our case the target variable (the output power) has the unit “Watt” then the RMSE error score will also have the unit “Watt” and not “Watt Squared” like the MSE.

A perfect RMSE value is 0.0, which means that all predictions matched the expected values exactly. This is almost never the case, and if it happens, it suggests your predictive modeling problem is trivial.

A good RMSE is relative to the dataset. It is a good idea to first establish a baseline RMSE for your dataset using a naive predictive model, such as predicting the mean target value from the training dataset. A model that achieves an RMSE better than the RMSE for the naive model has skill.

### 2.4.2 Mean Absolute Error:

Mean Absolute Error, or MAE, is a popular metric because, like RMSE, the units of the error score match the units of the target value that is being predicted.

Unlike the RMSE, the changes in MAE are linear and therefore intuitive. That is, MSE and RMSE punish larger errors more than smaller errors, inflating or magnifying the mean error score. This is due to the square of the error value. The MAE does not give more or less weight to different types of errors and instead the scores increase linearly with increases in error.

As its name suggests, the MAE score is calculated as the average of the absolute error values. Absolute is a mathematical function that simply makes a number positive. Therefore, the difference between an expected and predicted value may be positive or negative and is forced to be positive when calculating the MAE.

The MAE can be calculated as follows:

$$MAE = \frac{1}{N} \sum_{J=1}^{N} |yj - \hat{y}j| \quad \ldots\ldots. (2.10)$$

Where $yj$ is the j'th expected value in the dataset and ŷj is the j'th predicted value. We take the difference between in absolute value and sum them then take the average by dividing over N which is the number of samples in the testing set.

Same as RMSE, A perfect MAE is 0.0, which is almost never the case. A good MAE is relative to the dataset.

### 2.4.3 Median Absolute Error:

The Median Absolute Error, or MedAE, is particularly interesting because it is robust to outliers. This is because it is the median of all of the absolute values of the residuals, and the median is unaffected by values at the tails. So, this loss function can be used to perform robust forecasting

The loss is calculated by taking the median of all absolute differences between the target and the prediction.

If $\hat{y}j$ is the predicted value of the j-th sample and $yj$ is the corresponding true value, then the median absolute error (MedAE) estimated over n samples is defined as:

$$MedAE = median(|y1 - \hat{y}j|, \dots, |yn - \hat{y}n|) \quad \ldots\ldots (2.11)$$

Unlike RMSE and MAE, MedAE does not support multioutput

### 2.4.4 Explained Variance Score:

Explained variation measures the proportion to which a mathematical model accounts for the variation of a given data set. Often, variation is quantified as variance; then, the more specific term explained variance can be used.

In probability theory and statistics, variance is the expectation of the squared deviation of a random variable from its mean. In other words, it measures how far a set of numbers is spread out from their average value. The variance of a random variable y is the expected value of the squared deviation from the mean of y, and it is calculated as follow:

$$Var(y) = E[(y - \mu)^2] \quad \ldots\ldots (2.12)$$

Where; $\mu = E(y)$

If $\hat{y}$ is the predicted value $y$ is the corresponding true value, and Var is variance (the square of the standard deviation), then the explained variance is estimated as follow:

$$EVS = 1 - \frac{Var(y - \hat{y})}{Var(y)} \quad \ldots\ldots. (2.13)$$

The explained variance score is between 0.0 and 1.0, and the best score is 1.0

### 2.4.5 R2 Score:

The R2 score computes the coefficient of determination, usually denoted as R². It represents the proportion of variance (of y) that has been explained by the independent variables in the model. It provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model, through the proportion of explained variance. As such variance is dataset dependent, R² may not be meaningfully comparable across different datasets.

If $\hat{y}j$ is the predicted value of the j-th sample and $yj$ is the corresponding true value, then the R2 score estimated over n samples is defined as follow:

$$R2 = 1 - \frac{\sum_{j=1}^{n}(yj - \hat{y}\mathrm{j})^2}{\sum_{j=1}^{n}(yj - \mathrm{Y})^2} \quad \ldots\ldots. (2.14)$$

Where; $Y = \frac{1}{n}\sum_{j=1}^{n} yj$

Note that r2 score calculates unadjusted R² without correcting for bias in sample variance of y. Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse).

After seeing how we evaluated and compared between different models, we will see next the techniques that we used to fine tune these models and extract the best hyperparameters.

## 2.5 Fine tuning the models:

After we defined our models, trained them with preprocessed data, tested them using the test set, and finally evaluated them with the five-evaluation metrics discussed above, now we want to know if this is the best scores we can get from them, what if we changed the number of decision trees in random forest, or the degree of the polynomial regressor, or number of hidden layers and numbers of neurons in ANNs, or any other hyperparameters of the models, will it give as a better result?

How we can be sure that this is the best score we can get from a specific model? This is where fine tuning techniques are used.

Fine-tuning, in general, means making small adjustments to a process to achieve the desired output or performance.

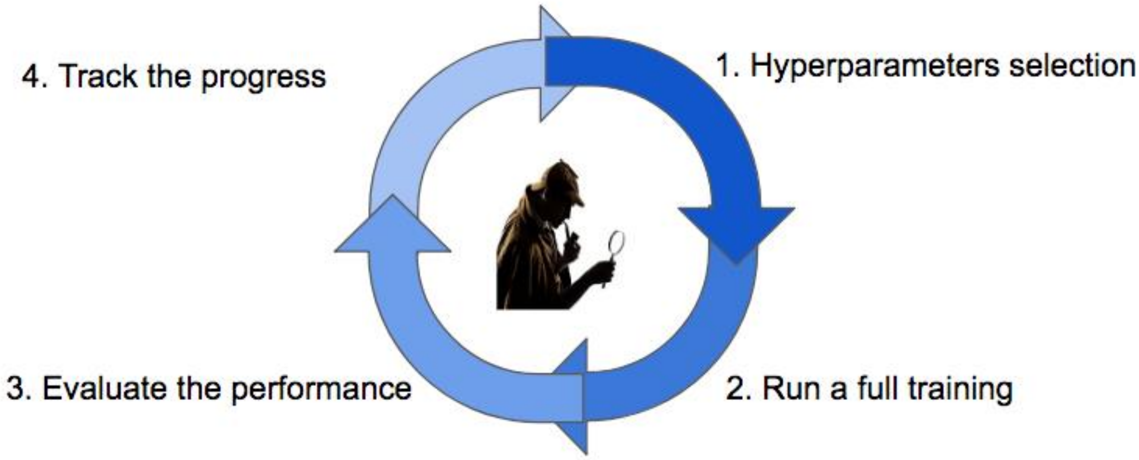


**Figure 2.14:** the hyperparameter search cycle [33].

Fine tuning machine learning predictive model is a crucial step to improve “accuracy” of the forecasted results, but in the same time it is very expensive and time consuming. In our project we used three known fine-tuning techniques depending on the model we are tuning.

### 2.5.1 Trial and Error:

As its name suggest, we improve the scores of our model by trying different combinations of different hyperparameters, which are the knobs that we can turn when building a machine learning model, then pick the hyperparameters that gives the best results.

This approach is 100% manual, we used this method for neural networks and LSTMs, the hyper parameters that we tuned are:

- Number of hidden layers
- Number of units in each layer
- Activation functions
- Learning rate

- Batch size
- Optimizers

We tried different combinations for each of these hyperparameters, by trial and error we were able to develop a pattern on how the model is learning and make modifications that improved its “accuracy” until we got the optimal hyperparameters. But this can be time consuming.

### 2.5.2 Grid Search:

Using this approach, we can optimize our time by defining an automatic strategy for hyperparameter searching, it is inspired by the naïve approach of simply trying every possible configuration. Here is its workflow :

1- We define a grid on n dimensions, where each of these maps for a hyperparameter.
2- For each dimension (hyperparameter), we define the range of possible values.
3- Try all possible configurations and wait for the results to establish the best one.

The real pain point of this approach is known as the curse of dimensionality. This means that more dimensions we add, the more the search will explode in time complexity (usually by an exponential factor), ultimately making this strategy unfeasible (unless we can afford to use huge computational power).

It's common to use this approach when the dimensions are less than or equal to 4. But, in practice, even if it guarantees to find the best configuration at the end, it's still not preferable. Instead, it's better to use Random Search; which we'll discuss next.

### 2.5.3 Random Search:

The only real difference between Grid Search and Random Search is on the step 1 of the strategy cycle, Random Search picks the point randomly from the configuration space.

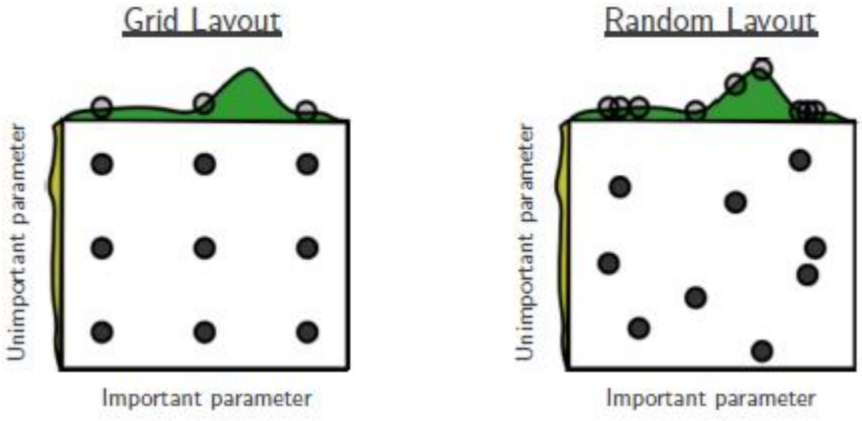


**Figure 2.15:** Grid search vs random search [34]

The image compares the two approaches by searching the best configuration on two hyperparameters space. It also assumes that one parameter is more important that the other one. This is a safe assumption because Machine Learning models are really full of hyperparameters, and usually we know which ones affect the training most significantly.

In the Grid Layout, it's easy to notice that, even if we have trained 9 models, we have used only 3 values per variable. Whereas, with the Random Layout, it's extremely unlikely that we will select the same variables more than once. It ends up that, with the second approach, we will have trained 9 models using 9 different values for each variable.

As we can tell from the space exploration at the top of each layout in the image, we have explored the hyperparameters space more widely with Random Search (especially for the more important variables). This will help us to find the best configuration in fewer iterations.

## 2.6 Proposed Approach:

There are many approaches to tackle the task of PV forecasting, for our case; we followed these steps:

1- Collect meteorological and geographical data from three different locations and climates; Florida (subtropical climate), Oregon (marine west coast climate), Colorado (semi-arid climate). So that the model can be robust and don’t “memorize” a specific location or specific climate, and it can be used in other locations and perform well (not just in the three locations used in training).

2- Preprocessing of the data by dealing with messing measurements, extracting useful features with the correlation test, and adding new features to the dataset.

3- Forecasting the irradiance using eight different models (seen earlier), fine tuning each one of them to get the best result from it, and then compare between models using the five evaluation metrices.

4- After forecasting the irradiance, and since PV output power is linearly related with it (as we will see), we can just use simple linear regressor to forecast the PV power using the irradiance.

We used the indirect approach to forecast the PV power so that the model can be robust and independent of the PV characteristics and technologies used.

In the following two chapters, we will dive in each step of this approach and discuss it in details.

## 2.7 Conclusion:

In this chapter we covered the fundamentals of machine learning as well as its different types, we gave some theoretical background about the various models that we used for the task of forecasting the PV generated energy, then we saw the evaluation metrics with which we evaluated the models and compared between them, after that we investigated how we can tune the models to get the best of them and which techniques to use, finally we explained briefly our proposed approach to forecast the PV generated power.

# CHAPTER 3:

# Data Analysis and Feature Engineering

Feature engineering is a topic that is absolutely known and agreed to be key to success in applied machine learning, in fact it is an understatement to say that it is a crucial step in the machine-learning pipeline as we can see in the figure below. Data preprocessing is the technique of transforming raw data into more meaningful data or the data which can be understood by the Machine Learning Model. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors, which explains the importance of data processing. In this chapter we will present the dataset we have been working with and the several steps we took in order to transform our data and select the optimal features [35][36][37].

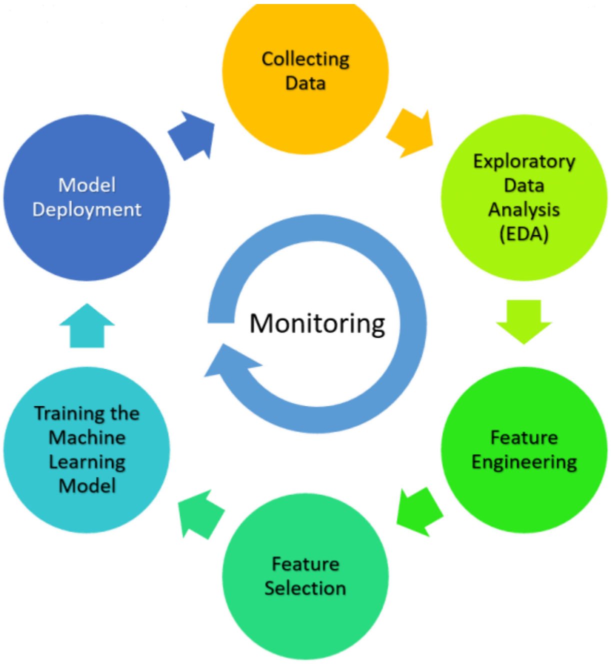


**Figure 3.1:** typical machine learning pipeline.

## 3.1 Importance of Feature Engineering:

The goals of feature engineering twofold:

- Process the input dataset in a way that is compatible with and suits the machine learning algorithm in an optimal way.
- Improving the performance of machine learning models

According to a survey in Forbes, data scientists spend 80% of their time on data preparation. The importance of feature engineering is realized through its time-efficient approach to preparing data that brings consistent output [38].

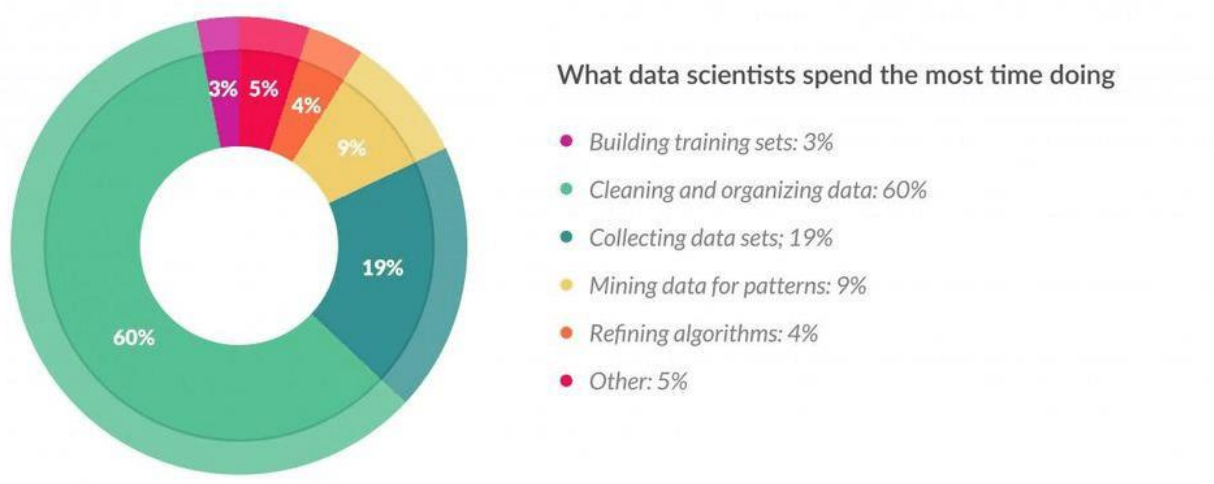


**Figure 3.2:** importance of data processing according to a survey in Forbes [39].

The result of feature engineering is an optimal dataset that will characterize all the essential factors and complex trends of problem. This dataset would in turn yield the best possible performance when fed to models.

In other words, the better the prepared features we select, the better the results the model will achieve. It is true, but it also misleading. Since there is a complex relation between number of features and accuracy more features do not always mean better accuracy since using too many features will likely render your model prone to overfitting. The results are a factor of the model, the available data and the features we prepared. Even the framing of the problem and objective evaluation metrics used to estimate accuracy play a part. the results depend on many inter-dependent properties.

## 3.2 Dataset presentation:

The Dataset we were able to collect contains meteorological data for photovoltaic (PV) modules representing all flat-plate PV technologies modules installed in three different locations and climates for approximately one-year periods observed in 5-minute interval. The data measurement locations were Cocoa, Florida (subtropical climate); Eugene, Oregon (marine west coast climate); and Golden, Colorado (semi-arid climate). The data sets were produced by the National Renewable Energy Laboratory under the Systems Integration Subprogram, which is funded and monitored by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy.

The data include a wide range of irradiance and temperature conditions representing each season for each location that cover the following periods:

- Cocoa – January 21, 2011, through March 4, 2012
- Golden – August 14, 2012, through September 24, 2013
- Eugene – December 20, 2012, through January 20, 2014.

## 3.3 Data measurement and equipment:

Measurement equipment was selected to provide low measurement errors. Data acquisition equipment was located inside a temperature-controlled environment. The sensors and data acquisition equipment are listed in the following Table. Equipment was selected to provide low measurement errors, with special attention paid to the selection of the solar radiation instrumentation, which is usually the largest source of error when measuring the performance of PV modules or systems. Moreover, to ensure PV modules were not excessively dirty, rendering the data questionable, all the PV modules were cleaned if judged necessary.

| Item | Parameter | Instrument |
|---|---|---|
| 1 | Wind Speed/Wind Direction/ Precipitation/Temperature/ Relative Humidity/Barometric Pressure. | Vaisala WXT520 Weather Sensor. |
| 2 | Plane-of-Array Irradiance. | Kipp & Zonen CMP 22 pyranometer LI-COR pyranometer. |
| 3 | PV Module Back-Surface Temperature. | Omega Model CO1-T Style I Thermocouple. |

**Table 3.1:** List of Sensors and Data Acquisition Equipment.

- The Vaisala Weather Transmitter WXT520 is a compact and lightweight multi-sensor instrument that measures the most essential weather parameters. It is a configurable product that can measure wind speed and direction, liquid precipitation, barometric pressure, temperature and relative humidity all in one transmitter [40].

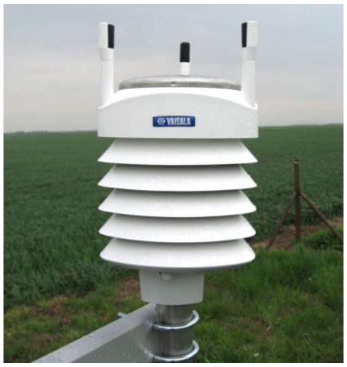


**Figure 3.3:** Vaisala Weather Transmitter WXT520 sensor [40].

- Kipp & Zonen CMP 22 pyranometer LI-COR pyranometer is one of the bets pyranometers in the market, it is a type of actinometer used to measure broadband solar irradiance on a planar surface and is a sensor that is designed to measure the solar radiation flux density (in watts per meter square) from a field of view of 180 degrees.

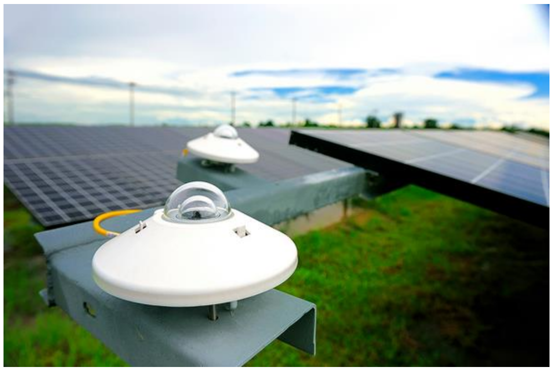


**Figure 3.4:** Kipp & Zonen CMP 22 pyranometer LI-COR pyranometer.

- Omega Model CO1-T Style I Thermocouple in other words it is a sensor that measures temperature. It consists of two different types of metals, joined together at one end. This model is a Type T thermocouple it is considered to be the best thermocouple to measure temperature, since it is very stable and the most accurate type of thermocouple

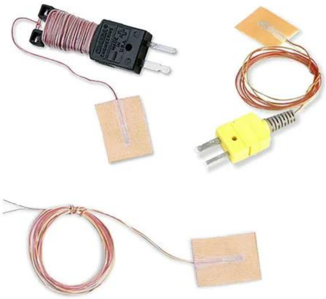


**Figure 3.5:** Omega Model CO1-T Style I Thermocouple.

## 3.4 Data preprocessing:

Machine learning in a nutshell consists of training and feeding data to algorithms to perform various computationally intensive tasks. The main challenge engineer may face while processing the data resides in feeding the right data to machine learning algorithms or cleaning of irrelevant and error-prone data. As mentioned earlier when it comes to utilizing ML data, most of the time is spent on cleaning data. In other words, Data cleaning is an important aspect of machine learning (ML). It involves ensuring that the right data is fed to the right algorithms and that the datasets are free of errors. Setting up a quality plan, filling missing values, removing rows, reducing data size are some of the best practices used for data cleaning in Machine Learning [41].

