
Universidad del País Vasco

Euskal Herriko Unibertsitatea

UNIVERSITY OF THE BASQUE COUNTRY UPV/EHU

DOCTORAL THESIS

# Application of machine learning techniques to weather forecasting

*Author:*
Pablo Rozas Larraondo

*Supervisors:*
Prof. José A. Lozano
Prof. Iñaki Inza Cano

*Dissertation submitted to the Department of Computer Science and Artificial Intelligence of the University of the Basque Country (UPV/EHU) as partial fulfillment of the requirements for the PhD degree in Computer Science.*

Donostia, October 24, 2018

# Declaration of Authorship

I, Pablo Rozas Larraondo, declare that this thesis titled, "Application of machine learning techniques to weather forecasting" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

*"Climate is what we expect, weather is what we get."*

Mark Twain

UNIVERSITY OF THE BASQUE COUNTRY UPV/EHU

# *Abstract*

Computer Science Faculty
Computer Science and Artificial Intelligence Department

**Application of machine learning techniques to weather forecasting**

by Pablo Rozas Larraondo

Weather forecasting is, still today, a human based activity. Although computer simulations play a major role in modelling the state and evolution of the atmosphere, there is a lack of methodologies to automate the interpretation of the information generated by these models. This doctoral thesis explores the use of machine learning methodologies to solve specific problems in meteorology and particularly focuses on the exploration of methodologies to improve the accuracy of numerical weather prediction models using machine learning. The work presented in this manuscript contains two different approaches using machine learning. In the first part, classical methodologies, such as multivariate non-parametric regression and binary trees are explored to perform regression on meteorological data. In this first part, we particularly focus on forecasting wind, where the circular nature of this variable opens interesting challenges for classic machine learning algorithms and techniques. The second part of this thesis, explores the analysis of weather data as a generic structured prediction problem using deep neural networks. Neural networks, such as convolutional and recurrent networks provide a method for capturing the spatial and temporal structure inherent in weather prediction models. This part explores the potential of deep convolutional neural networks in solving difficult problems in meteorology, such as modelling precipitation from basic numerical model fields. The research performed during the completion of this thesis demonstrates that collaboration between the machine learning and meteorology research communities is mutually beneficial and leads to advances in both disciplines. Weather forecasting models and observational data represent unique examples of large (petabytes), structured and high-quality data sets, that the machine learning community demands for developing the next generation of scalable algorithms.

# *Acknowledgements*

There are so many people I would to thank for helping me and supporting me along these years. I will try to keep this list as brief as possible.

- To both my PhD supervisors Prof. Jose A. Lozano and Prof. Inaki Inza Cano for their guidance and patience.
- To the Australian National University and the National Computational Infrastructure for their support in carrying out the last part of this research.
- To my family and friends for being there during these years offering their unconditional love.

# Contents

# List of Figures

*I would like to dedicate this work to Kate, Amaia and Tomas who, one by one, came into my life during this time and made it all possible.*

# Chapter 1

# Introduction

## 1.1 Contents and structure

This doctoral thesis is conceived with the objective of publishing its research outcomes on peered reviewed publications, as a way of measuring its relevance and impact within the meteorological and machine learning communities. At the moment of the defense of this thesis, two manuscripts have been published, another manuscript was presented in a workshop held as part of a well known machine learning conference and a third paper has been submitted to a weather forecasting journal. This document aims to provide the reader with an understanding of the context and research lines pursued by the author during the completion of this work.

This first chapter introduces the reader to the field of weather forecasting and its intersection with machine learning. It provides context about the theory behind numerical weather simulations and a historical perspective about the evolution and current state of weather models. Chapter 2 provides an introduction to the main research lines established for this doctoral work addressing the motivating challenges and main scientific outcomes. Chapter 3 reflects on the overall contributions presenting new avenues and lines of work that we consider promising but did not have the opportunity to explore further. Four appendixes at the end of this thesis contain the original material published by the different journals and conference proceedings.

## 1.2 The origins and evolution of weather forecasting

Weather forecasting is defined as the application of science and technology to predict the conditions of the atmosphere for a given location and time in the future (Vasquez, 2009). Historically, humans have tried to understand the behavior of the atmosphere by studying the different patterns and relationships between phenomena relating them with future events. For example, it is well known since centuries ago that a sudden descent in the barometric pressure was often followed by precipitation events.

In 1922, L. F. Richardson proposed the use of basic fluid mechanics equations to model the movements of the atmosphere. At that time, there was not way to automate calculations, so the author came up with the idea of splitting the surface of the Earth in cells and using persons to solve the differential equations that describe the movements of the atmosphere. According to his estimations, an army of 64,000 human computers would be required to generate an updated forecast for the whole planet (Richardson, 1922). Figure 1.1, represents the building conceived by Richardson to host his idea of a human "forecasting factory". Unfortunately, due to

the proportions of this project, Richardson's idea was never implemented and the availability of a global weather forecast had to wait two more decades.

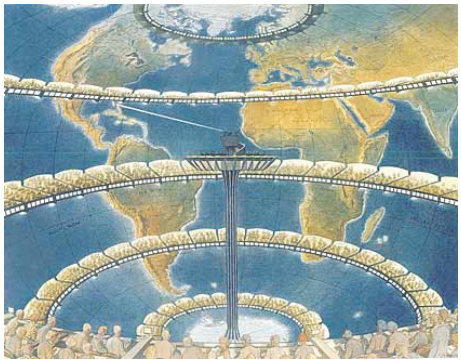

FIGURE 1.1: Representation of Richardson's idea of a "forecast factory", that would employ some sixty-four thousand human computers sitting in tiers around the circumference of a giant globe solving the equations to forecast the weather. (Source: http://cabinetmagazine.org/issues/27/foer.php).

At the end of World War II, when electronic computers and atmospheric radio sounding data became available, the United States led an ambitious project to implement the first automated forecasting system. In 1950 meteorologists Jule Charney, Agnar Fjörtoff, and mathematician John von Neumann published a paper named "Numerical Integration of the Barotropic Vorticity Equation" (Charney, Fjörtoft, and Neumann, 1950), which establishes the basis for computer weather models setting the foundations of Numerical Weather Prediction (NWP) as we know it today. This work resulted in the implementation of the first weather forecast by electronic computer. The model was implemented and run operationally on the ENIAC computer, at the University of Pennsylvania. Due to the capacity limitations of this computer, it took 24 hours of processing time to generate a 24 hour forecast, which limited its applications, as it could not effectively predict the future.

Since the 1950s computers have been used to simulate the state and evolution of the atmosphere using Numerical Weather Prediction models (Kimura, 2002). These models use a set of nonlinear differential equations to approximate the state and evolution of the atmosphere, which are known as the primitive equations. These primitive equations define the conservation of mass, momentum and thermal energy. The primitive equations are solved by NWP models using finite difference, or spectral methods, for the three dimensions of the space and time. NWP models initialise these equations using observed data, to create a snapshot of the state of the atmosphere. This process is known as "analysis" (Courtier, Thépaut, and Hollingsworth, 1994). To integrate the different parameters simulated through these equations, the Earth is divided in a discrete grid representing the evolution of the different regions of the atmosphere through time. Figure 1.2 represents the distribution of grid points in a regular Gaussian grid.

NWP models are therefore built using mathematical equations that describe the dynamical physical processes in the atmosphere. To simulate the future state of the atmosphere the same set of equations are solved iteratively using the output of the previous step. This process is repeated until the solution reaches the desired forecast time. However, errors in the simulated variables accumulate through time and the accuracy of the computed forecast deteriorates at each step. Using too coarse grid

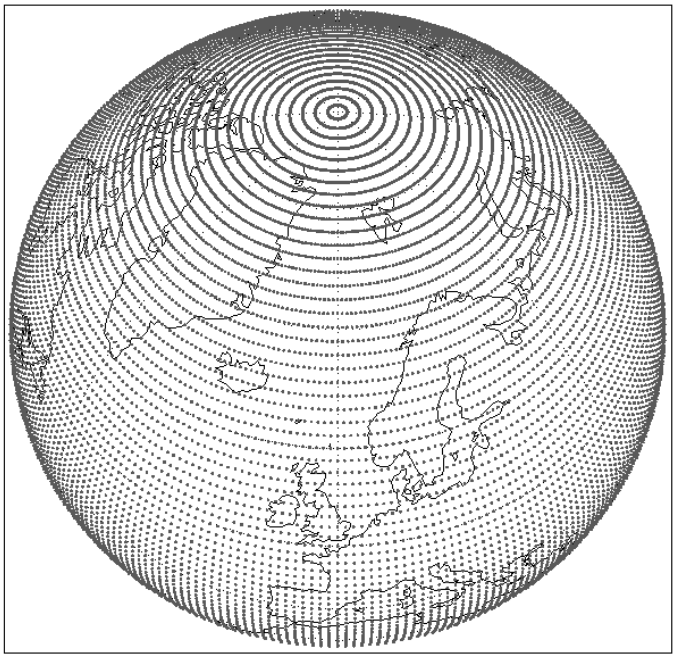

FIGURE 1.2: Example of an F80 regular Gaussian grid used by some NWP to represent the Earth's atmosphere (Source: https://www.ecmwf.int).

cells or long integration time steps are the main causes for NWP errors. The contribution of the small scale or sub-grid processes, which are not explicitly resolved by the physical equations in the model, becomes significant. Increasing the spatial and temporal resolution of NWP partially solves this problem but at the cost of significantly increasing the computational requirements of the model.

One of the main constraints limiting NWP is the resolution of the grid used to resolve the equations, which is usually in the order or kilometers. This resolution is usually smaller than the natural scale of some important atmospheric processes. Processes such as convention, radiative transfer or cloud formation happen at smaller scales than models can explicitly resolve. For these small-scale or complex processes, NWP models use simplified or approximated processes called parameterisations (Milton and Wilson, 1996; Delage, 1997).

Parameterisations are one of the most complicated components of NWPs. Parameterising small-scale processes correctly becomes crucial when forecasting events more than 48 hours in advance, when the effect of these processes usually becomes significant. Parameterisations are usually built using simplified models to approximate atmospheric processes. These models are often defined by discretising a process into different categories, where different linear models are applied in each case. Parameterisations for different processes are sometimes closely related and have to be specified in conjunction resulting in non-trivial interactions and dependencies. Figure 1.3 contains a representation of the output of the National Oceans and Atmosphere Administration (NOAA) Climate Forecast System (CFS) representing the global atmospheric precipitable water, which is a parameterised variable.

Another consequence of the NWP grid resolution is the limitation in accounting for the effect of topography. Global NWP models use a coarse representation of the shape and features of the surface of the Earth, averaging coastlines and mountain

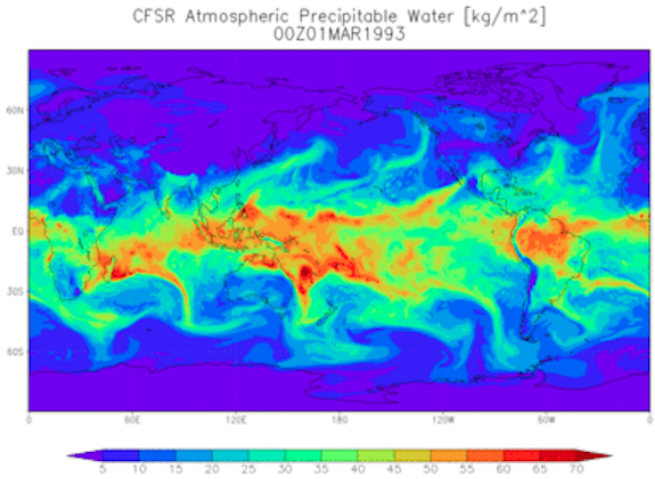


FIGURE 1.3: NOAA CFS output for the global atmospheric precipitable water on March 15, 1993 for a 12-hour accumulation period (Source: https://www.ncdc.noaa.gov).

heights over the extent of the corresponding grid cell, which is normally in the order of 10 to 100 kilometers. As a consequence, they are unable to represent important local effects that happen at a sub-grid scale.

Another major constraint that physical NWP models suffer is the difficulty to represent the chaotic nature of the atmosphere (Lorenz, 1982). The mathematical equations governing the dynamics of the atmospheric flows are nonlinear and they represent non-stable hydrodynamical and thermodynamical processes. Small differences in the initial state can amplify as the system evolves resulting in significantly different scenarios. NWP models are initialised in the analysis phase using observational data which has an intrinsic uncertainty associated with them. This uncertainty can be propagated through time resulting in a variability in the results. The instability of the atmosphere defines an upper and lower bound on the predictability of instantaneous weather patterns.

At the beginning of the 1990's, the meteorological community proposed ensemble forecasting as a methodology to measure the predictability of the atmosphere using NWP (Molteni et al., 1996). Instead of making a single high resolution simulation, ensemble forecasts run a set of forecasts with slightly different initial conditions. This set of forecasts provide an indication of the range of possible future states of the atmosphere and possible scenarios. Figure 1.4 represents the evolution of the temperature variable represented as a transformation in the shape of its probability distribution through time. The different members normally aggregate around different values which correspond to the most likely scenarios for a given physical parameter.

As we have seen, the representation of non-linear subgrid processes and the quantification and propagation of uncertainty are central to the process of weather forecasting. The former is closely related to the scale of the equations used to represent the atmospheric physical processes and the latter to the precision of the sensors and the confidence we have in their measurements.

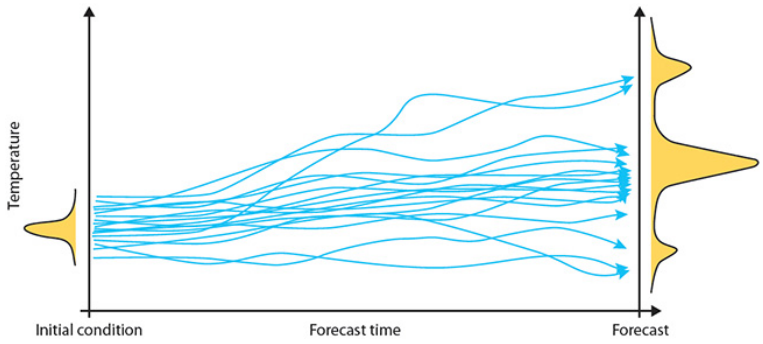


FIGURE 1.4: An ensemble of forecasts produces a range of possible scenarios given an initial probability distribution of a forecasted parameter. The different ensemble members provide an indication of the possible resulting scenarios based on a probability distribution. (Source: https://www.ecmwf.int).

## 1.3 Numerical weather prediction

As introduced in the previous section, computers have been used since the 1950's to simulate the state and evolution of the atmosphere. Ever since this time, capacity of the computers has doubled every 18 months following Moore's Law and similarly the resolution of the weather models. Although the physical equations and methodologies for resolving them have remained fundamentally the same as in the first weather models, the spatial and temporal resolution as well as the frequency of the runs of the models has been constantly increased up until nowadays. In terms of the accuracy – skill in the weather forecasting literature – of the forecasts, NWP performance has improved by two days per decade, as shown in Figure 1.5. This means that the accuracy of a 4-day forecast today is as good as it was 10 years ago for a 2-days forecast.

The field of NWP verification has been extensively studied in the field of weather forecasting (Ahijevych et al., 2009). The need for methods that can measure the accuracy and quality of the information produced is fundamental to identify weaknesses in the simulation of the atmospheric processes. The two baselines that are often used to measure the quality of the forecasts are persistence in short range (0-2 days) forecasts and climatology in mid- (2-5 days) to long-range (5+ days) forecasts. Persistence is the assumption that the meteorological conditions are going to stay the same. In this context a NWP model needs to be better at forecasting the weather than a simple model that keeps the variables constant in time. The climatology baseline, on the other hand, assumes that the weather is going to behave as the averaged historical records for that particular area.

One of the main limitations of NWP models has been the lack of observations over large regions of the world, which limited the initialisation of the initial state of models, which is called "analysis". For a long time, the conditions over the Pacific ocean or the southern hemisphere were vastly unknown due to a lack of ground station and atmospheric sounding data. This limitation has been greatly improved since the introduction of satellites and remote sensors. Since the 1990s satellite data has become available and the assimilation of these data by NWP models has resulted in an significant improvement in the quality of their forecasts. This is a trend that is expected to continue, as new satellites equiped with new and more capable sensors become available. For example, in August 2018, a joint initiative of the European

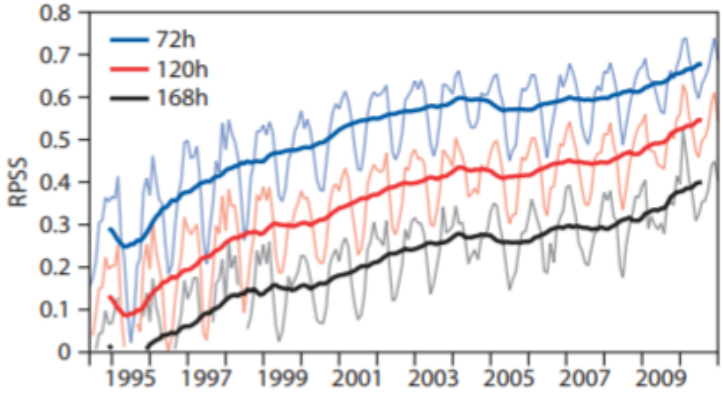


FIGURE 1.5: The performance of the EPS has improved steadily since it became operational in the mid-1990s, as shown by this skill measure for forecasts of the 850 hPa temperature over the northern hemisphere at days 3, 5 and 7. Comparing the skill measure at the three lead times demonstrates that on average the performance has improved by two days per decade. The level of skill reached by a 3-day forecast around 1998/99 (skill measure = 0.5) is reached in 2008-2009 by a 5-day forecast. In other words, today a 5-day forecast is as good as a 3-day forecast 10 years ago. The skill measure used here is the Ranked Probability Skill Score (RPSS), which is 1 for a perfect forecast and 0 for a forecast no better than climatology. (Source: https://www.ecmwf.int).

Union and the European Space Agency (ESA) has launched Aeolus, the first satellite able to monitor the Earth's winds. This satellite alone is expected to significantly increase the accuracy of NWP models by providing valuable information of the winds with a global coverage.

The synoptic scale in meteorology (also known as large scale or cyclonic scale) is a horizontal length scale of the order of 1000 kilometers or more. NWP models have been able to simulate the presence and evolution of the synoptic atmospheric systems since the beginning. This comprises phenomenons such as mid-latitude depressions, fronts and most high and low-pressure areas seen on weather maps. As the size of the grid cells used to simulate the weather has been reduced, models have been able to simulate atmospheric processes happening at lower scales. Current models are able to explicitly solve the processes that occur at the mesoscale level, which comprises phenomena that manifest at scales ranging 5 to 100 km, such as sea breezes, squall lines or medium to large-sized convective cells. Current global weather models operate at a resolution that ranges 25 to 100 km which allows for partially accounting for some of these mesoscale systems. There are important phenomena such as convection, turbulence, radiative processes or raindrop coalescence that occur at the micro-scale level, which comprises scales smaller than 5 km. NWP models cannot explicitly solve the physical equations ruling these processes.

The physical variables simulated by NWP can be separated in two main types, depending on the nature of the equations used to simulate them. Basic parameters are the ones explicitly computed by NWP solving the physical equations. Examples of these are atmospheric pressure, wind, temperature or humidity. Derived parameters are the ones that are not explicitly solved by NWP but are derived from basic parameters using parameterisations. Precipitation, convection, heat radiative processes or turbulence are examples of derived parameters.

Due to the relatively coarse resolution of NWP models, there are considerations to be made relatively to the representativeness of its variables and their interpretation at specific grid cells. Figure 1.6 compares the topography of Europe using two different resolutions. As it can be seen, the level of detail varies significantly between both representations and important details in mountainous and coastal areas cannot be represented using coarse representations. This implies that NWP models cannot account for the impact that topography has on weather variables at scales lower than its grid cell's size.

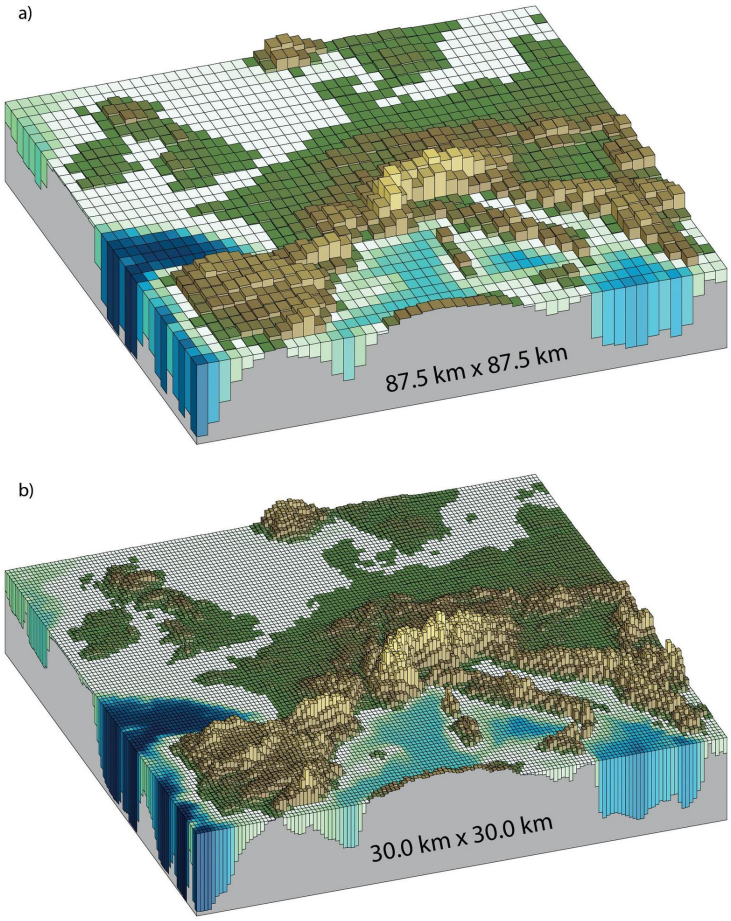


FIGURE 1.6: Comparison between the horizontal resolutions used in global models today [30 km] (a) and the equivalent models 10 years ago [87.5 km] (b). (Source: http://www.climatechange2013.org).

