FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Improving Solar Power Forecasting Through Advanced Feature Engineering

Rui Manuel Gonçalves do Couto

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Dr. Ricardo Bessa

Second Supervisor: Dr. Vladimiro Miranda

23 July 2020

Resumo

A redução drástica dos preços dos painéis fotovoltaicos que ocorreu na década passada permitiu que centrais fotovoltaicas fossem instaladas em massa, levando a que esta fonte ocupasse uma maior fatia da produção de energia elétrica.

Este tipo de energia apresenta baixo impacto ambiental, no entanto, a sua alta variabilidade e incerteza intrínsecas desafiam os agentes de mercado e operadores do sistema. De forma a ajudá-los a combater os dois fatores negativos apresentados são realizadas previsões de potência solar a cada instante definido. Estas previsões são, normalmente, acompanhadas por um determinado erro, que se pretende que seja o mais baixo possível a fim de otimizar as decisões com impacto futuro. Assim, vários estudos têm vindo a ser desenvolvidos nesta área com o intuito de maximizar a qualidade das previsões fornecidas. Apesar dos grandes avanços conseguidos, ainda há margem para melhoria e, por isso, é apresentado neste projeto uma nova abordagem ao problema.

Nesta tese, apresentam-se diversos modelos de previsão de potência solar com horizonte temporal de um dia para uma central fotovoltaica num cenário real. Estes modelos são desenvolvidos através de *deep learning* que extraem automaticamente *features* a partir de imagens baseadas numa grelha de pontos de *Numerical Weather Predictions*. Para determinar o desempenho dos mesmos, são concebidas redes que requerem a criação manual de *features*. Todos os modelos são sujeitos a uma seleção, a fim de comparar apenas os melhores de cada tipo numa posterior fase de avaliação. O desempenho geral e o desempenho em determinadas condições climatéricas de cada um dos referidos são avaliados de acordo com os valores de um conjunto de métricas aplicadas às previsões fornecidas pelos mesmos.

Abstract

The drastic reduction in the prices of photovoltaic panels that occurred in the past decade allowed photovoltaic plants to be installed in large scale, making this source occupy a greater share of the production of electricity.

This type of energy has a low environmental impact, however, its high intrinsic variability and uncertainty challenge market agents and system operators. In order to help them tackle the two negative factors presented, solar power forecasts are produced for specific timestamps. These predictions usually contain errors, which are intended to be as low as possible to optimize decisions with a future impact. Thus, several studies have been developed in this area in order to maximize the quality of the provided forecasts. Despite the great advances achieved, there is still room for improvement and, therefore, a new approach to the problem is presented in this project.

In this thesis, several models of day-ahead solar power forecasting for a photovoltaic plant in a real scenario are presented. These models are developed through deep learning that automatically extract features from images based on a grid of Numerical Weather Predictions points. To assess their performance, networks that require manual feature creation are designed. All of the created models are subject to a selection, in order to compare only the best of each type in a later evaluation phase. The global performance and the performance in certain climatic conditions of the selected networks are evaluated according to the values retrieved by a set of metrics applied to the forecasts which are provided by these models.

Acknowledgements

First of all, I would like to thank my thesis supervisor, Dr. Ricardo Bessa, for the opportunity and continuous support provided even after hours. This project, undoubtedly, could not be done without his ideas and vision. I also would like to thank Prof. Dr. Vladimiro Miranda for the help and motivation granted. A special word must be addressed to Ricardo Andrade for his assistance provided in the code development part and suggested ideas.

I am very thankful to all my closest friends for backing me in some tough moments and helping me forget all the work I had to do for some brief moments.

Finally, I must express my deepest gratitude to my parents, my sister, my brother-in-law and my niece for the given opportunities and support throughout my life. To the rest of my closest family, I would like to express my heartfelt thanks for following me and believing in me during my entire life.

Thank you.

Rui Couto

“No one wants to learn by mistakes,
but we cannot learn enough from successes to go beyond the state of the art.”

Henry Petroski

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Objectives and Contributions 3
1.3 Organisation of the Thesis 4

2 State of art **5**
2.1 Introduction 5
2.2 Solar Power Forecasting: Information Versus Time Horizon 5
2.3 Clear Sky-models and Naive Approaches 6
2.4 Statistical Methods for Solar Power Forecasting 7
2.5 The Importance of Feature Engineering 8
2.5.1 Manual Feature Creation 9
2.5.2 Automatic Feature Creation 10
2.6 Approaches for Automatic Feature Creation 10
2.6.1 Convolutional Neural Networks 11
2.6.2 Long Short-Term Memory Networks 12
2.6.3 Convolutional Long Short-Term Memory Networks 12
2.7 Conclusion 13

3 Methodology **15**
3.1 Data Description 15
3.2 Data Pre-processing 16
3.2.1 Manual Feature Creation 16
3.2.2 Image Creation 19
3.3 Neural Network Structures 20
3.3.1 Experimental Data Setup 21
3.3.2 Dense Neural Networks 22
3.3.3 Convolutional Neural Networks 23
3.3.4 Convolutional LSTM Neural Networks 26

4 Results **33**
4.1 Evaluation of Forecasting Accuracy 33
4.2 Hyperparameter Tuning 34
4.3 Comparison Between Forecasts Provided by Each Model 35
4.3.1 Overall Benchmark 36
4.3.2 Clear Sky Hours Performance 37
4.3.3 Partly Cloudy Sky Hours Performance 41
4.3.4 Cloudy Sky Hours Performance 44

4.4 Hardware Specifications 48
4.5 Final Remarks 49

5 Conclusions and Future Work **51**
5.1 Conclusions 51
5.2 Future Work 52

A Hyperparameter Tuning Results **55**

References **59**

List of Figures

1.1 Evolution of the cumulative installed capacity of PV solar in Portugal [1]. 2
1.2 Evolution of the cumulative installed capacity of PV solar in EU-28 [2, 3, 4, 5, 6, 7, 8]. 3

2.1 Time horizon and applications for each type of solar power forecasting [9]. . . . 6
2.2 Simplified diagram that represents the operations performed by convolution layers (extracted from [10]). 11
2.3 Simplified diagram of a LSTM cell, where small rectangles and circular shapes represent layers and pointwise operations, respectively. 12

3.1 Geographic location of the points of NWP grid. 17
3.2 Diagram that represents the spatial layers in different colours. 19
3.3 Evolution of $swflx$ from 5 a.m. to 7 p.m. on 1 May 2013. 20
3.4 Time diagram of the sets for each fold, where grey, yellow and blue shapes correspond to training, validation and test sets, respectively. 21
3.5 DNN1 and DNN2 structures. 22
3.6 Example of the transformations performed by Conv2D and MaxPooling2D in an image related to $swflx$ on 2 May 2014 at 1 a.m.. 24
3.7 CNN-I1 and CNN-IT1 structures. 26
3.8 CNN-I2 and CNN-IT2 structures. 27
3.9 CNN-I3 and CNN-IT3 structures. 28
3.10 Example of the transformations performed by ConvLSTM2D in images related to $swflx$ on 2 May 2014 from 10 a.m. to 1 p.m.. 29
3.11 CLSTMNN1 and CLSTMNN2 structures. 30

4.1 Time diagram of the sets for each fold, where grey and blue shapes correspond to training and test sets, respectively. 35
4.2 Graphs of forecasts retrieved by the proposed models versus the measured power in clear sky days, numbered according to the fold where they belong. 38
4.3 Graphs of forecasts retrieved by the proposed models versus the measured power in partly cloudy sky days, numbered according to the fold where they belong. . . 44
4.4 Graphs of forecasts retrieved by the proposed models versus the measured power in cloudy sky days, numbered according to the fold where they belong. 45

A.1 DNN2 hyperparameters related to the network structure and activation functions. 55
A.2 CNN-I2 hyperparameters related to the network structure and activation functions. 56
A.3 CNN-IT3 hyperparameters related to the network structure and activation functions. 56
A.4 CLSTMNN2 hyperparameters related to the network structure and activation functions. 57

List of Tables

3.1 Weather information provided by NWP data set for each point. 16
3.2 Transformations to be applied to the some NWP data set variables. 18
3.3 Time period of each test data fold. 21
3.4 Months that compose the validation set of each data fold. 21
3.5 Input variables divided by their type. 23

4.1 Values of RMSE for each fold and model. 36
4.2 Values of MAPE for each fold and model. 36
4.3 Values of MBE for each fold and model. 36
4.4 Metrics of each model for 3 May 2014. 37
4.5 Metrics of each model for 17 February 2015. 39
4.6 Metrics of each model for 7 March 2015. 39
4.7 Metrics of each model for 1 October 2015. 39
4.8 Metrics of each model for 21 June 2016. 40
4.9 Values of RMSE for each fold and model in clear sky hours. 40
4.10 Values of MAPE for each fold and model in clear sky hours. 40
4.11 Values of MBE for each fold and model in clear sky hours. 41
4.12 Metrics of each model for 20 August 2014. 41
4.13 Metrics of each model for 25 October 2014. 42
4.14 Metrics of each model for 23 April 2015. 42
4.15 Metrics of each model for 5 August 2015. 42
4.16 Metrics of each model for 13 March 2016. 43
4.17 Values of RMSE for each fold and model in partly cloudy sky hours. 43
4.18 Values of MAPE for each fold and model in partly cloudy sky hours. 43
4.19 Values of MBE for each fold and model in partly cloudy sky hours. 43
4.20 Metrics of each model for 6 June 2014. 46
4.21 Metrics of each model for 22 February 2015. 46
4.22 Metrics of each model for 3 May 2015. 46
4.23 Metrics of each model for 4 October 2015. 47
4.24 Metrics of each model for 2 April 2016. 47
4.25 Values of RMSE for each fold and model in cloudy sky hours. 47
4.26 Values of MAPE for each fold and model in cloudy sky hours. 48
4.27 Values of MBE for each fold and model in cloudy sky hours. 48
4.28 Average runtime of hyperparameter tuning and training of each model per fold. . 48

A.1 Model DNN2 hyperparameter sets selected for each data fold. 57
A.2 Model CNN-I2 hyperparameter sets selected for each data fold. 57
A.3 Model CNN-IT3 hyperparameter sets selected for each data fold. 58

A.4 Model CLSTMNN2 hyperparameter sets selected for each data fold. 58

Abbreviations

PV	Photovoltaic
EU-28	European Union (with United Kingdom included)
ANN	Artificial Neural Network
AR	Autoregressive
ARMA	Autoregressive Moving Average
NWP	Numerical Weather Predictions
MOS	Model Output Statistics
GHI	Global Horizontal Irradiance
ARIMA	Autoregressive Integrated Moving Average
SARIMA	Seasonal Autoregressive Integrated Moving Average
PCA	Principal Component Analysis
MLP	Multilayer Perceptron
LSTM	Long Short-Term Memory
DBN	Deep Belief Network
CNN	Convolutional Neural Network
1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
ConvLSTM	Convolutional Long Short-Term Memory
MBE	Mean Bias Error
MAE	Mean Average Error
MAPE	Mean Absolute Percentage Error
RMSE	Root Mean Square Error
THREDDS	Thematic Real-time Environmental Distributed Data Services
UTC	Coordinated Universal Time
API	Application Programming Interface
img	Images
4D	Four-Dimensional
5D	Five-Dimensional
CPU	Central Process Unit
GPU	Graphics Processing Unit
CUDA	Compute Unified Device Architecture
NOAA	National Oceanic and Atmospheric Administration
ReLU	Rectified Linear Unit
tanh	Hyperbolic Tangent
ELU	Exponential Linear Unit

Chapter 1

Introduction

The present chapter aims to provide the reader an overview regarding the motivation that supports the theme of the thesis. Additionally, this chapter intends to show to the reader the main objectives of the dissertation and the organisation of this document, describing each chapter briefly.

1.1 Motivation

Lately, the electrical power systems have been subject of various changes in order to improve not only its reliability and quality of service provided, but also reduce its environmental impact. Nowadays, electric power production is one of the main sectors that emit the most carbon dioxide into the atmosphere, worsening the effects of climate change. In order to reverse the situation, several countries and companies have been investing in alternative sources of energy that are more environmentally friendly, such as wind and solar power. While the installed wind capacity has increased in a very significant way over the last 20 years, the installed solar capacity has only started to expand at a similar pace to the mentioned one since approximately 2010. In the coming years, the installed solar capacity is expected to surpass installed wind capacity, because several countries are running out of economically viable places to install wind turbines and there has been a very steep decrease in the price of PV panels. In the figures 1.1 and 1.2 are represented the growth of the cumulative installed capacity of PV in Portugal and in EU-28, respectively. According to the National Energy and Climate Plan presented by the Government of Portugal, the cumulative installed capacity of PV solar, in Portugal, is expected to reach 6.6 GWp by 2025 and 9.0 GWp by 2030 [11]. In EU-28, this capacity can achieve values between 180.1 GWp and 276.8 GWp by 2023 [12].

This progressive penetration of solar power generation means that conventional sources, such as thermal power stations, have an increasingly diminished role in the electrical power systems. This event contributes to a greater difficulty in the management of the system and energy markets, because unlike the production of energy from traditional sources, the production of solar energy cannot be increased when it is desired and is dependent on external factors. Additionally, the generation of solar energy is highly variable, uncertain and difficult to predict with high accuracy.

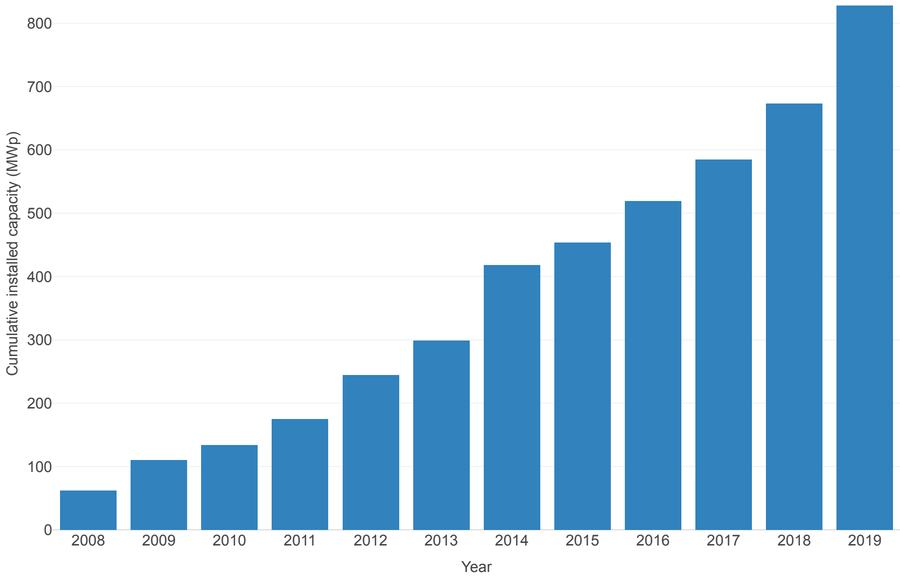


Figure 1.1: Evolution of the cumulative installed capacity of PV solar in Portugal [1].

Solar power forecasting is an important asset for market agents, solar power plant operators and system operators. These predictions optimise bids in the energy market and, consequently control the costs associated with balancing. They can help PV power plant operators scheduling maintenance in their facilities, minimizing potential economical losses. System operators need to receive accurate solar power predictions, in order to increase the system stability that is challenged by the inherent variability of solar power production. These forecasts can help determine operation reserve requirements, optimise production planning, establish strategies to guarantee grid security in any situation, avoid local voltage disruptions and minimise curtailment of solar power. The errors present in the forecasts can affect negatively the decisions taken by their users, both in economic and security terms.

Several studies conducted in the previous years showed a considerable error reduction by applying algorithms that had never been applied in the sector. However, new methods keep being discovered, namely in the field of deep learning, which can further reduce the errors provided by these predictions. Nowadays, deep learning is also used in several sectors such as: healthcare industry to detect tumors from clinical images as studied in [13] and farm industry to detect and count the number of trees in an orchard using high-resolution remote sensing images as proposed in [14].

The evolution of GPU over the last decade allowed them to become powerful computing accelerators for high-performance computing. GPU accelerators usually reduce significantly the computation time required to solve complex practical problems [15]. Therefore, the use of GPU in solar power forecasting allowed the researchers to design more sophisticated models without compromising the runtime and, consequently, delivering better results than classical approaches.

Most of the algorithms proposed by previous studies related to solar power forecasting rely on

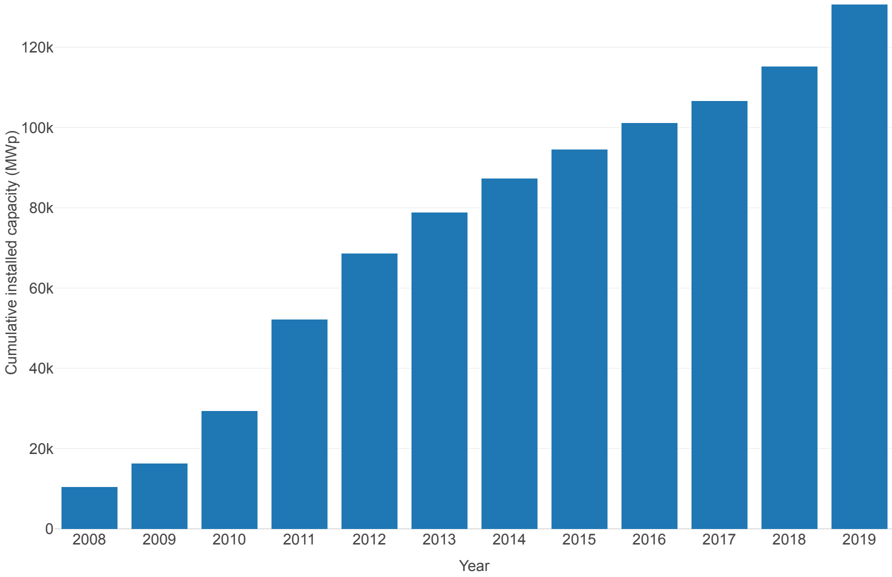


Figure 1.2: Evolution of the cumulative installed capacity of PV solar in EU-28 [2, 3, 4, 5, 6, 7, 8].

inputs that need to be manually pre-processed, known as manual features. This procedure is time consuming and, additionally, does no not allow the generalisation for the desired application. The remaining studies in the area were achieved using features generated by the models themselves, called automatic features. These models have the capability to learn from the input which one is best set of features and, consequently reduce the time spent by the researchers and, also, allow the generalisation for the intended use.

1.2 Objectives and Contributions

As mentioned before, solar power forecasts play an important role in electrical power systems with high solar power penetration. Their accuracy can be crucial to power system operators taking decisions, hence there is a need to do further studies in order to improve their errors.

In this dissertation, it is intended to develop a set of deep learning models capable of predicting solar power for a determined temporal horizon with a similar or even better performance than the ones used in the industry. The differentiating factor of this project is the automatic extraction of features from a grid of NWP points. This approach allows the researchers to spend more time optimizing the network instead of the inputs, since the models themselves define which features are the most crucial ones. The developed work will accomplish the following objectives:

- Development of models that predict solar power using automatic features, by testing and comparing different networks with capability to generate distinct features;
- Evaluation of the automatic features in solar power forecasting, by benchmarking the predictions delivered by the different proposed networks;

- Definition of a set of procedures that allow the replication of the models developed in this dissertation in other PV plants using the same type of inputs.

