REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ MOHAMED BOUDIAF - M'SILA

FACULTÉ : TECHNOLOGIE
DEPARTEMENT : ELECTRONIQUE
N° :

DOMAINE: Sciences et Technologies
FILIERE : Électronique
OPTION : Electronique des Systèmes Embarqués

Mémoire présenté pour l'obtention du diplôme de Master Académique

Par : Bouguerra Oussama

Benslimane Oussama

Intitulé

SOLAR RADIATION PREDICTION USING MACHINE LEARNING

Le jury composé de :

Mr. ATTALLAH Bilal	Université de M'sila	Président
Mr. DJERIOUI Mohamed	Université de M'sila	Encadreur
Mr. BRIK Youcef	Université de M'sila	Co-Encadreur
Mr. BOURAS Mounir	Université de M'sila	Examinateur

Année universitaire : 2019 /2020

THANKS

First of all, we thank our coach **Dr. DJERIOUI Mohamed** who agreed to mentor us to make this thesis. We are grateful to him. Throughout this work, we have benefited from his great competence, his intellectual rigor, his efficiency, and his invaluable advice.

We would also like to thank **Dr. BRIK Youcef** for reserving some of his time to follow our work. We would also like to thank the jury members who honoured us with the honour of judging this thesis and proclaiming its scientific value. We would also like to thank all those who contributed directly or indirectly to the realization of this thesis.

Dedication

I dedicate this modest work to my dear father for his sacrifices as a testimony of all my affection throughout my studies. To my dear mother who has not stopped encouraging me since my first school year. To my brother Omar Elfarouk ,Bilal. To my sisters Sanna And all the family ,To all my friends especially: Ahmed, Mohammed,Salah,Houcine ,Abd Halim,Adel.

To all the teachers in the Department of Electronics.

Bouguerra Oussama

Dedication

I dedicate this modest work to my dear father for his sacrifices as a testimony of all my affection throughout my studies. To my dear mother who has not stopped encouraging me since my first school year.

Benslimane Oussama

Table of Contents

CHAPTER I State of the Art

CHAPTER II Machine learning and deep learning techniques

CHAPTER III Results and Discussion

LISTE OF FIGURES

CHAPTER I State of the Art

CHAPTER II Machine learning and deep learning techniques

CHAPTER III Results and Discussion

LISTE OF TABLES

CHAPTER II Machine learning and deep learning techniques

CHAPTER III Results and Discussion

NOTATION AND ABBREVIATED TERMS

RE : Renewable Energy

DNI : Direct Normal Irradiance

DHI : Diffuse Horizontal Irradiance

GHI : Global Horizontal Irradiance

G_d **:** Diffuse Horizontal Irradiance

G_b **:** beam irradiance

θ_z **:** zenith angle

Φ : latitude of the place

δ **:** declination

G_{sc} : solar constant

r: distance

ω **:** Time Angle

TSV : dedicated solar time

A_z **:** Azimuth

I_{cc} **:** short circuit current

V_{co} **:** circuit voltage

DC : direct current

AC : alternating current

ML : Machine Learning

DL : Deep learning

AI : artificial intelligence

CART : Classification And Regression Trees

B_{opti} **:** the number of samples bootstrap

$mtry_{opti}$ **:** the number of variables selected at each separation

RF : Random forest

GBM : Gradient Boosting Machine

$\boldsymbol{F_{approx}(x)}$ **:** find an approximation

ANNs : Artificial Neural Networks

NNs : Neural Networks

MLP : Multilayer Perceptron

FNN : Feedforward Neural Networks

SGD : Stochastic Gradient Descent

DNN : Deep neural network

TNNs : Traditional neural networks

RNNs : Recurrent neural networks

LSTM : Long Short-Term Memory

MTSA : Multiple Travelling Salesman Assignment

NRNS : Nested RNS

$\boldsymbol{X_t}$ **:** current input

BILSTM : Bidirectional Long Short-Term Memory

BRNNs : Bidirectional Recurrent neural networks

$\boldsymbol{x_{i_n}}$ **:** The normalized value.

MAE : Mean Absolute Error

MSE : Mean Square Error

RMSE : Root Mean Square Error

RRMSE : Relative Root Mean Square Error

R : Correlation Coefficient

R ² : Coefficient Of Determination

GPU: graphics processing unit

API : *Application Programming Interface*

CPU : Central Processing Unit

RAM : *Random Access Memory*

INTRODUCTION

INTRODUCTION

Renewable Energy is becoming a technology and an ever more viable alternative to traditional non-renewable energy sources. Therefore, the relentless danger of climate change forces humanity to strive for new and more effective ways to produce energy. Particularly when we base all our basic needs on this form of energy.

According to the International Energy Agency (IEA) [1], global renewable electricity capacity is projected to rise by over 1 TW, a 46 percent increase over the period 2018 to 2023.

Solar photovoltaic (PV) represents more than half of this expansion and dominates the growth of renewable ability.

However, because the energy output of PV panels depends on weather conditions such as cloud cover and solar irradiance, the PV panels' energy output is unstable. To understand and manage the output variability is of interest for several actors in the energy market.

Nowadays, the use of Artificial Intelligence Technologies is becoming increasingly widespread due to their ability to solve highly complex problems. The increase in the efficiency of computers and algorithms has helped to solve problems, not only in engineering but also in many fields such as medicine[2], finance[3], and the environment[4]. That is why these instruments are becoming more popular and can still make some progress and can be implemented in more areas or issues.

Machine learning is a computer science subfield, and it is categorized as a form of artificial intelligence. It can be used in many domains and the advantage of this approach is that a model can solve problems that clear algorithms can not represent.

In the case of Renewable Energy Source (RES), these machine learning models are used to forecast the energy produced in a power station or even to forecast the actions of weather conditions. In the energy industry, such predictions are of great significance. The problem of Solar Irradiance variability and unpredictability which reaches the surface of the Earth is well known. A precise forecast of this variable will therefore facilitate better planning and operation of power delivery at the economic level or at the level of energy

output, either by making alternative arrangements for traditional power and overall timetables or by investing the correct amount of energy resources and reserves to minimize the operating costs of the power system.

Additionally, related literature on the prediction of solar surface irradiation, estimation of PV capacity and current state-of-the-art studies is reviewed [5]. Machine learning algorithms such as support vector machine (SVM) [6,7], k-nearest neighbor [8,9], and Random Forest [10] have been used in studies; others compared or combined the prediction results with multiple machine learning models.

Deep learning is a particular subfield of machine learning designed to allow machines to simulate the way the human brain thinks, and its operational model is based on neuroscience [11]. Deep learning is designed for the representation of input and target data using a neural network structure. Actually, several types of research have addressed in solar irradiance prediction based on deep learning models. In [12], the authors have proposed a deep learning hybrid model to predict solar radiation in 2 stages, a convolutional network is used to extract features, and a Long-Short-Term Memory (LSTM) network is used for the prediction stage. [13] proposed a recurrent neural network model to investigate how emerging deep learning algorithms, especially compared to swallowing artificial neural networks, contribute to accurate solar radiation prediction. Guijo-Rubio et al. [14] have presented how evolutionary artificial neural network algorithms can obtain extremely accurate solar radiation prediction, exclusively working with satellite-based data. To predict solar irradiation from weather measurements, an advanced machine learning-based model was used in [15]. This model is based on LSTM networks which, in time series data, have an extended capacity to explain time dependencies. A deep learning approach based on traditional neural networks has been used to predict solar irradiance at thirty stations in Turkey [16].

The purpose of this memory is to research the viability of machine learning algorithms to forecast day-ahead solar irradiance of the next hour and hourly. The studied machine learning algorithms are the networks Random Forest, Gradient Boosting Machine, Deep Neural Network ,LSTM, and Bidirectional LSTM. The research uses historical weather data from the HI-SEAS weather station (Dataset).

The rest of the memory is organized as follows:

Chapter I introduces a few of the features and behaviors of this RES. In this chapter, we will see the value of this form of technology increasing and understand why we should invest in it.

Chapter II offers an overview of how deep neural networked machine learning works. It begins with a general machine learning summary and then continues with a more thorough take on the core components of machine learning. The structure of a neural network and the algorithm for gradient descent optimization are discussed in some detail.

Chapter III focuses on the empirical results, analysis, and discussion of the results are presented.

Finally, the last section describes the key conclusions taken from the analysis along with implications for further field studies.

CHAPTER I

State of the Art

I. INTRODUCTION

Renewable energies are inexhaustibly provided by the Sun, the wind, the heat of the Earth, waterfalls, tides, or the growth of plants. These are the energies of the future. Today, they are underexploited about their potential. For example, renewable energy accounts for only 20% of global electricity consumption. Control of the random nature of renewable energy sources such as solar radiation on the ground could allow the proper sizing of solar systems of all kinds and allow power grid operators to integrate them better. Unfortunately, radiation data are not widely available. With this in mind, we have endeavored during this study to contribute to the research of modeling methodologies for the prediction of solar radiation. This type of prediction is essential because it could, in the long term, make it possible to exploit better the solar Renewable Energy (RE), whose intermittence heavily penalizes its use. To make this prediction, it is necessary to have statistical and mathematical tools dedicated to this type of analysis[17].

II.SOLAR RADIATION

II.1. Definition

Solar radiation is the energy per unit vicinity obtained from the Sun in the shape of electromagnetic radiation. The SI unit of photovoltaic irradiance is watt per rectangular meter W/m2. The find out about and size of photovoltaic irradiance is fascinating for the prediction of the power era of photovoltaic energy plants[18].

II.2. Type of Solar Radiation

II.2.1.Direct Normal Irradiance (DNI)

Additionally acknowledged as beam irradiance is the photovoltaic radiation measured at a floor of the earth perpendicular to the Sun. It solely measures the direct radiation from the solar disk and excludes the diffuse radiation[18].

II.2.2.Diffuse Horizontal Irradiance (DHI)

Is the radiation measured on a horizontal floor on Earth, coming from mild scattered by way of the atmosphere. It measures radiation from all points in the sky except for radiation

from the solar disk. In the absence of atmosphere, there has to be nearly no diffuse sky radiation[18].

II.2.3.Reflected Radiation

Is the radiation mirrored by using non-atmospherical factors such as the ground. However, photovoltaic panels tend to be tilted away from the mirrored radiation trajectory, so it hardly ever has relevance in the whole radiation acquired by using their surface. An exception is in stipulations the place the floor is surrounded with the aid of snow, which can make more significant extensively the mirrored radiation received [18].

II.2.4.Global Horizontal Irradiance (GHI)

Is the total irradiance from the solar on a horizontal floor on Earth. It is the sum of DHI, DNI (after accounting for the photovoltaic zenith perspective of the solar θ_z) and mirrored radiation. However, due to the fact reflected radiation is commonly insignificant in contrast to direct and diffuse radiation for all realistic functions, international horizontal radiation is stated to be the sum of direct and diffuse radiation solely :

$$\boldsymbol{G = G_d + G_b cos}(\theta_z) \quad \text{(I.1)}$$

the location G denotes the Global Horizontal Irradiance,$\boldsymbol{G_d}$ the Diffuse Horizontal Irradiance, $\boldsymbol{G_b}$ Direct Normal Irradiance or beam irradiance and θ_z the zenith angle. The referred to types of photograph voltaic radiation are established in **Figure I.1**

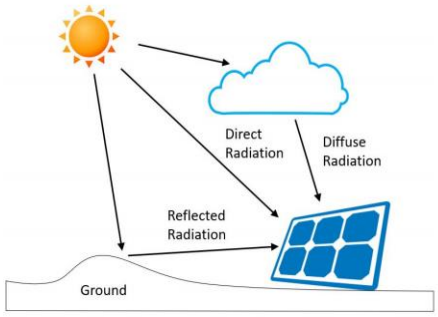

Fig.I. 1. Types of Solar Radiation.

The energy, E_{ph}, of each photon is directly related to the wavelength λ by the relationship

$$E_{ph} = \frac{hC}{\lambda} \quad \text{(I.2)}$$

Where h is the Planck constant, h = 6,62.10^{-34} J.s [18].

II.3. Why Predict Solar Radiation

The installation of any solar energy system in a given site requires preliminary studies. Indeed, sizing and simulation are essential to ensure optimal operation. To carry out such tasks, reliable measurements over relatively long periods of certain meteorological variables, and especially those of solar radiation, are essential. The lack of a long series of data or data series of low quality (discontinuity and unreliability) can combine errors in the design, sizing, and prediction of solar system performance, which hurts investment. Unfortunately, measurements of solar radiation are generally inaccurate and rare worldwide; Especially in Algeria, due to the high price of measuring devices. There are only a small number of solar radiation stations, which is why there is a lack of solar radiation measurements in large areas on the one hand. On the other hand, where these data exist, there are generally periods of failure due to failures or low monitoring, since the majority of these stations belong to establishments that do not benefit economically from these data.

However, other meteorological parameters such as temperature ambient, humidity, wind speed are relatively easily measured in a larger number of weather stations with a relatively low cost compared to that of radiation. On the other hand, the optimization of a solar system or the simulation of its performances require at least, daily data, even, schedule of solar irradiation. Therefore, it is desirable to develop relationships between available meteorological data and solar irradiation; and to develop precise techniques (models) to predict solar irradiation. These models are tools that allow the generation of long series of solar irradiation data at different time steps; this could provide useful information to decision-makers in terms of selection, design, and planning of new solar power plants [19].

II.4.Renewable Energy Issues

The world has developed, and energy needs are growing to support both economic development and the requirements in terms of comfort and consumption of populations. At the moment, we are coming to a critical moment in energy exploitation: we realize the fragility

and inconsistency of our functioning. Indeed, the planet's resources in fossil sediment are depleting, and oil is rarefied, and, in addition to the economic consequences, it is clear that we must either find alternatives to current energy sources or find an alternative to our mode of civilization itself. Without energy, our daily life disappears. Also, the exploitation of fossil fuels poses another problem: the impact on the environment is massive. If this has long been ignored, the preservation of the environment becomes a global issue, again with significant economic stakes. The environmental community has long been aware of this policy, and recently public opinion. The ecological and industrial disasters plunge people into a grip of awareness of the dangers generated by the impact of humanity on our planet. We shall see in this section the evolution of mentalities and policies over the past thirty years in the face of these difficulties and this ever-increasing demand for energy. We will then focus on the implications of these various directives at the international and national levels [19].

II.5.Extraterrestrial Radiation

Extraterrestrial solar radiation covers an extensive range of wavelengths. It does not depend on any meteorological parameters. Still, it depends on some parameters astronomical and geographic such as the latitude of the place (φ), the solar declination (δ), and the hour angle at sunset (ω0) [20].

On a horizontal surface, and for day n, extraterrestrial radiation G_0 (MJ.m^{-2}.Day^{-1}) is obtained from the following equation:

$$G_0 = \frac{24 \times 60}{\pi} G_{sc} \left[1 + 0.034 \cos\left(\frac{2\pi n}{365}\right)\right] [\omega_0 . \sin(\varphi) . \sin(\delta) + \cos(\varphi) . \cos(\delta) . \sin(\omega_0)] \quad \text{(I.3)}$$

G_{sc} : Is the solar constant equal to 0.0082 MJ.m^{-2}.mn^{-1}.

n: is the number of the day from the first of January.

II.6.Solar Radiation In The Earth's Atmosphere

The radiation received on the Earth's atmosphere occupies only a small portion of the spectrum of solar electromagnetic waves. Wavelengths between 0.2 and 2.5 µm characterize it; it includes the domain of the visible (light waves from 0.4 to 0.8 µm).

Solar energy collectors, which correspond to solar cells, will therefore have to be compatible with these wavelengths to be able to trap the photons and return them in the form of electrons [20].

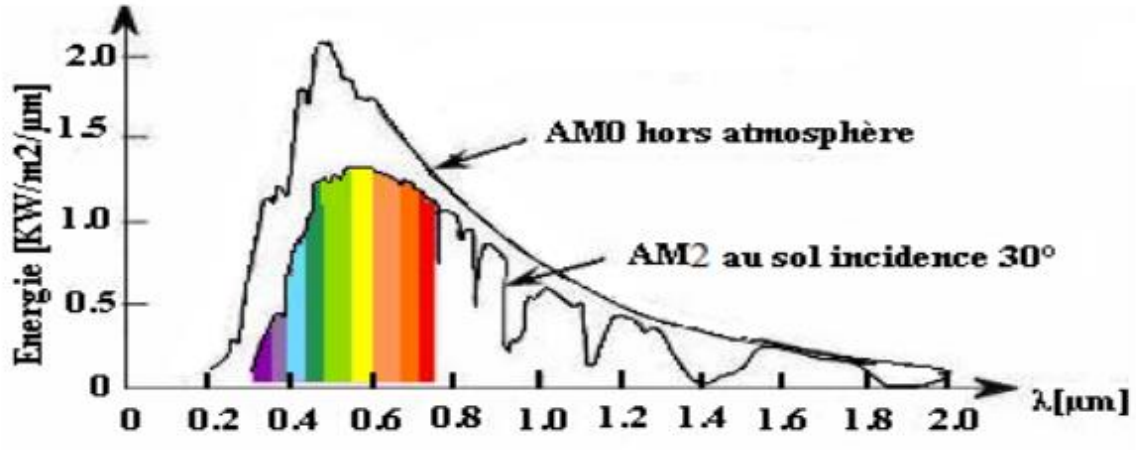

Fig.I. 2. Spectral distribution of solar radiation

III. LOCATION OF THE SUN IN THE SKY

There are two major movements of the earth: the revolution of the world around them The Sun and the earth's rotation around its polar axis. These two movements are Important in solar energy applications. Generally, it is more convenient to study the apparent motion of the Sun in the celestial vault.

The position of the Sun in the celestial vault is detected by two systems of classical coordinates: the time system and the horizontal system.

III.1.The Time System

The Earth rotates around the Sun in an eccentric elliptical trajectory E_0= 0.0167, as shown in **Figure (I.3).** The annual periodicity of this motion makes it possible to understand the phenomenon of the seasons. The distance earth-sun, therefore, varies during the year. On average, the earth-sun distance is used as the basis for the "astronomical unit," 1 ua corresponding to 150.106 km (r_0). It reaches its maximum at the summer solstice (Aphélie; 1,017 ua or 152,106 km) and its minimum at the winter solstice (Perihelia; 0,983 ua or

147,106 km). It depends on the day of the year number j, which varies from 1 to 365 (or 366 for leap years). The ground distance (r) is given by the **equation (I-4)** proposed by and which gives a good precision [19].

$$E_0 = \left(\frac{r_0}{r}\right) = 1.00110 + 0.034221cos\beta + 0.001280sin\beta + 0.000719cos\beta$$
(I-4)

$$+0.000077sin2\beta$$

With: $\beta = \frac{2\pi(j-1)}{365}$ in radians

It is, of course, possible to determine the apparent position, in hourly coordinates, using two angular coordinates: the solar declination (δ) and the hourly angle w.

Fig.I. 3. Earth's motion of revolution around the Sun

III.1.1.The Solar Declination (δ)

The solar declination represents the angle between the Earth-Sun direction and the plane of the equator at noon dedicated solar time. This is a magnitude that often intervenes in the various calculations related to solar radiation. It varies sinusoidally during the year between -23.45 (winter solstice) and 23.45 (summer solstice) and cancels out at equinoxes. Several mathematical formulae have been proposed to calculate approximate values of ($\boldsymbol{\delta}$). **Equation (I.5)** is the one we adopted for our calculations; it gives a ($\boldsymbol{\delta}$) with great precision :

$$\delta(rad) = 0.006918 - 0.399912cos\beta + 0.070257sin\beta - 0.006758cos2\beta$$

$$+0.000907sin2\beta + 0.002697cos3\beta + 0.00148sin3\beta \quad \text{(I.5)}$$

Wit: $\beta = \frac{2\pi(j-1)}{365}$ in radians

III.1.2.The Time Angle (ω)

$\boldsymbol{\omega}$ is the angle between the meridian and the hour circle that contains the Sun. It is positively counted westward from the meridian. It measures the Sun's course in the sky. The hourly angle varies from 15° per hour canceling at noon, dedicated solar time (TSV), it is calculated from the real solar time by the relationship:

$$\boldsymbol{\omega = \frac{\pi}{12}(TSV - 12)} \quad \text{(I.6)}$$

TSV represents dedicated solar time in hours based on earth rotation around its polar axis and its revolution around the Sun. The duration of the solar day varies during the year because:

- ✓ Earth sweeps different surfaces in its rotation around the Sun.
- ✓ The earth axis is tilted about the ecliptic plane.

The real solar time differs from the legal time (TL) of the site considered for three main reasons:

- ✓ The difference in longitude between the site under consideration and the longitude used as the legal time reference (TL) is 4 minutes per degree of longitude.
- ✓ Correction due to legal time changes between summer and winter.
- ✓ The difference in solar time from one day to the next. Moreover, it is the correction of the equation times. This correction varies during the year from -14.3mn to +16.4mn. The approximate formula can calculate it:

$$Et = 9.87sin2\beta - 7.53cos\beta - 1.5sin\beta \quad \text{(I.7)}$$

The relationship between TSV and TL taking these corrections into account is:

$$TSV = TL + Et + \frac{24L}{2\pi} + C \quad \text{(I.8)}$$

L is the longitude of the site considered about the meridian of Greenwich, affected by the sign (+) for the longitudes East and the sign (-) for the longitudes West.

C: correction due to time change.