A common problem in NWP data interpretation is to infer high-resolution information from low-resolution variables. This process is called "downscaling" in disciplines such as meteorology, climatology or remote sensing. This process can be based on dynamical or statistical approaches such as splines, kriging or nesting into higher resolution models (Peng et al., 2017). Machine learning based methods lie in the category of statistical methods, and they can be used to improve the output of NWP learning and extracting patterns of the relationship between the model's historical output and their corresponding observed values.

NWP models are complex systems which are usually developed by large organisations during decades. Their development requires a high degree of specialisation

on different areas and entire teams of highly skilled scientists are dedicated to different components of the model. Models are often designed using modules which can be run with a certain independence from the rest of the system. However, the atmosphere is a complex system in which most of its variables are interlinked defining dependencies and feedback processes between them. NWP are highly optimised to run on High Performance Computing (HPC) facilities using Fortran and C and libraries such as OpenMP (Dagum and Menon, 1998) and MPI (Gropp et al., 1999) to distribute the computation between large compute clusters and accelerate the generation of its output.

## 1.4 The sources of weather data

So far, we have focused our attention on NWP, as this is currently the only tool available that is able to "forecast" the evolution and future estate of the different physical parameters describing the atmosphere. However, NWP output is not the only source of information when studying the atmosphere. Sensors, under many forms and characteristics, provide accurate values of the observed conditions in a region of the atmosphere. Examples of sensors used in weather forecasting are: ground stations, atmospheric sounding balloons, weather radars or satellites. Each of these types of sensors has different characteristics in terms of the spatial and temporal resolution and the extent that they can cover.

Observational data sets are crucial in the process of weather forecasting as they provide a live stream of information describing the current state of the atmosphere. Forecasters use these data operationally to validate the NWP output and to correct for possible errors or local effects.

Specially relevant to this thesis work are METARs, which are weather observation reports generated in most of the airports in the world. These reports are used to plan air traffic control in airports and are one of the highest quality observed data sources available. The following code represents a METAR report for the airport of Donostia, Spain. The code starts with the International Civil Aviation Organization code of the airport, date of the report, wind conditions, visibility, cloud coverage, temperature and finishes with the pressure conditions.

```
LESO 071119Z 31013KT 280V340 3000 RA FEW018 SCT033 BKN040 14/12 Q1020
```

NWP models also combine all the different sources of observed data to establish the initial conditions of the model. This process, which is known as "analysis phase", is a complex process in which all the different sources of information, together with the previous output of the NWP model, need to be combined in a consistent manner. Figure 1.7 represents some of the most important sensor data that are used to initialise NWP models.

There are mainly two techniques used to perform the analysis of NWP models: Ensemble Kalman Filter (EKF) (Burgers, Leeuwen, and Evensen, 1998) and 4-Dimensional Variational assimilation (4D-Var) (Courtier, Thépaut, and Hollingsworth, 1994). These techniques basically integrate the different variables across the space and time minimising a cost function that measure the difference between the observed values and the ones in the NWP.

Historical observational data sets are kept in the records of the different weather organisations for performing climatological and research studies. A special kind of NWP models, called climatic or re-analysis models, simulate the state of the atmosphere in the past instead of in the future. These models are produced as a way

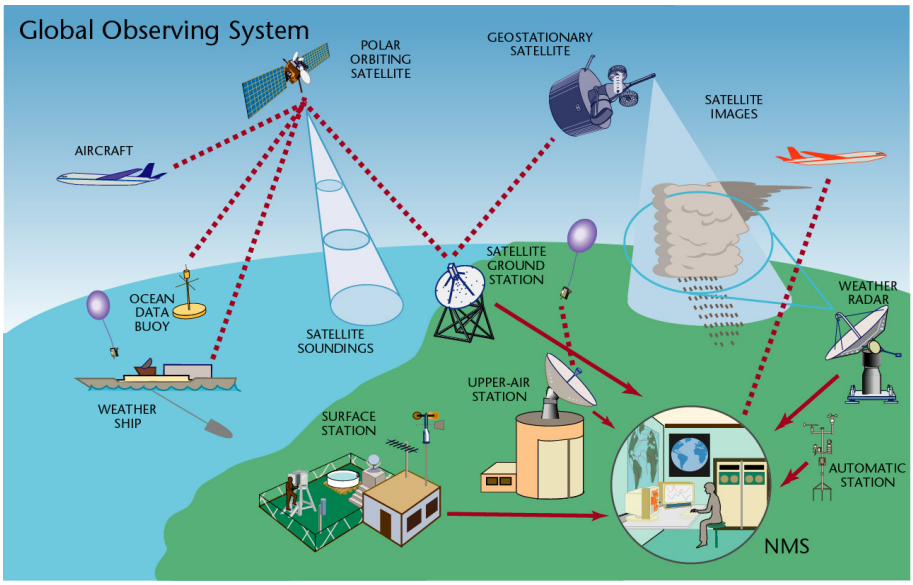


FIGURE 1.7: The Global Observing System (GOS) consists of a network of synoptic surface-based observations made at over 11000 land stations, by about 7000 ships and 750 drifting buoys at sea and around 900 upper-air stations, together with reports from aircraft and remotely sensed data from geostationary and polar orbiting satellites. (Source: http://www.wmo.int/pages/prog/www/OSY/GOS.html).

of having a consistent dataset that represents the weather of the past. These data sets are mainly used for performing research studies about the evolution of the atmosphere during the last years or centuries, depending on the temporal extent and resolution of the model. Examples of these models are ERA-Interim (Dee et al., 2011) and CMIP5 (Taylor, Stouffer, and Meehl, 2012).

Weather agencies around the world provide access to weather forecasts, severe weather warnings, observations, flood information and climate information to society. The amount of information generated by these centres is very large and its storage and management requires high capacity infrastructure to support it. The volume of some of these collections is in the order of Petabytes of information. For example, ERA5, one of the latest reanalysis data sets, has generated nearly 6 Petabytes of data. Although these data sets have been historically maintained in tape storage systems, there is an increasing demand to consume these data in real-time, which requires high-performance storage file systems (Evans et al., 2015).

Weather data is mostly represented using multidimensional numerical arrays. There are specific file formats, such as NetCDF-4 (Rew, Hartnett, and Caron, 2006) or HDF5 (Folk et al., 2011), which are optimised to provide fast access to the whole data or subsets of it implementing advanced compression techniques to reduce the size of the files. These formats also implement metadata standards to ensure the compatibility and interoperatibility of the files between organisations.

The high volumes of data generated at simulating the atmosphere and an increasing interest in these data by sectors such as agriculture, air and maritime transport or renewable energies, pose a challenge on how to make these data available. Currently, large investments are dedicated to infrastructure to facilitate the access to weather data by the public. Multiple national and international efforts, such as the European Copernicus program (*Copernicus Europe's eyes on Earth*), are currently

focused on designing and implementing new systems capable of processing and extracting value out of weather and climate data.

More generally, the ready availability of large datasets about the Earth has awaken the interest on combining different sources into new studies and analysis, coining the *data fusion* term (Wald, 1999). One of the biggest challenges when combining data from different sources is the heterogeneity of the data. Data coming from different NWP models or satellites is represented using different combinations of grids, units, projections, spatial and temporal resolutions and file formats. The process of data fusion is normally preceded by a laborious and error prone process to homogenise the data into a common representation. This process is commonly known as *data wrangling* (Goldston, 2008) and, unfortunately, is a common activity in weather sciences. For example, downscaling of NWP output constitutes an active field of research in meteorology which usually requires the combination of low resolution gridded NWP data with other higher resolution sources such as local weather stations or Earth Observation (EO) (Giebel et al., 2011; Renzullo, Sutanudjaja, and Bierkens, 2016).

During the last decade there has been an increasing interest coming from private companies into weather and Earth observation datasets. Weather datasets have been recently made available by large cloud computing companies, such as Amazon or Google. Also, these companies are investing on systems to access and process these large datasets using their infrastructure (Gorelick et al., 2017; *Earth on AWS*). This offers a new avenue for the use of these data, which has been historically dependent on the support of large national agencies and computational infrastructures.

These initiatives are popularising and democratising the access of the general public to weather data sets. However, the Cloud presents quite different architecture when compared to traditional High Performance Computing (HPC) centres. Systems, models and file formats have been developed during the last half century based on mature and stable technologies, such as POSIX file systems and the x86 CPU architecture. There is nowadays an unprecedented opportunity to redesign the systems that will bring weather data into new applications and ubiquitous uses in society.

This transition making weather data more accessible is happening at the same time to the development of new machine learning and large scale analytics algorithms. Due to the size of these datasets, computers have replaced humans as the main consumers of data. This trend is expected to continue with the release of new algorithms capable of analysing weather data and produce on-demand added value products (*Cloud AI*; *AWS Machine Learning*).

## 1.5 Machine learning

The term "machine learning" dates back to the middle of the last century. In 1959, computer scientist Arthur Samuel defined machine learning as "the ability to learn without being explicitly programmed." (Samuel, 1959). Machine learning can also be seen as a particular way to solve a more generic task expressed with the concept of "artificial intelligence", which involves machines being able to think and reason in a similar way to what humans do. The concept of "artificial intelligence" predates the one of "machine learning" by a few years. In 1950, Alan Turing published a groundbreaking paper with titled "Computing machinery and intelligence" (Turing, 1950), in which the question of whether machines can think was formally raised. In

1956, John McCarthy, computer scientist at Stanford University, used this term to express the idea that machines can simulate any form of human learning.

There are different ways of defining and representing the intersection between "artificial intelligence" and "machine learning". Depending on the author and scientific field, these terms are often found in conjunction with others, such as "computational statistics" or "data science". In the context of this thesis, we use the term "machine learning" to refer to the field of computer science that uses statistical techniques to give computer programs the ability to "learn" from data. This section introduces some of the main areas and algorithms in machine learning, with special emphasis in those that have been applied in this doctoral work to solve specific weather forecasting problems.

One of the main challenges in machine learning is to design generic algorithms that can find meaningful representations in the data. In practice, input data is usually adapted through a series of transformations to suit the requirements of a specific algorithm. For example, designing an algorithm that is able to extract the time from images of wall clocks is a non-trivial problem. If the same wall clock images are processed resulting in a new dataset that contains the angles of the clock hands for each image, the complexity of the problem is reduced significantly. This problem is called representation or feature learning (Bengio, Courville, and Vincent, 2013).

*Representation of features becomes a central topic of research in the first half of this thesis. Weather data and specifically wind data, is naturally represented as vectors defining the speed and directional components. The directional component is represented by a circular variable that, as opposed to linear variables, is not bounded. If represented in radians, this means that a circular variable with the value of 0 represents the same point in the variable space as $2\pi$. Most of the machine learning algorithms are not designed to work with circular variables and fail to represent them correctly. In our work, we explore methods to improve the representation of circular variables through use cases that contain wind direction, time and calendar date variables.*

From the point of view of the nature of the problems that machine learning algorithms try to solve, we can differentiate two main categories of problems: supervised and unsupervised. In supervised problems (Russell and Norvig, 1995), the learning algorithm is provided with training data that contains the output values and it extracts relationships or patterns present in the data. Once the model is trained, it can be used to predict the output of new input data samples, which label or output value is unknown. On the other hand, unsupervised learning (Hastie, Tibshirani, and Friedman, 2009) represent problems where the output remains unknown at the training stage of the model. These algorithms perform an exploratory analysis on the data trying to find its inherent structure or relationships between the input samples. It is used to draw inferences from data sets consisting of input data without labeled responses.

A central application of unsupervised learning is in the field of density estimation in statistics,[1] though unsupervised learning encompasses many other domains involving summarizing and explaining data features. Examples unsupervised problems are clustering, with algorithms such as k-means (Forgy, 1965) or dimensionality reduction with algorithms such as mixture models (Day, 1969) or

neural network methods such as deep belief networks (Hinton, 2009) and autoencoders (Hinton and Salakhutdinov, 2006).

Supervised problems, at the same time, can be divided in two categories: regression and classification, depending on the nature of the output variable to be predicted. In regression, the output is represented using a continuous variable whereas classification problems the output variable contains a discrete number of possible values or labels. Both classification and regression problems can be formally expressed using vectors of random variables to represent the input and output data. For the purposes of this work, we develop the theory around regression problems, but similar equations can be derived for classification problems, if we consider the output variable to be a discrete random variable instead of continuous.

A supervised regression problem can be described generically by the correspondence between a set of $n$ continuous predictive variables $\mathbf{X} = (X_1, \ldots, X_n)$ and a set of $m$ continuous explanatory or output variables $\mathbf{Y} = (Y_1, \ldots, Y_m)$. When both $n$ and $m$ are greater than one, the problem is called "multivariate multiple regression", or commonly, "multivariate regression". Each sample of the random vector $(\mathbf{X}, \mathbf{Y})$ is represented by $(\mathbf{x}, \mathbf{y}) = (x_1, \ldots, x_n, y_1, \ldots, y_m)$. The space defined by each input variable $X_i$ is denoted by $\mathcal{X}_i$ and the space of each output variable $Y_j$ is denoted by $\mathcal{Y}_j$. Therefore, the space defined by the random vector $(\mathbf{x}, \mathbf{y})$ is $(\mathcal{X}, \mathcal{Y}) = \mathcal{X}_1 \times \ldots \times \mathcal{X}_n \times \mathcal{Y}_1 \times \ldots \times \mathcal{Y}_m$ contains the set of all possible instances of $(\mathbf{x}, \mathbf{y})$.

A regression model defines a function $\Phi$ that maps each instance of input vector space into the output variable space:

$$\begin{aligned} \Phi : \quad & \mathcal{X} \to \mathcal{Y} \\ & (x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_m) \end{aligned}$$

Similarly, a classification problem, maps the input space into the space defined by a set of discrete variables $\mathbf{C}$.

However, it is not always possible to have a fully supervised data set during training, which introduces the concept of weakly (or semi-) supervised problems (Chapelle, Scholkopf, and Zien, 2009; Hernández-González, Inza, and Lozano, 2016). Weakly supervised learning deals with scenarios in which data sets present missing or inaccurate labels or target values. Rather than a binary differentiation between supervised and unsupervised problems, there exists a continuum of categories that cover a spectrum between both.

The subject of this thesis has been focused on exploring regression methods, so the rest of this section covers the different methodologies used to perform regression, with special mention to the algorithms explored in the proposed journal publications.

The most basic algorithm to perform regression is linear regression (Neter, Wasserman, and Kutner, 1989), which uses a linear function to relate the input or independent variables with an output or dependant variable. In mathematics, a linear system can be solved when we know a number of equations that is equal to the number of variables in the system. Linear regression presents the problem of having a data set normally containing a much larger number of points than variables in the data,

which can be seen as solving an over-determined system. Liner regression finds a solution to such system by considering a linear function that relates the different variables and minimises the overall error when predicting the output.

The following equation represents the linear relationship between one output element of the dataset and a set of $n$ input variables:

$$y = \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon$$

There are many approaches to solve linear regression models, which are all based on minimising the error, represented by the $\epsilon$ value in the equation, across the whole dataset. Least-squared methods are normally used to perform linear regression. Most implementations require a matrix inversion which makes the computational cost of these methods grow with the dimension of the data. An alternative approach is to use iterative methods, such as gradient descent, to make the model converge minimising the error value.

Linear regression models, although simple and effective in many situations, do not provide good representations when the data presents non-linear relationships. Non-linear regression methods allow training models that are able to represent non-linear relationships between the input variables. There are many methods to define non-linear methods such as Taylor or sinusoidal series, or by segmenting the space and using a series of linear regression models, also called piece-wise regression.

Non-linear regression models are more capable of representing relationships in the data than linear models. However, the larger representational capacity of non-linear regression models can lead to data representations with excessive detail. If the model learns to reproduce, with high precision, the relationships in the training dataset, sometimes is an undesirable effect as it can fail to generalise new data points, unseen during the training phase. This problem is known as overfitting (Hawkins, 2004) and its effect is represented in Figure 1.8. In this figure, we can see a non-linear regression model accurately representing all the points in a data set, as opposed to a linear model that approximates its values. In some cases, the linear model can provide a better representation of the problem than the non-linear version, and we'll say that the non-linear model is overfitting the data. Sometimes, overfitting can be easily solved by adding more data to the model. In other situations, avoiding overfitting requires a careful consideration about the model design and its parameters.

Another alternative for performing non-linear regression are the, so called, non-parametric regression methods. In non-parametric regression, there is not a single model that represents the whole data set. The model is constructed ad hoc for each individual case, according to the information derived from the data. Non-parametric regression models search the dataset looking for the elements that are "closer" to the input value. Therefore, they often require larger data sets than the equivalent parametric methods.

Kernel regression is a particular method of non-parametric regression. Kernel regression uses mathematical window functions, called kernels, to weight the contribution of input data points. This method estimates a continuous dependent variable from a limited set of data points, which the kernel selects by weighting the contribution of each data point, giving higher weights to nearby locations. Figure 1.9 shows seven different kernel functions which apply different weighting patterns to the data.

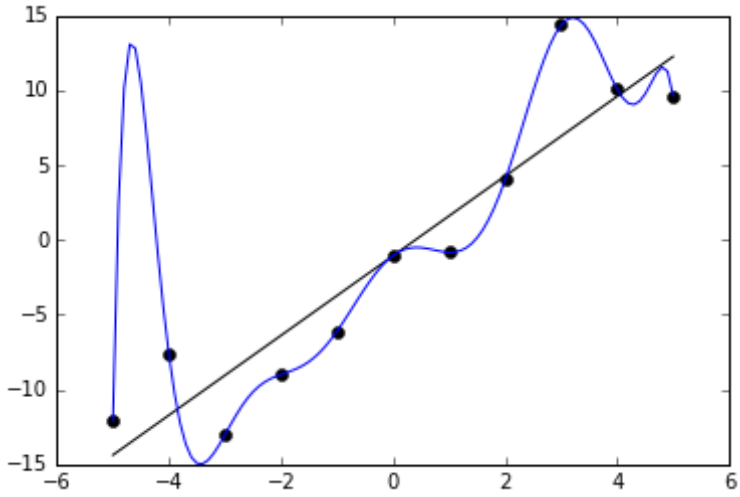


FIGURE 1.8: Comparison of a non-linear regression model (blue) and a linear model (black) representing a 2-dimensional data set. Source: Creative Commons by M. Giles

*Kernel regression is the method used in the first contribution of this doctoral thesis. It was used to resolve the local effects of local topography and improve the NWP forecasted wind speed values at airports. Cyclic kernels can be used to filter historical wind observations by their directional component. The resulting models provide a dynamic regression model that can account for the effects of surrounding physical features of an area, such as mountains or coast lines.*

We have covered some methodologies used to perform statistical regression on data based on least-squared like methods. However, there are other well known ways of building models from data that represent continuous variables. Decision trees, for example, provide a tool to model relationships in the data by partitioning the dataset space into independent regions. The algorithm of Classification and Regression Trees (CART), proposed by Leo Breiman in 1984 (Breiman et al., 1984), is based on the use of binary decision trees to generate classification and regression models.

Regression trees perform a recursive partitioning of a dataset based on a metric that maximises the homogeneity in the resulting children nodes, such as the variance. At each node of a binary regression tree, one variable is selected to perform a partition by selecting one value; typically data points with values larger than the splitting value end up in one child node and the smaller ones are assigned to the other child node. This partitioning process is performed until a stop criteria is met, usually based on the number of data points or a minimum variance value per node. When a node is no longer partitioned, it is called "leaf".

Regression trees provide an easy and intuitive way to model data. They are fast to train, scale well with large data sets and are easy to interpret and understand by looking at the decisions made at each node of the tree. These factors have contributed to their popularity, and nowadays regression tree applications can be found at nearly any field of science. There are many versions or possibilities to build regression trees and, since their invention, many authors have presented alternative methods to build trees, presented new metrics or proposed ways of pruning trees to better represent the data (Quinlan, 1993; Freund and Mason, 1999).

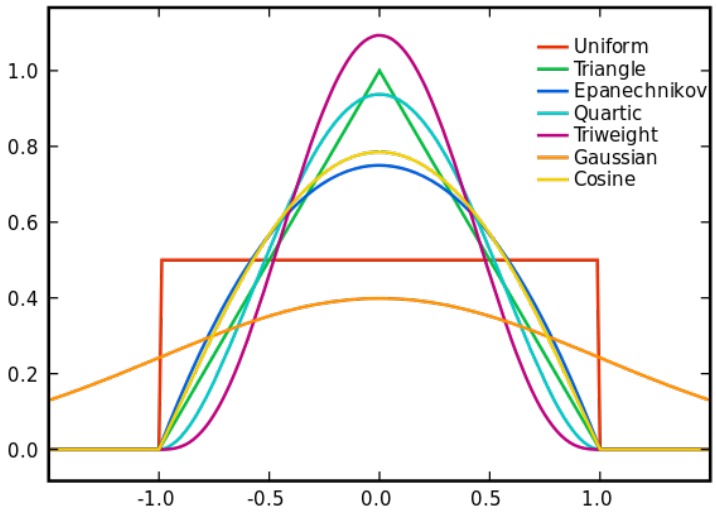


FIGURE 1.9: Comparison of the shape of seven common window functions used in kernel regression. Source: Creative Commons by Brian Amberg

Regression tree algorithms perform significantly better when trained in groups or ensembles (Breiman, 1996). Bagging, also known as bootstrap aggregation, is an ensemble tree method used to reduce the variance of individual decision trees. The different tree members of the ensemble are created using sub-data sets by randomly selecting data points, with replacement, from the training data. The average of all the individual predictions from the ensemble trees provides a more robust predictor than any of the individual regression trees. Ensemble methodologies have also become very popular and there is a wide range of algorithms available such as boosting or random forest (Dietterich, 2000; Breiman, 2001).