1.3 Organisation of the Thesis

The present section intends to explain the organisation of the five chapters that compose the thesis. The current chapter introduces the need for further studies in the area of solar power forecasting and contains the objectives that this work will achieve.

Chapter 2, which is followed by the introduction, delivers an overview about solar power forecasting and feature engineering. In the first part of this chapter, the solar power forecasts are classified and some of the available methods are referred. The importance of feature engineering is described in the second part as well as some of the studies conducted in this area that used manual features and automatic features are mentioned along with their results. In the last part three possible approaches to obtain automatic features are briefly explained.

In the chapter 3, the methodology followed in this work to obtain the models, that are later analysed, is explained. Firstly, in order to help the reader understand the used information in the models, the available data sets are described. Then, the methods used to pre-process the input data are defined and the models developed are compared according to each type.

The next chapter addresses the hardware specifications of the server where the models are tested and the hyperparameter tuning of the best models developed in the chapter 3. After this tuning, the models are subject to several comparisons, in order to understand the value that automatic features can bring to solar power forecasts.

Lastly, chapter 5 presents the conclusions reached throughout the work developed and possible methods to improve the performance of the forecasts provided by the models designed.

Chapter 2

State of art

2.1 Introduction

The production of solar energy is undoubtedly dependent on weather conditions and the time of day in which it occurs. Therefore, this source of energy is extremely variable and difficult to predict with high accuracy. There are several ways to forecast the power output of a PV plant and they can be classified according to their time horizon as it will be seen in this chapter.

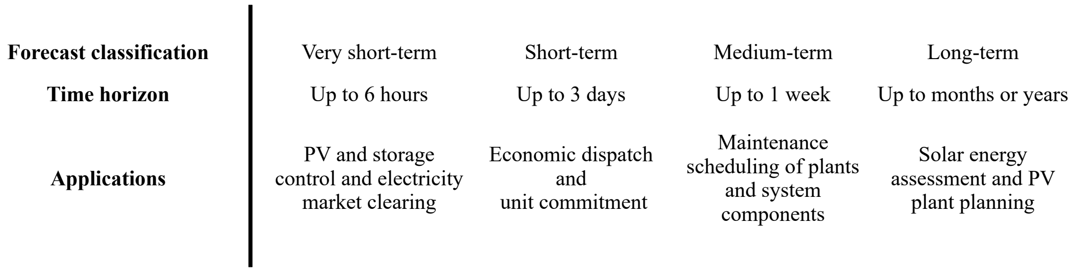
The most advanced prediction techniques rely on machine learning algorithms, whose inputs need to be pre-processed in order to obtain better forecasts. This pre-processing involves the creation of new variables, known as feature engineering, and respective selection. This part will have a special focus in the current chapter, since it plays an important role in deep learning networks and is one of the main motives for this project. The variables can either be generated manually or automatically, however both methods have their own upsides and downsides that need to be considered.

2.2 Solar Power Forecasting: Information Versus Time Horizon

Forecasting solar power involves a variety of methods based on the time frame being forecasted, the data available to the forecaster and the usage of the forecast [16]. Several factors influence its forecasted power output including, but not restricted to the measurement of solar irradiance, reflectivity, estimation of PV cell temperatures and efficiency of the inverter [9]. Solar power predictions can be classified by its time horizon: very-short term, short term, medium term and long term. Each type of prediction can provide additional information for power system operations and electricity markets as it can be seen in 2.1 [17, 9].

Very short-term solar power forecasts have a timescale from 5 minutes up to 6 hours and can be based on:

- Statistical models, such as AR, ARMA and ANN, with online measurements as input;
- Cloud motion vectors derived from the sequential satellite cloud images for the temporal range from 30 minutes up to 6 hours;



Forecast classification	Very short-term	Short-term	Medium-term	Long-term
Time horizon	Up to 6 hours	Up to 3 days	Up to 1 week	Up to months or years
Applications	PV and storage control and electricity market clearing	Economic dispatch and unit commitment	Maintenance scheduling of plants and system components	Solar energy assessment and PV plant planning

Figure 2.1: Time horizon and applications for each type of solar power forecasting [9].

- Cloud information from ground-based sky images for the sub hour range [18];
- Machine learning techniques combined with NWP data [19].

Recently, stochastic learning techniques, which attempt to remove the modeller's opinion in the optimisation of the the topology, received attention, since they can be highly competitive in accuracy and relatively simple to setup. The best approaches use measured data and NWP models [20].

In short-term horizon, NWP models are combined with statistical methods for better solar predictions. In cloudy conditions, the errors are higher, but in clear sky conditions are lower. The NWP models displays large positive errors which require a model output statistics approach. The NWP models conjugated with MOS correction method based on clear sky index and solar zenith angle provides an important base line prediction accuracy to evaluate other forecasting models [21].

For medium-term forecasts, it is possible to use information from NWP models, since they can provide data for up to 6 days-ahead [22] and statistical models with processed data [21].

In the long-term forecasting process, the neural network toolbox is used, in order to predict the future GHI. The historical and other weather data are fed to the model as input whereas the actual data is fed as target [23].

2.3 Clear Sky-models and Naive Approaches

The clear-sky model originally included water vapor and Rayleigh scatter, but gradually added ozone and aerosols [24]. Due to the diversity of atmospheric processes involved and their spatio-temporal variability, modelling solar radiation incident at the surface is a difficult task. The general irradiance modelling challenge is typically simplified by considering the following independent parts:

- Modelling under presumed cloudless conditions;
- Modelling of the superimposed effect of clouds, if present.

The all-sky irradiance is, therefore, assumed a direct function of its clear-sky counterpart, even when the sky is totally covered by clouds [25]. Such value can be used obtain the production of a solar plant under stationary conditions. Typically, meteorological variables and solar geometry feed clear sky models, using Radiative Transfer Models to establish the connections between the inputs [26].

Naive approaches can be considered as the reference models, in order to evaluate the performance of other forecast models. The development of prediction models tries to focus more on achieving better forecast accuracy instead of reducing the computational cost [27]. This model can be used to create forecasts for minutes to hours-ahead time scales or less frequently forecasts day-ahead restricted to clear sky conditions and its logic is to assume that the forecasted conditions are similar to the current. There are several variations of naive-based modelling in solar radiation predictions:

- Assuming that the forecasted and current values are equal;
- Assuming that the forecasted values are based on the current, but adjusted by the stage of diurnal cycle;
- Using trends of change;
- Assuming long-term persistence of climatological means.

The results obtained from this model are affected by the information source, because data acquired at the site provides better estimates than values derived from NWP [28].

2.4 Statistical Methods for Solar Power Forecasting

Statistical techniques extract relations of historical influential variables, in order to forecast its future behaviour [27, 26]. These models do not need any internal data from the system and its accuracy depends on the quality and quantity of historical meteorological and power measurements [26]. Statistical methods can be separated into two distinct subgroups: regressive methods and artificial intelligence techniques.

Regressive methods estimate the relationship between solar power output and independent variables, by finding a mathematical function that correlates them. According to how time series are treated, a further classification appears: linear or non-linear and stationary and non-stationary. A time-series is classified as stationary if it fluctuates around a static mean and non-stationary if that mean it is not shown. The regressive models used in solar power forecasting are:

- Linear stationary models: where is included AR model, Moving Average models, ARMA models, AR exogenous models and ARMA models with exogenous variables;
- Linear non-stationary models: ARIMA models and SARIMA models are incorporated in this category;
- Non-linear stationary models: composed by Non-linear AR-exogenous models [26].

Artificial intelligence systems learn from example are able to deal with non-linear problems and noisy and/or incomplete data (which is common in meteorological and power measurements). Once they are trained, they can forecast and generalise at high speed [29]. The main AI models applied in this field are: Artificial Neural Networks, k-Nearest Neighbours, Support Vector Machines and Random Forests. It will be done a further review on ANN, since they are the most adopted machine learning models in solar power prediction and the project is related to them.

Artificial Neural Networks are effective tools since they can correlate highly nonlinear behaviour and they are inspired in the neuron operation, where a group of neurons are interconnected to form a neural network [26]. These models have predicted solar irradiance with success for different time horizons (from intra-hour to years) and display good performance for both data-rich/poor environments [30]. To evaluate the performance of ANN, several studies were done such as the ones conducted by:

- Pedro *et al.*, where it is shown that ANN outperform Persistence, the ARIMA and the k-NN models on 1 hour and 2 hours-ahead solar power forecasting [31];
- Kardakos *et al.*, where it was concluded that ANN provide better day-ahead predictions than SARIMA and Persistence and similar to SARIMA with exogenous factor [32].

2.5 The Importance of Feature Engineering

Feature engineering, also called feature construction, is often described as the process of extracting new representations from the raw data that are more beneficial to a computing task [33]. These new representations are obtained via a single or set of mathematical operations between part of the original data set and provide another perspective of the data to the machine learning model as it can be seen in [34] and [35].

Feature construction plays a crucial role in many machine learning models, because their performance strongly depends on having insightful input representations that uncover the fundamental explanatory components of the output for the observed input [33]. Therefore, it can enhance the overall result and, also, help build simpler models that are simultaneously easier to understand and faster to run.

This process creates features that can be classified as:

- Relevant - shows strong influence on the output;
- Irrelevant - exhibits no effect on the output;
- Redundant - displays no impact when another feature that captures the same functionality is present.

Therefore, it is important to perform feature selection, which consists in the stage where only the relevant features are chosen to serve as the model's input. Although feature selection is mainly performed to enhance the model's performance, it can also decrease the operating time of a learning process and help build simpler models [36].

2.5.1 Manual Feature Creation

Manual feature creation requires humans to build new representations of the original data and, then select the relevant features. Both processes are time consuming and error prone. Although, these processes, usually, are possible to do in small data sets, larger data sets require a more complex feature representation, which is challenging for humans to realise based on domain knowledge alone [37].

Another issue that manual feature construction faces is not having capability to generalise, at least for each application. This can be seen in different PV plants, since the ideal set of features discovered for one, might not be even good for another, due to the different location where they are installed, which can make some factors more important in one and irrelevant in the other.

Chow *et al.* proposed an ANN that performs day-ahead solar power predictions based on a database with NWP for the PV plant site. They carried out data min-max normalisation and concluded that the most relevant variables to forecast the daily power output are: day of the month, the mean daily relative humidity, the mean daily wind speed, the mean daily solar irradiance and the daily air temperature [38].

Wang *et al.* created an ANN network capable of predicting solar irradiance with a time scale of 24 hours based on measurements taken on the previous day of the forecast. This model receives 5 inputs and retrieves 24 variables corresponding to the solar irradiance of each hour for the next day. From a data set with irradiance and temperature measurements, they extracted the following features: daily average surface irradiance, daily average temperature, normalised discrete difference of solar irradiance, maximum value of the third order derivative of solar irradiance difference and day of the year [39].

In the Global Energy Forecasting Competition 2014, participants studied probabilistic forecasting for hourly solar and wind production. For the task of solar prediction, they had available information regarding 12 independent NWP variables, 4 of which contained accumulated values. Gábor *et al.* decided to extract regular variables from the accumulated ones and time-related information from the given timestamp (hour in two different formats, day of the year and month) [40].

Andrade and Bessa suggested another method to day-ahead solar power forecasts, but with higher time resolution (1 hour). They had access to a database with a grid of NWP and engineered new variables based on: temporal variance, lags and leads, spatial PCA, spatial standard deviation and temporal and spatial means. Several approaches were tested according to the type of extracted features, it was concluded that the combination of both types of features resulted on the best performance [34].

The model developed by Andrade and Bessa is expected to achieve better results than the remaining aforementioned ones, because it uses both temporal and spatial features. However, this model certainly took longer to develop, as they had to discover spatial relationships with the output.

2.5.2 Automatic Feature Creation

The feature extraction stage is typically performed manually, additionally, depending on the type of data, it is possible to face a wide range of methods to extract features. In this sense, the process to create and select appropriate techniques, normally, takes a long time, so several methods were created to overcome this issue [41]. These methods automatically discover the best representations to perform a determined forecast, removing the need of feature selection.

Gensler *et al.* tested the performance between MLP, LSTM, DBN and LSTM Autoencoder networks for solar power forecasts with time horizon of 24 hours to 48 hours. The features of the first two models were obtained from manual feature engineering and the remaining automatically from the models themselves. It was concluded that models with inherent capability of feature extraction (in this case temporal features) led to better forecasts than the others [42].

Recent studies showed promising results when using CNN as a mean to learn and extract automatically from time series data. These types of neural networks predict future values, since it can learn filters that represent repeated patterns in the data. A closer analysis to CNN will be done in 2.6.1. Koprinska *et al.* transposed this knowledge to day-ahead solar power forecasting and adopted 1D-CNN and LSTM to extract temporal features by using arrays with hourly data of each day. Persistence and MLP models which require manual feature extraction were also tested to benchmark the performance. It was concluded that 1D-CNN under these conditions outperformed the other models Koprinska2018.

Higashiyama *et al.* suggested another approach, but now applied to wind power forecasting with time horizon between 30 minutes to 48 hours, which used 3D-CNN to automatically extract features from a grid of NPW data. As referred in the article, 2D-CNN and 3D-CNN are able to extract nonlinear intrinsic spatial features and spatio-temporal features from the input, respectively. In order to benchmark the performance of features extracted by 3D-CNN, the following methods were also studied to extract features: Naive approaches (direct use of NWP data), Principal Component Analysis (where there a dimensionality reduction of NWP data) and 2D-CNN. It was proven that 3D-CNN improve the accuracy of wind power forecasts, especially for a time horizon greater than 20 hours, which shows that it can be an effective tools to extract features from NWP data [43].

Since solar power data sets are organised in time-series, it is always possible to use 1D-CNN and LSTM to extract temporal features, however, to extract spatial and spatio-temporal features using 2D-CNN and 3D-CNN, respectively, it is necessary to have access to a grid of measurements. 3D-CNN models are expected to have better performance than the remaining aforementioned models, because they can extract features that others cannot.

2.6 Approaches for Automatic Feature Creation

Since the project will address automatic feature creation, it is essential to understand the underlying mechanisms of the structures that can automatically extract features. Therefore, in the following subsections, it will be explained how these structures work.

2.6.1 Convolutional Neural Networks

Convolutional neural networks, which are one of the most widely used categories of neural networks, especially for high-dimensional data, operate in a very similar manner to standard neural networks [44]. CNN use a specialised kind of linear operation named convolution instead of general matrix multiplication in at least one of their layers [10].

A CNN is composed by several basic building blocks that implement basic functionalities such as normalization, pooling, convolution, and fully connected layers. Convolutional layers contain a set of filters (also named convolutional kernels) that generate an output feature map after convolving with the input, as it can be seen in 2.2. In order to obtain the output feature map, the filter takes a determined number of steps, previously defined by the user, along horizontal and vertical position [44]. After this operation, some redundancies were stored in the feature map, since adjacent activities are usually related to each other. The use of a pooling layer after a convolutional layer may be beneficial, because it can filter the most prominent features [33].

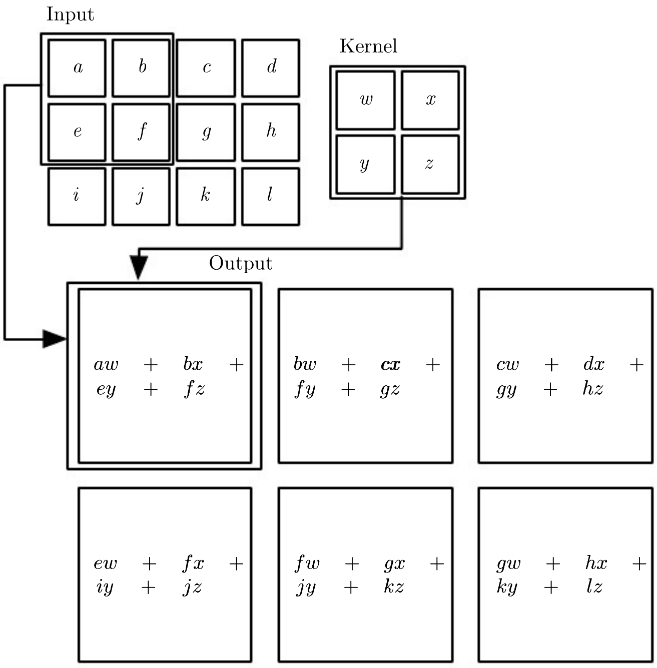


Figure 2.2: Simplified diagram that represents the operations performed by convolution layers (extracted from [10]).

In a standard neural network, every output unit interacts with every input unit, since its layers use matrix multiplication by a matrix of parameters. CNN have sparse interactions when the kernel is smaller than the input, reducing the number of parameters to store. Thereby, improving its statistical efficiency and decreasing the number of required operations [10]. Convolution takes advantage of three fundamental ideas that can help enhance machine learning systems: sparse interactions, parameter sharing and equivariant representations. Such operation allows to work with inputs with variable shapes [10].

A network has parameter sharing when the weight applied to one input is related to the weight applied elsewhere. In a CNN, such feature is verified, because each member of the kernel is used at every position of the input except some of the boundary pixels. With parameter sharing instead of learning a separate set of parameters for every location, it is learned only one set, which reduces even more storage requirements [10].

The particular form of parameter sharing causes the layer to be equivariant to translation, which means that a change in the input would cause a change in the same way in the output. The equivariance to translation makes it possible to appear the same output representation after moving an event later in time in the input [10].

2.6.2 Long Short-Term Memory Networks

A LSTM network is composed by a set of recurrently connected subnets called memory blocks. These blocks contain one or more self-connected memory cells and three adaptive multiplicative gating units (input, output and forget gates) as it can be seen in 2.3. These gates can be trained to learn what information to store for a determined period of time and when to read it out [45].

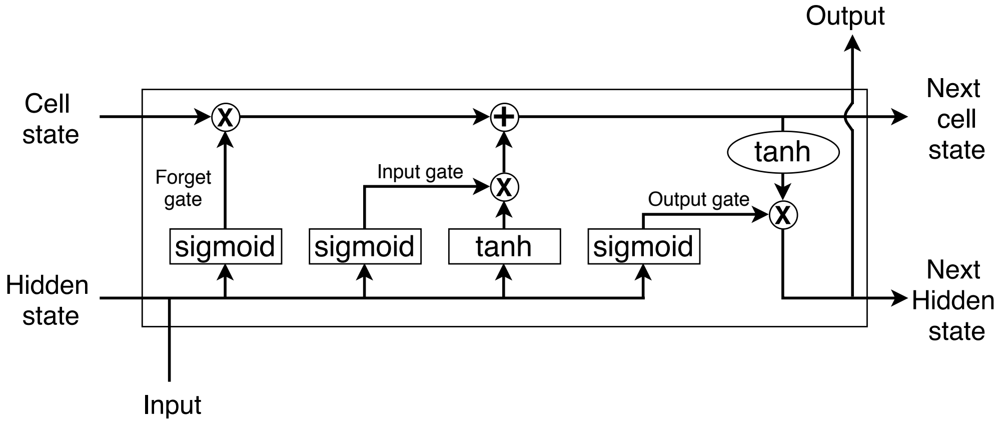


Figure 2.3: Simplified diagram of a LSTM cell, where small rectangles and circular shapes represent layers and pointwise operations, respectively.

