POLITECNICO DI TORINO

Massachusetts Institute of Technology

Master's Degree in Computer Engineering

Master's Degree Thesis

"A scalable approach for predicting the energy produced by a moving solar panel in the urban environment: the City Scanner case study"

Supervisors

Prof. Enrico MACII

Prof. Carlo RATTI

Dr. Simone MORA

Candidate

Lorenzo SANTOLINI

July 2020

Abstract

Estimating the energy produced by solar panels has been a crucial research field in the last fifty years, driven by the impellent need of finding alternative solutions to fossil fuels. The increasing affordability and technological advancements in this area, along with the exponentially growing number of IoT devices, allowed to create systems disconnected from the electrical grid capable of self-sustaining relying solely on the sun's energy. This research presents a scalable approach for forecasting solar irradiation and yield of a moving solar panel, taking into account dynamic location changes and the influence of the city shading that comes along with these different environments. The presented machine learning model allows predicting the Global Horizontal Irradiance (GHI) with an R2 score higher than 0.90, and for each of its output, a Google Street View imaging method is employed to assess the obstruction caused by buildings and trees, to have a more precise measurement at street level. The validation of the selected approach is performed on data collected in The Bronx, NY, during deployment of City Scanner, drive-by sensing platform conceived by the MIT Senseable City Lab. For its scalable nature, this approach can be ported in other cities with minimal effort and applied to solar panel equipped fleets of the most diverse types, from drones to electric vehicles, to better assess in advance their energetic potential in an untested environment.

Acknowledgements

First of all, I would like to thank Professor Enrico Macii for having encouraged me in the decision to pursue my final project abroad and for the support he gave me even with an ocean between us.

I am very grateful also to Professor Carlo Ratti and Senseable City Lab for having hosted me during these months and for having fuelled my curiosity and my appetite for research.

I also cannot forget to thanks Simone Mora for having supervised me during my stay at MIT, guiding me through the whole process of a research project, from the data collection phase to the formulation of the results.

Without any doubt, I owe a large part of the credit to my parents, for having taught me that to achieve your goals, you have to face challenges with courage.

I am very grateful to my lifetime friends of PartyConLaP and Pappagallo for sharing with me the experiences I want to keep in my memories.

I can't forget to thank all my friends from the MIT Visiting Student Association and my lab members, because, since the first week, it didn't feel to be at 6000 Km away from home.

Finally, I would like to mention with affection Professor Paolo Cortese for his words capable of opening the eyes of naive teenagers on the adventure that awaits them in university and life.

Last but not least, a special thank goes to Mr. G. for being a good c.

Table of Contents

List of Tables

List of Figures

Acronyms

ACF
Auto-Correlation Function

AI
Artificial Intelligence

ANN
Artificial Neural Network

API
Application Programming Interface

ARIMA
Auto-Regressive Integrated Moving Average

ARMA
Auto-Regressive Moving Average

CNN
Convolutional Neural Network

CS
City Scanner

DHI
Diffuse Horizontal Irradiance

DNI
Direct Normal Irradiance

EHI

Extraterrestrial Horizontal Irradince

GBR

Gradient Boosting Regression

GHI

Global Horizontal Irradiance

GIS

Geographic Information System

GPS

Geo Positioning System

GPU

Graphical Processing Unit

GSV

Google Street View

HDOP

Horizontal Dilution of Position

IoT

Internet Of Things

LA

Los Angeles

MAPE

Mean Absolute Percentage Error

MIT

Massachusetts Institute of Technology

MLP

Multi-Layer Perceptron

MSE

Mean Squared Error

NARX

Nonlinear Autoregressive Network with Exogenous Input

NREL

National Renewable Energy Lab

NSRDB

National Solar Radiation Database

NWP

Numerical Weather Prediction

PACF

Partial Auto-Correlation Function

PSP

Pyramid Scene Parsing

RMSE

Root Mean Squared Error

RQ

Research Question

SARIMA

Seasonal Auto-Regressive Integrated Moving Average

SCL

Senseable City Lab

SOLPOS
: Solar Position And Intensity

SAA
: Solar Azimuth Angle

SVF
: Sky View Factor

SZA
: Solar Zenith Angle

TVF
: Tree View Factor

UHI
: Urban Heat Island effect

WWO
: World Weather Online

XGB
: Extreme Gradient Boosting

Chapter 1

Introduction

1.1 General Overview

Cities are big data factories; cameras, smart buildings, and diverse types of IoT devices generate an enormous amount of data points every single day. According to Cisco, within 2030 the number of IoT devices will reach 500 billion [1], empowering engineers and city planners with an incredible tool to digitalise the urban environment, from self-driving cars and smart buildings to traffic control systems.

Collected data, once processed with analytical solutions and visualised intuitively, can also be served directly to citizens to allow informed choices and made available to city governments and administrators for data-driven decision making, democratising in this way the acquired knowledge [2].

To make reliable predictions and to provide consistent services to citizens, few and sporadic data points are not sufficient. The number of data collecting nodes must be well distributed and provide continuous insights into the analysed phenomena. However, one of the main challenges faced using the traditional strategies is the generation of these large datasets, because a significant number of stationary sensors are needed, often leading to considerable costs for deployment and maintenance [3].

A partial solution to this problem comes with the advent of new approaches to urban sensing, which exploit the recently achieved portability and precision of last generation sensors. These class of devices, with high accuracy and embedded communication technologies, can broadcast data at a fraction of the cost, opening up new possibilities in the city context [4].

Various studies have applied these sensors to several use cases, targeting the analysis of one or multiple features of the urban environment, e.g. infrastructure conditions [5] or air quality [6], [7].

Monitoring the levels of volatile pollutants became of vital importance to the citizen's life. In fact, air pollution is the cause of millions of premature deaths every year [8], and analysing its daily levels, e.g. Particulate Matter and NOx gasses, can help urban planners and policymakers to adopt effective strategies to address this compelling issue.

Natural phenomena are generally continuous signals both in temporal and spatial dimensions; each sampling device must be able to capture dense enough spatiotemporal data points to obtain their digital representation. However, in many environmental applications, the collected data from stationary sensors are constrained on spatial, temporal or both dimensions, limiting the amount of information that each analysis can derive. From the study of the Sotiris Vardoulakis and al. and from one of Apte et al. [9, 10], emerged that air pollutants gradients sensed in fixed locations are subject to high variance between very close locations and in different time frames during the day.

On the other hand, satellite measurements on this type of data can give excellent spatial coverage, but they lack in precision concerning the time scope. The mathematical models employed in this research field are usually efficacious but very expensive in term of computational complexity, such in the case of the estimation of surface temperature [11].

Finally, most of the mentioned projects that employed portable sensors in the urban environment made use of special-purpose vehicles to achieve their data collection goal, deploying them specifically for the experiment purposes [6].

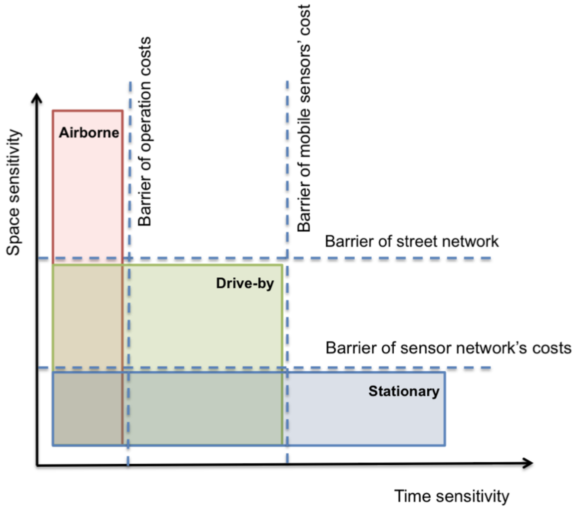

Space sensitivity
Time sensitivity
Airborne
Drive-by
Stationary
Barrier of operation costs
Barrier of mobile sensors' cost
Barrier of street network
Barrier of sensor network's costs

Figure 1.1: Space and time sensitivity of different approaches; drive-by solutions offer a trade-off between the two [3].

Drive-by sensing is a new approach to data collection that makes use of everyday vehicles as mobile sensors. Compared to traditional strategies, it grants various advantages, such as the creation of highly dense, hyper-local datasets at a fraction of the cost [3].

The City Scanner [2] project proposed by Senseable City Lab introduces this novel approach for developing a cost-effective, drive-by modular sensing platform to evaluate spatiotemporal environmental parameters in municipalities, such as air quality, ambient temperature and building thermal dissipation.

1.2 City Scanner

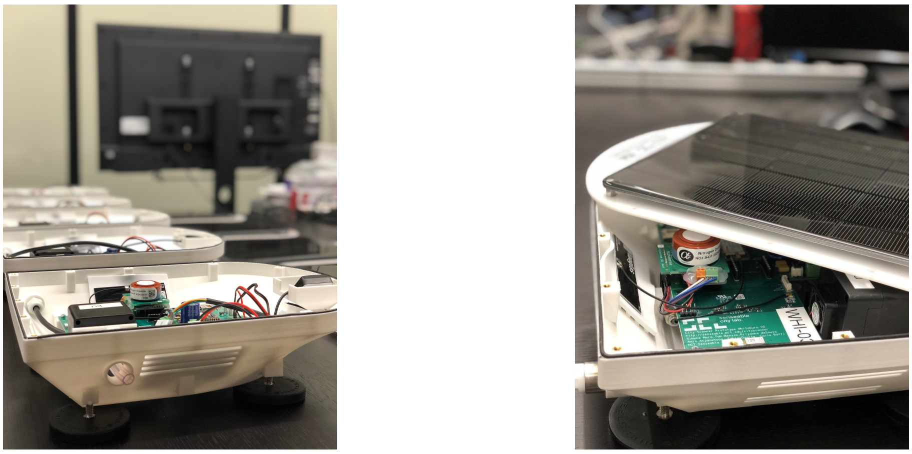
The City Scanner project, conceived by the Senseable City Lab, aims at developing a sensing platform to enable large scale drive-by deployments using everyday urban vehicles as sensing nodes.

City Scanner follows a centralised IoT regime to generate a near real-time map of sensed data. The individual sensing units are installed on urban vehicles to record data and stream it to the cloud for processing and analysis. Each sensing unit includes power management, data management, and cloud streaming components.

All these nodes also include air quality, temperature and humidity sensors. Apart from these core components, sensor nodes are designed in a modular way so they can be added or removed to build different sensing configurations. These devices are self-sufficient; they are equipped with a solar panel and provided with a battery to store the generated energy. Furthermore, since all components are encapsulated in the portable sensor platform, no additional resources (such as an external power source or an open window) are required other than some surface area on the external bodyshell. In the case of city-owned vehicles, this solution gives the local government the power to decide which and how many sensors to deploy to acquire the data they need for specific applications.

As stated above, the only source of energy of these devices is the solar panel installed on top of them. For this reason, this dissertation will assess the capabilities of these 'mobile' energy supply units to comprehend further the operational possibilities of these units before the deployment in an untested environment.

In the IV chapter, further details on the platform will be presented.



(a) **(b)**

Figure 1.2: City Scanner devices.

1.3 Problem Overview

IoT devices, thanks to their increasing portability, are used in locations where is not trivial to provide a constant energy supply from the electric grid. However, at the same time, these devices need a constant energy flow to fulfil their duties.

The convenience and the usability of solar energy harvesting systems have been improving over the last years; the efficiency increased while the manufacturing price decreased, facilitating their usage in a broader range of applications.

Even if for large scale implants bears some drawbacks, it offers a very convenient energy option in low-power sensors applications. Therefore it has been implemented in many projects, spanning in the most various fields, from agriculture [12] to medical devices [13], allowing the creation of solutions that use it as the primary energy source.

The drive-by sensor solution presented above, namely City Scanner, employs the sun's energy as the primary supply to perform sampling and to transfer the collected data to the cloud [14].

Nevertheless, given the constraints on the size of the panel and the variability of the weather, these devices may not continuously operate year-round, but work intermittently, collecting data points when possible. However, when the deployment of the project happens in a new, untested city, it is not trivial to understand the total available energy, directly correlated with the operational time of the devices, to achieve a data collection goal.

For this reason, this research will focus on the understanding of the amount of energy produced by a moving solar panel, to enable the fleets of tomorrow to achieve their goals, anticipating their operation time in an unexplored environment. This work will provide researchers and companies with an important instrument to understand better the needs of their prototypes and products, saving their time and leading them to better results in a fraction of the time.

1.4 Research Questions

The main research question for the thesis work is:

MRQ: Which is the estimate of the energy produced from a moving solar panel.

To answer the main research question the work has been broken down into two sub-questions:

RQ1: Which machine learning technique can forecast in the most accurate way how much solar irradiance hits a surface fully exposed to the sun in a specific geographical area at a given time?

RQ2: How much of this irradiance can be used by a moving solar panel considering the influence of the built environment?

1.5 Contributions

The goal of this thesis is to provide researchers with a tool that facilitates the deployment of solar-powered moving devices. The contribution will come on two aspects:

Theoretical. It will give an overview of the state of the art of solar panels, with a specific focus on the methods to predict the power output, helping in this way future researchers in their deployment.

Technological. This work will be provided open source as a Python3 module able to predict the power supply of a moving solar panel given a set of geographical coordinates.

Deploying a fleet of solar-powered devices will be better quantifiable in terms of costs and will drive to better decisions in the design phase.

1.6 Structure

The structure of this thesis is the following:

1. Chapter 2, **State of the Art**. In this section will be presented the modern methodologies to estimate the amount of solar irradiation (see appendix A) of an area, alongside the latest technologies to store and collect it, with a specific focus on the urban environment.

2. Chapter 3, **Design**. Hereby will be explained the design of the system implemented during my stay with a high-level description of the latter, together with an overview of the City Scanner project, the first use case where my model is applied.

3. Chapter 4, **Data Collection**. To validate in the most effective way the prototypes and the solar estimation model, empirical data are needed. This chapter depicts the preparations done for the deployment done in January 2020 in the city of New York, along with the expectations and the results

4. Chapter 5, **Implementation**. The technical implementation of the model, including the mathematical formulation of the problem and the chosen strategy to achieve the highest accuracy.

5. Chapter 6, **Evaluation**. After the implementation of the model, an important step in the research is the testing one. The dataset previously collected data in New York will be employed to quantify the accuracy of the predictions.

6. Chapter 7, **Conclusions and Future Works**. This will present the conclusions on the research, expected outcomes and actual results.

1.7 Senseable City Lab

The variety of miniaturised and connected digital technologies that spread into the urban environment, permeating our buildings, streets and communication devices is adding a new functional layer over cities. For this reason, emerges the need for collaboration across multiple disciplines to shape the urban future collectively.

SENSEable City Lab, located at Massachusetts Institute of Technology, stands on the front line of this change, partnering with cities and companies to address these challenges and opportunities [15].

With a multidisciplinary approach, it performs research focused on the interface between cities and technology. The key to the lab's success is the capability of bringing together people from a wide variety of sciences; thanks to this peculiarity, it investigates the impact of this revolution on the spaces in which peoples live.

This research work was carried out at the Senseable City Lab, in collaboration with Polytechnic of Turin.

Chapter 2

State of the art

In the first part of this chapter will be presented an overview of the historical development of photovoltaics and a brief outline of the technology used in the City Scanner project. Secondly, some modern methodologies to estimate its yearly power production will be presented.

2.1 Photovoltaics, historical overview

Photovoltaics is the process of converting sunlight directly into electricity using solar cells. Today it is an increasingly significant and renewable alternative to conventional fossil fuels, but in comparison to other electricity generation technologies, it is a relative newcomer.

The first discovery that put the basis of the photovoltaic energy collection method dates back to 1839, when Alexandre-Edmond Becquerel, a French physicist, discovered the photovoltaic effect while experimenting on solar light in his father's laboratory [16]. In his experiment, he placed silver chloride in an acidic solution and illuminated it while it was connected to platinum electrodes. As a result, it generated voltage and current.

The invention of the first solar cell instead, traces back to the American inventor Charles Fritt, which managed to produce current from solar light only half a century later [17]. He spread a wide layer of selenium onto a metal plate and covered it with a thin, semitransparent gold-leaf film and obtained an energy conversion efficiency between 1 and 2%.

Photovoltaic's first practical power harvesting devices were introduced in the 1950s, with Bell Laboratories intuition to switch from selenium to silicon [18], which allowed reaching the efficiency of 6%. They presented their invention in 1954 in a press conference where they impressed the media with the Bell Solar Battery powering a radio transmitter that was broadcasting voice and music.

Thanks to the fabrication of n-on-p silicon photovoltaic cells by the US Signal Corps Laboratories done in 1958, panels durability to space radiation improved, making them suited for space applications [19]. These space solar cells were thousands of times more expensive than today's ones, so their usage was limited to the missions outside of the atmosphere.

After the 1970s oil crisis, the world focus shifted on the research of alternative energy sources, which brought to the investigation of terrestrial applications of the photovoltaic. Even if both their price and their energy output was distant from the one of today, their advantages to the "remote" power supply area were quickly recognised and induced investments in the field. Small scale transportable applications, such as calculators, were utilised, and remote power applications began to benefit from photovoltaic.

In the 1980s, research into the field started to pay off. In fact, in 1985, the efficiency of silicon solar cells reached the 20% milestone [20]. Over the next year, the photovoltaic market steadily increased, with peaks such as in 1997, with an increase of 38%. Nowadays solar power is not only recognised for granting energy access to the areas disconnected from the grid, but also as a valid renewable alternative to polluting fossil fuels, diminishing in this way the impact of conventional electricity generation in advanced industrial countries.

2.2 Photovoltaic Effect

The photovoltaic effect is a process in which two dissimilar materials in close contact produce an electrical voltage when struck by light or other radiant energy. This phenomenon is the functioning principle of the photovoltaic cell, allowing to create a current from the potential difference created in this way.

The photovoltaic effect is a particular case of the photoelectric effect, discovered by Heinrich Hertz in 1937 but explained by the noble prize Albert Einstein in 1905, which had the intuition of considering light as packets of energy, photons.

These particles carry energy, depending on their frequency, which is transferred to an electron when a collision happens. If this is greater than a certain threshold, this electron will jump from the valence band to the conductive one, in a process called *absorption*. In the absence of an electric field, the excited particle will eventually bounce back to a lower orbital, emitting a photon with a wavelength equal to the difference in energy between those two levels in a process called *emission*.

A coherent motion of the electrons is needed to use this effect to generate an electric current. That is why some semiconductor materials, such as silicon and germanium, have been chemically and physically modified using the doping method and placed together to create a p-n junction.

Using this technique, we can create a material that has an area with an excess of electrons – the n side, which stands for negative – and another with an excess of holes – the *p* one, which stands for positive –. Some negative charges cross the junction, creating a hole electron pair across the depletion area, generating a small electric field with an opposite direction compared to the original one.

When the cell is stroke by the solar light, if we prevent the light-generated carriers from exiting the material, the number of free electrons on the *n* side and the number of holes on the *p* side increases because the previously generated electric field across the connection directs the pair towards their respective sides.

This separation produces a difference of potential which, after connecting the two regions with a conductor, will generate a current between the two areas, which can be used to power a load; this current will flow as long as solar light is provided.

Nevertheless, not all the energy from the solar beams is converted into electrical energy. Here are some of the main reasons:

1. Not all photons that strike the surface carry sufficient energy to excite

an electron.

2. Parasite resistances on the materials dissipate some energy in the process.
3. Due to the recombination effect, new excited electrons may encounter a hole in their path and recombine with another atom.
4. As the temperature of the cell increases, the band gap of the semiconductor materials decreases, allowing electrons to flow through the depletion area and thus decreasing the maximum potential difference achievable.
5. Not optimal exposition of the panel, shadow zones and debris present on the cell physically blocking light reduce as well the light intake.

2.3 Silicon mono-crystalline solar cell

Because of the reasons for which this thesis was produced, it is not possible to give an exhaustive overview of all the different technologies to transform sunlight into energy. This section will give an outline on the principles of operation of the panel chosen in the Cityscanner project, to facilitate a better understanding of the parameters investigated and the techniques employed during the dissertation for its power output estimation.

In the latest version of the prototype described in [14], the employed solar panel is a 9W, 6,2V mono-crystalline cells solar panel. Mono-crystalline cells are by far the most widely used solar photovoltaic technology, and they rely on the classic scheme of photovoltaic production, illustrated in chapter 2.1: solar light is converted to electrical current using a single pure semiconductor crystal. For this reason, the panel must be manufactured with an accurate method to avoid efficiency losses due to impurities.

The Czochralski process is employed to create this monolithic component, which takes the name from the polish inventor who discovered it in 1916. High-purity semiconductor silicon is melted in a crucible, doping atoms such as boron or phosphorus are added in exact quantities, then a precisely oriented seed crystal is dipped into the molten material and extracted rotating at a specific speed, temperature and angle allowing to extract a large, single-crystal, cylindrical ingot from the melt.