*The second contribution in this doctoral thesis proposes a new methodology to build regression trees that can incorporate circular variables. Based on a previous pioneering work that introduces circular trees, we build up and extend this concept with an alternative method that generates better partitions of the circular space. The methodology restricts the options to partition a circular variable by only allowing splits that generate contiguous regions, in a similar way to how linear variables are handled in classic regression trees.*

Alternatively to the covered methods, there are other kind of algorithms for performing regression that are often more capable of representing high-dimensional data sets and non-linearities in the data. Two very popular examples are Support Vector Machines (SVM) (Hearst et al., 1998) and multilayer feed-forward Artificial Neural Networks (ANN) (Hornik, Stinchcombe, and White, 1989). Both these methodologies use a series of "basis functions" to define hyperplanes and represent the data. Non-linearities in the data can be represented by using the "kernel trick" (Mika et al., 1999) in SVM and activation functions in ANN. Although there are similarities between SVM and multilayer feed-forward ANN, the former is non-parametric and has variable size whereas the latter is parametric and has fixed size, defined by the number and dimensions of its layers. Later in this section, we come back to ANNs in the context of image analysis and deep learning.

Within the area of machine learning that explores the applications of SVM and ANN methodologies, problems that involve working with structured and high-dimensional data sets are commonly known as "structured prediction" (Taskar et al., 2005). Structured prediction, focuses its attention on methodologies that deal with regular and large output spaces, such as data sets representing temporal or spatial dimensions (Gupta et al., 2010; Tran and Yuan, 2012).

In the field of spatial data analysis, computer vision has been traditionally focused on the development of methods to extract patterns from image data. Given the difficulty of coming up with generic algorithms that are able to interpret image data effectively, computer vision has been traditionally centred on developing feature engineering methods, such as SIFT (Lowe, 2004) and HOG (Dalal and Triggs, 2005), which describe the different features from basic, human-defined, building blocks. However, in the last decade, new methodologies based on the use of convolutional neural networks, have demonstrated to be more generic and offer substantial advantages over previous methodologies.

Deep Learning (LeCun, Bengio, and Hinton, 2015) methods have recently achieved unprecedented results in different supervised classification and regression tasks, using different kinds of high dimensional data sets, such as images or audio. These methods use large artificial neural networks composed of tens or in some cases hundreds of layers. Recent research has presented networks that are capable of surpassing human-level performance at different complex tasks such as image classification (Krizhevsky, Sutskever, and Hinton, 2012) or semantic description (Karpathy and Fei-Fei, 2015). Deep learning networks for image analysis are normally based on Convolutional Neural Networks (CNN) (Krizhevsky, Sutskever, and Hinton, 2012), which have been proven to be very effective at capturing intrinsic features represented at different scales of an image. CNNs learn several layers of convolutional kernels, which are able to establish local connections between nearby pixels in the image. Each layer in the network performs a sampling operation immediately after the convolution, reducing the dimensions of the image. Kernels in one layer cover larger areas than in the previous layer. The network is able to progressively aggregate the spatial features of images, creating high level representations.

Convolutional encoder-decoder networks are a type of CNN that provide state-of-the-art results at tasks such as image segmentation (Badrinarayanan, Kendall, and Cipolla, 2017), image denoising (Mao, Shen, and Yang, 2016) or image-to-image regression (Isola et al., 2017). These networks are based on autoencoders (Hinton and Salakhutdinov, 2006), which use CNNs to learn reduced but accurate representations of images, generalising its use to perform regression between images. Convolutions in the encoder half of the network perform a feature selection process by reducing the dimensionality of the data. The decoder part enlarges the feature space mapping it to the output space. Encoder-decoder networks offer an effective method for learning the relationship between high dimensional input and output spaces, such as the ones defined by images or video.

Convolutional encoder-decoder networks have recently opened an active and promising field of research in areas such as medicine (Greenspan, Ginneken, and Summers, 2016), astronomy (Shallue and Vanderburg, 2018) or high-energy physics (Baldi, Sadowski, and Whiteson, 2014). In the field of weather and climate sciences there is also an incipient interest in the introduction of convolutional networks to perform analysis and interpretation of NWP weather and climate data sets.

*Our third contribution applies convolutional encoder-decoder networks, originally designed to perform image segmentation tasks, to model the relationships between atmospheric variables. Convolutional encoder-decoder networks can learn to extract the 3-dimensional spatial structure represented by pressure fields from NWP, and predict precipitation. This application becomes specially relevant in the field of weather forecasting, as precipitation is a parameter that is not explicitly resolved by NWP and it is modeled or parameterised instead. This method provides therefore a viable alternative to generate parameterisations in NWP models.*

Within regression, there is an area specifically focused on analysing temporal series data, which is closely related to the topic of weather forecasting. The objective of these methods is to predict future values based on the short- and long-term trends and patterns in the data. This problem has been traditionally approached by the statistical community by applying auto-regressive methods, such as Auto-Regressive Moving Average (ARMA) or Auto-Regressive Integrated Moving Average (ARIMA), to predict the future state or transitions of temporal and sequential variables (Stram and Wei, 1986; Zhang, 2003). Although weather forecasting can be regarded as a time-series prediction problem, these methodologies have a limited application at predicting the weather. There are examples (Hodge et al., 2011; Eldali et al., 2016) where these techniques have been applied with satisfactory results to very short prediction windows (minutes), but in general, NWP achieves significantly better results by explicitly simulating the physical equations of the atmosphere.

## 1.6 Weather forecasting: The machine learning approach

Before the existence of NWP models, humans used a simple technique to forecast the weather based on the observed data collected over the years for a specific region, which is known as climatology. Rough estimations about long-range trends and cumulative values of variables, such as temperature or precipitation, can be inferred by applying basic statistics to observed weather events over long-enough periods of time.

At the beginning of the 20$^{th}$ century, with the advent of the technology to accurately measure the atmosphere and to communicate these values across geographically distant places, weather maps became available. These maps initially started representing the position and shape of the low and high pressure systems and allowed the development of weather forecasting methodologies, which predicted the movement and effects of these pressure system in the atmosphere. The application of statistical models to weather forecasting was proposed as early as in the 1950's (Malone, 1955). At the same time that the first NWP models were developed, Malone presented the argument that "statistics must eventually play some role" in the simulation of the atmosphere. The author demonstrated a methodology, using multiple linear regression, to forecast the sea-level pressure field. Figure 1.10 shows this first attempt of using regression methods to predict the evolution of the surface temperature at Indianapolis (USA). This model uses the atmospheric circulation pressure field in the previous 24 hours as input to forecast the next 24 hours.

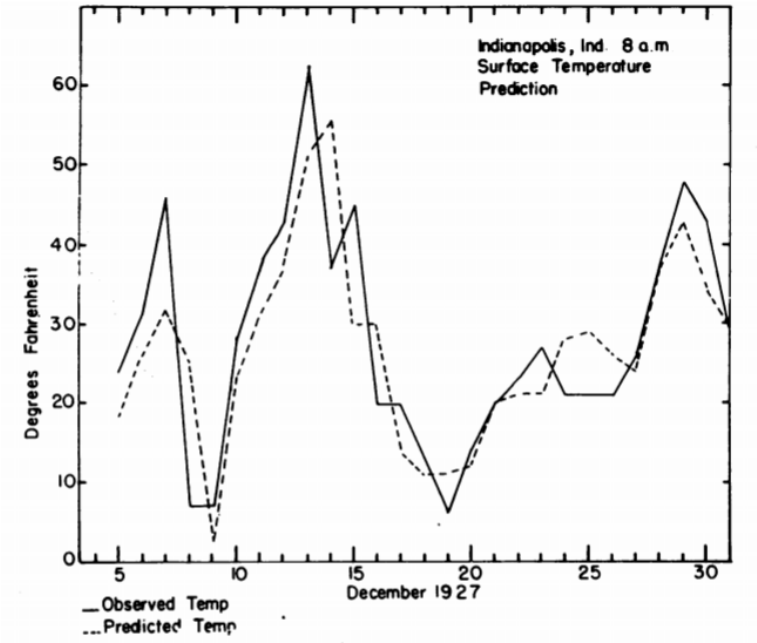


FIGURE 1.10: This figure represents a comparison between a 24-hour prediction of daily mean temperature and the observed temperature values at Indianapolis (USA). Source (Malone, 1955)

In spite of the optimism expressed by Malone in the early 1950's, the reality is that the application of machine learning and statistical methods has come as a complement of NWP rather than as an alternative to it. The power of the physical equations to simulate the evolution of the atmosphere along the spatial and temporal dimensions has not yet been matched by other methodologies. In this section, we cover examples on how different machine learning methodologies have been applied to solve different problems in the field of weather forecasting. Depending on the task to be solved and the nature of the underlying data, we can identify the following categories of problems, where different machine learning methodologies have been successfully applied to improve our understanding of the processes in the atmosphere.

- Correction of NWP systematic error: This category comprises the cases where NWP output data is post-processed to remove biases, increase the output resolution or resolve local topographic effects using observed data (Aznarte and Siebert, 2017; Buehner et al., 2010).

- Predictability assessment: Due to the chaotic nature of the atmosphere, weather forecasts have an intrinsic uncertainty value associated, which limits its value. Machine learning methods have been used to assess the uncertainty and associated confidence scores of ensemble forecasting (Wilks, 2002; Foley et al., 2012; Mallet, Stoltz, and Mauricette, 2009).

- Extreme detection: This category groups classification problems in which the outcome is the prediction or detection of a rare event. Examples of these applications are found in methods that predict phenomena such as hail, wind gusts or cyclones (McGovern et al., 2017; Williams et al., 2008; Herman and Schumacher, 2018).

- NWP parameterisations: NWP models generate multiple variables using approximate models or parameterisations to describe processes which cannot be simulated through explicit physical equations. Although these parameterisations have been historically based on empirical models, the use of machine learning is starting to grow. There are examples of machine learning methods applied to model processes, such as radiative transfer, convective and boundary-layer or turbulence (Szturc, Osrodka, and Jurczyk, 2007; O'Gorman and Dwyer, 2018; Gentine et al., 2018).

Although pure statistical based methodologies have not yet been able to replace NWP simulations in forecasting the weather, statistics have played a major role in improving the quality of the output of NWP. It is well known that forecasts from NWP models have certain defects that can be corrected by statistically post-processing their output (Wilks, 1995). The first category of problems, which try to improve the output of NWP, is the most common application of machine learning to weather forecasting. Observational data is often used to perform some sort of regression on the NWP data to enhance its accuracy.

Statistical models, used for post-processing NWP output, have evolved within three general frameworks: "Perfect Prog", Model Output Statistics (MOS), and Reanalysis (Marzban, Sandgathe, and Kalnay, 2006). "Perfect Prog" models (Vislocky and Young, 1989) use regression to represent the relationship between the initial state of NWP (analysis) variables, such as temperature or precipitation, and observations of those same parameters (Klein, Lewis, and Enger, 1959). The trained models are then used to correct NWP forecasted fields, such as temperature or precipitation, rectifying deficiencies in the NWP forecast. "perfect Prog" assumes that the accuracy of the forecast does not depend on the size of the forecasting window. In contrast, MOS fits a linear regression model between NWP output at a certain forecast time with observations at that time (Glahn and Lowry, 1972). Because MOS fits the NWP output directly, it can correct for biases and systematic errors in a model. When NWP model configurations are updated, MOS must be retrained after a sufficient number of new model forecasts are collected. "Perfect Prog" models are generally less accurate than an optimised MOS model, but they are less sensitive to model configuration changes and tend to be more robust over time. In the development of all the variations on MOS and "Perfect Prog", a limitation is the amount of data available for training the regression models (McGovern et al., 2017). The use of reanalysis data to train allows developing post-processing models similar to "Perfect Prog" and MOS without the limitation in the amount of observed data (Kalnay, 2003).

Linear regression models have been used in weather forecasting within a broader context than removing bias from NWP simulated fields. For example, regression methods have been proposed to downscale extreme precipitation events from NWP data (Friederichs and Hense, 2007). Similarly (Rozas-Larraondo, Inza, and Lozano, 2014) propose a form of parametric regression based on the use of kernels to resolve the local effects of winds non resolved by NWP. We can find other applications of regression to forecast the probability of severe weather (Kitzmiller, McGovern, and Saffle, 1995) or the maximum hail size and its probability (Billet et al., 1997). In cases where the output is not a continuous variable but a binary outcome or a set of categories, logistic regression can be used to forecast the occurrence of certain events such as convective processes (Mecikalski et al., 2015) or the selection of NWP ensemble members (Messner et al., 2014).

One of the main limitations of linear regression based models is that not all the relationships between the different atmospheric processes and its variables can be modeled using linear functions. Also, least-squared based methods used to solve linear regression, such as matrix inversion, do not scale well when a dataset presents a large number of input variables. To overcome these limitations different methods have been proposed in the literature, such as multilevel regression used to model climatic variability for non-linear processes (Kravtsov, Kondrashov, and Ghil, 2005) or quadratic regression for the generation of ensemble members (Hodyss and Campbell, 2013). In the case of having a large number of input variables or when these variables are closely related, the training process of regression models can be difficult. For example, ridge regression based techniques have been proposed as a method to select variables in multi-model scenarios (DelSole, Jia, and Tippett, 2013) where different models forecast the same variables with different levels of accuracy. Least Absolute Shrinkage and Selection Operator (LASSO) is another technique, similar to ridge regression, that has been applied to weather forecasting problems for NWP downscaling (Hofer et al., 2017) or long-range seasonal forecasting (DelSole and Banerjee, 2017). Also, Principal Component Analysis (PCA) methods have been proposed to parameterise sub-grid scale processes in the atmosphere (Godfrey and Stensrud, 2010).

Regression and classification tree based methods are a popular approach to model non-linear atmospheric processes. Before tree methods became popular in the 1980's, decision trees were introduced to provide diagnosis of upper-level humidity levels in the atmosphere (Chisholm et al., 1968). Trees can be easily scaled to train models using large data sets and with high number of input variables. The easy interpretability of the trained models has also contributed to the popularisation of these methods in weather forecasting. Decision tree based methods have proven to be a powerful tool in a wide variety of weather applications, such as the detection and diagnosis of thunderstorm turbulence (Williams et al., 2008), extreme precipitation events (Herman and Schumacher, 2018), or to represent the circular nature of wind (Larraondo, Inza, and Lozano, 2018). Figure 1.11 represents a decision tree for predicting hail precipitation (McGovern et al., 2017), which clearly communicates a method that human forecasters can follow to forecast hail.

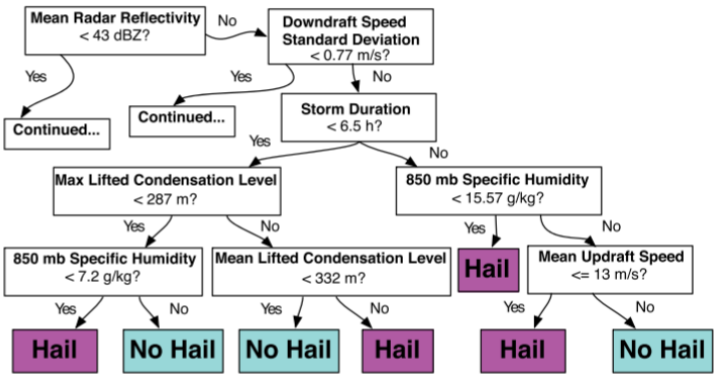


FIGURE 1.11: Example of a decision tree used to forecast the event of hail based on thresholds for different observed and NWP parameters. Source (McGovern et al., 2017)

Tree based methods are able to represent non-linearities in the data through a piece-wise approximation, by recursively splitting the input space. Artificial Neural

Networks (ANN), Support Vector Machines (SVM) or Support Vector Regression (SVR) provide a generic, more powerful alternative to modeling non-linear processes. Both ANN and SVM/SVR models are flexible and powerful, but produce models that are often difficult to interpret in terms of underlying physical concepts that the model has identified. This characteristic has limited the application of such models to weather forecasting problems, in domains where scientists require an understanding of the assumptions made by the model due to the need of coupling with other models or the need to account for inter-dependencies between variables.

ANNs suffer a similar problem where the resulting models are difficult to interpret through the weights and nonlinear activation functions. ANNs have been used in a wide variety of meteorology applications since the late 1980s (Key, Maslanik, and Schweiger, 1989), with applications such as cloud classification (Bankert, 1994), tornado prediction and detection (Marzban, Sandgathe, and Kalnay, 2006), cloud height and visibility (Marzban, Leyton, and Colman, 2007), precipitation classification (Anagnostou, 2004) or bias correction for precipitation and temperature forecasts (Moghim and Bras, 2017). ANNs have recently expanded into deep learning methods, whose applications in the field of weather forecasting are covered at the end of this section. SVM and SVR based methods have also been extensively used in weather applications such as to detect and predict tornadoes (Adrianto, Trafalis, and Lakshmanan, 2009) or for forecasting precipitation (Wei, 2012; Liu and Zhang, 2015)

More recently, deep neural network models have demonstrated to effectively and efficiently extract complex patterns from large structured data sets. Specifically, Convolutional Neural Networks (CNNs) provide a methodology to extract spatial information from image data sets. Similarly, Recurrent Neural Networks (RNNs), developed in the field of natural language processing, have found many applications in a broad range of data sets containing the temporal dimension. The combination of both models was first proposed in the context of precipitation now-casting (Xingjian et al., 2015) using radar data.

Another category of problem in weather forecasting, is accounting for the uncertainty associated with meteorological situations. Any given forecast should have associated a confidence value indicating how likely it is to happen. The chaotic and non-linear nature of the atmosphere (Lorenz, 1982) imposes a limit to our capacity to forecast its evolution. The weather forecasting community uses the term "predictability" (Palmer and Hagedorn, 2006) to express our capacity to accurately model the evolution of a certain parameter or situation.

A common approach to represent variability in the atmosphere is to use ensembles of numerical models which are constructed by running the same model by perturbing the initial conditions (Buizza et al., 2005) or by considering the output of different NWP models (Tebaldi and Knutti, 2007). The interpretation of ensemble NWP often requires the identification of groups of models or regions that present similar meteorological situations. Unsupervised methodologies, such as clustering, are used to represent the likelyhood of a forecast.

Examples of use of cluster analysis to perform ensemble analysis of weather and climate models includes: grouping daily weather observations into synoptic types (Kalkstein, Tan, and Skindlov, 1987), defining weather regimes from upper air flow patterns (Mo and Ghil, 1988; Molteni, Tibaldi, and Palmer, 1990) or grouping members of forecast ensembles (Tracton and Kalnay, 1993; Molteni et al., 1996). Clustering algorithms have also been proposed in the literature with applications to

precipitation map segmentation (Baldwin and Lakshmivarahan, 2005), precipitation distribution patterns using hierarchical clustering (Ramos, 2001), El Niño pattern identification (Johnson, 2013), or to predict typhoon trajectories (Camargo and Ghil, 2007).

Generative models, such as Variational Auto-Encoders (VAE) (Kingma and Welling, 2013) or adversarial models (Goodfellow et al., 2014) are being explored as ways to simulate variability and understand the stability of specific meteorological situations. This approach can generate variations from a basic meteorological situation similarly to the way ensemble methods operate but using a single model. Although there are not yet publications demonstrating the potential of these methods, several works exploring this idea have been presented in conferences lately.

A special and important category of problems in weather forecasting is the prediction of extreme events. There is great value in correctly identifying the occurrence of extraordinary atmospheric phenomena, which normally present a threat to human activities. This problem is normally not addressed accurately by regression methods, which tend to treat these extreme events as outliers in the training data and miss-represent their occurrence.

Several works have been presented proposing changes to regression methodologies to account for infrequent events. For example, a method based on multiple linear regression is used to detect hail size (Billet et al., 1997), support vector machines for tornado prediction (Adrianto, Trafalis, and Lakshmanan, 2009), or random forest for detection of convective cells (Ahijevych et al., 2016). However, the representation of extreme events remains a challenge in weather forecasting. Machine learning based methodologies are found to be underconfident in the estimation of these events, while classical empirical models forecasts tend to be overconfident (Herman and Schumacher, 2018). This problem of dealing with imbalanced data sets appears in the broader context of machine learning and which has lead to new proposals, such as re-forecasting analogues based techniques (Hamill and Whitaker, 2006).

Parameterisations are commonly used in NWP to represent atmospheric processes that are either too complex to be resolved explicitly, or happens at scales non resolved by the numerical model. Parameterisation is normally performed using empirical models to represent atmospheric processes such as convection or radiation. However, in the last decade statistical or probabilistic models have appeared as a viable alternative (Berner et al., 2017).

Pure machine learning methodologies have been recently proposed to parameterise moist convection (O'Gorman and Dwyer, 2018), convection (Gentine et al., 2018) or our work, submitted to the Monthly Weather Review proposing the use of deep learning convolutional networks to parameterise precipitation. All these proposals present a completely new approach compared to the traditional methodologies that have been used in NWP for the last half century. The effectiveness of machine learning based methods at representing weather phenomena and their simplicity, compared to classic approaches, open a promising future for these techniques.

An important aspect of machine learning in the field of weather forecasting is the volumes of the data involved. Although the availability of large amounts of data is an advantage, the high dimensionality of the data sets also becomes a challenge. Most examples of machine learning applications found in the literature perform a drastic simplification or reduction of the dimensionality in the data, for example, restricting the input space to individual grid points in space or time.

Creating generic machine learning models, that account for the effects of multiple input variables, usually requires the use of parametric approaches. These parametric models require training multiple versions of the same models for different points in time and space. This approach involves large processing resources and is often difficult to scale. Also, implementations of traditional machine learning algorithms, such as linear regression, tree based methods or artificial neural networks are designed to work on reduced input spaces and their complexity quickly becomes unmanageable as the number of input variables grows (Raible et al., 1999; Bowler, Pierce, and Seed, 2006). Training models on highly dimensional spaces has remained a challenge in the field of machine learning (Fan and Bifet, 2013) and has prevented the development of substantial alternatives to NWP for simulating the weather.