LSTM can generate a list of state representations during composition, however, as it is shown in 2.3, the next state is dependent from the current state. It is assumed that LSTM maintain unbounded memory [46].

2.6.3 Convolutional Long Short-Term Memory Networks

A Convolutional Long Short-Term Memory network, also known as ConvLSTM network, is an extent of LSTM network which has convolutional structures in the following transitions: input-to-state and state-to-state. By stacking multiple ConvLSTM layers and creating an encodingforecasting structure, it is possible to build a model that can handle spatio-temporal sequence problems.

ConvLSTM networks, unlike standard LSTM networks, can handle spatial data, because its inputs, cell outputs, hidden states and gates are 3D tensors, whose two last dimensions are spatial dimensions. The ConvLSTM predicts the future state of a determined cell in the grid based on inputs and previous states of its local neighbours. If states are seen as hidden representation of moving objects, faster motions should be captured by ConvLSTM layer with larger transitional kernel, while slower motions by one with smaller kernel [47].

2.7 Conclusion

Solar power forecasts can be classified according to its time horizon, and each type has different applications that can be important for electricity markets and system maintenance. NWP data has an essential role in these predictions, since PV power output depends on the weather conditions at the time of production.

Statistical methods show good performance for predictions with different time horizons, while persistence models are only suitable for very short-term forecasts. Different studies prove that ANN can outperform time-series models at several time horizons, since these models cannot capture some solar data features.

The performance of machine learning models highly depends on the input representation, so it is necessary to create new features from the raw data set, and then select the relevant ones to improve the input's correlation with the output. This process can be done manually or automatically. Manual feature engineering can lead machine learning models to good results, however it is time consuming, challenging for humans to do based on domain knowledge alone and incapable to generalise to each application. Automatic feature engineering requires only code implementation (which brings time savings) and can generalise to each application, because it learns the relations between the inputs and the output. Additionally, this process can create features that cannot be generated manually, originating better forecast results in some papers, on one hand, and, on the other, making the features more difficult to humans to understand.

Using CNN, LSTM and ConvLSTM networks are some of the ways to extract automatically features related to solar power forecasting. While CNN and LSTM networks are able to extract only spatial or temporal and temporal features, respectively, ConvLSTM network can extract both at the same time, leading to better predictions.

Chapter 3

Methodology

In this chapter, the procedures to choose the most adequate models to explore in the chapter 4 are explained. The data sets available are analysed to help the reader understand in which context they surge. As reviewed in the chapter 2, the inputs of each model must be properly pre-processed in order to ensure better forecasting performance. Therefore, the set of operations applied to each variable is further explored. Several structures are studied for each type of input, however, only the ones that show the most promising forecasts are selected for an additional review.

3.1 Data Description

The solar power data set is composed by hourly power measurements of a PV power plant, located in the city of Porto in Portugal, from 28 April 2013 to 28 June 2016. This plant has installed capacity of 16.32 kW and is divided by the following arrays: 3 × 2160 W, 2 × 3360 W and 1 × 3120 W.

The use NWP data is fundamental to achieve good results for short-term solar power forecasting, therefore a data set constituted by a grid of NWP is used. This grid covers an approximate area of 2400 km^2 and composed by 169 uniformly distributed points in a 13 × 13 spatial configuration, as it is possible to see in 3.1. Each point is separated from its closest neighbours by a 4 km distance. This NWP data set consists in 1014 variables, corresponding to the six categories present in the table 3.1 per point.

The NWP data were obtained from the MeteoGalicia THREDDS server, which provides publicly historical and daily forecasts with a time resolution of 1 hour. It has capability to generate new climate forecasts every 24 hours at 12 a.m. UTC with a time horizon up to 96 hours. Each variable at each hour present in this data set is represented by 4 forecasts, one generated on the current day at 12 a.m. UTC and remaining on the previous days.

Both solar power measurements and NWP data sets are publicly available in [48].

Table 3.1: Weather information provided by NWP data set for each point.

Category	Meaning
$swflx$ [W/m^2]	Surface downwelling shortwave flux
$temp$ [K]	Ambient temperature at 2 meters
cfl [0, 1]	Cloud cover at low levels
cfm [0, 1]	Cloud cover at mid levels
cfm [0, 1]	Cloud cover at high levels
cft [0, 1]	Cloud cover at low and mid levels

3.2 Data Pre-processing

Data pre-processing plays a decisive role in the machine learning algorithms' performance, as the collected data related to power and meteorological measurements have often errors and missing values. Their performance also depend on how the data is represented as it was seen in 2.5.

For this study case, it was found missing data in some periods of time in the solar power data set, which do not participate in the predictions performed in the following chapters. In the NWP data set, some values related to cloud cover variables were discovered to be higher than the established limits mentioned in 3.1, so they were replaced by the upper limit value.

3.2.1 Manual Feature Creation

As shown in 2.5.1, there are several ways to represent the original data to minimise the prediction error. One way is to normalise the input data, as several machine learning algorithms cannot correctly correlate data with different scales. There are several ways to normalise data and the most common in the field of solar power forecast are: min-max normalisation and clear sky normalisation. While the first mentioned normalisation is done by calculating values via a simple formula (that is represented in 3.1) to the values, the second one requires a clear sky-model (which was reviewed in the section 2.3) to predict the an intermediary value that is used in 3.2 [49].

$$X' = \frac{X - X_{min}}{X_{max} - X_{max}} \tag{3.1}$$

$$X' = \frac{X}{X_{pred}} \tag{3.2}$$

Where X, X', X_{min}, X_{max} and X_{pred} mean value before normalisation, value after normalisation, minimum value that X can be, maximum value that X can be and value predicted by the clear sky model, respectively.

Since a grid of NWP is at disposal, it is possible to extract both temporal and spatial features. Andrade and Bessa, as referred in 2.5.1, discovered the combination of both types of features leads to better solar power forecasts [34]. The features extracted are based on the ones suggested

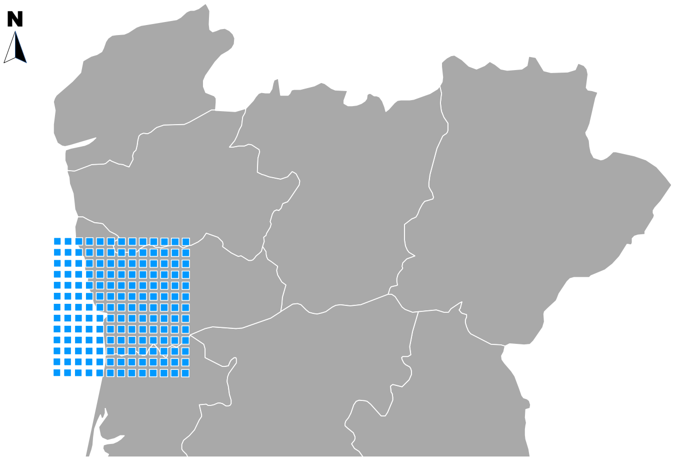


Figure 3.1: Geographic location of the points of NWP grid.

by Andrade and Bessa. In table 3.2, it is possible to see the transformations that some variables of the NWP data set will be subject.

For the temporal variance, it is considered a moving window of 3, 7, and 11 hours centred in a determined lead-time $t+k$. Therefore, it is possible to create three new variables for each one present in the NWP data set, by calculating the variance according to the formula 3.3 for the following intervals of time: $[t+k-1, t+k+1]$h, $[t+k-3, t+k+3]$h and $[t+k-5, t+k+5]$h.

$$\sigma^2_{time}(t+k) \quad = \quad \frac{\sum_{i=t+k-(N_h-1)/2}^{t+k+(N_h-1)/2}(x(i)-\bar{x})^2}{N_h} \tag{3.3}$$

Where $x(i)$ and $\bar{x}$ correspond to the value of a variable at a specific time i and the mean value that the variable has for the considered window, respectively. The temporal variance is only calculated for $swflx$ at the central point of the NWP grid, because Andrade and Bessa showed that this type of variance is correlated to power variability periods.

The temporal dependency of a NWP variable can expressed by adding its prior and subsequent values. The inclusion of lags and leads in the study conducted by Nagy *et al.* have shown an increase of predictions' accuracy [40].

There is a high similarity between the 4 forecasts provided by the NWP data set in clear days predictions, however, this similarity decreases in the remaining days. Therefore, to include the information given by each forecast in the model without excessively increase its number of inputs, it is calculated the mean between the 4 forecasts.

$$\bar{x}_{past}(t+k) \quad = \quad \frac{\sum_{d=1}^{N_{run}} x_d(t+k)}{N_{run}} \tag{3.4}$$

Table 3.2: Transformations to be applied to the some NWP data set variables.

Domain	Transformations
Temporal	Variance
	Lags and leads
	Mean
Spatial	PCA
	Standard deviation
	Mean

Where N_{run} and x_d mean number of forecast runs available at $t+k$ and value of the variable predicted by the forecast run d.

PCA is a method to reduce the dimensionality of data, exposing the most important patterns and dynamics of the original data. This is implemented by dividing all grid variables according to their type, normalising each value and then using sklearn.decomposition. PCA from the Scikit-learn library [50]. Andrade and Bessa concluded that the best overall results were achieved applying PCA to $swflx$, cfl, cfm and cft and choosing their respective number of principal components that assured 90% variance threshold.

Andrade and Bessa found a meaningful correlation between the dispersion of $swflx$ NWP grid values and distinct power variability time periods. So, to include this information in the prediction model they suggested computing individually a hourly standard deviation index (3.5) for each type of variable available in the NWP data set [34].

$$\sigma_{spatial}(t+k) = \sqrt{\frac{\sum_{i=1}^{169}(x_i(t+k)-\bar{x})^2}{169}} \tag{3.5}$$

Where x_i and $\bar{x}$ correspond to the value of a variable at point i and mean value of all points at $t+k$.

Spatial standard deviation index may not describe well enough the magnitude of power variations, leading to over/underestimations. Therefore, a new variable called Spatial Smoothing was created to smooth the $swflx$ forecasts for the central location based on surrounding predictions for the entire grid. This feature is generated by computing the hourly mean value of each layer represented in 3.2 and then the mean of the preciously calculated values (3.6).

$$\bar{x}_{spatial} = \frac{\sum_{i=1}^{6}\bar{x}_i(t+k)}{6} \tag{3.6}$$

Where $\bar{x}_i$ corresponds to the mean value of the layer i.

The features aforementioned use only variables given by the latest weather forecast run with the exception of temporal mean that use variables provided by every run.

It is possible to obtain non-weather features from the NWP data set. Every weather forecast has an associated timestamp, where time-related features can be extracted, such as hour and month.

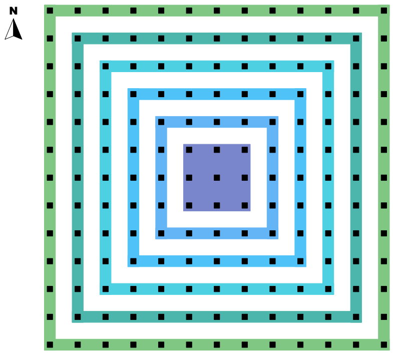


Figure 3.2: Diagram that represents the spatial layers in different colours.

These variables add information related to the sun's position and can be expressed via a min-max normalisation or by the expressions 3.7, 3.8, 3.9 and 3.10.

$$hour_{sin}(n) = sin\left(\frac{\pi \times n}{23}\right) \tag{3.7}$$

$$hour_{cos}(n) = cos\left(\frac{\pi \times n}{23}\right) \tag{3.8}$$

$$month_{sin}(n) = sin\left(\frac{\pi \times n}{12}\right) \tag{3.9}$$

$$month_{cos}(n) = cos\left(\frac{\pi \times n}{12}\right) \tag{3.10}$$

Where n corresponds to the hour or month when the forecast occurs.

3.2.2 Image Creation

As there are 6 types of variables described by a square NWP grid, it is possible to create 6 types of square matrices. These matrices represent the weather conditions at a determined hour for the space defined by the NWP grid.

To create these matrices it is necessary to, first, normalise the data, according to their type and then, assign the value of each grid point to its corresponding matrix element. Each element corresponds to the value of the NWP point with similar spatial location to the element (e.g. the first element of the matrix, which is located in the top right position, corresponds to the northwest point). Therefore, each matrix is be 13 × 13.

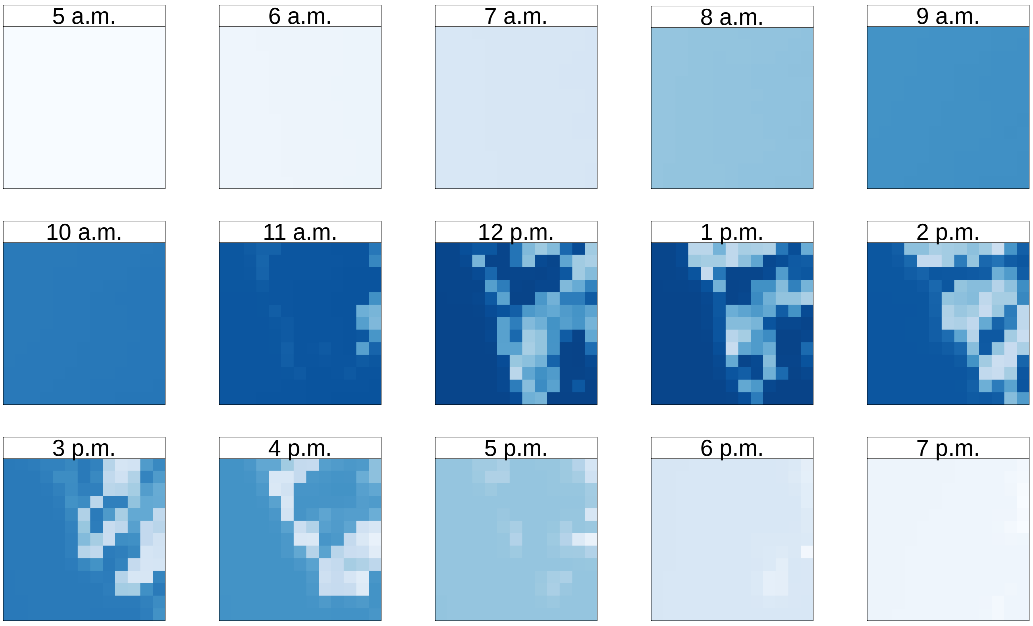


Figure 3.3: Evolution of $swflx$ from 5 a.m. to 7 p.m. on 1 May 2013.

These matrices can be seen as monochromatic images, because each element, which has only one value, can correspond to a pixel. The values of the variables used to create these images are the ones retrieved by the run closest to the moment of prediction, because in theory this run provides greater accuracy than the remaining which contain higher levels of uncertainty. In the figure 3.3, the evolution of $swflx$ for a specific period of time can be seen with a different scale of colours to help understand the concept, where lightest and darkest colours correspond to lowest and highest values of $swflx$, respectively. Thus, during the daytime the lighter the pixels, the more clouds or possibly fog there will be at the corresponding point on the NWP grid.

3.3 Neural Network Structures

The neural network structures are implemented via Keras [51], which is TensorFlow's high-level API for building and training deep learning models. The models, that are designed, receive the following set of inputs:

- Numerical variables;
- Images;
- Images and numerical variables.

Where the numerical variables refer to manual features created. Different structures are tested for each set of inputs and the ones that provide best results will be further reviewed in the chapter 4.

The developed models predict the power generated by the PV solar plant for the hours corresponding to the numerical variables and/or images created that feed the neuronal networks.

3.3.1 Experimental Data Setup

In order to evaluate the overall performance of every model, it is created 5 data folds. Each data fold is divided into the following sets: training, validation and test. In the table 3.3, it is represented the time periods of each test set.

Table 3.3: Time period of each test data fold.

Fold	Time period
1	01/05/2014 - 30/09/2014
2	01/10/2014 - 28/02/2015
3	01/03/2015 - 31/07/2015
4	01/08/2015 - 21/12/2015
5	01/01/2016 - 28/06/2016

The models are trained using the training sets and their selection will be done according to the error retrieved by each structure using validation set as input. The use of test data is exclusive to a final comparison between models that will be performed in the chapter 4. Each data fold has a validation set that is composed by 6 months that are mentioned in the table 3.4.

Table 3.4: Months that compose the validation set of each data fold.

Fold	Months
1	July 2013, January 2014, April 2014, September 2015, November 2015 and March 2016
2	July 2013, January 2014, April 2014, September 2015, November 2015 and March 2016
3	July 2013, January 2014, April 2014, September 2015, November 2015 and March 2016
4	July 2013, January 2014, April 2014, September 2014, December 2014 and March 2016
5	July 2013, January 2014, April 2014, March 2015, September 2015 and November 2015

The validations sets were chosen from the training sets to represent both cloudy and clear sky months. In the figure 3.4, the temporal distributions of the sets are represented for each fold.

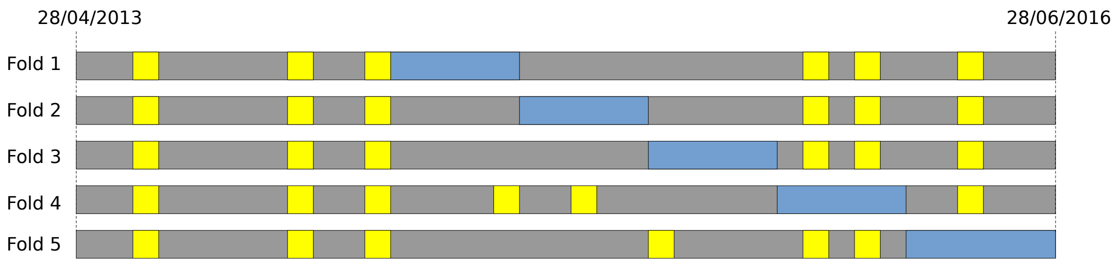


Figure 3.4: Time diagram of the sets for each fold, where grey, yellow and blue shapes correspond to training, validation and test sets, respectively.

3.3.2 Dense Neural Networks

These networks are composed by dense layers which are a regular densely-connected neural network layers. These networks predict the PV power plant output from the features generated previously. Since their input is 2D, the intrinsic dense layers will receive as well a 2D matrix, where the first and the second dimensions correspond to the batch size and input dimension, respectively. These layers generate the output by applying an activation function to the sum of the bias array to the scalar product between the input matrix and the generated weights matrix. The output of these layers can be described by the formula 3.11 and is 2D, whose dimensions match the batch size and the units.

$$output = activation(input \cdot kernel + bias) \tag{3.11}$$

Where *activation*, *kernel* and *bias* correspond to the element-wise activation function, the weights matrix and the bias vector, respectively.