This technology allows to reach high efficiency, but it comes with some downsides in comparison to other more modern technologies. The main issues are the costs and the environmental impact, due to the high quantity of used material.

2.4 Important factors in energy estimation

Discovering methods and technology for accurate and reliable solar forecasting will be imperative in the future of solar energy. With a lack of solar forecasting and underdeveloped technology surrounding it, inaccurate forecasts can have expensive consequences.

Global solar radiation has an essential role in this task; accurate information on the number of solar beams that irradiate the deployment area is often required to obtain a reliable estimate of the total energy production [21].

The measurement of solar radiation is available in some specific areas but, given the cost of measuring equipment, maintenance and calibration requirements, for the zones where no sampling has been done, estimation techniques are used [22].

Another critical factor, which considerably affects the power generation of solar panels is temperature. Since solar cells are semiconductors, the current and voltage of these cells are significantly affected by temperature. When the temperature increases, the excited electrons can more easily get through the depletion area, creating parasite currents that lower the difference of potential between the anode and the cathode of the circuit, negatively impacting the open-circuit voltage [23].

Solar panel data sheets refer to test conditions, which usually consider a temperature of 25 °C, a solar irradiance value of 1000 W/m^2 and air mass 1.5. However, these conditions are not generally valid in a real-case deployment, given the variability of the weather in different areas of the earth surface.

Therefore, the estimation and analysis of cell temperature have received extensive attention in recent years to predict better the power output generated by a solar panel. Some models have been proposed to enhance this prediction, introducing methods that weight the dynamic change in outdoor parameters [24].

Depending on the location and the inclination where the solar cell is located, the daily irradiation (see appendix A) hours change. For this reason, is vital to consider the exposition together with the shadows generated by close objects such as trees and buildings. In urban deployments comparable to this case study, this factor is especially important, because it can significantly affect the generated current [25].

Lastly, deposits of dirt or snow on the solar cell can prevent the solar beams to hit the semiconductor, lowering its efficiency. In a general case, rain along with the panel inclination is enough to clean the surface regularly, but in arid or heavily polluted areas regular maintenance helps to sustain maximum open-circuit voltage [26].

2.5 Irradiation Estimation

Before assessing the impact of urban configuration and the influence of the route done by a moving solar panel, it is necessary to review the currently existing methods to estimate the amount of solar irradiation impacting the area of the deployment.

The existing approaches to predict Global Horizontal Irradiance (GHI) can be divided into two main categories: Physical and Statistical. Also, a third category combines them to get the best of the two approaches: Hybrid models [27].

2.5.1 Physical

The most widely employed physical methods are of two kinds; the first one consists of Numerical Weather Prediction (NWP) [28], which employs mathematical models of the atmosphere and the oceans based on current conditions, to predict future ones. The second kind consists of using Cloud Imagery data collected by ground and satellite stations, of predicting the behaviour of the clouds [29].

It has been demonstrated that cloud cover and cloud depth are the most influential variables regarding the prediction of the GHI. Chow et al. [30] presented a technique for cloud shadow forecasting using ground-based sky imaging at UC San Diego, which allows predicting cloud motion, casted shadow and irradiance. Nevertheless, only short deterministic forecast horizons are feasible using a Total Sky Imager (TSI), due to the presence of low clouds and high weather variability.

2.5.2 Statistical

The inherent issues of the physical models lead to explore methods belonging to the second class, which rely more on statistics and historical conditions. The statistical approach uses mathematical models that employ historical data to predict future values. They can be divided into:

- **Persistence model**.
- **Time-series models**. that include the Autoregressive Moving Average (ARMA) and different variations of artificial neural networks (ANN).

Persistence Model

Persistence model is the simplest one; it predicts the next value assuming it will be similar to the previous one. It is a naive predictor and is often used to benchmark other models [31].

Time Series Models

Time series models instead, are based on the historical data and are defined as a sequence of observations measured over time, such as hourly, daily and weekly.

The ARMA [32] model is composed by two parts: the Autoregressive part (AR), which predicts next value based on its past ones, and the Moving Average (MA) part, which helps to model the error as a linear combination of the error values at various time in the past.

In the study of Rui Huang et al. [33], for each month of the year, the Mean Absolute Error (MAE) of the ARMA model has been proven smaller than the one of the persistence model in the case of 1 hour-ahead forecasting. In January, for example, the ARMA model shows an improvement of as much as 44.38% compared to the persistence one.

The Autoregressive Integrated Moving Average (ARIMA) is a version of ARMA equipped with a differencing component d, which aims at reducing seasonality. Its improvement compared to ARMA is that it allows treating non-stationary series. In the research of Atique et al. [34], they achieved a Mean Absolute Percentage Error (MAPE) of 17.70%.

Artificial Neural Networks (ANN) has been demonstrated as a valid alternative to the traditional approaches being able to recognise patterns in data, and have been widely applied to solar forecast. In [35] it has been demonstrated that their artificial neural network outperforms ARIMA, K Nearest Neighbours and Markov Chains methods, utilising 19 years of meteorological data from Ajaccio, France.

Alzahrani and his team [36] used a Nonlinear Autoregressive Network with Exogenous Inputs (NARX) for forecasting global solar radiation using data from Vichy National Airport in Rolla. They achieved a Mean Squared Error (MSE), ranging from 5.7% to 15.33%.

Hybrid models

The use of hybrid models became more popular, as it takes advantage of combining different techniques to assess the weak points of the single methods. Practical and theoretical studies have demonstrated that these combinations can help to increase the forecast performance.

Karkados et al. [37] proposed two seasonal auto-regressive integrated moving average (SARIMA) techniques; the first, based on seasonal ARIMA time-series analysis, is further improved by incorporating short-term solar radiation forecasts derived from NWP model, while the second is ANN-based. They achieved a NRMSE average over the year of 11,12% with the first one, while the neural network scored 11,26%.

2.5.3 Online Tools

Finally, there are online tools that can give an estimation of the solar irradiance for a specific area, making use of data extracted from databases collected by different institutions. The most famous examples are PVGIS [38] (done by the European Commission's science and knowledge service, the JRC), PVWatts (developed by NREL) and RETScreen (commissioned by the Canadian Government) [39].

These applications have been created for quick estimations and calculations relevant to photovoltaic (PV) electricity production, but some of them include an extension that can also give an evaluation of solar irradiation using historical data from that location.

	Space Resol.	Time Resol.	Type	Cited Studies
NWP	1-150Km [30]	1-1000h [27]	Physical	[28, 27, 30]
Cloud Imagery	0,1-100Km [30, 27]	6h [31]	Physical	[31, 30, 29, 27]
Persistence	0,01Km [27]	1h [31]	Statistical	[31, 27]
ARMA	0-0,1Km [27]	1-36h [33]	Statistical	[27, 33]
ARIMA	0-0,1Km [27]	1-36h [33]	Statistical	[27, 33]
ANN	0,1-5Km [27]	1-36h [33]	Statistical	[27, 33, 35, 36]
PVGIS	Adjustable	Adjustable	Online Tool	[38, 39]

Table 2.1: Different solar estimation methods compared.

2.6 Irradiance in the Urban Environment

Given the fact that the deployment of the City Scanner devices and other solar-powered IoT applications happens mostly in an urban context, it is necessary to assess the influence of the built environment on the irradiance that directly hits the ground level. Many have assessed the impact of radiation in urban street canyons: in studies like Wang and Png [40] or Thorsson et al. [41] for thermodynamic reasons (to evaluate the Urban Heat Island effect (UHI)), in other researches for meteorological modelling or photovoltaic generation [42].

Nevertheless, as indicated by Gong et al. [43], there is a shortage of an analytical quantification of solar radiance at street level mainly for two reasons: firstly because current meteorological data consider free horizon, and secondly because most of the existing studies rely on computationally heavy 3D models, which dramatically reduce scalability and adaptability among different locations.

However, among the existing studies and tools, are worth mentioning Gong et al.'s [43] model and the ArcMap Solar Radiation toolset to evaluate the street-level Global Horizontal Irradiance.

Gong et al.'s model makes use of images extracted from Google Street View (GSV) to determine the Sky View Factor (SVF); it indicates the portion of sky visible from the ground in a particular point. Afterwards, it uses sun data extracted from the Hong Kong Observatory (HKO) and the SOLPOS algorithm developed by NREL [44] to understand the direction and intensity of solar rays. Combining these pieces of information, they have been able to assess the amount of irradiation reaching the streets in Hong Kong with an overall correlation coefficient for global solar irradiance under all-sky conditions of 0.87.

ArcGIS [45] is a desktop application that supports data visualisation, advanced analysis and data maintenance in both 2D and 3D. The solar radiation analysis tools included in this software enable to map and analyse the effects of the sun over a geographic area for specific periods. The calculation is based on methods from the hemispherical viewshed algorithm developed by Rich et al. [46, 47] and further developed by Fu and Rich [48]. The calculation of direct, diffuse, and global irradiance are applied on the given topographic surface, producing heat maps for an entire geographic area.

Chapter 3

Design

3.1 Problem Elaboration

As introduced in the first chapter, City Scanner is a drive-by sensing platform aimed at analysing environmental parameters in the urban context. The system is composed of a fleet of devices installed on cars, equipped with several sensors, a solar panel and a battery, and of a back-end application capable of collecting data and able to send commands to the devices.

It is possible to adjust the behaviour of these modules depending on the aim of the deployment, using both these remote and local settings. To be able to optimise their functioning mode, it is first necessary to understand in detail the energetic needs of each module, to prevent prolonged periods of downtime due to lack of power. Being City Scanner a sensing platform, we will refer to the project goals as **sensing goals**; examples are *space coverage*, in case the focus is reaching as many streets as possible, and *time coverage*, where the interest is in having a constant data flow for a prolonged period.

Thus, to make data-driven decisions useful to reach our sensing goals, the system has to adapt real-time in the best way possible also to several external variables:

1. The meteorological conditions, directly correlated with the location and the period where deployment takes place.
2. The city configuration; the presence of obstacles, such as very high buildings, can obstruct the direct solar rays for most of the days, drastically

reducing the available solar energy at street level.

3. The route of the vehicles influences the sensing goals and effect of the previous point.

To take into account all these variables, in this chapter will be presented several components able to maximise the efficacy of every deployment.

3.2 System description

A complete system is necessary to create an automated platform orchestrator, capable of dynamically respond to the previously presented external factors.

This system can be logically separated into four different modules, each addressing one aspect of the bigger problem:

1. Device module, which focuses on the efficiency of the devices.

2. Solar module, which is the one presented in this thesis, focused on predicting the available energy.

3. Path forecasting module, which anticipates the path taken by the vehicles

4. Coordination module, which responds in real-time aggregating data from the other modules.

The focus of this dissertation is to extensively study and develop the second one, namely the Solar module, but following a brief description of each of them can be found.

To effectively evaluate the second module, a data collection phase was needed. Before this stage, some improvements were introduced in the prototypes, to increase the overall on-field performances. The upcoming sections will also include a concise description of the technical modifications propaedeutic to this thesis.

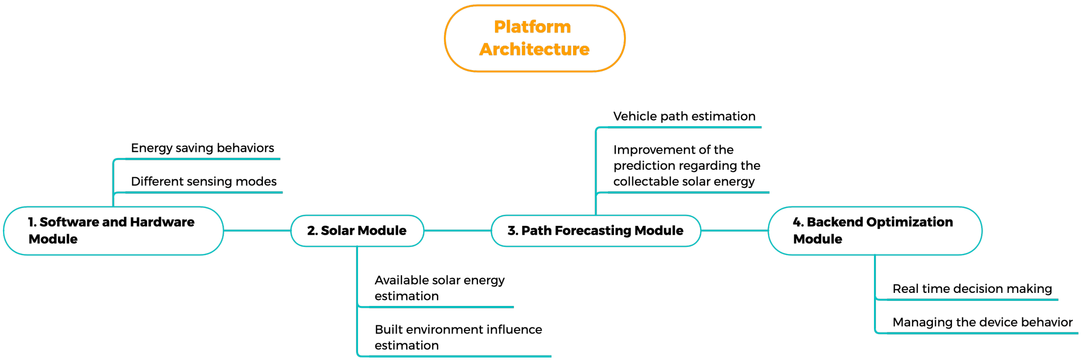

Platform Architecture
1. Software and Hardware Module
Energy saving behaviors
Different sensing modes
2. Solar Module
Available solar energy estimation
Built environment influence estimation
3. Path Forecasting Module
Vehicle path estimation
Improvement of the prediction regarding the collectable solar energy
4. Backend Optimization Module
Real time decision making
Managing the device behavior

Figure 3.1: Architecture chart

3.2.1 Device module

Before dynamically adjusting the behaviour of the devices, it is necessary to embed some energy-saving functioning modes, to prolong their base operating time.

The task of this module is to have a responsive device, which can save energy whenever possible, that can work in different operating modes depending on the objective, and that can be controlled remotely from a command line. These functionalities are critical for the success of City Scanner, aiming at an independent fleet that can sustain without any human intervention.

At the beginning of this research, the devices had only one functioning mode, which was collecting samples periodically, without considering the battery level and they had to be manually deactivated at the end of the day. The need for a more efficient implementation drove to this new prototype version, with the resulting additional characteristics:

- It enters a standby state, switching off sensors and connection if no movement is detected, to save energy in case the car stops moving. When motion is detected, the device awakens and starts sampling.
- It enters a standby state similar to the one previously described if the GPS signal is absent for a prolonged period. In this case, most of the times the car is parked inside a garage, and also samples without any location metadata are not usable.

Moreover, these devices were able to send data only through a proprietary API of the programmable board used, and these frames were managed at the server-side by an instance of Node-RED. This browser, flow-based programming tool built on Node.js, allows to wire together APIs and online services (cite) and allows to parse and operate on the incoming packets. During this first step, we introduced the possibility of transferring entire files through a TCP connection from the command line, with noticeable improvements in the transfer speed and decreasing costs.

Furthermore, in the previous implementation, the Node-RED service was managed by the producer of the board in the beta version. Because we needed more customisation and stability, we transferred the service into an Amazon Elastic Compute Cloud (EC2) server hosted by Amazon Web Services. The Node-RED instance was deployed inside a Docker container, reachable through an Apache reverse proxy and a protected HTTPS connection.

Lastly, some functioning modes were added to the devices, so that they could have different behaviours depending on the battery level and the sensing goals, with the sampling frequency remotely adjustable, in particular:

- Idle. The device waits for commands, mainly used for troubleshooting.
- Real-Time. Similar to the previously existing mode, the device sends a constant stream of data.
- Logging. The device records all samples on an SD card and transfers them to the server on demand.
- Hourly. The device records data and stores them on the SD card while being offline, and connects for 5 minutes per hour to receive commands.

3.2.2 Solar Module

The objective of the solar module is to understand how much energy is available to use in a specific location and period of the year. This information is decisive for a successful outcome, assisting the preliminary decision-making phase, during and after the deployment:

- Ahead, it can assist throughout the planning phase to predict the operating time and to understand how much data is potentially achievable.

Moreover, knowing this duration in advance allows informing the partners about the best time to recharge the device, if needed.

- During, because in combination with module 2 and 3, we can dynamically adjust the device operating mode to save as much energy as possible based on the route taken.

- After, to evaluate the quality of the deployment and to verify that everything operated properly.

This module is further composed of two sub-modules:

1. Solar irradiation forecast, which outputs the solar radiation on an area given the weather forecast

2. City coverage analysis, which output is used to understand how much the built environment influences the solar radiation at street level.

In chapter 5, this section will be described more in detail.

3.2.3 Path Forecasting Module

City Scanner, for its scalable, non-intrusive nature, aims to deploy its devices excluding the use of special-purpose carriers. In fact, the ideal hosts of the sensing nodes are vehicles that already travel through cities to perform other tasks, such as trash trucks, taxis and buses. In this way, environmental parameters can be continuously monitored at lower traffic and pollution impact. Depending on the selected medium, the path of the vehicles will differ; in the case of buses or trucks, their itinerary is deterministic, making it trivial to forecast the energy produced through the second module. In case taxis are chosen, we can have excellent coverage of the city utilising a few devices, as demonstrated by O'Keeffe et al. in [49] for Manhattan, but their behaviour is less predictable.

For this reason, this component responsibility is to understand in which streets the vehicles are headed whenever the selected carrier behaves in a non-deterministic fashion. Once combined with the previous module output and with the collected data, this information permits to dynamically select the most suitable operating mode to achieve our sensing goals while preserving battery life.

If, by predicting the path of a specific car, we understand that the current vehicle is heading towards an area of the city where at that time of the day the buildings obstruct most of the sun rays, and the device has low battery, it would be possible to suspend it to save battery life in case another one has already sampled those street segments.

3.2.4 Coordination Module

This last module is conceived to take decisions and changing dynamically the behaviour of the devices, based on the real-time analysis of the output of the previous modules during the deployment.

In fact, when the devices are active in a city, all the generated data and metadata are sent to the remote server, hosted on the cloud. Depending on the objectives of the specific deployment, this module has to take decisions at single-device and fleet level, working towards the maximisation of the selected sensing goal.

This component acts as a fleet coordinator; it is necessary to make all the previous modules work in an organised process towards the common purpose. When data are received, it parses and organises them in databases for further analyses, but at the same time, it can directly provide them to the second and third module. After having analysed their output, it successively sends to the single nodes fine-grained adaptations in every new state.

3.3 Chosen methods elaboration

Here will be presented the methods used to create the second module, in both its parts. Given the fact that City Scanner and other moving solar-powered IoT fleets have to operate in different environments and locations, they require high scalability. During the state-of-the-art evaluation, were preferred methods that keep in mind this requirement, so that they are tunable with minimal effort in changed conditions.

3.3.1 Irradiation forecast module

As introduced in the previous chapter, exists different methods to estimate the solar irradiation given the weather forecast. For our research goals, the ideal approach should be able to provide an adequate estimate of the solar radiation for any location in the world with minimal changes. As previously explained, it ought to be usable before, during and after deployment without a model re-adaptation.

For this reason, the model that we have chosen takes as input meteorological data of the location and outputs the Global Horizontal Irradiance (GHI), for allowing the user to provide either the weather forecast for the following days or the historical weather data. In our situation, statistical methods are the most suitable compared to physical ones, due to their ease of use and adaptability. In chapter 5 will be shown a performance comparison of different statistical approaches, with their associated pros and cons.

3.3.2 Urban influence on irradiation

In this second part, values most taken into account are again scalability and ease of use, with a difference; were favoured computationally lighter and open source methods. During the evaluation, methods that imply the usage of proprietary tools or techniques that require a 3D model of the city were put in second place. For this reason, were analysed mostly GSV image-based methods, to come up with a plug and play solution for this problem.

Chapter 4

Data collection

4.1 Deployments

During the last months, a City Scanner deployment took place in New York City on the 21st January and went on for three weeks, specifically in the borough of The Bronx. This on-field experiment was beneficial for testing the previously added software enhancements and essential to gather the necessary data for validating this research.

In fact, from this deployment, we were able to obtain more than 120'000 data points, containing air quality data and useful metadata, such as GPS coordinates, the energy produced and the battery levels.

In figure 4.1, it is possible to see the spatial distribution and density of the collected samples during these periods; the methods used in this study are validated with The Bronx data.

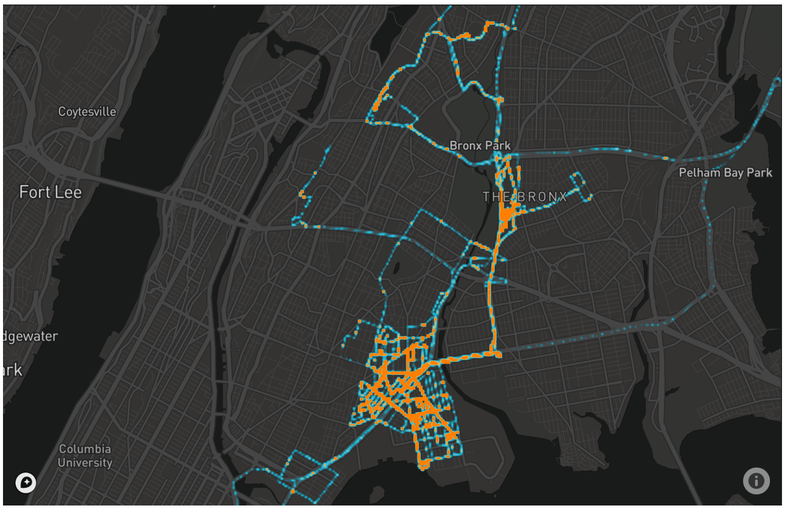


Figure 4.1: Collected points during the City Scanner deployment in The Bronx.

(a)

(b)

Figure 4.2: City Scanner deployment in The Bronx.