### 3.4.1 Data Cleaning:

The raw data present in the datasets needed to be processed and polished, it contained over 44 parameters, for our objective we would not need all of them hence a feature selection is mandatory, moreover as expected there were some missing data in certain columns the missing data was indicated by -9999 in fact about 25% of the meteorological data are missing in golden. Data may be missing due to a failure to meet QA thresholds or equipment problems, in addition to special type data as ‘timestamp’ that represents the date and time of the measurements in this section we will discuss the undertaken steps to deal with the data.

- **Missing data:** One of the first steps of fixing errors in a dataset is to find incomplete values and fill them out, in such cases, a common approach is to directly drop the rows with missing data, in our case since we are working with meteorological data and since the measurements are taken in a 5min interval the changes could be abrupt hence we decided to take the average of each 64 samples (approximately 5 hours) and use this value to fill the missing data. This means that missing values are filled with the mean of the 5 hours slice of the dataset they belong to.
- **Normalization:** Normalization is a scaling technique that moves values to a certain range so that they end up ranging between zero and 1. Normalization can be described by the following equation:

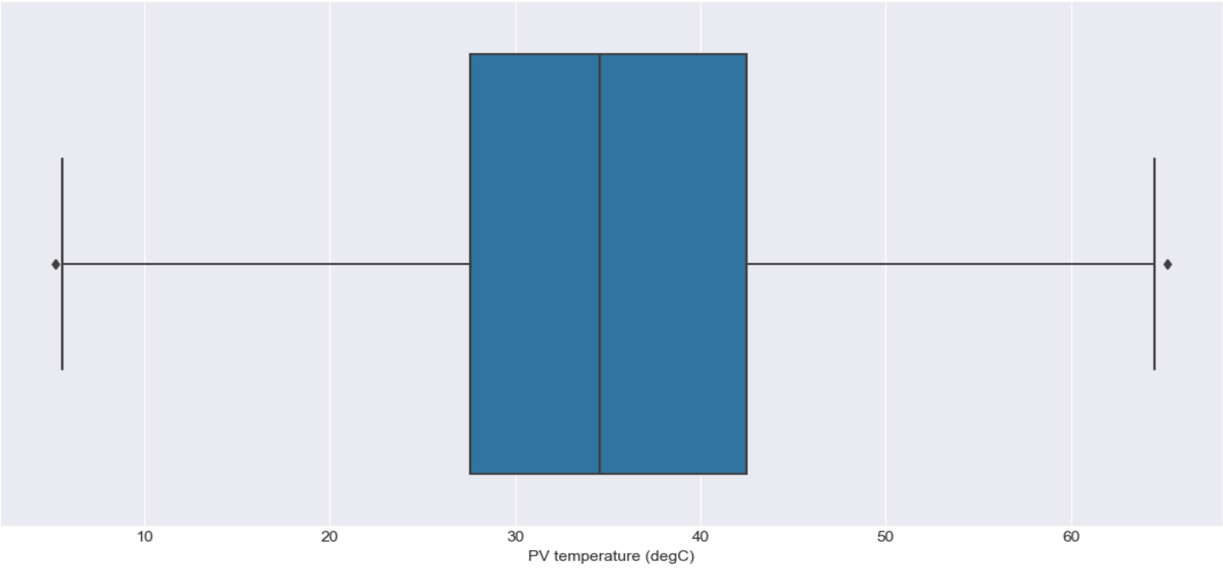
$$X' = \frac{X - Xmin}{Xmax - Xmin} \quad \ldots\ldots (3.1)$$

- **Standardization:** Similar to normalization, standardization is another feature scaling method, this technique works by focusing on the mean values with a unit standard deviation, in other words the mean becomes 0 and the deviation is present in both sides (positive and negative). Standardization can be described by the following equation:

$$X' = \frac{X - \mu}{\sigma} \quad \ldots\ldots (3.2)$$

$\mu$ is the mean of the feature values and $\sigma$ is the standard deviation of the feature values.

- **Dealing with Outliers:** An outlier is a data point that is noticeably different from the rest. They represent errors in measurement, bad data collection, or simply show variables not considered when collecting the data. Wikipedia defines it as 'an observation point that is distant from other observations'. Outliers threaten to skew your results and render inaccurate insights. We thoroughly analyzed the dataset for outlier values by plotting the box plot of data for each feature over the period of measurement. as expected, there was some outliers present in certain parameters such as precipitation and PV temperature but nothing that could harm the accuracy of the predictions. Especially since most of the ‘Extreme values’ seem to be present in the expected seasons of the year, i.e., outliers in ‘PV temperature’ are located in winter and summer as one would expect [42][43].



-

**Figure 3.6:** Box Plot of PV temperature.

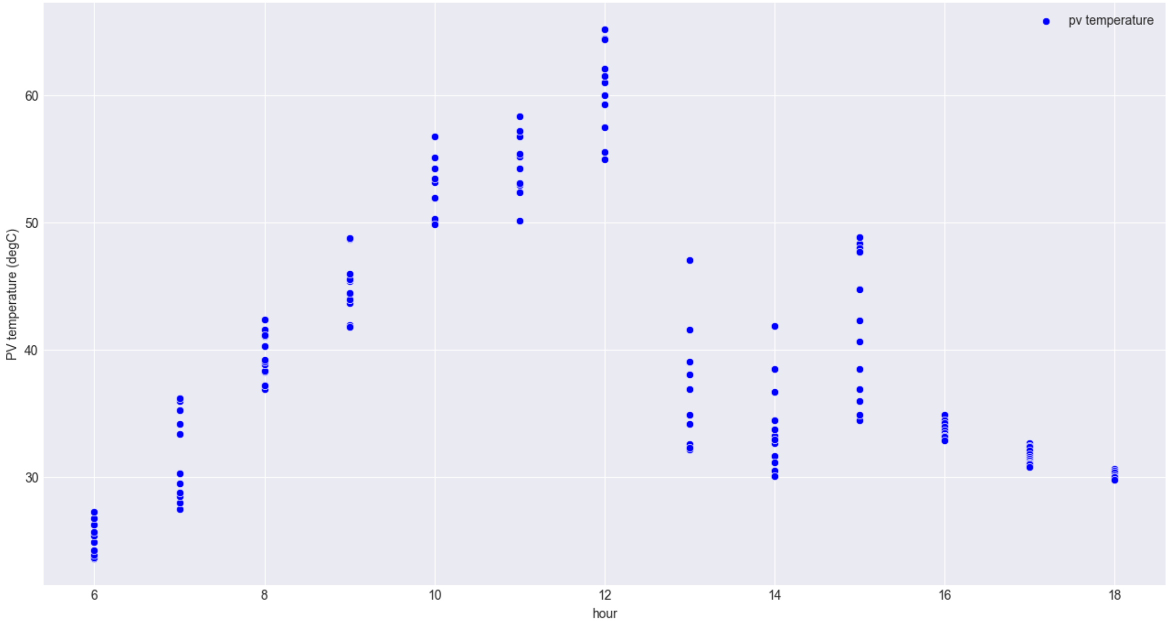


**Figure 3.7:** outlier day of PV temp.

The above figures represent respectively a box plot and a scatter plot of our ‘PV temperature’ feature present in our dataset. From the box plot we can clearly observe that this feature has two outliers one present roughly at PV temp 65° and another at 5°, after further examination we determined the day where the outlier with higher temperature occurred, after plotting the scatter plot of we can clearly observe that that value was not only recorded at noon but also the day itself was in the summer season. With these conditions such value can only be expected hence this cannot be considered as an outlier.

### 3.4.2 Feature construction:

at first glance at the dataset, we noticed the absence of geographical data that can be valuable for out model to improve its accuracy, fortunately the location of the PV plant was provided, after some manipulation and analysis we were able to construct new features such as: sunset and sunrise time, daylight duration, zenith and azimuth angles, and latitude and longitude, declination angle. Furthermore, we noticed that recorded time of each measurement was stored in a column with a special type of data known as ‘Timestamp’, this type of data cannot be fed to the model therefore after some tweaking, we split it into day, month, year, hour, minute, second. However, since our data was measured each 5minutes features like year/month/day would be redundant and would later create a bias in our predictions. In order to tackle this issue, we decided to create one new feature from the three mentioned above hence we created a new feature named ‘day_of_year’ which basically is a sequential day number starting with day 1 on January 1st.

### 3.4.3 Data visualization:

plotting the data and examining them is an important part of data analysis it helps quickly detect abnormalities and analyze trends and behavior our data. We will visualize the features individually in the first place then we will observe their impact on the irradiance.

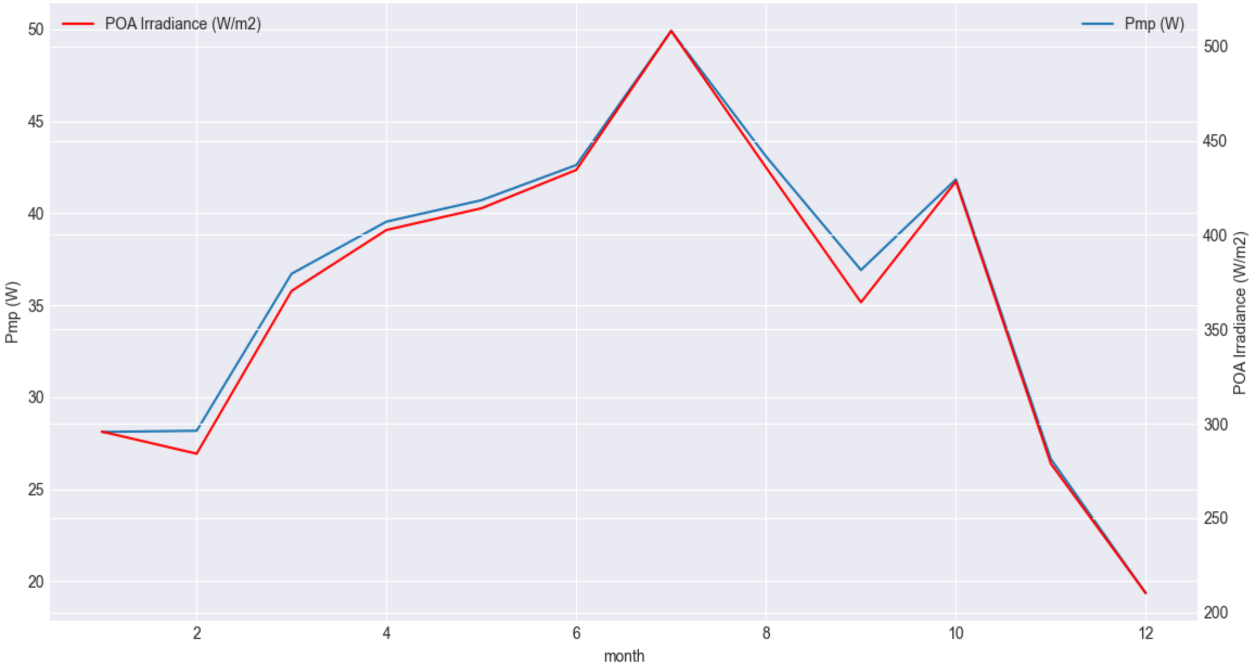


**Figure 3.8:** irradiance and generated power line plot.

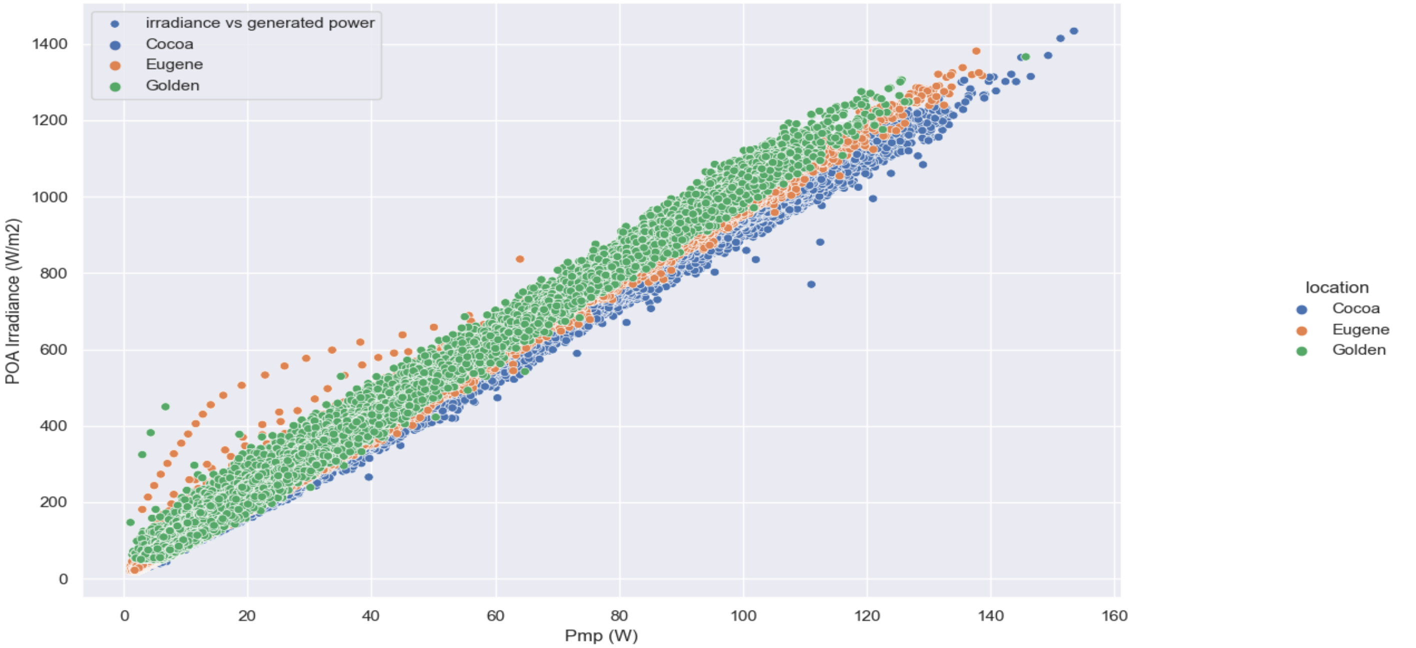


**Figure 3.9:** irradiance vs generated power scatterplot.

The above plots depict the evolution of irradiance and generated power, we can clearly observe the evident linear relationship between the irradiance and the PV output power. This was the main reason behind our proposed approach to first forecast the irradiance then with that result predict the generated power using a simple regression system.

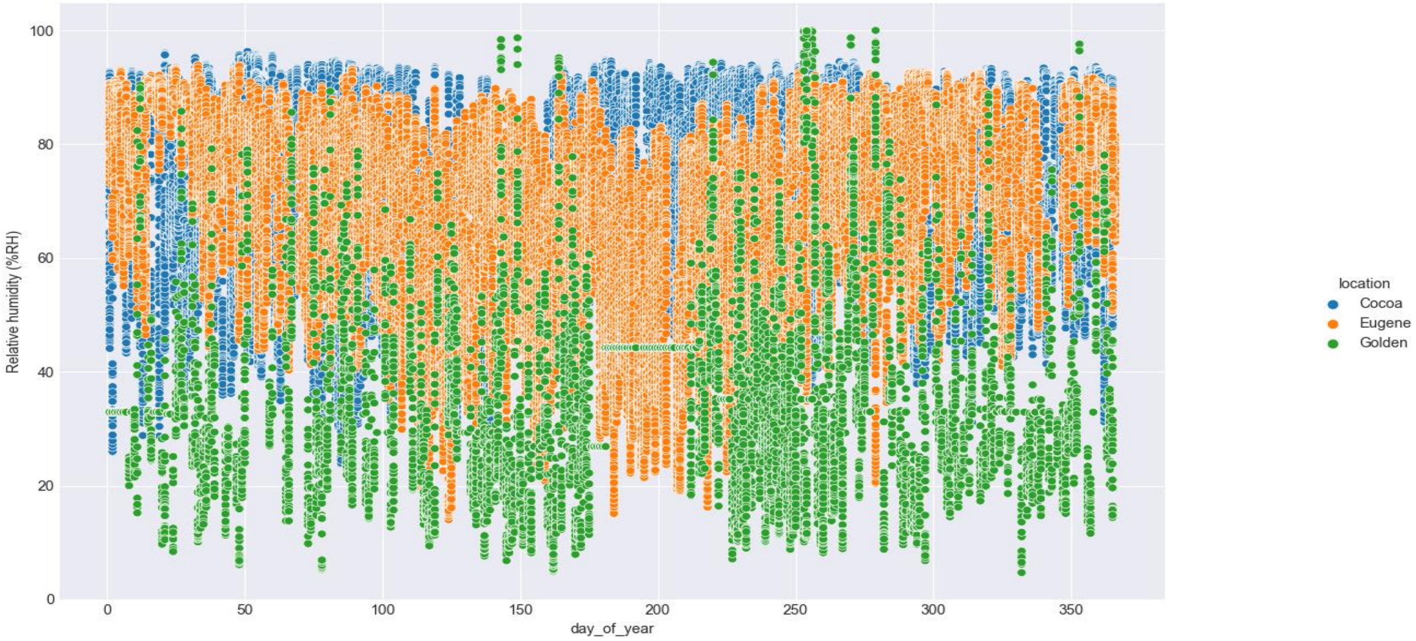


**Figure 3.10:** Evolution of Relative Humidity during the year per region scatterplot.

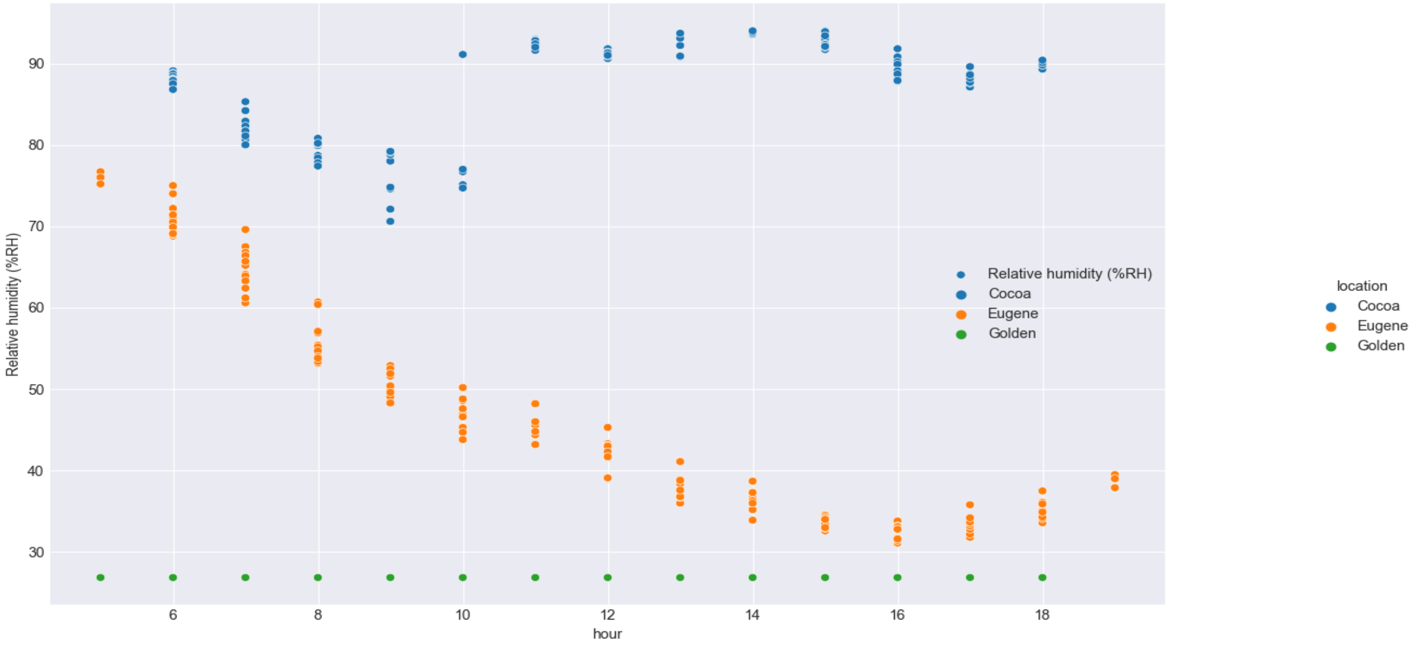


**Figure 3.11:** Evolution of Relative Humidity during a random day per region scatterplot.

The first figure illustrates the Evolution of humidity with respect to day of the year for each region, while in the second figures gives a more detailed idea of the trends in each area since it represents humidity vs hour for a randomly selected day, golden settling at about 27% RH while Eugene's humidity is steadily falling. Cocoa on the other hand experienced fluctuations in its level of humidity. We can observe 3 different trends each proper to a certain area this would have a beneficial effect on our model later since it would help the model to become robust.