It was created two networks, DNN1 and DNN2, which are represented in the figure 3.5. The only structural difference between both is that DNN2 has a hidden layer while DNN1 does not contain any. The main purpose of comparing these networks is to understand the value that hidden layers can add to the forecasts.

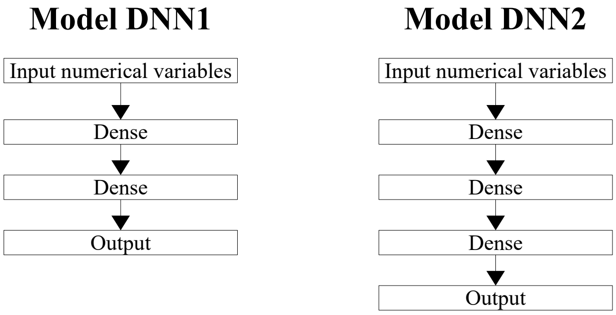


Figure 3.5: DNN1 and DNN2 structures.

Before performing this benchmark, it is necessary to define the hyperparameters of the networks and determine which is the most suitable set of inputs to this type of networks. The hyperparameters introduced in DNN1 and DNN2 are based on the approaches proposed by Moolayil [52] and preliminary tests conducted. The number of neurons in the first dense layer is chosen according to the closest number to double of inputs in the power of 2 and its weights and bias are initialised to random uniform and zero, respectively. Both networks are trained with:

- Batch size of 32;
- Adam as optimiser;

- Learning rate of 0.001.

After many experiments using the network DNN1 with different combinations of the features referred in the subsection 3.2.1, it was concluded that the best set of inputs is the one described in the table 3.5, since it provided the lowest average RMSE forecasts.

Table 3.5: Input variables divided by their type.

Type	Variables
Chronological	$hour_{sin}$ and $month_{sin}$
Temporal	Temporal variance of $swflx$ for 3, 7 and 11 hours
	Lags of 1h, 2h and 3h of $swflx$
	Temporal mean of $swflx$
Spatial	Spatial PCA of cfl, cfm and cft
	Spatial standard deviation of $swflx$ and cfl
	Spatial mean of $swflx$
Others	$swflx$ and $temp$ at the central position provided by the latest weather forecast run

Variables extracted from temperature and cloud data are min-max normalised. However, power measurements and variables related to $swflx$ can either be normalised by clear sky models or minimum-maximum values. Therefore, to choose the combination of normalisations that best fits the structure, tests with the same parameters were conducted using training and validation sets. From the experiments performed, it was concluded that power measurements and $swflx$ should be normalised according to clear sky models.

Using the same inputs, the models DNN1 and DNN2 were tested and it was verified that DNN2 had better performance than DNN1, as DNN2 provided forecasts with lower average RMSE. Other models with more hidden layers were analysed, but they didn't bring any improvements to the final results.

3.3.3 Convolutional Neural Networks

The convolutional neural networks, that will be analysed, are composed by several types of layers, such as 2D convolutional layers. These layers can be implemented in Keras using the Conv2D class, which convolves the input matrix with a generated convolution kernel and then applies an activation function to the result added by a bias vector, producing an output that feeds the next layer. This sequence of operations can be converted to the mathematical expression present in the formula 3.12.

$$output = activation(input * kernel + bias) \tag{3.12}$$

Where $activation$, $kernel$ and $bias$ correspond to the element-wise activation function, the weights matrix and the bias vector, respectively. The input of the Conv2D is a matrix with 4 dimensions that correspond to the batch size, rows, columns and channels. The produced output is also 4D, however, its dimensions match the batch size, new rows, new columns and filters. The number of new rows and new columns follow the formula 3.13 for square images with no padding.

$$L_{out} = \frac{L_{in} - K}{s} + 1 \tag{3.13}$$

Where L_{out}, L_{in}, K and s correspond to the number of output columns/rows, number of input columns/rows, kernel size and number of strides, respectively.

Convolutional layers are usually followed by pooling layers that downsample the output of convolutional layers by summarising the presence of features of the feature map. There are several types of pooling layers in Keras, however, the ones that are compatible with the output provided Conv2D are: MaxPooling2D, AveragePooling2D, GlobalMaxPooling2D and GlobalAveragePooling2D. MaxPooling2D and AveragePooling2D take the maximum or the average value over a defined window, depending on its type, and return a 4D matrix where its dimensions correspond to batch size, pooled rows, pooled columns and channels. The number of pooled rows and pooled columns is given by the formula 3.13 for square inputs. GlobalMaxPooling2D and GlobalAveragePooling2D present similar operation to the aforementioned ones, however, they take the maximum or average value of the entire feature map, downsampling it into a single value.

While dense neural networks rely on manual features created to predict solar power forecasting, Conv2D layers generate features for the convolutional neural networks from the images previously generated. The convolutional layers apply filters to the input images, changing the way these images are interpreted by the CNN, as it can be seen in the figure 3.6, where lightest and darkest colours correspond to lowest and highest pixel values, respectively.

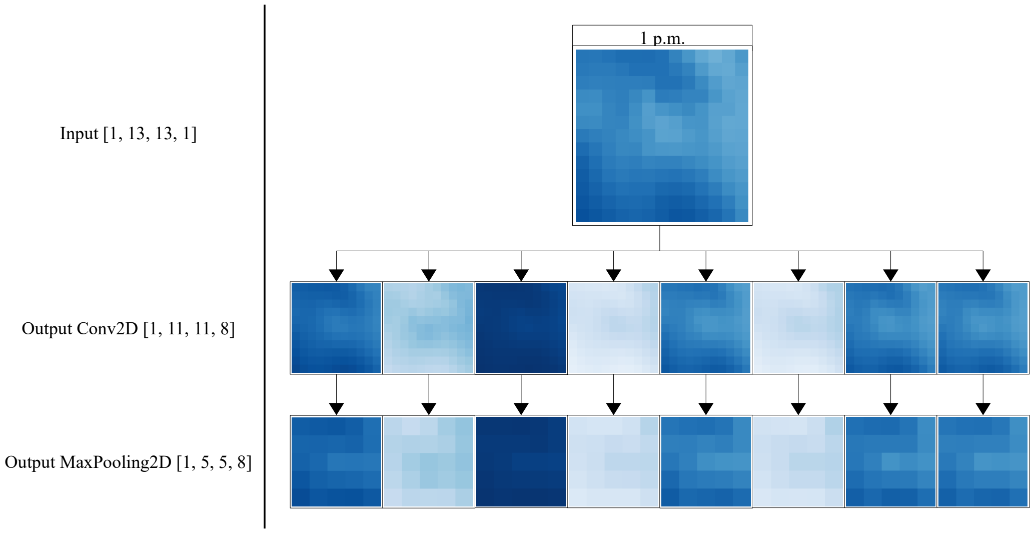

1 p.m.
Input [1, 13, 13, 1]
Output Conv2D [1, 11, 11, 8]
Output MaxPooling2D [1, 5, 5, 8]

Figure 3.6: Example of the transformations performed by Conv2D and MaxPooling2D in an image related to $swflx$ on 2 May 2014 at 1 a.m..

As the training progresses, the filters adapt in order to ensure the minimum loss, therefore, allowing to find the features that contribute the most to the performance of the network. The convolutional layers can extract spatial features of each type of variables present in the NWP data set, since the pixels present in the images are organised by geographic location as stated in the subsection 3.2.2 and, additionally, these layers can learn relations from each pixel's neighbours. The use of pooling layers after convolutional layers extracts the relevant features created during the convolution process, because as the kernel passes more than once through the same pixel, it creates redundant information.

This method should provide better features than the spatial ones created manually, since the model learns itself which are the most relevant and finds others that otherwise would be impossible due to the intricate connections of neural networks. These networks can also receive manual features as input, however, they only implemented after Conv2D and pooling layers.

Two different groups of CNN are analysed: one that only uses images as input (CNN-Ix) and the other receives the same images as the previous one, temporal features mentioned in the table 3.5, $hour_{cos}$ and $month_{cos}$ (CNN-ITx). It will be created three pairs of models in which each one is composed by networks from different groups with the same upper structure. The hyperparameters chosen for each model are based again on [52]. The number of filters in Conv2D selected is low, since these types of networks deal with simple images. Preliminary experiments conducted in the networks showed that they have better performance when: Conv2D have a reduced number of filters, since these types of networks deal with simple images, and they are trained with a batch size much higher than DNN1 and DNN2, such as 128. Additionally, from the same experiments it was concluded that smaller kernel size in Conv2D and reduced pool size in MaxPooling2D and AveragePooling2D improved the accuracy of the networks.

The first pair of networks, that is constituted by CNN-I1 and CNN-IT1, concatenates input images per hour and then applies a sequence of Conv2D and MaxPooling2D, extracting the features of the images with different variables together. These networks are represented in the figure 3.7. The input images of every CNN model were chosen based on tests performed on a model with a similar structure to CNN-I1. These tests showed that the most adequate set of images is obtained by $swflx$, cfl, cfm, cfh and cft variables from the NWP data set. Additionally, NWP variables should receive a min-max normalisation and power measurements a clear sky normalisation. In the second set of networks, which are composed by CNN-I2 and CNN-IT2 described in the figure 3.8, each type of type of images passes through a different sequence of Conv2D and MaxPooling2D and then they are concatenated. This method extracts features from each input independently. The models CNN-I3 and CNN-IT3, that belong in the third pair of networks and are characterised in the figure 3.9, have similar upper structures to the CNN-I2 and CNN-IT2, however, after the concatenation of feature maps, the output goes through the layers Conv2D and MaxPooling2D. This procedure provides a summary of all the features discovered by the upper layers.

Several extents to the proposed models were tested, some of which included adding more convolutional layers, removing pooling layers, replacing max pooling layers to average pooling layers and adding more dense layers. However, none of these approaches resulted in better performance.

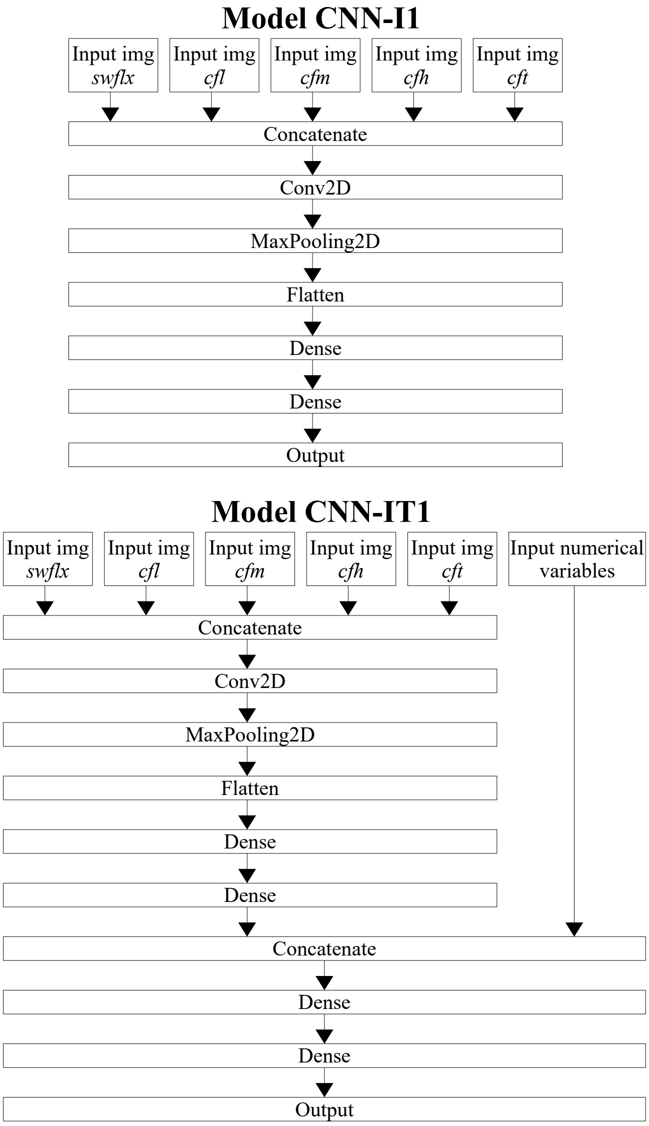


Figure 3.7: CNN-I1 and CNN-IT1 structures.

The last concatenation that can be found in the models CNN-IT1, CNN-IT2 and CNN-IT1 was also tested in different places, however the ideal position, according to tests conducted, is the one present in the figures 3.7, 3.8 and 3.9.

After performing tests in all of the aforementioned models, it was concluded that the best networks are CNN-I2 and CNN-IT3, since they delivered forecasts with lower RMSE than others.

3.3.4 Convolutional LSTM Neural Networks

The convolutional LSTM layers, that are one of the main layers in these networks, work in a similar manner to the LSTM layers, but with convolutional input transformations and recurrent transformations. They are implemented in Keras via ConvLSTM2D class and receive 5D inputs

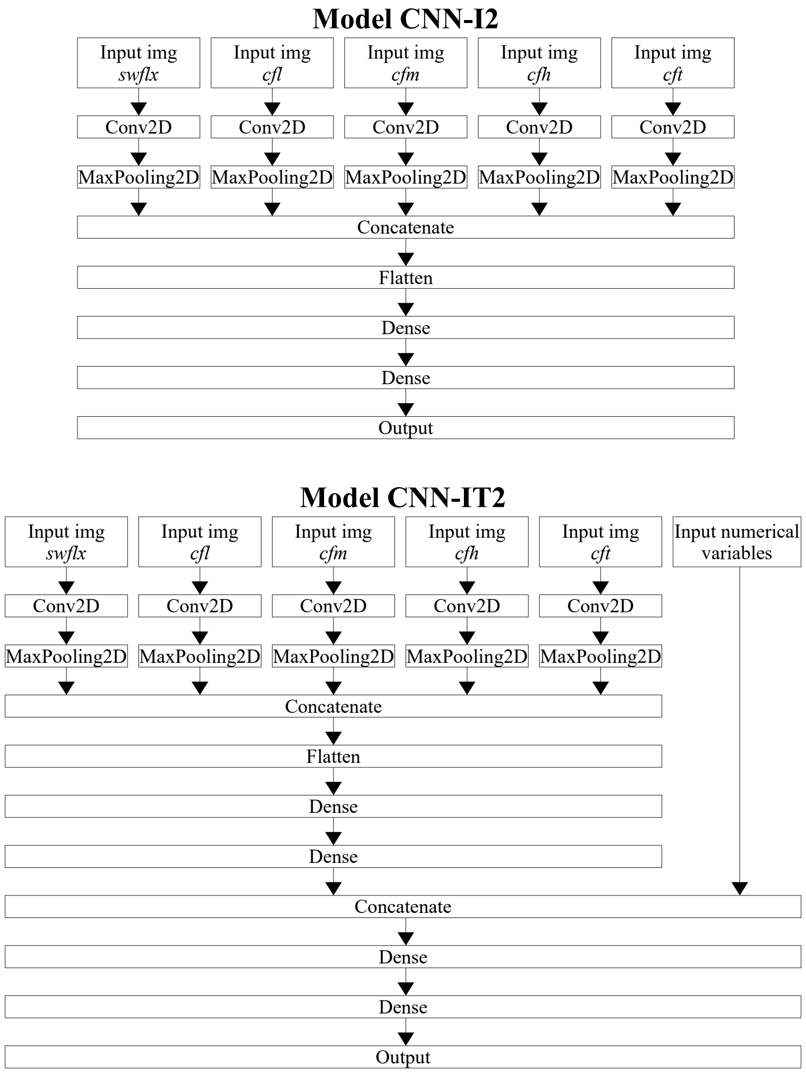


Figure 3.8: CNN-I2 and CNN-IT2 structures.

where the dimensions correspond to samples, time step, height, width and channels and output a matrix with samples, filters, output rows and output columns as dimensions. The number of rows and columns of the outputted matrix is given by the expression 3.13. Convolutional LSTM layers are composed by a set of memory blocks, where each one corresponds to a specific time step. In order to obtain the output of this layer, it is necessary to perform a series of sequential mathematical operations, which are defined in the formulas 3.14, 3.15, 3.16, 3.17 and 3.18.

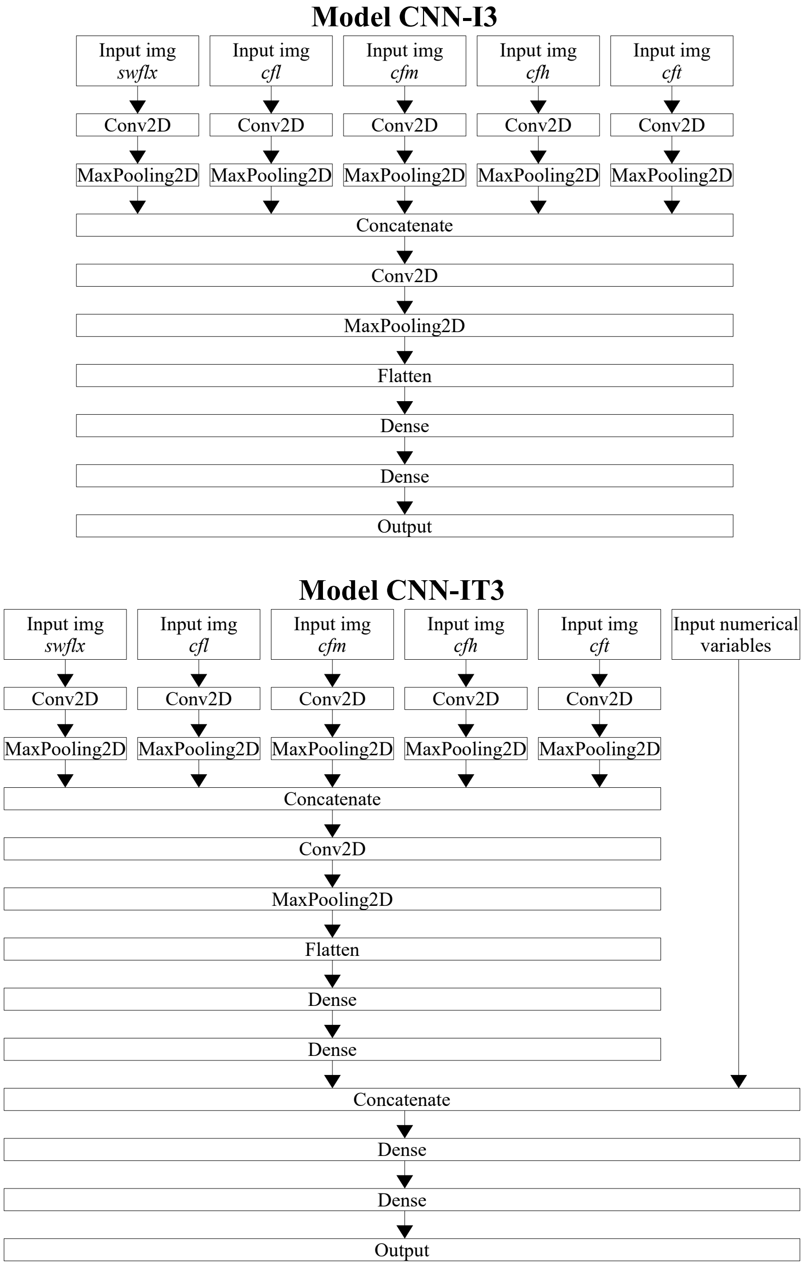


Figure 3.9: CNN-I3 and CNN-IT3 structures.

$$i_t = \text{sigmoid}(W_{xi} * X_t + W_{hi} * h_{t-1} + W_{ci} \circ C_{t-1} + b_i) \tag{3.14}$$

$$f_t = \text{sigmoid}(W_{xf} * X_t + W_{hf} * h_{t-1} + W_{cf} \circ C_{t-1} + b_f) \tag{3.15}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * h_{t-1} + b_c) \tag{3.16}$$

$$o_t = \text{sigmoid}(W_{xo} * X_t + W_{ho} * h_{t-1} + W_{co} \circ C_t + b_o) \tag{3.17}$$

$$h_t = o_t \circ \tanh(C_t) \tag{3.18}$$

Where '$\circ$' denotes the Hadamard product and i_t, f_t, C_t, o_t, h_t, W, X, b and t correspond to input gate, forget gate, cell output, output gate, hidden state, weight, cell input, bias vector and t-th time step, respectively. The functions sigmoid and tanh represent the logistic sigmoid function and the hyperbolic tangent function, respectively.