4.2 Datasets Description

In this section will be presented the dataset used in this research. Our devices, during the deployment, gathered two datasets; the first one contains air quality data, while the second one contains metadata about the devices status and the spacial and time location of the sample. The metadata dataset is the one that has more relevance to this research, and it contains the following features:

- Device Id, which identifies the device that recorded the sample.
- Timestamp, UNIX timestamp of the collected sample.
- Latitude.
- Longitude.
- Error on latitude and longitude.
- Internal device temperature.
- Solar panel instantaneous voltage.
- Solar panel instantaneous current.
- Battery instantaneous voltage.
- Battery instantaneous current.
- Battery level.

Moreover, New York City provided us with a dataset containing the deployment vehicles behaviour; the available information is:

- Vehicle Id.
- Start Date, which indicated the starting date and time of a drive.
- Driving Duration.
- Stop Date.
- Stop Duration.
- Miles Driven.
- Max Speed.
- Idling Duration.

4.2.1 Solar Estimation Model data

During the solar module study and testing, were used other three datasets. In particular, the first one, namely National Solar Radiation Database (NSRDB), contains historical meteorological data for the New York deployment area. It is available open-source for download through a web application or APIs from the National Renewable Energy Lab (NREL) website [50]. It includes weather data from 1997 to 2018 collected every half an hour, with a spatial resolution of 4x4 Km. The available features in this dataset are:

- Year, month, day, hour, minute.
- Temperature.
- Clearsky GHI, DNI, DHI.
- Cloud Type.
- Dew Point.
- GHI, DNI, DHI.
- Fill Flag.
- Relative Humidity.
- Solar Zenith Angle.
- Surface Albedo.
- Pressure.
- Precipitable Water.
- Wind Direction and Speed.

The data used in this research can be retrieved by supplying these parameters to the provided API:

```
lat, lon =  40.829, -73.897  # Location data
year = '2010'                # Starting year
n_years = 8                  # Number of years from the starting one
leap_year = 'true'           # Whether to include leap years or not
interval = '60'              # Time interval between samples
```

The issue with this dataset is that our deployments happened after 2018, so another dataset containing more recent data was employed.

To our knowledge, a free for use dataset containing very recent solar irradiation index was not available, so a weather dataset was joined with the previous one to create a predictive tool that could estimate Global Horizontal Irradiance (GHI) exclusively from forecast data; the resulting features of this merge will be presented in chapter 5. We chose to use World Weather Online (WWO) [51] in this phase because it has data about the needed locations and provides a free trial period, but similar services can be employed as well. In particular, its Python3 interface with the following parameters to retrieve the necessary weather information [52]:

```
frequency = 1                 # One sample per hour
start_date = '1-JAN-2010'     # Start date
end_date = '31-DEC-2018'      # End date
location_list = ['10467']     # The Bronx zipcode
```

In a similar way, the meteorological information about the deployment period was retrieved, namely from 21-JAN-2020 to 21-FEB-2020.

4.2.2 Google Street View Images

As introduced in section 3.3.2, the employed methods heavily rely on Google Street View Images. For this reason, the first step consists in obtaining the location points where the study is needed, as latitude and longitude pairs; this can be done either with the help of a Geographic Information System (GIS) software or manually.

As a next step, we developed a JavaScript tool able to, for each latitude and longitude pair, downloads the panoramic identifier of the image from Google Street View (GSV), namely panoId [53], along with the car facing direction in relation to North. These unique identifiers were later provided as input to a Python3 script that, for each location, retrieves all the tiles of the panorama, orients them towards the North and merges them, giving a complete fish-eye image of the area.

Performing this procedure with the data points sampled in the deployment and eliminating duplicates, we were able to obtain a complete photographic mapping of the interested area.

The last dataset utilised was retrieved from the Solar Position And Intensity (SOLPOS) website [54], and contained geometrical information about the sun position over the deployment area. It was essential for the second part of the solar module, to understand where the sun was in the sky at a certain point in time.

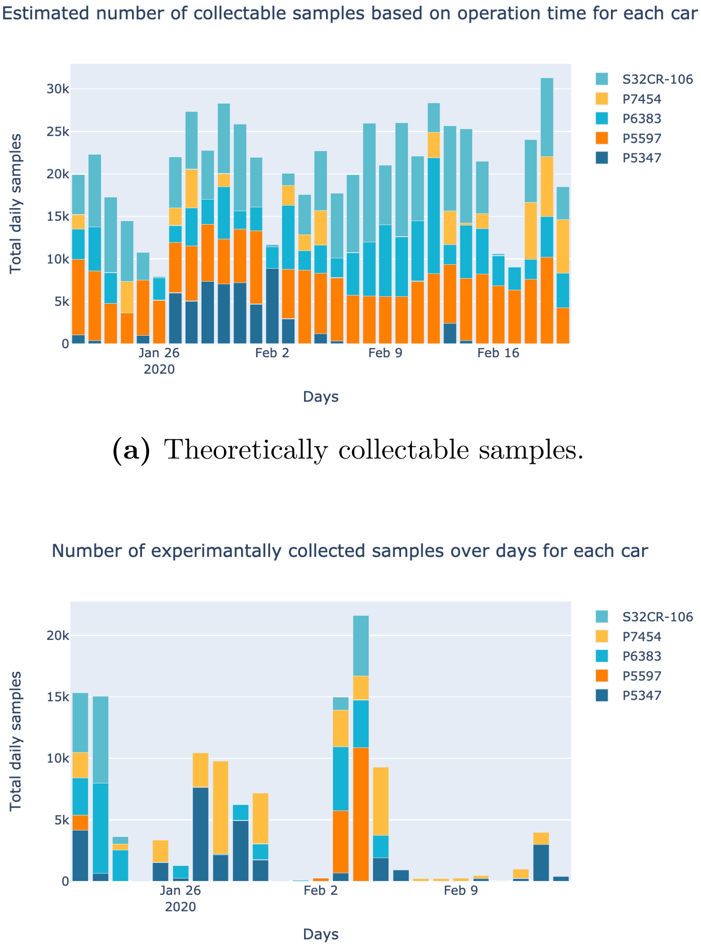
4.3 The Bronx Results

As a first step, we analysed the metadata dataset collected in New York, looking for meaningful insights about the deployment outcomes. This examination was necessary to understand if better energy management could effectively improve the results of the deployment, while it allowed us to understand some criticalities in the current implementation.

First of all, some pre-processing operations allowed to remove from the dataset outliers and meaningless data. In particular, were excluded from the dataset points with Horizontal Dilution of Position (HDOP) error on the GPS higher than 15. Afterwards, data points recorded in times where the cars were not operative, such as night hours, were also removed.

The first inquiry was to understand if the devices had been recording samples for the total of the driving hours or if they failed due to external or internal variables. To obtain this information, we begun with understanding for how many hours each day the deployment cars drove.

In the vehicles behaviour dataset, grouping by samples by the vehicle Id and day allowed to understand the total driving time in each working shift. To that time, we added three minutes per each stop, because the devices entered standby after three minutes without any solicitation, continuing to record data in that lag time. Finally, we obtained the maximum theoretical number of collectable samples by multiplying this total operational time for the devices sampling frequency. It is possible to see these results in figure 4.2.



(a) Theoretically collectable samples.

(b) Experimentally collected samples.

Figure 4.3: New York deployment results.

By comparing this number with the one recorded, it is possible to see that some devices had some operative issue. Most of them, such as the vehicle P7457 identified by the yellow line, worked properly having downtime only in case of low battery. However looking at the data can be seen that others, such as P5597 identified by the orange line, stop working on the first day. This behaviour cannot be due to battery issues; in this case, some debugging to understand the underlying reason is necessary.

Finally, these data show that the actual operational time, compared to the theoretically compute one, is overall much lower. From these graphs, excluding software issues, it is evident that a study about the energy produced by the solar panel is needed, to be able to predict the number of collectable samples in advance.

Chapter 5

Implementation

At the beginning of the chapter, will be presented the methods and procedures utilised to obtain a forecast on the hourly GHI levels. The second part, instead, will show how the city influences this estimate.

5.1 Solar Irradiation Forecast

First of all, it is crucial to understand and correctly pre-process of the available data to build a powerful regression model.

5.1.1 Pre-processing

As a first step, the information regarding the Solar Zenith Angle and Global Horizontal Irradiance were extracted from the NREL dataset and merged over date and time with the historical weather dataset previously obtained from World Weather Online. Considering that the Solar Zenith Angle is yearly recurrent, the only parameter that we cannot access using weather data exclusively is the Global Horizontal Irradiance, which becomes our target variable.

As a next step, some features not useful for our goals such as minimum daily temperature, wind chill information and moon illumination, were excluded from the data. All the numeric features misinterpreted as strings were converted into floating-point or integer values.

Successively, the rows containing information about nightly hours were filtered out, because in that case, the irradiation is always zero. This operation was easily achievable comparing the hour of the sample with the hour of the sunset and sunrise, present in the dataset.

As a next step, we added to the data frame the Extraterrestrial Solar Radiation (EHI) using the following formula [55]:

$$G_{on} = G_{sc}\left[1 + 0.033\cos\left(\frac{360N}{360}\right)\right] \tag{5.1}$$

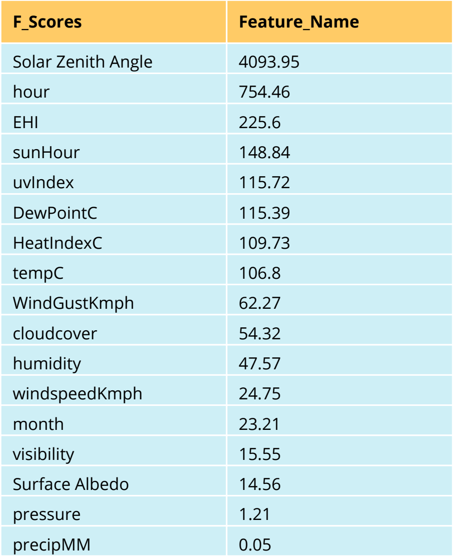
where:

G_{on} = the extraterrestrial radiation measured on the plane normal to the radiation on the Nth day of the year (W/m^2).

G_{sc} = solar constant (W/m^2).

Once the dataset was complete, it was necessary to understand the importance of the different features adequately. For this reason, the `SelectKBest` function from `sklearn` Python3 package was applied to the dataset using the `f_regression` scoring function.

This method tests the individual effect of each of the multiple regressors, calculating the correlation between the single regressor and the target variable, allowing to understand which are the most prominent features on the decision. The results of this method, along with the final dataset features, can be seen in the following table:



F_Scores	Feature_Name
Solar Zenith Angle	4093.95
hour	754.46
EHI	225.6
sunHour	148.84
uvIndex	115.72
DewPointC	115.39
HeatIndexC	109.73
tempC	106.8
WindGustKmph	62.27
cloudcover	54.32
humidity	47.57
windspeedKmph	24.75
month	23.21
visibility	15.55
Surface Albedo	14.56
pressure	1.21
precipMM	0.05

Figure 5.1: Features with the corresponding f-score function.

As expected, the height of the sun and the hour of the day are the most significant ones, while cloud cover has a secondary impact. Some columns, such as pressure and rain millimetres have a shallow effect on the decision but were kept in the dataset because we have not experienced performances issues, and the tests showed that the accuracy of the prediction was superior including them. No dimensionality reduction technique was employed for the same reason.

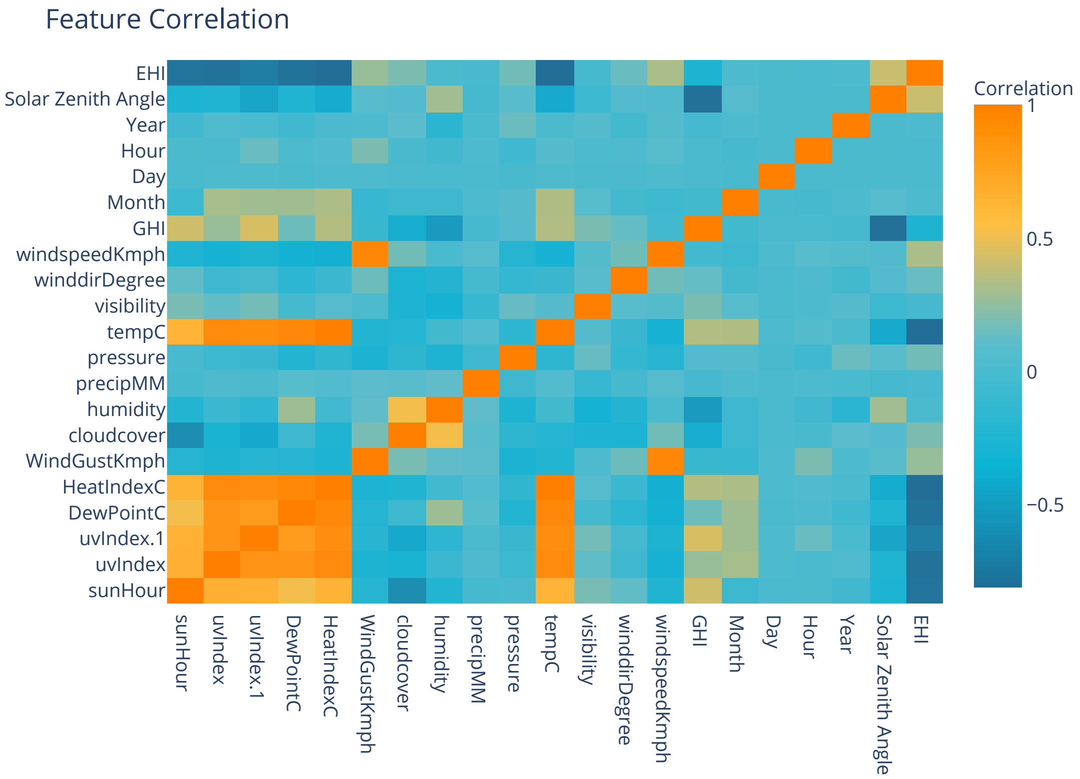


Figure 5.2: Correlation of the various features.

From the correlation matrix shown in figure 5.2, it is possible to see that the only noticeable correlation of the GHI feature is a negative one with the sun height with respect to the zenith (the higher the angle, the lower is the sun position). The only slight positive correlation of our target variable is with the Uv Index, with a value of 0,44.

Once decided the features to use, the shape of the dataset was analysed; the resulting frame contains 39694 rows and contains only numeric values, so no encoding technique was necessary.

5.1.2 Metrics

Before presenting the various regression models employed, it is necessary to give an overview of the performance metrics used. The first one is the root mean squared error, which is defined as:

$$RMSE = \sqrt{\frac{\Sigma_{i=1}^{N}(Y_i - \hat{Y}_i)^2}{N}} \tag{5.2}$$

This quantity indicates the average geometrical distance of the predicted values from the real ones, all under the square root; the smaller it is, the better is the model.

The second evaluation metric employed is the R squared one, which is defined like this in its more general formulation:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{5.3}$$

where:

$SS_{res} = \Sigma_{i=1}^{N}(y_i - f_i)^2$ is the total sum of squares, which is proportional to the variance of the data.

$SS_{tot} = \Sigma_{i=1}^{N}(y_i - \overline{y})^2$ is the residual sum of squares.

It is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. The higher this value is, the better the prediction. Minimising these values during training is not enough to evaluate the performances of a model; even with the best possible value for RMSE or R2, the model has to achieve a score as high during the test phase; otherwise, it is a sign of overfitting the data.

Successively, a copy of the dataset was made. One of these data frames was used as previously described, without further modification, while the other was scaled and centred. In the scaling phase, the `MinMaxScaler` from `sklearn` library was employed, because different solutions such as `RobustScaler` and `StandardScaler` yielded inferior regression performances. This specific scaler transforms each feature in this way:

$$X_{sc} = \frac{X_i - X_{min}}{X_{max} - X_{min}} \tag{5.4}$$

Bringing them in a range between 0 and 1 if positive, and [-1,1] if also negative values are present. It was convenient to separate the two datasets in this way because, in some of the tree-based methods employed, scaling is unnecessary or even detrimental.

To conclude the pre-processing phase, the two datasets were split into test and training sets using the `train_test_split` function of `sklearn`, with a test size of 20%.

5.1.3 Linear Regression

As the first step in this phase, a straightforward model was used to have a general idea of how complex the problem is. For this reason, a Linear Regression model was fitted and tested on the scaled dataset; it produced a cross-validation R2 score of 0.78.

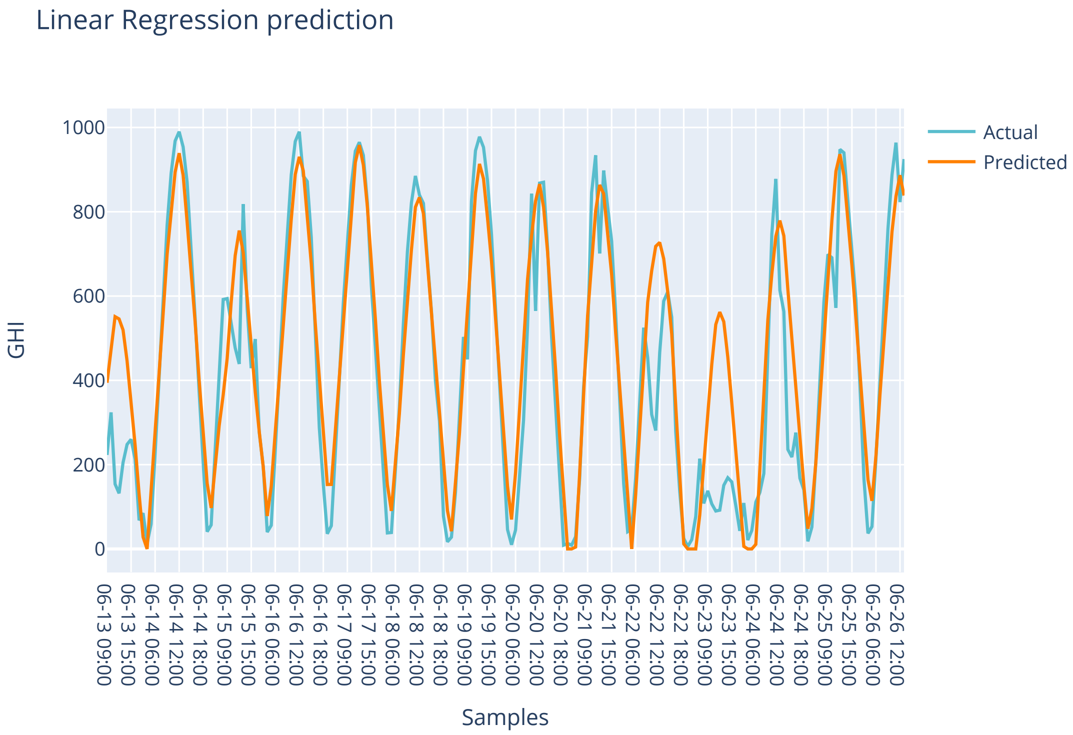


Figure 5.3: Results of LinearRegression model over 14 days.

It is possible to notice from the figure that the simple classifier can grasp the trend of the GHI during sunny days, but in more overcast conditions tends to stray from the actual values. For this reason, more complex classifiers will be presented successively.

5.1.4 SARIMA

ARIMA is the acronym for Autoregressive Integrated Moving Average, and it is one of the most widely used models in time series forecasting. The downside of this method is that it cannot support data that have a seasonal trend, and requires the removal of these cycles with techniques such as seasonal differencing.

SARIMA model, where S stands for Seasonal, adds three hyperparameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well an additional parameter for seasonality.

Before modelling, it is interesting to look at the behaviour of the Global Horizontal Irradiance during the day; we can see from the graphs that the trend repeats after 24 samples. This behaviour was predictable, being our target variable heavily dependent on the sun position. While in simpler models like ARMA is necessary to transform the input data in a stationary series if they are not, in the SARIMA model, the differencing component (I) can resolve the non-stationarity in most of the cases, so no further transformation is required. Moreover, the Augmented Dickey-Fuller test was done on the data; it returned a *p-value* of ~0, so the null hypothesis can be safely rejected and we can say that the target variable is in fact stationary.

For fitting the regression model to our data, it is necessary to find out the best value for all the components in the formula SARIMA(p, d, q)x(P, D, Q, m), where:

- p and seasonal P indicate the number of autoregressive terms.
- d and D represent the differencing needed to make the series stationary.
- q and Q indicate the number of moving average components, which are the lags of the forecast errors.
- m represents the seasonal period.