Recently, there has been a strong focus in the machine learning community to explore new methodologies that can be applied to large volumes of data or high dimensionalities. In particular, deep learning methods (LeCun, Bengio, and Hinton, 2015) have demonstrated state-of-the-art results using large data sets (Deng et al., 2009; Krasin et al., 2017). The application of these methodologies has also been recently explored in the field of weather forecasting with unprecedented results (Xingjian et al., 2015; Liu et al., 2016; Rasp, Pritchard, and Gentine, 2018). For example, Figure 1.12 shows the results of a deep learning model trained using a large collection of NWP pressure fields to detect jet-stream flows. This model was trained on a large cluster of compute nodes at the Lawrence Berkeley Supercomputing centre in the USA demonstrating the scalability and capacity of CNNs to extract complex spatial patterns in the data.

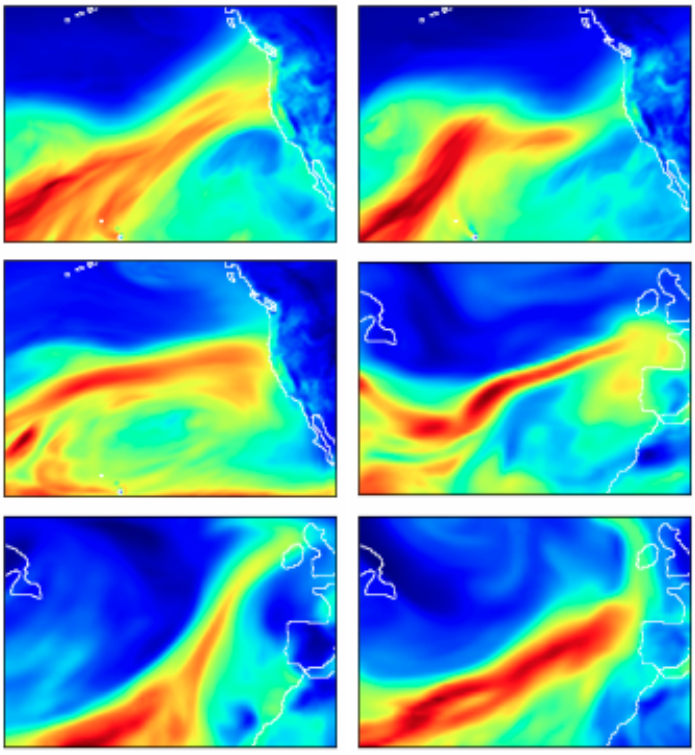

FIGURE 1.12: Sample images of atmospheric rivers (jet-streams) correctly classified and extracted from a multi-Terabyte NWP dataset by a deep CNN model. Source (Liu et al., 2016)

Even if new methodologies demonstrate capable of learning the physics ruling NWP models, it is hard to imagine that they will substitute NWP, at least in the short- mid-term. Basic fields in NWP, such as pressure or winds are accurately modelled using physical models, which are solved using highly-optimised numerical methods. The code and compilers running these models have been continuously improved during the last 60 years. Machine learning models would still need some time to achieve similar levels of efficiency in running simulations at comparable resolutions than NWP.

Regarding the trend in the volume of data generated by NWP models and observation systems during the past decades, we can foresee that the size and complexity of the data sets is going to continue growing at an exponential rate. Interpreting and analysing such volumes of data will require methodologies that are able to extract patterns from large and complex data sets, making an efficient use of the available computational and storage resources.

Assuming the existence of a system capable enough to analyse the whole collection of historical observations of the atmosphere extracting the patterns and spatio-temporal relationships between the variables, such system would be able to replace NWP entirely. In this scenario, there would be no need to understand the physical equations that rule the atmosphere, as algorithms would be able to extract models that simulate it. Figure 1.13, represents the different approaches taken by NWP and machine learning into the weather forecasting problem.

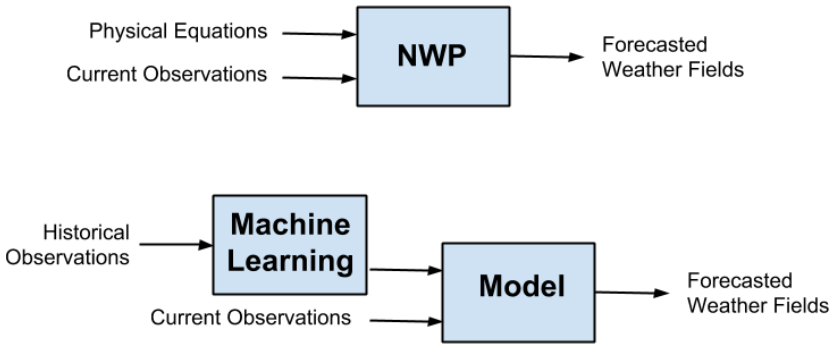


FIGURE 1.13: Comparison of the traditional NWP and machine learning approaches to weather forecasting.

NWP and observed data collections from ground stations and remote sensing sources provide a unique resource for the exploration of structured weather prediction problems. Experts working on different domains of weather and climate modeling agree that we will see a proliferation of automated statistical methods to help interpreting weather information in the near future (Jones, 2017). First, as a complement to NWP, machine learning based methods will help us to have a better understanding of complex processes occurring in the atmosphere. In a second stage, more comprehensive methodologies will become available, as the efficiency and capacity of the algorithms improves and more computational power becomes available.

# Chapter 2

# Line of research and contributions

## 2.1 Improved wind forecasting using kernel regression for circular variables

### 2.1.1 Introduction

The first part of this thesis is focused on exploring how classic non-linear regression methodologies can be used to improve the output of numerical weather models. Multivariate linear regression is a classic method for modelling the relationship between a scalar output variable and one or more input variables. This method serves in many cases as the baseline method when comparing new approaches to do regression, so we decided to start our research by exploring the most fundamental techniques.

The challenge in this case is to perform regression on the NWP wind speed variable using observed data. Wind is a weather phenomenon that is highly dependent on topography. Wind is normally represented using vectors, where the module provides the speed and the angle is the direction of the wind. Weather models decompose the wind speed variable into its Cartesian components to simplify its representation. The problem with this approach is that performing operations such as regression is not straight forward because of the relationship and dependence between both components.

For this study, we use the Global Forecasting System (GFS) NWP model and Aviation Routine Weather Reports (METARs) from three airports in northern Spain, Vitoria-Gasteiz, Bilbao and San Sebastian-Donostia. The GFS model represents data using a regular grid which covers the whole world, with a spatial resolution of approximately 50 km and a temporal resolution of 3 hours. This model is operated by the National Oceanographic and Atmospheric Administration (NOAA) and the global dataset for the latest 15 years is made publicly available. METARS are high quality meteorological reports drafted in most of the civil airports in the world. These reports are made available every 30 minutes and describe variables such as wind, temperature, visibility and cloud coverage at the airport's runway. METARS are encoded as text messages and distributed worldwide so flight control groups and pilots can plan take off and landing procedures.

For each of the three airports we selected the NWP grid points closest to the observation points, creating a temporal series that contains both the modeled and observed values for different weather parameters. We compare different linear and non-linear regression methodologies using different combinations of NWP variables in the input and using the observed wind speed as the output variable.

The reason for choosing wind speed as the output is because this variable is highly affected by the local topography surrounding the airports. Airports are normally located in open spaces with constant and wind patterns. These three airports in northern Spain are placed in a mountainous region near the Atlantic coast and have very specific wind regimes which are highly affected by the local topography. The resolution of the NWP model used in this study is in the order of 75 km, which means that these three airports are modeled in contiguous grid cells. The work is focused on studying the potential of observed wind speed data to correct the NWP values for specific locations.

### 2.1.2 Research contribution

Non-parametric regression is a category of regression analysis in which the predictor does not take a predetermined form but is built each time from the data. Non-parametric regression requires larger sample sizes than regression based on parametric models because the data must supply the model structure as well as the model estimates. This work demonstrates a technique using non-parametric regression for improving wind speed prediction by clustering wind speed data around specific directional components. This regression is performed dynamically, selecting historical data with similar characteristics. The level of similarity and the relative weights for each element in the regression is controlled using different shapes of kernels.

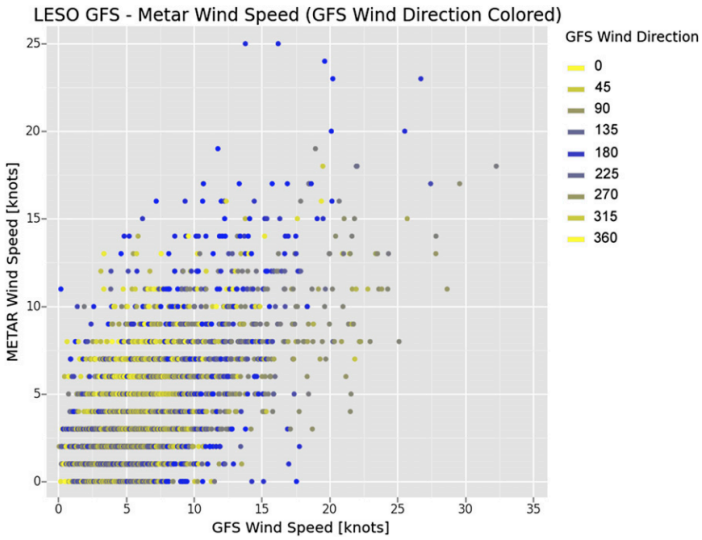


FIGURE 2.1: Relationship between GFS and METAR wind speed values from San Sebastian. GFS wind direction is represented using a color scale, with colors around yellow showing northerly winds and colors around blue representing southerly winds.

For example, Figure 2.1 contains a representation of the relationship between the observed and forecasted the wind values generated for one of the studied airports. In this case, wind direction is used in a cyclic kernel to dynamically weight the contribution of each input data point in the regression.

This work introduces the concept of cyclic kernel which provides a way of clustering elements using a circular variable. We demonstrate how circular kernels can successfully assimilate directional data into a regression model. This concept can

also extend and improve other circular or temporal variables and can be used to extract seasonal or daily patterns from data.

### 2.1.3 Publication: A Method for Wind Speed Forecasting in Airports Based on Nonparametric Regression

This work was published in the "Weather and Forecasting" journal, which is a scientific publication from the American Meteorological Society. This publication covers articles on weather forecasting and analysis techniques, forecast verification studies, and case studies useful to forecasters. It was first submitted to the journal in March 2014 and was finally published in December 2014 after a major revision process that required work on improving the statistical method used to validate the methodology.

# A Method for Wind Speed Forecasting in Airports Based on Nonparametric Regression

PABLO ROZAS-LARRAONDO

*Commonwealth Scientific and Industrial Research Organisation, Canberra, Australian Capital Territory, Australia*

IÑAKI INZA AND JOSE A. LOZANO

*Intelligent Systems Group, Computer Science Faculty, University of the Basque Country, Donostia-San Sebastian, Spain*

(Manuscript received 8 January 2014, in final form 10 June 2014)

ABSTRACT

Wind is one of the parameters best predicted by numerical weather models, as it can be directly calculated from the physical equations of pressure that govern its movement. However, local winds are considerably affected by topography, which global numerical weather models, due to their limited resolution, are not able to reproduce. To improve the skill of numerical weather models, statistical and data analysis methods can be used. Machine learning techniques can be applied to train a model with data coming from both the model and observations in the area of interest. In this paper, a new method based on nonparametric multivariate locally weighted regression is studied for improving the forecasted wind speed of a numerical weather model. Wind direction data are used to build different regression models, as a way of accounting for the effect of surrounding topography. The use of this technique offers similar levels of accuracy for wind speed forecasts compared with other machine learning algorithms with the advantage of being more intuitive and easy to interpret.

## 1. Introduction

Global numerical weather prediction (NWP) models are run with a spatial resolution that is not able to explicitly represent the effects of local topography. Several tools and methodologies have been developed for downscaling global NWP forecasts to regional or local scales. Basically, all of them could be classified as dynamical and statistical approaches. For the dynamical downscaling methods, the aim is to use a high-resolution physical model nested and initialized with the boundary conditions of a low-resolution model, which usually covers a more extensive area (Wilby and Wigley 1997). Statistical downscaling is based on statistical analysis of the output of the NWP and observational data for a location.

According to Kannan and Ghosh (2013), statistical downscaling can also be grouped into three categories: (i) weather classification/typing identifies patterns or synoptic weather schemes and analyzes data according to each case (Conway and Jones 1998; Schnur and Lettenmaier 1998), (ii) regression/transfer function techniques fit NWP and observational data using different regression and other machine learning algorithms (Sloughter et al. 2008; Sailor et al. 2000), and (iii) weather generators are based on the idea of creating a stochastic time series as a pipeline process for the different parameters (Khalili et al. 2009).

Nonparametric regression downscaling techniques are based on the idea that the predictor cannot be stated using a unique formula, but may be constructed during the execution time considering the whole dataset and selecting a subset of it to build a different regression model for every case. Nonparametric regression requires larger datasets than does regression based on parametric models, because only a limited portion of the data is used to construct the predictor each time.

Nonparametric regression has already been applied into meteorological problems for downscaling precipitation

Denotes Open Access content.

*Corresponding author address:* Pablo Rozas-Larraondo, CSIRO Black Mountain, Bldg. 5, Clunies Ross St., Acton ACT 2601, Australia.
E-mail: pablo.rozaslarraondo@csiro.au

DOI: 10.1175/WAF-D-14-00006.1

patterns (Kannan and Ghosh 2013). In this paper, a simple form of kernel nonparametric regression is used to improve forecasts of wind speed coming from the NWP, considering wind direction and wind speed variables to filter out data. This form of regression is particularly suitable for real-time forecasting, because it can be updated to include the most recent data, which makes it a perfect candidate for operational on-demand applications.

Wind statistical analysis cannot be performed directly by applying out-of-the-box machine learning or downscaling algorithms to data, because of the cyclic nature of the wind direction. The proposed regression model uses a cyclic kernel approach to select similar wind direction cases. This way of fitting circular data into a model is an approach unlike that taken by other directional statistics techniques, such as circular regression (Downs and Mardia 2002), or using generalized additive models on separate wind components (Salameh et al. 2009).

The aim of this technique is to present a method for measuring the systematic error of an NWP when forecasting wind speed. The systematic error of an NWP is mainly caused by its limited spatial resolution. If this error can be measured taking into account the difference in wind direction and how it affects wind speed, it will be possible to subtract this error from any of the leading times of the model improving its skill. The proposed method builds a nonparametric regression model to estimate the relationship between NWP-forecasted wind speed and observed wind speed when filtering the data by wind direction.

Airports are usually located in wide-open areas, with particular wind regimes favorable to air traffic. The surrounding topography affects wind behavior, by blocking, intensifying, or changing its direction as it travels, generating local wind effects not resolved by NWP. If wind speed has to be forecasted for a particular direction, grouping together similar wind direction cases to build a regression model is justified, as all are affected by the same topographic configuration.

Civil airports also offer high quality observational data that are publicly accessible. These characteristics make airports especially suitable for studying the problem of wind speed downscaling. To carry out this study, the airports of Foronda, Loiu, and Hondarribia in northern Spain have been chosen. At these sites, the wind is highly affected by adjacent steep topography and the nearby sea. Ultimately, better wind speed forecasts mean better quality terminal aerodrome forecasts (TAFORs), which implies safer air traffic operations.

All the datasets and algorithms used in this paper have been published in a public repository (https://code.google.com/p/wind-kernel-regression/) using a GNU General Public License, version 3. Any experiment contained in this article can be reproduced and freely modified.

## 2. Data sources and processing

To test the performance of the present downscaling technique, both NWP and observational data have to be collected. In the proposed regression model, the dependent variable is the observed wind speed and the independent variables are the wind speed and wind direction coming from the NWP. These data have to be represented as time series, using the same units and time resolutions for each of the selected airports.

### a. *Observations*

Observational weather data from civil airports are publicly available through aviation routine weather reports (METARs; WMO 1995), a form of coded aeronautical weather reports regulated by the International Civil Aviation Organization (ICAO). These reports are normally produced every 30 min and contain many different observed weather conditions affecting the airport at the time of observation.

Each METAR contains information such as the airport identifier, date and time of the observation, wind, cloud cover, temperature, dewpoint, and pressure, using a coded format specified by the World Meteorological Organization (WMO). To perform the tests, only the observed wind speed value is extracted along with its corresponding time stamp value for each of the airports. METAR wind speeds and directions are stated as the measured or estimated mean over the 10 min prior to the time of issue of the report. Gust wind, if present, is encoded as a separate variable and is not taken into account.

ICAO uses a four-character code to identify each airport. The ICAO codes for the selected airports in Spain are LEVT for Foronda, LEBB for Loiu, and LESO for Hondarribia. METARs for these airports are collected during the period from March 2011 to March 2013.

### b. *NWP data*

The Global Forecast System (GFS; Campana and Caplan 2013) is a global numerical weather model run operationally by the National Weather Service since March 2011 and all its historical netCDF files are available through the National Oceanic and Atmospheric Administration (NOAA) National Operational Model Archive and Distribution System (NOMADS; Rutledge et al. 2006) public repository online. GFS has a version with spatial resolution of 0.5° (~55 km) and a temporal resolution of 3 h with a new run available every 6 h. To assess the NWP systematic error, reanalysis data should be used. The use of time plus 3 h ($T + 3$) forecast data doubles the number of points available for the regression, at the expense of introducing additional inherent uncertainty into the forecasting model.

This extra uncertainty is assumed to be reasonably small 3 h away from the reanalysis and its use is compensated by the fact that the number of points used in the regression model is doubled.

For this study, a simple approach is used to extract the time series of a site. The closest GFS grid point to each airport is selected without considering any other form of spatial interpolation or correction. For each location the zonal and meridional 10-m wind speed components are extracted into a time series, corresponding to the GFS variables labeled "*U* component of wind height above ground" and "*V* component of wind height above ground," respectively. These variables contain instantaneous values measured in meters per second. Proceeding the same way as with observational data, GFS wind data are collected for the closest grid points to the airports during the period from March 2011 to March 2013.

### c. Time series

NWPs usually represent wind through Cartesian components, which is very convenient for computing averages and other statistical analysis. However, wind data coming from weather stations are normally described using their directional and speed components. For this study, wind direction is used to filter out data included in the nonparametric regression. Wind values coming from GFS have to be converted from their Cartesian components into direction and speed components before being included in the time series.

As the GFS data have been collected at a 3-h resolution and the METARs are available every 30 min, only a subset of the values can be compared. A combined time series for 0000, 0300, 0600, 0900, 1200, 1500, 1800, and 2100 UTC is created using data from the GFS model and METARs; all the extra METARs are ignored.

LEVT is the only airport of the three that records METARs 24 h a day. For this airport, 5840 data points are collected. The other two airports are closed during part of the night. METARs for 0000 and 0300 UTC are missing for LEBB, as are those for 2100 UTC from LESO, giving totals of 4380 and 3650 data rows, respectively. Table 1 shows a sample of the combined time series data for the airport at Loiu (Fig. 1).

TABLE 1. Sample time series data from LEBB on 16 May 2012, combining data from METARs and the GFS model. To save space, only three parameters are shown.

| Time stamp (UTC) | METAR wind speed (kt) | GFS wind speed (kt) | GFS wind direction (°) |
|---|---|---|---|
| 0900 | 6.0 | 12.060 | 95 |
| 1200 | 8.0 | 15.114 | 81 |
| 1500 | 12.0 | 16.094 | 79 |
| 1800 | 9.0 | 14.291 | 79 |
| 2100 | 2.0 | 13.870 | 100 |

## 3. Methodology fundamentals

An improvement in the wind speed–forecasting error could be achieved by applying a simple univariate regression model that combines wind speed values derived from METARs and the GFS model. However, NWPs contain many different physical parameters that can be included in a regression method to improve the results. Nonparametric regression techniques require large datasets to be tested, as they use subsets to calculate the regression, which is the reason for collecting 2 years of time series data in the database. Nonparametric regression is a form of regression in which the predictor cannot be expressed through a single function; rather, it calculates a new regression for each forecasted value. This kind of algorithm is especially suitable for real-time forecasting, as the regression model is built on execution time. Locally weighted regression is a form of nonparametric regression wherein values close to the forecasted point have a stronger influence in the regression. The following sections outline the basic techniques used in this regression model.

### a. Regression model

Weighted least squares is the most basic way of fitting data into a model when weighting is required. The equation in matrix notation can be expressed as follows:

$$(\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X})\boldsymbol{\beta} = \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{Y},$$

where $\mathbf{X}$ is the independent variables matrix, $\mathbf{W}$ is a diagonal matrix containing the weights of each point, $\mathbf{Y}$ is the vector containing the values of the dependent variable, and $\boldsymbol{\beta}$ is the vector containing the coefficients for each regression variable.

If a training set is defined where the values of both $\mathbf{X}$ and $\mathbf{Y}$ are known, the value of $\boldsymbol{\beta}$ can be determined and used to predict new values of the dependent variable.

A form of locally weighted regression can be implemented using a weighted least squares approach and defining a weighting function to select local points around the regression point.

The comparison between simple linear regression and locally weighted regression for wind speed data from LESO (Fig. 2) demonstrates how nonparametric regression adapts to nonlinearities in the data at the expense of the computational cost of calculating new regression parameters for each point in the plot. The weighting function used to create the locally weighted regression shown in this figure is explained in the next section.