The images of each variable created in the subsection 3.2.2 were organised in arrays where the dimension of the array corresponded to the time step.

ConvLSTM2D layers combine the capability of LSTM layers to generate temporal features with the ability of Conv2D layers to create spatial features. The intrinsic LSTM structure present in the ConvLSTM2D layers allows them to extract temporal relations between each point and its predecessors. Since some of the operations performed in this structure require intermediate convolutional layers, spatial features can be extracted the same way they were in the subsection 3.3.3. Therefore, convolutional LSTM networks can extract more features than convolutional networks, leading to possibly better solar power predictions. In the figure 3.10, an example of the application of a convolutional LSTM layer in a sequence of images is shown, where lightest and darkest colours correspond to lowest and highest pixel values, respectively. These networks like convolutional networks can receive manual features as input, but only after ConvLSTM2D and pooling layers.

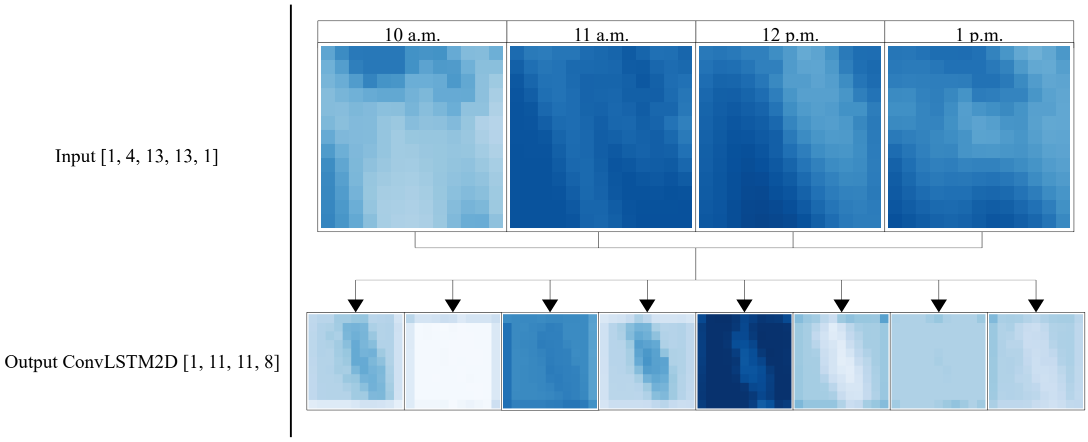


Figure 3.10: Example of the transformations performed by ConvLSTM2D in images related to *swflx* on 2 May 2014 from 10 a.m. to 1 p.m..

In order to determine which are the most adequate input images, several experiments were performed in a network similar to CNN-I1, where the Conv2D layers were replaced by ConvLSTM2D layers. The hyperparameters chosen for the network are based on the ones presented in 3.3.3, because of the existing similarity. From these experiments, it was verified that images

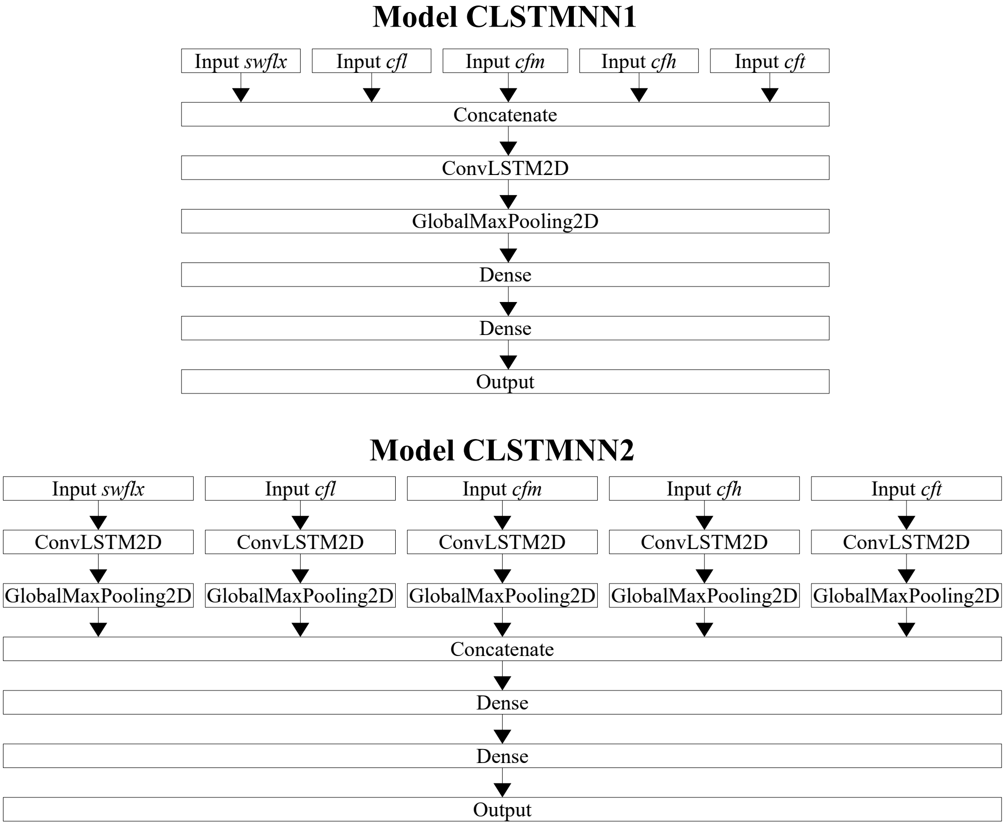


Figure 3.11: CLSTMNN1 and CLSTMNN2 structures.

based on $swflx$, cfl, cfm, cfh and cft variables from the NWP are the most suitable for input, since the images obtained using $temp$ variables don't add any information. In the same tests performed, it was discovered that power measurements and $swflx$ images must be subject to a clear sky and a min-max normalisations, respectively. Additionally, it was verified that ideal time step is 4, which allows the convolutional LSTM networks to extract temporal features between $t+k-3$ and $t+k$, where $t+k$ corresponds to the time when the forecast occurs.

Similar structures to CNN-I2 and CNN-I3, where the convolutional layers were also replaced by convolutional LSTM layers, were tested with, again, similar hyperparameters presented in 3.3.3. Other experiments that consisted in replacing max pooling layers by global max pooling layers in the aforementioned models showed better results. The networks similar to CNN-I1 and CNN-I2 are the only ones compatible to this replacement, originating the models CLSTMNN-1 and CLSTMNN-2, represented in the figure 3.11. The input of global pooling layers is 4D whose dimensions correspond to batch size, rows, columns and channels. These layers return a 2D matrix constituted by batch size and channels. For this reason, a model based on CNN-I3 was impossible to test.

After the tests performed, it was verified that the model CLSTMNN-2 provided forecasts with lower average RMSE than the model CLSTMNN-1. Both structures were subjected to various adjustments, such as adding more convolutional LSTM layers and removing pooling layers, however, these changes led to worse predictions.

Chapter 4

Results

In this chapter, the solar power forecasts provided by different networks are compared in order to determine the value that automatic features add. These models are benchmarked according to the predictions provided on clear sky, partly cloudy and cloudy hours and to metrics that evaluate the overall performance.

The reader is advised to visit the chapter 3 to understand the data used in the models and the selection process of the networks. Each of the compared models goes through a procedure of hyperparameter tuning and is trained using a server.

4.1 Evaluation of Forecasting Accuracy

The solar power forecasts given by different models can be evaluated via several metrics, which allow a performance comparison between themselves. In 4.1, 4.2, 4.3 and 4.4, it is possible to see the formulas of some of the most common metrics: MBE, MAE, MAPE and RMSE.

$$MBE = \frac{1}{N}\sum_{i=1}^{N}(P_{pred} - P_{meas}) \tag{4.1}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|P_{pred} - P_{meas}\right| \tag{4.2}$$

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{P_{pred} - P_{meas}}{P_0}\right| \tag{4.3}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_{pred} - P_{meas})^2} \tag{4.4}$$

Where P_{pred}, P_{meas}, P_0 and N mean predicted power, measured power, power capacity of the PV plant and number of predictions/measurements, respectively.

MAE displays the average distance between the predicted power and the respective measurement. Both MAE and MAPE are suitable to evaluate uniform forecast errors. While MBE can determine if a model is over/underestimating, RMSE penalises large errors in a square order [26].

4.2 Hyperparameter Tuning

It was seen in the subsection 3.3, that the models which provided better solar power forecasts were:

- DNN2: a dense neural network with a hidden layer that receives manual engineered variables as input;
- CNN-I2: a convolutional neural network whose inputs are generated images from variables present in the NWP data set;
- CNN-IT3: a convolutional neural network with an upper structure more complex than CNN-I2, which extracts features from the same images used in CNN-I2 and uses temporal and chronological features manually created;
- CLSTMNN2: a convolutional long-short-term memory neural network that extracts both spacial and temporal features with a time horizon of 3 hours using exclusively the same set of images that CNN-I2 and CNN-IT3 received as input.

Hyperparameters are essential since they influence the overall performance of ANNs. The main goal of hyperparameter tuning is to discover of the optimal set of hyperparameters that minimises a specific loss function and, consequently, ensures the best predictions. Failure to do so provides sub-optimal forecasts. Although this procedure is important for the performance of each model, it is computationally expensive and time-consuming. Additionally, it is extremely difficult to find the optimum set of parameters.

The selection of hyperparameters of each network presented is done using Keras Tuner [53], which is a library that helps pick the optimal set of hyperparameters for each model. The first step to perform a hyperparameter tuning consists in defining the training and validation sets which will be used to obtain intermediate errors that will allow the selection of the best hyperparameters. Then, it is necessary to define which hyperparameters to search and their respective search space. Lastly, the number of trials, number of epochs per trial and tuner must be chosen.

Each of the proposed models has its hyperparameters tuned according to the test and validations sets referred in the subsection 3.3.1. Therefore, each network is subject of five processes of hyperparameter selection corresponding to each data fold.

The number of epochs per trial is defined by early-stopping, which is implemented in Keras using EarlyStopping class. This class allows to stop the training of a neural network after a determined number of epochs in which any improvement in the result occurred. The early-stopping implemented in each model stops the training after 25 epochs where the validation set loss dos not show any improvement.

For this case study, Keras Tuner provides the following tuners: Random Search, Hyperband and Bayesian Optimisation. Random Search and Hyperband tries random combinations of hyperparameters, however, Hyperband focus on speeding up the process by adaptive resource allocation and early-stopping [54]. Unlike the other tuners, Bayesian Optimisation monitors the previous evaluation results and uses it to build a probabilistic model, which is then exploited to make decisions about the next set of hyperparameters to evaluate [55]. The tuner chosen for the hyperparameter optimisation is Bayesian Optimisation, because it is more likely to lead to better results in lesser trials since its choice of hyperparameters set is based on a determined function and not a random distribution.

The appendix A contains the structures with the representation of the hyperparameters related to the network structure and activation functions that are subject to a hyperparameter tuning. Additionally, it includes tables with search space for each hyperparameter of the proposed networks. The ideal hyperparameter values found for each fold is also present in this appendix.

4.3 Comparison Between Forecasts Provided by Each Model

After tuning the hyperparameters, it was obtained the forecasts for each fold of models DNN2, CNN-I2, CNN-IT3 and CLSTMNN2. The input data of each model come from the data sets reviewed in the subsection 3.3.1. However, the validation sets merge with the training sets, creating more extensive training sets which can lead to an increase model performance. The training and test sets are represented in the time diagram present in the figure 4.1. The obtained predictions allow an analysis between the proposed models, and consequently help determine which one is the most adequate.

To determine the errors in the following subsections, the forecasts were filtered according to the value of the zenith angle in the PV plant site using solar calculations proposed by NOAA [56]. During daytime, this angle is between 0^o and 90^o, and during nighttime, it is higher than 90^o.

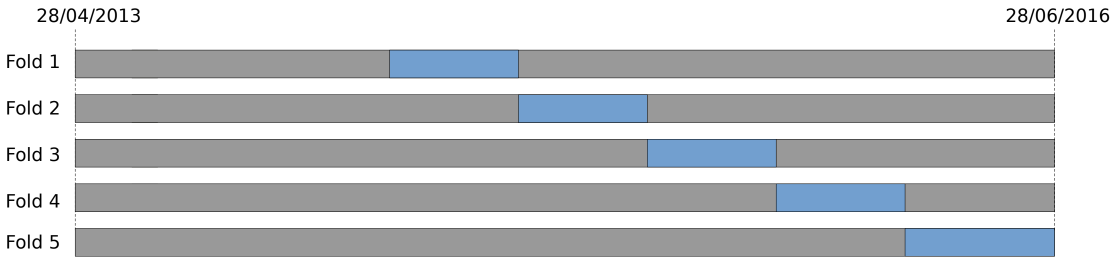


Figure 4.1: Time diagram of the sets for each fold, where grey and blue shapes correspond to training and test sets, respectively.

This filtration of predictions prevents errors from being underestimated, since at night the power produced is zero.

4.3.1 Overall Benchmark

To evaluate the overall performance of the forecasts retrieved by the models, the metrics RMSE, MAE, MAPE and MBE are used. In this case, MAE is ignored, since MAPE is an extent of the referred metric. The tables 4.1, 4.2 and 4.3 contain the RMSE, MAPE and MBE associated with each model and fold, respectively.

Table 4.1: Values of RMSE for each fold and model.

Fold	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
1	9.98%	10.27%	10.44%	9.86%
2	9.50%	10.16%	9.89%	9.98%
3	9.57%	10.23%	10.60%	10.33%
4	10.17%	10.01%	9.64%	9.65%
5	9.81%	11.91%	10.43%	9.97%
Average	9.84%	10.60%	10.25%	9.96%

Table 4.2: Values of MAPE for each fold and model.

Fold	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
1	6.00%	6.18%	6.09%	5.94%
2	6.33%	7.30%	6.79%	6.94%
3	5.55%	5.95%	6.24%	6.09%
4	6.16%	6.44%	6.26%	5.96%
5	7.05%	7.78%	6.83%	6.63%
Average	6.23%	6.72%	6.43%	6.29%

Table 4.3: Values of MBE for each fold and model.

Fold	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
1	-58.20W	29.84 W	-129.57 W	-88.75 W
2	235.45 W	200.07 W	171.92 W	74.69 W
3	-192.09 W	-269.85 W	-150.45W	81.93 W
4	179.68 W	-71.69 W	189.15 W	-14.94 W
5	76.45 W	-554.94 W	-269.19 W	-99.80 W
Average	35.54 W	-156.40 W	-60.79 W	-17.17 W

The model DNN2 has a better overall performance than the reaming ones, as it has the lowest average RMSE and MAPE. This can be justified by the exhaustive optimisation of features carried out by Andrade and Bessa for this data set.

The network CNN-I2 is the one that presents the worst global performance as it can be seen in the metrics represented in the tables 4.1 and 4.2. Unlike the other models, CNN-I2 uses exclusively the spatial features generated in the Conv2D layers of the network to predict the power produced by the PV panels, thus it is at disadvantage compared to the others.

CNN-IT3, followed by CNN-I2, is the model with lowest performance, therefore it can be concluded, in a quick analysis, that the spatial features created automatically are worse than the ones engineered manually. From the comparison between these two models it is possible to deduce that the use of manual features that could not be generated by the model itself improves the overall performance of the network.

Despite CLSTMNN2 being the second-best performing model, according to the table 4.1 and 4.2, it presents a similar global performance to DNN2, as the difference of average RMSE and MAPE between these two models is relatively small. The network CLSTMNN2 is capable of making better predictions in the folds 1 and 4 than DNN2. CLSTMNN2, unlike DNN2, relies only on spatial and temporal features created automatically with a time horizon of 3 hours.

From the table 4.3, it can be concluded that model DNN2 tends to underestimate the values of the produced power in the PV plant, while the remaining models tend to overestimate them.

The analysis performed above showed that convolutional neural networks and convolutional long-short-term memory neural networks can extract valuable features that can surpass the performance of those created manually. In the following subsections, the performance of each proposed model is further evaluated by benchmarking them according to different weather conditions.

4.3.2 Clear Sky Hours Performance

It is considered clear sky when there no clouds present in the sky over the PV plant site, therefore, in this analysis only measured and predicted solar power associated with cfl, cfm, cfh and cft values of 0 in the central point of NWP grid are evaluated. These values that allow the stated selection are given by the latest run.

In the figure 4.2 are present the some graphs of the measured and predicted powers by the different models in clear sky days. Most of the forecasts that are represented in the referred figure display good accuracy as the predicted power is close to the measured one.

For the first example, 3 May 2014, it is verified that the measured production is very close to the forecasted one by the proposed models from the beginning of the day until 10 a.m., where there is a small gap between them. The peak production predicted by the models with the exception of DNN2 has a delay, however, from 4 p.m. till dusk a new approximation of values occurs. For the case presented, the model DNN2 performs better than the others, since it has the lowest RMSE and MAPE, as it can be seen in the table 4.4. In addition, it is concluded that all models underestimate the value of generated power.

Table 4.4: Metrics of each model for 3 May 2014.