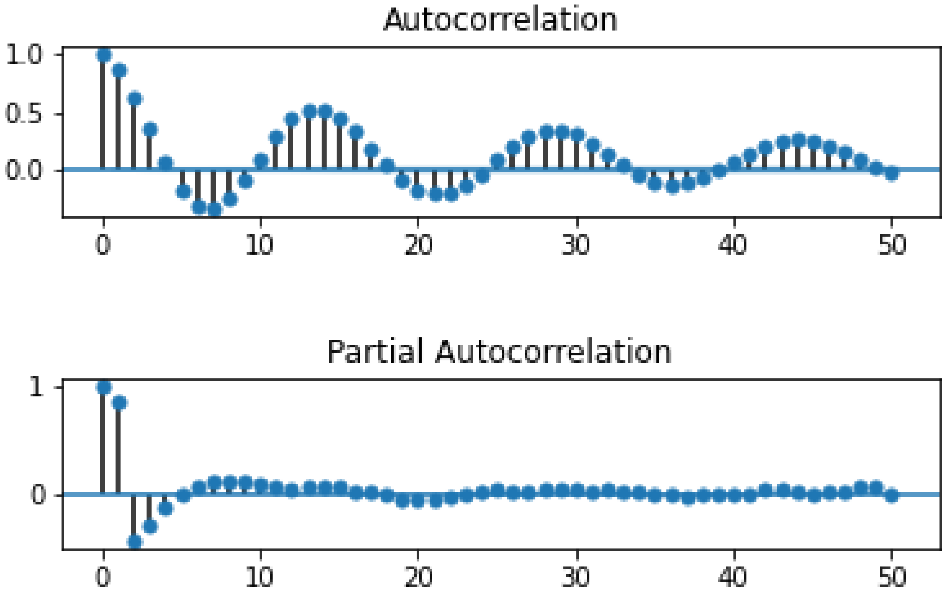


Figure 5.4: Auto correlation and partial auto correlation plots of the scaled data.

As a first step, it is useful to look into the plot of the autocorrelation function (ACF), intending to reduce the searching space for these parameters. It is possible to see that the trend oscillates, showing that we are dealing with periodic data. We set m equals 24, being our points sampled every hour over the course of the day.

Analysing the graph, it is possible to initialise the other values in this way:

- p and P equal 1, because of the significant positive spike in the PACF plot.

- d and D respectively at 1 and 0, because simple differencing could be enough, being our series already stationary.

- q and Q respectively at 2, due to the negative spike in the PACF plot at that position.

Plotting the ACF and the PACF using these test values, we noticed that no noticeable spikes or trends emerge. At this point, the function `auto_arima` from the `pyramid` Python3 package is used to perform a grid search over the specified parameter ranges to seek the optimal order for the model, providing the remaining features as exogenous variables.

The function returns the following values: (2,1,3)x(1, 0, [1, 2], 24).

Fitting with the Powell optimisation method with the `SARIMAX` function from `statsmodel` and evaluating the model, it gives an R2 score of 0.78 and an RMSE of 99. The plot in 5.6 shows the model accuracy over a week; it is possible to see that the model is able to grasp the trend of the data, but in cloudy days it has poor performances. Plotting the ACF for one last time shows that the process is now wholly uncorrelated, meaning that the used parameters are adequate, as it is possible to see in 5.5.

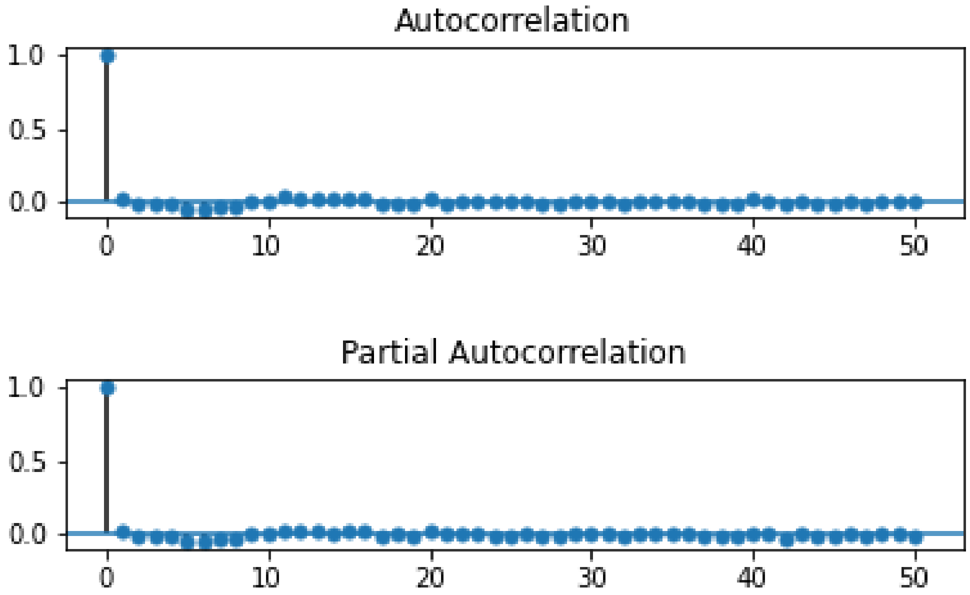


Figure 5.5: Auto correlation and partial auto correlation plots of the residuals.

To be able to compare these scores with the ones from the other models, a re-evaluation is needed after removing the night hours. After this procedure, the final score achieved is an R2 score of 0.73.

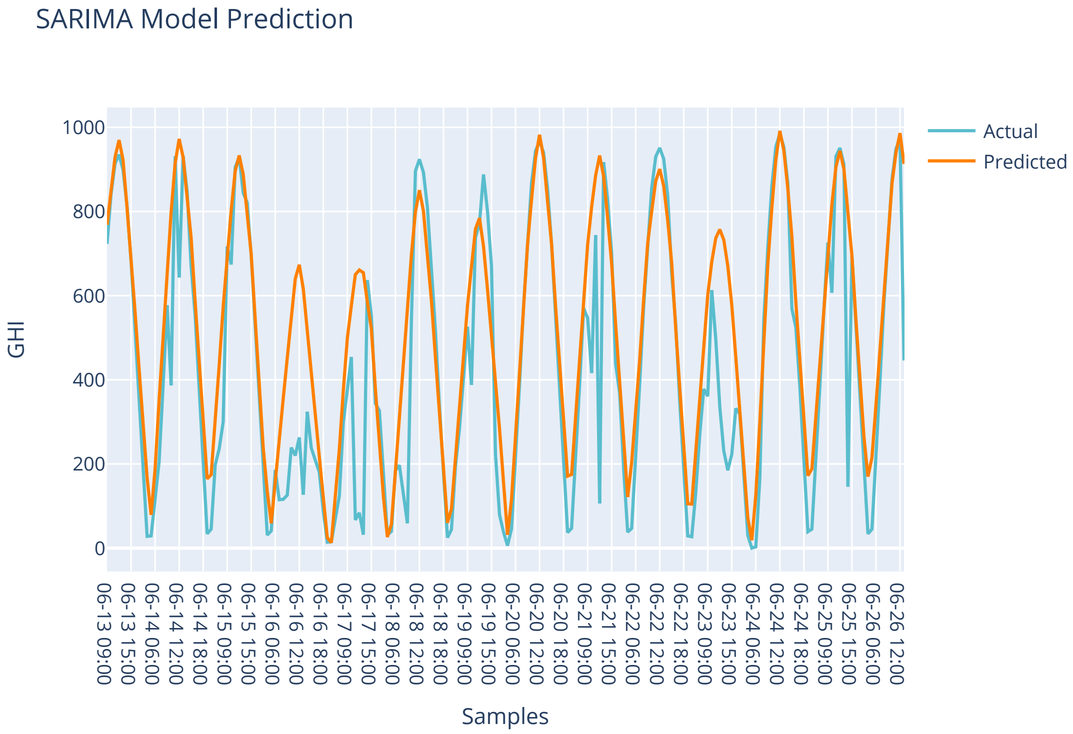


Figure 5.6: Results of SARIMA model over 14 days.

5.1.5 Tree based methods

Tree-based methods are considered one of the best and most widely used models in supervised learning. Their capacity to model non-linear relationships and their ease of interpretation make them suitable for both classification and regression problems. Even though neural network models tend to outperform tree-based models in unstructured data such as text or images, these methods have proved themselves to be better in case of tabular data.

We applied to our problem three different tree-based models and then improved their single performances by creating a stacked classifier. For optimising the hyperparameter, the Python3 package `hyperopt` was used; it allows to specify ranges for each parameter and performs a heuristic grid search over the possible values, improving the model performances in a reasonable time.

Gradient Boosting Regression

Gradient Boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, in this case, decision trees. It is an example of a boosting algorithm because, during training, subsequent predictors learn from the mistakes of the previous predictors. In fact, the observations have an unequal probability of appearing in subsequent models. Only the ones with the highest error appear most; samples are not chosen based on the bootstrap process but based on the error.

The hyperparameters of the model were optimised in 100 iterations, and the best ones were:

```
best_gb = {
        'learning_rate': 0.05,
        'max_depth': 11,
        'max_features': 'sqrt',
        'min_samples_leaf': 8,
        'min_samples_split': 0.1,
        'n_estimators': 975,
        'subsample': 0.9345503093012107
}
```

The final score of the model on the test set is an R2 of 0.856 and a RMSE of 99.18.

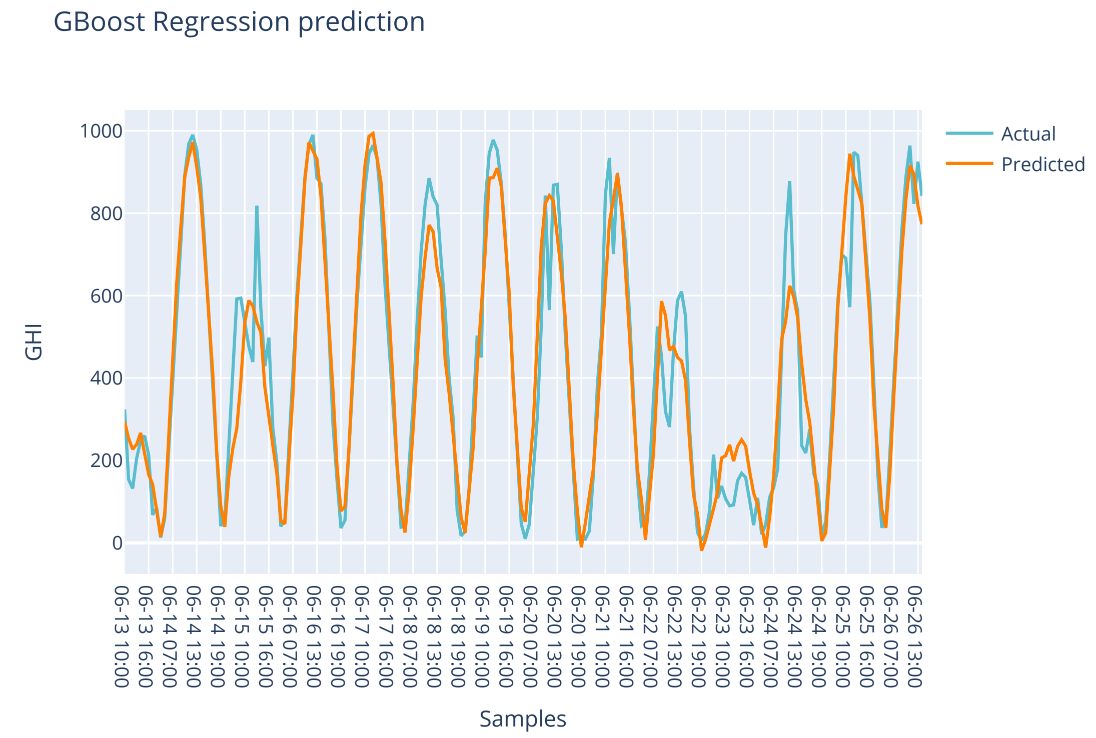


Figure 5.7: Results of Gradient Boosting Regression model over 14 days.

Is it possible to see in the graph that the accuracy is higher than the simpler linear model, even though some fluctuations cannot be adequately detected.

XGBoost

The second model employed is XGBoost [56], which is a is a decision-tree-based ensemble Machine Learning algorithm which uses a gradient boosting framework as well. Compared to the previously used method, it uses a parallelised implementation, and it has been designed to make optimised use of the hardware resources, making it very fast. Moreover, it presents more regularisation options and more randomisation, improving overall performance by reducing overfitting.

After the optimisation, the parameters used are:

```
best_xgb = {
        'colsample_bytree': 0.6,
        'eta': 0.9543574177848344,
        'gamma': 7.493135053072937,
        'learning_rate': 0.05,
        'max_depth': 11,
        'min_child_weight': 7,
        'n_estimators': 1375,
        'reg_alpha': 166,
        'reg_lambda': 0.8509315054191489,
        'subsample': 0.9,
        'booster' : 'gbtree'
    }
```

The score achieved using the `dart` booster was slightly better, but the training time was exponentially higher, so `gbtree` was chosen. The R2 and RMSE of the prediction on the test set are respectively 0.8972 and 86.57.

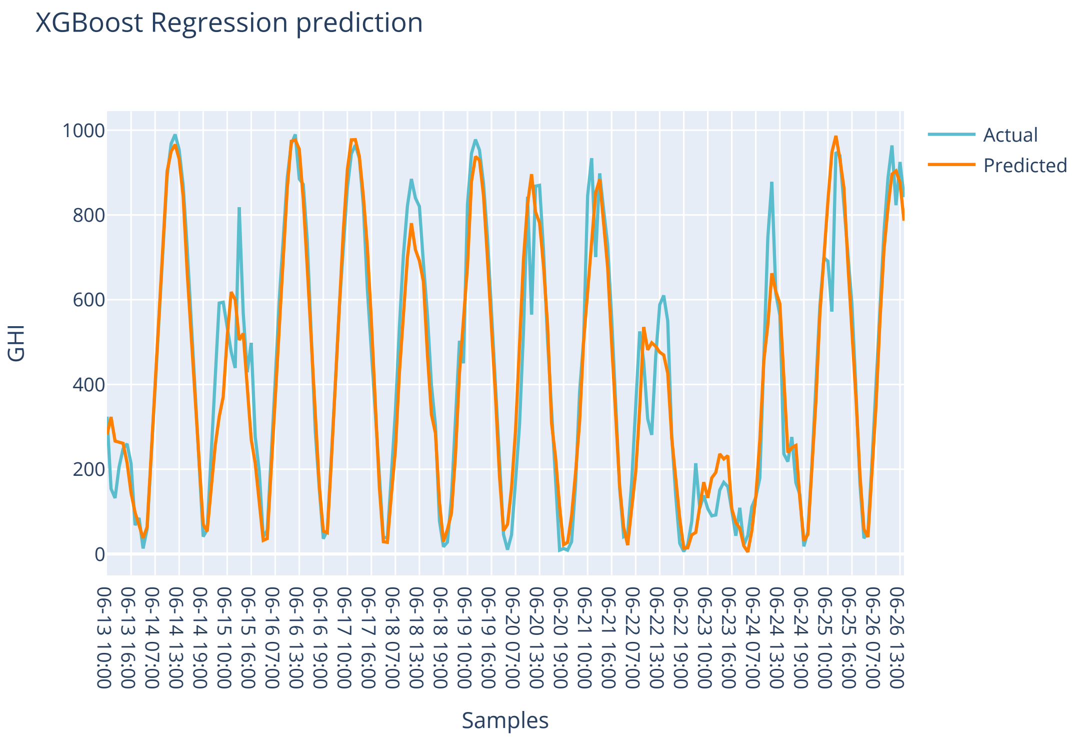


Figure 5.8: Results of XGB Regression over 14 days.

The score of this model is slightly higher than the previous ones; from the graph, it is possible to see a minor improvement, in particular on less regular days.

LGBM

The second method employed is LightGBM, another tree-based model developed by Microsoft Research [57]. It is based on Gradient Boosting Decision Trees, but presents some differences compared to XGBoost, such as the Exclusive Feature Bundling; it takes advantage of the sparsity of large datasets. The essential observation behind this method is that the sparsity of features means that some features are never non-zero together. In this case, they will be joined in a single one without losing information.

Moreover, it implements Gradient-based One-Side Sampling; the essential observation behind this method is that not all data points contribute equally to training; data points with small gradients tend to be more well trained (close to a local minimum). This means that it is more efficient to concentrate on data points with more significant gradients.

After the optimisation using `hyperopt`, the final parameters used are these:

```
best_lgbm_found = {
    'bagging_fraction': 0.9,
    'bagging_freq': 4,
    'boosting_type': 'dart',
    'colsample_bytree': 0.5,
    'learning_rate': 0.3,
    'max_depth': 15,
    'min_child_weight': 2,
    'n_estimators': 2900,
    'subsample': 0.5
}
```

And the final score on the test set is 0.894 for R2 and 88.1 for RMSE.

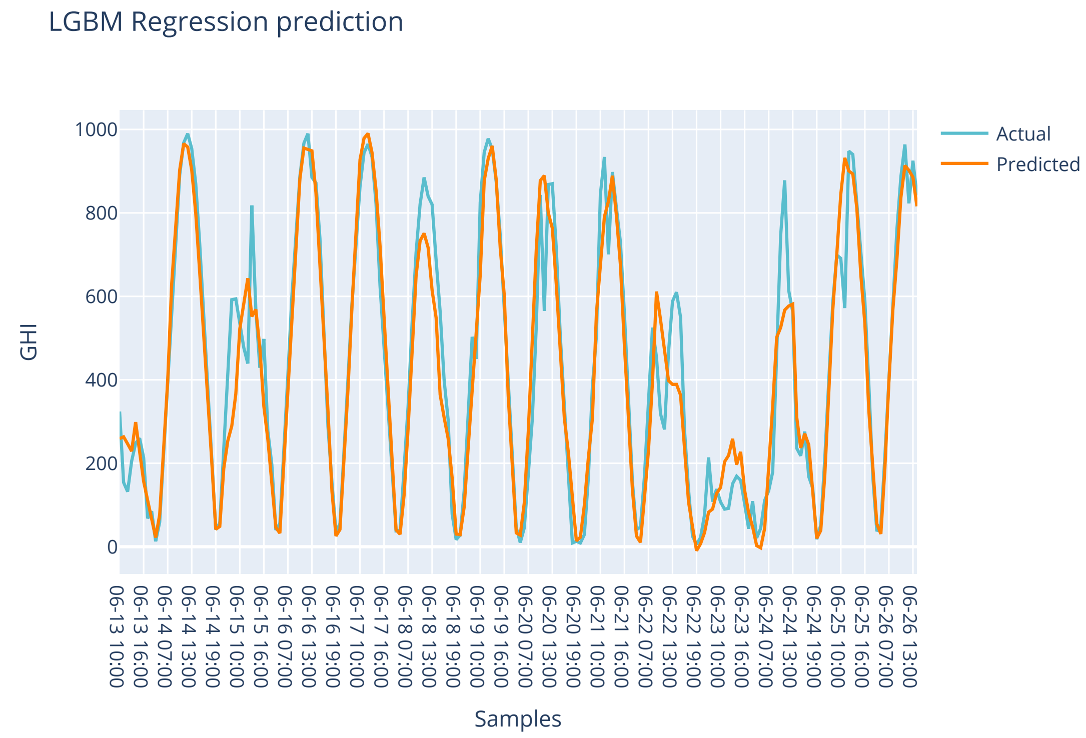


Figure 5.9: Results of LGBM Regression model over 14 days.

Again, the accuracy is high compared to the simple linear regression, and it can predict fluctuations better than XGBoost, even if its score is a bit lower.

5.1.6 Neural Networks

Neural networks are a useful technique also for tabular data analysis, requiring little feature engineering and less maintenance than other methods. For this reason, as presented in chapter 2, this method has been applied to this problem in several studies [58].

Feed Forward Neural Network

Feed-forward neural networks are artificial neural networks where the connections do not form a cycle. The name feed-forward indicates that the information travels only in one direction; every layer takes input from the upper layer and propagates into the successive ones. These architectures, also called Multi-Layer Perceptrons, are composed by one input layer, one or more hidden layers and an output layer.

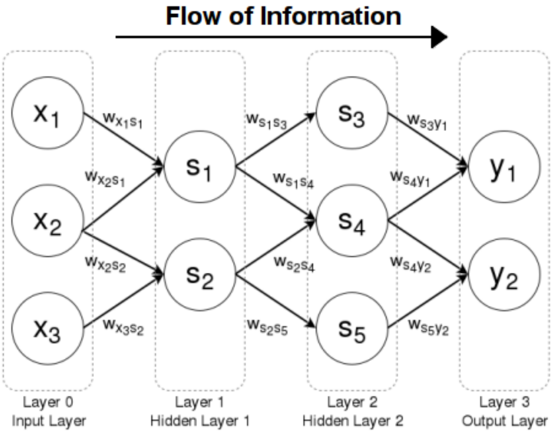


Figure 5.10: Architecture of a Feed-forward neural network.