The time angle is linked to legal time (TL) by the relationship :

$$\omega = \frac{2\pi}{24}(TL + Et + \frac{24L}{2\pi} + C - 24\ ) \quad \text{(I.9)}$$

ω is expressed in degrees, the TL and Et parameters are expressed in hours. **Figure(I.4)** illustrates the entire time coordinate system.

We also calculate the hourly angle of sunset and sunrise by the relationship following :

$\omega_s = cos^{-1}(-\tan(\varphi).\tan(\delta))$ where ω is the latitude and δ the angle of declination [22].

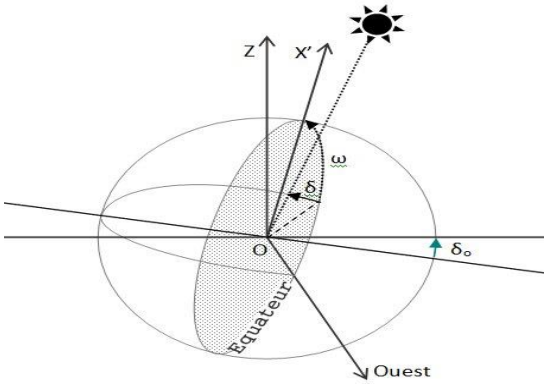

Fig.I. 4. Time Coordinate System

III.2. System of Horizontal Coordinates

The horizontal system is most convenient for common applications. Two angular sizes determine the position of the Sun: The height of the Sun (h) and azimuth ($\boldsymbol{A_z}$)(**Figure I. 5**).

III.2.1.Azimuth (A_z)

It is the horizontal angle of the direction of the Sun with the direction of the south. The knowledge of the azimuth.' $\boldsymbol{A_z}$' makes it possible to calculate the angle of incidence of the rays on a non-horizontal surface [19].

III.2.2.Angular Height

It is the vertical angle of the direction of the Sun with the horizontal plane. Sometimes one talks about the zenith angle, which is the complement to h, such as Az = 90°-h. The height angular of the sun h measures the angular distance of the Sun from the horizon[19].

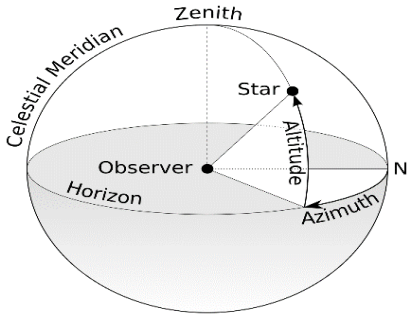

Fig.I. 5. Horizontal Coordinate System.

IV. SOLAR PHOTOVOLTAIC ENERGY

Solar photovoltaic energy has been of great interest in recent years. It is non-polluting energy and provides real solutions to the various problems that The European Union and its Member States are currently discussing climate change and the energy crisis [20].

IV.1.Photovoltaic Effect

The photovoltaic effect is a physical phenomenon specific to certain materials called semiconductors (the best known is silicon). It converts energy directly light from solar rays into electricity through production and transport in a semiconductor material of positive or negative electrical loads as a result of the light. The electrical energy obtained is called photovoltaic energy [21].

IV.2.Photovoltaic System

The photovoltaic system consists of a field of modules and a set of components that adapt the electricity produced by the modules to the specifications of receptors[21].

The following figure represents the synoptic diagram of an autonomous photovoltaic system

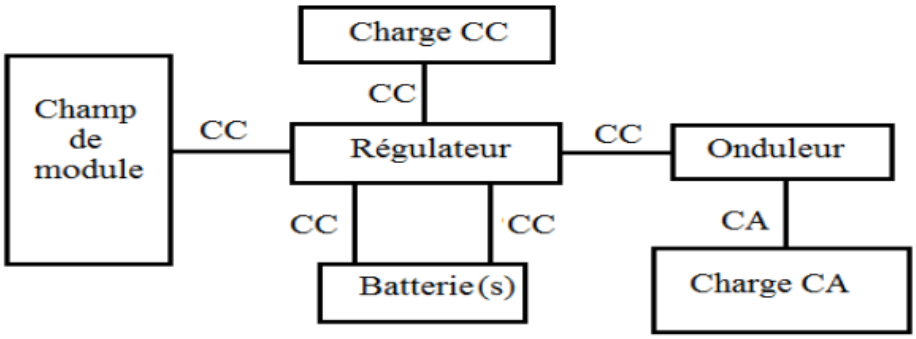

Fig.I. 6. Synoptic diagram of an autonomous photovoltaic system.

IV.2.1.photovoltaic Cell

IV.2.1.1.Description

The photovoltaic cell is a thin layer (two-tenths of a millimeter) of a semiconductor metal, processed to be able to convert light into direct current.

It consists of the stacking of two layers of silicon previously exposed to ion beams, one to phosphorus(-) ions and the other to boron(+) ions. The first layer has an excess of electron and the other a deficit. They are said respectively doped N and doped P. This process is called the «doping» and serves to create an electric field between the two zones where is created a junction called PN, and directed of the zone (P) to the zone (N).

The area (N) is covered by a metal grid that acts as a cathode K while a metal plate A covers the other side of the crystal and acts as an anode. A light beam that strikes the device can penetrate the crystal through the grid and cause a voltage to appear between the cathode and anode.

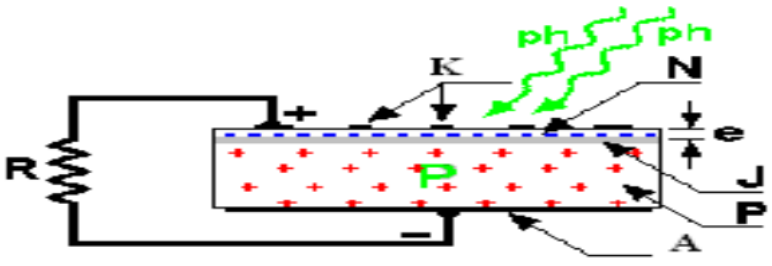

Fig.I. 7. Structure of a photovoltaic cell.

IV.2.1.2.Operating Principle

When the two doped layers are introduced into contact, the extra electrons in the doped N cloth diffuse into the doped P material. The in the beginning doped N vicinity will become positively charged, and the at first doped P region is negatively charged. An electric field is thus created between them, which tends to push the electrons back into the N zone and the holes towards the P zone; a junction known as PN has been formed. By adding metal contacts to the N and P zones, a diode is obtained.

When this diode is illuminated, photons having an energy (h_v) greater than or equal to the bandwidth of the forbidden band E_gExcite the silicon atoms and create positive and negative charges; thus, the electrons and holes created in the P and N regions respectively diffuse and reach the space charge zone, accelerated by the internal electric field, they cross the transition zone. The N region receives electrons and charges negatively. The P region accepts holes and charges positively.

If a charge is placed at the cell terminals, the electrons in the N-zone join the holes in the P-zone via the external connection, creating an electrical current [21].

IV.2.1.3.Characteristic of A Photovoltaic Cell

The function of a cell can be represented by the curve I=f (V), which indicates the evolution of the current generated by the photovoltaic cell according to the voltage at these terminals from the short circuit to the open circuit.

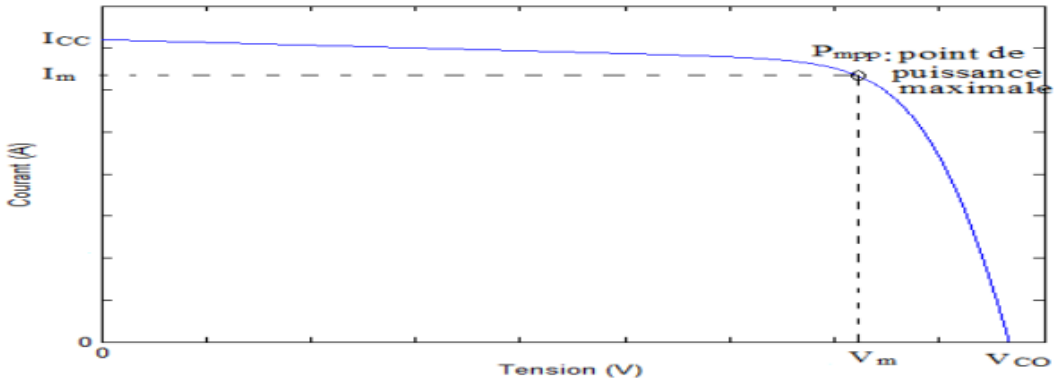

Fig.I. 8. Current-voltage characteristic of a photovoltaic solar cell.

According to the current-voltage characteristic, it is possible to deduce the electrical parameters of the cell:

- ✓ the short circuit current (I_{cc}) corresponding to the current supplied by the cell when the voltage at its terminals is zero.
- ✓ the circuit voltage (V_{co}) corresponding to the voltage that appears at the terminals of the cell when the current flow is zero.

Between these two values, there is an optimum, at a voltage known as the maximum voltage V_m and a maximum current I'm, giving the greatest power (P_{mpp}) or peak power[21].

IV.2.1.4.Influence of Temperature And Illumination

The pattern of the current-voltage characteristic (**Figure (I.9) , Figure (I.10)**) varies according to the environmental conditions (illumination and temperature).

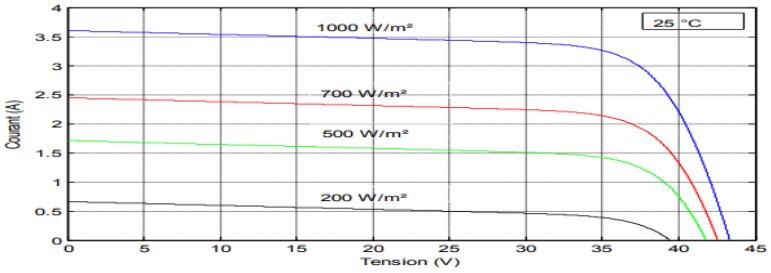

Fig.I. 9. Influence of light on current-voltage characteristics of a photovoltaic cell.

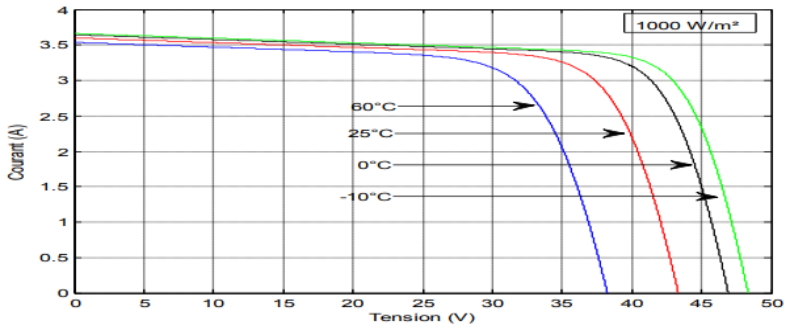

Fig.I. 10. Influence of temperature on current-voltage characteristics of a photovoltaic cell.

The influence of temperature is significant on the current/voltage characteristic. The open-circuit voltage decreases with the temperature increase, but the current varies very little with

temperature. Therefore the maximum power (**Figure (I.11)**) delivered by the photovoltaic cell decreases.

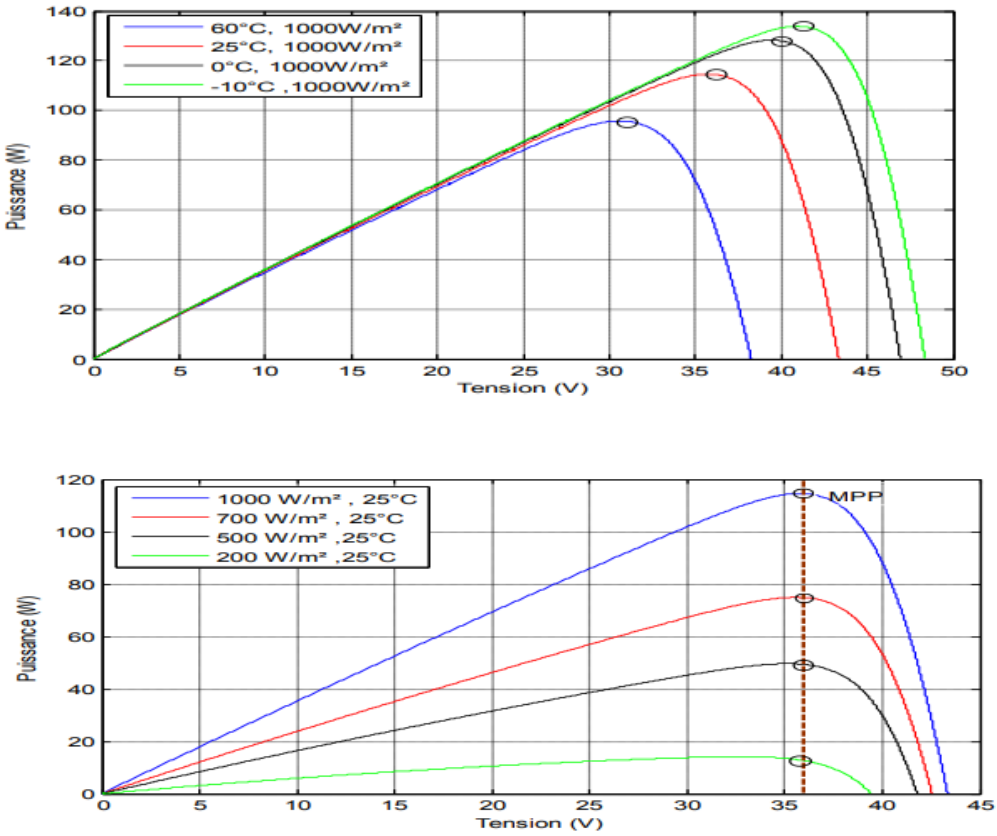

Fig.I. 11. Influence of temperature on the power-voltage characteristic of a cell photovoltaic.

Unlike the variation in temperature, the variation in illumination affects the short circuit current that decreases as the illuminance decreases. The open-circuit voltage is not sensitive to this variation. This implies that:

- ✓ The top-quality energy of the telephone is proportional to the illumination.
- ✓ The most electricity factors are at about the identical voltage (Figure (I.11)).

IV.2.1.5.Efficiency

Apart from the specific efficiency of each cell type (depending on the properties of the material used), the final efficiency depends on the energy captured on the cell surface. This depends on the solar irradiation arriving on the surface of the cell, which, in addition to the

factors mentioned above (latitude, declination, solar angle, etc.), depends on the angle of incidence.

The maximum efficiency, in the same place, is obtained when the solar radiation is perpendicular to the catchment area, that is to say, the angle of incidence of the radiation on the cell is 90° [22].

- **Photovoltaic Cell Manufacturing Technology**

Several technologies are currently being developed. They are classified into three categories (generations) whose detailed description is widely discussed in the literature specialized

- ✓ The first generation consists of monocrystalline silicon cells and Poly-crystalline. Monocrystalline cells have a small yield superior to poly-crystalline cells (15-16%). Nevertheless, they remain relatively expensive due to their high purity requirements. Poly-crystalline cells are cheaper than the previous ones but

  Lower yields. Poly-crystalline silicon is the most widespread technology in the global market due to its good result (11-14%) for controlled manufacturing costs.
- ✓ The second generation: thin-film technology enables the use of fragile layers of silicon that are applied to glass, plastic flexible, or metal. It is cheaper than others and works with low light (in overcast conditions or inside a building), but there

  Performance is less useful than others.
- ✓ The third generation: The third-generation concepts are
  - The photo pile with one or more intermediate bands,
  - Conversion of photons not used directly in the PV cell,
  - Quantum Well Cells

Developments around the third generation target yields ranging from 30-70% and significant cost reductions[22].

IV.2.2.Storage Battery

The fact that solar energy is not available for operation of the powered system requires the use of batteries in the installations autonomous to store energy.

In autonomous solar systems:

- ✓ **Lead-acid batteries:** They constitute the overwhelming majority of the accumulators. Its good technological mastery, its low cost, its good load/discharge efficiency. Its operating conditions are not difficult to satisfy militate in favor of its wide use.
- ✓ **Nickel Cadmium Batteries:** Elles sont les plus chers, mais aussi très résistant aux surcharges et aux décharges, et résistent bien aux basses températures.

IV.2.3.Regulators

In any autonomous photovoltaic system, a so-called regulation system, which is used to control the current going through the accumulators, protecting them against overloads and deep discharges, to maximize its service life. The regulator also allows an optimal transfer of energy from the field photovoltaic to use.

IV.2.4.Inverters

To supply AC equipment, a device Static conversion electronics or DC/AC converter is used for the transformation AC direct current.

IV.2.5.Charge(Users)

There are two types of devices powered by the system, the direct current such as telecommunications equipment, water pumping, and AC in the case of domestic use. This case requires an inverter. The use of photovoltaic energy must be thought in terms of power. It is, therefore, more advantageous to look for consumers running continuously rather than adding an inverter and a 220 V_{ac} consumer.

V. Conclusion

In this chapter, we have defined the majority of meteorological parameters, which are included in the calculations of solar radiation, whether out of the atmosphere or soil. We have

recalled the most important empirical models of solar irradiation. Finally, we presented the experimental data, which are necessary for forming a database for our prediction models throughout this study. There are several ways to process solar radiation data. The next chapter is devoted to the state of the art of solar radiation prediction as that a time series.

CHAPTER II

Machine learning and deep learning techniques

I.INTRODUCTION

Artificial intelligence (AI) has over the past decade become a popular topic both within and outside the scientific community; a wealth of articles in technology and non-technological journals have the subject of machine learning (ML), Deep Learning (DL) and AI. Still there is confusion about AI, ML and DL. The terms are strongly interconnected, but not interchangeable. In this review we do not (try) use technical jargon to better explain these concepts to a clinical audience.

In 1956, a group of computer scientists suggested that computers could be programmed to think and argue, "that every aspect of learning or any other property of intelligence [could] be described in such a precise way that a machine [could] be induced to simulate it." They described this principle as "artificial intelligence." Simply put, AI is an area focused on automating the intellectual tasks normally performed by humans, and ML and DL are specific methods to achieve this goal. This means that they are in the field of AI, but AI includes approaches that do not include any form of "learning." Thus, the sub-field known as symbolic AI focuses on hardcoding rules (i.e., explicit writing) for each possible scenario in a particular area of interest. These rules, written by people, come from a priori knowledge of the particular topic and task to be completed. For example, if you would program an algorithm to modulate the room temperature of an office, he or she probably already know what temperatures are convenient for people to work at and would cool the room, when temperatures rise above a certain threshold and heat when they fall below a lower threshold to program. Although symbolic AI can solve clearly defined logical problems, it often fails in tasks that require higher pattern recognition, such as speech recognition or image classification. These more complicated tasks are where ML and DL methods perform well. This report summarizes machine learning and deep learning methodology for the audience without extensive technical computer programming background [23].

II.MACHINE LEARNING

Machine learning is one of the domains of artificial intelligence. A machine learning algorithm has to aims to analyze the information available from a large number of statistical data to learn to carry out a study without having explicitly been programmed for this upstream.

Machine learning has been a real success in recent years. With the growth exponential number of digital data available, we need to use new analysis methods, and so-called machine learning methods correspond to this need. In this paper, we will study several types of machine learning algorithms, and in this part, we will explain different generalities common to all. We'll start with a quick introduction of the other learning families, and then the general operation of a machine learning algorithm. Then we will list the main advantages and disadvantages, with a closer look at over-learning. Finally, we will explain the simplest machine learning model, as well as its use in this memory [24].

II.1.Basic Principle

II.1.1.Types of Learning in Machine Learning

II.1.1.1.Supervised Learning

Supervised learning is intended to create a predictive function for one of the variables Of our database as a function of others. The variable we want to predict is called "variable to be explained," the other variables used to guide the prediction are the "variables Explanatory". The algorithm tries to learn, by browsing the database available, The different causal links between the explanatory variables and the variable to be explained. Once our model is created, it matches each possible combination of explanatory variables, a prediction of the corresponding variable to be explained. For this, he must group the individuals of the base into subgroups by maximizing the homogeneity of the variable to be defined. Still, the groups must be separated only according to their explanatory variables.

There are two categories of supervised learning:

- Regression algorithms, when the variable to be explained, is quantitative. In this case, the prediction is a value.
- Classification algorithms, when the variable to be explained, is qualitative. In this case, the prediction is often a probability of belonging to the different modalities of the variable in question [25].

II.1.1.2.Unsupervised Learning

Non-served learning occurs when there is no response variable to predict. These algorithms are also used on a database, but in this case, its purpose is to determine the structure present in that database.

To do this, it must, like the supervised algorithms, group individuals into the most homogeneous subgroups possible, however here we no longer have variables to explain, so homogeneity must be done on all variables.

What interests us here is not the prediction of new data, but instead, how groups are determined and what commonalities exist between the individuals of each subgroup.

Of the two learning families discussed above, we will only use supervised learning algorithms in this brief. We aim to estimate a variable to be explained, the S/P ratio of contracts, based on the information available about these contracts, which are the explanatory variables. Supervised learning methods, therefore, appear to be adapted to our problems, such as the CART algorithm (Classification And Regression Trees), bagging, or random forest. The purpose of this section is to present the operation and the main characteristics common to these methods [25].

II.1.1.3.Reinforcement Learning

In this type of study, each marked and unlabeled statistic may want to be utilized to shape the indispensable knowledge. The framework receives a reward for each good or incorrect forecast. Depending on the reward, the subsequent forecast should be generated. At the factor when new information is given to the framework, the framework will pastime to stumble on the excellent execution way or be part of a couple of execution pathways for forecasting and pause for the reward. When the received reward takes place to be greatest with recognition to the previous rewards for equal input, at that point, this pathway flips out to be agreeable Reinforcement gaining knowledge of is utilized in net-primarily based games, for example, Chess [25].