Metric	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	1.03%	2.18%	1.75%	1.92%
MAPE	0.88%	1.78%	1.41%	1.55%
MBE	66.48 W	152.38 W	59.92 W	144.71 W

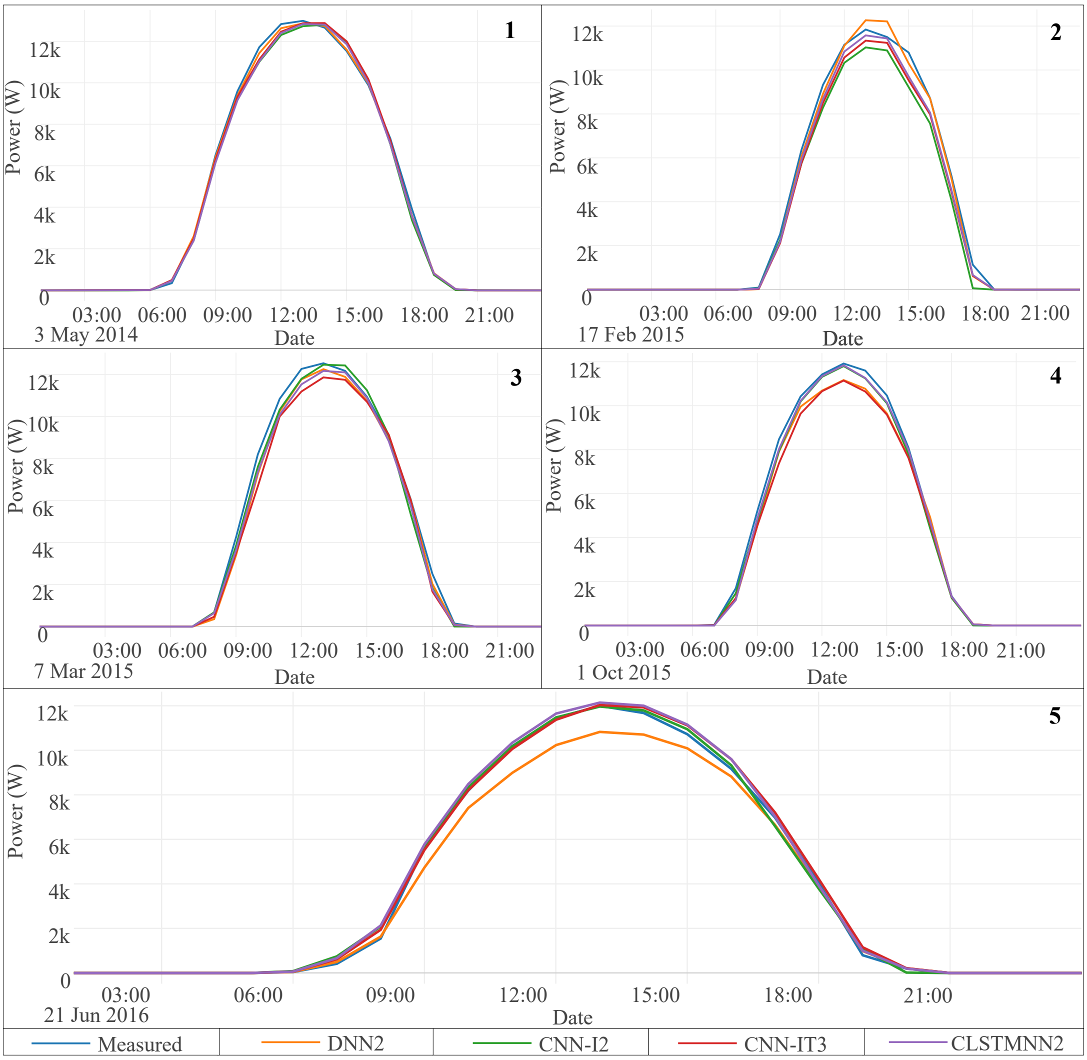


Figure 4.2: Graphs of forecasts retrieved by the proposed models versus the measured power in clear sky days, numbered according to the fold where they belong.

The case of 17 February 2015, which belongs to the fold 2, there is a more noticeable difference between the forecasts provided by each model than the aforementioned example. It is demonstrated again that the predictions at the beginning of the day assume similar values to ones measured, however, at 10 a.m. the forecasts retrieved by different models follow different paths. The predictions originated from network DNN2 overestimate the peak power production that occurred between 12 p.m. and 2 p.m., while the forecasts provided by the remaining networks underestimates them. The created spatial features on convolutional neural networks worsened the predictions, as CNN-I2 and CNN-IT3 had lower performance than DNN2. The use of manual engineered temporal and chronological features in the same type of networks seems to improve the performance of the models since CNN-IT3 originates better predictions than CNN-I2. From all the presented models, DNN2 is again the best performing network as it can checked in the table 4.5.

Table 4.5: Metrics of each model for 17 February 2015.

Metric	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	2.32%	5.66%	4.05%	3.28%
MAPE	1.91%	5.07%	3.61%	2.77%
MBE	104.16 W	827.15 W	588.48 W	452.48 W

The example shown for fold 3, unlike the aforementioned cases, displays a noticeable dispersion of forecasts during the morning. The model CNN-I2 has a better overall performance during 7 March 2015 compared to the other models, as it delivers predictions with lower underestimations between 7 a.m. to 1 p.m.. The performance of each model can be checked in the table 4.6. It can be concluded that during this day, the use of manual features in convolutional neural networks worsened the quality of the forecasts, since the ones generated by CNN-IT3 have higher RMSE and MAPE than the ones provided by CNN-I2, which only uses the extracted features from the input images.

Table 4.6: Metrics of each model for 7 March 2015.

Metric	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	3.10%	2.70%	4.60%	3.23%
MAPE	2.61%	2.26%	3.73%	2.67%
MBE	426.06 W	256.14 W	596.19 W	435.09 W

On 1 October 2015, it is verified that during the morning the predictions follow a similar path presented on 7 March 2015. By 11 a.m., it is clear that the forecasts provided by DNN2 and CNN-IT3 deviate from the measured power values, and they only return to their encounter only after 4 p.m.. The forecasts originated from CNN-I2 and CLSTMNN2 seem to describe properly the evolution of the generated power in the photovoltaic plant, which can be proven by the low RMSE and MAPE that are described in the table 4.7. In short, during this day the models that use exclusively automatic features have better performance than the ones which use features manually created.

Table 4.7: Metrics of each model for 1 October 2015.

Metric	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	3.51%	1.83%	4.18%	1.65%
MAPE	3.09%	1.59%	3.59%	1.27%
MBE	474.27 W	259.42 W	586.02 W	207.23 W

The figure 4.2 shows, on 21 June 2016, that the model DNN2 provides good forecasts for the first two hours of the day, however these predictions in the next hours deviate significantly from the actual PV power production values. This model tends to underestimate the values of the produced power during that day, while the others tend to overestimate them. This statement is confirmed according to the MBE metric present in the table 4.8. At the end of the day, the forecasts retrieved by the studied networks assume similar values to the measured ones. In this case, the use of automatic temporal features whether automatically or manually generated reduces the overall performance of the forecasts.

Table 4.8: Metrics of each model for 21 June 2016.

Metric	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	4.29%	1.44%	1.47%	1.68%
MAPE	3.29%	1.11%	1.18%	1.36%
MBE	486.92 W	-68.03 W	-148.09 W	-207.66 W

In the tables 4.9 and 4.10 are presented the values of RMSE and MAPE, respectively, of the predictions generated from DNN2, CNN-I2, CNN-IT3 and CLSTMNN2 in clear sky hours, which allow to evaluate the performance of the mentioned models. The network DNN2 along with CLSTMNN2 provides better overall forecasts as they contain the lowest average RMSE and MAPE, respectively. Although CNN-I2 and CNN-IT3 have worse global metrics, they present lower RMSE or/and MAPE in the some of the folds, showing their potential. The models CLSTMNN2, CNN-IT3 and DNN2 can be considered identical, in terms of performance, since they assume average RMSE and MAPE values close to each other. According to the table 4.11, the models designed tend to underestimate the value of the production PV power in clear sky conditions.

Table 4.9: Values of RMSE for each fold and model in clear sky hours.

Fold	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
1	8.17%	8.56%	8.40%	8.19%
2	7.15%	9.24%	8.23%	8.49%
3	7.72%	7.27%	8.34%	7.93%
4	8.11%	8.37%	8.27%	8.16%
5	9.06%	9.42%	8.36%	8.63%
Average	8.12%	8.49%	8.33%	8.24%

Table 4.10: Values of MAPE for each fold and model in clear sky hours.

Fold	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
1	4.44%	4.57%	4.27%	4.35%
2	4.75%	7.01%	5.95%	6.14%
3	3.95%	4.05%	4.51%	4.49%
4	5.18%	5.06%	5.35%	4.57%
5	6.24%	5.52%	4.95%	5.25%
Average	4.85%	5.00%	4.87%	4.81%

Table 4.11: Values of MBE for each fold and model in clear sky hours.

Fold	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
1	-59.74 W	-82.69 W	-193.04 W	-126.27 W
2	190.83 W	578.34 W	427.57 W	439.66 W
3	-153.06 W	-171.86 W	5.43 W	71.72 W
4	243.49 W	119.16 W	264.80 W	21.02 W
5	462.30 W	30.31 W	76.60 W	195.38 W
Average	107.39 W	34.58 W	69.95 W	81.01 W

4.3.3 Partly Cloudy Sky Hours Performance

The performance evaluation of the models in partly cloudy sky hours is done considering only hours when the central values of cfl, cfm, cfh and cft of the NWP grid given by the last run are in the following interval [0, 0.3[. It should be noted that hours when all the values of the variables mentioned are equal to zero are not considered, as they belong to the group of forecasts analysed in subsection 4.3.2.

The figure 4.3 shows 5 graphs that represent the predicted and measured powers in days composed mostly by partly cloudy sky. Each graph present in the aforementioned figure can be considered as an illustrative example of each data fold.

On 20 August 2014, the NWP at the central point determined the occurrence of sky partially clouded from 7 a.m. to 5 p.m.. On this day there is a constant underestimation of production with the exception of the period between 7 a.m. and 9 a.m.. The models, in general, show difficulty in predicting the generated power values between 10 a.m. and 4 p.m., which may result from the existence of fewer clouds than predicted by the NWP. These networks were unable to predict the time corresponding to the peak of production with the exception of CNN-IT3. According to table 4.12, the model DNN2 is the best performing one on this day, as it shows lower RMSE and MAPE than the remaining networks.

Table 4.12: Metrics of each model for 20 August 2014.

Metric	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	6.68%	8.45%	9.84%	9.81%
MAPE	4.79%	6.46%	7.06%	7.01%
MBE	613.47 W	951.52 W	896.47 W	895.44 W

The second example, corresponding to the 25 October 2014, there is a significant underestimation of the power values until approximately 1 p.m. which is followed by an overestimation until 6 p.m.. During this day there may have been fewer clouds than expected during the underestimation period and more clouds during the overestimation period. Unlike the example corresponding to 20 August 2014, the models were able to predict the time when the peak production occurred. On this day, the exclusive use of automatically extracted spatial features originated forecasts with higher quality, since the predictions generated from CNN-I2 have lower RMSE and MAPE than ones provided by CNN-IT3 and CLSTMNN2, as shown in table 4.13.

Table 4.13: Metrics of each model for 25 October 2014.

Metric	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	9.03%	4.94%	7.45%	7.32%
MAPE	7.38%	3.85%	5.78%	6.19%
MBE	678.49 W	24.21 W	94.48 W	253.74 W

In the case of 23 April 2015, which belongs to the fold 3, it is verified that the forecasts generated by the proposed models are generally of good quality, since they follow a similar path to the one taken by the power measurements. It should be noted that at 2 p.m., the sky is clear, hence there was an increase in power generation at that same hour. The models tend to underestimate the solar production, as can be seen from the values of the MBE metric present in table 4.14. As on 25 October 2014, the exclusive use of automatically extracted spatial features led to predictions with smaller errors which can be seen in table 4.14. In this specific case, the use of manually created features leads to a considerable increase in error.

Table 4.14: Metrics of each model for 23 April 2015.

Metric	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	5.20%	3.35%	4.94%	4.70%
MAPE	3.99%	2.62%	3.97%	3.44%
MBE	294.08 W	20.37 W	481.87 W	186.89 W

The graph 4, which corresponds to 5 August 2015, shows that every model was able to adequately predict the power values generated for most of the useful period. During the period when the zenith angle is lower than 90^o, the sky is partially cloudy except between 7 a.m. and 10 a.m. and at 2 p.m.. On this day, the sky has very few clouds since the production curve is similar to that one found on clear days. The model CLSTMNN2 behaved in a remarkable way, which can be confirmed by the values of the metrics present in table 4.15. The use of spatial, temporal and chronological features engineered manually as network inputs worsens the quality of the forecasts.

Table 4.15: Metrics of each model for 5 August 2015.

Metric	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	3.67%	3.10%	3.91%	2.33%
MAPE	3.14%	2.75%	3.38%	1.71%
MBE	450.05 W	368.49 W	506.40 W	162.51 W

On March 13, 2016, all the models studied underestimated the power values produced during the entire period in which production occurred. The possible cause for this event is that there may have fewer clouds than NWP forecasted. On this day, the model CNN-IT3 is the best performing one, which is shown in table 4.16. From the referred table, it is concluded that the spatial features created manually worsened the quality of the predictions, since DNN2 is the one that presents the worst performance and is the only one that uses the mentioned features.

Table 4.16: Metrics of each model for 13 March 2016.

Metric	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	6.69%	5.32%	4.95%	5.38%
MAPE	5.98%	4.70%	4.39%	4.78%
MBE	976.23 W	766.79 W	716.82 W	779.71 W

The tables 4.17 and 4.18 show that the best performing networks for predicting the power generated in hours with partly cloudy sky are DNN2 and CLSTMNN2. Both models have similar overall performance in these events. The models DNN2 is the one with the best performance in folds 1 and 3, while the models CNN-IT3 and CLSTMNN2 are the ones that provide the best predictions in folds 2 and 5, respectively. The forecasts originated from the model CNN-I2, on the other hand, did not stand out in any of the folds and presents the worst global metrics. According to the table 4.19, model DNN2 is the only one that tends to underestimate the production values, whereas the other models tend to overestimate them.

Table 4.17: Values of RMSE for each fold and model in partly cloudy sky hours.

Fold	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
1	10.71%	11.02%	11.20%	10.77%
2	11.61%	11.66%	11.45%	11.76%
3	12.38%	12.79%	12.93%	13.34%
4	10.18%	10.82%	10.32%	10.32%
5	11.54%	13.50%	12.42%	11.13%
Average	11.30%	12.01%	11.72%	11.47%

Table 4.18: Values of MAPE for each fold and model in partly cloudy sky hours.

Fold	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
1	6.80%	7.07%	6.97%	6.87%
2	8.11%	8.40%	7.99%	8.20%
3	7.25%	7.43%	7.88%	7.97%
4	6.73%	7.08%	6.76%	6.55%
5	8.31%	8.86%	8.31%	7.62%
Average	7.43%	7.76%	7.57%	7.42%

Table 4.19: Values of MBE for each fold and model in partly cloudy sky hours.

Fold	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
1	-130.21 W	-37.89 W	-260.07 W	-154.84 W
2	316.64 W	189.56 W	141.78 W	137.70 W
3	-203.92 W	-280.28 W	-212.94 W	124.88 W
4	232.39 W	-77.54 W	176.02 W	-70.85 W
5	28.68 W	-760.42 W	-383.40 W	-220.39 W
Average	28.28 W	-204.67 W	-135.77 W	-51.37 W

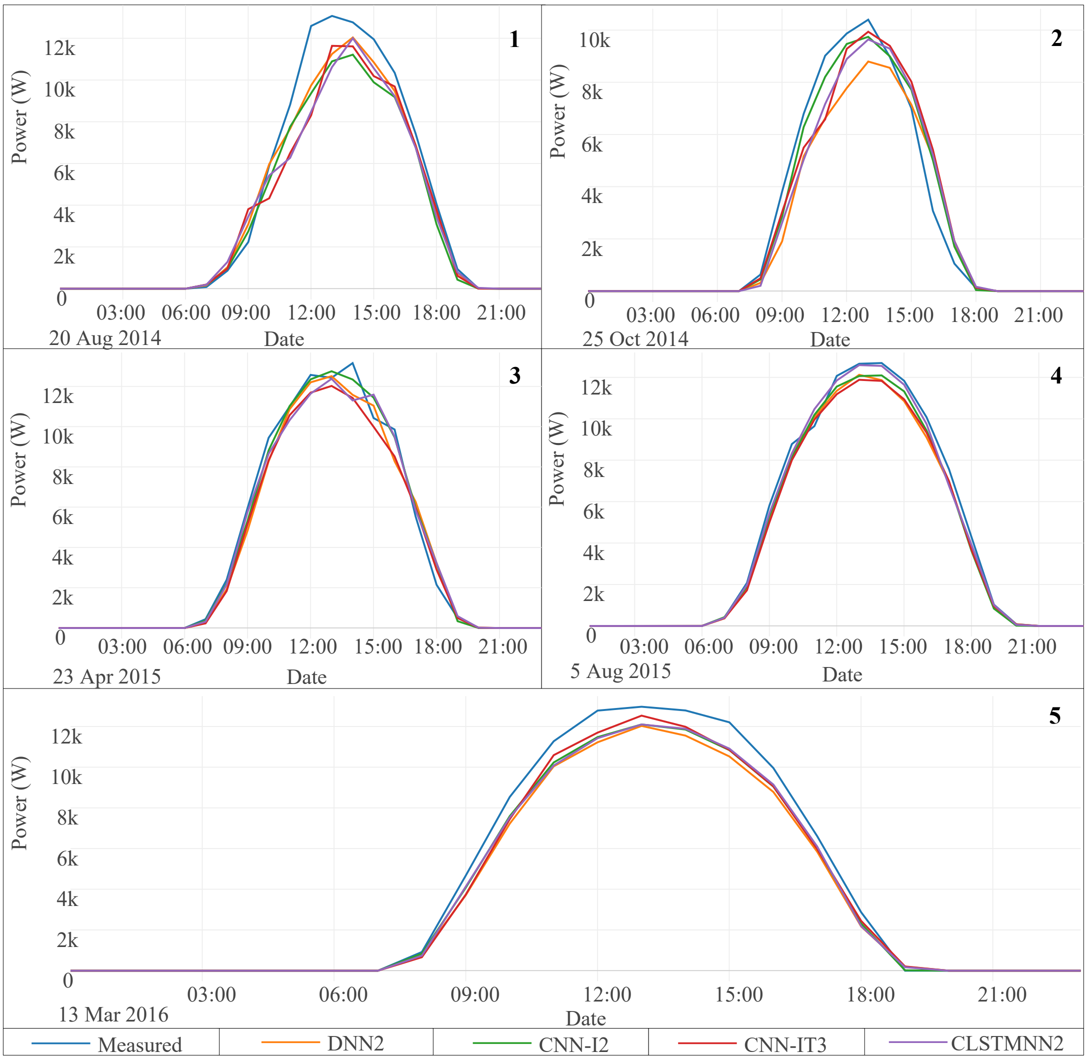


Figure 4.3: Graphs of forecasts retrieved by the proposed models versus the measured power in partly cloudy sky days, numbered according to the fold where they belong.

4.3.4 Cloudy Sky Hours Performance

The performance of the proposed models under cloudy sky conditions is assessed by selecting the hours when at least one of the central values of cfl, cfm, cfh and cft provided by NWP grid on the latest run is higher or equal to 0.3. Under these circumstances, the production of solar power is expected to be lower than the ones presented in the subsections 4.3.2 and 4.3.3, as the values of short wave flux are lower.