In our case, as a starter model, we used one similar to [58], with one single hidden layer. This model could learn all the complexity adequately, and its test RMSE score applied to our problem was low. As a next step, we started adding fully connected hidden layers and augmenting the number of units, while tracking the shape of the loss on test data to avoid overfitting. Successively, we used Adam [59] optimiser, which gave us better results and faster convergence compared to Adadelta [60] and Stochastic Gradient Descent. With several trials, we set the learning rate to 0,0008 because it achieved the highest score.

At last, we introduced some regularisation techniques to reduce the overfitting on the input data and some normalisation layers to speed up training.

The first one of these layers is the Batch Normalisation [61]. This layer normalises the activations of the previous layer at each batch; it applies a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1. Even if the problem of covariance shift becomes more significant in deeper architectures, one of these layers after three fully connected layers helps to speed up training time by increasing the learning rate with a slight performance increase.

Afterwards, we introduced one dropout layer after some fully connected layer in the network [62]. The idea is to randomly drop units (along with their connections) from the neural network during training, which prevents input units from co-adapting too much.

Regarding the activation layers, in the hidden layers, the ReLu function was used, because it is more computationally efficient than other solutions, e.g. sigmoid, and introduces some non-linearities that can boost prediction score. Whereas in the last layer, the ’insert final layer’ function allowed to output a value between zero and one

The Python3 package `hyperopt` was again employed to fine-tune the number of units, the dropout coefficients and the number of layers to maximise its efficiency on our problem.

The final architecture is defined as:

```
random_norm_init = RandomNormal(seed=1)

model = Sequential()
model.add(Dense(128, input_shape=(X.shape[1],),
        kernel_initializer=random_norm_init,
        bias_initializer=random_norm_init,
        activation='relu'))
model.add(Dense(256, kernel_initializer=random_norm_init))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.11))
model.add(Dense(256, kernel_initializer=random_norm_init,
        activation='relu'))
model.add(Dropout(0.09))
model.add(Dense(256, kernel_initializer=random_norm_init,
        activation='relu'))
model.add(Dropout(0.11))
model.add(Dense(1, activation='relu'))
```

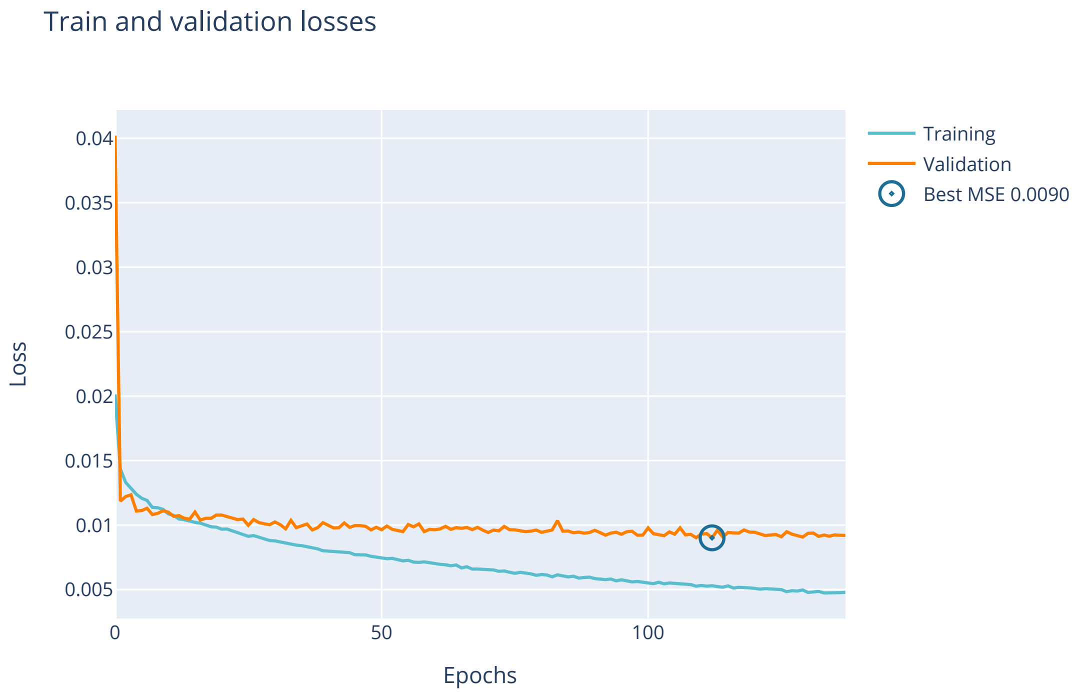


Figure 5.11: Training and validation losses of the Neural Network.

To better understand the correct number of epochs necessary to achieve the maximum score, an `EarlyStop` callback was employed. This function monitors the validation loss of the network through the training; if for n iterations the specified parameter has not seen any improvement, the training stops and the is restored the configuration with the best weights. The model in analysis, as it is possible to notice from the line chart, converged after 112 epochs with n set to 25, achieving a normalised validation score of 0.0090.

The neural network achieved similar performances to the trees based methods; its R2 score on the test set reached 87.4 while its RMSE touched a value of 93.

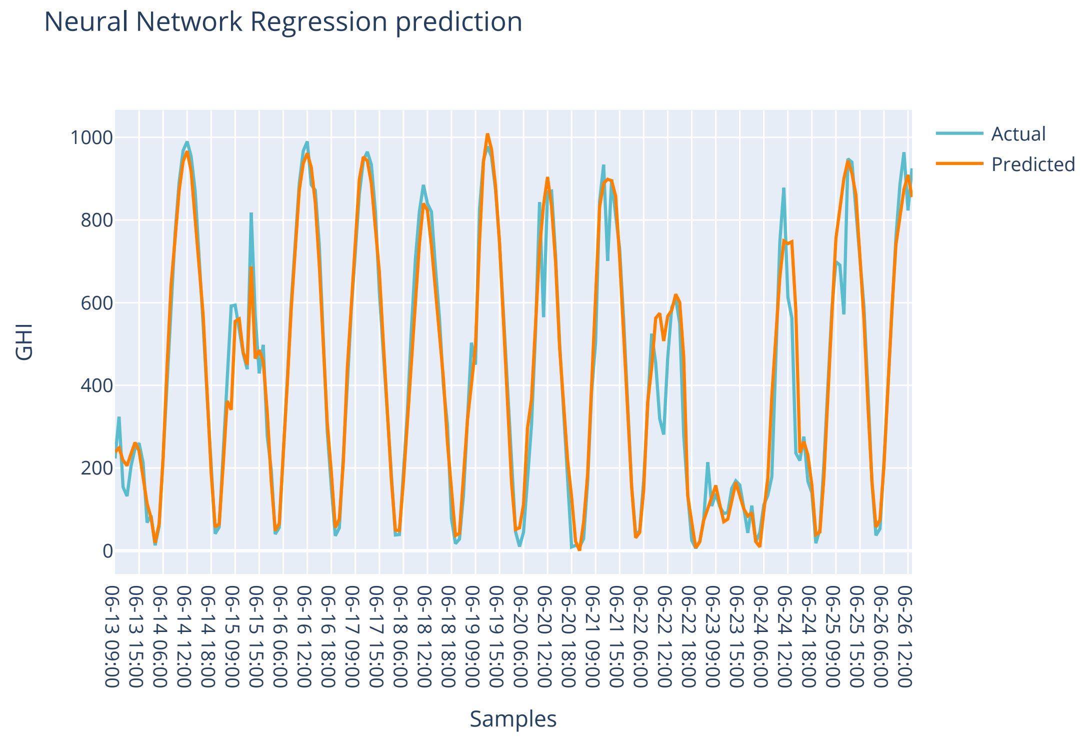


Figure 5.12: Results of Neural Network over 14 days.

5.2 Urban Irradiation

The output of the previous step, to assess the influence of the city on its estimate, has to take into account the type of environment where the fleet deployment takes place. For this reason, this section will present two similar approaches to identify and quantify the obstruction caused by trees or buildings. Both these methodologies are based on Google Street View images, but the assessment of the percentage of the covered sky is different.

5.2.1 Preliminary steps

As introduced in the data collection chapter, the first step consists of retrieving the images of the deployment's area. This collection was achieved using a JavaScript tool, which takes as input a file with the latitudes and longitudes of the streets of interest. For each of these coordinates pairs, it uses the google street view service [53] to retrieve metadata about the provided locations. The file collected in this way contains, for each point, this information:

- Date.
- Panorama Id, which identifies the unique code of the panoramic picture in Google's databases.
- Sample number, determines to which input pair this information corresponds.
- Latitude.
- Longitude.
- Heading, that indicates which direction the GSV car was facing when acquiring the picture, in relation to North. This information is crucial to be able to rotate the image to face that direction; otherwise, the geometrical calculations about the sun position cannot be successfully completed.

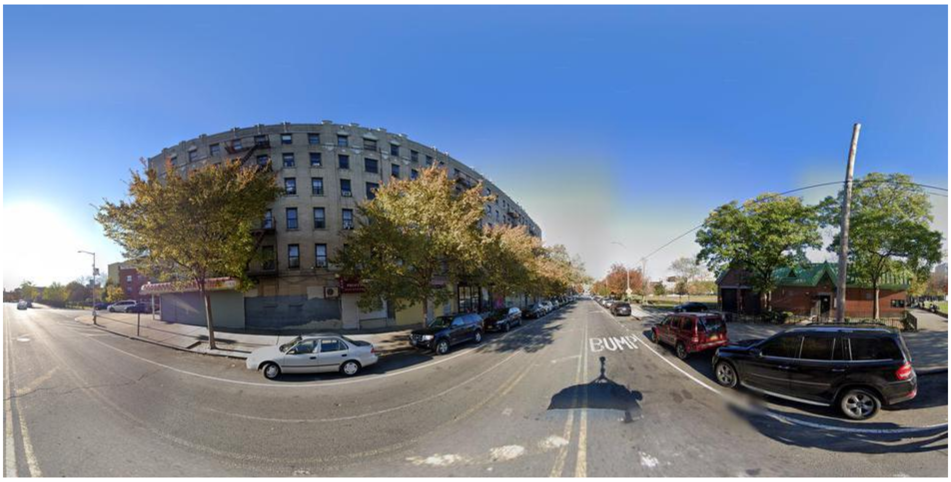


Figure 5.13: Example panoramic image, it corresponds to a street where the deployment cars drove by in The Bronx, in particular the coordinates are 40.818816, -73.89859162.

Once this file is available, the next step is to download the images corresponding to the supplied locations. We created a Python3 tool that could retrieve in parallel the needed images; the images are downloaded from the API in tiles with a shape of 26 horizontal tiles and 13 vertical ones, using a zoom level of 5. These tiles were successively joined together to obtain an image with 2:1 aspect ratio and 946x473 resolution; this specific resolution was chosen for the next step.

5.2.2 Image segmentation

Deep learning method

This methodology, proposed by Gong et al. in [63] makes use of Pyramid Scene Parsing [64] deep neural network to segment images in the different city components to identify the sky region.

PSPnet uses a first Convolutional Neural Network to create a feature map of the last convolutional layer, which sub-region representations are collected by a pyramid parsing module. This module is then followed by upsampling and concatenation layers to form the final feature representation, which carries both local and global context information. Lastly, this representation is fed into a convolution layer to obtain the final per-pixel classification.

This network reached optimum performances in classifying the different components of the urban environment, achieving a score of 80.2% accuracy on the Cityscapes dataset [65].

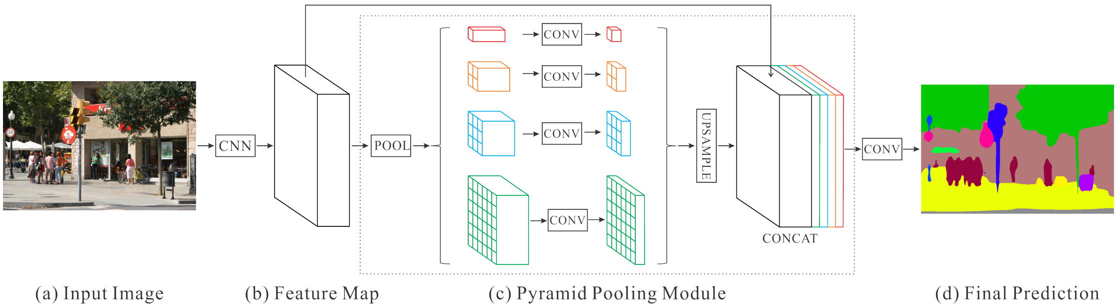


Figure 5.14: PSP net architecture.

Regarding the implementation, this [66] PSPnet Python3 version was used. Because the network takes as input bitmaps with 473x473 resolution, the downloaded pictures were split into two images with the requested size, fed into the model and re-joined after the classification.

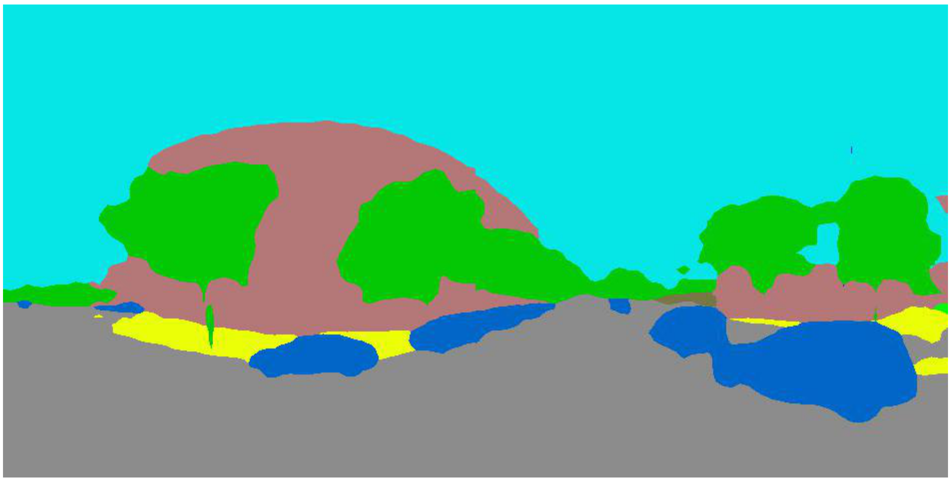


Figure 5.15: Segmented image using the PSP neural network.

The next step consists of changing the orientation of the images so that all are facing North. Using the information retrieved from Google Street View service, a portion of the image, equivalent to the rotation needed, either from the right or the left side, is moved on the opposite side of the image, obtaining in this way the correctly oriented panorama.

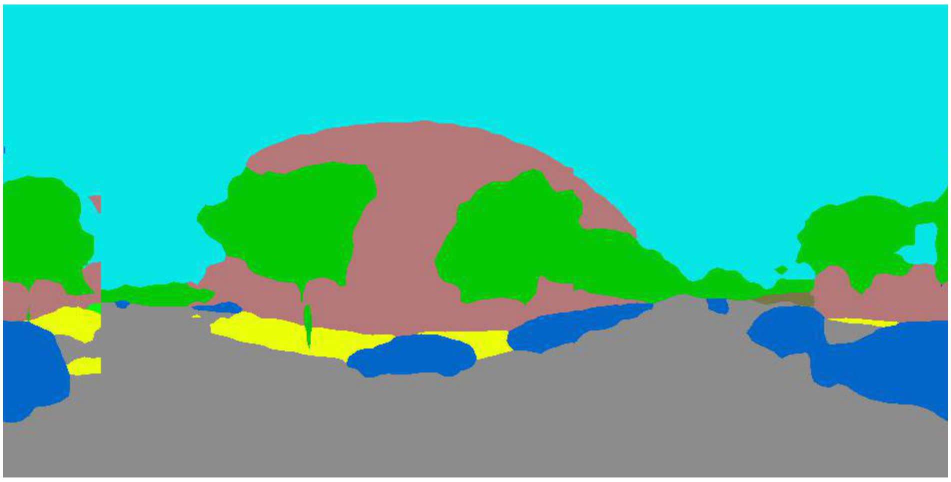


Figure 5.16: From the Google Street View API, we found out that the orientation of the car when the picture was taken was of 37.5 degrees to North. The section of the image corresponding in proportion to this value is moved on the other side.

Once we obtained these segmented views, a polar transformation allows converting the photos from a wide-angle panorama to fish-eye images, with the zenith in the centre. This procedure is the one employed in [67]. Having W_c and H_c as the height and the width of the cylindrical panorama, the radius of the transformed image becomes $r_0 = W_c/2\pi$, while the new dimensions of the image become W_c/π. The central point of the new representation (C_x, C_y) becomes:

$$C_x = C_y = \frac{W_c}{2\pi} \tag{5.5}$$

Each pixel of the original image (x_c, y_c) corresponds to (x_f, y_f) in the new one, following this relationship:

$$x_c = \frac{\theta}{2\pi} W_c \tag{5.6}$$

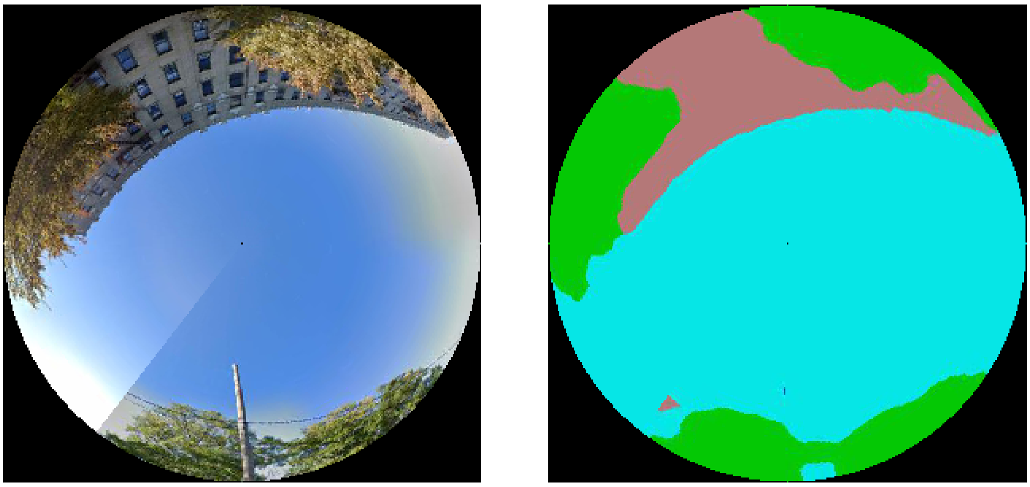
$$y_c = \frac{r}{r_0} H_c \tag{5.7}$$

where:

$$\theta = \begin{cases} \frac{\pi}{2} + \arctan\left(\frac{y_f - C_y}{x_f - C_x}\right), x_f < C_x \\ \frac{3\pi}{2} + \arctan\left(\frac{y_f - C_y}{x_f - C_x}\right), x_f > C_x \end{cases} \tag{5.8}$$

$$r = \sqrt{(x_f - C_x)^2 + (y_f - C_y)^2} \tag{5.9}$$

The result of this transformation can be seen in the following figure.

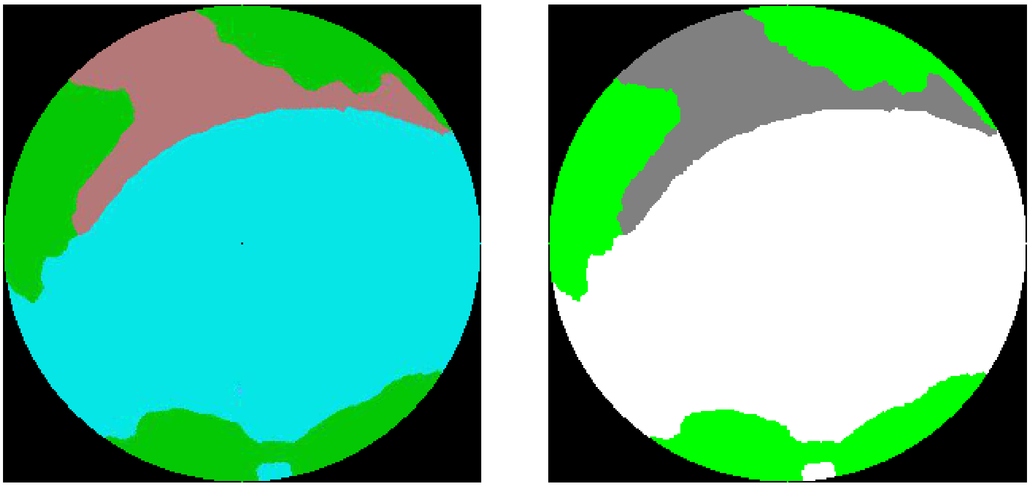


(a) Azimuthal projection of the cilindrical panorama.

(b) The transformation allowed us view the cilindrical panorama as an azimuthal fish-eye image.

Figure 5.17: Azimuthal projection of the panoramas pictures.