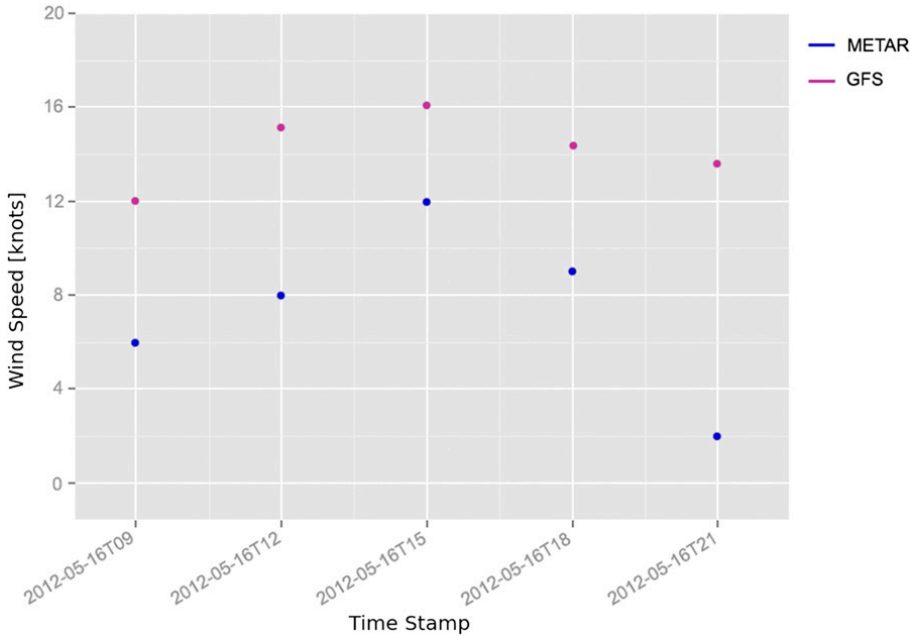


FIG. 1. Sample of time series data for METAR and GFS wind speeds from LEBB.

### b. *Kernel function*

A kernel is a well-known weighting function used in nonparametric estimation techniques to shape the influence of the different data points that take part in a nonparametric regression. There are many functions that could be used as kernels: uniform, triangle, cosine, or tricube. The tricube is one of the most popular kernels and the one used in the model proposed in this study:

$$K(d) = \begin{cases} \frac{81}{90}\left(1 - \left|\frac{d}{d_{\max}}\right|^3\right)^3, & \text{if} \quad d \le d_{\max} \\ 0 & \text{if} \quad d > d_{\max} \end{cases}.$$

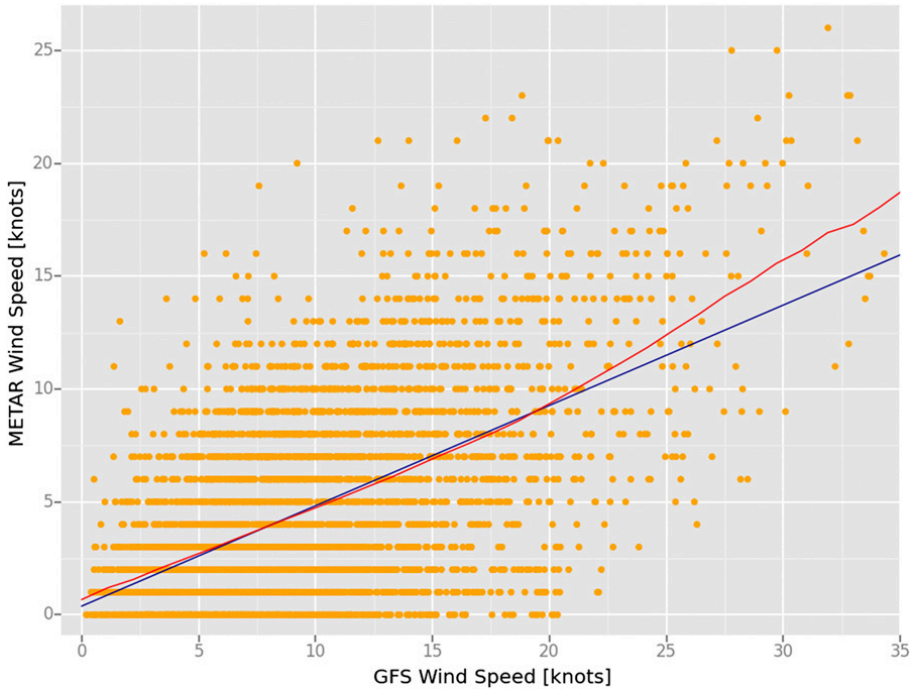


FIG. 2. Simple linear regression and locally weighted regression comparison for wind speed data from LEBB.

This function returns the relative weight of every point in the model, where $d$ represents its distance to the forecasted point and $d_{\max}$ is the maximum distance from which any point will not be included in the regression model. This kernel determines both the number of points and their relative weight in the regression model, depending on how far each point is from the forecasted value.

Figure 3 shows how the tricube kernel weights the different data points around the values of 15 and 20 knots (kt; $1\,\text{kt} = 0.51\,\text{m s}^{-1}$) to create a regression model. In this sample, the relative weight of each point in the regression decreases from its maximum value at 15 and 20 kt and becomes zero for points farther than 10 kt on each side, which is the fixed value of $d_{\max}$.

### c. *Cross-dimensional weighting*

In the previous section, there is an example of how a kernel shapes the weight of the points depending on their distance from the forecasted point. Every point considered in the regression contains many other parameters apart from wind speed. This technique is inspired by the idea of fitting regression models using historical data with similar characteristics to the day we are trying to forecast. For example, if the NWP model forecasts a wind blowing from the north, better results should be obtained when filtering the data to use the values of the database where the wind is blowing from the north. However, data can also be filtered using any other variable contained in the dataset. Wind speed can be forecasted by filtering the data points to include those showing similar characteristics to the forecasted day. In this paper wind direction is proposed as a good filtering variable but any other variable can also be used.

Kernels are used to define the weights matrix $\mathbf{W}$ used in the regression. The square matrix $\mathbf{W}$ is determined by

$$\mathbf{W} = \mathbf{I}\mathbf{K}_1\mathbf{K}_2\cdots\mathbf{K}_n,$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{K}_i$ is the $i$th kernel matrix. A kernel matrix $\mathbf{K}$ is a diagonal matrix in which each value of the main diagonal corresponds to the weight of each data point in the regression. Therefore, $\mathbf{K}$ is a square matrix with a dimension equal to the number of elements in the dataset. The weight of each value is determined by the tricube kernel function. As all the matrices used to calculate the weights matrix $\mathbf{W}$ are same-dimensional diagonal matrices, the commutative property can be applied, which means the order of the kernels does not affect the result (Fig. 4).

Figure 5 shows an example of this cross-dimensional weighting, using wind direction to filter the data. A wind direction tricube kernel selects a subset of the original points with wind directions around 0° and 180°.

## 4. Proposed methodology

The idea of considering historically similar cases comes naturally in the activity of weather forecasting. In the previous section, some mathematical tools and ideas were introduced, which can be helpful to filter out those ''similar situations'' from the whole dataset. In the particular case of wind forecasting, topography has a major influence on defining the pattern of winds at any place.

Wind direction classifies winds blowing from different places and it can be used to introduce local effects on winds. Grouping same-direction wind data together is a way of implicitly introducing the effect of local topography.

In this section, the idea of using wind direction values to forecast wind speed is explored. Kernel matrices are used as a way of weighting subsets of data into the regression model. However, the kernel function, as introduced in the previous section, is designed to work with linear variables, and the wind direction is circular.

### a. *Cyclic kernel*

Wind direction has the particularity of defining a circular space instead of a linear one. Measured in degrees, wind direction can take values in the range (0°–360°), where 0° and 360° represent the same point. To calculate a distance between two angles, the minimum of the two possible distances around the cycle must be chosen:

$$\text{Dist}(a,b) = \min\begin{cases} b-a \\ a+360-b \end{cases} \quad \text{where} \quad b \geq a.$$

This distance and a defined maximum angular distance $d_{\max}$ are used in the tricube kernel to assign weights to the different data points and derive the best estimates of wind speed. Choosing an optimal value for $d_{\max}$ is key to building an accurate regression model. Too small $d_{\max}$ values consider only small sectors of the data, which can cause overfitting and a poor generalization of the model, while, on the other hand, too large values give extremely general regressions that are not able to discriminate among the different cases.

### b. *Building and validating the model*

For each airport, wind speed and direction data from the GFS model and from METAR are used. GFS wind speed data determine the dependent variable matrix $\mathbf{X}$, METAR-observed wind speed values form the explanatory variable matrix $\mathbf{Y}$, and GFS wind direction data are used to build the weight matrix $\mathbf{W}$ using cyclic kernels.

To validate the proposed model, a methodology to test the results must be defined. A model validation technique has to be defined in order to assess how the wind forecasts generalize to an independent dataset.

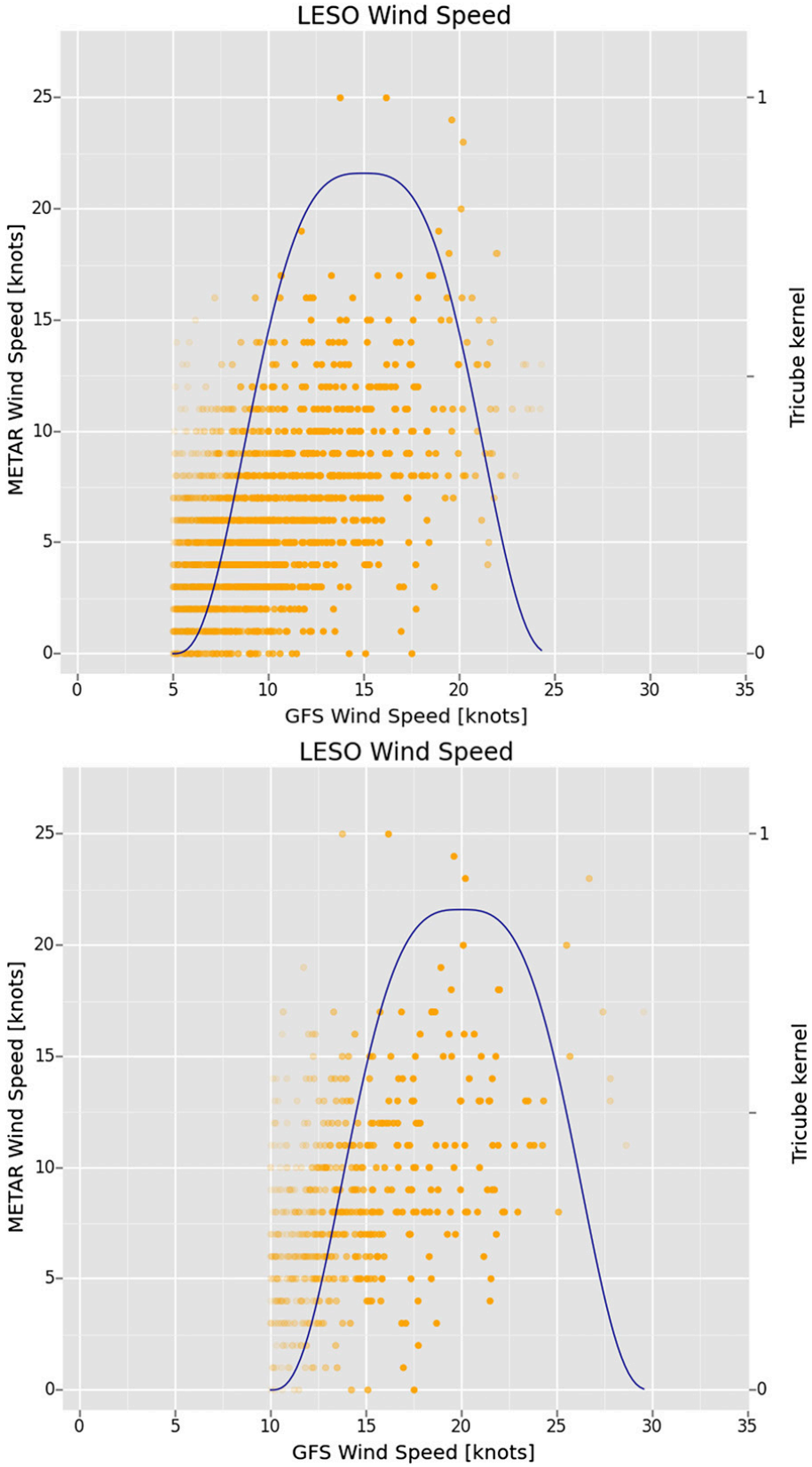


FIG. 3. Selection of wind speed points from LESO using two tricube kernels centered around 15 and 20 kt with a $d_{max}$ value of 10 kt to weight the data. Points are faded by the effect of the kernel; color intensity represents their corresponding weight in the regression.

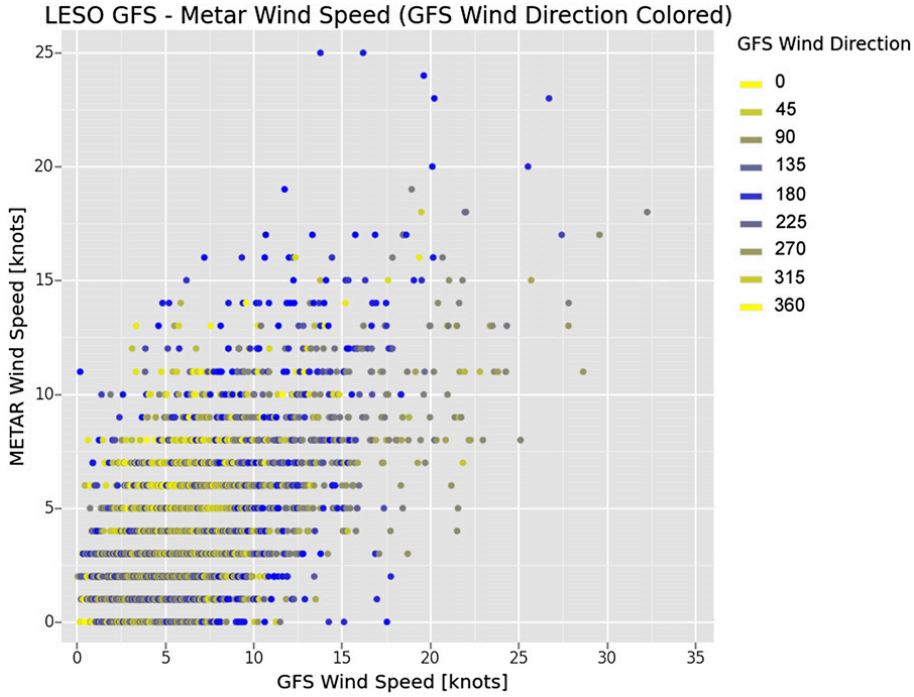


FIG. 4. Relationship between GFS and METAR wind speed values from LESO. GFS wind direction is represented using a color scale, with colors around yellow showing northerly winds and colors around blue representing southerly winds.

The holdout, cross-validation, and bootstrapping methods are different techniques for randomly splitting a dataset and validating a statistical method. The holdout approach divides the dataset into two different groups; one being used to train the statistical model and the other to validate or test the performance of the trained model. Cross validation divides the dataset into $n$ different groups and carries out the validation. One data group at a time is excluded from the training, using that excluded group to conduct testing. This process is then repeated, changing the group selected for exclusion each time, until all groups have been covered. Bootstrapping is a variation of the holdout method where each of the subsets is obtained by random sampling with replacement from the original dataset.

To test the proposed model, a repeated holdout method is chosen. The decision to use a repeated holdout method instead of a cross-validation or bootstrapping approach is made based on the size the dataset. For large datasets, the difference between randomly holding out data points or cross validating fixed subsets of the data is negligible.

For each experiment a repeated holdout estimation is performed, where the whole dataset of each airport is randomly split into two sets: one set containing 75% of the data is used to train the regression model and the other 25% is used for validation. In the training set, the observational values of wind speed are used to train the regression model and the same variable is hidden and used to estimate the error in the validation set. Root-mean-square error (RMSE) of wind speeds is used as the estimator for the error of the model. For each experiment and airport, this procedure is carried out 10 times and its RMSE values are averaged.

Different values of $d_{\max}$ used in the regression model yield different RMSE values: the larger the value of $d_{\max}$, the more points are included in the regression of each point. The experiment forecasts every observed wind speed contained in the validation set, using the data from the training to create a regression model for each value. The result obtained from the regression model is compared with the observed wind speed to estimate the error. This experiment is repeated using different values of $d_{\max}$ to identify the value or values that minimize the error of the model.

## 5. Results

The model is compared with other state-of-the-art regression techniques to assess its performance. To validate each technique, a holdout evaluation process is repeated 10 times, as explained in the preceding section. Comparisons between different methodologies are always performed using the same repeated holdout splits. The significance of the difference between compared types of regression models is statistically assessed using

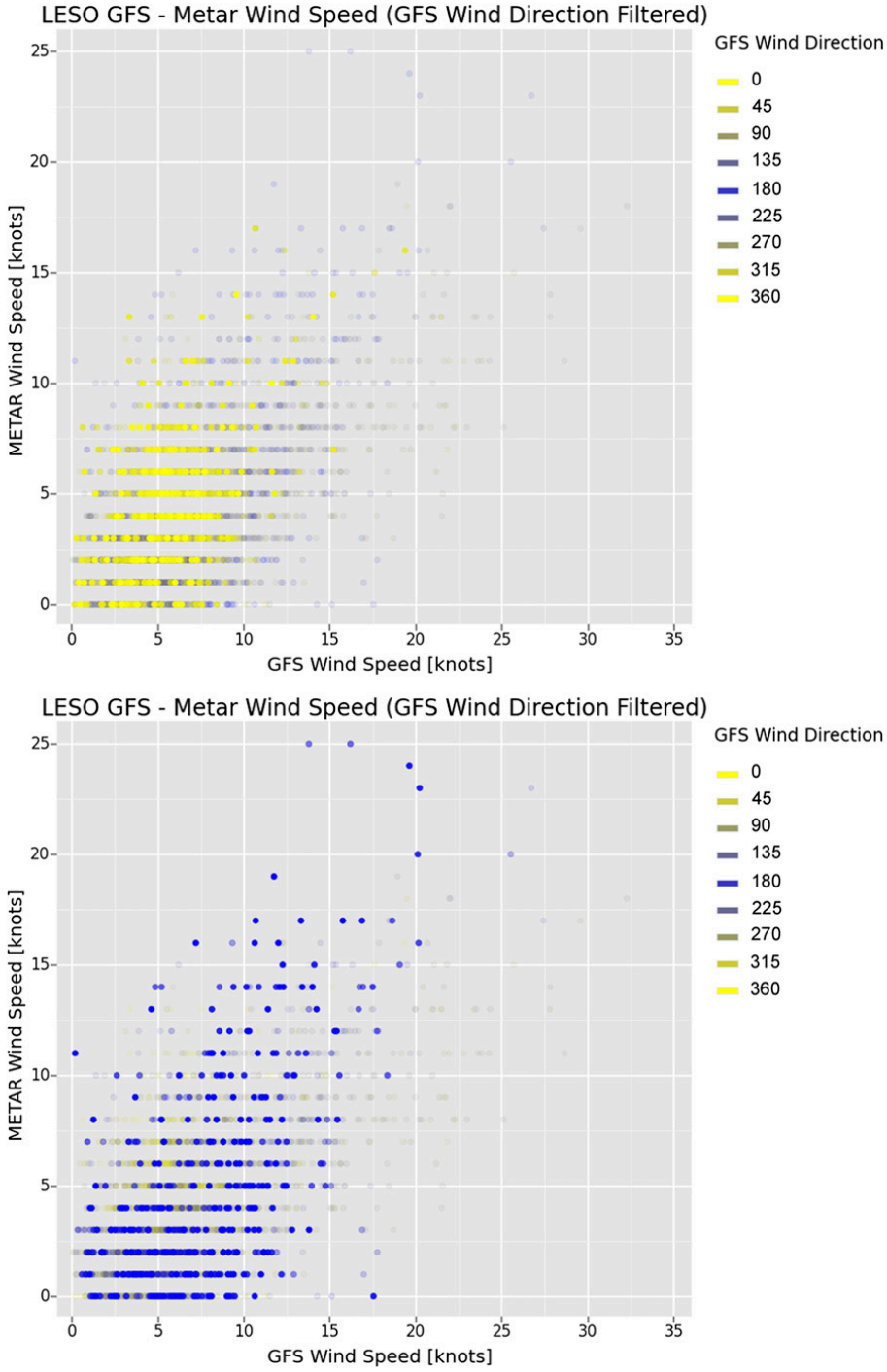


FIG. 5. Selection of wind speed points from LESO using a two wind–direction tricube kernel around 0° and 180°. A $d_{max}$ value of 35° to weight the data is used. Note the different intensities of the faded points across the plot as the wind direction is being used as a weighting variable.

a paired Student's *t* test. The use of a parametric test is justifiable, as the Shapiro–Wilks test has ensured the normality assumption of the compared RMSE samples.

First of all, a benchmark is established as a reference, so the results of the proposed regression model can be compared with it. The most basic and least accurate wind-forecasting method uses the wind speed value from the numerical weather model to predict the observed wind speed at the airport. This result could be easily improved by applying a simple linear regression to relate wind speed values from the NWP and observational data from METARs. Table 2 contains the RMSE

TABLE 2. Wind speed mean RMSE and std dev $\sigma$ results [$E$(RMSE) $\pm$ $\sigma$(RMSE)] after directly forecasting the observed wind speed using the wind speed output of the GFS model compared with the RMSE obtained when applying an univariate linear regression model containing the same data. Note that $E(\cdot)$ indicates the mean of the variable within the parentheses.

| Method | LESO | LEVT | LEBB |
|---|---|---|---|
| GFS output | 4.361 ± 0.067 | 4.015 ± 0.074 | 6.863 ± 0.107 |
| Linear regression | 2.765 ± 0.110 | 3.929 ± 0.083 | 3.560 ± 0.068 |

results of a direct comparison between wind speeds forecasted by GFS and the corresponding observed METAR values. Table 2 also contains RMSE results achieved using a univariate linear regression model to predict wind speeds for the different airports. The first result is obtained using the whole dataset, whereas in the case of the regression, the methodology explained at the end of the previous section is used to calculate the RMSE values of each airport. As shown in Table 2, linear regression introduces a notable improvement in forecasting wind speed. The proposed nonparametric regression model aims to improve these RMSE values.

Table 3 contains the results of running the experiments using different $d_{\max}$ wind direction values to filter data used in the regression, as explained in the methodology section. As indicated in Fig. 6, the three airports present a similar behavior showing minimum wind speed mean RMSE values with $d_{\max}$ ranging between 10° and 30°, but the overall improvement achieved is different for each of them.