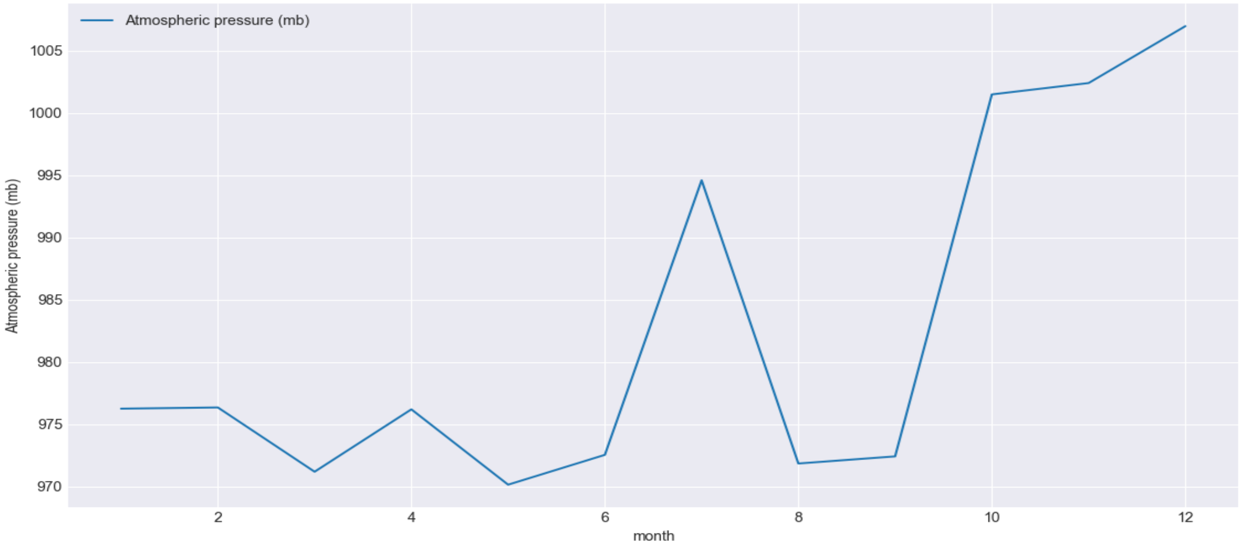


**Figure 3.12:** Mean Atmospheric Pressure for the 3 regions during the year 2013-line plot.

The above line plot illustrates the mean atmospheric pressure for all the regions during the year 2013. at first glance we can notice the peak occurring in winter rather than in summer this is caused by the relatively cold conditions experienced in this region compared to summer. Since Cold air is denser than relatively warm air, it has a tendency to sink, and thus has a higher pressure than warm air. From this, it can be gleaned that the winter season would have higher average atmospheric pressure than during summer.

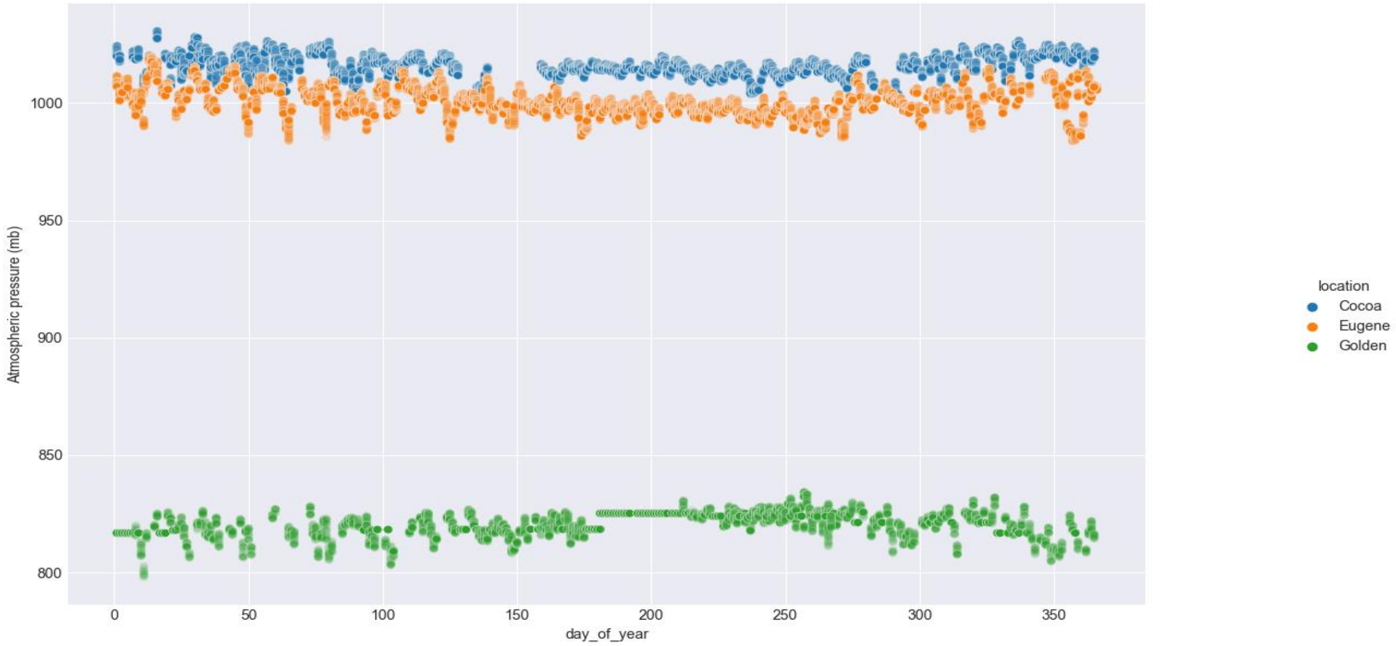


**Figure 3.13:** Atmospheric Pressure for each region during the year 2013 scatterplot.

The above scatterplot provides more information on each region in terms of Atmospheric pressure during the year 2013, we can observe two different clusters the first one regroups the regions of Cocoa and Eugene who shares roughly the same trends and values, while the second cluster includes Golden has significantly less atmospheric pressure than the other regions.

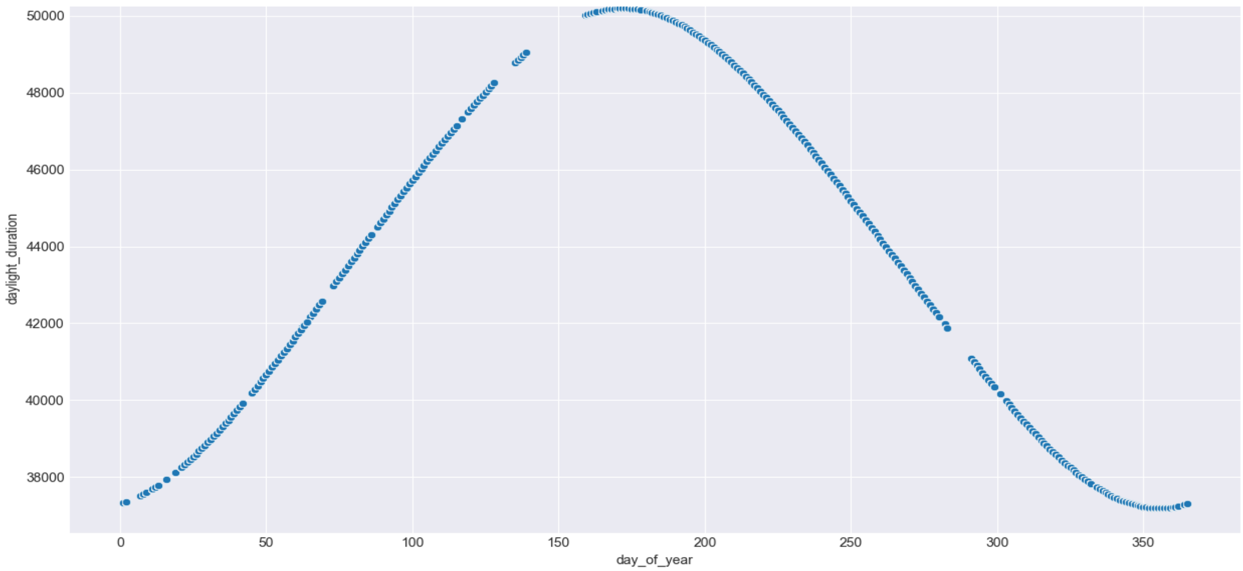


**Figure 3.14:** Evolution of daylight duration during the year 2013 scatterplot.

Daylight duration is one of the features we constructed and added to the dataset, in order to construct this feature, we used the geographical locations, the time zone and city information as input for a python library that adapt its calculations for Python, from the spreadsheets provided by the National Oceanic and Atmospheric Administration (NOAA), after carefully verifying the validly of the results we got, we can observe the daylight duration in seconds for a sinusoidal curve during the year 2013 which is expected considering the nature of the movement of the earth around the sun.

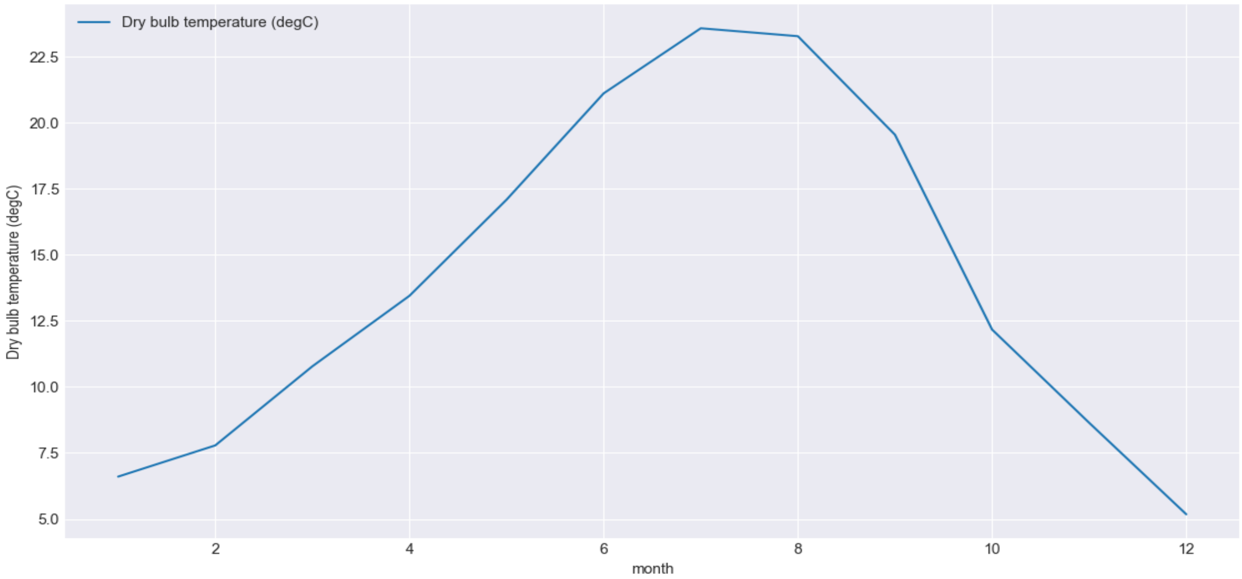


**Figure 3.15:** Evolution of the mean Dry Bulb Temperature during the year 2013-line plot.

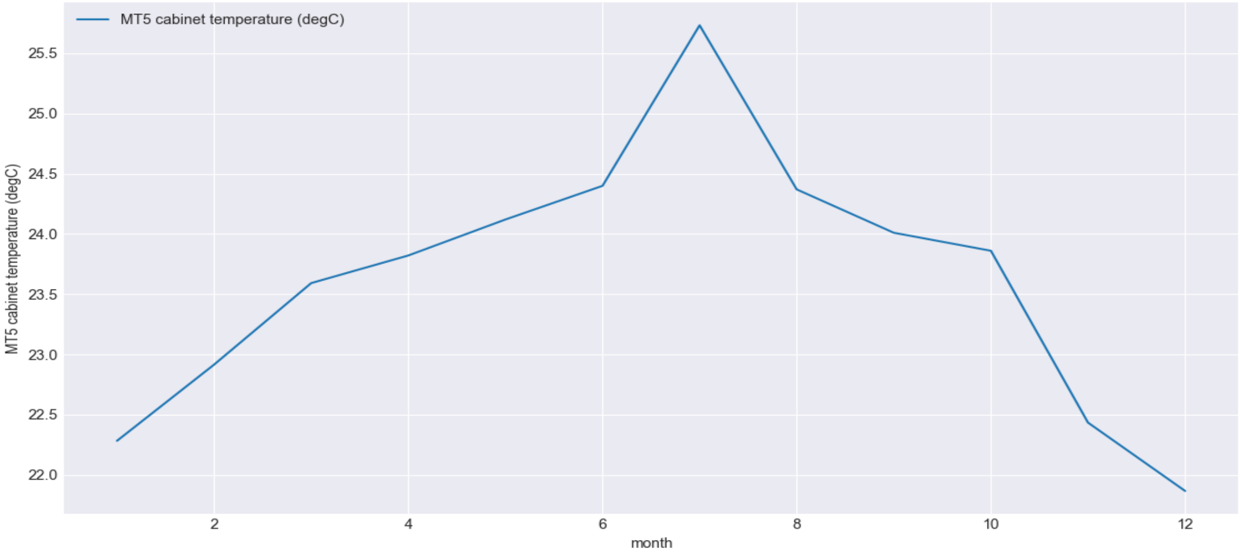


**Figure 3.16:** Evolution of the mean MT5 cabinet temperature during the year 2013.

The above graphs represent the fluctuation in ‘Dry bulb temperature’ and ‘MT5 cabinet temperature’ respectively during the year 2013. As expected, both features peak during summer with a maximum of 27.5°C for dry bulb temperature in July and 25.75°C for MT5 cabinet temperature that occurred also in July.

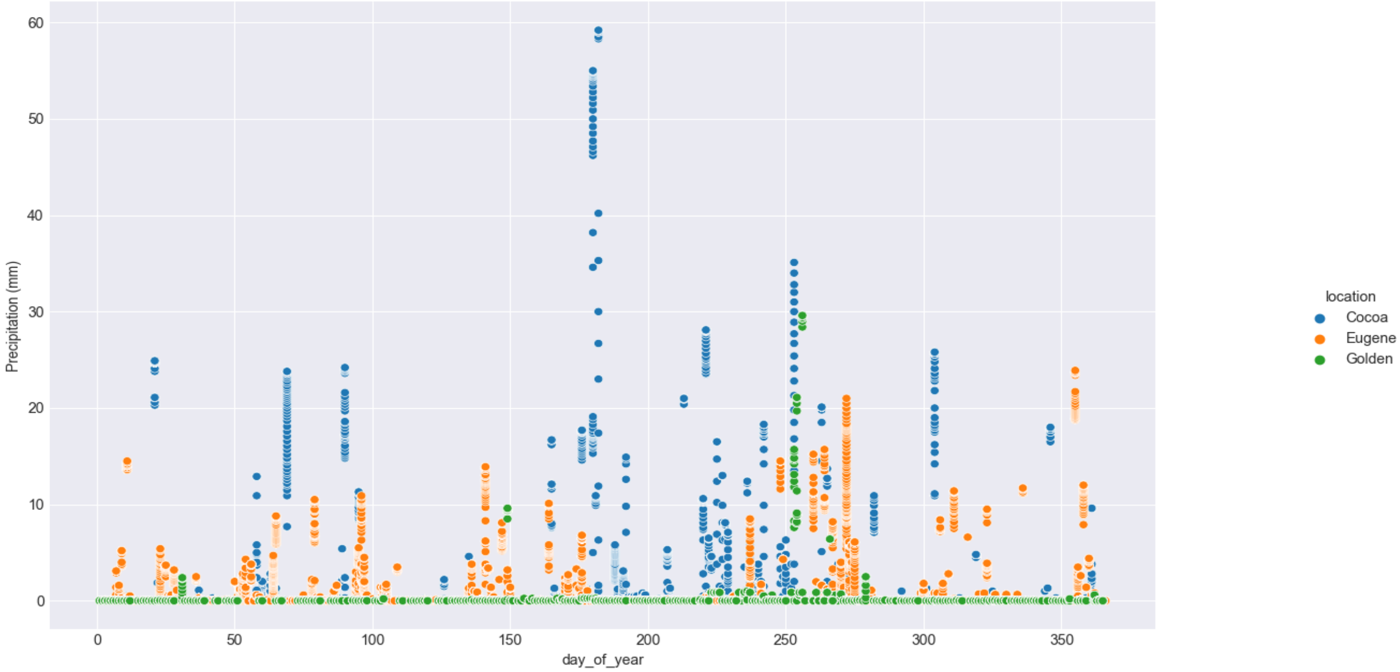


**Figure 3.17:** Evolution of precipitation in each region during the year 2013.

The above scatter plot illustrates the precipitation in each region during the year 2013. We can clearly see that each region differs from the others. Cocoa experienced a rainy couple of days in the early and late summer season. Golden went through a dry year which is understandable since it is located in a desert. Finally, Eugene had a relatively stable year in terms of precipitation.

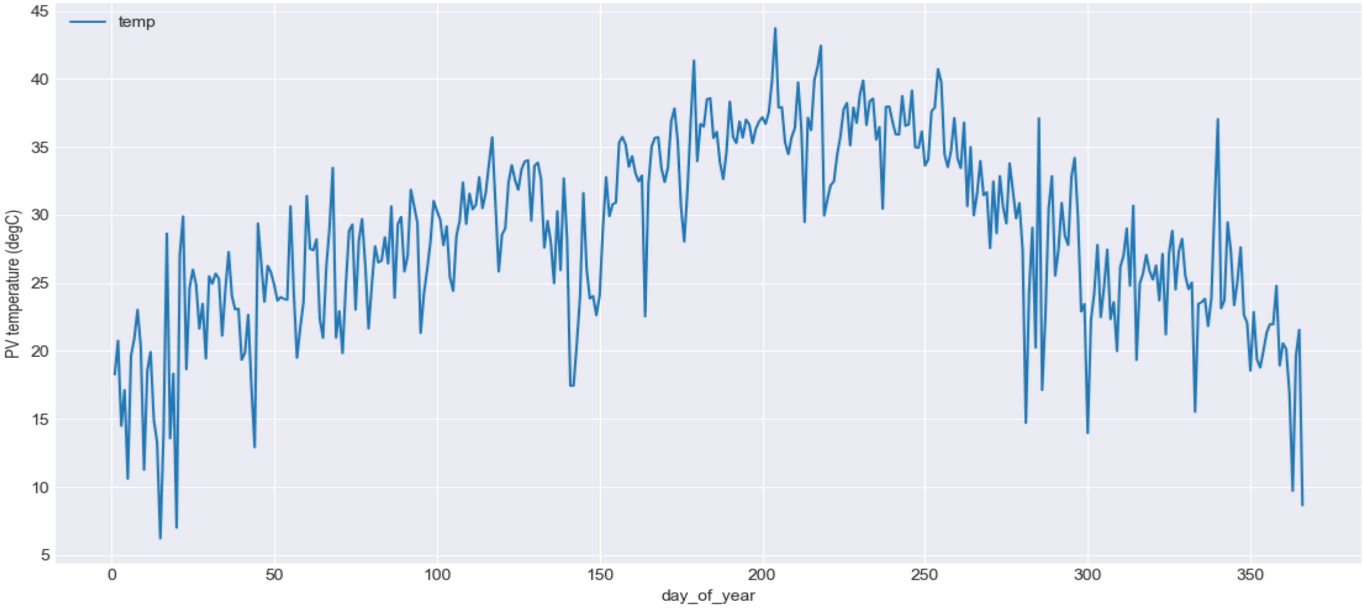


**Figure 3.18:** Evolution of the mean PV Temperature during the year 2013.

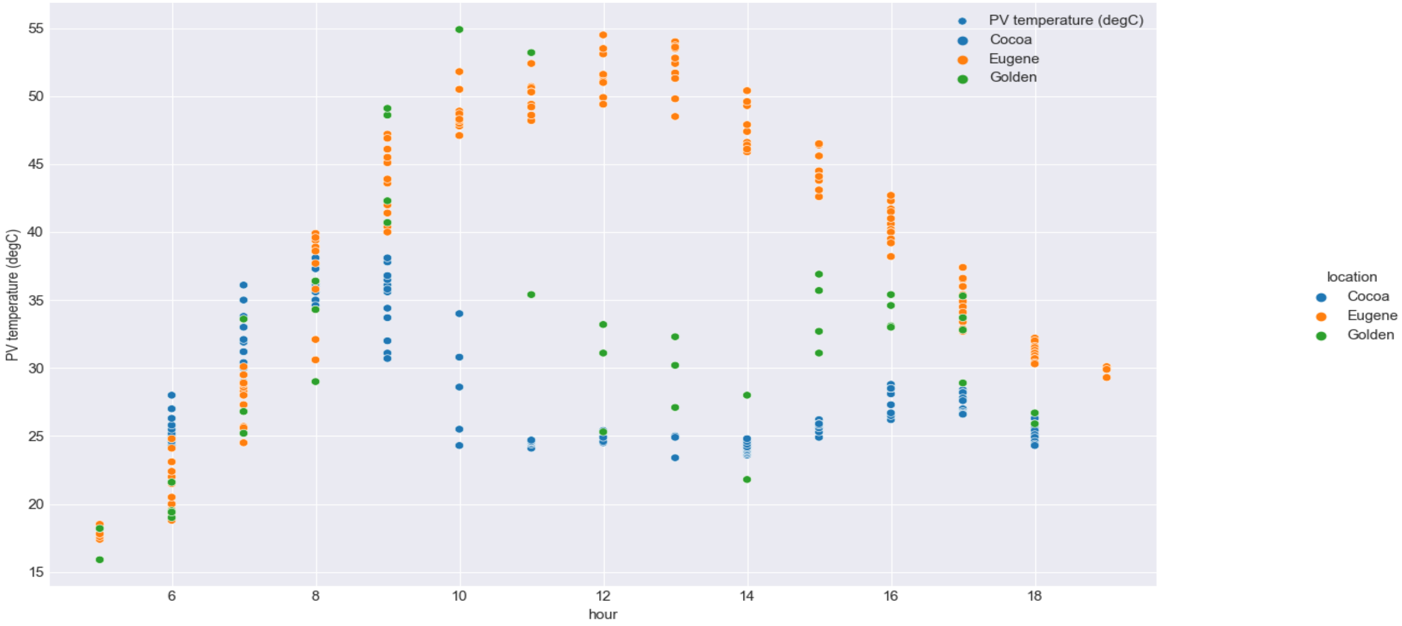


**Figure 3.19:** Evolution of temperature during one day in three different locations

The two above plots provide information on the PV temperature during the year 2013 and during a randomly chosen day respectively as expected the peak appear to occurs in the summer season and more specifically from 10AM to 15PM.