In contraposition with the original method described in the paper [43], some test showed that applying the mean-shift algorithm to the segmented fish-eye image on our deployment area yielded better results compared to the suggested approach. This algorithm in fact, which will be further explained in the subsequent section, allows reducing the shades of colours, decisively simplifying the classification process. As the last processing step, the function `cdist` from `scipy.spatial.distance` was chosen to calculate the distance of every colour of the picture from reference ones defined in a palette, to repaint the image with the preferred shades.



(a) Azimuthal projection of the cylindrical panorama after mean-shift algorithm; edges are more defined and there is less noise.

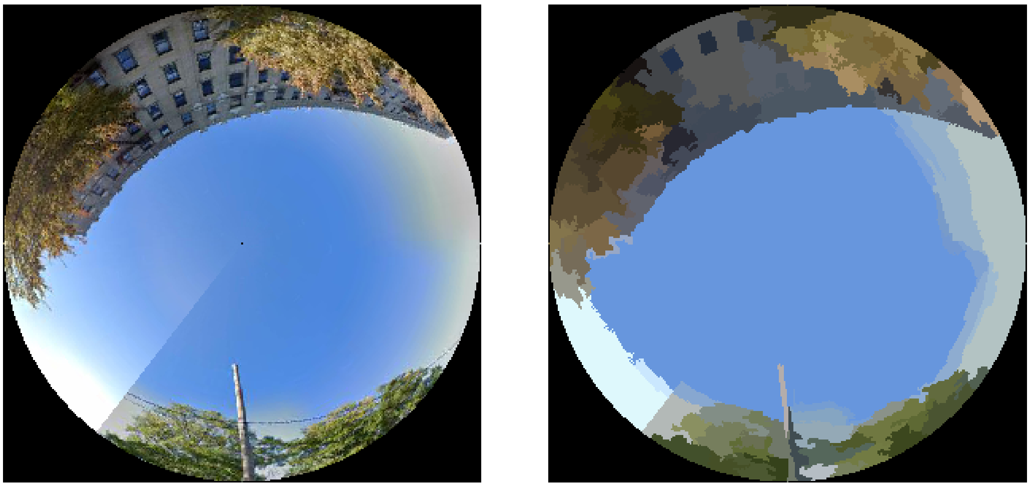
(b) Recoloring of the segmented image using geometrical distance from reference colours.

Figure 5.18: Mean-shited colours and repaint of the image.

Brightness Based Method

This other method, described in Li et al. [67], makes use of the mean shift algorithm to limit the number of different colours and to segment the pictures in more specific colour areas. With the segmentation results, it is possible to differentiate the sky pixels from the non-sky ones by calculating a brightness value for each of them and compare this value with a threshold.

First of all, the images are oriented to face North and transformed into azimuthal views using the same methods presented in the previous section. The mean shift algorithm, firstly presented in 2002 by Comaniciu et al. in [68] is a clustering algorithm that assigns data points to the clusters iteratively by shifting them towards the mode. In fact, given a set of data points, the algorithm iteratively assigns each datapoint towards the closest cluster centroid. The Python3 `pymeanshift` package was used as an implementation of this procedure.



(a) Azimuthal projection of the cilindrical panorama.

(b) Meanshifted azimuthal image.

Figure 5.19: Azimuthal projection of the panoramas pictures.

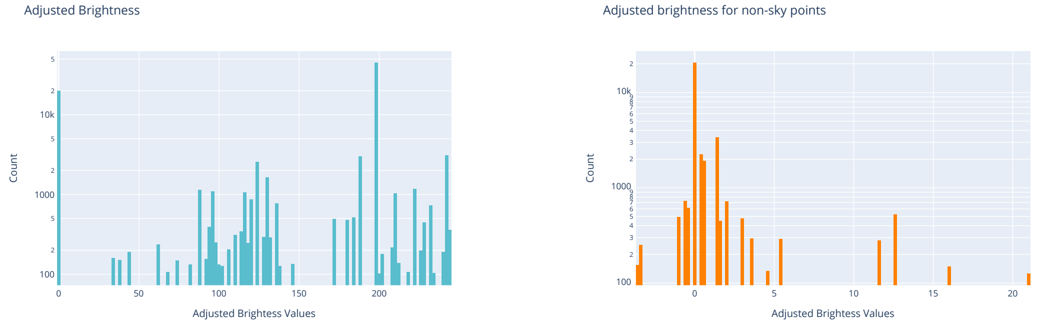
Regarding the classification, the intuition is that the sky points have the blue band more accentuated compared to buildings and trees; the brightness was adjusted giving more weight to this particular channel using the formula presented by Li et al. in [67]:

$$Brightness = \frac{(0.5 \times Red + Green + 1.5 \times Blue)}{3} \tag{5.10}$$

All the points that had this value higher than a certain threshold were classified as sky, while the remaining pixels were further separated into vegetation and buildings, using a similar technique to the one before, but with the colour green:

$$ExG = 2 \times Green - Blue - Red \tag{5.11}$$

As suggested in the paper, Otsu's method [69] was used to pick these thresholds; this technique looks at the set of values and tries to minimise the within-class variance. In other words, it works to find a cut-off value such that the variance of the resulting both classes in the Gaussian distributions is as small as possible.

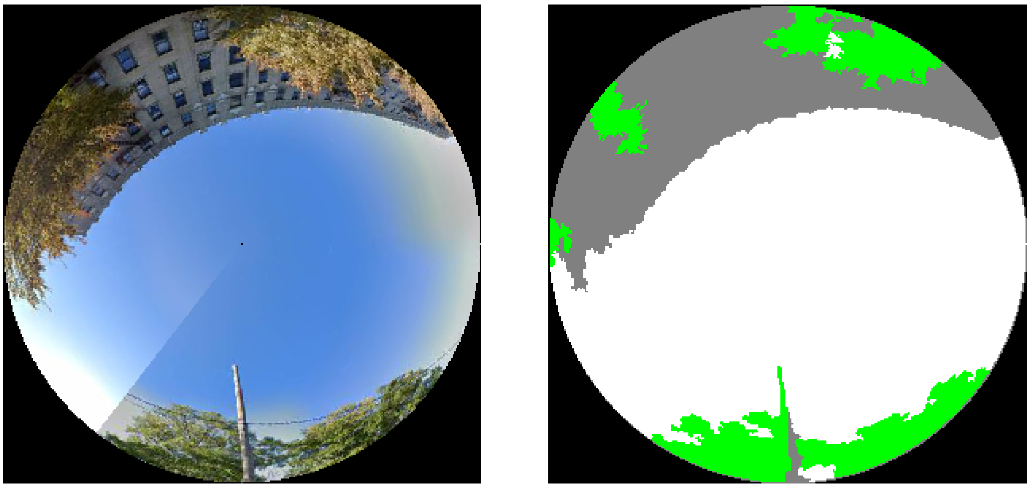


(a) Distribution of blu-band enhancing formula results.

(b) Distribution of green-band enhancing formula results on remaining points.

Figure 5.20: Distribution of brightness values on example picture containing sky, trees and buildings.

As it is possible to see from the figure, it is possible to separate the two classes respectively for values around 150 in the (a), and 6 in (b); applying Otsu's method for this sample picture the values returned are very close to these values.



(a) Azimuthal projection of the cilindrical panorama.

(b) Each pixel in the meanshifted image has been classified using adjusted brightness rules.

Figure 5.21: Azimuthal projection of the panoramas pictures.

As pointed out in Li et al.'s work, this method can be enhanced using some geometrical rules, but for this research, it showed a satisfying accuracy also without further modifications.

As it is possible to see in the picture, the trees that have a colour towards brown can be mistaken for buildings. In the irradiation assessment, they are treated as an obstruction to direct irradiation equally, so it is not problematic.

5.2.3 Irradiation reduction assessment

As a next step, is calculated the Sky View Factor (SVF) for each of the analysed points. The panorama images, after the polar transformation, have a resolution of 301x301. It is possible then to calculate the number of pixels contained in it using the formula for the circle area, being the actual image shaped like a circle with diameter 301. Once the photo is segmented, the number of pixels classified as the sky is calculated.

This number is then divided by the total pixel number, obtaining in this way a coefficient between 0 and 1, which indicates the sky portion sky visible from that spot.

Once is identified which part of the image is the sky and which one consists of buildings or trees, and after having estimated the SVF, is necessary to understand in which hours that street segment is illuminated. Using the information about the Solar Zenith Angle and the Solar Azimuth Angle contained in the SOLPOS [54] dataset presented in chapter 4, it is possible to determine the position of the sun at a specific time in the fish-eye images.

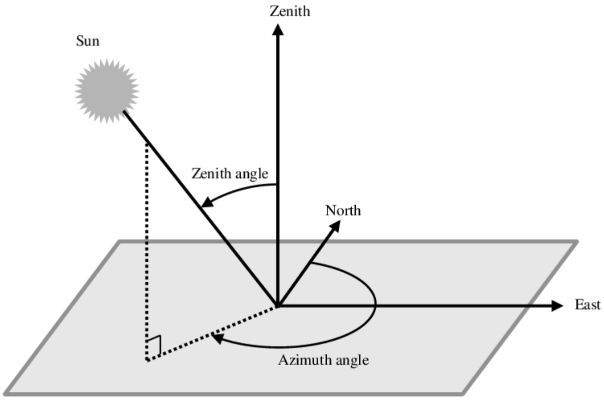


Figure 5.22: Using zenith and azimuth angle is possible to identify the sun position on a hemispheric image.

Once the point (x_f, y_f) is established, the nearest 8 pixels around the point are analysed to classify the spot using the majority class (obstructed or not obstructed).

The GHI value, estimated in 5.1, can be decomposed into DNI and DHI, to better assess its impact at street level. This decomposition is done using the well known Erbs et al.'s model [70]; in particular, the implementation done in `pvlib` is chosen. Once the two components of the GHI are obtained, the SVF and the information about the building and trees obstruction can be combined to have the street-level GHI, using this formula:

$$GHI_{street} = DNI \times \cos\alpha \times f + DHI \times SVF \tag{5.12}$$

where:

- GHI_{street} is the total irradiance at street level.
- α is the solar zenith angle.
- f is a binary value that indicates whether the sun path is obstructed or not; this information can be retrieved from the GSV images.
- SVF is the Sky View Factor.

Chapter 6

Evaluation

After having presented all the techniques tested on the weather dataset, to reach the best results on the prediction, an ensemble regressor is used.

6.0.1 Stacking Classifier

Multiple approaches exist to combine several models into one to improve the overall performances. For example, a simple weighted average of the different classifier predictions can boost the base score and the robustness of the single ones.

Stacking, similar to bagging and boosting methods, involves joining the predictions from multiple machine learning models on the same dataset. The architecture of stacking unravels into two levels; at the first level, two or more classifiers deliver a prediction in parallel on the same dataset. At the second level, a meta-model takes as input features the output of each model at the upper level, and learns how to combine them in the best way possible to predict the target variable.

The efficiency of stacking is maximised when the predictions made at the first level or their forecast errors have a low correlation or better they are uncorrelated. Regarding the types of models to use at the different levels, while the base models are usually complex, the meta-model instead is often simple, providing a smooth interpretation of the predictions made by the base models.

Using as base models LGBM, GBTree, XGBoost and the ANN, and using

Lasso regressor as a meta-model, we were able to increase the prediction accuracy, reaching an R2 score of 90.12.

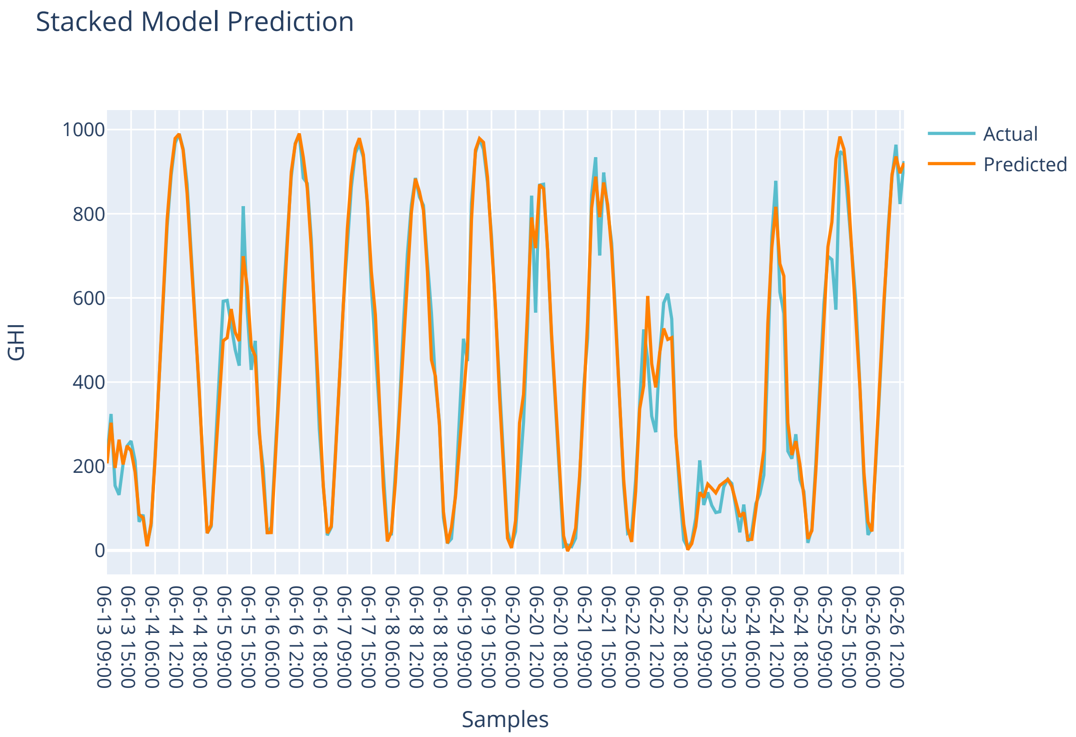


Figure 6.1: Results of Stacked model prediction over 14 days.

Another consideration needs to be taken into account on this forecast accuracy. We chose to exclude the feature Cloud Type, contained into the NSRDB dataset because it is not included in the most common weather datasets and APIs, and may result hard to retrieve for some locations.

Including that feature, the accuracy increased dramatically, reaching values of R2 of 0.96 on average, while staying over 0.99 for sunny days. Even if, with this modification, we were able to reach these high scores, we preferred scalability over accuracy, because the current implementation allows to port the system to another city with minimal effort. The network was tested on Los Angeles weather data to validate this claim, achieving a score higher than the one for The Bronx, namely 0.921 as R2 accuracy, most likely because of lower weather variability and a higher presence of sunny days.

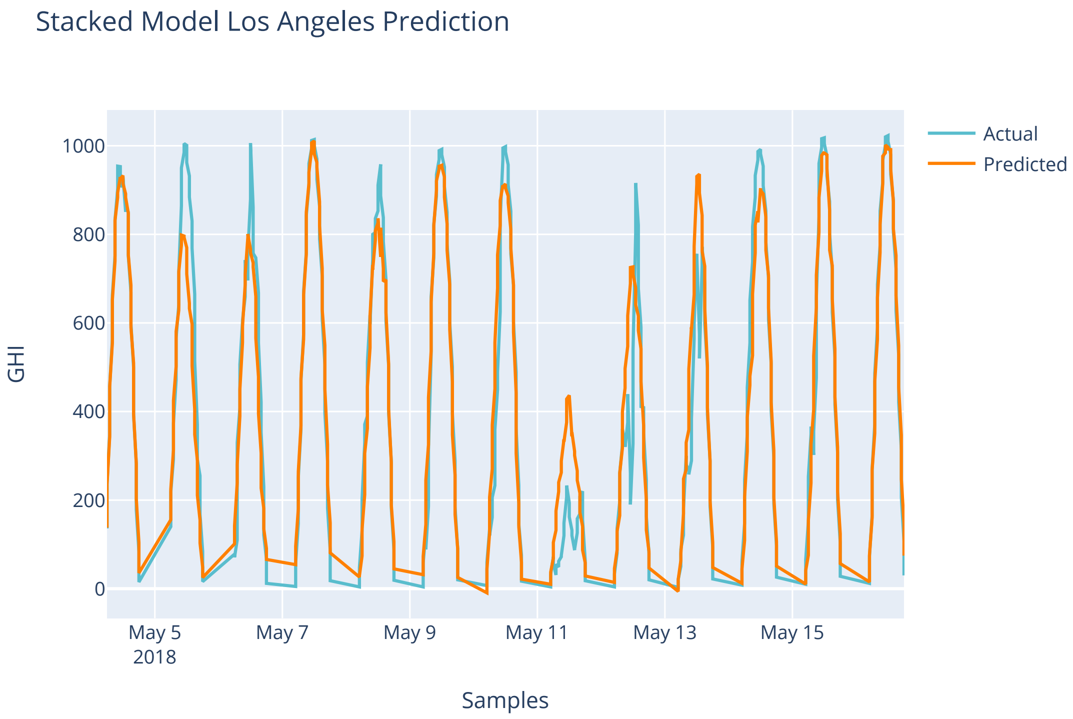


Figure 6.2: Results of Stacked model prediction on Los Angeles data without re-training.

There is also another way to increase prediction accuracy further. The data employed for the training, contained in the NSRDB, are calculated with a physical estimation model, and not sampled using tailored equipment. If the network is re-trained with data recorded with accurate instruments, we believe that the model's scores can reach higher values.

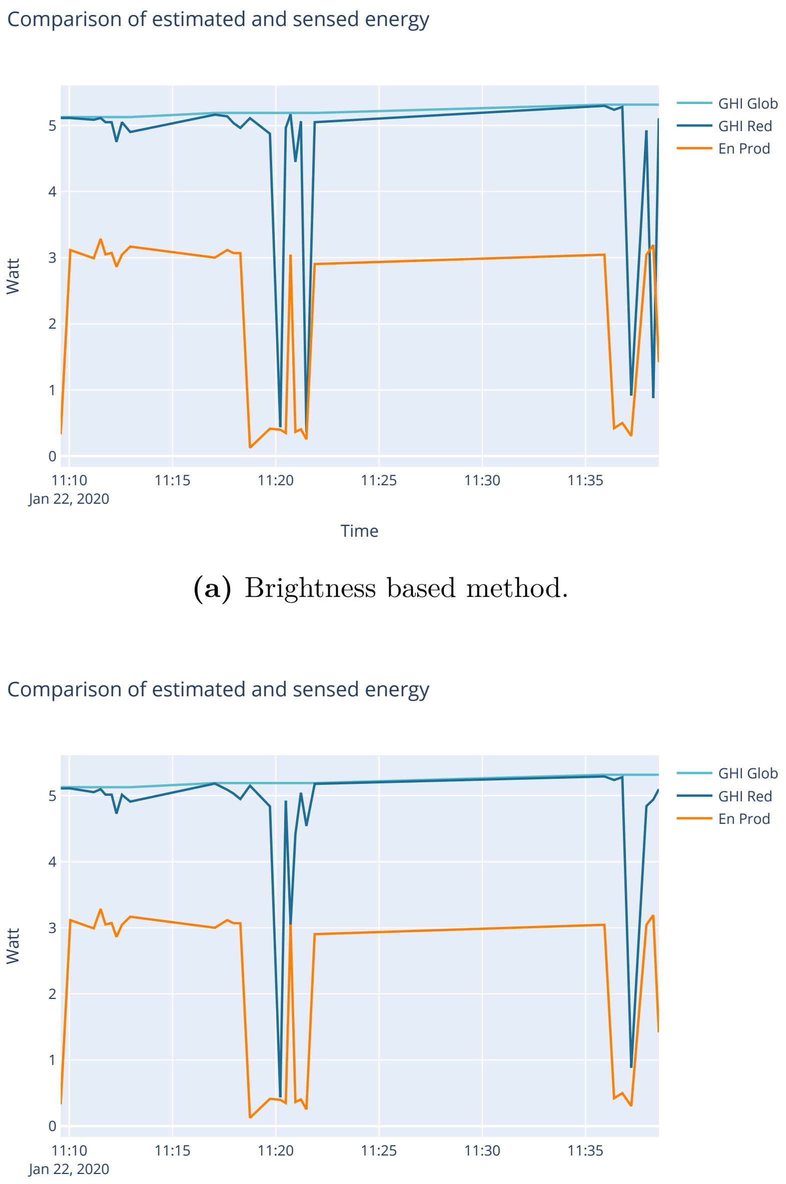
6.1 Irradiation reduction assessment

The two photographic methods described in section 5.2 were compared to assess the city's influence on the estimate of the first part. After the download and the pre-processing phase, the two techniques were implemented, applied and then compared with the data experimentally collected. From the tests, it emerges that the deep learning-based method is able to classify the different city components with higher accuracy. In particular, the brightness-based method performs poorly in case of trees with no leaves or with colours shifted towards autumnal shades.