TABLE 3. Wind speed mean RMSE and $\sigma$ results [$E$(RMSE) $\pm$ $\sigma$(RMSE)] for the different airports using different values of $d_{\max}$ wind direction in a tricubic kernel to weight the data in a nonparametric regression.

| $d_{\max}$ (°) | LESO | LEVT | LEBB |
|---|---|---|---|
| 5 | 2.758 ± 0.104 | 3.434 ± 0.059 | 3.426 ± 0.079 |
| 10 | 2.735 ± 0.104 | 3.414 ± 0.059 | 3.378 ± 0.074 |
| 20 | 2.711 ± 0.100 | 3.428 ± 0.063 | 3.370 ± 0.070 |
| 30 | 2.706 ± 0.097 | 3.459 ± 0.067 | 3.381 ± 0.069 |
| 40 | 2.706 ± 0.095 | 3.496 ± 0.070 | 3.401 ± 0.069 |
| 60 | 2.710 ± 0.095 | 3.571 ± 0.079 | 3.441 ± 0.070 |
| 90 | 2.725 ± 0.100 | 3.657 ± 0.089 | 3.487 ± 0.069 |
| 120 | 2.741 ± 0.105 | 3.711 ± 0.090 | 3.509 ± 0.069 |
| 180 | 2.760 ± 0.109 | 3.762 ± 0.087 | 3.544 ± 0.069 |

Once the value of wind direction $d_{\max}$ is found that minimizes the error, it can be fixed and a new kernel can be introduced to filter data using a different variable. The data points with similar wind directions selected by the first kernel are weighted again using a second kernel with a new variable. Using multiple kernels allows us to filter the data according to different variables and thereby achieve better results in the regression. For example, if the value of the wind direction $d_{\max}$ is fixed to 30° in one kernel, a secondary kernel can be used again to weight the resulting subset of data according to another variable.

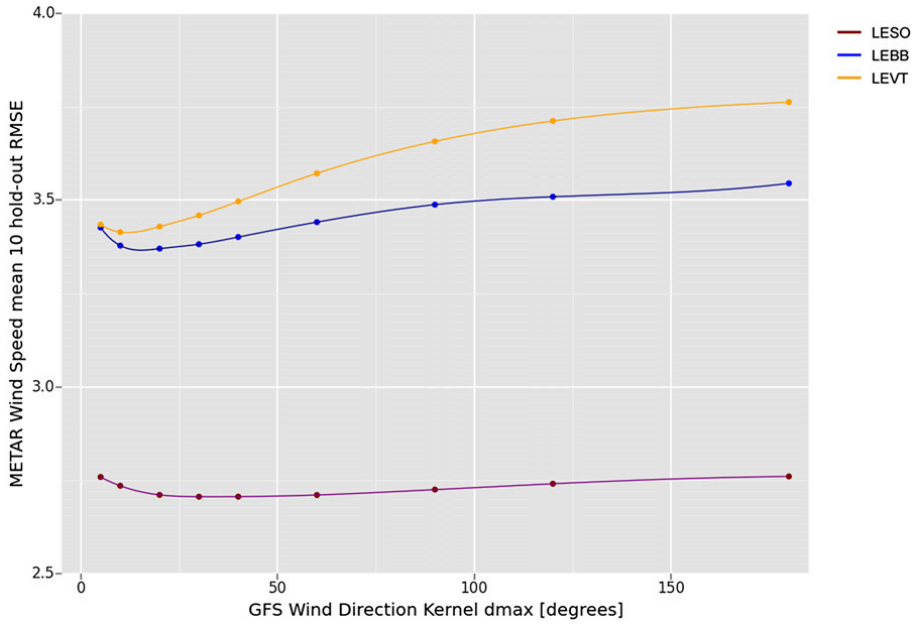


FIG. 6. Evolution of the wind speed mean RMSE for the different airports as a function of the wind direction kernel $d_{\max}$ value. Dots represent the computed values and the line has been plotted using a spline interpolation.

TABLE 4. Mean RMSE and $\sigma$ results [$E$(RMSE) $\pm$ $\sigma$(RMSE)] when using a fixed value of wind direction $d_{max}$ and different wind speed $d'_{max}$ values as kernel parameters in a nonparametric regression.

| $d_{max}$ (°) | $d'_{max}$ | LESO | LEVT | LEBB |
|---|---|---|---|---|
| 30 | 0.5 | 2.691 ± 0.093 | 3.442 ± 0.059 | 3.329 ± 0.054 |
| 30 | 1.0 | 2.666 ± 0.095 | 3.426 ± 0.062 | 3.303 ± 0.055 |
| 30 | 2.0 | 2.634 ± 0.096 | 3.419 ± 0.063 | 3.294 ± 0.054 |
| 30 | 4.0 | 2.655 ± 0.094 | 3.430 ± 0.062 | 3.291 ± 0.053 |
| 30 | 8.0 | 2.684 ± 0.092 | 3.438 ± 0.060 | 3.318 ± 0.054 |

As the objective is to improve the wind speed forecast coming from the model, GFS wind directions and wind speeds can be combined to select the data points where winds come from the same direction and have similar speed. As explained in the previous paragraph, two kernels can be applied, one using wind direction and the other using wind speed, to select the NWP points with similar wind characteristics (direction and speed) to the predicted example. Using the results of the previous experiment, which determined an optimal value of wind direction $d_{max}$ around 30°, a new experiment is proposed combining two kernels. The first kernel filters the data by its GFS wind direction, using the optimal $d_{max}$ value, and the second kernel filters the data using the GFS wind speed variable. As carried out in the previous experiment, different values for the wind speed are tested to find an optimal value that minimizes the error of the regression. To avoid confusion in the notation of the parameters of the two kernels, references to the wind speed kernel use the prime symbol. Table 4 contains the RMSE results of this experiment using different values of $d'_{max}$ in the regression model.

Analyzing the results contained in Table 4 indicates that the proposed nonparametric regression using kernels statistically outperforms the standard linear regression ($p$ value = 0.001). A pattern in the mean values can be observed, where performance improves as the value of $d'_{max}$ increases until a point where it degrades again. Different airports show different optimal values of $d'_{max}$, but all of them have a minimum in the range between 2 and 4 kt.

Applying this technique, a large number of different combinations for fitting a nonparametric regression model arise, depending on the number and type of variables, the order in which they are applied, and the shape of the kernel used.

To evaluate the performance of this wind-forecasting model, a comparison with a popular regression machine learning technique is performed. The random forest technique (RF; Breiman 2001) is used as the reference. Random forest is an ensemble learning method for regression. It operates by constructing a multitude of decision trees at training time and outputting the average of the outputs from individual trees. Every tree is trained using a random subset of the whole dataset.

TABLE 5. Mean RMSE and $\sigma$ results [$E$(RMSE) $\pm$ $\sigma$(RMSE)] for the different airports, when applying the optimized nonparametric regression model (NP) and the RF.

| Model | LESO | LEVT | LEBB |
|---|---|---|---|
| NP (30/2) | 2.764 ± 0.096 | 3.433 ± 0.063 | 3.328 ± 0.054 |
| RF | 2.682 ± 0.172 | 3.448 ± 0.117 | 3.362 ± 0.139 |

To compare the proposed model, a random forest model is used, containing 100 trees, and using wind speed and wind direction values from the GFS model, as well as observational wind speed values from the METARs. As carried out with the nonparametric regression, 75% of the data are used for training the random forests and the other 25% are used for validating the model, where METAR wind speed values are used for estimating the errors. For each airport, the process is repeated 10 times, averaging the RMSE results.

Table 5 presents very similar results in the performance of both the proposed model and the random forest technique. A paired Student's $t$ test does not show statistically significant differences between the two techniques ($p$ value = 0.36).

One important advantage of this model is that it is very intuitive, and kernels could be customized to maximize the performance of the model for each airport or location. This is a major benefit when compared with the black box results of other techniques such as random forest or neural networks. The random forest algorithm used in this experiment did not consider the circular nature of the wind direction. Therefore, an improvement in its performance would be expected if a version capable of dealing with directional data is used.

## 6. Conclusions

Different kinds of statistical postprocessing can be used to improve the performance of NWP. In some cases, when local forecasts are needed, statistical analysis and machine learning algorithms can dramatically reduce the error of the forecasted variables.

Nonparametric regression models introduce a simple and yet efficient way of representing nonlinear relationships between the variables. The use of kernels inside these models allows us to shape the influence (weight) that nearby points have on the regression depending on their proximity to the forecasted point. In this study, the use of kernels inside a nonparametric regression has been proven to work especially well when applied to wind speed forecasting in airports showing marked local wind regimes. Kernels also offer a simple way of fitting directional data into a regression model.

As pointed out in the previous section, there is a large number of possibilities for fitting data into a nonparametric regression model depending on the number of variables used as kernels to weight the data and the order in which they are applied. Solar irradiance, for example, which is available as a variable in most NWPs, also resulted in a solid filtering kernel when forecasting wind speeds. Its high correlation with wind turbulence, originating from the sun heating the surface of the earth, makes it a good way to account for daily and seasonal turbulent wind regimes. The results of using this variable are not included in this paper, as it mainly focuses on the idea of using wind direction as the main variable to determine the local effects of topography.

Cyclic kernels have proven to successfully assimilate directional data into a regression model. Other circular variables, normally contained in time-stamped datasets are time of day and day of year. The same nonparametric regression technique using cyclic kernels can be applied to these variables, extracting seasonality and daily patterns from the data. Time series analysis of airport data, as seasonal or daily pattern filtering, has not been explored in this paper but merits further research.

## REFERENCES


Breiman, L., 2001: Random forests. *Mach. Learn.,* **45,** 5–32, doi:10.1023/A:1010933404324.

Campana, K., and P. Caplan, Eds., cited 2013: ''Technical Procedures Bulletin'' for the T382 Global Forecast System. NCEP. [Available online at http://www.emc.ncep.noaa.gov/gc_wmb/Documentation/TPBoct05/T382.TPB.FINAL.htm.]

Conway, D., and P. D. Jones, 1998: The use of weather types and air flow indices for GCM downscaling. *J. Hydrol.,* **212–213,** 348–361, doi:10.1016/S0022-1694(98)00216-9.

Downs, T. D., and K. V. Mardia, 2002: Circular regression. *Biometrika,* **89,** 683–697, doi:10.1093/biomet/89.3.683.

Kannan, S., and S. Ghosh, 2013: A nonparametric kernel regression model for downscaling multisite daily precipitation in the Mahanadi basin. *Water Resour. Res.,* **49,** 1360–1385, doi:10.1002/wrcr.20118.

Khalili, M., F. Brissette, and R. Leconte, 2009: Stochastic multi-site generation of daily weather data. *Stochastic Environ. Res. Risk Assess.,* **23,** 837–849, doi:10.1007/s00477-008-0275-x.

Rutledge, G. K., J. Alpert, and W. Ebuisaki, 2006: NOMADS: A climate and weather model archive at the National Oceanic and Atmospheric Administration. *Bull. Amer. Meteor. Soc.,* **87,** 327–341, doi:10.1175/BAMS-87-3-327.

Sailor, D. J., T. Hu, X. Li, and J. N. Rosen, 2000: A neural network approach to local downscaling of GCM wind statistics for assessment of wind power implications of climate variability and climatic change. *Renewable Energy,* **19,** 359–378, doi:10.1016/S0960-1481(99)00056-7.

Salameh, T., P. Drobinski, M. Vrac, and P. Naveau, 2009: Statistical downscaling of near-surface wind over complex terrain in southern France. *Meteor. Atmos. Phys.,* **103,** 253–265, doi:10.1007/s00703-008-0330-7.

Schnur, R., and D. P. Lettenmaier, 1998: A case study of statistical downscaling in Australia using weather classification by recursive partitioning. *J. Hydrol.,* **212–213,** 362–379, doi:10.1016/S0022-1694(98)00217-0.

Sloughter, J. M., T. Gneiting, and A. E. Raftery, 2008: Probabilistic wind speed forecasting using ensembles and Bayesian model averaging. Dept. of Statistics Tech. Rep. 544, University of Washington, 20 pp. [Available online at http://www.stat.washington.edu/research/reports/2008/tr544.pdf.]

Wilby, R. L., and T. M. L. Wigley, 1997: Downscaling general circulation model output: A review of methods and limitations. *Prog. Phys. Geogr.,* **21,** 530–548, doi:10.1177/030913339702100403.

WMO, 1995: Part A—Alphanumeric codes. Manual on codes: International codes, Vol. I.1, WMO-306, Code Forms FM 15–XIV METAR, 503 pp. [Available online at http://www.wmo.int/pages/prog/www/WMOCodes/Manual/WMO306_Vol-I-1-PartA.pdf.]

## 2.2 Circular regression trees

### 2.2.1 Introduction

Most of the current regression machine learning algorithms are focused on modelling the relationships between linear variables. Circular variables have a different nature to linear variables, so traditional methodologies are not able to represent their content thoroughly, leading to suboptimal results in most cases.

Continuing with the study of methodologies for modelling circular variables, we decided to focus on regression trees for this new research study. Regression trees and tree based ensemble methods, such as bagging or Random Forest, are a versatile, computationally efficient and accurate method for performing regression. Building upon the pioneering concept of circular regression tree (Lund, 2002), we decided to explore alternative and more efficient methods for introducing circular variables in regression trees.

Traditionally, circular variables have been used in regression trees using two approaches. The first one ignores the circular nature of a variable treating it as linear variable. The problem with this approach is that 0 and 360 degrees are represented at each end of the variable range, when in reality they are the same value. The second approach is to decompose a circular variable into its Cartesian components and use them as separate variables in a tree. Both of these approaches introduce constrains and limit the performance of regression trees. With the first approach, trees cannot perform splits that cross the origin, and in the second case splits performed on the Cartesian components independently lead to an excessive and unnatural partition of the space defined by circular variables having a negative impact in the accuracy of the resulting models.

For this work we use a similar dataset to the one introduced in the previous section about non-linear kernel regression. We choose to forecast the observed speed of the wind at 5 different locations in Europe. Data from airport METARS of Berlin Tegel, London Heathrow, Barcelona El Prat, Paris Charles de Gaulle and Milano Malpensa are used to train the different tree models and to analyse the results. Similarly to the previous work we extract the time series data from the GFS model for the closest grid cells to each of these airports. The tree models are trained using three-hourly data for the years 2011, 2012 and 2013, providing approximately 8760 samples per airport.

### 2.2.2 Research contribution

The key idea behind circular trees is that circular variables can be naturally integrated in regression trees by considering splits that cross the origin [0-360] point in the search space. The original implementation of circular trees performs splits on non-contiguous regions of the variable space leading to excessive fragmentation of the dataset. The improvement proposed in our work restricts the search space to contiguous regions of the variable space leading to an improvement in computation performance and accuracy of the results. Figure 2.2 contains a representation of the splits performed by our circular tree for a dataset that contains one circular and one linear variable. The splits generated by this tree define contiguous regions in the space.

In this work we introduce a new methodology that restricts the search space of each partition in a tree generating contiguous partitions. This constraint added

to the initial idea of circular regression trees in Lund's work (Lund, 2002) has the implication of generating more simple partitions which improve the accuracy of the tree model and is more efficient to train. Although the search space for optimal splits at each node of the tree is more limited than the original version, which allows non-contiguous splits, the overall error of the tree results to be lower as the depth of the tree increases in our case. In this work we found that even if non-contiguous splits provide good results at the top part of a tree, these partitions tend to degenerate as the partition process evolves resulting in trees with poor performance.

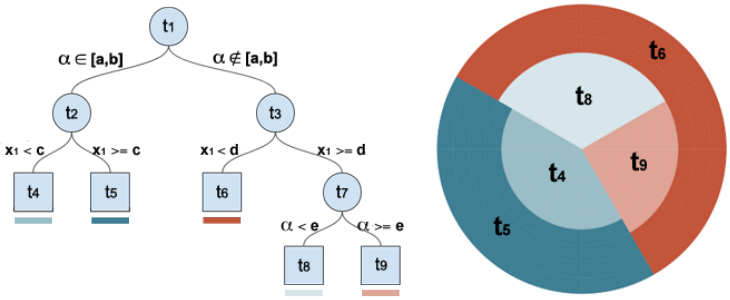


FIGURE 2.2: Example of the proposed circular regression tree and a representation of how the space is divided in contiguous regions.

The software developed in this work is presented in the form of a self-contained Python library. This library implements a general version of regression tree which can be used with both linear and circular variables in the input and output. Also, there is an option to perform non-contiguous partitions, as in the original proposal by Lund (Lund, 2002) and contiguous as in the metodology that we propose. The library also offers a command line interface that allows users to train and test different regression tree models and download datasets for any airport in the world. These tools have been designed so users can create their own forecasts experimenting and exploring the differences between models, input variables and airports. The scripts and libraries are written in a simple way so users can read the code to understand what the program is doing and also modify or extend its functionalities. AeroCirTree comes with a GNU GPLv3 licence so anyone can use, modify and share this program for any purpose.

### 2.2.3 Publication: A system for airport weather forecasting based on circular regression trees

In March 2017 it was submitted to the "Environmental Modelling & Software", a peer-reviewed scientific journal which publishes work involving modelling and software within the Environmental Science domain. The work was finally published in February 2018, after a major revision process that required extra work to demonstrate the benefit of our circular methodology in comparison to other linear trees alternatives.

Environmental Modelling & Software 100 (2018) 24–32

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

# A system for airport weather forecasting based on circular regression trees

Pablo Rozas Larraondo [a, *], Iñaki Inza [b], Jose A. Lozano [b, c]

[a] National Computational Infrastructure, Building 143, Australian National University, Ward Road, ACT, 2601, Australia
[b] Intelligent Systems Group, Computer Science Faculty, University of the Basque Country, Paseo de Manuel Lardizabal, Donostia, 20018, Spain
[c] Basque Center for Applied Mathematics (BCAM), Mazarredo 14, Bilbao, 48009, Spain

ARTICLE INFO

*Article history:*
Received 18 March 2017
Received in revised form
29 August 2017
Accepted 8 November 2017

*Keywords:*
Circular variables
Weather forecasting
Meteorology
Regression trees
Machine learning

ABSTRACT

This paper describes a suite of tools and a model for improving the accuracy of airport weather forecasts produced by numerical weather prediction (NWP) products, by learning from the relationships between previously modelled and observed data. This is based on a new machine learning methodology that allows circular variables to be naturally incorporated into regression trees, producing more accurate results than linear and previous circular regression tree methodologies.

The software has been made publicly available as a Python package, which contains all the necessary tools to extract historical NWP and observed weather data and to generate forecasts for different weather variables for any airport in the world. Several examples are presented where the results of the proposed model significantly improve those produced by NWP and also by previous regression tree models.



## Software availability

Name of software: AeroCirTree
Developer: Pablo Rozas Larraondo
Contact Address: National Computational Infrastructure, Building 143, Australian National University, Ward Road, ACT, 2601, Australia (pablo.larraondo@anu.edu.au)
Source: http://github.com/prl900/AeroCirTree
Programming Language: Python 3
Dependencies: Numpy, Pandas
Licence: GNU GPL v3

## 1. Introduction

Modern weather forecasting relies mostly on numerical models that simulate the evolution of the atmosphere, based on fluid dynamics and thermodynamics equations. These equations are solved for the discrete points of a regular grid covering the region of interest. Higher resolution models generate more detailed forecasts, but also require large computational resources and longer running times. Operational models trade off resolution quality for shorter processing times. The need for higher resolution forecasts has driven numerous methodologies to generate more detailed outputs, which is known as downscaling. Dynamic downscaling uses the output of a coarser model as the initial condition of a higher resolution local model, which better resolves sub-grid processes and topography (Carvalho et al., 2011). Another approach is statistical downscaling, where historical observed data are used to enhance the output of a numerical model. There are numerous methodologies for statistical downscaling based on different principles, such as analogues (Bannayan and Hoogenboom, 2008), interpolation (Plouffe et al., 2015) or machine learning models (Rozas-Larraondo et al., 2014; Salameh et al., 2009).

Aviation operations are highly affected by the weather and require the best quality meteorological information to maximise efficiency and safety. The International Civil Aviation Organization (ICAO) and the World Meteorological Organization (WMO) have established international standards to ensure high quality meteorological reports (WMO, 1995). To generate these reports, national weather services across the world employ highly qualified personnel who continuously observe and forecast conditions around the airport, such as visibility, direction and speed of the wind or proximity of storm cells. Aviation weather forecasters rely mainly on their knowledge of the airport and the quality of the NWP used.

* Corresponding author.
*E-mail address:* pablo.larraondo@anu.edu.au (P. Rozas Larraondo).

https://doi.org/10.1016/j.envsoft.2017.11.004
1364-8152/

There are a number of tools that facilitate the process of generating airport weather forecasts (Ghirardelli and Glahn, 2010; Jacobs and Maat, 2005), being an area of active research at the moment. Airports usually have long and regular series of high quality historical observation data that can be used to create statistical downscaling models to help forecasters in their work. The effect of non-resolved surrounding mountains, water bodies or local climate conditions can be incorporated by these models, by studying the local effects produced by weather patterns in the past.

Circular variables are present in any directional measurement or variable with an inherent periodicity. Weather data contain many parameters that are represented as circular variables, such as wind direction, geographical coordinates or timestamps. Most of the current regression machine learning algorithms focus on modelling the relationships between linear variables. Circular variables have a different nature to linear variables, so traditional methodologies are not able to represent their content thoroughly, leading to sub-optimal results in most cases. The model presented in this article builds upon the concept of circular regression trees introduced by Lund (2002). Our model is computationally more efficient and generates contiguous splits for circular variables, which results in improved accuracy when compared to its precursor.

Circular regression trees can better represent circular variables, as they consider more possibilities for splitting the space than linear regression trees do. Circular regression trees can define subsets of data around the origin $0, 2\pi$ radians point. For example, when predicting an event that shows a high correlation with the winter months in the northern hemisphere, a circular tree would be able to isolate the months from December to March in one group. On the other hand, a linear tree would most likely consider splits starting or ending at the beginning of the year, failing to create a group containing these months.