In the figure 4.4 are represented 5 illustrative graphs, one for each data fold, that show the values of forecasted and measured powers under cloudy sky conditions during one day. The production curves that are present in the aforementioned figure assume different forms compared to those that in the figures 4.2 and 4.3 due to a higher cloud coverage variability verified in these

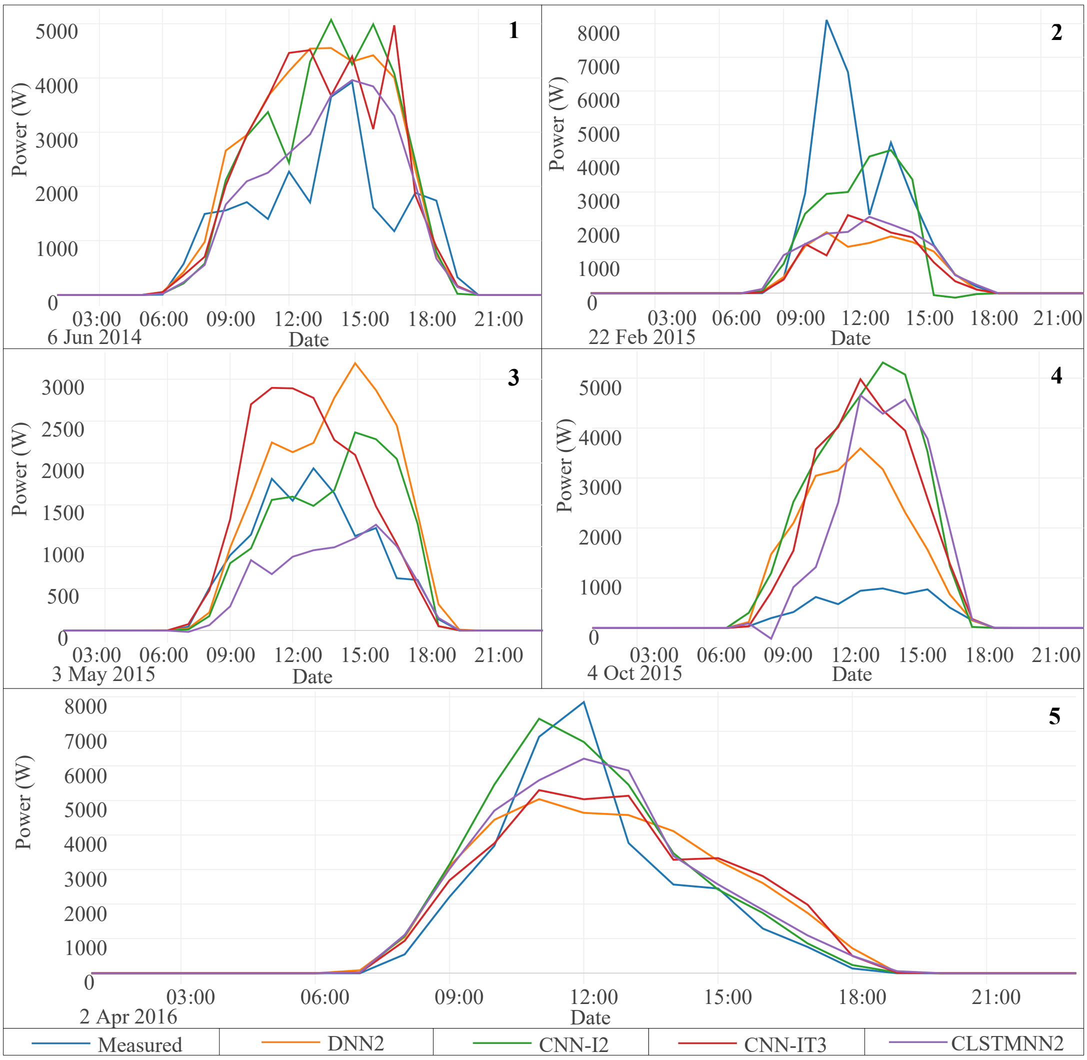


Figure 4.4: Graphs of forecasts retrieved by the proposed models versus the measured power in cloudy sky days, numbered according to the fold where they belong.

situations.

The graph 1 of figure 4.4 shows that there is an overestimation of solar production, with the exception at the beginning and at the end of the day. The model CLSTMNN2 is the only one that is able to detect peak production and is the one that best described the production curve between 9 a.m. and 6 p.m., which is reflected in the value of its metrics that is present in table 4.20. The excessive overestimation of the model DNN2 devalued its performance and led it to have the worst quality forecasts of those studied.

On 22 February 2015, it appears that there was an error in the NWP, since none of the proposed models managed to detect a sharp increase in the power produced between 10 a.m. and 11 a.m.. The NWP must have foreseen the existence of more clouds for the referred period. The model

Table 4.20: Metrics of each model for 6 June 2014.

Metrics	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	9.72%	9.59%	9.71%	5.99%
MAPE	7.57%	7.22%	6.86%	4.12%
MBE	-993.02 W	-842.74 W	-846.79 W	-335.06 W

CNN-I2 is the one that presents forecasts with lower underestimation of solar power values generated between 10 a.m. and 12 p.m. and at 2 p.m.. Despite the poor forecasts presented after 4 p.m. and the anomaly that occurs at 5 p.m., this network is the one with the best overall performance during this day, as can be seen from table 4.21. It should be noted that the use of temporal features seems to improve the models' performance at the end of the afternoon.

Table 4.21: Metrics of each model for 22 February 2015.

Metrics	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	16.44%	12.52%	16.31%	15.72%
MAPE	10.22%	8.19%	9.84%	9.47%
MBE	1651.22 W	833.18 W	1601.11 W	1389.14 W

In the case of example 3, corresponding to 3 May 2015, it is verified that the forecasts at the beginning of the day provided by model CNN-IT3 are very accurate, but end up diverging in the following hours to values excessively high values during peak production. The model CNN-I2 presents a production curve from 9 a.m. to 2 p.m. similar to the one that contains the measured power. From 3 p.m. until the end of the day, the model CLSTMNN2 is the best performing one, however it underestimates significantly the production values in the remaining hours of the day. The high divergence of values verified between 3 p.m. and 7 p.m. in the forecasts of the model CNN-I2 lead the network CLSTMNN2 to present the best global performance during this day, which can be checked in the table 4.22.

Table 4.22: Metrics of each model for 3 May 2015.

Metrics	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	5.95%	3.86%	4.60%	3.24%
MAPE	4.30%	2.58%	3.39%	2.33%
MBE	-657.48 W	-219.65 W	-526.26 W	318.49 W

The graph, that corresponds to 4 October 2015, shows that NWP contain considerable errors as every model overestimated the values of the solar power generated from 9 a.m. to 6 p.m.. Contrary to what happened on 22 February 2015 (graph 2), the NWP must have foreseen less clouds than expected. At 9 a.m. it appears that the network CLSTMNN2 produced an anomalous forecast and this, like the others models, failed to determine the time when the peak production occurred. According to table 4.23, the network DNN2 is the one that delivers forecasts with lower RMSE and MAPE, since this model is the one that produces the least underestimation of values compared to the others.

Table 4.23: Metrics of each model for 4 October 2015.

Metrics	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	10.48%	16.73%	14.64%	13.61%
MAPE	8.26%	13.41%	11.26%	9.96%
MBE	-1347.56 W	-2165.76 W	-1837.29 W	-1556.51 W

The NWP seems to have an adequate quality on April 2, 2016, as the forecasts provided by the models tend to follow the measured power values. The CLSTMNN2 is the only model able to detect the time when peak production occurs, however during that hour the network that presented the best forecast is the CNN-I2. Additionally, the referred model delivered the best hourly predictions at 11 a.m. and between 3 p.m. and 6 p.m., which led this network to be considered best overall performing one for this day, as can be seen in table 4.24. The use of manual time and chronological features seems to originate worse forecasts between 11 a.m. and 5 p.m. (except for the model CNN-IT3 at 2 p.m.). The table previously mentioned shows that all models tend to underestimate the generation of PV power as the values of the metric MBE are negative.

Table 4.24: Metrics of each model for 2 April 2016.

Metrics	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
RMSE	8.27%	5.54%	7.46%	6.11%
MAPE	6.78%	4.20%	5.82%	4.90%
MBE	-271.34 W	-488.77 W	-223.99 W	-317.79 W

Under cloudy sky conditions the model DNN2 has better overall performance than the other networks, as it delivers predictions with the lowest RMSE and MAPE, as it can be seen in the tables 4.25 and 4.26. Although CLSTMNN2 is the second-best performing network, it delivers predictions with lower RMSE and MAPE on the fold 1. The model DNN2 is considered the worst performing one as its associated RMSE and MAPE are higher than the remaining.

The networks tend to retrieve forecasts that overestimate the value of the PV power, since the average MBE of each model is negative, as it can be verified in the table 4.27.

Table 4.25: Values of RMSE for each fold and model in cloudy sky hours.

Fold	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
1	11.61%	11.76%	12.30%	11.13%
2	9.30%	9.67%	9.76%	9.58%
3	10.36%	11.83%	11.65%	10.87%
4	10.67%	11.15%	10.62%	10.75%
5	10.14%	12.58%	10.60%	10.21%
Average	10.39%	11.53%	10.94%	10.46%

Table 4.26: Values of MAPE for each fold and model in cloudy sky hours.

Fold	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
1	7.56%	7.71%	7.97%	7.43%
2	6.10%	6.71%	6.47%	6.54%
3	6.68%	7.69%	7.61%	7.15%
4	6.94%	7.65%	7.00%	7.21%
5	6.94%	8.85%	7.40%	7.09%
Average	6.84%	7.82%	7.28%	7.07%

Table 4.27: Values of MBE for each fold and model in cloudy sky hours.

Fold	DNN2	CNN-I2	CNN-IT3	CLSTMNN2
1	22.66 W	279.42 W	111.78 W	41.87 W
2	210.95 W	-39.96 W	24.91 W	-205.22 W
3	-256.65 W	-421.69 W	-358.53 W	55.08 W
4	66.94 W	-295.25 W	108.07 W	-18.95 W
5	-183.25 W	-874.25 W	-461.89 W	-251.33 W
Average	-37.48 W	-329.04 W	-147.88 W	-98.97 W

4.4 Hardware Specifications

The CPU, whose main purpose is to sequential run tasks that are kept in the computer memory, processes the mathematical operations during the training phase of neural networks. Since CPU usually contain few strong processing cores, they are ideal for running sequential tasks [57]. These components, whether they are used to train neural networks, handle all the input/output operations [58].

The models developed were tested in a server with the following specifications:

- CPU: AMD EPYC 7351P 16-Core (up to 2.9GHz , 64MB of cache) [59];
- GPU: GEFORCE GTX 1080 Ti with 11GB of memory and 3584 cores (up to 1.582GHz) [60].

The table 4.28 contain the average runtime of hyperparameter tuning and training of each model per fold using the server with the aforementioned specifications.

Table 4.28: Average runtime of hyperparameter tuning and training of each model per fold.

Models	Hyperparameter tuning	Training
DNN2	2 h	2 min
CNN-I2	1 h 15min	3 min
CNN-IT3	1 h	3 min
CLSTMNN2	19 h	15 min

4.5 Final Remarks

The present chapter proves the value of the features created automatically in the field of solar power forecasting, as the models studied that rely on them have similar performances to the one designed whose inputs are manually engineered features.

The use of night values in the training set reduces forecast errors at the beginning and at the end of the day, increasing the overall performance of each model.

The errors present in the NWP prejudice the predictions of the models, since they will be predicting the values of PV power under non-real conditions. As NWP has a greater difficulty in predicting atmospheric conditions when there are clouds in the sky, due to the increased uncertainty brought by them, there are higher errors associated with the power forecasts in the models designed.

A more in-depth tuning, doing more trials and using independent hyperparameters, could further reduce the error present in the forecasts provided by the studied networks, mainly of CNN-I2, CNN-IT3 and CLSTMNN2, as these networks depend on their hyperparameters to extract the features by themselves.

The use of GPU instead of CPU reduced by approximately 3 times the runtime of hyperparameter tuning of CNN-I2, CNN-IT3 and CLSTMNN2, thus it is an important tool in the deep learning process.

In this chapter, it is concluded that the model CLSTMNN2 is the most adequate one to use since it provides, generally, forecasts of good quality, does not require a long search for manual features and is scalable to other PV plants. The application of the exact same features determined in the model DNN2 in other solar power plants can deliver predictions with high RMSE and MAPE, as those features might not be the ideal ones for that plant, due to the existing different relations between weather conditions and power produced. Thus, the capacity to learn which spatial and temporal features are most suitable considering the data introduced is one of the main factors that makes the model CLSTMNN2 so appealing.

Chapter 5

Conclusions and Future Work

This chapter is divided in two sections, the first one addresses the major conclusions obtained from the conducted work, while the last one introduces some lines of possible future research that could lead to better forecast accuracy.

5.1 Conclusions

There is a continuous need to improve the quality of solar power forecasts, due to increasing penetration of PV power in electrical power systems. Over the years, several approaches have been tested, the common ones being those that require the creation of manual features. This process of engineering features is usually time-consuming, as it involves manual testing in order to verify which set of features is the most adequate. In addition, the features created for a model designed for a determined photovoltaic plant may not be suitable for another, since the relationships between climatic conditions and power may be different. Therefore, in this work, several networks were developed that automatically extract features from the inputs, in order to provide good quality forecasts and allow generalisation to other PV plants.

The use of deep learning structures allows the models to learn for themselves which features to extract from the network inputs according to each study case. The hyperparameters play an important role in these networks, since the features extracted depend on them. Thus, better forecasts require more time to tune the hyperparameters.

Keras proved to be a useful tool in the development of the models proposed in this research, as it allowed an easy construction of those networks. This is mainly due to the fact that this platform is user-centred instead of machine-centred, allowing an intricate model to be designed with a small number of actions without compromising its performance. The variety of types of layers available and their modularity make Keras a good option for creating even the most mathematically complex models.

In general, the four models studied in deeper detail are capable of generating good quality solar power forecasts. The model DNN2 is the one that generally presents the best performance, which is followed by CLSTMNN2. CNN-I2 is considered the model with the worst performance of all since it has the highest overall RMSE and MAPE, which can be justified by the exclusive use of spatial features. During the research, it was confirmed that the model CLSTMNN2 is the one that requires more computational power both in the hyperparameter tuning and in the network training, due to the complex temporal relations that the ConvLSTM layers extract. Future advances in GPU and CPU will reduce these runtimes in all models.

In the daily cases analysed, it was found that the existence of errors in the NWP significantly impairs the performance of the networks by underestimating or overestimating production according to the error type. These errors are considered to be one of the main constraints in solar power forecasts, as once NWP performance is improved, they will be enhanced as well.

For another photovoltaic plant in which the same type of data is available, it is concluded that the most indicated network to use, in situations when there is a rigid time constrain, is the convolutional neural network, as it presents a reasonable performance for its hyperparameter tuning and training time. If there is more flexibility in the development time, the use of a convolutional LSTM neural network is recommended, because although DNN2 provided a better performance in the case study, the search time for manual features can be used in the hyperparameter tuning of a convolutional LSTM neural network resulting in better forecasts.

5.2 Future Work

Although the designed models that rely exclusively on automatic features perform in a similar manner to the one that depends on manually engineered features, some enhancements can be made with future work. These upgrades could lead to a significant boost of performance, retrieving forecasts with lower errors than the ones outputted by networks whose inputs are manual features. In general, the following ideas are proposed:

- Implementation of neural attention: this mechanism provides the neural network capability to focus on specific parts of inputs images linked to a determined context to extract more relevant features [61];

- Use of similar structures to the ones studied in this thesis in other sources of inputs: as CNNs and ConvLSTM neural networks were successful at extracting features from NWP data, they might be able to learn features from satellite and sky images;

- An alternative approach using Graph Convolutional Neural Networks: after building adjacency matrices between the points of the NWP grid, it is possible to extract automatically features using the suggested network [62];

- Further research on ConvLSTM networks: performing additional tests in structures whose inputs contain a higher temporal horizon might increase the overall accuracy, since it may find other relevant temporal features;

- Forecasting solar power via Gradient Boosting Trees using the automatically extracted features: simple tests, that were performed using this technique on the features extracted by the model CNN-I2, showed promising results.

Appendix A

Hyperparameter Tuning Results

The present chapter provides the hyperparameters determined on the tuning of each model. The figures A.1, A.2, A.3 and A.4 show the variables related to the layers of each model.

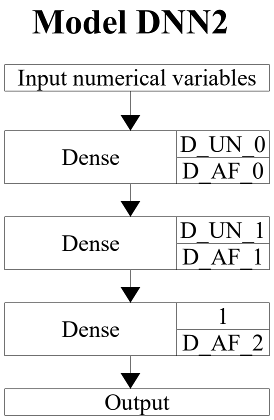


Figure A.1: DNN2 hyperparameters related to the network structure and activation functions.

The tables A.1, A.2, A.3 and A.4 contain the space search and the values of each variable retrieved by the hyperparameter tuning performed in each data fold and model.

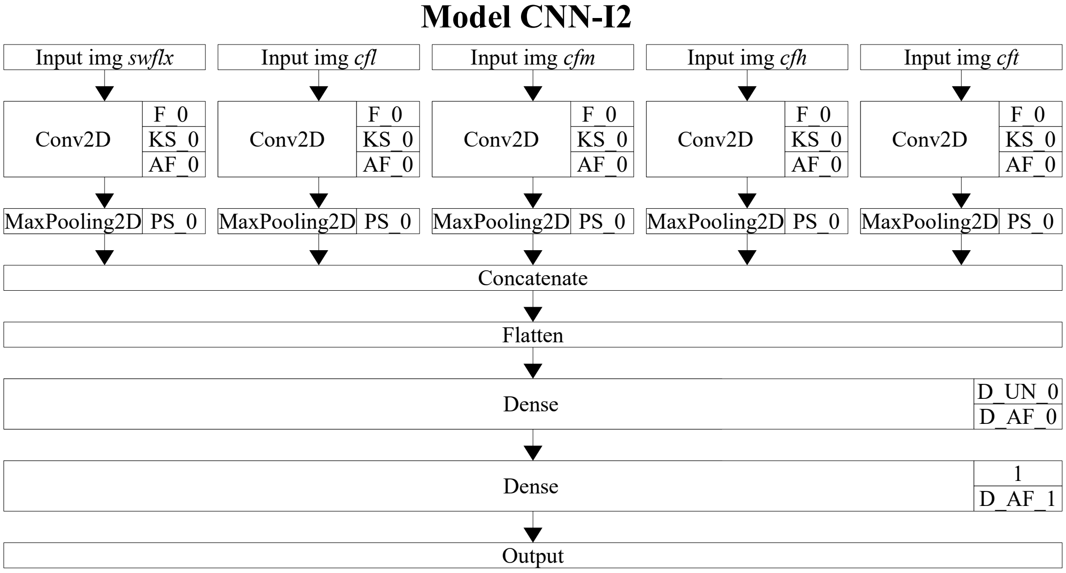


Figure A.2: CNN-I2 hyperparameters related to the network structure and activation functions.