II.1.2. Regression And Classification

II. 1.2.1. Classification

The classification consists of determining the instructions of belonging of new objects from recognized previous examples, the variable to consider can therefore take discrete values called classes. We will see that our problem is multi-class, that is, the exchange of opinions can be represented using several courses, as an antagonist to the binary classification represented using a two-class output variable [26].

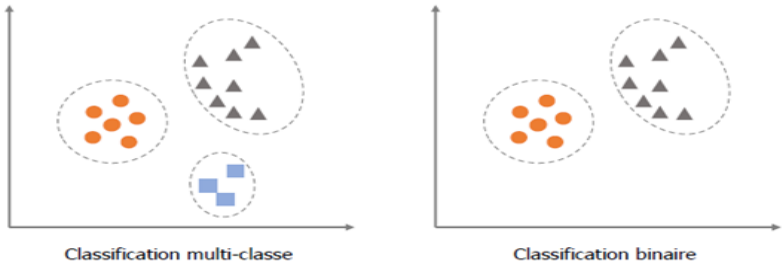

Fig.II. 1. Representation of Observation

II. 1.2.2.Regression

Regression is used when predicting a continuous variable, which can therefore take any value. The classes representing the note change have therefore were considered constant and the regression algorithm allowed to obtain decimal values that we finally rounded to get a vector of predicted levels [26].

III. ALGORITHMS OF MACHINE LEARNING

III .1. Random Forest

III .1.1. Definition Of The Model

The method of machine learning that we will see in this part is the "random forest," better known as random forest, introduced in 2001 by L. Breiman in his publication bearing the name of the method in question. The random forest (RF) method is part of the family of set methods, as is the bagging is seen earlier. As a reminder, the purpose of these so-called established methods is to create several slightly different trees from a single initial base to group these trees and thus reduce the variance of our model. To make a large number of other trees, we can either modify the learning base of each tree or change the algorithm used to create it [24].

III .1.2. How It's Work

In the bagging section, the trees were created from the CART algorithm. It consists in choosing at each node the separation maximizing the intergroup variance among all possible, a separation representing a test on one of the p explanatory variables of the model. However, in this part, this algorithm will be modified so that before each separation, which is called a node, we randomly pull m explanatory variables among the p's of our model. The separation chosen for the node in question will therefore be the optimal one only among the tests involving the selected m variables. Thus, at each node, the information carried per variable p-m drawn randomly is ignored. Apart from the choice of separation, the rest of the algorithm takes place Then, for each bootstrap sample, we create the maximum tree using our new modified algorithm. As for the bagging method, we do not prune the trees because we want to keep the bias as low as possible. Finally, we aggregate the maximum trees together, so that the prediction of our random forest model is the average of the predictions of the maximum trees.

It is interesting to note that bagging could be considered as a method of Random forest, in the particular case where m = p. Indeed, if m = p, then at each node, we choose optimal separation among the tests for one of the selected p variables, that is, all Variables. We find the CART algorithm used in the bagging method.

To remain in line with the ratings used previously:

- B number of bootstrap samples created
- $n_{learning}$ The size of the sample of data used for learning
- $\hat{Y}_i$ estimation of the output variable of individual I by our random forest model
- $\hat{Y}_{RF(m)_{i,k}}$ the estimation of the output variable of individual I buy the maximum tree created from of the bootstrap k sample with a random forest algorithm with the parameter mtry = m

it is achieved that

$$\hat{Y}_i = \frac{1}{B}\sum_{k=1}^{B} \hat{Y}_{RF(m)_{i,k}} \qquad \text{(II.1)}$$

The formula is, therefore, the same as for a bootstrap, the difference in how to build the tree after creating the bootstrap sample [24].

As in the previous part, this method is explained in a prominent and concise manner with a diagram:

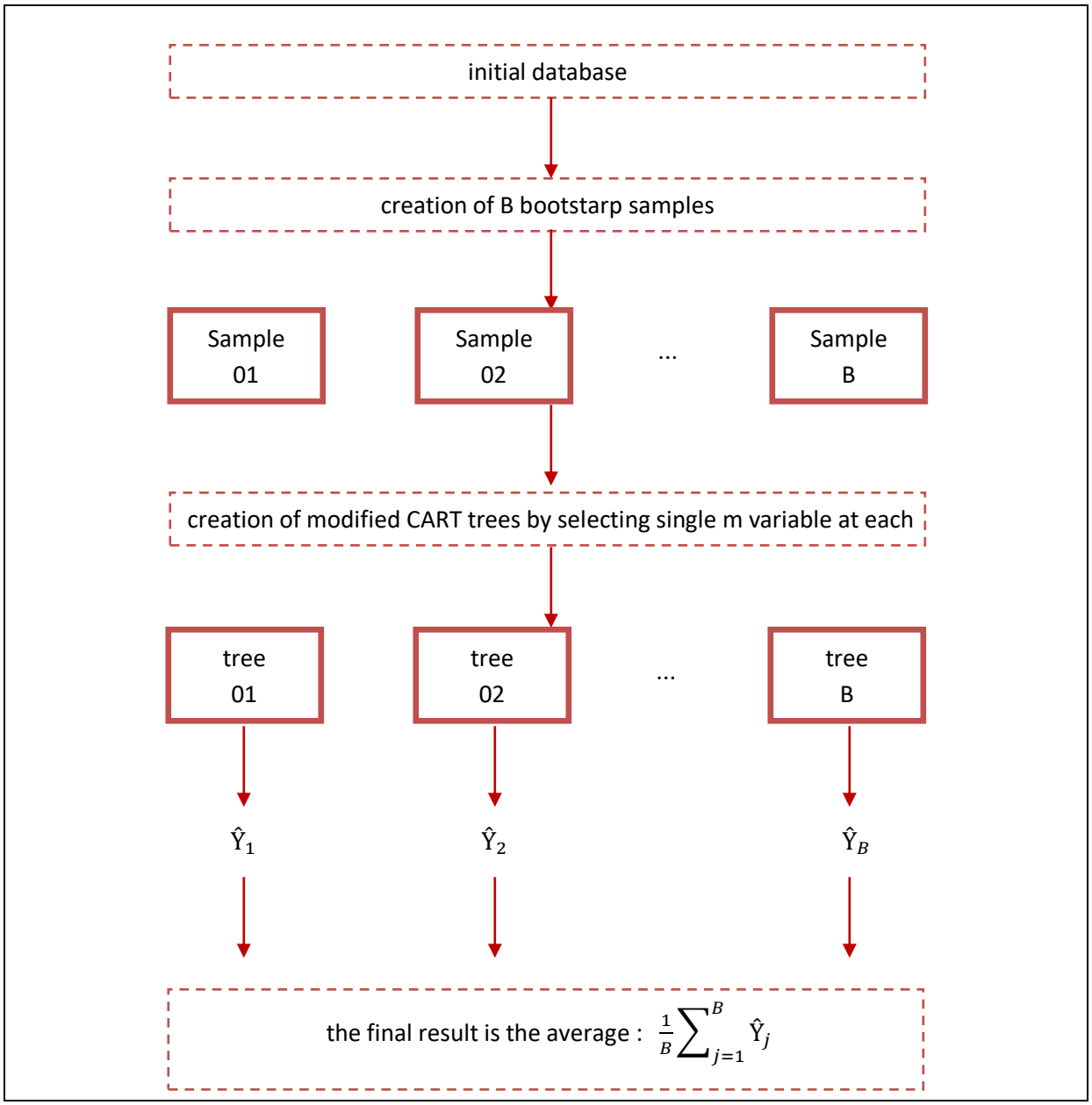

Fig.II. 2. Summary diagram of the random forest method

III .1.3.Creating The Random Forest

As with the bagging model, the first step in creating our random model forest is the determination of the parameters. There are two of them: B, the number of samples bootstrap, and mtry the number of variables selected at each separation. To stay consistent with the previous part, we will note a B_{opti} and $mtry_{opti}$ the chosen parameters.

For parameter B, we will operate in the same way as for the bagging model. We are therefore going to test a random forest model with a large number of trees, here 1000. Then we will find from what number of trees the OOB error, defined in the previous part, is stabilized.

To create our random forest model with 1000 trees to determine B_{opti} We had to set a value for mtry. As a reminder, a good indication of this parameter is mtry $\approx \sqrt{p}$, with p the number of variables that can potentially be selected for each draw.

Here, p = 29, this number is more significant than that mentioned in the section on elimination variables. Indeed, in a random forest, for the category variable, for example, the model can draw separately each of the six variables created during the aggregation of the database. So if we seek to determine the number of variables, the category of the vehicle counts for six here. In contrast, it counted for only one in the selection of variables, which explains this difference.

For now, we set the mtry parameter as follows: mtry $\approx \sqrt{29} \approx 5.4$, we will take so for this test the rounding to the nearest integer, that is to say, mtry = 5. Of course, we will then determine the value of $mtry_{opti}$ using a more sophisticated method. We set it to 5 only the time of the tests to determine B_{opti} [26].

III .1.3.1. Hyperparameter tuning random forest

Although random forests perform well out-of-the-box, there are several tunable hyperparameters that we should consider when training a model. The main hyperparameters to consider include:

n_estimators = number of trees in the forest

max_features(mtry) = max number of features considered for splitting a node

max_depth = max number of levels in each decision tree

min_samples_split = min number of data points placed in a node before the node is split

min_samples_leaf = min number of data points allowed in a leaf node

bootstrap = method for sampling data points (with or without replacement)

III .1.3.2. Selection Of The Number Of Trees

We can see on this graph the evolution of the OOB error as a function of the number of bootstrap samples, and therefore of the number of trees. We can see that the error stabilizes later as for the bagging model. Here we will now fix B_{opti} = 400 [26].

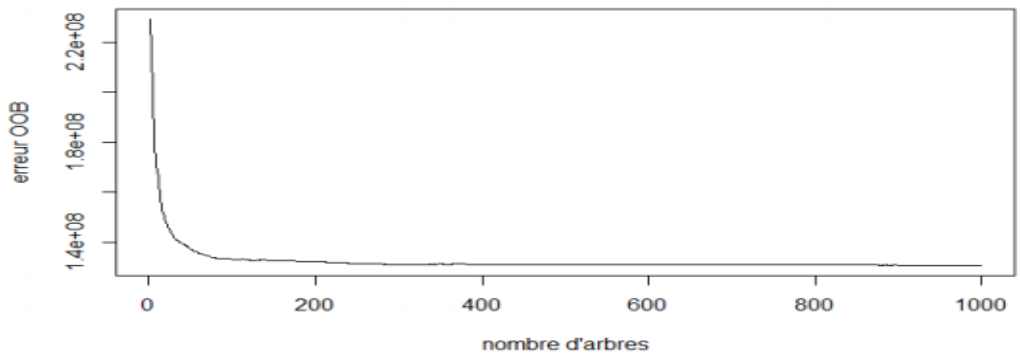

Fig.II. 3.OOB error depending on the number of trees in the random forest model.

III .1.3.3. Selecting The Mtry Parameter

Now that we have determined the first parameter, Bopti, let's move on to the choice of the second, $mtry_{opti}$.

For the mtry parameter, we will create models with different values for mtry, and then we will test these models on our validation basis to calculate the MSE error. We will choose the mtry that gave the model with the lowest MSE.

We first had to choose a range of mtry values that we are going to test. As the indication $\text{mtry} \approx \sqrt{p}$ gave us a result between 5 and 6, we took a range around these values, so we have arbitrarily set it between 2 and 12. If we subsequently find that this range is not sufficient to make our choice, we give ourselves the option to change it later [26].

Finally, we specify that all models tested on this range are tested with 400 bootstrap samples, as we have chosen previously.

We can see that the indication $\text{mtry} \approx \sqrt{p}$ looks good because models with mtry values of 5 and 6 are better than others. We now fix $mtry_{opti} = 6$, whose model got a slightly smaller error than the one with mtry = 5.

Thus, the estimate made in this part will be composed of 12 random forest models, each having 400 bootstrap samples and a value of 6 for the parameter mtry. The models thus created obtained an MSE on the test basis of 0.87 by predicting the S / P and 0.74 by the number of claims. These are the best scores so far [24].

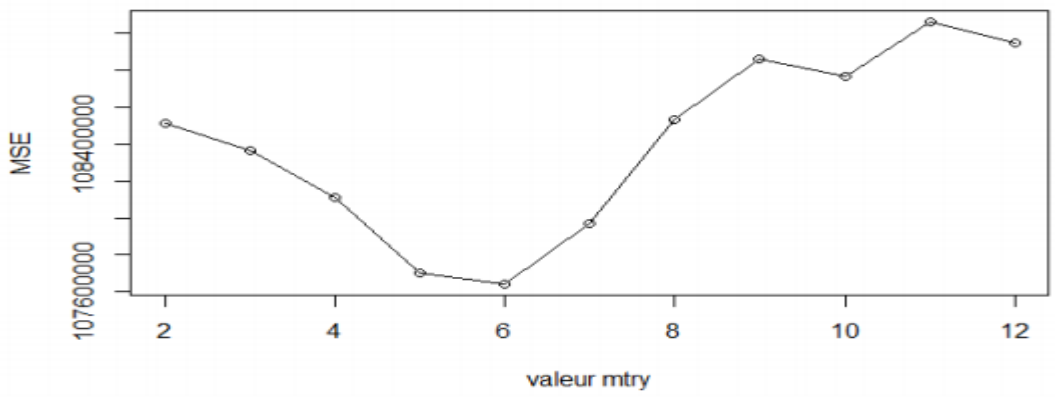

Fig.II. 4.MSE error depending on mtry value.

III .2. Gradient Boosting Machine

III .2.1. Definition of The Model

Gradient Boosting Machine (GBM) is a machine learning technique used for regression or classification problems.

The most usual approach of boosting is likely the one developed via Freund and Schapire in 1995: the AdaBoost (for Adaptive Boosting). However, it will now not be precise in this brief. However, the reader is referred to Article A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting for more critical data about this method. Like different boosting methods, the Gradient Boosting optimizes the overall performance of a set of so-called "low" prediction fashions by means of assembling them into a remaining model. This is known as a "low" prediction model, a classification or regression approach that is simply barely greater nice than a random draw. The low prediction mannequin normally used with a Gradient Boosting is a CART choice tree. More concretely, this approach of the Gradient Tree Boosting then consists in making a succession of choice bushes the place every mannequin is constructed on the residual error of the preceding one [25].

III .2.2. The Cart Decision Tree

The Classification and Regression Tree (CART) is a supervised statistical learning method designed to construct a prediction model. The main idea of the CART tree is to recursively and binary partition the space of the explanatory variables to obtain all possible values of the variable to be predicted. Suppose we have an n-sample of a variable Y with p predictors [x_1,..., x_p]. The goal is to be able to predict Y values for new [x_1,..., x_p] values. Let's insert t_p, the parent node and t_g, t_d the child nodes, respectively, left and right. From the set of observations on the variable Y, the CART tree selects at each iteration and, therefore, at each nodet_p, the variable x_j and its value x_j which best segment the space of the variables in two sub-most homogeneous areas possible in the tg and td nodes. This notion of homogeneity results in a function of impurity noted I (.). Several functions define the impurity i(t) of a node t. Still, in the case of the study of a continuous random variable Y, as is the case here, the node variance may represent a good indicator of the heterogeneity present in the node. The impurity of a parent node remaining constant regardless of the cutting performed to create the yarn nodes, maximizing the homogeneity in the yarn nodes is then equivalent to optimizing the variation of impurity Δ_i defined as:

$$\Delta_i = i(t_p) - E(i(t_{fils})) \quad \text{(II.4)}$$

This concept of impurity reduction can be seen in (**Figure II.5**), where the removal of impurity is more critical in the illustration on the left than in the illustration on the right.

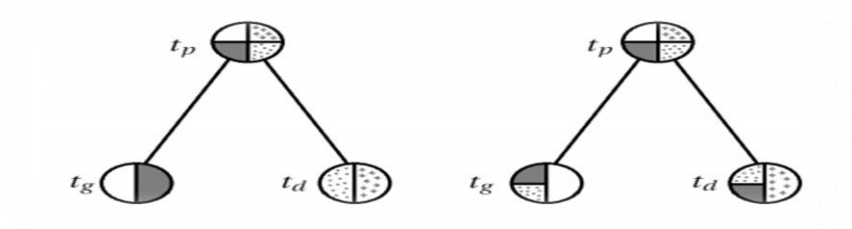

Fig.II. 5. Illustration of the CART tree impurity reduction concept.

The problem of maximizing impurity Δ_i defined in equation (II.4) can also be written:

$$max_{x_i<x_j, j=1,\dots,p} [i(t_p) - P_g * i(t_g) - P_d * i(t_d)] \quad \text{(II.5)}$$

where P_g and P_d are the probabilities of the left and right nodes, respectively. This problem translates the way the algorithm works: it scans all possible values for all variables x_i for j = 1,...,p to find the best cutout that will maximize the variation of impurity Δ_i. If the node variance represents the impurity function, then the maximization problem 3.2 can be rewritten:

$$max_{x_i<x_j, j=1,\dots,p} [P_g * Var(t_g) + P_d * Var(t_d)] \quad \text{(II.6)}$$

For more details on the machine learning method CART, the reader can refer to Roman Timofeev, Classification, And Regression Trees[27].

III .2.3. Gradient Boosting Machine Algorithm

The Gradient Boosting is an iterative algorithm that initially distributes weights equal to all predictions and then adapts them to each step, so that bad predictions are over-weighted to the next step so that the "low" prediction model Pay more attention to it. Let's go back to Friedman's original article by using his notes to develop the idea of Gradient Boosting in more detail. We have a sample size (n) consisting of (p) explanatory variables. Note x_i = ($x_{i,1}$,... $x_{i,p}$) $\in R^p$, The vector of the explanatory (p) variables corresponding to the observation i. The objective is to find an approximation $F_{\text{approx}}(x)$ of a function F(x) linking the explanatory variables x to the response variable y and minimizing the expectation of a certain loss function

L(y, F(x)) on the attached distribution of x and y [28]. Mathematically, the function F(x) is then defined by:

$$F^*(x) \in argmin_F E_{y,x}[\mathrm{L(y,F(x))}] \quad \text{(II.7)}$$

Algorithm Gradient Boosting Machine

- Initialize $F_0(x) = argmin_p \sum_{i-1}^{N} L(y_i, p)$
- For $m = 1$ to M do :
  - Step 1. Compute the negative gradient
$$\bar{y}_i = -\frac{\partial L(y_i, F(x_i))}{\partial F_{x_i}}$$
  - Step 2. Fit a model
$$\alpha_m = argmin_{\alpha,\beta} \sum_{i=1}^{N} [\bar{y} - \beta h(x_i; \alpha_m)]^2$$
  - Step 3. Choose a gradient descent step size as
$$p_m = argmin_p \sum_{i-1}^{N} L(y_i, F_m - 1(x_i) + ph(x_i; \alpha))$$
  - Step 4. Update the estimation of F(x)
$$F_m(x) = F_{m-1}(x) + p_m h(x; \alpha_m)$$
- End for
- Output the final regression function $F_m(x)$

Table.II.1. Algorithm Gradient Boosting Machine

III .2.4. Hyperparameter Tuning Gradient Boosting Machine

To avoid over-learning of the data and to degrade future predictions, the Gradient Boosting has several hyper-parameters that allow optimizing its use by limiting this over-learning phenomenon [29]:

A-Min_Samples_Split

- Defines the minimal range of samples (or observations) which are required in a node to be viewed for splitting.

- It is used to manage over-fitting. Higher values stop a model from mastering members of the family, which would possibly be noticeably unique to the specific pattern chosen for a tree.
- Too extreme values can lead to under-fitting; hence, it ought to be tuned to the use of CV.

B- Min_Samples_Leaf

- Defines the minimal samples (or observations) required in a terminal node or leaf.
- It is used to manage over-fitting comparable to min_samples_split.
- Generally, decrease values ought to be chosen for imbalanced type troubles due to the fact that the areas in which the minority classification will be in the majority will be minimal.

C- Max_Depth

- The most depth of a tree.
- Used to manipulate over-fitting as greater depth will permit mannequin to analyze family members very particular to a specific sample.
- Should be tuned the usage of CV.

D-Max_Features

- The range of facets to reflect on consideration while looking for an exceptional split. These will be randomly selected.
- As a thumb-rule, the rectangular root of the complete quantity of facets works excellent. However, we ought to take a look at up to 30-40% of the total variety of features.
- Higher values can lead to over-fitting; however, relies upon case to case.

E- Learning_Rate

- He determines the have impact of every tree on the closing effect. GBM works by way of beginning with a preliminary estimate, which is up to date the usage of the output of

every tree. The gaining knowledge of parameter controls the magnitude of this alternate in the forecast.

- Lower values are commonly favored as they make the mannequin strong to the unique traits of the tree and, as a consequence permitting it to generalize well.
- Lower values would require a greater wide variety of timber to mannequin all the family members and will be computationally expensive.

F- N_estimators

- The number of sequential trees to be modeled.
- Though GBM is relatively robust at a higher number of trees, it can still overfit at a point. Hence, this should be tuned using a CV for a particular learning rate.

G- Loss

- It refers to the loss function to be minimized in each split.
- It can have various values for classification and regression case. Generally, the default values work fine. Other values should be chosen only if you understand their impact on the model.