Moreover, it misclassifies some components of buildings if they heavily reflect the sky (e.g. glass skyscrapers) or if they present bright colour facades. The advantage of this approach is in its speed; the Pyramid scene parsing network takes a noticeable amount of time to classify the images, especially on machines where a dedicated GPU is not available.

If high classification accuracy is needed, the best method is the first; the second, which gives slightly worse results but it is orders of magnitude faster, can be chosen in case of a very extended area of interest and a short amount of time. The advantage of both these approaches is that, for every street that the devices are expected to cross in deployment, a database of irradiation reduction can be computed in advance, reducing in this way the assessment phase to a simple table lookup.

In the following graphs, it is possible to see the performances of the two methods applied to the data collected during a deployment day by one of the vehicles, in particular on the 22nd January 2020.



Comparison of estimated and sensed energy

(a) Brightness based method.

Comparison of estimated and sensed energy

(b) Deep learning based method.

Figure 6.3: Comparison of brightness based and deep learning methods.

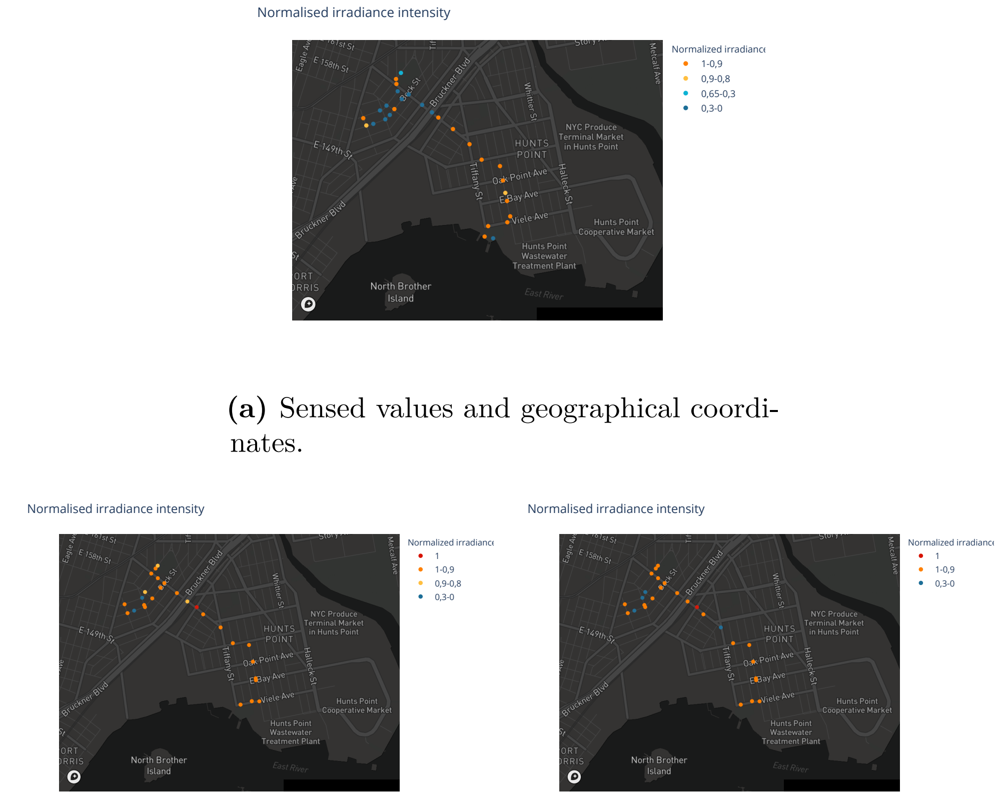
It is possible to see that the prediction, represented by the dark blue line, is similar. The brightness based method, being intrinsically noisier, has more obstruction points, which in this example follow better than the other the trend of the real value.

Moreover, it can be seen that between 11:10 am and 11:15 am there are some fluctuations in the energy produced; both models managed to predict in an accurate way this behaviour.

In the chat, all the measures are represented in Watts. As introduced in chapter 2, solar panels lose some of their efficiency when they heat up, due to parasite currents that appear between the adjacent cells. We did not consider this yield reduction because in the deployment period temperatures never exceeded 15 degrees Celsius and, at that temperature, the decrease is negligible. The sun energy was transformed to Watts through a simple multiplication for the panel surface area, namely 0,0583m^2.

There is an energy gap between the peak wattage supplied by the sun and the actual energy produced. This gap can be explained by the settings of the employed solar panel; the reason resides in the possibility to set the voltage threshold at which it starts to produce energy. If this threshold is high, the peak power yielded is higher, but it necessitates of more energy to start producing current. On the other hand, if this limit is low, the peak power will be lower, but the production is also triggered in overcast situations.

Being the deployment planned during the cold season, we decided to set the threshold lower to have a more constant energy flow, generating some current even in harsher conditions.



(a) Sensed values and geographical coordinates.

(b) Estimated values using deep learning method in GSV geographical coordinates.

(c) Estimated values using brightness method in GSV geographical coordinates.

Figure 6.4: Intensity calculated using the two methods compared to the one sensed in one street.

This alternative representation shows the differences between the two methods and the sensed values. First of all, it is essential to consider that the main limitation of these methods resides in the distance between one GSV photo and the next one. In fact, GSV API, given a latitude and longitude pair, will return the closest image to the needed location. This location could coincide with the requested point, but most of the time, it is slightly translated to a contiguous site.

These methods are susceptible to the precise position because a tree or a building could obscure one section of the street where the picture is available, and not obscure the remaining part.

Moreover, from the tests in The Bronx area, we noticed that the GSV car usually shoots panoramas at intersections, probably to maximise the field of view, and often, the SVF at the intersections is higher than the one in street tunnels.

This last hypothesis can be confirmed comparing the top-left zone of the figure 6.2a with the same area in figure 6.2b/c; it is possible to notice that in the second and third ones, the points are all placed at intersections with streets oriented towards the south-east, where the sun is located in that season in the morning. This phenomenon causes some misclassification, mistakenly increasing the number of illuminated points.

Chapter 7

Conclusions and Future Works

This last chapter aims to conclude this thesis, summarising what has been presented so far, highlighting the most important goals achieved and the limits of the proposed solution.

7.1 Overview

Recent technological advances allowed IoT devices, thanks to their low power usage and reduced overall dimensions, to have increasing relevance in applications such as remote sensing. In fact, in most of this kind of applications, the power from the electric grid is not always available, and alternative sources of energy are employed to sustain them. At the same time, the discovery of different materials to build solar panels was able to reduce their price, offering a convenient solution to harvest a resource widely available at low costs.

Many studies assessed the efficiency of solar panels placed in a precise location and inclination, but most of these models lose their efficacy when their position is not stationary anymore.

For this reason, the focus of this research is the understanding of the amount of energy produced by a moving solar panel, to enable the fleets of tomorrow to achieve their goals, anticipating their operation time in an unexplored environment. The presented method was validated on the data collected in The Bronx, NY, by the City Scanner project; a novel approach, conceived by MIT's Senseable City Lab, for developing a cost-effective, drive-by modular sensing platform to evaluate spatiotemporal environmental parameters in municipalities, such as air quality, ambient temperature and building thermal dissipation. These solar-powered devices follow a centralised IoT regime to generate a near real-time map of sensed data, and the individual sensing units are installed on urban vehicles to record data and stream it to the cloud for processing and analysis.

7.2 Objectives

The objective is to obtain a consistent method able to predict the irradiance over an area of a city and assess the influence of the buildings and trees on the final street-level values for the provided locations. This model must be able to to take decisions in all the phases of a solar-powered fleet deployment:

- Before, to forecast the collectable energy and thus the operating time to adjust the devices to achieve their goals.
- During, to make data-driven decisions on whether to enable power saving behaviours or to increase uptime (e.g. using a higher sampling frequency).
- After, to assess the performances of the devices.

This method must have the flexibility to adapt to different environments and has to be functional with no or minimal change in configurations. Moreover, the presented methodology is conceived as the first step towards a fully automated fleet manager, that can maximise the efficacy of the fleet deployment under the energetic aspect.

7.3 Methodology and results

As a first step, some software and hardware enhancements were introduced in the devices and the backend of City Scanner, that allowed to deploy the project in New York for one month at the beginning of 2020. During this deployment, have been collected more than 120'000 data points containing information about air quality and metadata about the energy produced by the panel and stored in the battery.

Successively, to obtain data to use as ground truth for irradiation, some models found in the literature and some not yet applied on this problem were tested on a combined database containing weather and irradiation data of the City Scanner deployment area. This approach resulted in a flexible procedure to predict the Global Horizontal Irradiance (GHI) in different locations with an R2 score higher than 0.90; it has also been tested in another city, namely Los Angeles reaching, without re-training, similar scores.

A further analysis allowed to assess the influence of the built environment on the previously obtained estimate; in fact, being the solar panels installed on vehicles at street level, they frequently enter zones shaded by human-made or natural obstacles. Two different Google Street View (GSV) imaging methods found in literature were modified and compared, to quantify the visible portion of the sky and the presence of obstructions for several streets in the area of the deployment.

One method required a pre-processing phase using a deep neural network, while the other classified images through brightness rules. The first one yielded better results but needed more time throughout the segmentation phase, while the second one was faster but less precise to identify the different urban components. In both techniques, panoramic images downloaded by the provided API were transformed into azimuthal fisheye views, and successively the sky pixels were classified from the non-sky ones. Combining this information with solar geometry data about the location of the deployment allowed the estimate the reduction on the solar irradiation values obtained in the first step.

The advantage of these techniques over software that makes use of 3D models resides in its inherent scalability, which allows creating a dataset containing this information for an untested location even if the 3D scan of the city area is unavailable.

The proposed method has been published on the platform GitHub [71] divided into three repositories, each solving a sub-problem of the bigger one:

- `pygsvpano` [72], which contains the tool needed to download the images from the Google Street View database.

- `fos` [73], from the Greek word light ($\varphi\tilde{\omega}\varsigma$), which contains the first part of the second module, namely the model to forecast the irradiation given the weather forecasts.
- `skia` [74], which means shade in Greek ($\sigma\kappa\iota\acute{\alpha}$), contains the function to assess the obstruction of the provided points given all the needed input data.

In the README.md files contained in these repositories is possible to find more precise instructions on how to use them. A high-level description of the steps needed to reproduce this research are:

1. Retrieve the car headings data and the panorama ids using the web tool inside the `pygsvpano` repository.
2. Download the needed images using the `pygsvpano` download tool.
3. Either load the pre-trained model or train a new one to forecast Global Horizontal Irradiance (GHI) values with the `fos` model.
4. Evaluate the reduction on the clear-sky GHI values using the function `perform_classification`, found in the `skia` repository.

7.4 Future works

As future steps of this project, the third and fourth modules presented in chapter 3 are needed to obtain a complete fleet manager. In fact, a component able to forecast the path of the devices can help to evaluate the available energy through the deployment day; using the energetic map created with the proposed method is possible to assign to a path a value that indicates its “energetic potential”. The forecasted power production is then provided to the last module, which aggregates the information from each device and dynamically adjusts their behaviour to achieve the selected sensing goal.

Furthermore, we will work on a higher-level interface on top of the provided python3 packages, capable of automatically gathering all the needed data and of performing the estimate straightforwardly. The only input data will be a set of latitude and longitude points with the associated timestamp in which the position of the panel matches with that location.

Given the increasing interest in renewable energy and the developments of the solar industry, the number of applications of devices powered by photovoltaics is expected to increase both in transportation and remote sensing fields. The presented methodology demonstrates then its future applicability also because of its flexibility, offering open-source components that can be employed on all kinds of solar panel powered fleets, to schedule in the most efficient way their behaviour in view of their goals.

Appendix A

Solar Energy

Solar irradiation is the incident energy per unit area on a surface, found by integration of irradiance over specified time (hour/day); the measure unit is J/m^2 [75].

Solar Irradiance is defined as the amount of electromagnetic energy produced by the sun incident on a surface per unit area (W/m^2). This energy calculated on the earth surface can be decomposed in different components.

The Direct Normal Irradiance (DNI) is the energy received by a surface perpendicular to the current position of the sun carried only by rays that come in a straight line, namely the ones that directly hit the area without any reflection. This value is directly dependent on the height of the sun with respect to the azimuth, in particular to Solar Zenith Angle (SZA). The Diffuse Horizontal Irradiance (DHI) is composed by the light which does not come directly from the direction of the sun, but instead is scattered by molecules in the atmosphere or reflected by buildings or ground, and comes equally from every direction. At last, the Global Horizontal Irradiance (GHI) is the combination of these two values on a horizontal surface; it is calculated with the following formula:

$$GHI = DNI * cos(SZA) + DHI \tag{A.1}$$

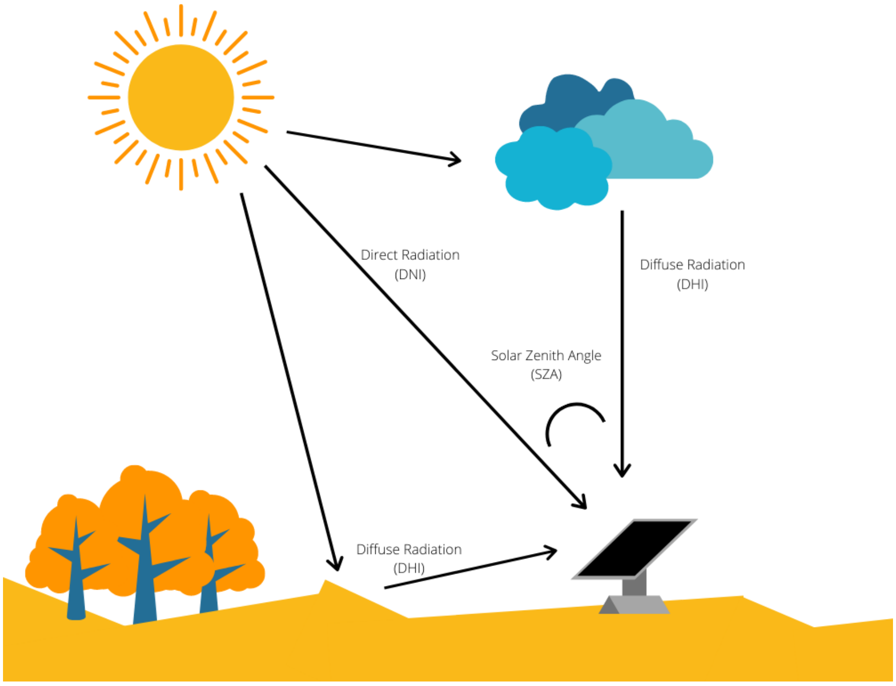


Figure A.1: Components of solar radiation.

Bibliography

[1] CISCO. *CISCO: Internet of Things At a Glance*. Tech. rep. June 2016. URL: https://www.cisco.com/c/dam/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf (visited on 02/12/2020) (cit. on p. 1).

[2] Amin Anjomshoaa, Simone Mora, Philip Schmitt, and Carlo Ratti. «Challenges of Drive-By IoT Sensing for Smart Cities: City Scanner Case Study». en. In: *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers - UbiComp '18*. Singapore, Singapore: ACM Press, 2018, pp. 1112–1120. ISBN: 978-1-4503-5966-5. DOI: 10.1145/3267305.3274167. URL: http://dl.acm.org/citation.cfm?doid=3267305.3274167 (visited on 02/12/2020) (cit. on pp. 1, 3).

[3] Amin Anjomshoaa, Fábio Duarte, Daniël Rennings, Thomas J. Matarazzo, Priyanka deSouza, and Carlo Ratti. «City Scanner: Building and Scheduling a Mobile Sensing Platform for Smart City Services». In: *IEEE Internet of Things Journal* 5.6 (Dec. 2018). Conference Name: IEEE Internet of Things Journal, pp. 4567–4579. ISSN: 2327-4662. DOI: 10.1109/JIOT.2018.2839058 (cit. on pp. 1, 3).

[4] Ankit Kekre and Suresh K. Gawre. «Solar photovoltaic remote monitoring system using IOT». In: *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*. ISSN: null. Oct. 2017, pp. 619–623. DOI: 10.1109/RISE.2017.8378227 (cit. on p. 2).

[5] Ming Wang, Ralf Birken, and Salar Shahini Shamsabadi. «Framework and implementation of a continuous network-wide health monitoring system for roadways». In: *Nondestructive Characterization for Composite*

Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security 2014. Vol. 9063. International Society for Optics and Photonics, Mar. 2014, 90630H. DOI: 10.1117/12.2047681. URL: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/9063/90630H/Framework-and-implementation-of-a-continuous-network-wide-health-monitoring/10.1117/12.2047681.short (visited on 02/12/2020) (cit. on p. 2).

[6] Laure Deville Cavellin, Scott Weichenthal, Ryan Tack, Martina S. Ragettli, Audrey Smargiassi, and Marianne Hatzopoulou. «Investigating the Use Of Portable Air Pollution Sensors to Capture the Spatial Variability Of Traffic-Related Air Pollution». In: *Environmental Science & Technology* 50.1 (Jan. 2016), pp. 313–320. ISSN: 0013-936X. DOI: 10.1021/acs.est.5b04235. URL: https://doi.org/10.1021/acs.est.5b04235 (visited on 02/12/2020) (cit. on p. 2).

[7] Grant R. McKercher, Jennifer A. Salmond, and Jennifer K. Vanos. «Characteristics and applications of small, portable gaseous air pollution monitors». eng. In: *Environmental Pollution (Barking, Essex: 1987)* 223 (Apr. 2017), pp. 102–110. ISSN: 1873-6424. DOI: 10.1016/j.envpol.2016.12.045 (cit. on p. 2).

[8] World Health Organization. *WHO | Ambient air pollution: Health impacts*. Library Catalog: www.who.int Publisher: World Health Organization. URL: http://www.who.int/airpollution/ambient/health-impacts/en/ (visited on 07/08/2020) (cit. on p. 2).

[9] Sotiris Vardoulakis, Norbert Gonzalez-Flesca, Bernard E. A. Fisher, and Koulis Pericleous. «Spatial variability of air pollution in the vicinity of a permanent monitoring station in central Paris». en. In: *Atmospheric Environment*. Fourth International Conference on Urban Air Quality: Measurement, Modelling and Management, 25-28 March 2003 39.15 (May 2005), pp. 2725–2736. ISSN: 1352-2310. DOI: 10.1016/j.atmosenv.2004.05.067. URL: http://www.sciencedirect.com/science/article/pii/S1352231005001743 (visited on 02/12/2020) (cit. on p. 2).

[10] Joshua S. Apte et al. «High-Resolution Air Pollution Mapping with Google Street View Cars: Exploiting Big Data». In: *Environmental Science & Technology* 51.12 (June 2017). Publisher: American Chemical Society, pp. 6999–7008. ISSN: 0013-936X. DOI: 10.1021/acs.est.

7b00891. URL: https://doi.org/10.1021/acs.est.7b00891 (visited on 07/08/2020) (cit. on p. 2).

[11] Adar Rosenfeld, Michael Dorman, Joel Schwartz, Victor Novack, Allan C. Just, and Itai Kloog. «Estimating daily minimum, maximum, and mean near surface air temperature using hybrid satellite models across Israel». eng. In: *Environmental Research* 159 (2017), pp. 297–312. ISSN: 1096-0953. DOI: 10.1016/j.envres.2017.08.017 (cit. on p. 2).

[12] Deepak Vasisht, Zerina Kapetanovic, Jongho Won, Xinxin Jin, Ranveer Chandra, Sudipta Sinha, Ashish Kapoor, Madhusudhan Sudarshan, and Sean Stratman. «FarmBeats: An IoT Platform for Data-Driven Agriculture». en. In: 2017, pp. 515–529. ISBN: 978-1-931971-37-9. URL: https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/vasisht (visited on 02/17/2020) (cit. on p. 5).

[13] Taiyang Wu, Fan Wu, Jean-Michel Redouté, and Mehmet Rasit Yuce. «An Autonomous Wireless Body Area Network Implementation Towards IoT Connected Healthcare Applications». In: *IEEE Access* 5 (2017), pp. 11413–11422. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2716344 (cit. on p. 5).

[14] Simone Mora, Amin Anjomshoaa, Tom Benson, Fabio Duarte, and Carlo Ratti. «Towards Large-scale Drive-by Sensing with Multi-purpose City Scanner Nodes». en. In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. Limerick, Ireland: IEEE, Apr. 2019, pp. 743–748. ISBN: 978-1-5386-4980-0. DOI: 10.1109/WF-IoT.2019.8767186. URL: https://ieeexplore.ieee.org/document/8767186/ (visited on 02/12/2020) (cit. on pp. 5, 12).

[15] Nashid Nabian and Prudence Robinson. *SENSEable CITY GUIDE*. English. 2011. URL: http://senseable.mit.edu/papers/pdf/20110412_Ratti_etal_SenseableCity_SAP.pdf (visited on 02/13/2020) (cit. on p. 7).