This paper introduces AeroCirTree, a system based on the described circular regression tree model, which is able to generate improved airport weather forecasts for any airport in the world. This software presents a general solution where all the necessary tools required to extract historical weather data, train models and generate new forecasts are made available. This system is intended to help aviation weather forecasters to produce better quality reports and for machine learning researchers to build upon more sophisticated models.

The paper is structured as follows: Section 2 contains the methodology used to create the model. Section 3 contains an introduction to the observed and numerical weather datasets used to develop and test the system. Section 4 presents results where the proposed model is compared with other regression tree methodologies. This section also contains a discussion of the results, providing the reader with deeper insight into the novelty of the proposed model. Section 5 provides a high level description of the model implementation, including its key components and their functionality as well as examples on how to use the software. Section 6 concludes this paper, revisiting the research highlights and proposing some ideas on future developments to carry this work forward.

## 2. Methodology

Because of their simplicity, training speed and performance, regression trees are a popular and effective technique for modelling linear variables. Classification and Regression Trees (CART) (Breiman et al., 1984) is one of the most popular versions of regression trees.

Linear regression trees recursively partition the space, finding the best split at each non-terminal node. Each split divides the space in two sets using a cost function, which is usually based on a metric for minimising the combined variance of the resulting children nodes.

Fig. 1 contains an example of a regression tree based on two linear variables $x_1$ and $x_2$. On the right side, there is a graphical representation of how the space is divided by creating splits on these two variables.

Circular variables are numerical variables whose values are constrained into a cyclical space - for example, a variable measuring angles in radians, spans between 0 and $2\pi$, where both values represent the same point in space. Although these variables can be included in a linear regression tree, they have to be treated as linear variables, which is an oversimplification and normally leads to suboptimal results (Lund, 2002).

A circular variable defines a circular space. A circular space is cyclic in the sense that it is not bounded; for instance, the notion of a minimum and maximum value does not apply. The distance between two values in the space becomes an ambiguous concept, as it can be measured in clockwise and anticlockwise directions, yielding different results. Also, this space cannot be split in two halves by selecting a value, as the '$<$' and '$>$' operators are not applicable.

In order to split a circular variable, at least two different values need to be defined. These two values describe two complementary sectors, each containing a portion of the data. Circular regression trees use this splitting approach for incorporating circular variables into regression trees.

There are many examples of circular variables. Any variable representing directional data or a periodic event is circular. More specifically, in the field of airport weather forecasting, wind direction, the time of the day or the date are examples of circular variables.

Lund (2002) proposes a methodology that allows circular variables to be incorporated into regression trees. Fig. 2 contains a similar representation to the previous example, but considering one circular variable $\alpha$ and a linear one $x_1$. On the right side, there is a chart representing how the space is partitioned using polar coordinates.

The methodology presented in this work builds upon the concept of circular regression trees, presenting an alternative that improves computational performance and the accuracy of its results. Fig. 3 shows how the space is partitioned using the proposed methodology.

Visually comparing Figs. 2 and 3, it is evident that regions are split differently. The novelty of this methodology, when compared to the original version proposed by Lund, is that it always generates contiguous splits. In doing so, we avoid an excessive fragmentation of the space, and the splits provide a better generalisation for its child nodes. The original methodology uses the '$\in$' and '$\notin$' operators to generate all the splits for circular variables. This usually generates partitions in which the subsets defined by the $\in$ clause are surrounded by the complementary $\notin$ subset. Our methodology uses these operators to create just the first split of a circular variable and, after that, uses the '$<$' and '$>$' operators to create the subsequent splits. This change also results in a reduction of the search space for possible splits. The proposed algorithm for generating circular trees has, as a consequence, $\mathcal{O}(n)$ cost instead of $\mathcal{O}(n^2)$, when compared to Lund's original proposal. The only exception is when computing the first split of a circular variable, which has a computational cost of $\mathcal{O}(n^2)$, as it has to consider all the different splits around the circle.

## 3. Software and datasets

AeroCirTree is a collection of Python scripts which provides the tools to train and test the three previously described regression tree

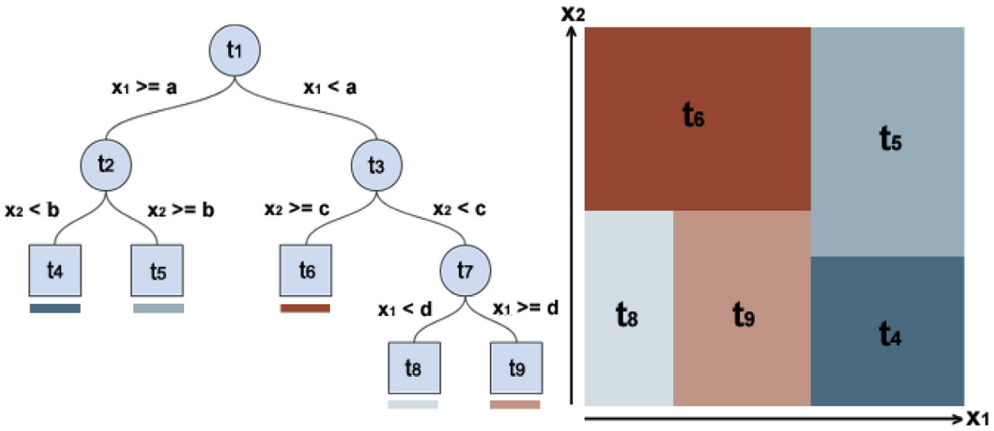


**Fig. 1.** Example of a classic linear regression tree and a representation of how the space is divided.

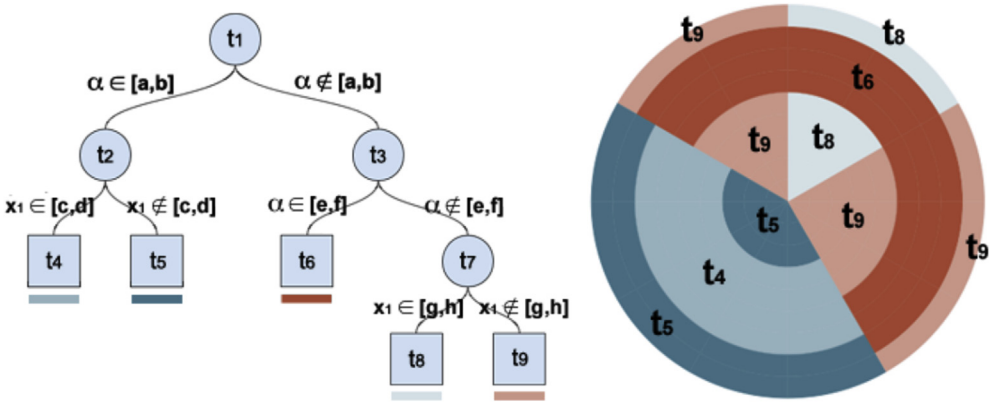


**Fig. 2.** Example of Lund's original proposal of circular regression tree and a representation of how the space is divided.

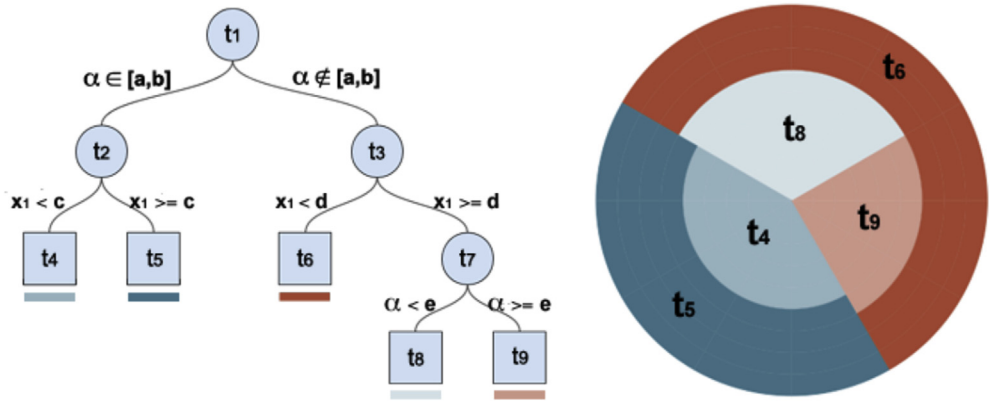


**Fig. 3.** Example of the proposed circular regression tree and a representation of how the space is divided.

methodologies using airport weather data. It uses NWP variables as the input and generates a more accurate value for the selected output variable by learning from the observed values for a certain location. Once the model has been trained, it can be used to improve the accuracy of the forecasted output value provided by new incoming NWP data.

It is worth noting that regression tree models are presented in this work as a method to statistically downscale the output of NWP for specific locations. They are not used to predict future values of a time-series but to improve the values produced by NWP. Analysis data from the NWP model and observed data are used to train the regression trees. These trees can account for biases and systematic errors of the NWP model. Trained models can be applied to any forecasting horizon produced by the NWP to correct systematic and random errors.

The AeroCirTree software presented in this work offers a general implementation of a regression tree. AeroCirTree allows its users to train linear regression trees as well as circular versions using non-contiguous or contiguous splits, as we propose. To determine which methodology is used, each variable in the input or output can be

tagged as being either [linear, circular] using a configuration file. An extra tag, contiguous, which can be set to [true, false], indicates the split methodology applied to circular variables. Different values of these tags indicate different versions of regression trees. For example, classic linear regression trees can be generated by tagging all their input variables as linear and contiguous = true. Lund's proposal of circular tree would require the circular input variables to be tagged as circular and contiguous = false. Lastly, our proposed methodology would require the same circular input variables to be contiguous = true.

AeroCirTree makes use of two weather datasets. The first is the output of a global NWP, called the Global Forecast System model (GFS) (Campana and Caplan, 2005), which is run operationally by the National Oceanic and Atmospheric Administration (NOAA). The second uses Meteorological Aerodrome Reports (METARs) (WMO, 1995), which contain periodic meteorological observations from airports around the world.

Each of these datasets contains several variables describing different weather parameters, such as the temperature, humidity, wind speed or cloud cover at the different locations they represent. The GFS model represents data using a regular grid which covers the whole world with a spatial resolution of approximately 50 km and a temporal resolution of 3 h. NOAA maintains an Operational Model Archive and Distribution System (NOMADS) to publish the GFS data. This archive contains the GFS outputs for the last 10 years.

METARS are weather text reports that encode observed meteorological parameters at airport runways using a well defined code. METARS are produced with an hourly or half-hourly frequency and are also made publicly available through the WMO Global Telecommunication System (GTS). The National Centers for Environmental Prediction (NCEP) maintains a system called Meteorological Assimilation Data Ingest System (MADIS), which archives all the METAR reports that have been produced in the world for the last 10 years. Each report is uniquely identified by its header, which contains the International Civil Aviation Organization (ICAO) airport code and a UTC time stamp.

The provided AeroCirTree software contains a command line utility that extracts the information from these two datasets for any given airport and date range. The output is presented as a convenient csv file containing the values of the different variables as a time series. All operations, such as locating the airport coordinate in the GFS grid, parsing and extracting METARs or homogenising variable units, are handled by the software, so the user can easily get a clean dataset for the desired airport. This csv file is the input used to train new models.

## 4. Experiments and results

The hypothesis of this study is that our proposed methodology for generating regression trees provides better generalisation and accuracy than previous non-contiguous circular regression trees when using circular variables and the equivalent classic linear methodologies.

The next sections go through the required steps to extract the necessary data, train the models and generate the forecasts. The last section contains an analysis of the proposed model accuracy and a comparison with the results provided by the GFS raw output, Lund's methodology and classic linear regression trees.

### 4.1. Data extraction and model training

To compare the differences in performance between methodologies, we use weather data coming from simulated NWP and observed data from different airports. Regression trees are trained using NWP as input and the observed speed of the wind as the output variable. It is worth noting that regression tree models are not used to forecast wind speeds into the future. These models are used to statistically downscale NWP data, correcting biases and systematic errors.

We choose to forecast the observed speed of the wind at 5 different locations in Europe. Data from the airports of Berlin Tegel (EDDT), London Heathrow (EGLL), Barcelona El Prat (LEBL), Paris Charles de Gaulle (LFPG) and Milano Malpensa (LIMC) are used to train the different models and to analyse the results. The models are trained using three-hourly data for the years 2011, 2012 and 2013, providing approximately 8760 samples per airport.

Each model generates the required partitions to predict the observed wind speed using the following GFS parameters as input variables: relative humidity, speed and direction of the 10-meter wind as well as the time of the day associated with the values. Wind speed is one of the most important weather variables affecting airport operations. This variable is also highly dependent on another variable, wind direction, which is circular. The reason for including these two variables in our experiments is that, in conjunction, they can represent local topography effects non resolved by weather models. Surface relative humidity is used as an indicator for phenomena such as rain or fog conditions. Lastly, time of the day, also a circular variable, is highly correlated with the daily patterns of the wind.

The stop criterium for all the considered trees is based on the number of elements in a node. Splits are recursively performed until the number of data entries in a node falls below a certain value. Then, the splitting process is stopped and the node is denoted as a leaf. This value receives the name “maximum leaf size”. Large values of “maximum leaf size” generate shallow trees, whereas small values generate deep trees with a larger number of nodes. For each airport, different versions of the model are generated using different maximum tree leaf sizes. The maximum leaf size values considered in this experiment are: 1000, 500, 250, 100 and 50. This is the content of the config file used to train our proposed model for the comparison defining a maximum leaf size of 100 (*please refer to* Section 5.2 *for more details on how these files are used and defined.*):

```
{"output":{"name":"metar_wind_spd","type":"linear"},
 "input":[{"name":"gfs_wind_spd","type":"linear"},
          {"name":"gfs_wind_dir","type":"circular"},
          {"name":"gfs_rh","type":"linear"},
          {"name":"time","type":"circular"}],
 "contiguous":true

 "max_leaf_size":100}
```

### 4.2. Experimental analysis

Following the process described in the previous sections, data from 2011 to 2013 is extracted for the 5 selected airports. For each airport and value of maximum leaf size, three different models are generated: classic linear regression tree (using the **u**, **v** components of the wind speed and time of the day), Lund's and our proposed circular regression tree.

To evaluate the differences in accuracy between these three methodologies, a 5-fold cross validation procedure is used. This validation process ensures that models are tested using data that has not been used at training time. In order to avoid differences in the results caused by different partitions in the validation process, the same 5-fold partition is used to validate all the methodologies for the different values of the "maximum leaf size" parameter. The error in forecasting is defined as the difference between the speed of the wind predicted by the tree, which is the mean of the target values contained in the corresponding leaf, and the observed METAR wind speed value. The Refined Index of Agreement (RIA) (Willmott et al., 2012) is used to measure the differences in accuracy between methodologies. This index provides greater separation when comparing models that perform relatively well and is less sensitive to errors concentrated in outliers when compared to other methods such as absolute or root mean squared error. The RIA can be expressed as

$$RIA = 1 - \frac{\sum_{i=1}^{n} |P_i - O_i|}{2\sum_{i=1}^{n} |O_i - \overline{O}|}$$

Where $O_i$ represents the observations and $P_i$ the predictions produced by the model.

Table 1 contains the resulting RIA values for each tree methodology as well as the reference value of the 10-meter wind speed produced by GFS in the airports previously referenced. Higher values of RIA indicate better accuracy in the results. Similar results using different combinations of input and output variables combining linear and circular variables are made available, as a text file, at the main code repository.

**Table 1**
Comparison of the RIA values when forecasting the observed METAR wind speed for the different airports using the direct output of GFS, a classic linear regression tree, Lund's circular tree and the proposed model.

| Airport | Method | RIA per Max Leaf Size | | | | |
|---|---|---|---|---|---|---|
| | | 1000 | 500 | 250 | 100 | 50 |
| EDDT | GFS (ref.) | 0.669 | 0.669 | 0.669 | 0.669 | 0.669 |
| | Linear | 0.684 | 0.695 | 0.710 | 0.716 | 0.713 |
| | Lund | 0.700 | 0.713 | 0.720 | 0.715 | 0.702 |
| | AeroCirTree | 0.700 | 0.712 | 0.717 | 0.721 | 0.714 |
| EGLL | GFS (ref.) | 0.653 | 0.653 | 0.653 | 0.653 | 0.653 |
| | Linear | 0.687 | 0.703 | 0.716 | 0.728 | 0.730 |
| | Lund | 0.702 | 0.721 | 0.731 | 0.735 | 0.729 |
| | AeroCirTree | 0.702 | 0.720 | 0.730 | 0.737 | 0.737 |
| LEBL | GFS (ref.) | 0.362 | 0.362 | 0.362 | 0.362 | 0.362 |
| | Linear | 0.591 | 0.601 | 0.607 | 0.613 | 0.607 |
| | Lund | 0.602 | 0.608 | 0.615 | 0.606 | 0.590 |
| | AeroCirTree | 0.601 | 0.607 | 0.619 | 0.619 | 0.606 |
| LFPG | GFS (ref.) | 0.604 | 0.604 | 0.604 | 0.604 | 0.604 |
| | Linear | 0.674 | 0.691 | 0.702 | 0.711 | 0.707 |
| | Lund | 0.704 | 0.716 | 0.719 | 0.706 | 0.691 |
| | AeroCirTree | 0.704 | 0.712 | 0.715 | 0.714 | 0.707 |
| LIMC | GFS (ref.) | 0.401 | 0.401 | 0.401 | 0.401 | 0.401 |
| | Linear | 0.517 | 0.519 | 0.519 | 0.509 | 0.496 |
| | Lund | 0.521 | 0.520 | 0.518 | 0.500 | 0.482 |
| | AeroCirTree | 0.522 | 0.521 | 0.521 | 0.513 | 0.501 |

Looking at the RIA values contained in Table 1, it can be noted that the use of regression tree models significantly improves the level of accuracy from the output of the GFS model. The level of improvement is highly dependent on the selected airport. This may be due to the fact that each grid point of the GFS model contains a representation of the weather in an area of approximately 50 square kilometres, and some locations and variables are better represented by this simplification than others. For example, airports surrounded by mountains will benefit more from statistical models than airports located on large plains.

Comparing the differences in accuracy between the three regression tree models shown in Table 1, the use of the proposed model provides better results in most of the cases. The level of improvement also varies significantly between different airport locations. Results are analysed considering the case of shallow and deep trees. For shallow trees, the two circular models show very similar behaviour outperforming the linear approach. As the maximum leaf size parameter gets smaller, we see an improvement in accuracy for all three models. Deeper trees still show better results for the circular models, but Lund's proposal starts showing signs of premature over-fitting when compared to the other two models. In the case of the deepest tree (maximum leaf size equal to 50), all three models show a deterioration of performance, with Lund's being the most noticeable case.

In the case of Paris Charles de Gaulle (LFPG), shallow circular trees show an improvement of around 4–5% when compared to the classic linear tree version. This improvement is maintained by our proposed model when considering deeper trees. However, Lund's model does not improve at the same rate. A more systematic analysis of the results of this test is offered at the end of the section, providing the statistical significance of the differences between methodologies.

Figs. 4 and 5 show a graphical representation of the evolution of the RIA when predicting wind speed for the airports of London Heathrow (EGLL) and Barcelona El Prat (LEBL) respectively. All the regression tree methodologies improve their accuracy as the maximum leaf size decreases, showing signs of overfitting for the smallest leaf size case. The value of the GFS wind speed value at the closest grid point is shown as a reference to represent the relative improvement achieved by each model.

As introduced in Section 2, the circular methodologies have the benefit of considering extra partitions for circular variables, those that cross the origin, when compared to linear methods. The benefits of using circular trees are more noticeable for the case of shallow trees, the ones with larger values of maximum leaf size. The first split of a circular variable normally happens at one of the first nodes of the tree, near the root node. Splits that happen at the top part of a tree have a major impact on its performance, because they divide a bigger proportion of the dataset. For shallow trees, finding a good partition at these levels is critical, whereas deeper trees can improve poor partitions by creating new ones.

Non-contiguous circular regression trees generate partitions that seem to provide a poorer generalisation for subsequent splits than the other two methodologies. The good results shown by Lund's method for shallow trees quickly deteriorate for deeper trees. The proposed methodology, based on contiguous circular trees, achieves a similar performance to Lund's method for shallow trees and also better results than the other two methodologies for deeper ones. Moreover, as mentioned in Section 2, the proposed methodology is more efficient computationally than the non-contiguous version.

In order to evaluate the results, the methodology proposed by Demsar (2006) is used to assess the statistical significance of the differences between methods. The null hypothesis of similarity is rejected for linear and both circular regression trees. This justifies

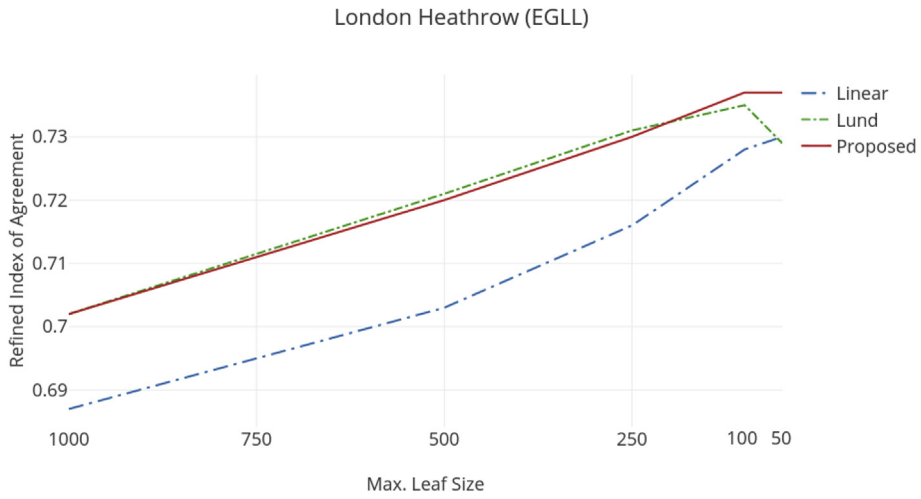


**Fig. 4.** RIA values for the airport of London Heathrow (EGLL), comparing the accuracy of the output for different maximum leaf sizes.