III .3.Machine Learning versus Deep Learning

Machine learning (ML) is extensively utilized in identifying several types of attacks. An ML methodology could assist the network admin in performing required actions for averting intrusions. Nevertheless, most traditional ML methods reside within shallow learning and usually emphasize to feature engineering and selection. With enormous intrusion data that rises in the real-time network environment, external learning is unable to resolve the classification issue efficiently [30]. A deep learning approach possesses the potential to extract improved representations from the dynamic datasets and is capable of creating many efficient prototypes. G.Hinton et al. [31] introduced the hypothesis of DL in the year 2006, and over the years, the theory underwent a spectacular rise in the area of ML. Most recent ML methodologies work well because of human-designed representations and input features. In ML, ‘representation learning’ or ‘feature learning’ represents a set of methods that makes a system capable of automatically learning the representations required for detecting features

from the training dataset. DL, on the contrary, could be considered as establishing both representation learning and machine learning together. DL pursuits to together learn essential features along with multiple levels of cumulative intricacy and abstraction and the concluding prediction. **Figure (II.6)** illustrates the fundamental difference between ML and DL, where traditional ML involves manual feature selection, and on the contrary, DL employs automated feature selection.

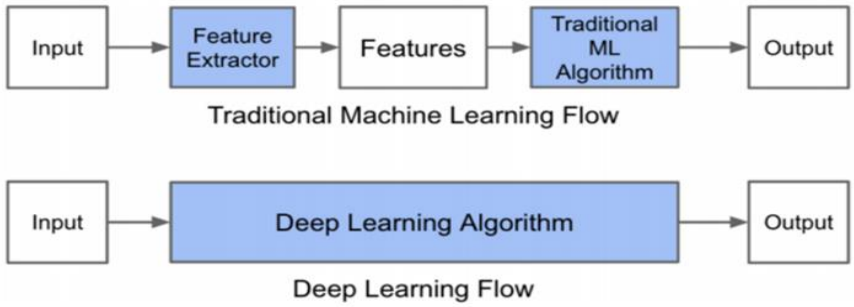

Fig.II. 6. Machine learning and Deep learning

IV.DEEP LEARNING

Deep Learning is a new area of ML research, which has been introduced to bring the ML closer to its primary objective: artificial intelligence. It concerns algorithms inspired by the structure and functioning of the brain. They can learn multiple levels of representation to model complex relationships between data.

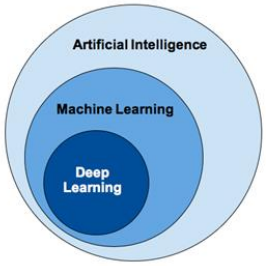

Fig.II. 7. The relationship between artificial intelligence, ML, and deep learning.

Deep Learning is based on the idea of artificial neural networks and is designed to manage large amounts of data by adding layers to the network. A deep learning model can extract characteristics from raw data through multiple layers of processing consisting of multiple linear and non-linear transformations and learn about these characteristics step by

step across each layer with minimal human intervention. Over the past five years, deep learning has moved from a niche market where only a handful of researchers were interested in the area most sought by researchers. Research related to deep learning now appears in top journals like Science, Nature, and Nature Methods, to name a few. Deep learning has conquered the GO, learned to drive a car, diagnosed cancer and autism, and even become an artist. The term "Deep Learning" The time period "Deep Learning" used to be first added to ML via Dechter (1986) and artificial neural networks by using Aizenberg et al. (2000) [32].

IV.2. History of Deep Learning

This paper presents a history of deep learning from Aristotle to the present. The various milestones are summarized in this table

Year	contributor	Contribution
300 AC	Aristotle	introduction of associationism, beginning of the history of humans trying to understand the brain
1873	Alexander Bain	introduction of Neural Groupings as the first models of neural networks
1943	McCulloch and Pitts	Introduction of the McCulloch–Pitts (MCP) model considered as the ancestor of artificial neural networks
1949	Donald Hebb	consider as the father of neural networks, and it introduces the Hebb learning rule that will serve as the foundation for modern neural networks
1958	Frank Rosenblatt	introduction of the first perceptron
1974	Paul Werbos	introduction of retro propagation
1980	Teuvo Kohonen	introduction of self-organizing cards
1980	Kunihiko Fukushima	introduction of Neocognitron, which inspired convolutive neural networks
1982	John Hopfield	introduction of Hopfield networks
1985	Hilton and Sejnowski	introduction of Boltzmann machines
1986	Paul Smolensky	introduction of Harmonium, which will later be known as restricted Boltzmann machines

1986	Michael. Jordan	definition and introduction of recurrent neural networks
1990	Yann LeCun	Introduction of LeNet and demonstrated the capabilities of deep neural networks
1997	Schuster and Paliwal	introduction of bi-directional recurrent neural networks
1997	Hochreiter and Schmidhuber	introduction of LSTM, which have solved the problem of vanishing gradient in recurrent neural networks
2006	Geoffrey Hinton	introduction of the Deep Belief Network
2009	Salakhutdinov and Hinton	introduction of Deep Boltzmann Machines
2012	Alex Krizhevsky	Introduction of AlexNet, which won the ImageNet challenge

Table.II.2. History of Deep Learning [33].

IV.3. Why Deep Learning?

ML algorithms work well for a wide variety of problems. However, they have failed to solve some significant issues of AI, such as voice recognition and object recognition. The development of deep learning was motivated in part by the failure of traditional algorithms in such a task of AI. But it wasn't until more data was made available thanks to Big Data and connected objects and computing machines became more powerful that the real potential of Deep Learning was understood.

One of the significant differences between Deep Learning and traditional ML algorithms is that it adapts well. The greater the amount of data provided, the better the performance of a Deep Learning algorithm. Unlike several ML classics algorithms that have a terminal higher than the amount of data they may receive, sometimes called "performance plateau," Deep Learning models do not have such limitations (theoretically), and they even went so far as to surpass human performance in areas such as image processing [34].

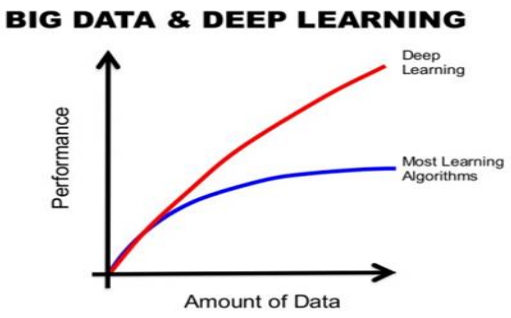

Fig.II. 8. The performance difference between Deep Learning and most ML algorithms depends on the amount of data.

IV.4. Deep Learning Application Areas

In-depth learning is gradually investing in our daily lives:

- Voice recognition
- Automatic tagging of music pieces
- Advanced speech synthesis
- The design of new pharmaceutical molecules

All these applications now use Deep Learning techniques [34].

V- ALGORITHMS OF DEEP LEARNING

There are a large number of variations of deep architectures. Most of them are derived from some original parenting architectures. It is not always possible to compare the performance of all architectures, as they are not all evaluated on the same datasets. Deep Learning is a rapidly growing field, and new architectures, variants, or algorithms appear every week.

V.1.Deep Neural Networks

Deep Neural Networks (DNNs), additionally acquainted as Neural Networks (NNs), are computing device mastering fashions which are stimulated from the human brain. The first fashions of neural community developed by way of a neurophysiologist Warren McCulloch

and a mathematician Walter Pitts in 1943. They mentioned how neurons would possibly work. Neural networks consist of the layer of entering neurons or alerts, which can be different characteristic values, an output layer the place the result of the community obtained, and a quantity of a range of hidden layers between the enter and output layers. Also, each layer has a few or various neurons.

Input alerts are surpassed thru the network, layer by way of layer, by way of the usage of the weighted connections to, in the end attain the output layer. At some neurons, a nonlinear feature can be triggered. The purpose of gaining knowledge of the method is discovering weights that would make the neural community display favored behavior. This is the instance of Multilayer Perceptron (MLP), which is additionally referred to as Feedforward Neural Networks (FNN). An accepted structure of artificial neural networks is proven in **Figure (II.9)**.

Despite the reality that the feedforward neural networks have been efficaciously applied and bought higher overall performance in many tasks, it does now not think about the transient prospect that characterizes sequential data. That is, they are no longer moderately proper when it comes to the statistics that rely on the preceding data. To remedy such a variety of troubles, the neural networks have advanced to so-referred to as recurrent neural networks. In the subsequent subsections [43].

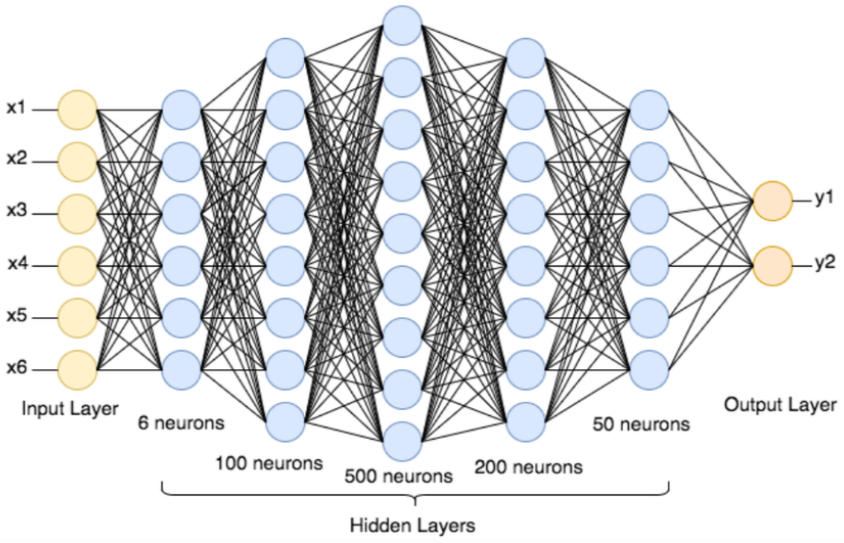

Fig.II. 9. Deep Neural Network architecture[35].

V.1.1.Perceptron

The fundamental unit of neural networks is one of the types of artificial neurons, called a perceptron. A perceptron takes a fixed number of inputs and produces a single output. The way of computing the result essentially has several steps, such as taking an input, calculating the weighted sum by introducing weights, bias term, and applying activation function. The process described above can be formalized as follows:

The perceptron has n inputs represented as an input vector x = $(x_1, x_2, \ldots, x_n)$. Each input has an assigned weight that defined by a vector of weights w = $(w_1, w_2, \ldots, w_n)$. Consequently, the weighted input values are combined, which gives the weighted sum:

$$\varepsilon = w \cdot x = \sum_{i=1}^{n} w_i \cdot x_i \qquad \text{(II.8)}$$

At this step, the activation function is applied to the weighted sum to calculate the output, and the weighted sum is compared with a threshold θ to produce an output y that is either 0 or 1, depending on whether or not it exceeds the threshold. Thus,

$$y = \sigma(\varepsilon) = \begin{cases} 1, & \varepsilon \geq 0 \\ 0, & \varepsilon < 0 \end{cases} \qquad \text{(II.9)}$$

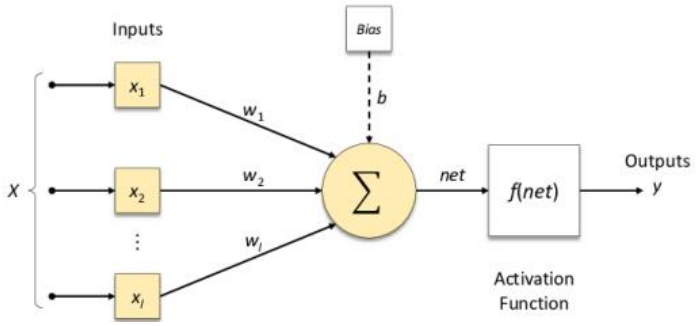

Fig.II. 10. The Perceptron.

V.1.2.Multi-layer Perceptron

A multilayer perceptron (or neural network) is a shape composed of countless hidden layers of neurons the place the output of a neuron of a layer will become the enter of a neuron of the subsequent layer. Moreover, the end result of a neuron can additionally be the enter of a

neuron of the identical layer or of the neuron of preceding layers (this is the case for recurrent neural networks)

nodes that are no goal of any connection are referred to as entering neurons. An MLP that must be utilized to enter patterns of dimension n needs to have n enter neurons, one for every dimension. Input neurons are usually enumerated as neuron 1, neuron 2, neuron three

- nodes that are no supply of any connection are referred to as output neurons. An MLP can have extra than one output neuron. The range of output neurons relies upon on the way the goal values (desired values) of the education patterns are described.
- all nodes that neither enter neurons nor output neurons are referred to as hidden neurons.

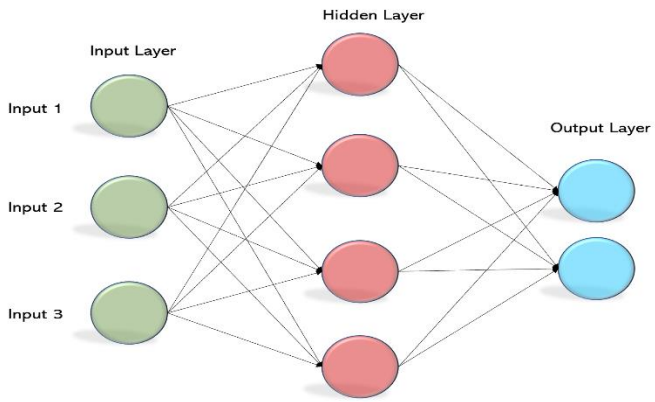

Fig.II. 11. Multilayer Perceptron (MLP) structure.

V.1.3.Activation Functions

An activation function is a mathematical function applied to a combination of x_i inputs. It thus introduces the principle of non-linearity, which makes it possible to adapt and differentiate the results. The choice of the activation function depends on the application, whether it is necessary to have binary outputs or not. In this work, the functions Sigmoid, Softmax, Tanh as well as ReLu are used [36].

- **Sigmoid**

The sigmoid function is defined by:

$$f(x) = \frac{1}{1+e^{-x}} \; Or \; x \in \mathrm{R} \tag{II.9}$$

The goal is to convert the input value into a probability of 1 if it is a very large positive number, and conversely, to 0 if the input is a very large negative number

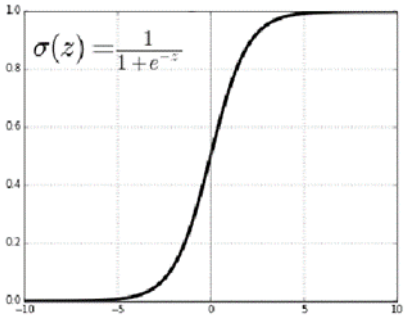

Fig.II. 12. Sigmoid Function Graph.

- **Softmax**

The Softmax function, or normalized exponential function, is defined by:

$$f(x)_j = \frac{e^{z_j}}{\sum_{K=1}^{K} e^{z_k}} \text{ for all } \mathrm{j} \in \{1, \dots, \mathrm{K}\}, \mathrm{x} = \{\mathrm{x1}, \dots, \mathrm{xk}\} \text{ ou } \mathrm{k} \in \mathrm{R} \tag{II.10}$$

The function gives an output vector of K strictly positive real numbers and sum 1.

- **Tanh**

The hyperbolic tangent function, noted Tanh, is defined by:

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{II.11}$$

The output in Tanh will be stored in the field [-1,1].

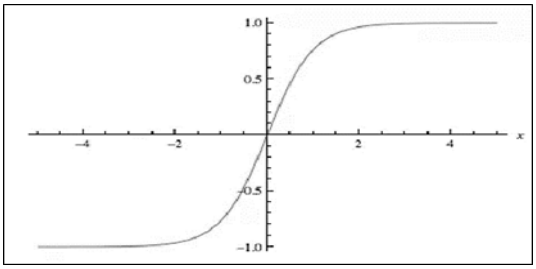

Fig.II. 13. Tanh Function Graph.

- **ReLu**

The ReLu linear grinding unit function is the simplest function, defined by:

$$f(x) = \begin{cases} 0 \ for & x < 0 \\ x \ for & x \geq 0 \end{cases} \quad \text{(II.12)}$$

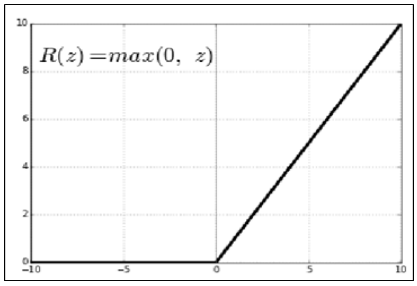

Fig.II. 14. ReLu Graph.

The output is therefore 0 if the input is less than 0. Otherwise, it will be equal to the input.

V.1.4.Optimization functions

Generally, the functions of optimization in neural networks have the objective of calculating the gradient to find optimized values (weight), and this by modifying the weights in the opposite direction, this cycle is repeated until the objective function is minimized. In this work, different optimizations functions are tested: Adam, Adamax, Adadelta, and SGD [37].

- **SGD**

The word “stochastic” refers to a random probability system or process. Therefore, in stochastic gradient descent, some samples are randomly selected instead of the dataset of each

iteration SGD (Stochastic Gradient Descent), which updates the parameters for each example of the x_i dataset and y_i label (label).

- Initialize with x_0 (at random)
- Repeat: $x_t + 1 = x_t - \eta \times \nabla(x_t)$ Until convergence
- η: modulates the correction (η too low, slow convergence; η too high, oscillation)
- $\nabla(x_t)$: The gradient at point x_t shows the direction and importance of the slope in the vicinity of x_t.

This method is usually faster. However, due to frequent updates, convergence becomes more difficult (find the minimum objective function).

- **Adam**

"Adam Adaptive Moment optimization" is one of the latest and most effective algorithms for gradient descent optimization. Adam calculates the exponential mean of the gradient as well as the squares of the gradient for each parameter. The learning rate is then multiplied by the mean of the gradient and divided by the square root of the exponential mean of the gradients. Then the update is added.

$$v_t = \beta_1 \times v_{t-1} - (1 - \beta_1) \times g_t \quad \text{(II.13)}$$

$$s_t = \beta_2 \times s_{t-1} - (1 - \beta_2) \times g^2{}_t \quad \text{(II.14)}$$

$$\Delta w_t = -\eta \frac{v_t}{\sqrt{s_t + \epsilon}} \times g_t \quad \text{(II.15)}$$

$$w_{t+1} = w_t + \Delta w_t \quad \text{(II.16)}$$

- Η: initial learning rate.
- g_t: the gradient in a time t.
- v_t: the exponential average of gradients.
- s : the exponential average of the squares of the gradients.
- $\beta_1, \beta_2, \epsilon$: hyperparameters usually initialized at 0.90, 0.99, and 1e-10.

- **Adamax**

Adamax is a generalization of Adam to get a more stable convergence, or instead of taking:

$$s_t = \beta_2 \times s_{t-1} - (1 - \beta_2) \times g^2{}_t \quad \text{(II.17)}$$

s_t become :

$$s_t = \beta_2^p \times s_{t-1} - \left(1 - \beta_2^p\right) \times g_t^p \quad \text{(II.18)}$$

Ou p = ∞.

- **Adadelta**

Adadelta proposes an adaptation of the gradient depending on its second-order moment and the state of the corrected network layer and uses a fixed size w to take into account the sum of the squares of gradients accumulated in the past.

V.1.5.Dropout Regularization

Deep neural network (DNN) fashions have several parameters and can model relatively composite functions. This capability is a boon and a bane. Such prototypes would often overfit on the training-set and would drop accuracy and generalizability over the test-set. Regularization in ANN terminology speaks of the technique of regulating neural network layers for stopping the over-fitting. Dropout (also recognized as dropout chance or dropout rate) is the most extensively utilized regularization approach in DL. During the learning manner, the hidden layer(s) neurons are chosen randomly and are discarded, relying on the dropout rate. Precisely, randomly chosen neurons are dropped-out, i.e., dropped out neurons could not replace weights anymore, accordingly supporting the learning manner to avert the problem of overfitting [38].

V.2.Recurrent Neural Networks

Humans don’t start thinking from scratch every second. By reading this essay, you understand each word according to your understanding of the preceding words. You don’t throw everything away and start thinking from scratch again. Your thoughts have perseverance.

Traditional neural networks (TNNs) can't do that, and that seems to be a significant gap. For example, imagine that you want to classify the type of event that occurs at each stage of a movie. It is not known how a traditional neural network could use its reasoning on previous events in the film to inform them later.

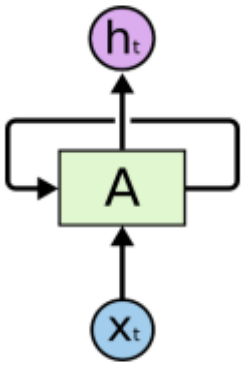

Fig.II. 15. Recurrent neural networks have loops.

Recurrent neural networks (RNNs) solve this problem. They are networks with loops that allow information to persist. In **figure II.15** above, a neural network segment, "A," looks at an Xt input and provides a ht value. A loop allows information to be passed from one stage of the network to another.