[16] Alexandre-Edmond Becquerel. «Memoire sur les effets electriques produits sous l'influence des rayons solaires». French. In: Comptes Rendus 9 (1839), pp. 561–567 (cit. on p. 9).

[17] Lewis Fraas. «Chapter 1: History of Solar Cell Development». In: June 2014. ISBN: 978-3-319-07529-7. DOI: 10.1007/978-3-319-07530-3_1 (cit. on p. 9).

[18] J. Perlin. *Silicon Solar Cell Turns 50*. English. Tech. rep. NREL/BR-520-33947. National Renewable Energy Lab., Golden, CO. (US), Aug. 2004. URL: `https://www.osti.gov/biblio/15009471` (visited on 02/21/2020) (cit. on p. 10).

[19] Lewis Fraas. *Low Cost Solar Electric Power*. June 2014. ISBN: 978-3-319-07529-7. DOI: `10.1007/978-3-319-07530-3` (cit. on p. 10).

[20] IEEE Photovoltaic Specialists Conference and Institute of Electrical and Electronics Engineers, eds. *The conference record of the eighteenth IEEE Photovoltaic Specialists Conference–1985: Las Vegas, Nevada, October 21-25, 1985*. English. OCLC: 13805157. New York, N.Y.: Institute of Electrical and Electronics Engineers, 1985 (cit. on p. 10).

[21] Fariba Besharat, Ali A. Dehghan, and Ahmad R. Faghih. «Empirical models for estimating global solar radiation: A review and case study». en. In: *Renewable and Sustainable Energy Reviews* 21 (May 2013), pp. 798–821. ISSN: 1364-0321. DOI: `10.1016/j.rser.2012.12.043`. URL: `http://www.sciencedirect.com/science/article/pii/S1364032112007484` (visited on 03/23/2020) (cit. on p. 13).

[22] Ozge Ayvazogluyuksel Erol and Ümmühan Başaran Filik. «Estimation of Monthly Average Hourly Global Solar Radiation from the Daily Value in Çanakkale, Turkey». In: *Journal of Clean Energy Technologies* 5 (Sept. 2017). DOI: `10.18178/jocet.2017.5.5.403` (cit. on p. 13).

[23] Davud Mostafa Tobnaghi, Rahim Madatov, and daryush naderi daryush. «The Effect of Temperature on Electrical Parameters of Solar Cells». In: 2.12 (Dec. 2013). ISSN: 2278-8875 (cit. on p. 13).

[24] Jussi Ekström, Matti Koivisto, John Millar, Ilkka Mellin, and Matti Lehtonen. «A statistical approach for hourly photovoltaic power generation modeling with generation locations without measured data». en. In: *Solar Energy* 132 (July 2016), pp. 173–187. ISSN: 0038-092X. DOI: `10.1016/j.solener.2016.02.055`. URL: `http://www.sciencedirect.com/science/article/pii/S0038092X1600164X` (visited on 03/24/2020) (cit. on p. 13).

[25] Hussein A Kazem, Miqdam Chaichan, Ali Al-Waeli, and Kavish Mani. «Effect of Shadows on the Performance of Solar Photovoltaic». In: *Mediterranean Green Buildings and Renewable Energy: Selected Papers from the World Renewable Energy Network's Med Green Forum*. Journal Abbreviation: Mediterranean Green Buildings and Renewable Energy:

Selected Papers from the World Renewable Energy Network's Med Green Forum. Dec. 2017, pp. 379–385. ISBN: 978-3-319-30745-9. DOI: 10.1007/978-3-319-30746-6_27 (cit. on p. 14).

[26] Shaharin Anwar Sulaiman, Atul Kumar Singh, Mior Maarof Mior Mokhtar, and Mohammed A. Bou-Rabee. «Influence of Dirt Accumulation on Performance of PV Panels». en. In: *Energy Procedia*. Technologies and Materials for Renewable Energy, Environment and Sustainability (TMREES14 – EUMISD) 50 (Jan. 2014), pp. 50–56. ISSN: 1876-6102. DOI: 10.1016/j.egypro.2014.06.006. URL: http://www.sciencedirect.com/science/article/pii/S1876610214007425 (visited on 03/24/2020) (cit. on p. 14).

[27] Hadja Maïmouna Diagne, Philippe Lauret, and Mathieu David. «Solar irradiation forecasting: state-of-the-art and proposition for future developments for small-scale insular grids». In: *WREF 2012 - World Renewable Energy Forum*. Denver, United States, May 2012. URL: https://hal.archives-ouvertes.fr/hal-00918150 (visited on 05/05/2020) (cit. on pp. 14, 17).

[28] Akinobu Murata, Hideaki Ohtake, and Takashi Oozeki. «Modeling of uncertainty of solar irradiance forecasts on numerical weather predictions with the estimation of multiple confidence intervals». en. In: *Renewable Energy* 117 (Mar. 2018), pp. 193–201. ISSN: 0960-1481. DOI: 10.1016/j.renene.2017.10.043. URL: http://www.sciencedirect.com/science/article/pii/S0960148117309813 (visited on 05/08/2020) (cit. on pp. 14, 17).

[29] S. Cros, O. Liandrat, N. Sébastien, and N. Schmutz. «Extracting cloud motion vectors from satellite images for solar power forecasting». In: *2014 IEEE Geoscience and Remote Sensing Symposium*. ISSN: 2153-7003. July 2014, pp. 4123–4126. DOI: 10.1109/IGARSS.2014.6947394 (cit. on pp. 14, 17).

[30] Chi Wai Chow, Bryan Urquhart, Matthew Lave, Anthony Dominguez, Jan Kleissl, Janet Shields, and Byron Washom. «Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed». en. In: *Solar Energy* 85.11 (Nov. 2011), pp. 2881–2893. ISSN: 0038-092X. DOI: 10.1016/j.solener.2011.08.025. URL: http://www.sciencedirect.com/science/article/pii/S0038092X11002982 (visited on 05/05/2020) (cit. on pp. 14, 17).

[31] Richard Perez, Sergey Kivalov, James Schlemmer, Karl Hemker, David Renné, and Thomas E. Hoff. «Validation of short and medium term operational solar radiation forecasts in the US». en. In: *Solar Energy* 84.12 (Dec. 2010), pp. 2161–2172. ISSN: 0038-092X. DOI: 10.1016/j.solener.2010.08.014. URL: http://www.sciencedirect.com/science/article/pii/S0038092X10002823 (visited on 05/06/2020) (cit. on pp. 15, 17).

[32] Terence C. Mills. «Chapter 3 - ARMA Models for Stationary Time Series». en. In: *Applied Time Series Analysis*. Ed. by Terence C. Mills. Academic Press, Jan. 2019, pp. 31–56. ISBN: 978-0-12-813117-6. DOI: 10.1016/B978-0-12-813117-6.00003-X. URL: http://www.sciencedirect.com/science/article/pii/B978012813117600003X (visited on 05/06/2020) (cit. on p. 15).

[33] Rui Huang, Tiana Huang, Rajit Gadh, and Na Li. «Solar generation prediction using the ARMA model in a laboratory-level microgrid». In: *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*. Nov. 2012, pp. 528–533. DOI: 10.1109/SmartGridComm.2012.6486039 (cit. on pp. 15, 17).

[34] Sharif Atique, Subrina Noureen, Vishwajit Roy, Vinitha Subburaj, Stephen Bayne, and Joshua Macfie. «Forecasting of total daily solar energy generation using ARIMA: A case study». In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. Jan. 2019, pp. 0114–0119. DOI: 10.1109/CCWC.2019.8666481 (cit. on p. 16).

[35] Christophe Paoli, Cyril Voyant, Marc Muselli, and Marie-Laure Nivet. «Forecasting of preprocessed daily solar radiation time series using neural networks». en. In: *Solar Energy* 84.12 (Dec. 2010), pp. 2146–2160. ISSN: 0038-092X. DOI: 10.1016/j.solener.2010.08.011. URL: http://www.sciencedirect.com/science/article/pii/S0038092X10002793 (visited on 05/06/2020) (cit. on pp. 16, 17).

[36] A. Alzahrani, J. W. Kimball, and C. Dagli. «Predicting Solar Irradiance Using Time Series Neural Networks». en. In: *Procedia Computer Science*. Complex Adaptive Systems Philadelphia, PA November 3-5, 2014 36 (Jan. 2014), pp. 623–628. ISSN: 1877-0509. DOI: 10.1016/j.procs.2014.09.065. URL: http://www.sciencedirect.com/science/article/pii/S1877050914013143 (visited on 05/06/2020) (cit. on pp. 16, 17).

[37] E. G. Kardakos, M. C. Alexiadis, S. I. Vagropoulos, C. K. Simoglou, P. N. Biskas, and A. G. Bakirtzis. «Application of time series and artificial neural network models in short-term forecasting of PV power generation». In: *2013 48th International Universities' Power Engineering Conference (UPEC)*. Sept. 2013, pp. 1–6. DOI: 10.1109/UPEC.2013.6714975 (cit. on p. 16).

[38] Thomas Huld, Richard Müller, and Attilio Gambardella. «A new solar radiation database for estimating PV performance in Europe and Africa». en. In: *Solar Energy* 86.6 (June 2012), pp. 1803–1815. ISSN: 0038-092X. DOI: 10.1016/j.solener.2012.03.006. URL: http://www.sciencedirect.com/science/article/pii/S0038092X12001119 (visited on 03/25/2020) (cit. on pp. 16, 17).

[39] Constantinos S. Psomopoulos, George Ch. Ioannidis, Stavros D. Kaminaris, Kostas D. Mardikis, and Nikolaos G. Katsikas. «A Comparative Evaluation of Photovoltaic Electricity Production Assessment Software (PVGIS, PVWatts and RETScreen)». en. In: *Environmental Processes* 2.1 (Nov. 2015), pp. 175–189. ISSN: 2198-7505. DOI: 10.1007/s40710-015-0092-4. URL: https://doi.org/10.1007/s40710-015-0092-4 (visited on 03/25/2020) (cit. on pp. 16, 17).

[40] Binfang Wang and Ching Eng Png. «Solar irradiance simulation for evaluating thermal comfort in urban environment». In: *Architectural Science Review* 62.1 (Jan. 2019). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00038628.2018.1535421, pp. 14–25. ISSN: 0003-8628. DOI: 10.1080/00038628.2018.1535421. URL: https://doi.org/10.1080/00038628.2018.1535421 (visited on 05/07/2020) (cit. on p. 17).

[41] Sofia Thorsson, Fredrik Lindberg, Jesper Björklund, Björn Holmer, and David Rayner. «Potential changes in outdoor thermal comfort conditions in Gothenburg, Sweden due to climate change: the influence of urban geometry». en. In: *International Journal of Climatology* 31.2 (2011). _eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/joc.2231, pp. 324–335. ISSN: 1097-0088. DOI: 10.1002/joc.2231. URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/joc.2231 (visited on 05/07/2020) (cit. on p. 17).

[42] Ha T. Nguyen and Joshua M. Pearce. «Incorporating shading losses in solar photovoltaic potential assessment at the municipal scale».

en. In: *Solar Energy* 86.5 (May 2012), pp. 1245–1260. ISSN: 0038-092X. DOI: 10.1016/j.solener.2012.01.017. URL: http://www.sciencedirect.com/science/article/pii/S0038092X12000333 (visited on 05/07/2020) (cit. on p. 17).

[43] Fang-Ying Gong, Zhao-Cheng Zeng, Fan Zhang, Xiaojiang Li, Edward Ng, and Leslie K. Norford. «Mapping sky, tree, and building view factors of street canyons in a high-density urban environment». en. In: *Building and Environment* 134 (Apr. 2018), pp. 155–167. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2018.02.042. URL: http://www.sciencedirect.com/science/article/pii/S0360132318301148 (visited on 05/09/2020) (cit. on pp. 17, 56).

[44] National Renewable Energy Laboratory. *Solar Position and Intensity (SOLPOS) calculator*. English. May 2020. URL: https://midcdmz.nrel.gov/solpos/solpos.html (visited on 05/07/2020) (cit. on p. 18).

[45] ESRI. *ArcMap*. English. May 2020. URL: https://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/an-overview-of-the-solar-radiation-tools.htm (visited on 05/07/2020) (cit. on p. 18).

[46] Paul Rich. «Characterizing Plant Canopies With Hemispherical Photographs». In: *Remote Sensing Reviews* 5 (Jan. 1990), pp. 13–29. DOI: 10.1080/02757259009532119 (cit. on p. 18).

[47] Paul Rich, W.A. Hetrick, and S.C. Saving. «Using viewshed models to calculate intercepted solar radiation: Applications in ecology». In: *Am. Soc. Photogram. Remote Sens. Tech. Pap.* 4 (Jan. 1994) (cit. on p. 18).

[48] Pinde Fu and Paul M Rich. «A geometric solar radiation model with applications in agriculture and forestry». en. In: *Computers and Electronics in Agriculture* 37.1 (Dec. 2002), pp. 25–35. ISSN: 0168-1699. DOI: 10.1016/S0168-1699(02)00115-1. URL: http://www.sciencedirect.com/science/article/pii/S0168169902001151 (visited on 05/07/2020) (cit. on p. 18).

[49] Kevin P. O'Keeffe, Amin Anjomshoaa, Steven H. Strogatz, Paolo Santi, and Carlo Ratti. «Quantifying the sensing power of vehicle fleets». en. In: *Proceedings of the National Academy of Sciences* 116.26 (June 2019), pp. 12752–12757. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.

1821667116. URL: https://www.pnas.org/content/116/26/12752 (visited on 02/12/2020) (cit. on p. 23).

[50] NREL. *Physical Solar Model (PSM)*. URL: https://developer.nrel.gov/docs/solar/nsrdb/psm3_data_download/ (visited on 06/16/2020) (cit. on p. 29).

[51] World Weather Online. *World Weather Online*. URL: https://www.worldweatheronline.com/developer/ (visited on 06/16/2020) (cit. on p. 30).

[52] EkapopeV. *ekapope/WorldWeatherOnline*. June 2020. URL: https://github.com/ekapope/WorldWeatherOnline (visited on 06/16/2020) (cit. on p. 30).

[53] Google. *Street View Service*. en. Library Catalog: developers.google.com. URL: https://developers.google.com/maps/documentation/javascript/streetview (visited on 06/16/2020) (cit. on pp. 30, 52).

[54] NREL. *SOLPOS Calculator*. URL: https://midcdmz.nrel.gov/solpos/solpos.html (visited on 06/17/2020) (cit. on pp. 31, 60).

[55] Soteris A. Kalogirou. «Chapter 2 - Environmental Characteristics». en. In: *Solar Energy Engineering (Second Edition)*. Ed. by Soteris A. Kalogirou. Boston: Academic Press, Jan. 2014, pp. 51–123. ISBN: 978-0-12-397270-5. DOI: 10.1016/B978-0-12-397270-5.00002-9. URL: http://www.sciencedirect.com/science/article/pii/B9780123972705000029 (visited on 06/17/2020) (cit. on p. 35).

[56] Tianqi Chen and Carlos Guestrin. «XGBoost: A Scalable Tree Boosting System». In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Aug. 2016). arXiv: 1603.02754, pp. 785–794. DOI: 10.1145/2939672.2939785. URL: http://arxiv.org/abs/1603.02754 (visited on 06/20/2020) (cit. on p. 44).

[57] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. «LightGBM: A Highly Efficient Gradient Boosting Decision Tree». In: *NIPS*. 2017 (cit. on p. 46).

[58] Saad Parvaiz DURRANI, Stefan BALLUFF, Lukas WURZER, and Stefan KRAUTER. «Photovoltaic yield prediction using an irradiance forecast model based on multiple neural networks». en. In: *Journal of Modern Power Systems and Clean Energy* 6.2 (Mar. 2018), pp. 255–267. ISSN: 2196-5420. DOI: `10.1007/s40565-018-0393-5`. URL: `https://doi.org/10.1007/s40565-018-0393-5` (visited on 05/20/2020) (cit. on pp. 47, 48).

[59] Diederik P. Kingma and Jimmy Ba. «Adam: A Method for Stochastic Optimization». In: *arXiv:1412.6980 [cs]* (Jan. 2017). arXiv: 1412.6980. URL: `http://arxiv.org/abs/1412.6980` (visited on 06/23/2020) (cit. on p. 48).

[60] Matthew D. Zeiler. «ADADELTA: An Adaptive Learning Rate Method». In: *arXiv:1212.5701 [cs]* (Dec. 2012). arXiv: 1212.5701. URL: `http://arxiv.org/abs/1212.5701` (visited on 06/23/2020) (cit. on p. 48).

[61] Sergey Ioffe and Christian Szegedy. «Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift». In: *arXiv:1502.03167 [cs]* (Mar. 2015). arXiv: 1502.03167. URL: `http://arxiv.org/abs/1502.03167` (visited on 06/23/2020) (cit. on p. 48).

[62] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. «Dropout: A Simple Way to Prevent Neural Networks from Overfitting». In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: `http://jmlr.org/papers/v15/srivastava14a.html` (cit. on p. 49).

[63] Fang-Ying Gong, Zhao-Cheng Zeng, Edward Ng, and Leslie K. Norford. «Spatiotemporal patterns of street-level solar radiation estimated using Google Street View in a high-density urban environment». en. In: *Building and Environment* 148 (Jan. 2019), pp. 547–566. ISSN: 0360-1323. DOI: `10.1016/j.buildenv.2018.10.025`. URL: `http://www.sciencedirect.com/science/article/pii/S0360132318306437` (visited on 05/06/2020) (cit. on p. 53).

[64] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. «Pyramid Scene Parsing Network». In: *arXiv:1612.01105 [cs]* (Apr. 2017). arXiv: 1612.01105. URL: `http://arxiv.org/abs/1612.01105` (visited on 06/25/2020) (cit. on p. 53).

[65] Cityscapes Dataset. *Cityscapes Dataset.* en-US. URL: https://www.cityscapes-dataset.com/news/ (visited on 06/25/2020) (cit. on p. 53).

[66] VladKry. *Vladkryvoruchko/PSPNet-Keras-tensorflow.* June 2020. URL: https://github.com/Vladkryvoruchko/PSPNet-Keras-tensorflow (visited on 06/26/2020) (cit. on p. 53).

[67] Xiaojiang Li, Carlo Ratti, and Ian Seiferling. «Quantifying the shade provision of street trees in urban landscape: A case study in Boston, USA, using Google Street View». en. In: *Landscape and Urban Planning* 169 (Jan. 2018), pp. 81–91. ISSN: 0169-2046. DOI: 10.1016/j.landurbplan.2017.08.011. URL: http://www.sciencedirect.com/science/article/pii/S0169204617301950 (visited on 05/10/2020) (cit. on pp. 55, 57, 58).

[68] D. Comaniciu and P. Meer. «Mean shift: a robust approach toward feature space analysis». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.5 (May 2002). Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 603–619. ISSN: 1939-3539. DOI: 10.1109/34.1000236 (cit. on p. 57).

[69] N. Otsu. «A threshold selection method from gray-level histograms». In: *Automatica* 11.285-296 (1975), pp. 23–27 (cit. on p. 58).

[70] D. G. Erbs, S. A. Klein, and J. A. Duffie. «Estimation of the diffuse radiation fraction for hourly, daily and monthly-average global radiation». en. In: *Solar Energy* 28.4 (Jan. 1982), pp. 293–302. ISSN: 0038-092X. DOI: 10.1016/0038-092X(82)90302-4. URL: http://www.sciencedirect.com/science/article/pii/0038092X82903024 (visited on 06/27/2020) (cit. on p. 61).

[71] github. *Build software better, together.* en. Library Catalog: github.com. URL: https://github.com (visited on 07/13/2020) (cit. on p. 72).

[72] Lorenzo Santolini. *modusV/pygsvpano.* original-date: 2020-07-13T16:24:16Z. July 2020. URL: https://github.com/modusV/pygsvpano (visited on 07/15/2020) (cit. on p. 72).

[73] Lorenzo Santolini. *modusV/fos.* original-date: 2020-07-11T15:18:12Z. July 2020. URL: https://github.com/modusV/fos (visited on 07/15/2020) (cit. on p. 73).

[74] Lorenzo Santolini. *modusV/skia*. original-date: 2020-07-12T08:56:45Z. July 2020. URL: `https://github.com/modusV/skia` (visited on 07/15/2020) (cit. on p. 73).

[75] Ali H. A. Al-Waeli, Hussein A. Kazem, Miqdam Tariq Chaichan, and Kamaruzzaman Sopian. *Photovoltaic/Thermal (PV/T) Systems: Principles, Design, and Applications*. en. Springer International Publishing, 2019. ISBN: 978-3-030-27823-6. DOI: `10.1007/978-3-030-27824-3`. URL: `https://www.springer.com/gp/book/9783030278236` (visited on 07/16/2020) (cit. on p. 75).