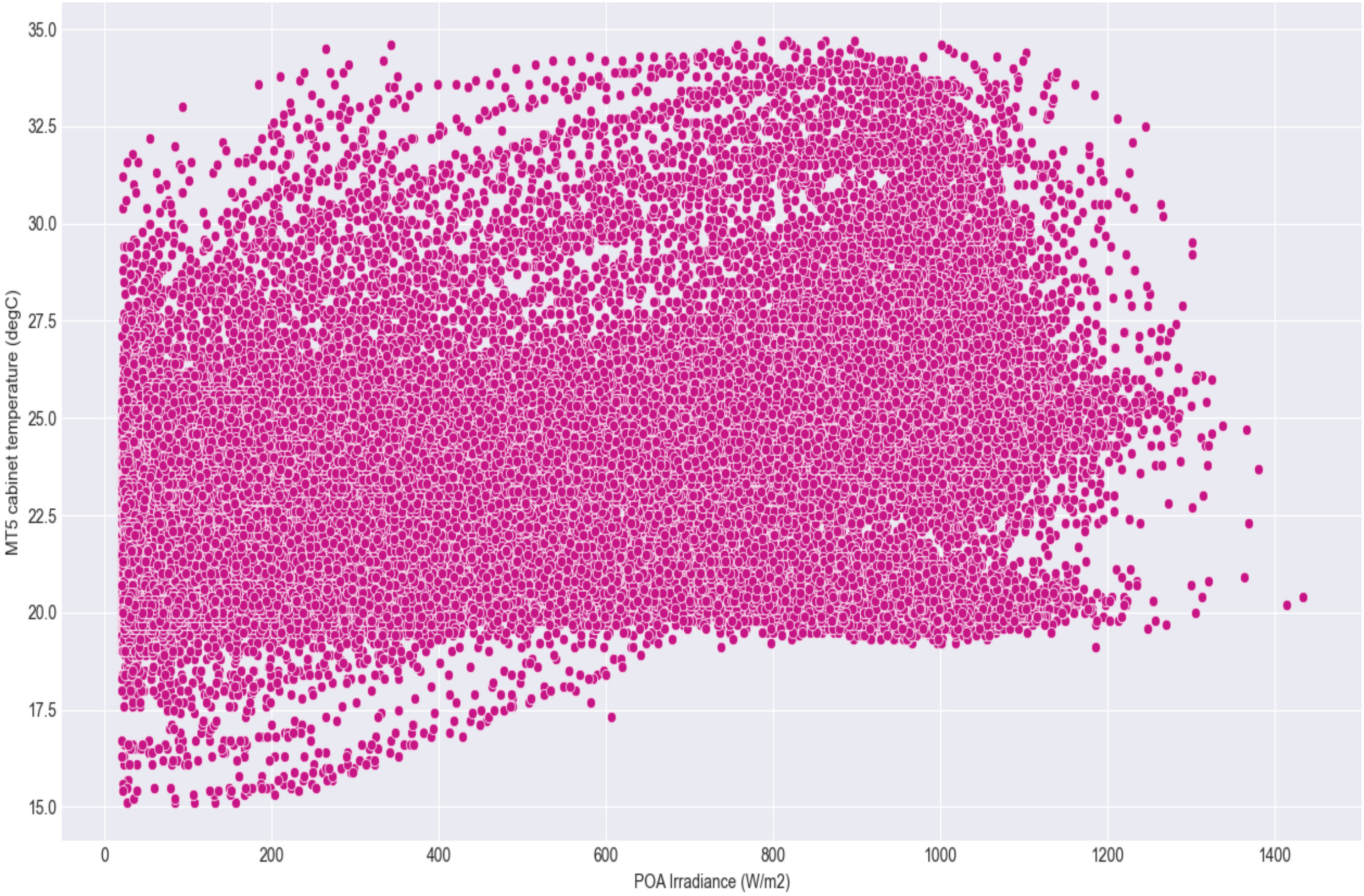


**Figure 3.20:** Irradiance vs MT5 cabinet temperature Scatterplot.

In the above Scatter plot, we can observe the relationship between the irradiance and MT5 cabinet temperature, this relationship is a positive one, since the irradiance seem to increase as the MT5 cabinet temperature increases, but this will be further developed later in this chapter.

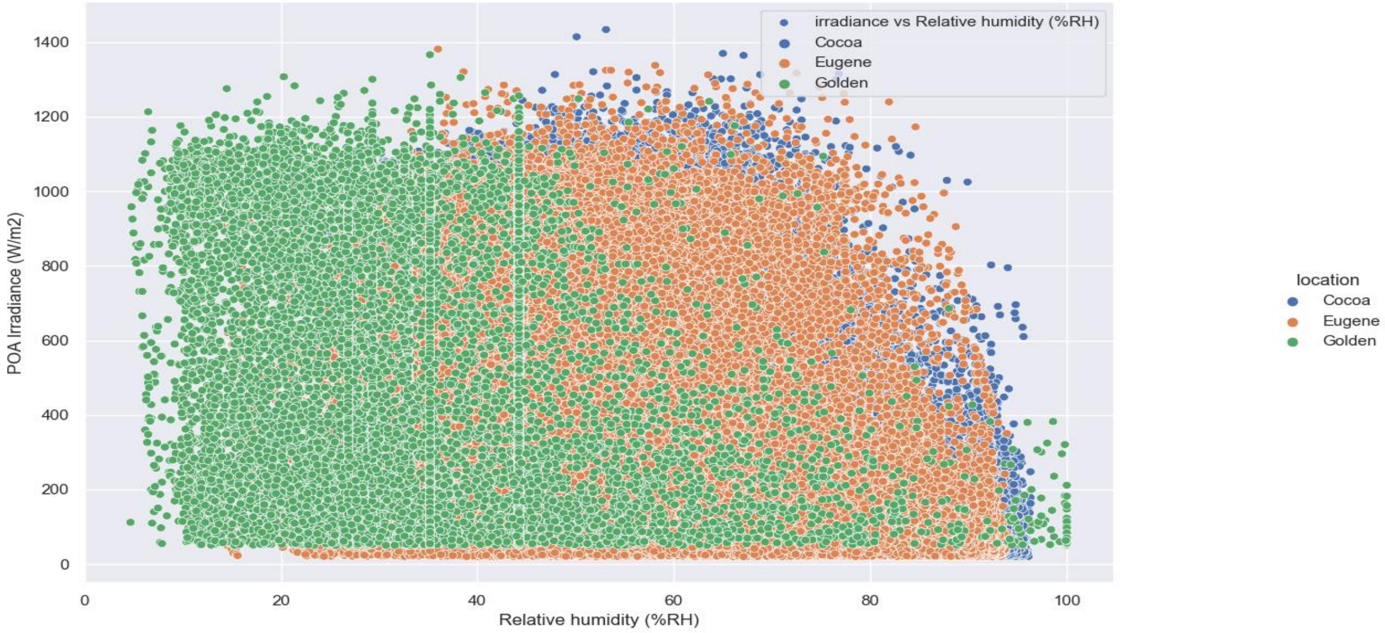


**Figure 3.21:** Irradiance vs Relative Humidity Scatterplot.

We can see in the previous figure that the correlation between the irradiance and the relative humidity is negative, the irradiance tends to be higher when the humidity is low. In other words, when there is high relative humidity (wet period), solar radiation will be low while in dry season, the solar irradiance will be high, showing that relative humidity has a negative effect on solar irradiance.

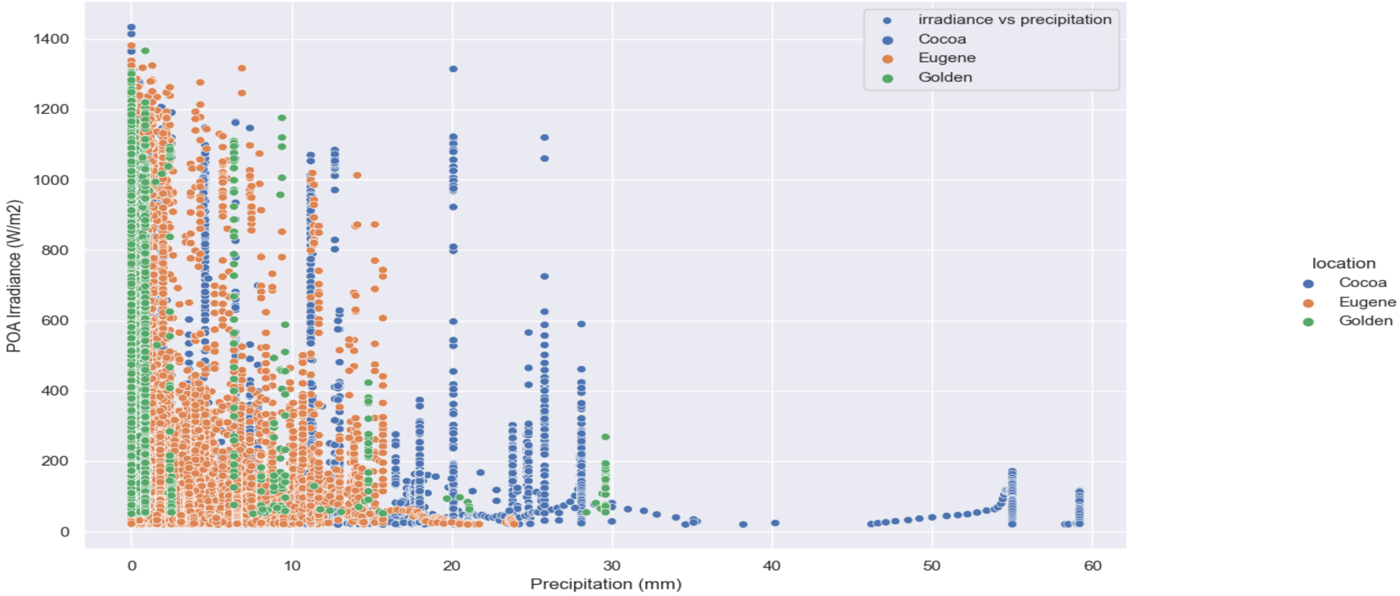


**Figure 3.22:** Irradiance vs Precipitation Scatterplot.

Using a regression plot we can clearly observe the negative relationship between the precipitation and the irradiance. With some rare exceptions in Cocoa, it is clear that precipitation and irradiance are not positively correlated.

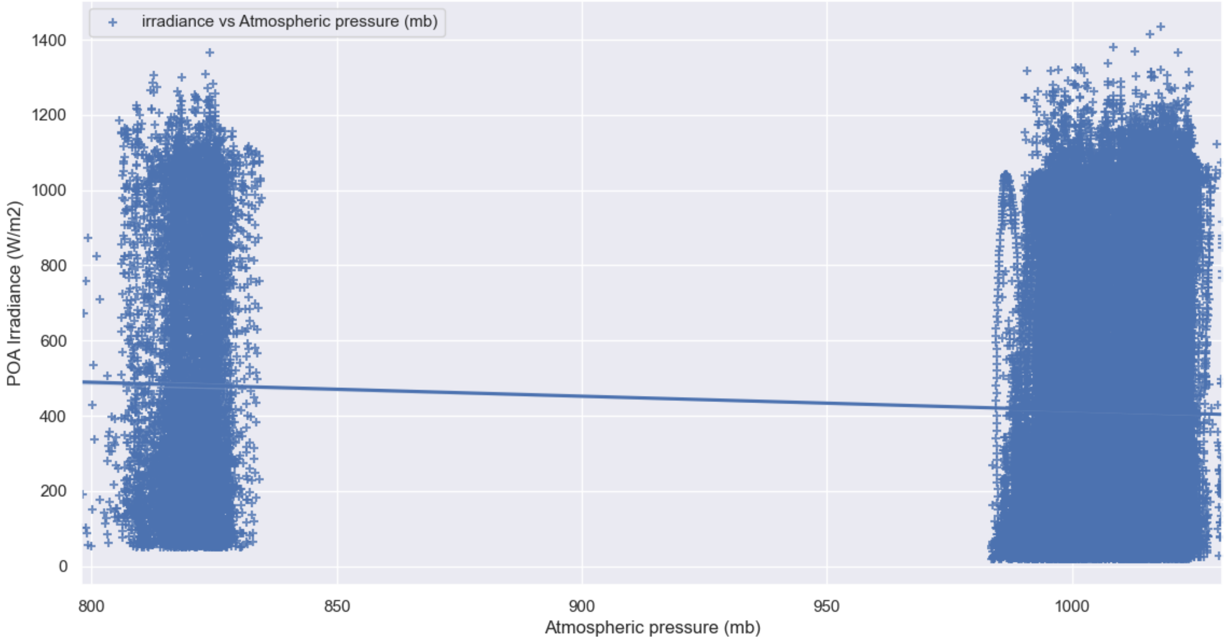


**Figure 3.23:** Irradiance vs Atmospheric Pressure Regression plot.

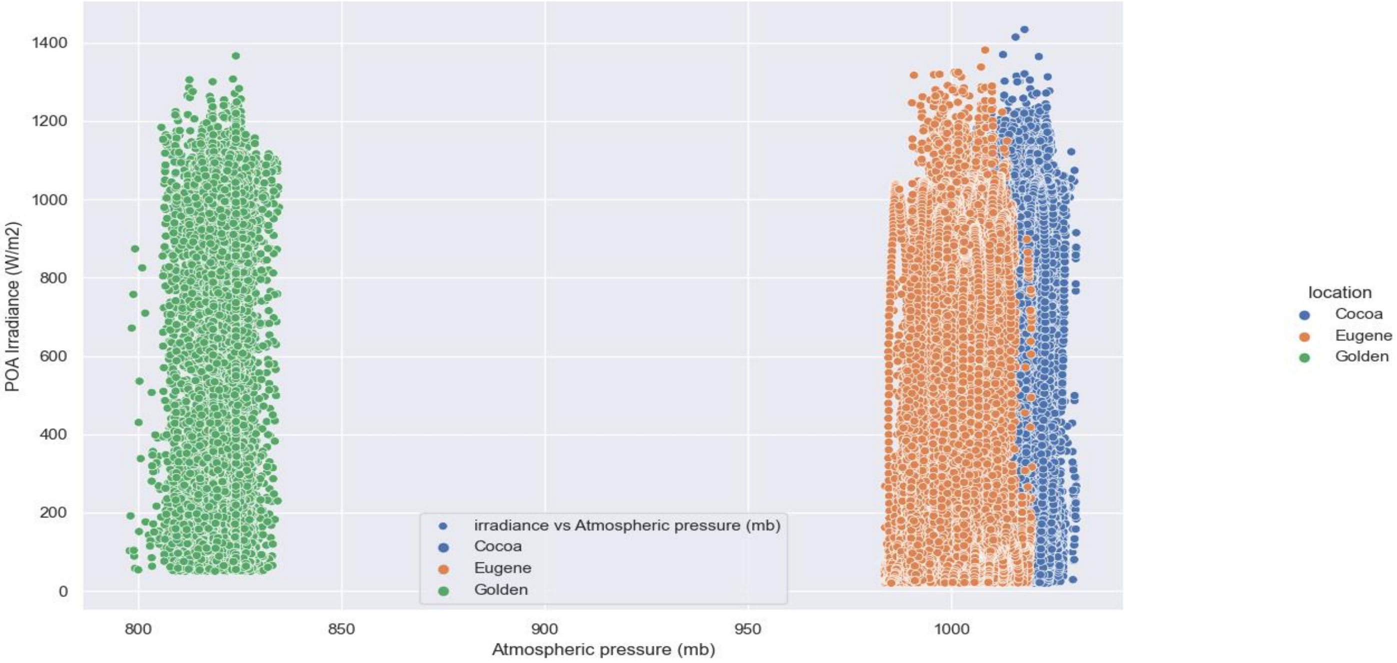


**Figure 3.24:** Irradiance vs Atmospheric Pressure Scatterplot.

Again using a regression plot we can clearly observe the negative relationship between the atmospheric pressure and the irradiance, we can also notice that as mentioned before the measurement plant located in ‘Golden’ has a different atmospheric pressure than the other two locations this will be render our model robust and will help them adapt more.