These loops make recurring neural networks a bit mysterious. However, if you think a little more, it turns out that they are not all different from a standard neural network. A network of recurrent neurons can be considered as multiple copies of the same network, each transmitting a message to a successor, as shown in **Figure II.16**. Consider what happens if we unravel the loop:

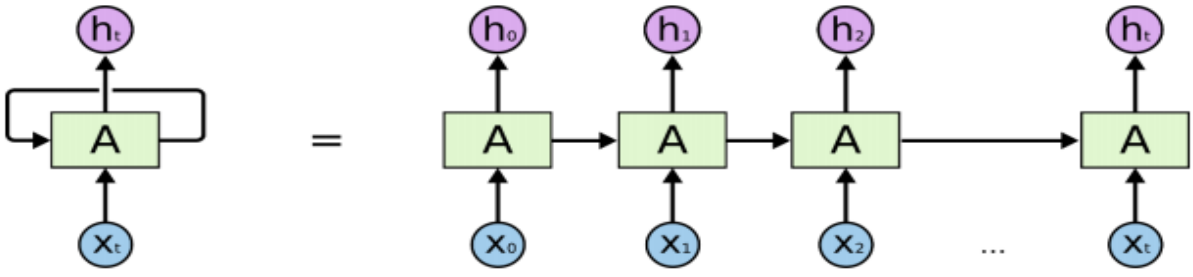

Fig.II. 16. A recurring neural network unfolded.

This chain nature reveals that recurring neural networks are intimately linked to sequences and lists. They are the natural architecture of the neural network to be used for such data.

In recent years, there has been an incredible success in applying NRNs to a variety of issues: speech recognition, language modeling, translation, captioning of images, etc [39].

V.2.1.The Problem of Long-Term Dependencies

One of the appeals of RNNs is the concept that they may be capable of joining preceding records to the current task, such as the use of remaining video frames may inform the appreciation of the contemporary structure. If RNNs ought to do this, they'd be instrumental. But can they? It depends [39].

Sometimes, we solely want to seem at current statistics to operate the current task. For example, reflect on consideration on a language mannequin is making an attempt to predict the subsequent phrase-based totally on the preceding ones. If we are making an attempt to envision the ultimate phrase in "the clouds are in the sky," we don't want any in addition context – it's rather apparent the subsequent expression is going to be the sky. In such cases, the place the hole between the applicable records and the region that it's wanted is small, RNNs can examine to use the previous data.

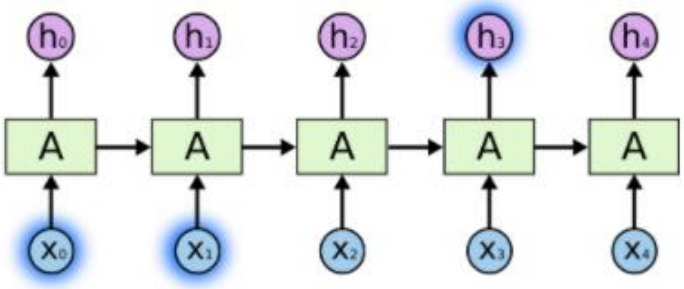

Fig.II. 17. The Problem of Long-Term Dependencies-1.

But there are additionally instances the place we want extra context. Consider making an attempt to predict the remaining phrase in the textual content "I grew up in France… I communicate fluent French." Recent data suggests that the next phase is likely the identity of a language. However, if we favor to slender down which language, we want the context of

France, from in addition back. It's absolutely feasible for the hole between the applicable statistics and the factor the place it is required to emerge as very large.

Unfortunately, as that gap grows, RNNs become unable to learn to connect the information.

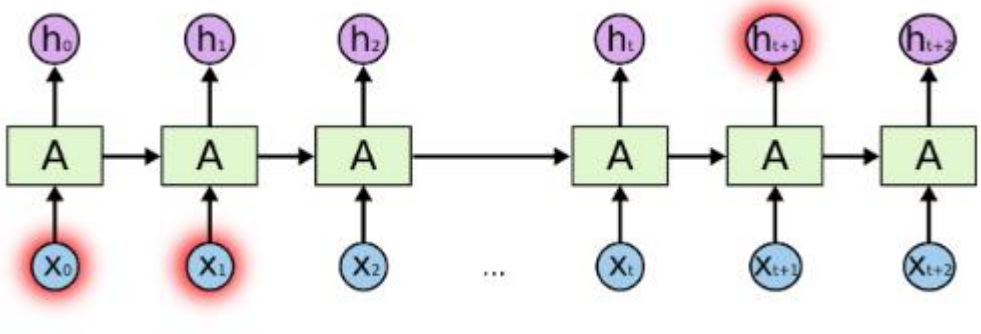

Fig.II. 18. The Problem of Long-Term Dependencies-2.

In theory, RNNs are succesful in coping with such "long-term dependencies." A human ought to cautiously pick out parameters for them to resolve the toy troubles of this form. Sadly, in practice, RNNs don't show up to be in a function to find out about them. The hassle used to be explored in depth by using way of Hochreiter (1991) [German] and Bengio et al. (1994), who decided some pretty essential causes why it may additionally be problematic. Thankfully, LSTMs don't have this problem!

V.3.Long Short-Term Memory (LSTM)

V.3.1.Introduction

Short-term, long-term memory networks (LSTM) is a synthetic structure of recurrent neuronal community (RNN) used in the discipline of deep learning. ... LSTM networks are well suited to classify, process, and make predictions based entirely on time collection data, as there may be delays of unknown time between vital opportunities in a time series.LSTM is a kind of recurrent neural community that addresses the disappearing and exploding gradient problems that can occur with NRNs, preventing them from having long time sequences. Therefore, the MTSA may have a "short-term memory," as the name suggests. The LSTM incorporates extraordinary devices called reminiscence blocks into the recurrent hidden layer. Reminiscence blocks include reminiscence cells with auto-connections storing the

community's temporal nation in addition to specific multiplicative tools called gates to manage movement with the flow of information. Each reminiscence block in the single structure contained an entrance door and an exit door. The entrance door controls the floating of the entrance activations in the reminiscence cell. The output door controls the output float of telephone activations in the relaxation of the network. Later, the overhanging door was once brought to the reminiscent block. The negligence door scales the inner nation of the mobile phone before including it as an entry into the phone via the cell's auto-recurring connection, therefore by adapting or resetting the cell's reminiscence [39].

V.3.2.LSTM Principle

LSTMs have internal mechanisms called gates, which are used to regulate the flow of information by learning what information is useful to keep or forget in time. In doing so, the network will be able to transmit relevant data, thanks to the state of the cell, acting as a transport route that transfers related information throughout the sequence chain. As the status continues, information is added or deleted through three main doors. Initially, the input is the output predicted previously (h_{t-1}) as well as the current input X_t). This data will be broken down into three streams. The objective is to update the status of the cell from (C_{t-1} to C_t.) [40].

A- Input Gate

The Input Gate updates the status of the cell. The activation function is Sigmoid. For each input, the gate will provide an output value between 0 and 1, and decide which value to update (0 means not necessary and 1 means important). This result is then multiplied by the current state

$$f_t = (W_t\,[h_{t-1}, X_t] + b_t) \quad \text{(II.19)}$$

$$Ct = ft \times Ct{-}1 \quad \text{(II.20)}$$

With:

- h_{t-1}: The exit at the moment t-1
- X_t : Current input at time t
- b_t : Biases

- W_t : Weight
- σ: Sigmoid function

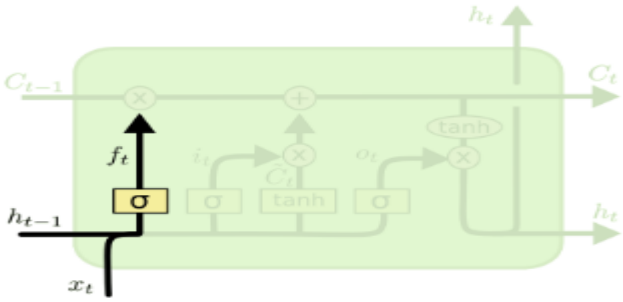

Fig.II. 19. Diagram represents an LSTM Input Gate.

B- Forget Gate

The Forget Gate forgets, decides what information should be discarded or kept. The information of the previous state is passed to the Tanh function giving values between -1 and 1. Then, to select the important characteristics, a sigmoid layer will decide which value will be updated

$$f_c = \tanh(W_c[h_{t-1}] + b_c) \quad \text{(II.21)}$$

$$I_t = (W_i[h_{t-1}, X_t] + b_i) \quad \text{(II.22)}$$

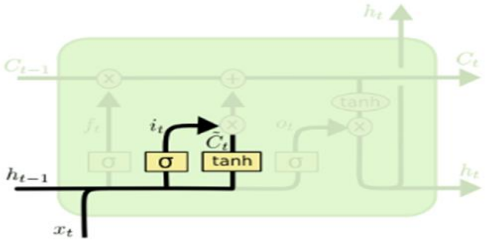

Fig.II. 20. Diagram represents a LSTM Forget Gate.

With:

- tanh : Hyperbolic Tangent activation function
- C : A candidate value

These two results will be multiplied and added to the state. At this stage, the state of the cell is:

$$C_t = f_t \times C_{t-1} + f_c \times I_t \quad \text{(II.23)}$$

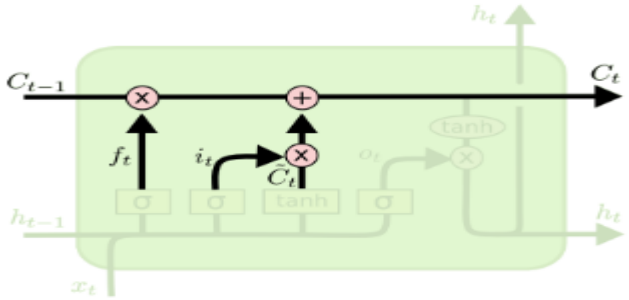

Fig.II. 21. Diagram represents an LSTM Forget Gate.

C- Output Gate

The output Gate will pass the previous hidden state and the current entry in a sigmoid function

$$o_t = (W_o[h_{t-1}, X_t] + b_o) \quad \text{(II.24)}$$

The new hidden state is, therefore:

$$h_t = o_t \times \tanh(C_t) \quad \text{(II.25)}$$

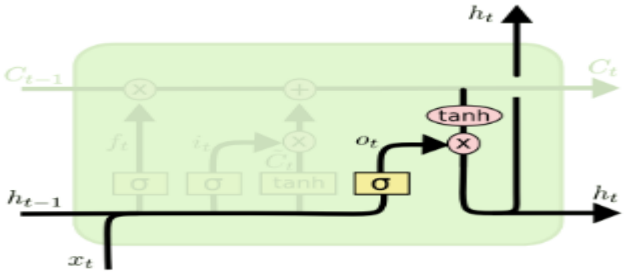

Fig.II. 22. Diagram represents an LSTM Output Gate.

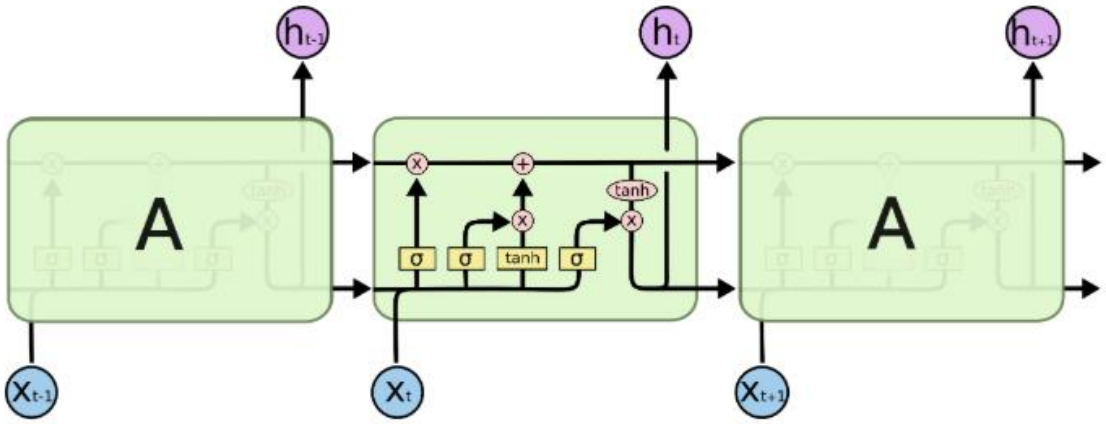

Fig.II. 23. Diagram represents a chain of three LSTM cells.

V.3.3.LSTM Algorithm

Algorithm: Operating algorithm of the doors according to three algorithms "if-then otherwise."

If Value Input " 0, then
Input Gate: Transmission of external information inside the block
If Value is Forgotten " 0, then
The previous state of the CEC affects the calculation of the present state
If Value Output Gate " 0, then
Input Gate: Transmission of Inside Information to the Outside of the Block
otherwise
Output Gate: No information provided on the exit
End if
otherwise
the cell forgets its past
end if
otherwise
Output Gate: Information blocked
End if

Table. II. 3.LSTM Algorithm [41].

V.3.4.Advantages of LSTM

- The constant error backpropagation within memory cells results in LSTM's ability to bridge very long time lags in case of problems similar to those discussed above
- For long time lag problems such as those discussed in this paper, LSTM can handle noise, distributed representations, and continuous values. In contrast to nite state automata or hidden Markov models, LSTM does not require an a priori choice of a nite number of states. In principle, it can deal with unlimited state numbers.
- For troubles mentioned in this paper, LSTM generalizes nicely even if the positions of broadly separated, applicable inputs in the enter sequence do now not matter. Unlike preceding approaches, ours shortly learns to distinguish between two or greater extensively isolated occurrences of a specific component in an input sequence, besides relying on excellent quick time lag training exemplars. There appears to be no need for parameter no tuning. LSTM works well over a broad range of parameters such as learning rate, input gate bias, and output gate bias. For instance, to some readers, the learning rates used in our experiments may seem extensive. However, a large learning rate pushes the output gates towards zero, thus automatically countermanding its adverse effects [42].

V.4. Bidirectional Long Short-Term Memory (BLSTM)

The thought of Bi-directional Long Short-Term Memory (BLSTM) originates from bidirectional RNN that approaches sequence facts in each frontward and backward instructions the use of two excellent hidden layers. BLSTMs be a part of these hidden layers with the equal output layer. One inadequacy of traditional RNNs is that they are only successful in the use of the preceding context. BRNNs [42] repair this by way of doling out records in both directions. A BLSTM network computes the ahead hidden layer sequence output, the output sequence of the backward hidden layer, and the output layer y by reiterating the ahead layer beginning t =1 to T, backward hidden layer because t = T to 1, and then the final output is upgraded using the following equations:

$$\vec{h_t} = H(W_{x\vec{h}}\, X_t + W_{\vec{h}\vec{h}}\, \vec{h_{t+1}} + b_{\vec{h}}) \quad \text{(II.26)}$$

$$h_t^{\leftarrow} = H(W_{xh^{\leftarrow}} X_t + W_{h^{\leftarrow}h^{\leftarrow}} h_{t+1}^{\leftarrow} + b_{h^{\leftarrow}}) \quad \text{(II.27)}$$

$$y_t = W_{h_y^{\rightarrow}} h_t^{\rightarrow} + W_{h_y^{\leftarrow}} h_t^{\leftarrow} + b_y \quad \text{(II.28)}$$

Both the output of the ahead and backward layers are calculated by way of potential of the preferred LSTM equations, Equations (II.27) - (II.28). The BLSTM layer produces an output vector, y_t , which is calculated via the equation:

$$y_t = \sigma(h_t^{\rightarrow}, h_t^{\leftarrow}) \quad \text{(II.29)}$$

Where σ function combines each the output sequences, the σ function could be of 4 kinds: concatenating, summation, common and multiplication function, and incorporating BRNNs with LSTM neurons outcomes a bidirectional LSTM recurrent neural network (BLSTM RNN). The BLSTM RNN is successful in gaining access to long term context statistics in each backward and ahead direction. The aggregate of both the forward and backward LSTM layers is regarded as a single BLSTM layer. It has been proven that the bidirectional fashions are appreciably higher than regular unidirectional models in some domains, like phoneme classification and speech recognition [42]. (**Figure II.24)** illustrates a bidirectional LSTM shape with three consecutive time steps.

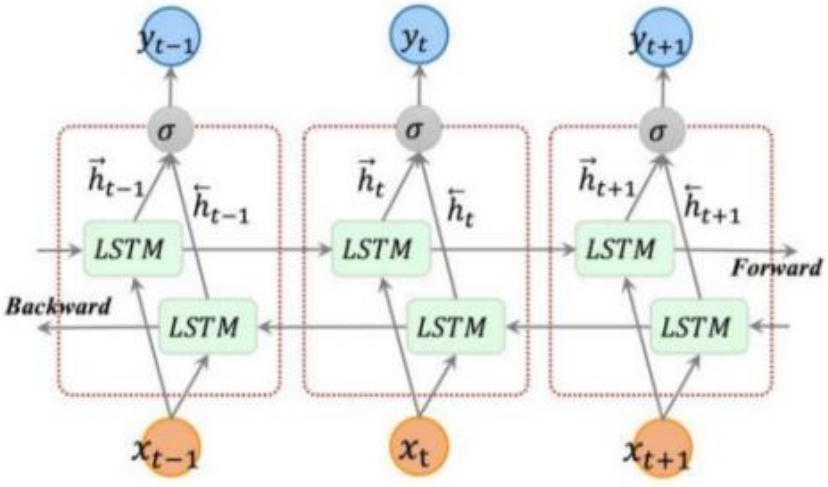

Fig.II. 24. Unfolded BLSTM RNN structure with three consecutive time steps[42].

VI.CONCLUSION

In this chapter we have presented the basic concepts of Machine learning and deep learning including its operating principle, its main components and also its limitations. Then, regarding machine learning, we talked about Random forest (RF) and Gradient boosting Machine (GBM), about these methods and their use, and we mentioned the pros and cons of both types.

Then we described a new variant of neural network called Deep Learning (DL). This technique is characterized by its ability to solve the problem of the complexity of training (NN) as well as its power to represent the forms (inputs) in a powerful, automatic and discriminating way.

Finally, we discussed the different models of Deep Learning namely the CNN, the LSTM, and BILSTM. The latter was very detailed because it will be the subject of several experiments in the next chapter.

CHAPTER III

Results and Discussion

I.INTRODUCTION

This last chapter is devoted to a simulation study. It is mainly aimed at assessment of the performance of the proposed techniques as a solution in the design of a solar radiation prediction system in the future. So it is to validate and evaluate the performance of each of the methods presented (RF-GBM - BILSTM-LSTM -DNN) in terms of learning time, and other regression parameters are then presented. The main efficiency requirements are formulated on the two essential points, namely, specification tests that verify that the program is performing the task for which it was designed, and performance tests that will be used to measure the effectiveness with which this task is completed. To carry out a comparative study.

II. Proposed System

All algorithms based on machine learning follow a predictive model that estimates a certain type of data with high accuracy. A large data set is essential for the learning algorithm to understand the behavior of the system. The first step for machine learning is data acquisition. The collected data were shared by various interested parties and summarized in useful information. The steps included in this process are data purification and data delimitation. The data were separated into two disjoint sets, training, testing . The training dataset was used for model training and testing. The dataset was used for model optimization and evaluation.

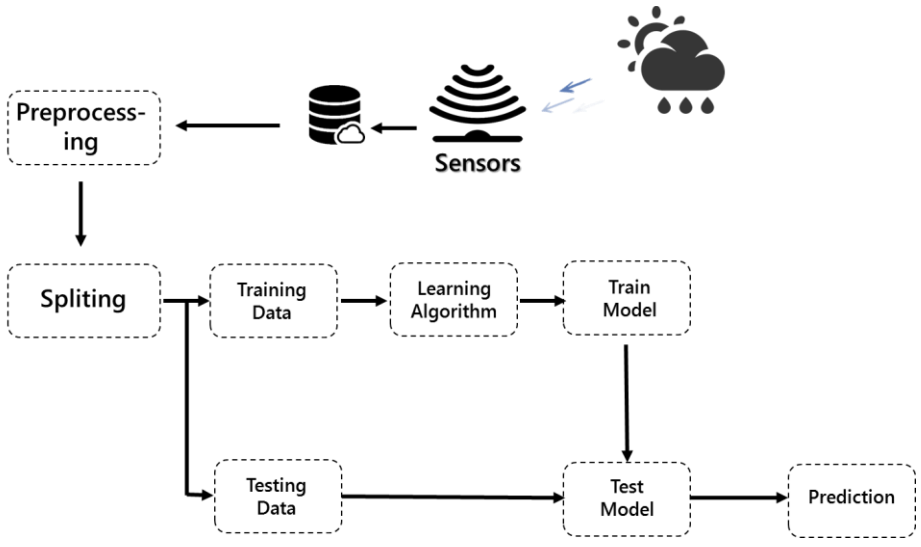

Fig. III. 1. Correlation matrix to identify the essential characteristics of the whole.

Training Dataset: The sample of data used to fit the model.

Testing Dataset: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

Dataset: A dataset consists of about two components, the two components are rows and columns. In addition, a main feature of a record is that it is organized in such a way that each row contains an observation.

Machine learning Algorithm : An "algorithm" in machine learning is a procedure that is run on data to create a machine learning "model".

Machine Learning Model : A "model" in machine learning is the output of a machine learning algorithm run on data.

III. DESCRIPTION OF THE DATASET

NASA HI-SEAS missions serve as a testbed and training ground for humans as we develop the capability to explore Mars. A recent NASA Space Apps Challenge hackathon asked participants to use the data collected on the HI-SEAS site to predict solar radiation given a set of measurable weather conditions. Knowing when the conditions are most favorable to solar radiation incident is crucial in deciding when and where to deploy solar energy recovery equipment, especially for settlers or astronauts on the surface of Mars.