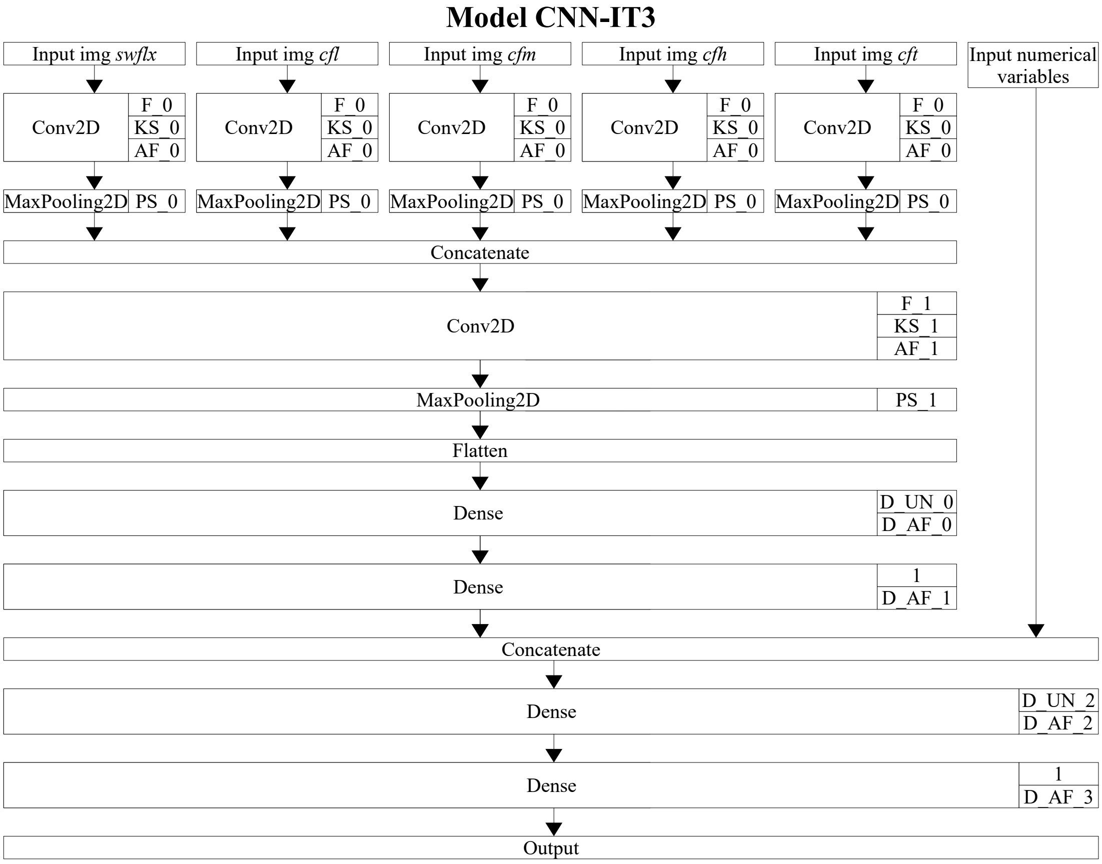


Figure A.3: CNN-IT3 hyperparameters related to the network structure and activation functions.

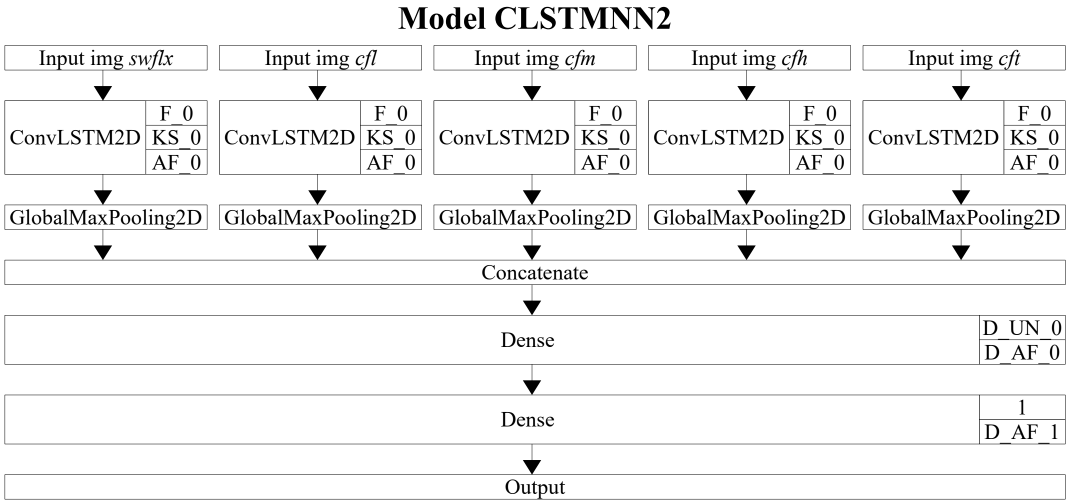


Figure A.4: CLSTMNN2 hyperparameters related to the network structure and activation functions.

Table A.1: Model DNN2 hyperparameter sets selected for each data fold.

Category	Hyperparameter	Search space	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Number of units	D_UN_0	[16, 64], step of 4	60	64	60	64	64
	D_UN_1	[16, 64], step of 4	52	40	52	8	8
Activation function	D_AF_0	ReLU, tanh, linear	ReLU	ReLU	ReLU	ReLU	ReLU
	D_AF_1	ReLU, tanh, linear	ReLU	ReLU	ReLU	ReLU	ReLU
	D_AF_2	ELU, tanh, linear	ELU	linear	ELU	ELU	ELU
Training algorithm	Optimiser	Adam					
	Learning rate	0.01, 0.001, 0.0001	0.0001	0.0001	0.0001	0.001	0.001
	Batch size	[16, 128], step of 16	16	16	16	96	96
	Number of epochs	[1, 1000]	52	80	84	65	35
Maximum number of trials = 50							

Table A.2: Model CNN-I2 hyperparameter sets selected for each data fold.

Category	Hyperparameter	Search space	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Number of filters	F_0	[4, 32], step of 2	32	32	32	26	4
Width/height of kernel	KS_0	[1, 6]	1	1	1	2	1
	PS_0	[1, 4]	4	3	4	4	4
Activation function	AF_0	ReLU, tanh	tanh	ReLU	tanh	tanh	tanh
	D_AF_0	ReLU, tanh, linear	tanh	tanh	tanh	tanh	tanh
	D_AF_1	ELU, tanh, linear	linear	tanh	linear	ELU	tanh
Number of units	D_UN_0	[8, 64], step of 4	8	64	64	64	64
Training algorithm	Optimiser	Adam					
	Learning rate	0.01, 0.001, 0.0001	0.0001	0.0001	0.0001	0.0001	0.01
	Batch size	[128, 512], step of 8	512	512	128	240	128
	Number of epochs	[1, 1000]	566	540	232	148	12
Maximum number of trials = 50							

Table A.3: Model CNN-IT3 hyperparameter sets selected for each data fold.

Category	Hyperparameter	Search space	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Number of filters	F_0	[4, 32], step of 2	26	24	4	8	12
	F_1	[4, 32], step of 2	26	18	20	4	20
Width/height of kernel	KS_0	[1, 4]	2	4	2	4	1
	KS_1	[1, 4]	2	2	4	3	1
	PS_0	[1, 2]	1	1	2	2	2
	PS_1	[1, 2]	2	2	1	2	2
Activation function	AF_0	ReLU, tanh	ReLU	ReLU	tanh	ReLU	ReLU
	AF_1	ReLU, tanh	ReLU	ReLU	ReLU	tanh	ReLU
	D_AF_0	ReLU, tanh, linear	ReLU	ReLU	linear	ReLU	ReLU
	D_AF_1	ELU, tanh, linear	ELU	ELU	linear	ELU	tanh
	D_AF_2	ReLU, tanh, linear	ReLU	ReLU	ReLU	ReLU	ReLU
	D_AF_3	ELU, tanh, linear	tanh	tanh	linear	tanh	ELU
Number of units	D_UN_0	[8, 64], step of 4	32	32	16	8	44
	D_UN_2	[8, 64], step of 4	52	64	20	52	60
Training algorithm	Optimiser	Adam					
	Learning rate	0.01, 0.001, 0.0001	0.001	0.001	0.001	0.001	0.001
	Batch size	[128, 512], step of 8	400	304	512	264	408
	Number of epochs	[1, 1000]	65	39	15	33	35
Maximum number of trials = 50							

Table A.4: Model CLSTMNN2 hyperparameter sets selected for each data fold.

Category	Hyperparameter	Search space	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Number of filters	F_0	[4, 32], step of 2	4	4	30	4	8
Width/height of kernel	KS_0	[1, 6]	6	6	1	6	1
Activation function	AF_0	ReLU, tanh	tanh	tanh	tanh	tanh	tanh
	D_AF_0	ReLU, tanh, linear	ReLU	ReLU	tanh	ReLU	tanh
	D_AF_1	ELU, tanh, linear	linear	linear	linear	linear	tanh
Number of units	D_UN_0	[8, 64], step of 4	48	48	56	16	20
Training algorithm	Optimiser	RMSprop					
	Learning rate	0.01, 0.001, 0.0001	0.0001	0.0001	0.01	0.0001	0.01
	Batch size	[128, 512], step of 8	200	200	408	240	192
	Number of epochs	[1, 1000]	86	115	70	104	52
Maximum number of trials = 50							

References

[1] Potência, 2020. https://www.apren.pt/pt/energias-renovaveis/potencia.

[2] EurObserv'ER. Photovoltaic barometer. Technical report, EurObserv'ER, 2020.

[3] EurObserv'ER. Photovoltaic barometer. Technical report, EurObserv'ER, 2018.

[4] EurObserv'ER. Photovoltaic barometer. Technical report, EurObserv'ER, 2016.

[5] EurObserv'ER. Photovoltaic barometer. Technical report, EurObserv'ER, 2014.

[6] EurObserv'ER. Photovoltaic barometer. Technical report, EurObserv'ER, 2012.

[7] EurObserv'ER. Photovoltaic barometer. Technical report, EurObserv'ER, 2010.

[8] EurObserv'ER. Photovoltaic barometer. Technical report, EurObserv'ER, 2009.

[9] Can Wan, Jian Zhao, Yonghua Song, Zhao Xu, Jin Lin, and Zechun Hu. Photovoltaic and solar power forecasting for smart grid energy management. *CSEE Journal of Power and Energy Systems*, 2016.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[11] Ministério do Ambiente e da Transição Energética. *Plano Nacional Energia e Clima 2021-2030 (PNEC 2030)*. 2019.

[12] SolarPower Europe. EU Market Outlook For Solar Power / 2019 - 2023. Technical report, SolarPower Europe, 2018.

[13] Seung Seog Han, Myoung Shin Kim, Woohyung Lim, Gyeong Hun Park, Ilwoo Park, and Sung Eun Chang. Classification of the clinical images for benign and malignant cutaneous tumors using a deep learning algorithm. *Journal of Investigative Dermatology*, 138(7):1529 – 1538, 2018.

[14] Weijia Li, Haohuan Fu, Le Yu, and Arthur Cracknell. Deep learning based oil palm tree detection and counting for high-resolution remote sensing images. *Remote Sensing*, 9:22, 01 2017.

[15] V. Boyer, D. [El Baz], and M.A. Salazar-Aguilar. Chapter 10 - gpu computing applied to linear and mixed-integer programming. In Hamid Sarbazi-Azad, editor, *Advances in GPU Research and Practice*, Emerging Trends in Computer Science and Applied Computing, pages 247 – 271. Morgan Kaufmann, Boston, 2017.

[16] Aidan Tuohy, John Zack, Sue Ellen Haupt, Justin Sharp, Mark Ahlstrom, Skip Dise, Eric Grimit, Corinna Mohrlen, Matthias Lange, Mayte Garcia Casado, Jon Black, Melinda Marquis, and Craig Collier. Solar Forecasting: Methods, Challenges, and Performance. *IEEE Power and Energy Magazine*, 2015.

[17] R. J. Bessa, A. Trindade, and V. Miranda. Spatial-temporal solar power forecasting for smart grids. *IEEE Transactions on Industrial Informatics*, 11(1):232–241, 2015.

[18] Maimouna Diagne, Mathieu David, Philippe Lauret, John Boland, and Nicolas Schmutz. Review of solar irradiance forecasting methods and a proposition for small-scale insular grids, 2013.

[19] S. Jaidee and W. Pora. Very short-term solar power forecast using data from nwp model. In *2019 4th International Conference on Information Technology (InCIT)*, pages 44–49, 2019.

[20] Sophie Pelland, Jan Remund, Jan Kleissl, Takashi Oozeki, and Karel De Brabandere. Photovoltaic and solar forecasting: state of the art. *IEA PVPS Task*, 14:1–36, 2013.

[21] S. Sreekumar and R. Bhakar. Solar power prediction models: Classification based on time horizon, input, output and application. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 67–71, 2018.

[22] 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. *Solar Energy*, 84(12):2161 – 2172, 2010.

[23] M. Alanazi, A. Alanazi, and A. Khodaei. Long-term solar generation forecasting. In *2016 IEEE/PES Transmission and Distribution Conference and Exposition (T D)*, pages 1–5, 2016.

[24] Steven Miller, Andrew Heidinger, and Manajit Sengupta. Physically based satellite methods. *Solar Energy Forecasting and Resource Assessment*, pages 49–79, 07 2013.

[25] Christian A Gueymard. Clear-sky radiation models and aerosol effects. In *Solar Resources Mapping*, pages 137–182. Springer, 2019.

[26] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. J. Martinez-de Pison, and F. Antonanzas-Torres. Review of photovoltaic power forecasting, 2016.

[27] Muhammad Qamar Raza, Mithulananthan Nadarajah, and Chandima Ekanayake. On recent advances in PV output power forecast, 2016.

[28] V Kostylev and A Pavlovski. Solar power forecasting performance - towards industry standards. In *1st International Workshop on the Integration of Solar Power into Power Systems, Aarhus, Denmark*, 2011.

[29] Radian Belu. Artificial intelligence techniques for solar energy and photovoltaic applications. *Handbook of Research on Solar Energy Systems and Technologies*, 3:376–436, 01 2012.

[30] Rich Inman, Hugo Pedro, and Carlos Coimbra. Solar forecasting methods for renewable energy integration. *Progress in Energy and Combustion Science*, 39:570, 12 2013.

[31] Hugo T.C. Pedro and Carlos F.M. Coimbra. Assessment of forecasting techniques for solar power production with no exogenous inputs. *Solar Energy*, 86(7):2017 – 2028, 2012.

[32] 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)*, pages 1–6, 2013.

[33] Guozhu Dong and Huan Liu. *Feature engineering for machine learning and data analytics*. CRC Press, 2018.

[34] J. R. Andrade and R. J. Bessa. Improving renewable energy forecasting with a grid of numerical weather predictions. *IEEE Transactions on Sustainable Energy*, 8(4):1571–1580, 2017.

[35] Emil Isaksson and Mikael Karpe Conde. Solar power forecasting with machine learning techniques, 2018.

[36] Md Rahat Hossain, Amanullah Maung Than Oo, and A. B. M. Shawkat Ali. The effectiveness of feature selection method in solar power prediction. *Journal of Renewable Energy*, 2013:1, Aug 2013.

[37] Arne De Brabandere, Pieter Robberechts, Tim Op De Beéck, and Jesse Davis. Automating feature construction for multi-view time series data. In *ECMLPKDD Workshop on Automating Data Science*, pages 1–16, 2019.

[38] Changsong Chen, Shanxu Duan, Tao Cai, and Bangyin Liu. Online 24-h solar power forecasting based on weather type classification using artificial neural network. *Solar Energy*, 85(11):2856 – 2870, 2011.

[39] Fei Wang, Zengqiang Mi, Shi Su, and Hongshan Zhao. Short-term solar irradiance forecasting model based on artificial neural network using statistical feature parameters. *Energies*, 5:1355–1370, 12 2012.

[40] Gábor I. Nagy, Gergő Barta, Sándor Kazi, Gyula Borbély, and Gábor Simon. Gefcom2014: Probabilistic solar and wind power forecasting using a generalized additive tree ensemble approach. *International Journal of Forecasting*, 32(3):1087 – 1093, 2016.

[41] Gonzalo Farias, Sebastián Dormido-Canto, Jesús Vega, Giuseppe Rattá, Héctor Vargas, Gabriel Hermosilla, Luis Alfaro, and Agustín Valencia. Automatic feature extraction in large fusion databases by using deep learning approach. *Fusion Engineering and Design*, 112:979 – 983, 2016.

[42] A. Gensler, J. Henze, B. Sick, and N. Raabe. Deep learning for solar power forecasting — an approach using autoencoder and lstm neural networks. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2858–2865, 2016.

[43] Kazutoshi Higashiyama, Yu Fujimoto, and Yasuhiro Hayashi. Feature extraction of nwp data for wind power forecasting using 3d-convolutional neural networks. *Energy Procedia*, 155:350 – 358, 2018. 12th International Renewable Energy Storage Conference, IRES 2018, 13-15 March 2018, Düsseldorf, Germany.

[44] S. Khan, H. Rahmani, S. A. A. Shah, M. Bennamoun, G. Medioni, and S. Dickinson. *A Guide to Convolutional Neural Networks for Computer Vision*. Morgan & Claypool, 2018.

[45] Felix Gers, Nicol Schraudolph, and Jürgen Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of Machine Learning Research*, 3:115–143, 01 2002.

[46] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading, 2016.

[47] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai kin Wong, and Wang chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting, 2015.

[48] Solar power forecasting: measurements and numerical weather predictions, 2016. https://rdm.inesctec.pt/dataset/pe-2020-002.

[49] Peder Bacher, Henrik Madsen, and Henrik Aalborg Nielsen. Online short-term solar power forecasting. *Solar Energy*, 83(10):1772 – 1783, 2009.

[50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[51] François Chollet et al. Keras. https://keras.io, 2015.

[52] Jojo Moolayil. *Learn Keras for Deep Neural Networks*. Apress, 2019.

[53] Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Keras Tuner. https://github.com/keras-team/keras-tuner, 2019.

[54] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.

[55] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[56] Solar calculation details, 2016. https://www.esrl.noaa.gov/gmd/grad/solcalc/calcdetails.html.

[57] Zulfiqar Memon, Fahad Samad, Zafar Awan, Abdul Aziz, and Shafaq Siddiqi. Cpu-gpu processing. *International Journal of Computer Science and Network Security*, 17:188–193, 09 2017.

[58] Steven W. D. Chien, Stefano Markidis, Chaitanya Prasad Sishtla, Luis Santos, Pawel Herman, Sai Narasimhamurthy, and Erwin Laure. Characterizing deep-learning i/o workloads in tensorflow. *2018 IEEE/ACM 3rd International Workshop on Parallel Data Storage Data Intensive Scalable Computing Systems (PDSW-DISCS)*, Nov 2018.

[59] Epyc™ 7351p single socket server processor | amd. https://www.amd.com/en/products/cpu/amd-epyc-7351p.

[60] Geforce® gtx 10 series graphics cards. https://www.nvidia.com/en-us/geforce/10-series/.

[61] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention, 2015.

[62] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs, 2016.