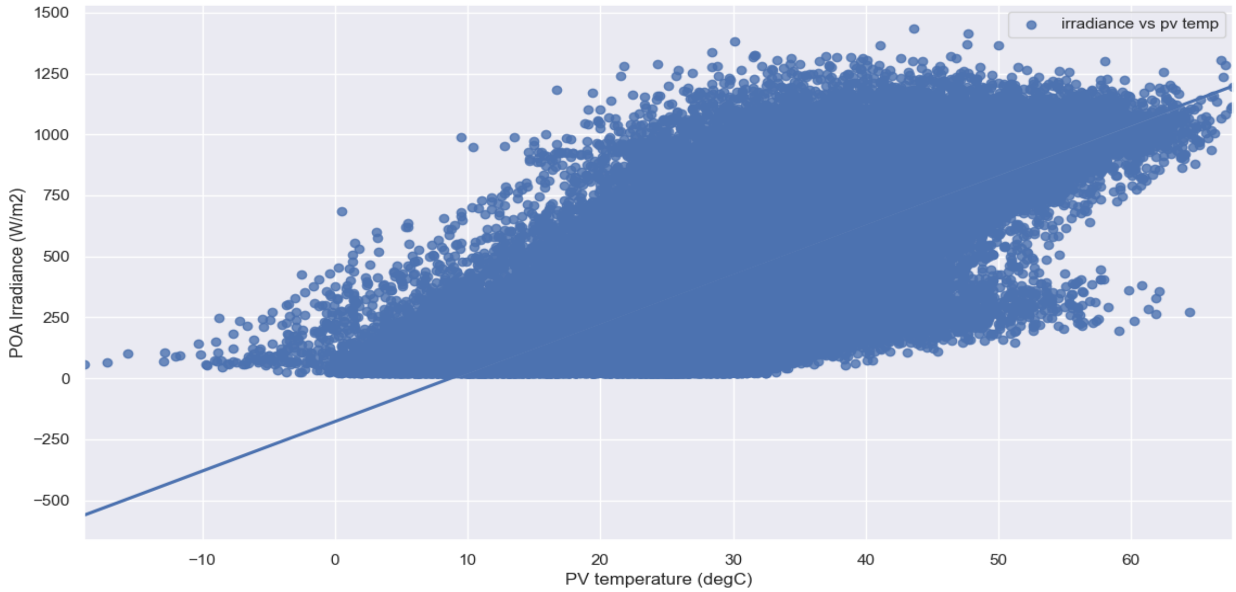


**Figure 3.25:** Irradiance vs PV Temperature Regression plot.

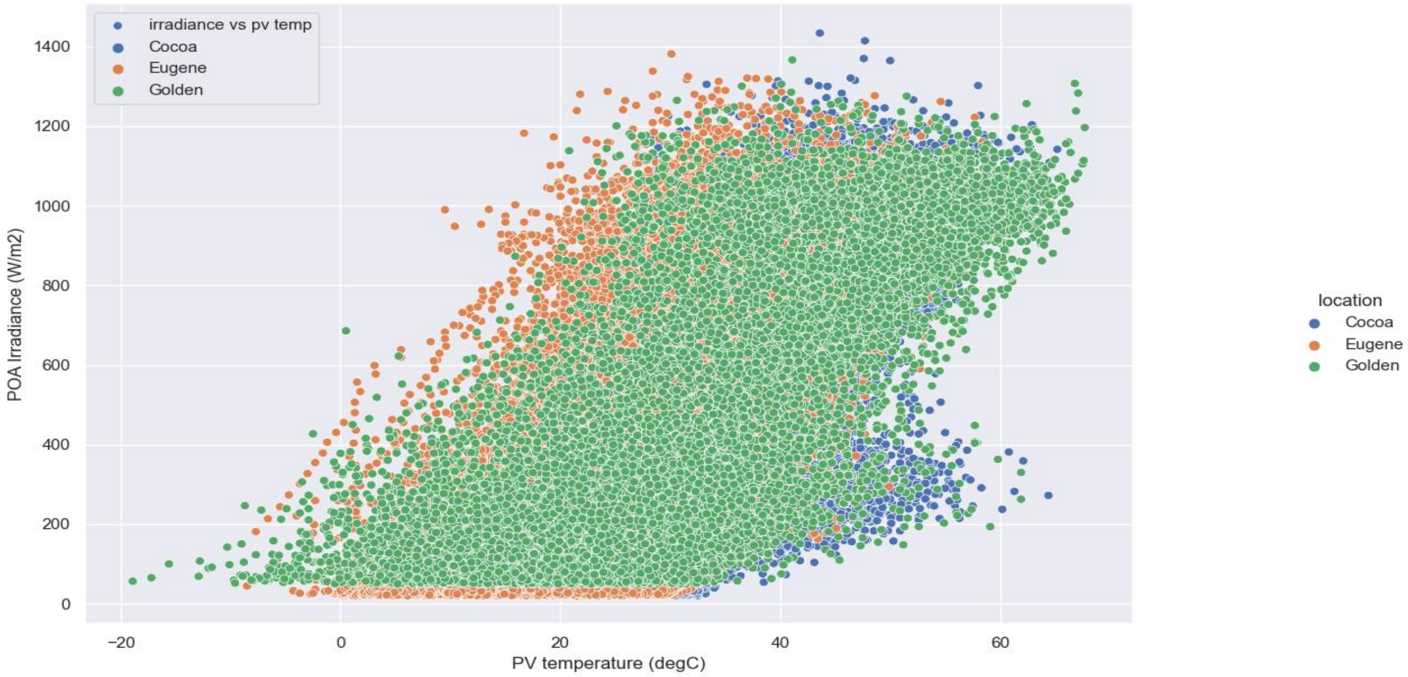


**Figure 3.26:** Irradiance vs PV Temperature Scatterplot.

According to the above graphs Irradiance and PV Temperature have a positive linear relationship and seem to be highly correlated, in fact we can clearly see the positive slope of the regression between the PV temperature and the irradiance.

### 3.4.4 Correlation Matrix:

Apart from the data quality verifications a second important analysis performed on the acquired datasets was to assess the importance of the input features on the output by investigating the correlation between the different parameters. Any typical machine learning or deep learning model is made to provide a single output from huge amounts of data be it structured or unstructured. These factors may contribute to the required result at various coefficients and degrees. They need to be filtered out in a way based on their significance in determining the output and also considering the redundancy in these factors. Correlation is a statistical measure that indicates the extent to which two or more variables fluctuate together. In simple terms, it tells us how much does one variable changes for a slight change in another variable [44]. Correlation between 2 variables can be found by various metrics such as Pearson r correlation, Kendall rank correlation, Spearman rank correlation, etc. Pearson r correlation is the most widely used correlation statistic to measure the degree of the relationship between linearly related variables. The Pearson correlation between any 2 variables x, y can be found using:

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\,\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

The resulting correlation matrix is:

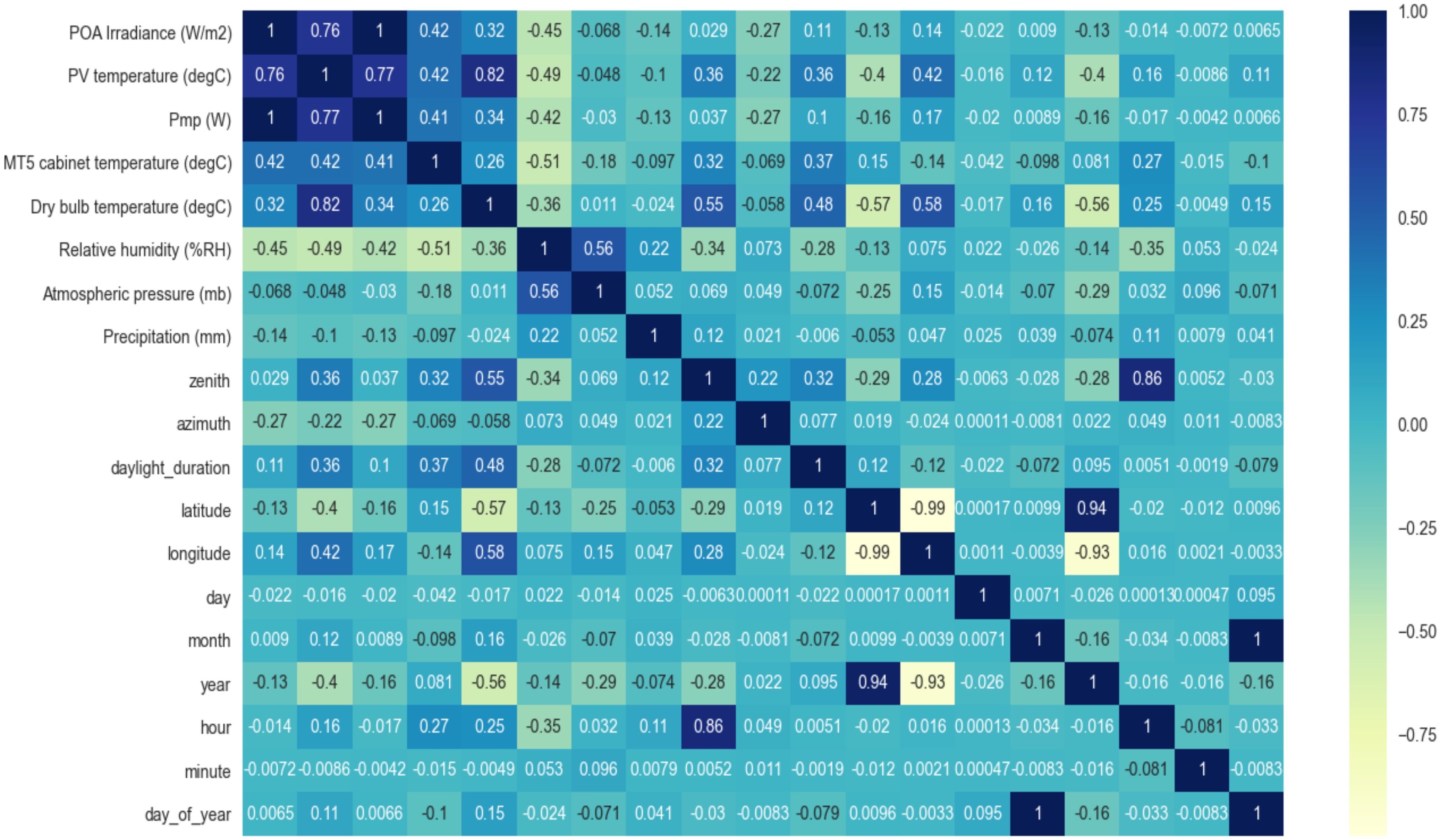


**Figure 3.27:** Correlation matrix

### 3.4.5 feature selection:

After analyzing the input features and studying the correlation, the insignificant weather parameters were removed from the input dataset and the final set of features were fed as an input to the model. For the first model the target variable was solar power as for the second model the target variable was generated power. After sensitivity analysis, We can clearly observe that there is a strong correlation between irradiance and generated power which makes it easy for us to predict the generated power once we have predicted the irradiance, moreover it is important to note that the positive correlation between the 'MT5 cabinet temperature', the 'PV temperature' and the 'dry bulb temperature' (Dry bulb temperature at the site) as well as day light duration with irradiance, even though that correlation seem to be low. We also notice a negative relationship between ‘relative humidity’, ’atmospheric pressure and ‘precipitation’ which was expected knowing the nature of those features. As we already know time of the day (hour) and day of year are very important for forecasting and yet the correlation is almost inexistent, this can be explained by the fact that they possess a high non-linear correlation with irradiance and PV system output. Therefore, we cannot rely solely on linear correlations to decide the input parameters for model building [45]. Taking all this into consideration the final set of parameters we decided to keep is:

1. **Plane-of-Array (POA) Irradiance [As Target Value for First Model]:** Amount of solar irradiance in watts per square meter received on the PV module surface at the time indicated. It comprises all types of irradiances being received at the Earth's surface i.e. direct irradiance, diffuse sky irradiance and reflected irradiance.

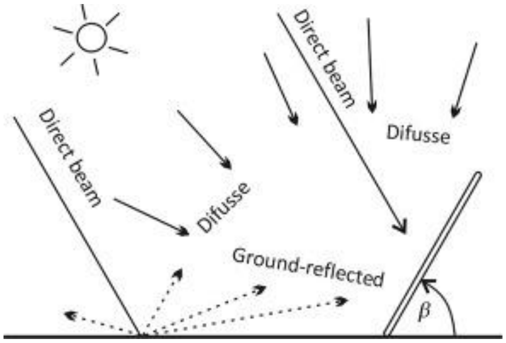


**Figure 3.28:** different types of irradiances at ground level.

2. **Generated power Pmp (watts) [As Target Value for Second Model]:** Maximum power of PV module in watts at the time indicated.
3. **PV Module Back-Surface Temperature:** PV module back-surface temperature in degrees Celsius at the time indicated, measured behind center of cell near center of PV module.
4. **MT5 cabinet temperature (degree C):** Air temperature within cabinet containing the MT5 multi-tracer in degrees Celsius at the time indicated.
5. **Dry Bulb Temperature:** Dry bulb temperature at the site in degrees Celsius at the time indicated for Golden, nearest 5-second average to the time indicated for Cocoa and Eugene.
6. **Relative Humidity:** Relative humidity at the site in percent, nearest 5- second average to the time indicated.

7. **Atmospheric Pressure:** Atmospheric pressure at the site in millibars, nearest 5-second average to the time indicated.
8. **Precipitation:** Accumulated daily total precipitation in millimeters at the time indicated.
9. **zenith:** Accumulated daily total precipitation in millimeters at the time indicated.
10. **Precipitation:** Accumulated daily total precipitation in millimeters at the time indicated.
11. **azimuth:** an angular coordinate system for locating positions in the sky. Azimuth is measured clockwise from true north to the point on the horizon directly below the object.
12. **zenith angle:** an angular measurement from straight up (zenith) to a point in the sky. Zenith angle can be used along with azimuth to indicate the position of a star or other celestial body. Zenith angle is the complementary angle of the elevation (elevation = 90° - zenith). The cosine of the solar zenith angle is used to calculate the vertical component of direct sunlight shining on a horizontal surface.

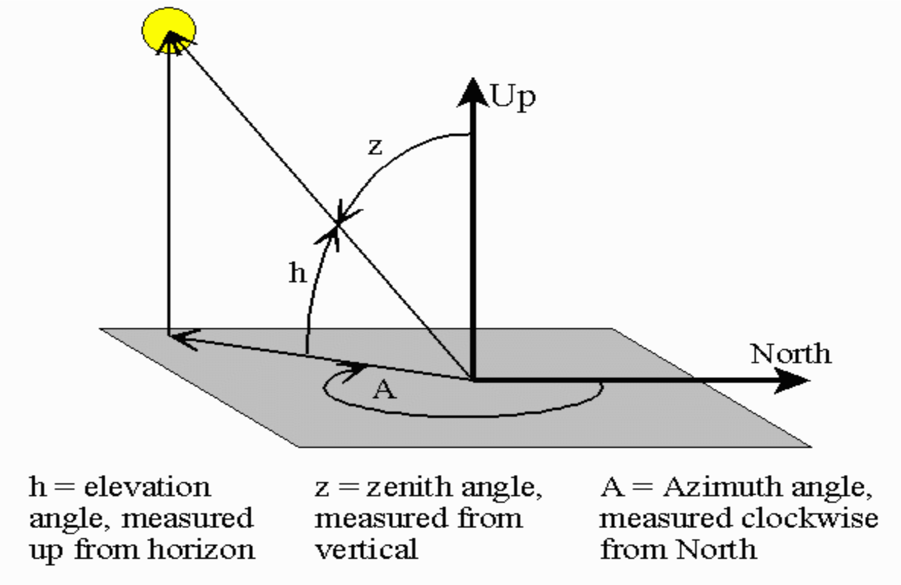

Up
z
h
North
A
h = elevation angle, measured up from horizon
z = zenith angle, measured from vertical
A = Azimuth angle, measured clockwise from North

**Figure 3.29:** Azimuth, elevation and Zenith angles [46].

13. **Daylight duration (micro seconds):** the amount of time the sun will shine during the day basically it is the difference between sunset and sunrise times in microseconds

14. **Day of the year:** The day of year is a numerical interpretation of the position of a day in a year often called DOY it starts with day 1 on January 1st till the day 365 or 366 in some cases in December 31th.

15. **Latitude:** an angular measurement of north-south location on Earth's surface. Latitude ranges from 90° south (at the south pole), through 0° (all along the equator), to 90° north (at the north pole). Latitude is usually defined as a positive value in the northern hemisphere and a negative value in the southern hemisphere.

16. **Longitude:** an angular measurement of east-west location on Earth's surface. Longitude is defined from the prime meridian, which passes through Greenwich, England. The international date line is defined around +/- 180° longitude. (180° east longitudes is the same as 180° west longitudes, because there are 360° in a circle.).

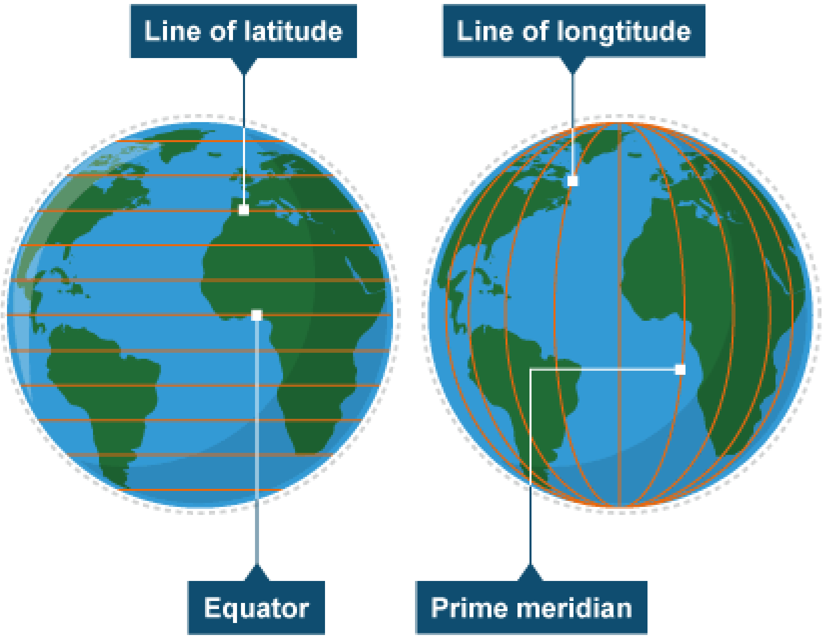


**Figure 3.30:** Lines of latitude and longitude [46].

### 3.5 Conclusion:

This chapter presented the data we used worked with, staring from its origin its format and how the data was actually measured, after that we discussed the different steps we took in order to extract, construct and select the optimal features that we would need. We started by cleaning the data and analyze it looking for outliers, visualization of the data revealed interesting behaviors and trend in the data we also plotted a convolution matrix to display the relationship between different features. In the next chapter, we are going to evaluate the performance for the selected machine learning models and discuss those results.

# Chapter 4:
# Evaluation and Results

After taking a theoretical overview about the two axes of this study; PV systems and Machine Learning Models, we presented our approach for the study and discussed the different steps we took in order to extract, construct and select the optimal features from the data that we worked with. Now we can proceed to the evaluation of the different machine models on the task of forecasting PV output power, and then compare them and discuss the results.

As discussed earlier, an indirect approach has been used in this study to forecast the PV output power, by first estimating the irradiance with meteorological and geographical data as input to the different machine learning models, then, based on the linear dependence between the PV output power and the irradiance, we were able to use a simple linear regressor model to forecast the PV power.

The forecasting process was carried out using python programming language with various library functions such as scikit-learn, keras, pandas and numpy, whereas the visualization process was conducted using matplotlib and seaborn libraries.

## 4.1 Performance evaluation of models:

To best visualize the performance of the different models, we picked one week from the training set (from 29 May to 4 June 2013) that incorporated all different weather conditions (clear and unclear days), to see how the models will react to abrupt changes and how the performance changes between regular sunny days and rainy or cloudy days.

We start by the simplest model which is the linear regressor;

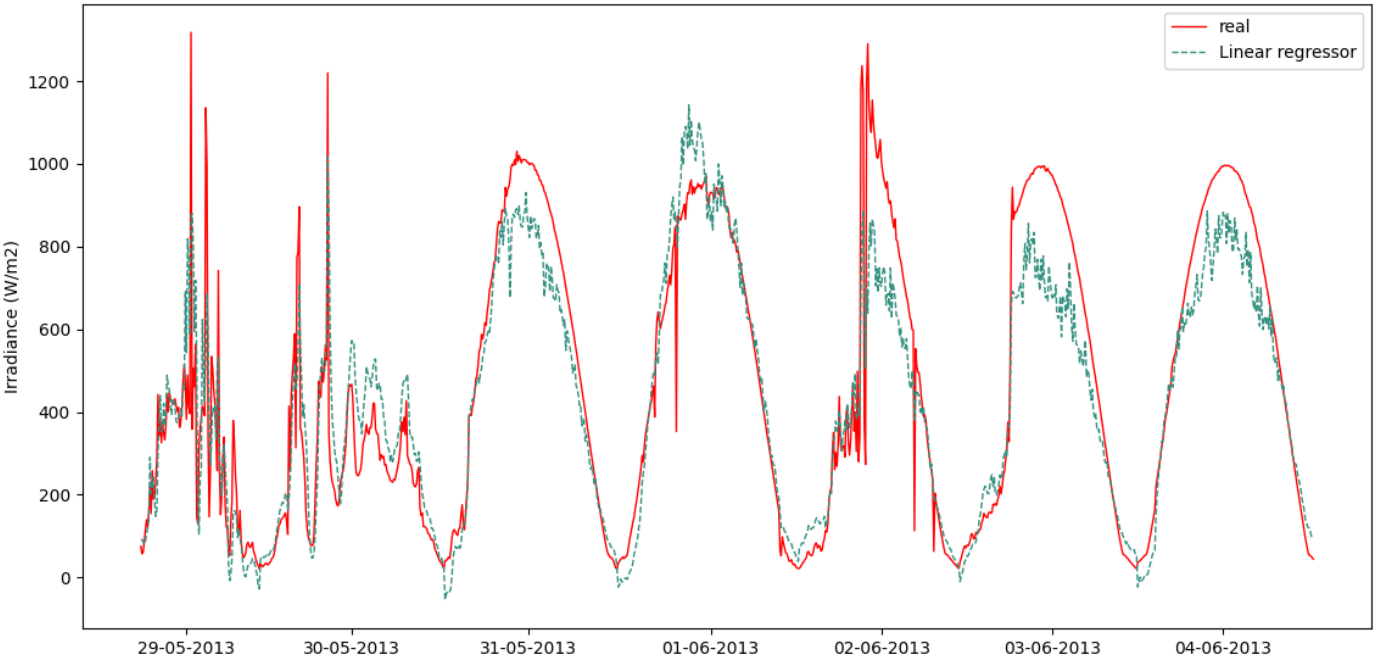


**Figure 4.1:** Forecasting performance of the Linear regressor.

Using the linear regressor model, we can clearly see that the forecasted output fails to follow the real output, and this is due to the nonlinear relationship between humidity, pressure, precipitation and other weather parameters on the one hand, and irradiance on the other hand (between the inputs and the output of the model). Hence, the linear nature of the model makes impossible to capture the actual irradiance.