These datasets are four-month weather data from the HI-SEAS weather station (September to December 2016). For each dataset, the fields are:

A-line number (1-n) is useful to sort the results of this export The UNIX date time_t (seconds since January 1, 1970). Useful for sorting the results of this export with other export results Date in yyyy-mm-dd format Local time in hh: mm: ss Format 24 hours Digital data, if applicable (maybe an empty string) Text data, if functional (can be an empty string).[44]

The units of each dataset are:

Solar radiation: watts per meter^2 (W/m2)	**Temperature:** degrees Fahrenheit (°F)
Humidity: percentage (%)	**Barometric pressure:** (Hg)
Wind direction: degrees (°)	**Wind speed:** miles per hour (mph)
Time at sunrise: Hawaii time	**Time at sunset:** Hawaii time

Table. III. 1. The units of each dataset.

II .1.Characteristics

At each timestamp of each day, there are values for all other variables. No other variables affect time or date values. Therefore, the date and time of day are independent variables.

For each date, there is a value for "Time at Sunrise" and "Time at Sunset." The difference in these values gives the length of a given day, which is directly related to the date. Further exploration of the dataset is required to determine whether the size of a given date outweighs the amount of useful information it provides.

Temperature, pressure, and humidity do not directly affect each other significantly, but since they are all properties that describe the local atmosphere, they do not vary independently of each other. Similarly, these three variables have a strong relationship with the time of day.

III.2.Correlation Matrix

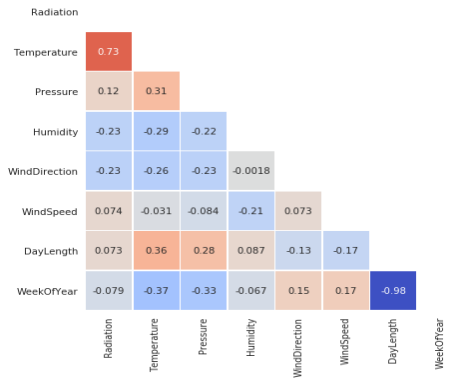

Fig. III. 2. Correlation matrix to identify the essential characteristics of the whole.

First, a fundamental correlation matrix is generated to remove irrelevant data and identify the essential characteristics of the set.

III.3.Bar Charts

Then, to better understand the data, hourly, and monthly averages of several variables were viewed using bar charts.

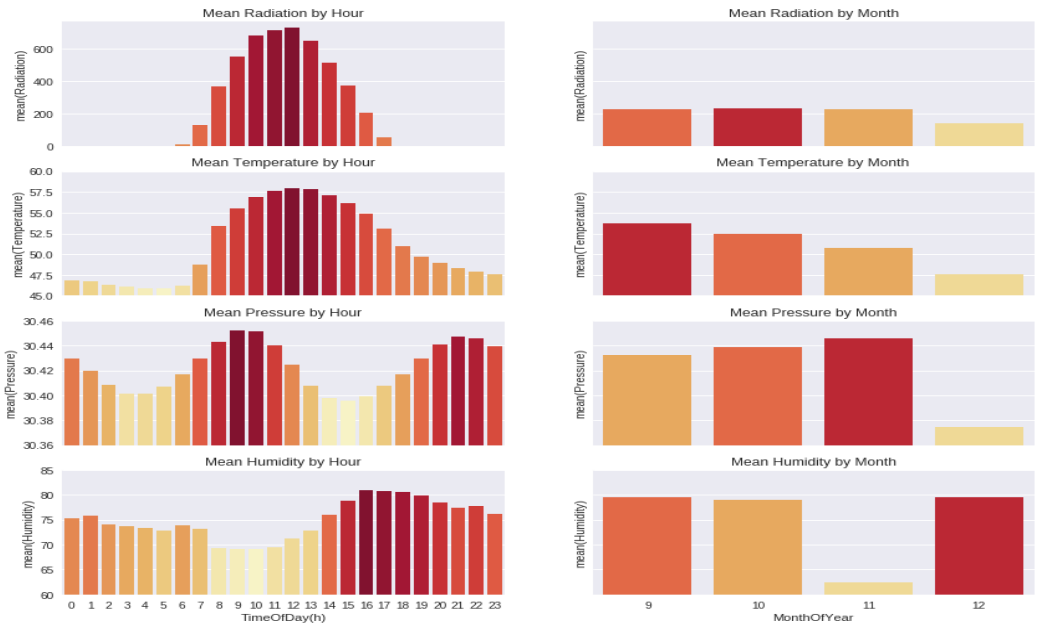

Fig. III. 3. Bar charts to better understand the data.

IV. Preprocessing Dataset

Data preprocessing is a data mining technique in which raw data is converted into an understandable format. The real data is often incomplete: missing attribute values, missing specific attributes of interest, data preprocessing is a proven method of solving such problems[45].

IV.1. Steps involved in data preprocessing

- ✓ Importing the required Libraries
- ✓ Importing the data set
- ✓ Handling the Missing Data.
- ✓ Encoding Categorical Data.

- ✓ Splitting the data set into test set and training set.
- ✓ Feature Scaling.

IV.2. Data Standardization

The standardization of data helps to overcome the differences in the "standards" of variables. Indeed, variables with large values may have a more significant Influence than variables with small values, but may not be more significant. Normalization is a data pre-processing tool used in the data mining system. Such as 0 to 1. Standardization is particularly useful for classification and regression algorithms. To optimize the prediction models, we introduce the normalization of the data by the min-max method. This technique provides a linear transformation over the original data range. Rescheduling is often accomplished using a linear interpretation formula, such as:

$$x_{i_n} = \frac{x_i - min_{(x)}}{max_{(x)} - min_{(x)}} \quad \text{(III.1)}$$

Avec :

- x_{i_n} : The normalized value.
- $X = [x_1, x_2, \dots xi]$: The vector to normalized.
- min(X): The minimum value of X.
- max(X): The maximum value of X.

V.PREDICTED EVALUTION

V.1.DEVELOPMENT ENVIRONMENT

V.1.1.Google Colab

Google Colab is a free cloud service that now supports free GPUs, improving coding skills in the Python programming language. Develop in-depth learning applications using popular libraries such as Keras, TensorFlow, PyTorch. The most important feature that distinguishes Colab from other free cloud computing services is: Colab provides a GPU and is entirely free [46].

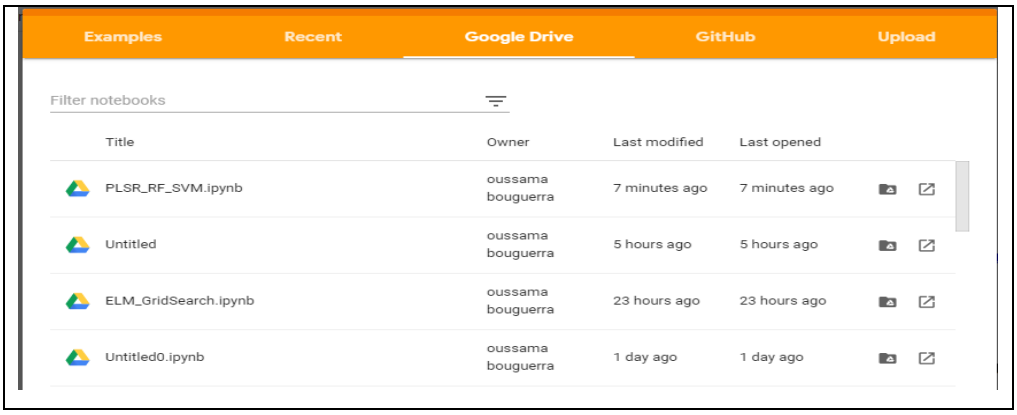

Fig. III. 4. Google Colab Environment.

V.1.2.Python

Python is a high-level programming language used for general programming. Created by Guido van Rossum and released in 1991, Python has a design philosophy that emphasizes the readability of the code, including using virtual spaces. It provides constructions that allow explicit programming at a small and large scale. Python has a dynamic gadget and automated reminiscence management. It helps more than one programming paradigms, inclusive of object-oriented, imperative, functional, and procedural, and has a complete general library. Python interpreters are reachable for many running structures [47].

Fig. III. 5. The Python Logo

V.1.3.Jupyter Notebook

The Jupyter Notebook is an open-source net utility that lets you create and share files that comprise stay code, equations, visualizations, and narrative textual content. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more [48].

Fig. III. 6. The Jupyter Notebook logo.

V.1.4. Visual Studio Code

Is a source code editor developed by Microsoft for Windows, Linux and macOS. It includes built-in Git and support for debugging, syntax highlighting, smart code completion, snippets, and code refactoring. It is exceptionally customizable, permitting customers to exchange the theme, keyboard shortcuts, preferences, and set up extensions that add extra features. The supply code is free and open-source, launched beneath the permissive MIT license. Compiled binaries are freeware for any purpose [49].

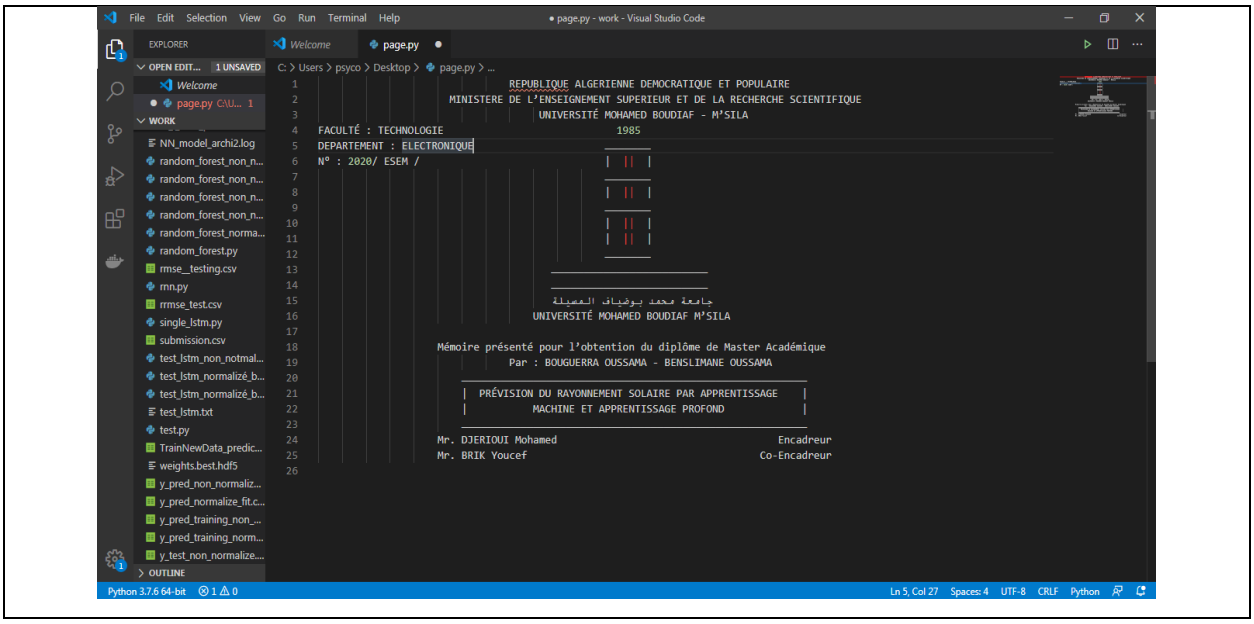

Fig. III. 7. Visual Studio Code Software Interface.

V.1.5.Tools Presentation

V.1.5.1.The Software

- ✓ **Tensorflow**

TensorFlow was created by the Google Brain team to research ML and Deep Learning. It is considered a modern version of Theano.

• Advantages

- ✓ supported by Google
- ✓ A very large community
- ✓ Multi-GPU support

• Disadvantages

- ✓ Slower than other frameworks in many benchmarks, although Tensorflow is catching up.
- ✓ Theano still outperforms RNN support

- ✓ **Keras**

The highest level, the most user-friendly framework on the list. It allows users to choose whether the models they build are running on Theano or TensorFlow.

• Advantages

- ✓ Python
- ✓ The perfect backend for Theano or TensorFlow
- ✓ High-level, intuitive interface

• Disadvantages

- ✓ Less flexible than other APIs

V.1.5.2.Hardware

Deep Learning and machine learning is an area with intense computational requirements, and the availability of resources (especially in GPU) dedicated to this task will fundamentally influence the user experience because, without its resources, it will take too long to learn from one's mistakes what can be discouraging. The experiments were all carried

out on a machine that offers acceptable performance, the characteristics of which are as follows:

CPU	**Intel Core i5-5200U @ 2.20GHz**
GPU	**NVIDIA GeForce 820M GPU**
RAM	**8GB**

Table. III. 2.machine characteristics.

V.2. EVALUATION CRITERIA

IV.1.Estimation Of Pre section Error

To make the analysis as rigorous as possible, the tests must be performed under the same conditions whenever possible. All experiments were taken on the same computer in order not to compromise the performance analysis. Each test was repeated four times. This is because some results were not equal with each run of a simulation. In both methods, the simulation time presents small related variations to obtain the performance error of each prediction and to analyze each parameter and better, and it was necessary to use some mathematical tools, pervasive in this case type of studies:

A - Mean Absolute Error (MAE)

The absolute mean error is a quantity often used to measure deviation from predictions and measurements. Its formula is given by:

$$\mathbf{MAE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\mathbf{j=1}}^{\mathbf{n}}|\mathbf{y} - \mathbf{y_i}| \qquad \text{(III.2)}$$

- n = The number of errors
- Σ = Symbol of summation
- y = Measured solar irradiation
- y_i= Solar irradiation estimated by a neural network.

B- Mean Square Error (MSE)

The mean square error is handy for comparing several estimators, especially when one of them is biased; it is given by :

$$\boldsymbol{MSE = \frac{1}{n}\sum_{j=1}^{n}(y - y_i)^2} \quad \text{(III.3)}$$

- n = The number of errors
- Σ = Symbol of summation
- y = Measured solar irradiation
- y_i= Solar irradiation estimated by a neural network.

C- Root Mean Square Error (RMSE)

The RMSE is a measure of the variation of the predicted values around the measured values. The smaller the value, the better the model, it is given by

$$\textbf{RMSE}=\sqrt{\boldsymbol{MSE}}=\sqrt{\frac{1}{n}\sum_{j=1}^{n}(y - y_i)^2} \quad \text{(III.4)}$$

- n = The number of errors
- Σ = Symbol of summation
- y = Measured solar irradiation
- y_i= Solar irradiation estimated by a neural network.

D- Relative Root Mean Square Error (RRMSE)

The percentage of RRMSE is obtained by dividing RMSE by the measured mean values, which are defined by:

$$\textbf{RRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{j=1}^{n}(y-y_i)^2}}{\frac{1}{n}\sum_{j=1}^{n} y} \times \mathbf{100} \quad \text{(III.5)}$$

Different ranges of RRMSE can be defined to show the capability of the models, so that model accuracy is:

- Excellent for RRMSE 10%;
- Good for 10% RRMSE 20%;
- Fair for 20% RRMSE 30%;
- Low for RRMSE > 30%.

E- Correlation Coefficient (R)

The correlation coefficient measures how close the predicted values are to the actual values. Clearly, the value of the correlation coefficient more comparable to the unit implies a better prediction.

$$\mathbf{R}=\frac{cov(y_i,y)}{\sigma_{y_i}\sigma_y} \quad \text{(III.6)}$$

F- Coefficient Of Determination (R^2)

The coefficient of determination is a statistical measure, which indicates how the regression line adjusts the actual data. A value of R^2 close to 1 means that the regression line changes the data well.

This indicator varies between 0 and 1. A value of 1 indicates a perfect agreement between measure and model, while the value 0 indicates a total disagreement.

$$R^2 = 1 - \frac{\frac{1}{n}\sum_{j=1}^{n}(y-y_i)^2}{\frac{1}{n}\sum_{j=1}^{n} y} \quad \text{(III.7)}$$

V.3. SIMULATION RESULTS

V.3.1.Random Forest

- The hyperparameters of the random forest technique in 4 learning and testing time

Parameter / cases	Number estimators	Max depth	Max features	Min samples split	MinSamples Leaf	Test size
1^{er} CAS	**1640**	**None**	**Auto**	**2**	**1**	**0.20**
$2^{éme}$CAS	840	None	Auto	15	10	0.20
$3^{éme}$CAS	200	64	Sqrt	2	1	0.33
$4^{éme}$CAS	4000	None	0.5	2	1	0.33

Table. III. 3. The hyperparameters of the random forest technical dataset.

Random forest										
ERROR / CAS	LEARNING					TEST				
	MSE	MAE	R^2	RRMSE	T_{app}	MSE	MAE	R^2	RRMSE	T_{test}
1^{er}CAS	**13.99**	**1.99**	**0.999**	**1.603**	**336**	**268.90**	**9.64**	**0.995**	**12.23**	**6.12**
$2^{éme}$CAS	88.69	4.92	0.999	4.177	149	364.81	11.22	0.993	14.25	1.98
$3^{éme}$CAS	19.75	2.45	0.999	1.956	118	527.83	15.23	0.993	13.80	7.70
$4^{éme}$CAS	13.69	2.08	0.999	1.62	351	315.35	11.34	0.995	10.66	15.64

Table. III. 4. Results were obtained from the random forest technical basis.

Next, we compared the actual outputs with the RF regression outputs in the learning and testing base. We obtained the best results in the 1^{st} the case according to the evaluation criteria (R^2 =0.995, RRMSE=12.23, MAE = 9.64), Figure III.10-11 reports the correspondence between the two test outputs in the 1^{st} case

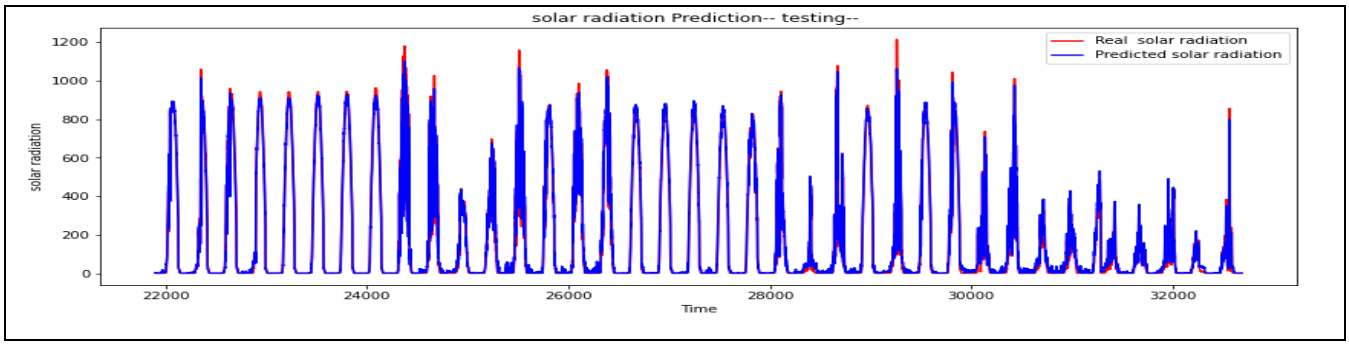

Fig. III. 8. Actual and calculated outputs for the random forest test.

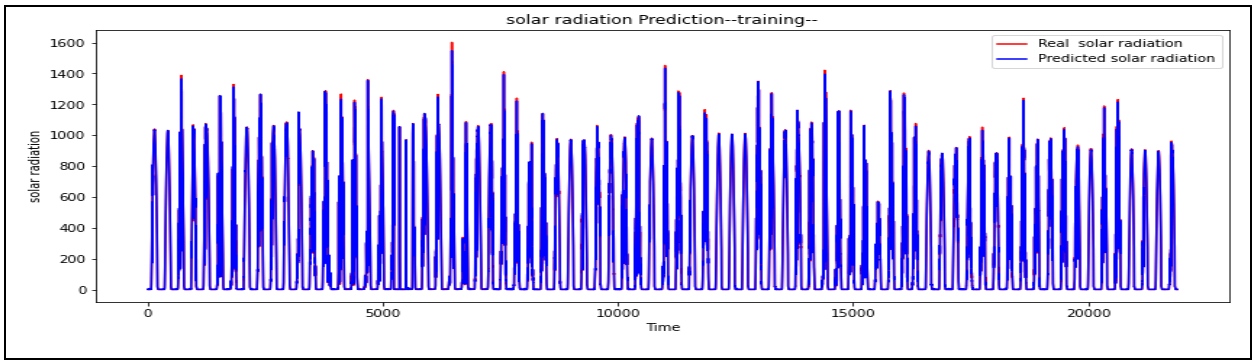

Fig. III. 9. Actual and calculated outputs for learning with random forest.

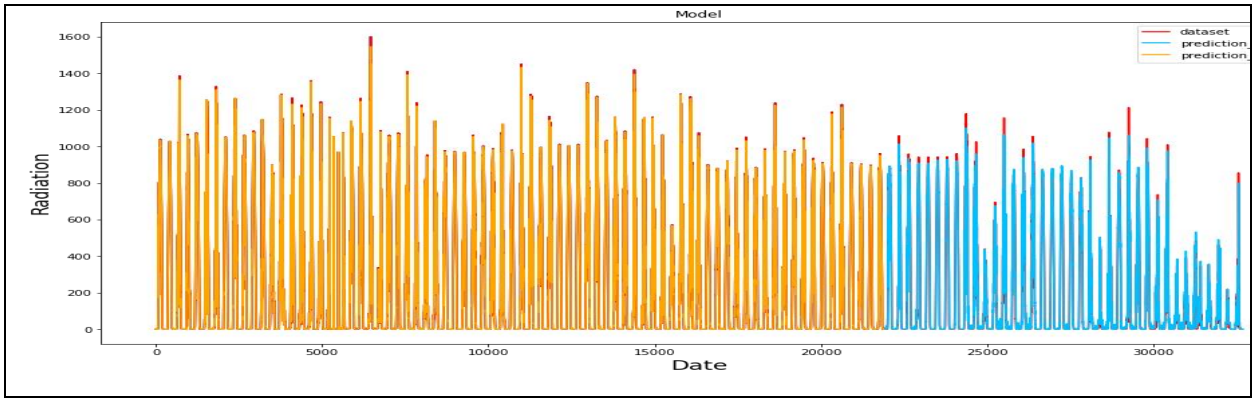

Fig. III. 10. Actual and calculated outputs for learning and testing with random forest.

V.3.2.Gradient Boosting Machine (GBM)

Gradient boosting regression hyperparameters in 4 learning and testing time

Parameter / cases	Number estimators	Learning rate	Max depth	Max features	Loss	Test size	Samples split	Samples leaf
1^{er} CAS	3000	0.01	None	sqrt	ls	0.20	2	1
$2^{éme}$CAS	2000	0.1	64	0.5	ls	0.20	15	10
$3^{éme}$CAS	**5000**	**0.1**	**128**	**0.5**	**huber**	**0.20**	**15**	**10**
$4^{éme}$CAS	4000	0.1	32	auto	huber	0.33	15	10

Table. III. 5. The hyperparameters of the technical basis Gradient boosting regression.

Technique: **Gradient boosting regression**										
ERROR / Cases	LEARNING					TEST				
	MSE	MAE	R^2	RRMSE	T_{app}	MSE	MAE	R^2	RRMSE	T_{test}
1^{er}CAS	0.79	0.71	0.999	0.39	1.34	368.98	12.27	0.993	14.33	45.36
$2^{éme}$CAS	0.78	0.48	0.999	0.39	0.30	195.69	8.45	0.996	10.44	18.00
$3^{éme}$CAS	**12.09**	**0.97**	**0.999**	**1.54**	**0.40**	**183.41**	**8.04**	**0.996**	**10.10**	**49.19**
$4^{éme}$CAS	15.14	1.04	0.999	1.71	0.33	269.61	9.83	0.996	9.86	41.69

Table. III. 6. Results were obtained from the technical basis of the gradient boosting regression.

Then we compared the actual outputs with the GBR regression outputs in the learning and testing base. We obtained the best results in the 3^{rd} the case according to the evaluation criteria (R^2=0.996, RRMSE=10.10, MAE= 8.04) Figure III.13-14 reports the correspondence between the two test outputs in the 3^{rd} case

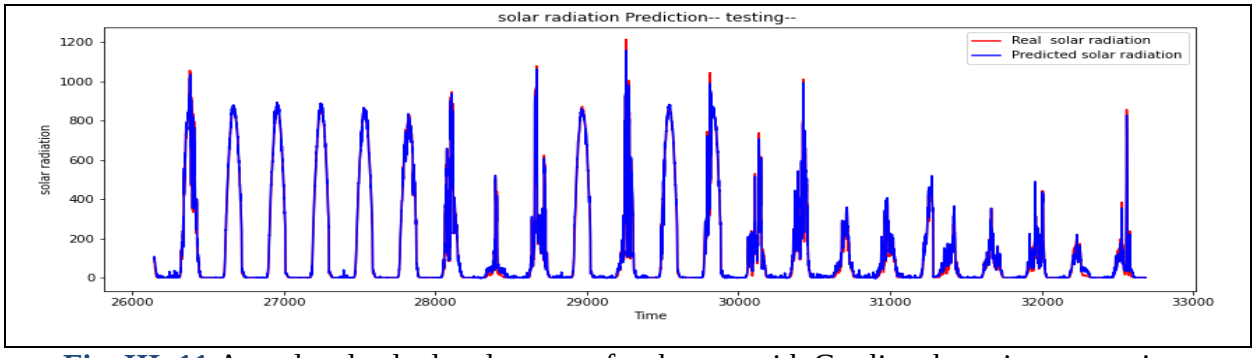

Fig. III. 11. Actual and calculated outputs for the test with Gradient boosting regression.

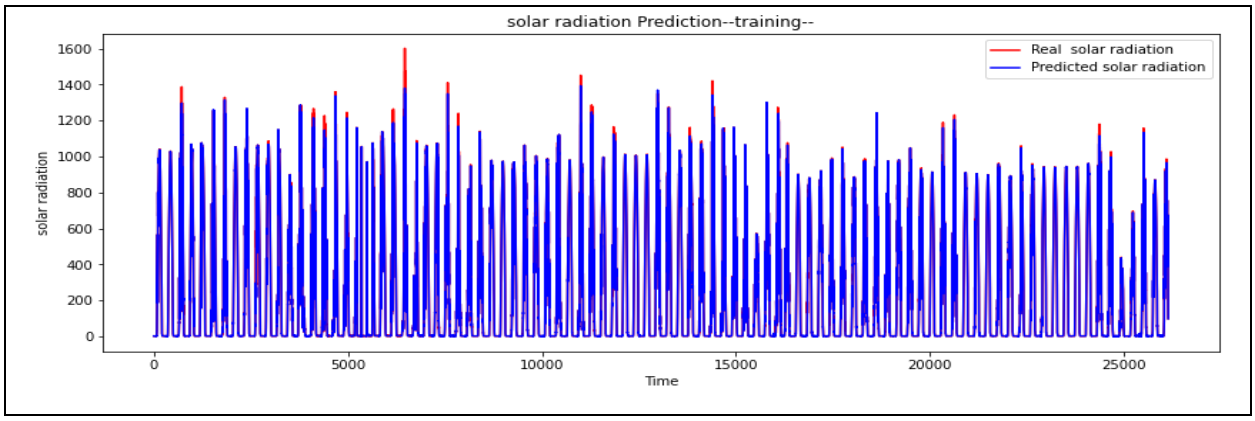

Fig. III. 12. Actual and calculated outputs for learning with Gradient boosting regression.

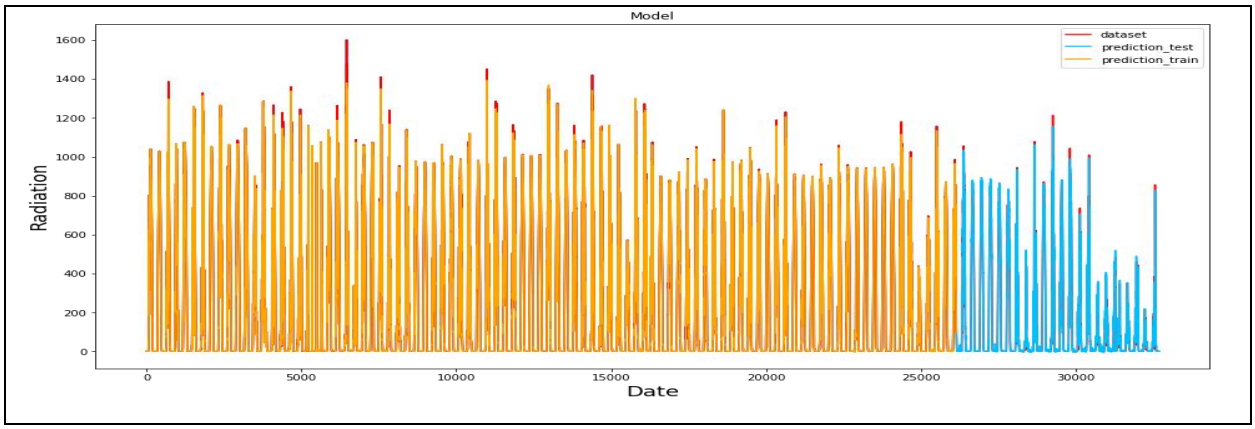

Fig. III. 13. Actual and calculated outputs for learning and testing with GBR.

V.3.3.Bidirectional LSTM (BI-LSTM)

The hyperparameters of the BI-LSTM technique in 4 learning time and test

parameter / cases	Number Epoch	Batch size	Number neurons	Batch size	optimizer	Validation	activation
1^{er} CAS	500	32	150	0.15	Nadam	True	linear
$2^{éme}$ CAS	**430**	**64**	**100**	**0.33**	**Nadam**	**True**	**linear**
$3^{éme}$ CAS	179	32	100	0.33	Nadam	True	linear
$4^{éme}$ CAS	235	128	200	0.33	Nadam	True	Relu

Table. III. 7. Hyperparameters of the BI-LSTM technical dataset.

BI-LSTM										
Error / CASES	LEARNING					TEST				
	MSE	MAE	R^2	RRMSE	T_{app}	MSE	MAE	R^2	RRMSE	T_{test}
1^{er}CAS	2.88	1.53	0.999	0.757	3h.45 m	1.53	0.98	0.999	1.13	5.03
$2^{éme}$CAS	**0.61**	**0.59**	**0.999**	**0.34**	**1h.24 m**	**0.85**	**0.74**	**0.999**	**0.69**	**3.13**
$3^{éme}$CAS	3.05	1.38	0.999	0.763	37m.6 s	2.39	1.17	0.999	0.92	3.05
$4^{éme}$CAS	8.89	1.93	0.999	1.31	1h.35 m	17.93	2.88	0.999	2.54	11.96

Table. III. 8. Results were obtained from the Bi-LSTM technical basis.

Then we compared the actual outputs with the Bi-Directional Long Short Term Memory regression outputs in the learning and testing base. We obtained the best results in the 2^{nd} the case according to the evaluation criteria (R^2=0.999, RRMSE=0.67, MAE = 1.24) Figure III.19-20 reports the correspondence between the two test outputs in the 2^{nd} case.

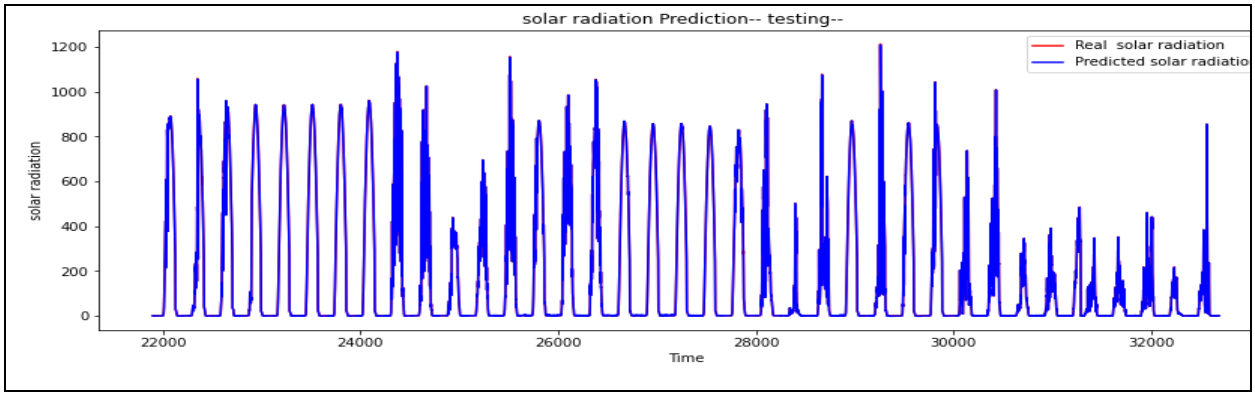

Fig. III. 14. Actual and calculated outputs for the test with BI-LSTM.

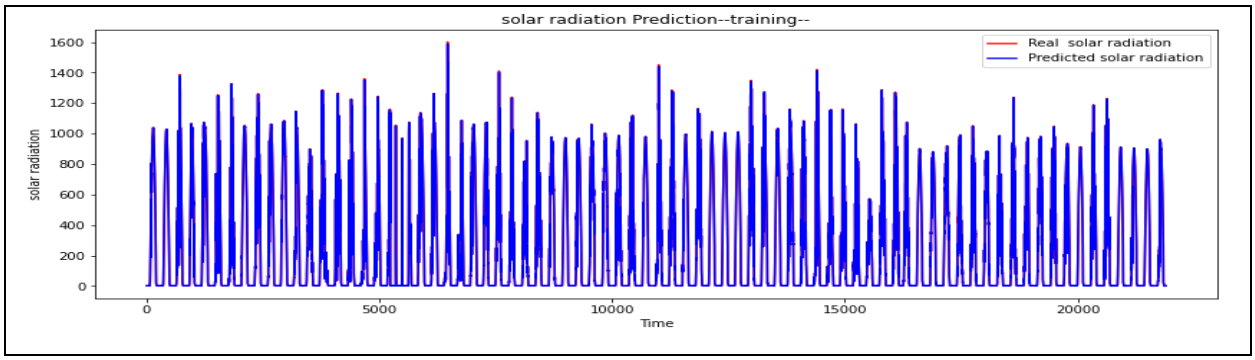

Fig. III. 15. Actual and calculated outputs for learning with BI-LSTM.

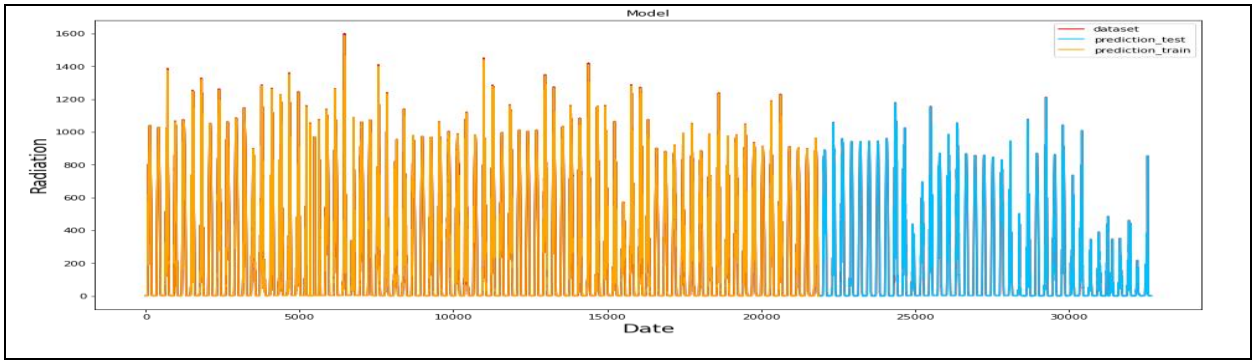

Fig. III. 16. Actual and calculated outputs for learning and testing with BI-LSTM.

V.3.4.Deep Neural Network (DNN)

Parameter / cases	Number Epoch	Batch size	Number neurons	test size	Optimizer	Learn rates	Activa-tion	Dropout rates
1er CAS	144	64	(64,128, 256,512,1)	0.2	Adam	0.005	Relu	0.30
2éme CAS	96	64	(64,128,256, 512,512,1024,1)	0.2	Adam	0.01	Relu	0.20
3éme CAS	318	64	(64,128,256, 512,1024,1)	0.2	NAdam	0.005	Relu	0.30
4éme CAS	**281**	**128**	**(64,128,256, 512,512,1)**	**0.2**	**NAdam**	**0.005**	**linear**	**0.30**

Table. III. 9. The hyperparameters of the Deep Neural Network technical dataset.

Technique: **Deep Neural Network**										
Error / CASES	LEARNING					TEST				
	MSE	MAE	R^2	RRMSE	T_{app}	MSE	MAE	R^2	RRMSE	T_{test}
1^{er}CAS	20.326	3.48	0.999	1.99	175.2	16.13	3.05	0.999	2.99	1.88
$2^{éme}$CAS	241.01	10.00	0.997	6.88	126.3	104.76	6.37	0.998	7.63	1.36
$3^{éme}$CAS	24.635	3.852	0.999	2.20	256.3	16.10	3.62	0.999	2.99	2.35
$4^{éme}$CAS	**8.00**	**2.24**	**0.999**	**1.25**	**235.3**	**6.01**	**1.99**	**0.999**	**1.83**	**2.23**

Table. III. 10. Results were obtained from the Deep Neural Network technical basis.

Then we compared the actual outputs with the Deep Neural Network regression outputs in the learning and testing base. We obtained the best results in the 4^{th} the case according to the evaluation criteria (R^2=0.999, RRMSE=2.21, MAE = 8.77) Figure III.22-23 reports the correspondence between the two test outputs in the 4^{th}case.

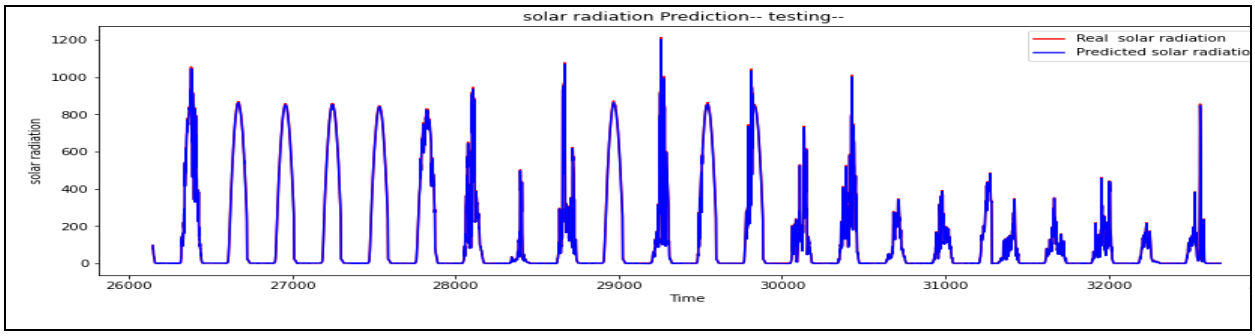

Fig. III. 17. Actual and calculated outputs for the test with Deep Neural Network.

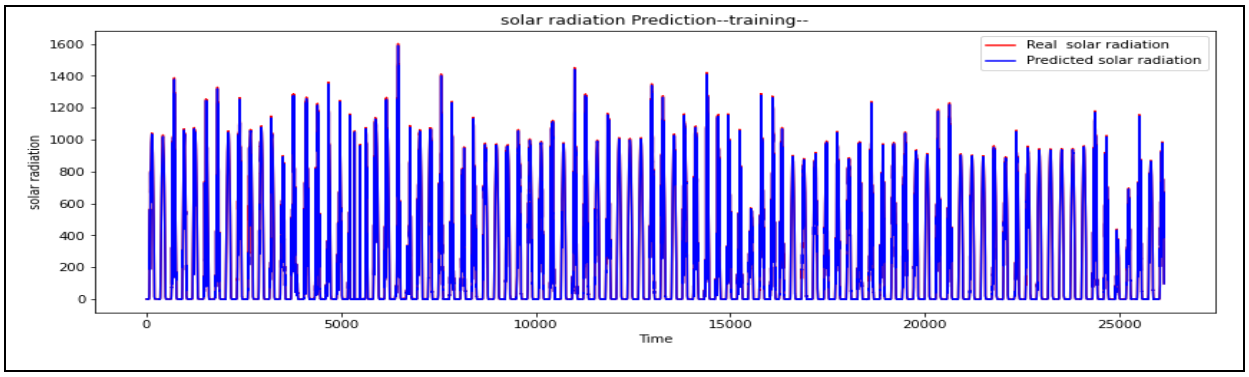

Fig. III. 18. Actual and calculated outputs for learning with Deep Neural Network.

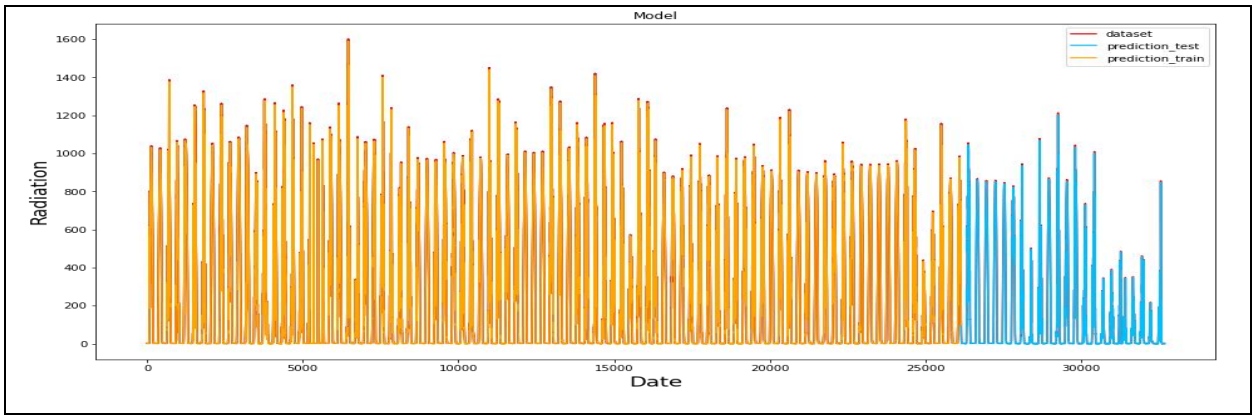

Fig. III. 19. Actual and calculated outputs for learning and testing with Deep Neural Network.

V.3.5.Long Short Term Memory (LSTM)

Parameter / cases	Number Epoch	Batch size	Number neurons	test size	Optim-izer	Valida-tion	Activa-tion	Dropout rates
1^{er} CAS	74	32	60.80.80.1	0.15	adam	True	Relu	0.2
$2^{éme}$CAS	171	64	60.80.80.1	0.20	adam	True	Relu	0.2
$3^{éme}$CAS	**300**	**64**	**60.80.80.1**	**0.20**	**Nadam**	**True**	**linear**	**0.2**
$4^{éme}$CAS	283	64	128.256.512.1	0.20	Nadam	True	linear	0.3

Table. III. 11. The hyperparameters of the Long Short Term Memory technical dataset.