However, we can remark that the model has some logic and were able to learn from other features that have a linear relationship with the irradiance, such as daylight duration and temperature.

One important advantage of the linear model is its simplicity and its fast response time, which allows it to deal with abrupt changes, thus making it ideal for online forecasting (real time forecasting) given linearly related input features and output target. These features make it the best model for forecasting PV output power given the irradiance as input.

Next, we will see the Polynomial Regressor, which is one of the derivatives of the linear regressor, in which the relationship between the input X and the output y is modeled as an nth degree polynomial.

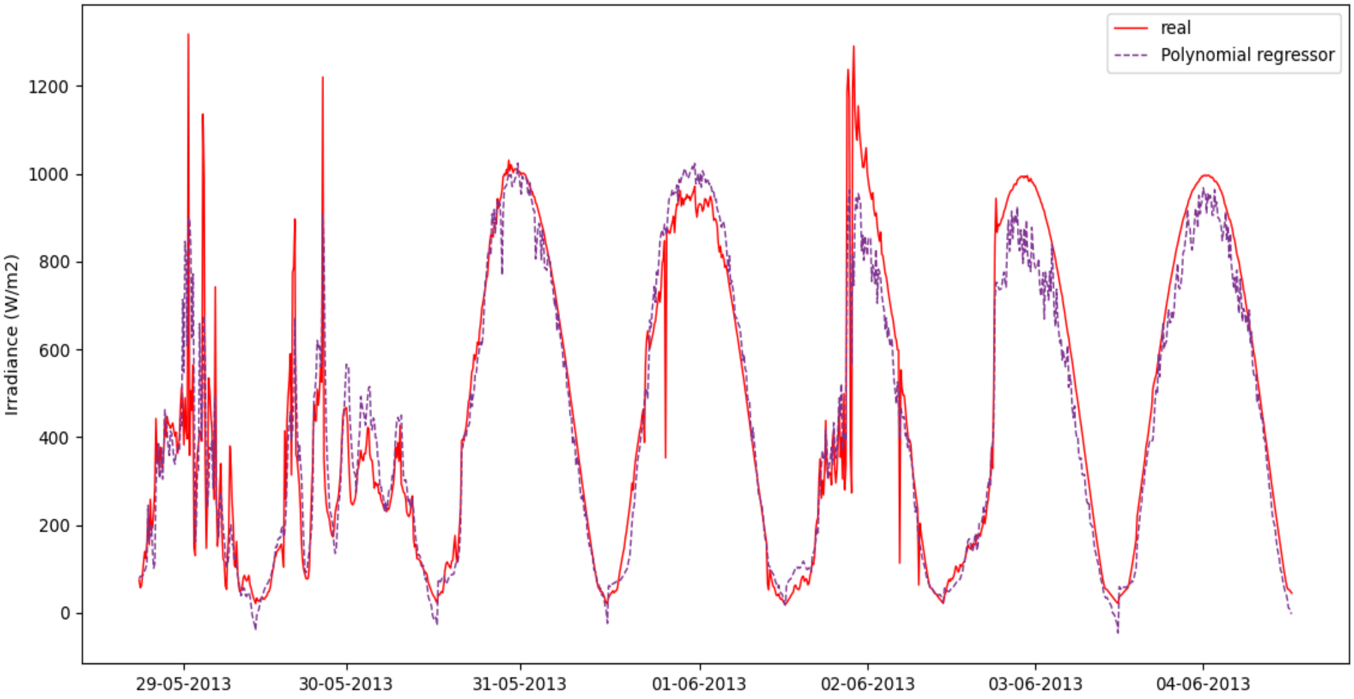


**Figure 4.2:** Forecasting performance of the polynomial regressor.

We can see that the polynomial regressor model is doing better than the linear model in following the real output irradiance. This is due to its ability to capture the nonlinearities between the input and the output, however, it failed to follow the irradiance peaks at noon and performed poorly in that period of the day. It also failed to give smooth forecasted output, which means that the model did not learn adequately the “profile” of the output.

A similar approach that depends on representing the model with a mathematical function is Support Vector Regression (SVR). SVRs are great with medium and small size datasets with a large number of features, which is not the case for our dataset, that is why the model took hours to train, yet it performed almost the same as the simple Linear Regression model and worse than the Polynomial Regression model.

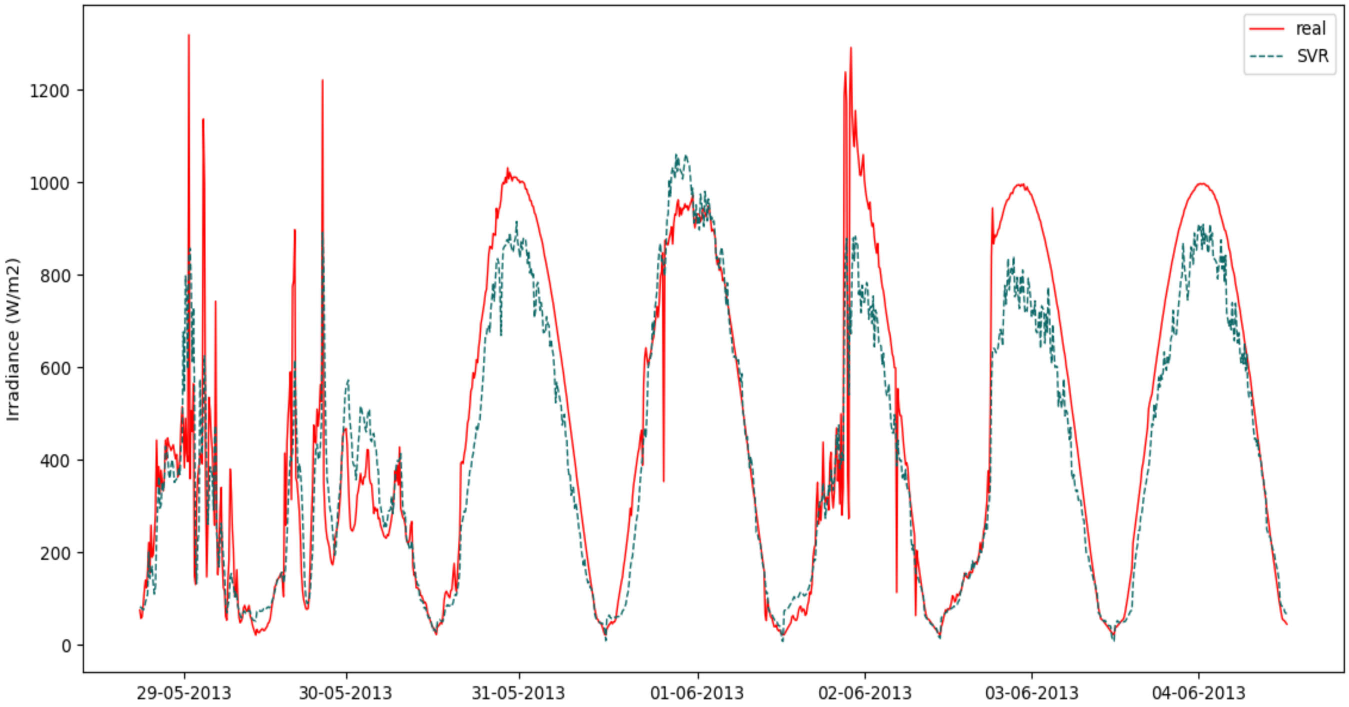


**Figure 4.3:** Forecasting performance of the SVR model.

In general, we got a similar performance for the Linear Regression and SVR models and a slightly better results with the Polynomial Repressor, but there is no room for improvements using the approach of representing the model with direct mathematical equations, at that time, we try a different approach; Decision trees.

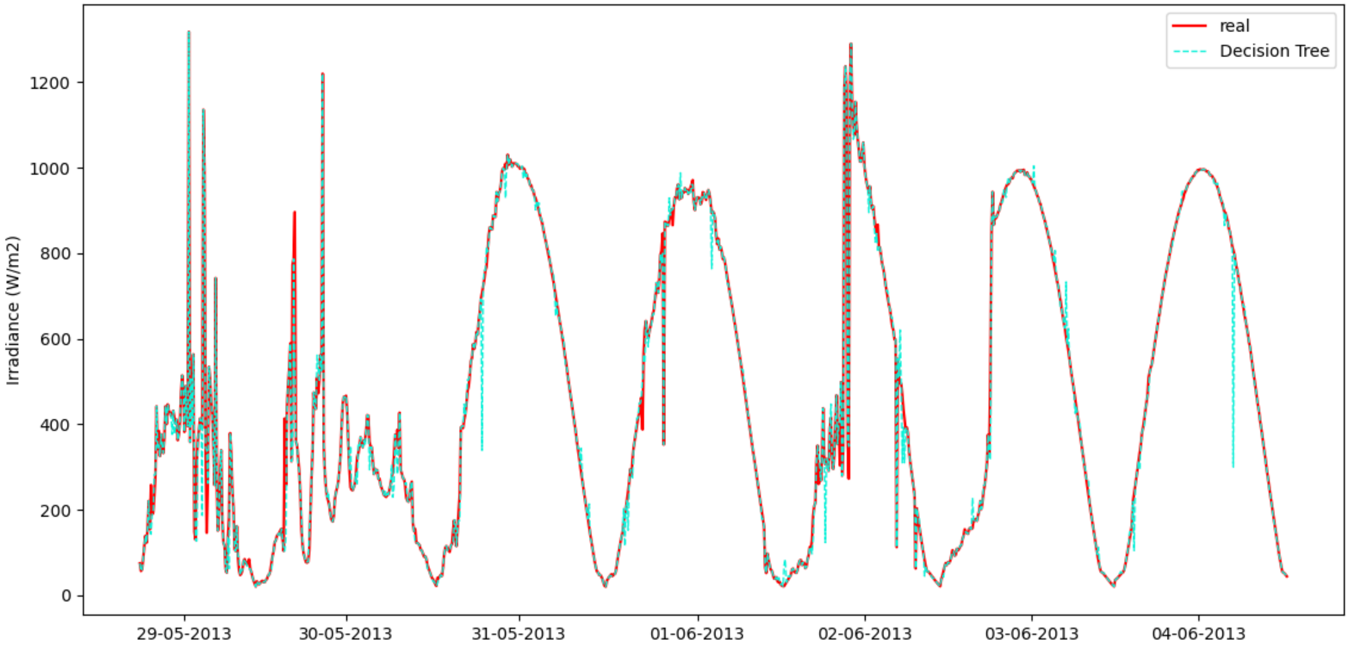


**Figure 4.4:** Forecasting performance of the Decision Tree model.

A decision tree is developed by breaking down a dataset into smaller and smaller subsets by asking a series of questions to the data, each question narrowing our possible values until the model gets confident enough to make a single prediction. This nature of the model allows it to deal perfectly with nonlinearities, and thus can “understand” the dataset better and gives finer forecasting results.

We can see that the forecasted output follows the real irradiance almost perfectly, and the model is sensitive to peaks and abrupt changes, but sometimes it can be too sensitive and generates some undesirable negative peaks, this is due to “wrong questions” asked by the model, this problem can be tackled by either feeding the model with more data, or by training other trees such that the decision is made through a vote of all the trees. This can lead to better decisions as it prevents the overall model from being “too sensitive” to certain features (overfit the dataset).

Since the available data is limited, we cannot feed the model any further, but the second approach can easily be done through a random forest.

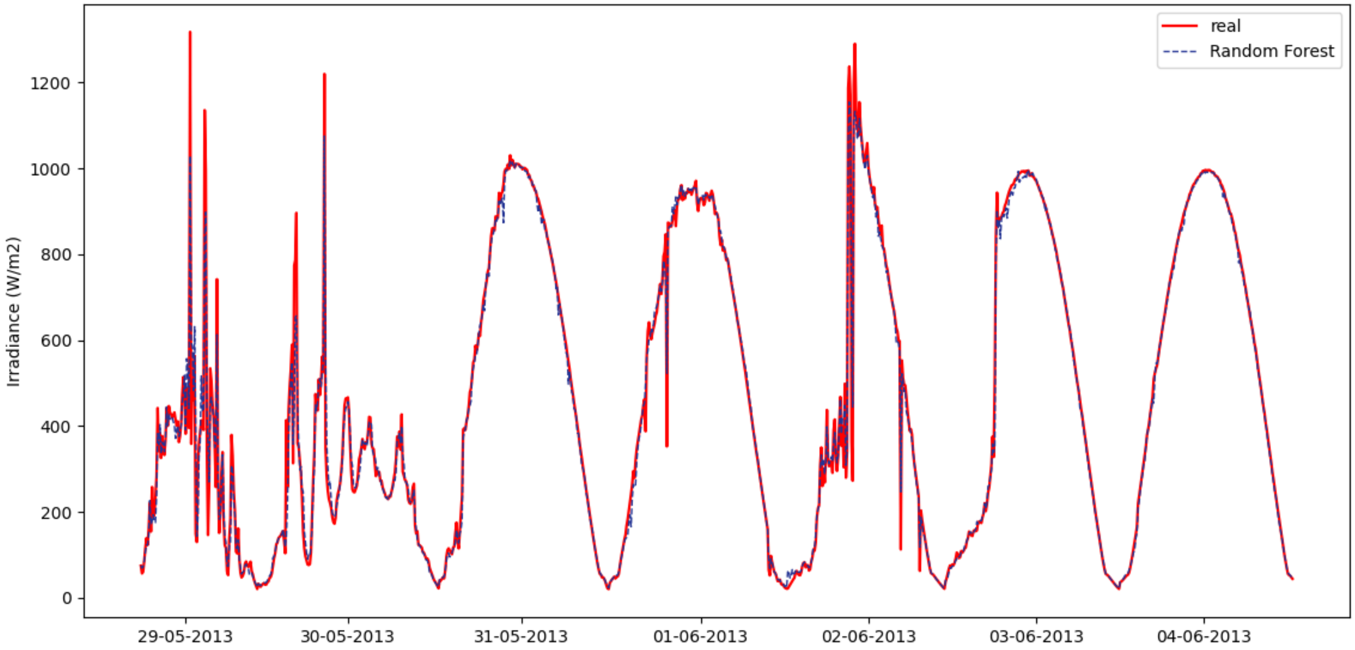


**Figure 4.5:** Forecasting performance of the Random Forest model.

As can be seen from the above figure, the forecasted output follows perfectly the real output, the Random Forest model was able to deal with the sensitivity problem of the Decision Tree, it is sensible to peaks and abrupt changes, but not too sensible. That is why, the model follows the peaks but to a certain level, even though it could not catch up with some peaks, but it generates well and gives great results.

The model was able to learn the profile of the irradiance and how it changes according to meteorological and geographical data.

But is this the best results we can get from ensemble learning techniques? while the Random Forest splits the dataset and train each decision tree on one subset of the data, the AdaBoost algorithm takes combinations of these subsets and train more decision trees, which leads to a slightly better results as we can see in the figure below;

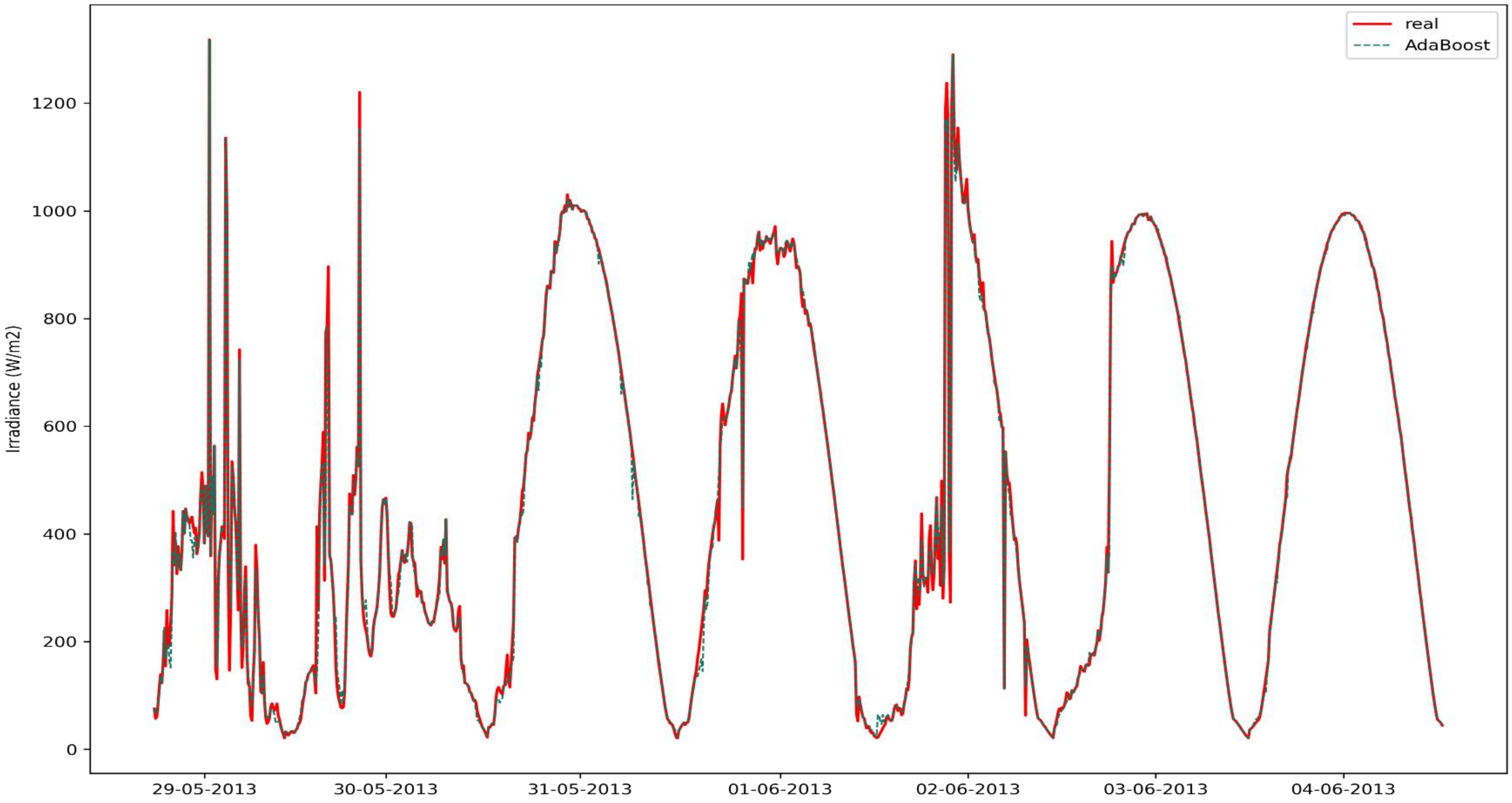


**Figure 4.6:** forecasting performance of the AdaBoost model.

The model dealt better with the peaks and at the same time generated well in clear days.

Another ensemble learning technique that has been used is Voting Regressor, which combines the predictions of several base estimators, each trained on the whole dataset, then it averages the individual predictions to form a final prediction, in order to improve the generalizability / robustness over a single estimator.

We used 5 base estimators to develop our Voting Regressor, namely: Linear regressor, SGD regressor, K Neighbors regressors, Decision Tree regressor and the Gradient Boosting regressor.

Even though we did not tune the model (because it requires a huge computational power and days or even weeks to train) an acceptable performance has been obtained, and we have a room for more improvements in later studies.