Technique: **LSTM**										
Error / CASES	LEARNING					TEST				
	MSE	MAE	R^2	RRMSE	T_{app}	MSE	MAE	R^2	RRMSE	T_{test}
1^{er}CAS	187.47	7.94	0.998	6.07	49m.6 s	213.61	8.68	0.995	12.19	5.23
$2^{éme}$CAS	144.67	7.72	0.998	5.33	1h.49 m	165.51	7.95	0.997	12.86	7.98
$3^{éme}$CAS	**6.40**	**1.66**	**0.999**	**1.12**	**3h.22 m**	**9.11**	**2.17**	**0.999**	**2.25**	**7.82**
$4^{éme}$CAS	33.99	4.15	0.999	2.58	2h.02 m	37.65	4.42	0.999	4.57	5.78

Table. III. 12. Results obtained from the Long Short Term Memory technical basis.

Next, we compared the actual outputs with the Long Short Term Memory regression outputs in the learning and testing base. We obtained the best results in the 3^{rd} the case according to the evaluation criteria (R^2=0.999, RRMSE=2.65, MAE = 12.62) Figure III.25-26 reports the correspondence between the two test outputs in the 3^{rd} case

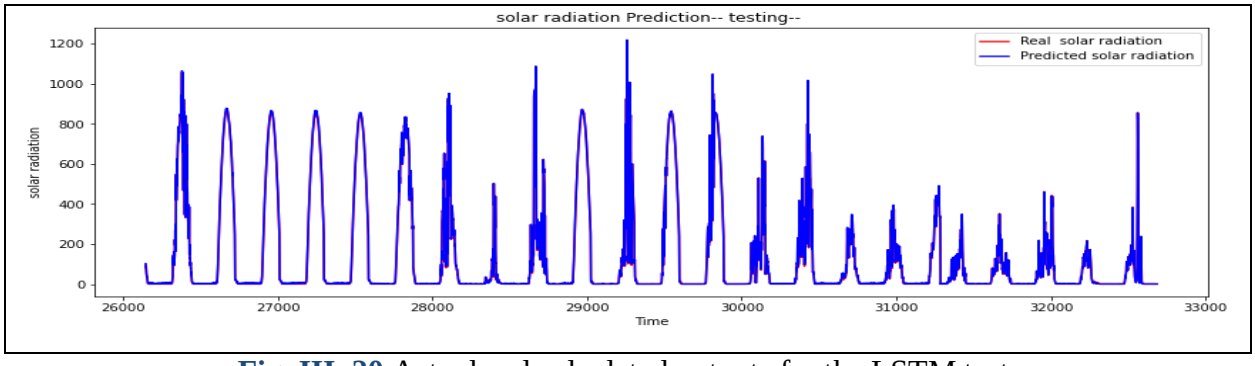

Fig. III. 20. Actual and calculated outputs for the LSTM test.

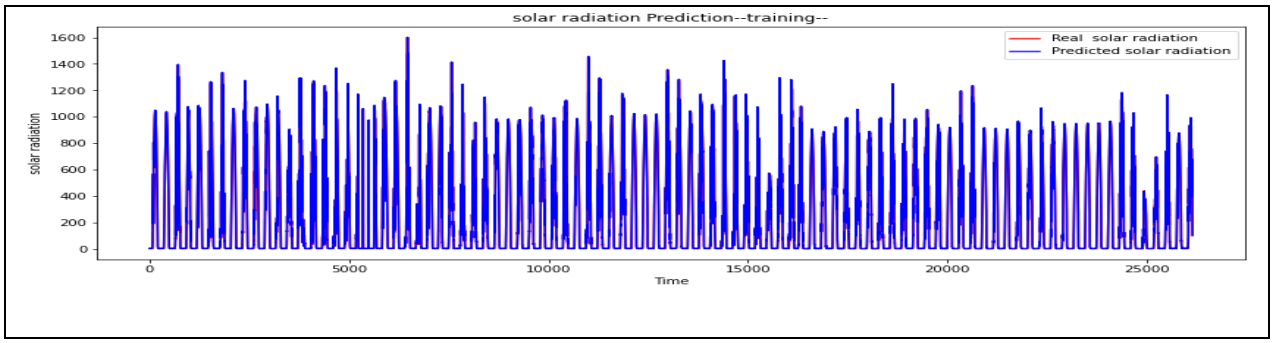

Fig. III. 21. Actual and calculated outputs for learning with LSTM.

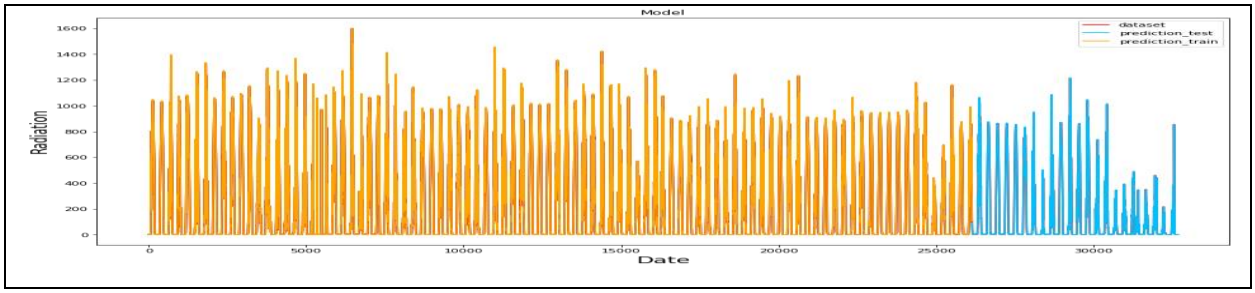

Fig. III. 22. Actual and calculated outputs for learning and testing with LSTM.

V.3.6.COMPARISON OF RESULTS

The table below makes a comparison of in terms of Training accuracy and Validation accuracy with the same NASA HI-SEAS dataset.

comparison of the results

Error / Tech	LEARNING					TEST				
	MSE	MAE	R^2	RRMSE	T_{app}	MSE	MAE	R^2	RRMSE	T_{test}
LSTM	6.40	1.66	0.999	1.12	3h.2m	9.11	2.17	0.999	2.25	7.82
DNN	8.00	2.24	0.999	1.25	235.3	6.01	2.24	0.999	1.25	2.23
BI.LSTM	**0.61**	**0.59**	**0.999**	**0.34**	**1h.2m**	**0.85**	**0.74**	**0.999**	**0.69**	**3.13**
GBR	12.09	0.97	0.999	1.54	0.40	183.41	8.04	0.996	10.10	49.19
RF	13.99	1.99	0.999	1.603	336	268.90	9.64	0.995	12.23	6.12

Table. III. 13. Performance evaluation for different techniques.

Based on the results obtained from Tables III.14, the following can be noted.

One of these results is that the BI-LSTM method One of the best ways used to predict solar radiation using artificial intelligence

VI.CONCLUSION

Throughout this work, it was possible to learn more about artificial intelligence techniques, in particular about the ML and DL models and how these models could be applied to solar irradiance Prediction.

After this, two different algorithms were developed using the studied LSTM and DNN methods to perform solar irradiance predictions. Before the final tests, the models were adjusted to perform the best predictions.

Conclusion

Conclusion

As we have seen in the introduction to this thesis , the Solar system sizing requires reliable, comprehensive and long-term solar radiation data. To overcome the lack of long series of reliable solar data, necessary for the optimization and optimal sizing of solar systems, an approach to predict hourly global solar radiation from cheaper meteorological data was presented in this manuscript. Ten different associations of five meteorological variables were used for develop two types of ANN models. The model that gives the best performance is a neuronal autoregressive with external BI-LSTM inputs with eight inputs. It was used to predict hourly solar radiation from more available and cheaper weather data (Solar radiation, Humidity, Wind direction, Time at sunrise, Temperature ,Barometric pressure ,Wind speed ,Time at sunset). The prediction accuracy of the proposed model is about 0.64% with a R^2 value of 0.999. In addition, different learning samples were used, the results showed that the proposed model required concrete training beforehand in order to give an acceptable predictive accuracy.

Future research could include a model that re-evaluates the times for each region and updates the input masks daily; such a model could improve the accuracy of the morning and dusk time region, as changes in season and obstacles affect dawn and dusk.

In conclusion, our approach provides radiation series synthetic solar to be used in the optimal sizing and planning of solar energy systems.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] *Renewables 2018. Power*. Oct. 8, 2018. URL: https://www.iea.org/reports/renewables-2018 (visited on 08/08/2019).

[2] Xiangyang Yea, Qing T. Zenga,b , Julio C. Facellia , Diana I. Brixnerc , Mike Conwaya , Bruce E. Bray Predicting Optimal Hypertension Treatment Pathways Using Recurrent Neural Networks. International Journal of Medical Informatics 2020.

[3] Chunchun Chena , Pu Zhang b , Yuan Liuc , Jun Liud, Financial quantitative investment using convolutional neural network and deep learning technology, Neurocomputing ,2019

[4] Mohammad Mehedi Hassan , Abdu Gumaei , Ahmed Alsanad , Majed Alrubaian , Giancarlo Fortino , A Hybrid Deep Learning Model for Efficient Intrusion Detection in Big Data Environment , Information Sciences,2019

[5] A. Khosravi* , R.N.N. Koury, L. Machado, J.J.G. Pabon , Prediction of hourly solar radiation in Abu Musa Island using machine learning algorithms , Journal of Cleaner Production ,2018

[6] Junho Lee, Wu Wang, Fouzi Harrou∗ , Ying Sun , Reliable solar irradiance prediction using ensemble learning-based models: A comparative study, Energy Conversion and Management, 2020

[7] Jianwu Zeng, Wei Qiao , Short-term solar power prediction using a support vector machine , Renewable Energy,2013

[8] Nor Azuana Ramli, Mohd Fairuz Abdul Hamid, Nurul Hanis Azhan, and Muhammad Alif As-Siddiq Ishak , Solar power generation prediction by using k-nearest neighbor method , AIP Conference Proceedings 2019

[9] Chao-Rong Chen and Unit Three Kartini , k-Nearest Neighbor Neural Network Models for Very Short-Term Global Solar Irradiance Forecasting Based on Meteorological Data , 2017

[10] J Liu1 , M Y Cao2 , D Bai1 and R Zhang, Solar radiation prediction based on random forest of feature-extraction ,2019

[11] JürgenSchmidhuber, Deep learning in neural networks: An overview, Neural Network ,2015

[12] S. Ghimire, R. C. Deo, N. Raj, J. Mi. Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms. Appl. Energy 2019;253:113541.

[13] Z. Pang, F. Niu, Z. O'Neill. Solar radiation prediction using recurrent neural network and arti_cial neural network: A case study with comparisons. Renew. Energy 2020;156:279-289.

[14] D. Guijo-Rubio et al. Evolutionary artificial neural networks for accurate solar radiation prediction. Energy 2020

[15] Qing X, Niu Y. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. Energy 2018;148:461–8.

[16] Kaba K, Sarيgül M, Avcي M, Kandيrmaz HM. Estimation of daily global solar radiation using deep learning model. Energy 2018;162:126–35.

[17] Kira M. Sargent « Supporting Renewable Energy: Lessons from the Deer Island Treatment Plant » Doctoral thesis, Washington University in St. Louis Environmental Studies Program, Spring 2010 St. Louis, Missouri

[18] Alberto E. Gabás Royo « SOLAR IRRADIANCE FORECASTING USING NEURAL NETWORKS », Faculty of Information Technology and Communication Sciences (ITC) Master of Science Thesis September 2019

[19] M.A. Atwater, J.T. Ball, “A numerical solar radiation model based on standard meteorological observations”, Sol. Energy 21, pp. 163–170, 1978

[20] TRAHI Fatiha « Prédiction de l’irradiation solaire globale pour la région de Tizi-Ouzou par les réseaux de neurones artificiels. Application pour le dimensionnement d’une installation photovoltaïque pour l’alimentation du laboratoire de recherche LAMPA. » MEMOIRE DE MAGISTER EN ELECTRONIQUE , Université Mouloud Mammeri de Tizi-Ouzou

[21] Lila Croci « Gestion de l'énergie dans un système multi-sources photovoltaïque et éolien avec stockage hybride batteries/supercondensateurs » Doctoral thesis, L'UNIVERSITE DE POITIERS, Submitted on 7 Feb 2014

[22] Saad Motahhir, Abdelaziz El Ghzizal, Aziz Derouich « Modélisation et commande d'un panneau photovoltaïque dans l'environnement PSIM » Doctoral thesis, Submitted on 19 Apr 2018

[23] Rene Y. Choi; Aaron S. Coyner; Jayashree Kalpathy-Cramer; Michael F. Chiang; J. Peter Campbell, Introduction to Machine Learning, Neural Networks, and Deep Learning, Translational Vision Science & Technology , February 2020

[24] Adrien BELLEVILLE « Prédiction des S/P individuels sur le produit flottes automobiles » Mémoire présenté le : pour l'obtention du Diplôme Universitaire d'actuariat de l'ISFA et l'admission à l'Institut des Actuaires

[25] Shalev-Shwartz, S. & Ben-David, S., "Understanding Machine Learning: From Theory to Algorithms", Cambridge University Press, 2014

[26] Morgane LAUR « Anticipation des changements de notes des obligations du portefeuille d'un assureur par méthode de machine learning » Mémoire présenté devant l'Université Paris Dauphine pour l'obtention du diplôme du Master Actuariat et l'admission à l'Institut des Actuaires

[27] Classification And Regression Trees for Machine Learning. Jason Brownlee. April 8, 2016 URL:https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/

[28] Alexey Natekin, Alois Knoll, Gradient boosting machines, a tutorial, fortiss GmbH, Munich, Germany, NEUROROBOTICS, Gradient boosting machines, a tutorial,2013

[29] Complete Machine Learning Guide to Parameter Tuning in Gradient Boosting (GBM) in Python, **AARSHAY JAIN,** FEBRUARY 21, 2016, URL : https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/

[30] Sepp Hochreiter Fakultat “LONG SHORT-TERM MEMORY” Technische Universitat Munchen 1997

[31] Mike Schuster and Kuldip K Paliwal, “Bidirectional recurrent neural networks,” Signal Processing, IEEE Transactions, vol. 45, no. 11, pp. 2673–2681, 1997

[32] Moualek Djaloul Youcef « Deep Learning pour la classification des images » Mémoire de fin d’études pour l’obtention du diplôme de Master en Informatique 2016

[33] Haohan Wang ,Bhiksha Raj “On the Origin of Deep Learning” Article · February 2017

[34] HAWKINS Jeff, BLAKESLEE Sandra. On intelligence : how a new understanding of the brain will lead to the creation of truly intelligent machines. 2007

---

[35] Meriem Bahi , Mohamed Batouche , Deep Learning for Ligand-Based Virtual Screening in Drug Discovery , University Constantine-2 Abdelhamid Mehri Constantine, Algeria ,2018

[36] Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall , Activation Functions: Comparison of Trends in Practice and Research for Deep Learning ,2018

[37] Adam — latest trends in deep learning optimization. Vitaly Bushaev. Oct,22,2018. URL : https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c

[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, pp. 1097–1105, 2012.

[39] “Understanding LSTM Networks -- colah’s blog.” [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed: 08-Jun-2018].

[40] Understanding LSTM Networks. Oinkina . Posted on August 27, 2015 . URL : https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[41] Huimei Han , Xingquan Zhu ,Ying Li , Generalizing Long Short-Term Memory Network for Deep Learning from Generic Data , 2020

[42] Z. Cui, S. Member, R. Ke, S. Member, and Y. Wang, “Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction,” pp. 1–12, 2018.

[43] Daniel Durstewitz , Georgia Koppe, Andreas Meyer, Lindenberg Deep neural networks in psychiatry,2019

[44] Solar Radiation Prediction .Task from NASA Hackathon . URL:https://www.kaggle.com/dronio/SolarEnergy

[45] Data Preprocessing in Machine Learning Model . Jack Daniel .URL : https://medium.com/analytics-vidhya/data-preprocessing-in-machine-learning-model-3af34d0f3ceb

[46] Google Colab Free GPU Tutorial .Fuat . Jan 26, 2018 .URL : https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

[47] Pythpn.URL: https://www.python.org./

[48] Jupyter .URL : https://jupyter.org/

[49] Visual Studio Code . URL : https://en.wikiversity.org/wiki/Visual_Studio_Code

Abstract

This thesis presents a prediction study of the different components of the solar radiation using artificial neural networks (ANN). The results of this study are crucial for the design and sizing of any solar energy system. A series of experimental hourly measurements of year variables were available for this study. Models (ANN) with different structures, in particular, different combinations of inputs as well as other numbers of hidden neurons, have been set up. To evaluate these models, the regression coefficient (R2) and the error estimators Relative Root Mean Square Error (RRMSE) and Mean Square Error (MSE) were used. Random Forest (RF) and Gradient Boosting Machine (GBM) and Deep Neural Network (DNN), Long Short Term Memory (LSTM) were compared with Bidirectional LSTM to generate horizontal hourly global solar radiation from less expensive exogenous variables. The results show BI-LSTM superiority with 8 entries. The test of this model to produce accurate forecasts offers good accuracy (R^2=0.999, MSE = 0.85 and RRMSE = 0.69%). Using different sizes of the learning sample we showed that from one year of data, our model gives satisfactory results. A comparison of our results with the literature confirmed that our models (ANN) exceed other estimation methods and that the proposed models ensure an authentic prediction of the different components of hourly solar irradiation from endogenous and exogenous variables that are more available and less expensive.

Keywords: ANN, solar power, forecasting, renewable energy, machine learning, deep learning.

Résumé

Cette mémoire présente une étude prédictive des différentes composantes de la rayonnement solaire utilisant des réseaux neuronaux artificiels (RNN). Les résultats de cette étude sont cruciaux pour la conception et le dimensionnement de tout système d'énergie solaire. Une série de mesures horaires expérimentales de variables de huit ans était disponible pour cette étude. Des modèles (RNN) avec différentes structures, en particulier, différentes combinaisons d'entrées ainsi que différents nombres de neurones cachés ont été mis en place. Pour évaluer ces modèles, on a utilisé le coefficient de régression (R2) et les estimateurs d'erreur Erreur quadratique moyenne relative (RRMSE) et Erreur quadratique moyenne (MSE). On a comparé Random Forest (RF) et Gradient Boosting Machine (GBM) et Deep neural Network (DNN), Long Short Term Memory (LSTM) avec Bidirectional LSTM pour générer un rayonnement solaire mondial horaire horizontal à partir de variables exogènes moins coûteuses. Les résultats montrent une supériorité BI-LSTM avec 8 entrées. Le test de ce modèle pour produire des prévisions authentiques montre une bonne précision (R^2=0,999, MSE = 0,85 et RRMSE = 0,69 %). En utilisant différentes tailles de l'échantillon d'apprentissage, nous avons montré qu'à partir d'une année de données, notre modèle donne des résultats satisfaisants. Une comparaison de nos résultats avec la littérature a confirmé que nos modèles (RNN) dépasser les autres méthodes

d’estimation et que les modèles proposés assurent une prédiction authentique des différentes composantes de l’irradiation solaire horaire à partir de variables endogènes et exogènes plus disponibles et moins coûteuses.

Mots-clés : ANN, énergie solaire, prévision, énergies renouvelables, apprentissage automatique, apprentissage profond.

ملخص

تقدم هذه الرسالة دراسة تنبؤ للمكونات المختلفة للإشعاع الشمسي باستخدام شبكات عصبية اصطناعية .(RNA) إن نتائج هذه الدراسة تشكل أهمية حاسمة في تصميم أي نظام للطاقة الشمسية وتحجيم حجمه .وقد توفرت لهذه الدراسة سلسلة من القياسات التجريبية بالساعة لمتغيرات ثماني سنوات .وتم إعداد نماذج (RNA) ذات هياكل مختلفة، وخاصة تركيبات مختلفة من المدخلات فضلاً عن أعداد مختلفة من الخلايا العصبية المخفية .لتقييم هذه النماذج، تم استخدام معامل الانحدار (R2) ومقدرات الأخطاء النسبة للخطأ المتوسط الحسابي للجذر المتوسط (RMSE) وخطأ مربع المتوسط .(MSE) تمت مقارنة الغابة العشوائية (RF) وماكينة تعزيز التدرج (GBM) والشبكة العصبية العميقة(DNN) ، الذاكرة طويلة الأجل (LSTM) مع تقنية LSTM ثنائية الاتجاه لتوليد إشعاع شمسي عالمي بالساعة أفقي من المتغيرات الخارجية الأقل تكلفة .وتظهر النتائج تفوق بي-إل تي سي مع 8 مداخل .إن اختبار هذا النموذج لإنتاج توقعات حقيقية يبين دقة جيدة(R^2=0.999) ، MSE = 0.85، RMSE = 0.69%).باستخدام أحجام مختلفة من عينة التعلم التي أظهرنا أنه من عام واحد من البيانات يقدم نموذجنا نتائج مرضية .وقد أكدت مقارنة النتائج التي تحققناها بالمؤلفات أن نماذجنا تتجاوز أساليب التقدير الأخرى وأن النماذج المقترحة تضمن التنبؤ الحقيقي بالمكونات المختلفة للإشعاع الشمسي بالساعة من المتغيرات الداخلية والخارجية الأكثر توافراً وأقل تكلفة .

الكلمات المفتاحية : شبكة العصبية اسطناعية , الطاقة الشمسية , الطاقات المتجددة , تعلم الآلة , تعلم العميق