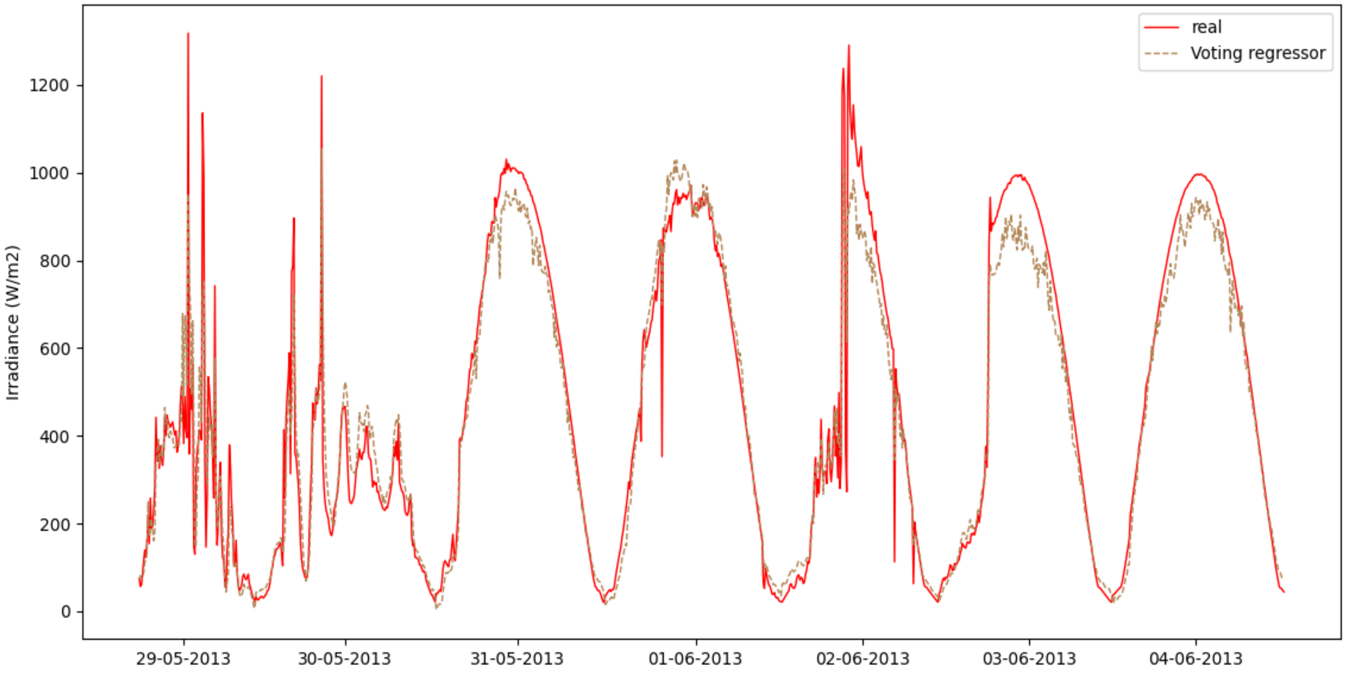


**Figure 4.7:** Forecasting performance of the Voting Regressor model.

As can be observed, the model performed well in following abrupt changes, but did poorly in the middle of the day and needs to be well tuned so it can deal with that.

Now we will take a totally different approach, which is Neural Networks.

Neural Networks are versatile, powerful, and scalable, making them ideal to tackle large and highly complex Machine Learning tasks like forecasting the irradiance using geographical and weather data.

Neural network model can fit any type of datasets, especially nonlinear ones, but needs huge amounts of data, and performs way better with more data than other models. For this task, and since the available data set is of medium size (94529 sample) the model performs well, but not better than Adaboost and Random Forest models.

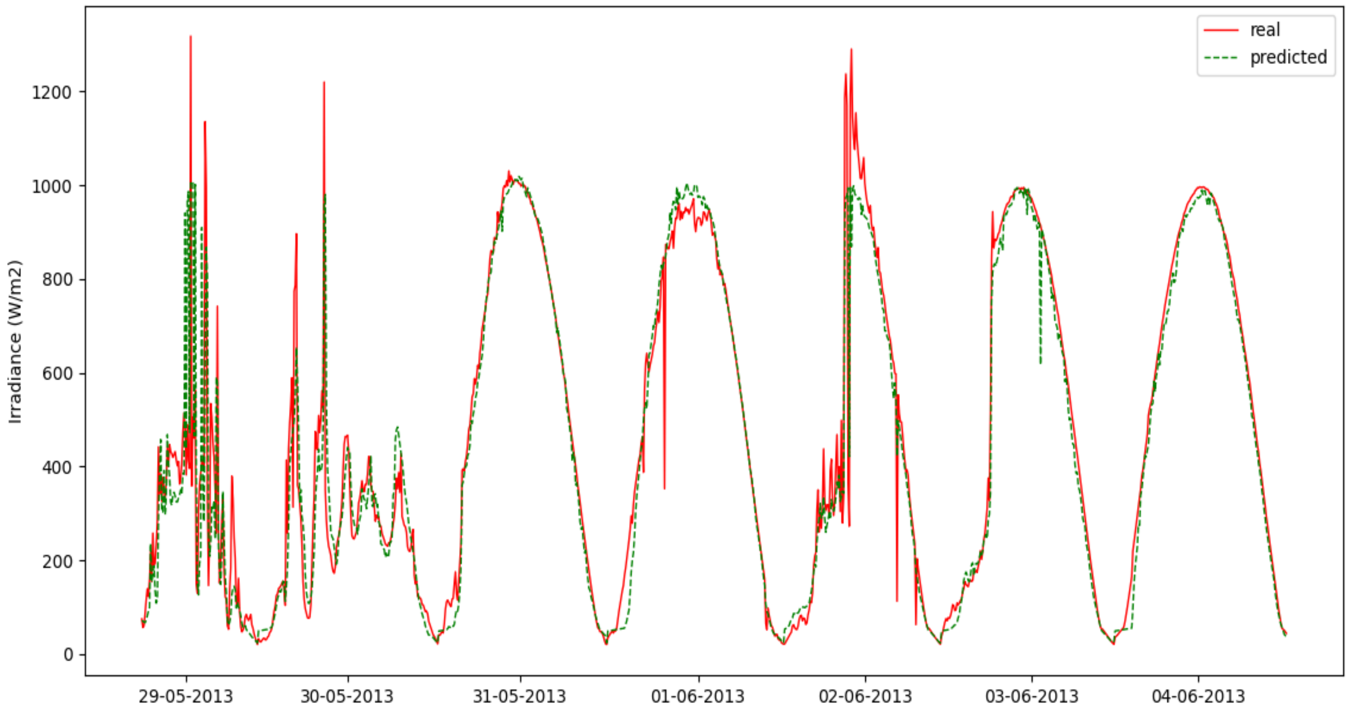


**Figure 4.8:** Forecasting performance of the Neural Network Model.

We can see that the model was able to establish a pattern in the dataset, but did not perform well in extremes. It is more stable and as a compromise, it does not follow the peaks.

A solution for this issue is to use Long Short-Term Memory networks (LSTMs), they are capable of learning long-term dependencies, which make them do great with extreme weather conditions. But compared to neural networks, they perform poorly in clear days.

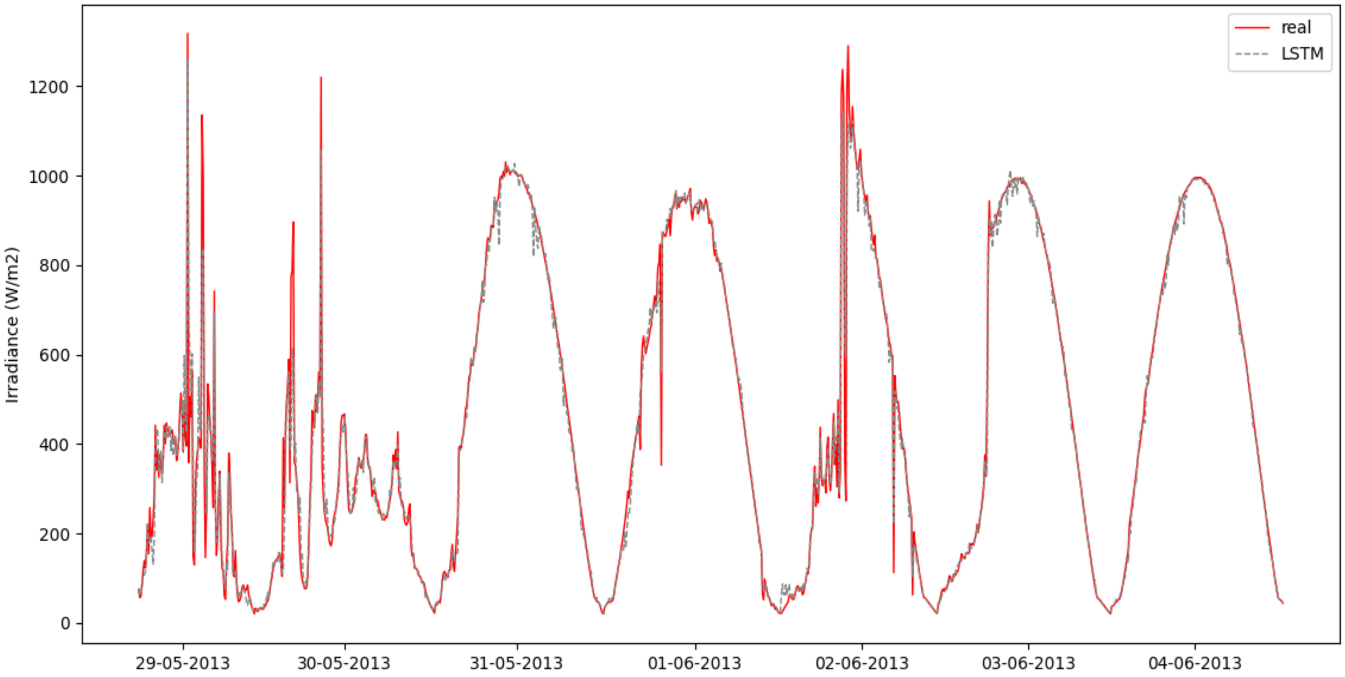


**Figure 4.9:** forecasting performance of LSTM model.

As can be seen, a better performance with the peaks and abrupt changes is obtained, however some noise can be noticed in clear days like day 3, 6 and 7.

Based on analyzing all these different models, we can observe some kind of a tradeoff between doing well on extreme weather conditions and generalizing well in clear days (a more stable system).

After forecasting the irradiance using different models, we can now proceed to our main task which is forecasting the PV output power, as we have seen in chapter 3 when analyzing the data; the irradiance and PV output Power are linearly related, so we can train a simple linear regressor model with the irradiance as input and the PV output power as the output. We got a strict line with the following equation;

$$PV\ output = 0.103718\ \times irradiance - 1.04324$$

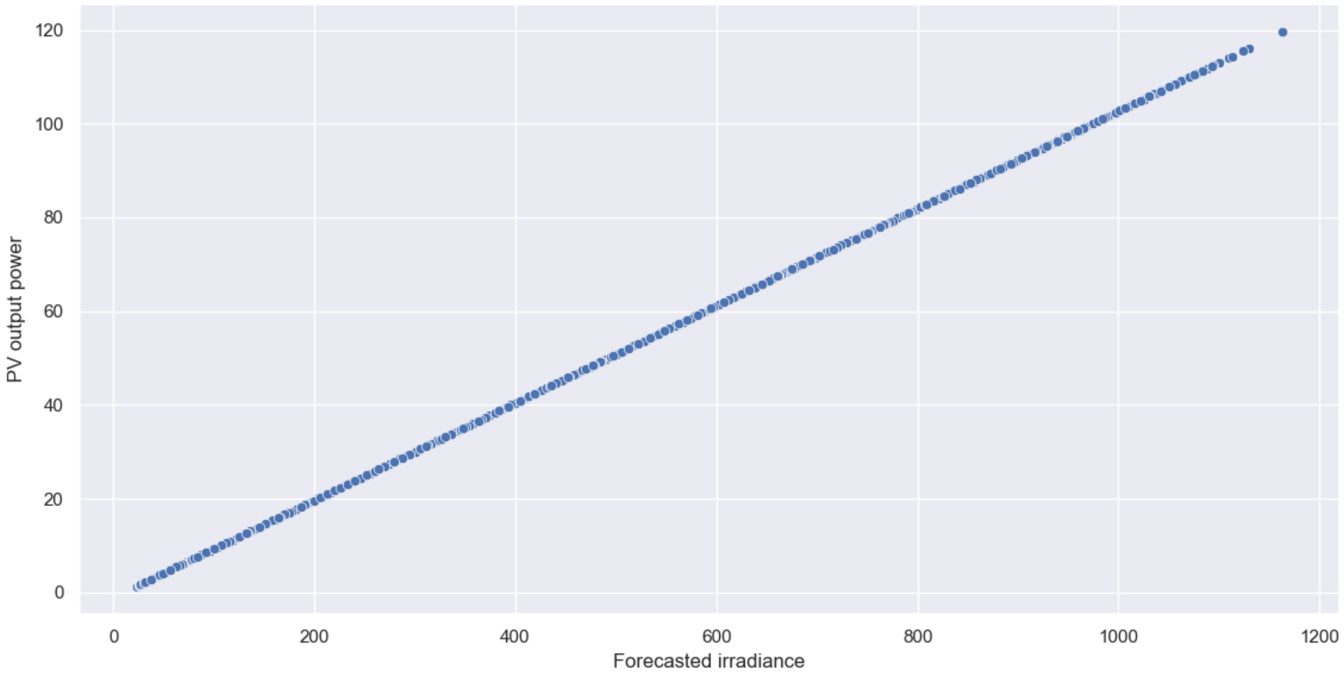


**Figure 4.10:** Forecasted irradiance vs PV output power.

Now, to compare the overall performance of these models and find the most adequate model for the task of PV forecasting, five different evaluation metrics have been used, each one of them focuses on certain characteristics of the model.

## 4.2 Comparison between models:

We were able to compare and classify all models according to their performance measured by the five evaluation metrics explained earlier in chapter 2, the two tables below show the obtained results. The first one is by feeding directly the data to the models and the second one is by using standardization technique, which is a scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. Using standardization, we got a better performance for the SVR, Neural Networks and Voting regressor models.

| | MSE | MAE | Med-AE | EVS | R2 |
|---|---|---|---|---|---|
| Linear regressor | 127.9962 | 0.4190 | 61.8088 | 0.8635 | 0.8635 |
| Polynomial regressor | 106.2382 | 0.3098 | 38.9909 | 0.9059 | 0.9059 |
| Decision Tree | 137.2180 | 0.2654 | 25.80 | 0.8430 | 0.8431 |
| Random Forest | 92.8959 | 0.1975 | 19.2063 | 0.9280 | 0.9281 |
| AdaBoost | 93.2548 | 0.1934 | 17.0 | 0.9276 | 0.9275 |
| Neural network | 109.8310 | 0.2553 | 28.4337 | 0.8995 | 0.8994 |
| LSTM | 97.3665 | 0.2052 | 19.1938 | 0.9211 | 0.9210 |
| SVR | 363.8237 | 0.5893 | 217.3964 | 0.0180 | -0.1029 |
| Voting regressor | 106.5449 | 0.2994 | 40.4559 | 0.9055 | 0.9054 |

**Table 4.1:** Comparison of performance metrics between various models using un-standardized data.

| | MSE | MAE | Med-AE | EVS | R2 |
|---|---|---|---|---|---|
| Linear regressor | 128.0308 | 0.4240 | 62.2913 | 0.8634 | 0.8634 |
| Polynomial regressor | 113.8816 | 0.4409 | 44.0561 | 0.8944 | 0.8919 |
| Decision Tree | 140.4257 | 0.2791 | 27.40 | 0.8357 | 0.8357 |
| Random Forest | 93.4549 | 0.2102 | 21.7881 | 0.9272 | 0.9272 |
| AdaBoost | 93.8345 | 0.2016 | 18.3667 | 0.9266 | 0.9266 |
| Neural network | 100.1439 | 0.1869 | 17.9082 | 0.9165 | 0.9164 |
| LSTM | 98.5443 | 0.2243 | 20.4575 | 0.9193 | 0.9190 |
| SVR | 141.4775 | 0.4006 | 65.7659 | 0.8339 | 0.8332 |
| Voting regressor | 105.0658 | 0.2922 | 39.4678 | 0.9080 | 0.9080 |

**Table 4.2:** Comparison of performance metrics between various models using standardized data.

To best compare between the models, the obtained results are represented on the following plots:

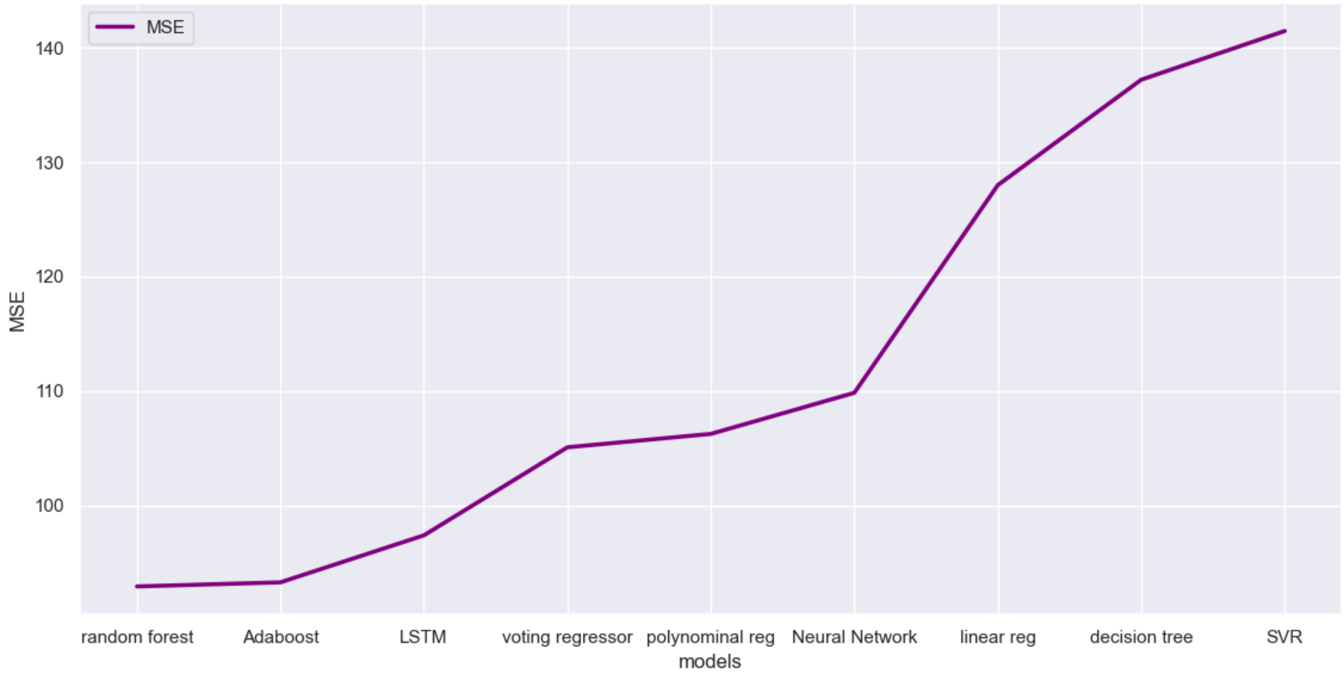


**Figure 4.11:** Mean Squared Error representation.

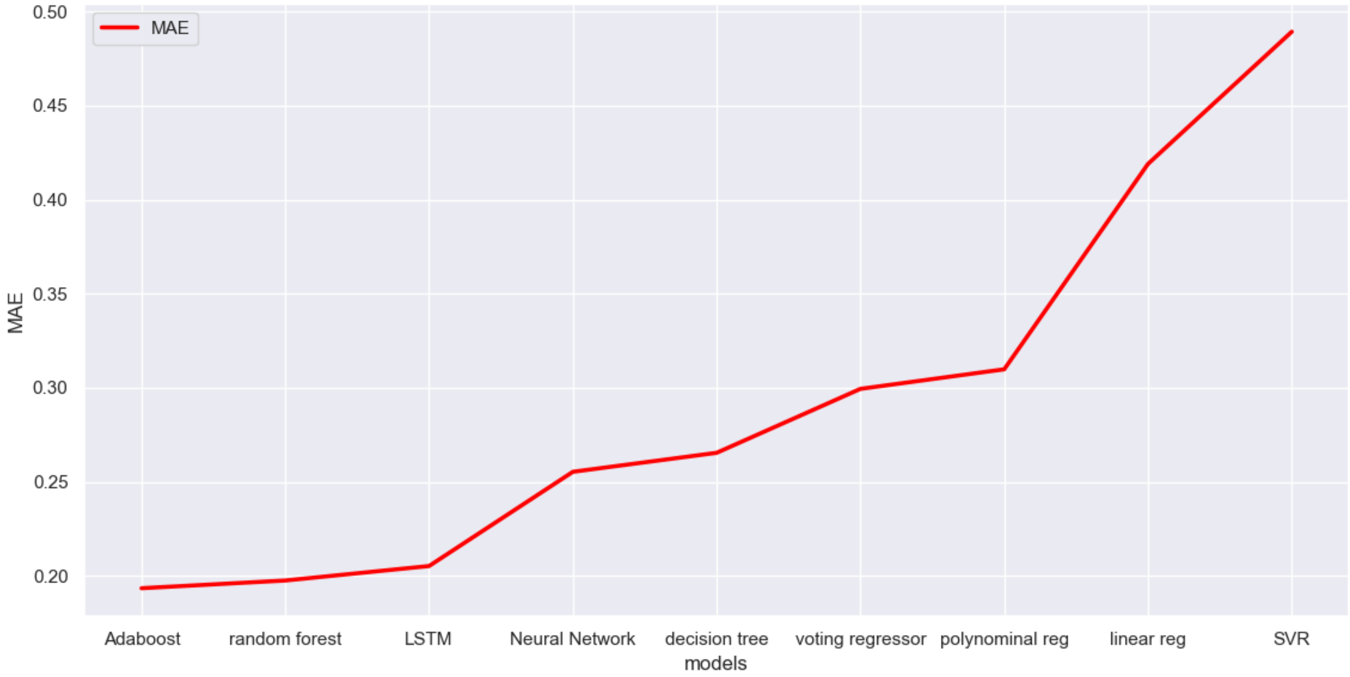


**Figure 4.12:** Mean Absolute Error representation.

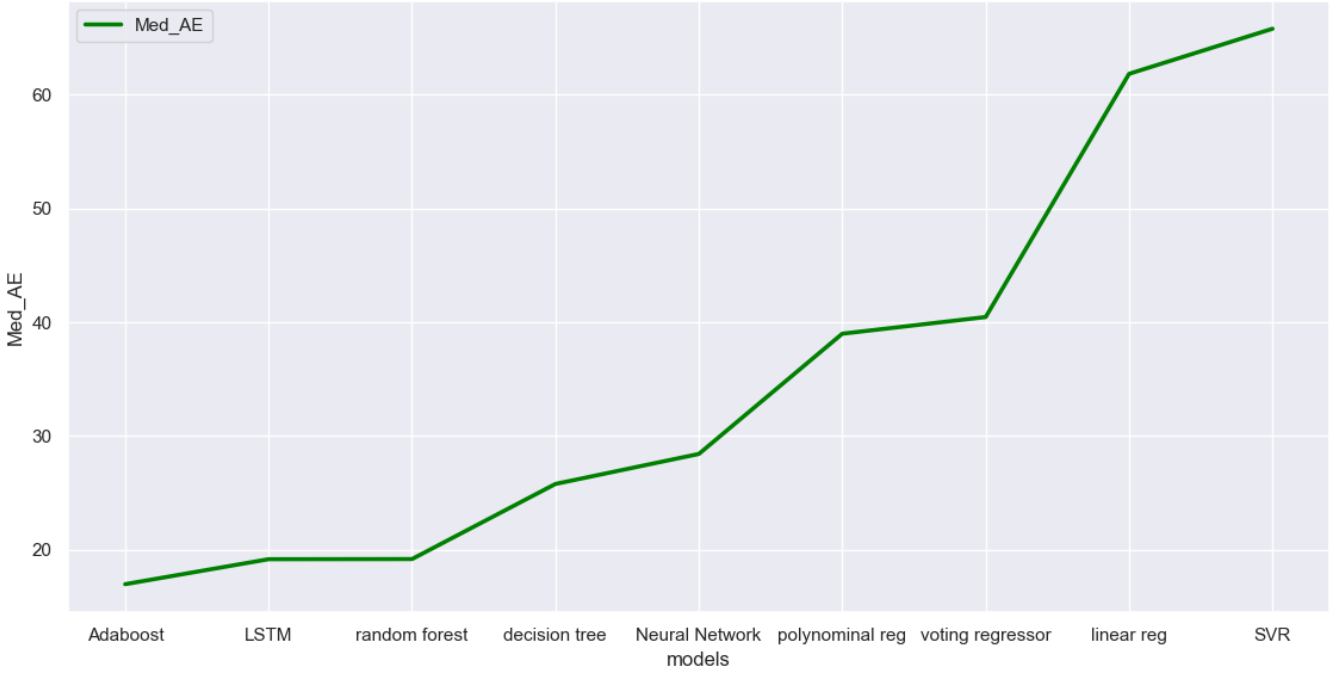


**Figure 4.13:** Median Absolute Error representation.

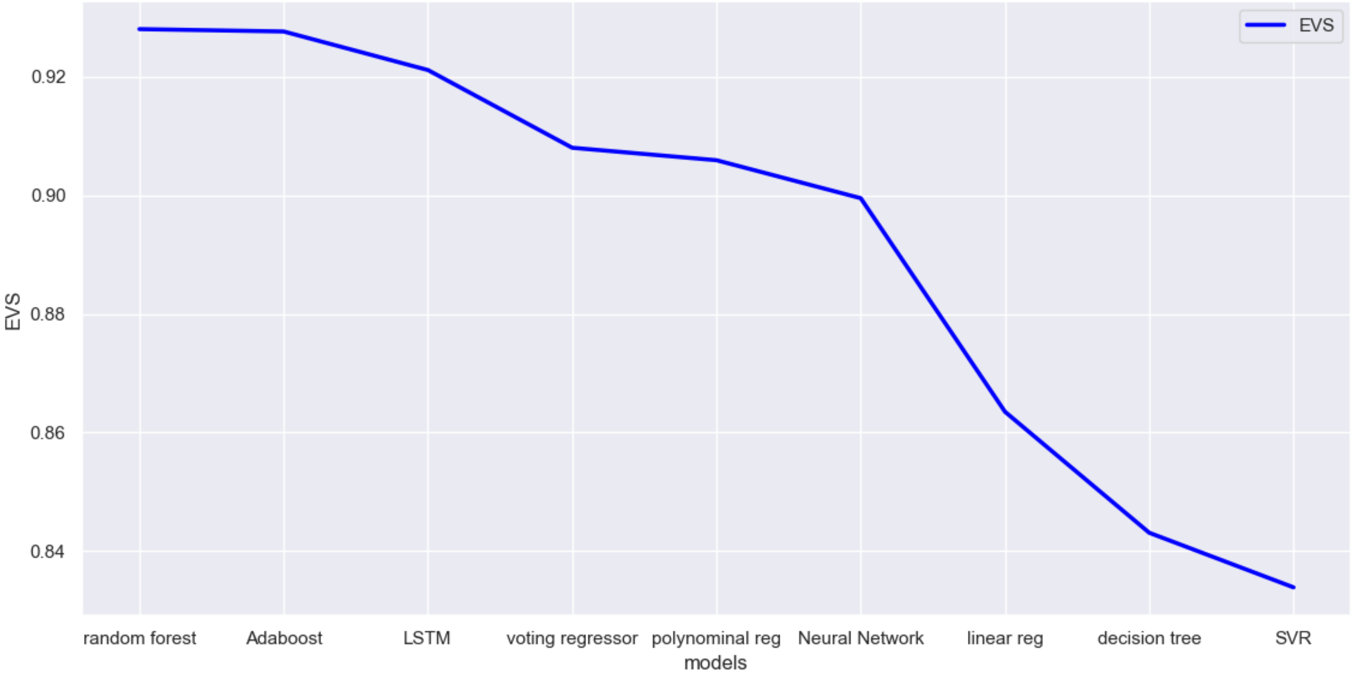


**Figure 4.14:** Explained Variance Score representation.

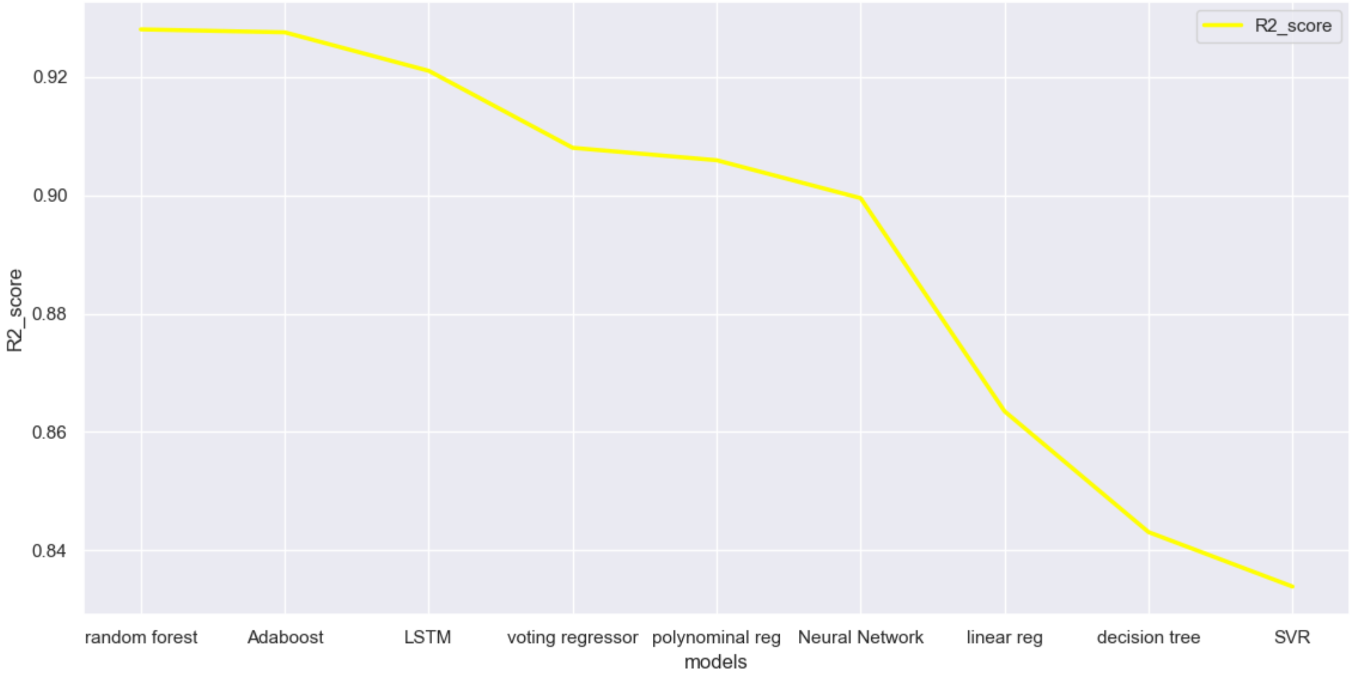


**Figure 4.15:** R2 score representation.

The reader should know that for MSE, MAE and Med_AE a lower value means a better performance, whereas for the EVS and R2 scores a higher value means a better performance. For this reason, the models were arranged from left to right (from the best performant to the least one).

We can summarize the previous graphs with the following one:

| MSE | MAE | Med AE | EVS | R2 Score |
|---|---|---|---|---|
| Random Forest | AdaBoost | AdaBoost | Random Forest | Random Forest |
| AdaBoost | Random Forest | LSTM | AdaBoost | AdaBoost |
| LSTM | LSTM | Random Forest | LSTM | LSTM |
| Voting regressor | Neural Network | Decision Tree | Voting regressor | Voting Regressor |
| Poly regressor | Decision Tree | Neural Network | Poly Regressor | Poly Regressor |
| Neural Network | Voting Regressor | Poly Regressor | Neural Network | Neural Network |
| Linear Regressor | Poly Regressor | Voting Regressor | Linear Regressor | Linear Regressor |
| Decision Tree | Linear Regressor | Linear Regressor | Decision Tree | Decision Tree |
| SVR | SVR | SVR | SVR | SVR |

We can see that Random Forest and AdaBoost models gave the best overall performance. Besides, LSTM has also demonstrated a good performance as it was close to RF and AdaBoost in multiple observations. Polynomial regressor, Voting Regressor and Neural network were very close to each other and performed reasonably well, however, with compromised reliability. On the other hand, Decision Tree, Linear Regressor and SVR performed Poorly under most of the measurements.

## 4.3 Discussion and Conclusion :

The purpose of the current study was to provide a machine learning-based model for the task of forecasting PV power generation. The originality of our system lies in the fact that we took the approach of forecasting the irradiance first based on meteorological and geographical data, then use a simple Linear regression model to forecast the generated PV power, so that the system can be autonomous and used with different PV installations and different PV cell technologies.

As has been demonstrated, the irradiance is 100% correlated with the PV generated power, which justifies the chosen approach. We have also built our dataset using data from three different locations and climates, so that the system can be robust and would not get stagnated/stuck to a specific location or a specific climate, and hence will perform well when used in other locations.

A comparative analysis using various machine learning approaches, including Linear Regression, Polynomial Regression, Decision Tree Regression, Random Forest regression, Neural Networks, Support Vector regression, Long Short-Term Memory AdaBoost and Voting regressor, is provided. The comparison between different models has been done using five performance metrics; mean squared error, mean absolute error, median absolute error, explained variance score and R2 score.

An analysis of the potential impacts of geographical and meteorological data on the irradiance reveals that the relative humidity, temperature, azimuth angle and daylight duration substantially impact the irradiance, whereas daily precipitation and atmospheric pressure appears to be a less significant dominating factor.

Due to its robustness to different geographical locations, the proposed system can be readily used in forecasting PV generation power on installations here in Algeria, with -normally- public available weather and geographical data.

The insights gained from this study might be of remarkable assistance in raising the rate of PV integration in conventional electric systems and help greening the grid. Further work needs to be done to understand the large- scale PV power generation forecasting and more efforts must be done on getting more data, and tuning complex models like the voting regressor and LSTM using more computational power.

# General Conclusion

This study is aimed at forecasting PV power generation using several machine learning techniques such as Random Forest, AdaBoost, Neural network and LSTM, these techniques were described, characterized and evaluated individually then compared together using 5 assessment metrics. The investigated approach consists in predicting the solar irradiance first by preprocessing the data and building accurate and robust models, then using the predicted values as an input to a simple Linear regression model to predict the generated power since the irradiance and generated power are highly correlated.

The results of the comparative study indicated that Ensemble learning models such as Random Forest and AdaBoost in addition to deep learning models like neural networks and LSTMs could achieve the best results and decrease error compared to other approaches. The ensemble methods outperformed the other methods due to their ability to merge several techniques which enhance the performance of the overall model even if the merged models are considered as weak learners.

As a further research work, we suggest to polish the linear regression model by adding with the irradiance more features that describe the state of the PV cells and panels such as fill factor and short circuit current, as well as soiling derate which describe the losses of the PV module due to dust, snow and shadow. Based on the performance evaluation of the different models, we can say that we still have a room for improvement, this can be done by acquiring more data and adding new features such as classification of the day (i.e., sunny, rainy, cloudy), wind direction, wind speed and other meteorological data. In addition, we can use more computational power to fine tune complex models like the voting regressor and LSTM. Also, further work must be done to understand the large-scale PV power generation forecasting.

# References

[1] Heangwoo Lee, 2019, Performance evaluation of a light shelf with a solar module based on the solar module attachment area, Building and Environment, Volume 159.

[2] Dr. Colleen Spiegel, ‘Components of a Photovoltaic System’, Accessed on June 23 2021

[3] IEA (2020), Solar PV, IEA, Paris, Accessed on June 20 2021

[4] ‘Data is compiled by Our World in Data based’ on two sources: BP Statistical Review of World Energy and Ember Climate, Accessed on June 20 2021

[5] Becquerel Institute and International Energy Agency (IEA) PVPS.

[6] Lafond et al. (2017) & IRENA Database, Accessed on June 12 2021.

[7] Our World in Data based on Global Carbon Project. (2020). Supplemental data of Global Carbon Budget 2020 (Version 1.0) [Data set]. Global Carbon Project.

[8] Liang Zhao, Wei Wang, Lingzhi Zhu, Yang Liu, Andreas Dubios, Economic analysis of solar energy development in North Africa, Global Energy Interconnection, Volume 1, Issue 1, 2018, Pages 53-62, ISSN 2096-5117,

[9] BP Statistical Review of World Energy; Shift Project; Maddison Project Database; UN Population Prospects

[10] ESMAP, Published on June 2020, Global photovoltaic power potential by country. Global Solar Atlas

[11] IEA/IRENA Renewables Policies Database, August 2016 Renewable Energy and Energy Efficiency Development Plan 2015-2030, Link: Renewable Energy and Energy Efficiency Development Plan 2015-2030 – Policies - IEA

[12] Hamdan, I., Maghraby, A. & Noureldeen, O. Stability improvement and control of grid-connected photovoltaic system during faults using supercapacitor SN Appl. Sci. 1, 1687 (2019).

[13] Harpreet Kaur and Inderpreet Kaur, Energy Return on Investment Analysis of a Solar Photovoltaic System chapter 3, published on 2019.

[14] Solar Facts and Advice: Monocrystalline Silicon, 2013. Link: Monocrystalline Solar Panels: Advantages and Disadvantages (solar-facts-and-advice.com), Accessed on June 10 2021

[15] EL-Shimy, Mohamed & AbdElAziz, A.A. & Oliba, K.A. & AbdEl-hameed, Kh.A. & Selim, O.A. & Helal, R.M. & Asaad, R.A. & Nasser, Rola & Essam, Reham & Email, R. & Tolba, S.H. & Zain, S.S. (2018). Experimental Analysis of Conditions Based Variations of Characteristics and Parameters of Photovoltaic Modules. 10.6084/m9.figshare.6199253.

[16] Characteristics of a Solar Cell and Parameters of a Solar Cell October 27, 2020, link: Characteristics of a Solar Cell and Parameters of a Solar Cell | Electrical4U, Accessed on June 12 2021

[17] Majid Jamil, M. Rizwan, D. P. Kothari, published on 2018, title: ‘Grid Integration of Solar Photovoltaic Systems’, Chapter 11 pages [184:203]

[18] K.N. Nwaigwe, P. Mutabilwa, E. Dintwa, An overview of solar power (PV systems) integration into electricity grids, Materials Science for Energy Technologies, Volume 2, Issue 3, 2019, pp. 629-633.

[19] Sun Shot U.S Department of Energy, July 2016, Systems Integration ‘Solar Forecasting: Maximizing its value for grid integration’.

[20] Thomas P. Trappenberg, “Fundamentals of Machine Learning”, page 98, Oxford university press 2020.

[21] Shai Shalev-Shwartz and Shai Ben-David, “understanding machine learning: from theory to algorithms”, page 97, Cambridge University Press 2014.

[22] John D. Kelleher, Brain Mac Namee, Aoife D’Arcy, “Machine Learning and predictive data analytics”, Second edition, page 361, The MIT Press 2020.

[23] Alex J. Smola, Bernhard scholkopf, “A tutorial on support vector regression”, 2004, Kluwer Academic Publishers

[24] Peter Flach, Machine Learning: the art and science of algorithms that make sense of data, second edition, page 132, Cambridge university press 2012

[25] Thomas P. Trappenberg, Fundamentals of Machine Learning, page 52, Oxford university press 2020.

[26] Georgios Drakos, 2019, “Decision Tree Regressor explained in depth”, gdcoder, accessed on: 07/06/2021

[27] Ankit Jain, Armando Fandango, Amita Kapoor, “TansorFlow Machine Learning Projects”, November 2018, Packt Publishing

[28] Junho Lee, Wu Wang, Fouzi Harrou, Ying Sun,“Reliable solar irradiance prediction using ensemble learning-based models: A comparative study”, Energy Conversion and Management, Volume 208, 2020, 112582

[29] D. P. Solomatine and D. L. Shrestha, "AdaBoost.RT: a boosting algorithm for regression problems," 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541), 2004, pp. 1163-1168 vol.2, doi: 10.1109/IJCNN.2004.1380102.

[30] John D. Kelleher, Brain Mac Namee, Aoife D’Arcy, “Machine Learning and predictive data analytics”, Second edition, page 361, The MIT Press 2020.

[31] Alexander Amini, MIT 6.S191 Course “Introduction to Deep Learning”, January 2020

[32] Alexei Botchkarev, “Performance Metrics (Error Measures) in machine learning regression, forecasting and prognostics: properties and typology”, 2018.

[33] Alessio Gozzoli, FloydHub Blog, “Practical guide to hyperparameters optimization for deep learning models”, 5 September 2018,

[34] James Bergstra, Yoshua Bengio, “Random Search for Hyper-Parameter Optimization”, Journal of Machine Learning Research 13 (2012) 281-305

[35] Jason Brownlee on August 15, 2020 “Discover Feature Engineering, How to Engineer Features and How to Get Good at It”, Accessed on June 12 2021.

[36] Effective Data Preprocessing and Feature Engineering | by Ali Hamza | Becoming Human: Artificial Intelligence Magazine, Accessed on June 10 2021.

[37] Feature Engineering for Machine Learning O’REILY by Alice Zheng, Amanda Casari (April 2018), chapter 1 pages [17 – 20]

[38] Shanawaz sheriff, April 15, 2020, “The What, Why and How of Feature Engineering”.

[39] Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task (forbes.com)

[40] Vaisala Weather Transmitter WXT520 Users Guide

[41] Jason Brownlee, June 30, 2020 ‘Data Preparation for Machine Learning: Data Cleaning, Feature Selection and Data Transform in Python’ pages [1 – 17]

[42] Jason Brownlee, June 30, 2020 ‘Data Preparation for Machine Learning: Data Cleaning, Feature Selection and Data Transform in Python’ pages [54 – 65].

[43] Mayank Tripathi, 16 June 2020 ‘Knowing all about Outliers in Machine Learning’.

[44] Tarun Acharya, march 30th 2020, “Understanding Feature extraction using Correlation Matrix and Scatter Plots”

[45] Arpit Bajpai and Markus Duchon on 25-28 June 2019, ‘A Hybrid Approach of Solar Power Forecasting Using Machine Learning’

[46] ESRL Global Monitoring Laboratory - Global Radiation and Aerosols (noaa.gov).