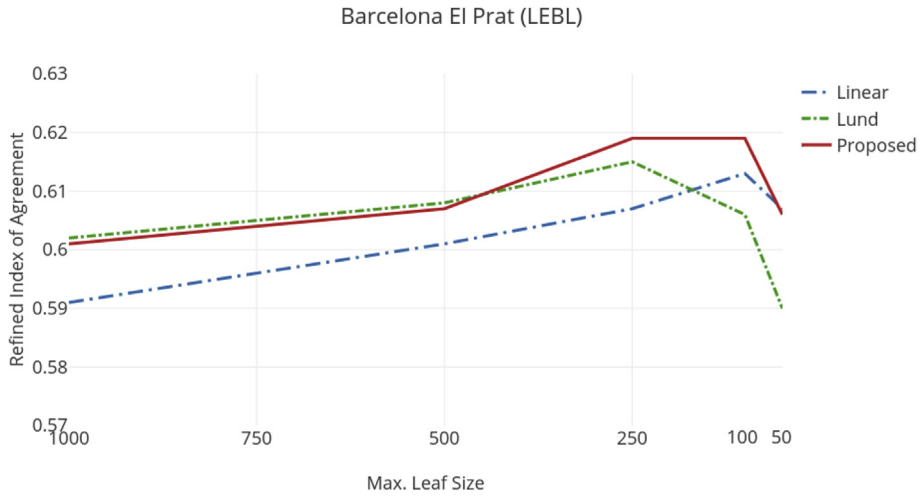


**Fig. 5.** RIA values for the airport of Barcelona El Prat (LEBL) comparing the accuracy of the output for different maximum leaf sizes.

the use of post-hoc bivariate tests, Nemenyi in our case, which assess the statistical difference between pairs of algorithms. The results of these tests can be graphically expressed using Critical Difference (CD) diagrams. The Nemenyi test pairwisely compares every methodology. The accuracy of any two methodologies is considered significantly different if the corresponding average rank differs by at least the critical difference.

Fig. 6 represents the RIA results of the Nemenyi test ($\alpha = 0.05$) making use of CD diagrams for the maximum leaf sizes of 1000, 100 and 50, as they represent both extremes of the proposed range.

CD diagrams connect the groups of algorithms for which no significant differences were found, or in other words, those whose distance is less than the fixed critical difference, shown above the graph. Note that algorithms ranked with lower values in CD diagrams imply higher RIA scores. These tests have been performed using the scmamp R package, which is publicly available at the Comprehensive R Archive Network (CRAN) (Calvo and Santafe, 2016).

As can be seen in the CD diagrams in Fig. 6, for shallow trees, both circular methodologies outperform the linear approach (maximum leaf size 1000). As the experiment progresses into deeper trees (maximum leaf size 100), the proposed methodology statistically outperforms the other two in the considered datasets. Even for the case of maximum leaf size 50, when all the methods show a deterioration in accuracy, the proposed methodology shows the best results. Lund's methodology, on the other hand, reveals a major degradation in accuracy for the smallest maximum leaf size. These results corroborate our experimental hypothesis: the proposed circular regression tree is able to generate models that provide better generalizations for circular variables.

## 5. Design and use of the software

AeroCirTree is a Python 3 package implementing regression trees and a set of command line tools to extract weather data and train tree models for any airport in the world. Users will normally use the provided package by using three scripts, named aerocirtree_extract, aerocirtree_train and aerocirtree_test, which fetch historical time-series weather data, train models and test results respectively, for any airport in the world.

### 5.1. Implementation design

The proposed circular regression tree has been implemented as

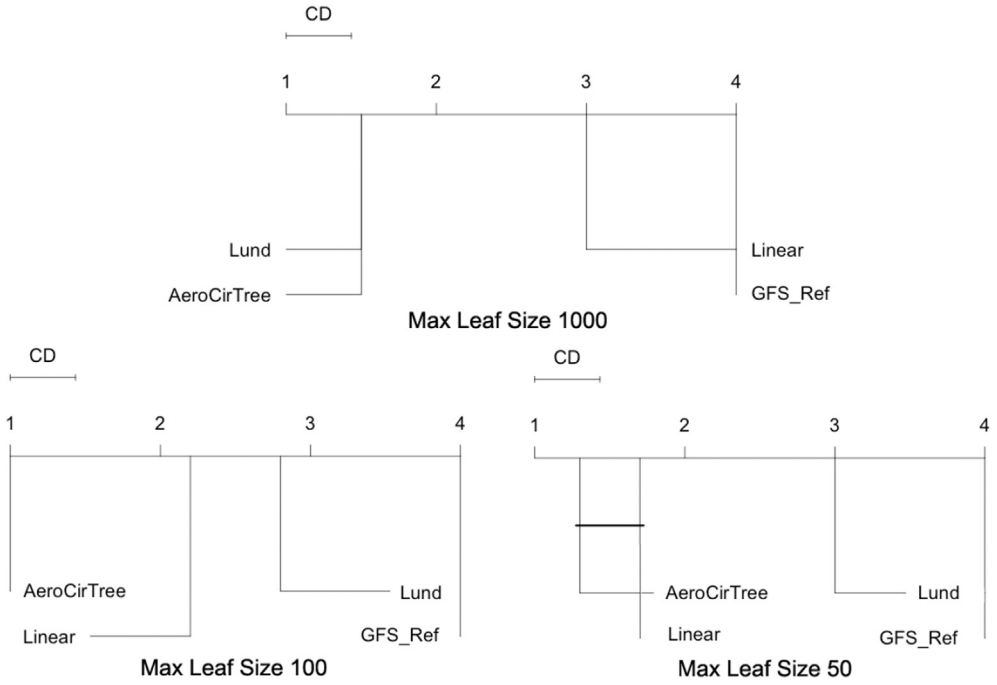


**Fig. 6.** Critical Differences comparing the three methodologies for shallow and deep trees. $\alpha = 0.05$.

a Python package. Most of its functionality is contained in two classes, called Data and Node. A tree is modelled as a nested structure of Node instances. Each Node in the tree contains an instance of the Data class, which represents the subset of the dataset contained in that node. The Data object is built around the Python Pandas DataFrame class.

Node contains two class attributes of type Node, named left_child and right_child, defining a recursive structure. Each non-terminal node in a tree contains two Node instances which constitute its left and right children. On the other hand, terminal nodes or leaves are characterised by having the contents of its children set to the None value.

Node defines also the .Split() method which creates a split generating two new instances of the Node class. Each of these two new Node instances contains one part of the original Data and is assigned to the left_child and right_child attributes. A tree is built by recursively calling the .Split() method on each of the children Node until the stop criteria is satisfied. The stop criteria can be configured to be a minimum number of elements or variance value for the Data contents of a node.

Each column of a node's Data has to be tagged as linear or circular to designate the nature of the data it represents. By tagging columns, we can dynamically train different tree versions and compare their results. Classic regression trees consider all the variables as linear, whereas our proposed methodology allows some of the variables to be treated as circular. For example, by tagging all variables as linear, we will get a classic regression tree.

This implementation is generic and can be applied to data from any field if made available in csv format.

### 5.2. User guide

AeroCirTree also provides a series of scripts to extract weather data, train and test regression tree models. These scripts make use of the previously described package to train specific models for any airport in the world.

Here is an example that shows how to extract the data for the airport of London Heathrow from the 1st of January 2016 to the 1st of June 2016:

```
$ ./aerocirtree_extract --airport EGLL --start_date 20160101\
--end_date 20160601
metar_press,metar_rh,metar_temp,metar_wind_spd,gfs_press,\
gfs_rh,gfs_temp,gfs_wind_dir,gfs_wind_spd,time,date
1025.0,75.5,6.0,2.57,1016,92,3,280,3,45.0,0
1024.0,80.92,5.0,4.12,1016,96,3,290,3,90.0,0
1024.0,80.92,5.0,2.57,1015,97,4,300,3,135.0,0
1024.0,86.99,6.0,2.57,1016,93,6,340,3,180.0,0
...
```

Note that the values of time and date are transformed to their numerical values as circular variables, where the origin [0–360] corresponds to 00:00 h and the 1st of January respectively. The

output of this command can be redirected to a local file. These files are used as the input required to train tree models.

Once a dataset is available for a given airport, a model can be trained by defining its input and target variables. The output variable has to be one of the observed variables coming from the METAR reports and the input variables are the GFS forecasted variables or a subset of them.

Doing it this way, when new forecast data from the GFS is available, the model can be used to generate an enhanced forecast of the target variable. The different options to create a model are specified through a configuration file. This configuration file contains a JSON object with three fields: “output”, “input” and “max_leaf_size”. The name of the target variable produced by the tree is specified in “output”. Input variables are listed in the “input” field along with a tag to treat them as either circular or linear. The max_leaf_size parameter specifies the value to control the depth of the resulting tree. For example, to specify a model to forecast temperature using GFS relative humidity, wind direction as a circular variable and a maximum leaf size of 100, a file with the following content should be specified:

```
{"output":{"name":"metar_temp","type":"linear"},
 "input":[{"name":"gfs_wind_dir","type":"circular"},
          {"name":"gfs_rh","type":"linear"}],
 "contiguous":true
 "max_leaf_size":100}
```

To train a model we use aerocirtree_train, which receives as arguments the paths of a file containing the data and a configuration file. Supposing the output of the data extracted in the previous section has been saved in a file named EGLL.csv and the presented configuration file is saved as Model_A.json, a model can be trained by running:

```
$ ./aerocirtree_train --data EGLL.csv --config Model_A.json
```

This command learns the specified model and saves it using a name that combines both input file names and using the extension .mod. The previous model would be saved on disk with the file name EGLL_Model_A.mod.

Finally, aerocirtree_test can be used to run the model on new data. This script receives the path to a saved model file and input csv as arguments. The script returns the resulting model outputs for each line of the input file.

For example, supposing we want to test our previously trained model EGLL_Model_A.mod with new data contained in the file EGLL.csv, we could run:

```
$ ./aerocirtree_test --data EGLL_new.csv --model EGLL\_Model\_A.mod
```

This command computes the resulting temperature values for each of the input values at the airport of London Heathrow.

## 6. Conclusions

This work presents a software application for forecasting the weather in any airport of the world. It also proposes a new circular regression tree methodology which offers better accuracy when compared to classic linear methods, and also better accuracy and computational efficiency than Lund's original proposal of circular regression trees.

This software contains a library that implements a general version of regression trees as well as the command line tools to train, test and download new airport datasets. These tools have been designed so users can create their own forecasts and also so that they can experiment and explore the differences between models, input variables and airports. Scripts and libraries are written in a simple way so users can read the code to understand what the program is doing and also modify parts of it. AeroCirTree comes with a GNU GPLv3 licence so anyone can use, modify and share this program for any purpose.

The model proposed in this work is based on a new methodology to build a basic circular regression tree. Regression trees have evolved with the introduction of many different techniques that improve both their accuracy and efficiency. Well known techniques that modify standard regression trees such as pruning, balancing, smoothing (Breiman et al., 1984; Quinlan, 1993) or random forests (Breiman, 2001) and ensembles (Bühlmann, 2012) can be also applied to circular regression trees and can improve the accuracy of results when compared to basic regression trees. Future work could implement the ideas presented in the referred publications offering more advanced models.

## Acknowledgements

We would like to thank the National Computational Infrastructure (NCI) at the Australian National University and the University of the Basque Country for their support and advice in carrying out this research work.

We are grateful for the support of the Basque Government (IT609-13), the Spanish Ministry of Economy and Competitiveness (TIN2016-78365-R) and a University-Society Project (15/19 Basque Government and UPV/EHU).

Jose A. Lozano is also supported by BERC program 2014–2017 (Basque Gov.) and Severo Ochoa Program SEV-2013-0323 (Spanish Ministry of Economy and Competitiveness).

## References

Bannayan, M., Hoogenboom, G., 2008. Weather analogue: a tool for real-time prediction of daily weather data realizations based on a modified k-nearest neighbor approach. Environ. Model. Softw. 23, 703–713.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Breiman, L., Friedman, J., Olshen, R., Stone, C., 1984. Classification and Regression Trees. Wadsworth Books.

Bühlmann, P., 2012. Handbook of Computational Statistics. Springer Berlin Heidelberg.

Calvo, B., Santafe, G., 2016. scmamp: statistical comparison of multiple algorithms in multiple problems. R J. 8, 248–256.

Campana, K., Caplan, P., 2005. Technical Procedure Bulletin for t382 Global Forecast System.

Carvalho, A.C., Carvalho, A., Martins, H., Marques, C., Rocha, A., Borrego, C., Viegas, D.X., Miranda, A.I., 2011. Fire weather risk assessment under climate change using a dynamical downscaling approach. Environ. Model. Softw. 26, 1123–1133.

Demsar, J., 2006. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30.

Ghirardelli, J.E., Glahn, B., 2010. The meteorological development laboratorys aviation weather prediction system. Weather Forecast. 25, 1027–1051.

Jacobs, A.J.M., Maat, N., 2005. Numerical guidance methods for decision support in aviation meteorological forecasting. Weather Forecast. 20, 82–100.

Lund, U.J., 2002. Tree-based regression for a circular response. Commun. Statistics Theory Meth. 31, 1549–1560.

Plouffe, C.C.F., Robertson, C., Chandrapala, L., 2015. Comparing interpolation techniques for monthly rainfall mapping using multiple evaluation criteria and auxiliary data sources: a case study of Sri Lanka. Environ. Model. Softw. 67, 57–71.

Quinlan, J.R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Inc.

Rozas-Larraondo, P., Inza, I., Lozano, J.A., 2014. A method for wind speed forecasting in airports based on nonparametric regression. Weather Forecast. 29, 1332–1342.

Salameh, T., Drobinski, P., Vrac, M., Naveau, P., 2009. Statistical downscaling of near-surface wind over complex terrain in Southern France. Meteorol. Atmos. Phys. 103, 253–265.

Willmott, C.J., Robeson, S.M., Matsuura, K., 2012. A refined index of model performance. Int. J. Climatol. 32, 2088–2094.

WMO, 1995. Manual on Codes International Codes VOLUME I.1. Geneva.

## 2.3 Convolutional neural networks for image regression

### 2.3.1 Introduction

During the time of working on the circular regression tree, deep learning techniques where presented to the research community demonstrating unprecedented results in many different domains. Although, as far as we knew, there were not applications in the meteorological domain in the literature we decided to learn and apply these techniques to weather forecasting problems. Convolutional neural networks, used mainly on image classification tasks, seemed a good candidate to model NWP gridded data.

Up until this point in our research, we extracted data from the closest NWP grid point to the desired observed dataset resulting in a time series. Convolutional Neural Networks (CNN) offer the possibility of treating a whole image as the input to a model and the neural network can learn to detect the regions of the image that are most correlated with the output. This is a major change on how we were doing research compared to our two previous works. CNNs do not require to know the location of the individual grid points as an input, they can treat the whole weather grids (images) as their only input avoiding the need of extracting temporal series beforehand.

CNNs are mainly used to perform classification and extraction of spatial information in images, building from fine grained details into higher level structures. In this work we explored the use of CNNs to classify the event of precipitation (rain/dry) at different cities in Europe.

CNN require large data sets to be trained. For this work we used a different dataset called ERA-Interim (Dee et al., 2011) which contains data since the year 1979, with a temporal resolution of 3 hours. This dataset is publicly available from the European Centre for Medium-Range Weather Forecasts (ECMWF). This dataset is generated using a numerical weather model which simulates the state of the atmosphere for the whole planet, with a spatial resolution of approximately 80 km. The output is presented in the form of regular numerical grids and there is a large number of physical parameters available, representing variables such as temperature, wind speed and relative vorticity.

For determining the event of rain at the different locations we use Aviation Routine Weather Reports (METARs) similarly to our previous works. We consider 5 main airports located in different cities across Europe and a period of 5 years (2012-2017). The airports are: Helsinki-Vanta, Amsterdam-Schiphol, Dublin, Rome-Fiumicino and Vienna.

We extract an extended area over Europe from ERA-Interim, creating a 3 hourly series of images composed by 3 bands, corresponding to the geopotential height $\mathbf{z}$ at the 1000, 700 and 500 pressure levels of the atmosphere. This parameter represents the height in the atmosphere at which a certain pressure value is reached. These levels correspond typically to 100, 3000 and 5500 metres above the mean sea level respectively.

### 2.3.2 Research contribution

In this work we experimented adding the temporal dimension to the CNN network. The temporal dimension is added to these networks by adding a fourth axis to the

convolutional kernels (latitude, longitude, height, time). Using the collection of geopotential fields as input and the precipitation conditions for the different locations as output, the CNN networks are trained to predict the rain conditions at each point. Although we are not giving the coordinates of the different locations within the pressure field, the network succeeds at finding the relationship between both. This work demonstrates therefore that CNNs can be used to interpret the output of NWP to generate local forecasts automatically.

Also as part of this work we used a technique called Class Activation Mapping (CAM) (Zhou et al., 2016) to introspect inside the CNN models and visually assess the regions of the input space that have more influence in the output. The use of CAM is represented in Figure 2.3 using a heat-map to represent the relative weight of each pixel in the network. It can be seen how the network learns to give more weight to the pixels surrounding the region even when this parameter is not known to the network.

Also in this work, 3D convolutions are proven to be able to naturally incorporate the temporal component of the data into neural networks, resulting in a significant improvement in the accuracy of the results when compared to a similar network trained with individual frames.

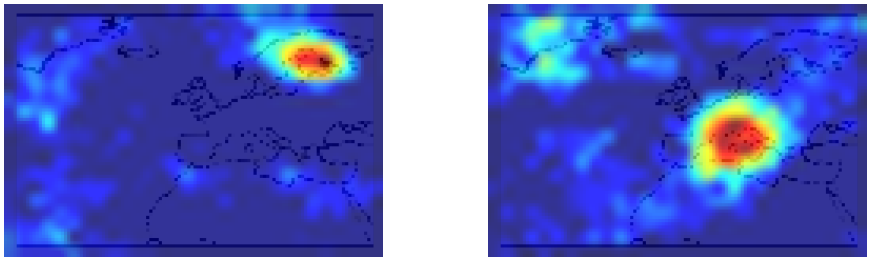

FIGURE 2.3: Example of the resulting Class Activation Maps for an ERA-Interim CNN trained using the observed precipitation at Helsinki-Vantaa airport, EFHK, (left) and Rome Fiumicino airport, LIRF, (right). Coastlines have been overlaid as a reference for readers.

### 2.3.3 Publication: Automating weather forecasts based on convolutional networks

This work was accepted for the "Deep Structured Prediction" workshop celebrated as part of the 2017 ICML conference in Sydney. The work was presented in July 2017 at the workshop and I received positive feedback on the techniques and a few people asked about the nature and availability of the dataset.

# Automating weather forecasts based on convolutional networks

**Pablo Rozas Larraondo** [1] **Iñaki Inza** [2] **Jose A. Lozano** [2 3]

## Abstract

Numerical weather models generate a vast amount of information which requires human interpretation to generate local weather forecasts. Convolutional Neural Networks (CNN) can extract features from images showing unprecedented results in many different domains. In this work, we propose the use of CNN models to interpret numerical weather model data which, by capturing the spatial and temporal relationships between the input variables, can produce local forecasts. Different architectures are compared and a methodology to introspect the models is presented.

## 1. Introduction

Weather forecasting is based on Numerical Weather Predictions (NWP) that capture the state of the atmosphere and simulate its evolution based on physical and chemical models. Global NWP models normally provide a large number of parameters representing different physical variables in space and time. Because of the lack of spatial and temporal resolution, these fields need to be interpreted by highly qualified personnel to produce forecasts for any specific region. This is still today a human based process, which relies on specifically trained and experienced professionals to interpret modeled and observed data (Wilson et al., 2017; Gravelle et al., 2016). NWP variables define the state of the atmosphere and its changes through space and time. NWPs define a highly structured dataset in which the relationships between its variables are defined by physics equations, such as conservation of mass, momentum and energy.

Recent advances in neural networks have proven that by increasing the number of general hidden layers, unprecedented results can be achieved in many different domains. More specifically, research around Convolutional Neural Networks (CNN) (Krizhevsky et al., 2012) has proven to be very effective in solving image classification and segmentation problems.

Machine learning has been applied to different areas of weather forecasting, such as downscaling (Tripathi et al., 2006) or nowcasting (Xingjian et al., 2015). The main deficiency of the traditional methodologies is their inability to incorporate both the spatial and temporal components present in the data. Most of the existing research in this field has been based on manually extracting the points in a model representing a certain location, and training models with the resulting data. The problem with this approach is that weather is a dynamic system, and analysing individual points in isolation misses important information contained in the synoptic and meso-scales.

CNNs enable analysis and extraction of the spatial information in images, building from fine grained details into higher level structures. The temporal dimension can be added to these networks by adding a third axis to the convolutional kernels. This work demonstrates how CNNs can be used to interpret the output of Numerical Weather Prediction (NWP) automatically to generate local forecasts. The main outcomes of this work are:

- CNNs are able to provide a model to interpret numerical weather model fields directly and to generate local weather forecasts.
- Class Activation Mapping (CAM) provides a valuable mechanism to assess the spatial and temporal correlations of the different fields visually, helping to introspect and develop new models.
- 3D convolutions can naturally incorporate the temporal component into neural networks, significantly improving the accuracy of the results.

## 2. Datasets

For this work, we propose the use of NWP and observed precipitation data from different locations, to experiment with different configurations of CNN models. The objective is to train a model which predicts the event of rain for a

[1]National Computational Infrastructure, Australian National University, Australia [2]University of the Basque Country, Spain [3]Basque Center for Applied Mathematics, Spain. Correspondence to: Pablo Rozas Larraondo <pablo.larraondo@anu.edu.au>.

*Proceedings of the ICML 17 Workshop on Deep Structured Prediction*, Sydney, Australia, PMLR 70, 2017. 

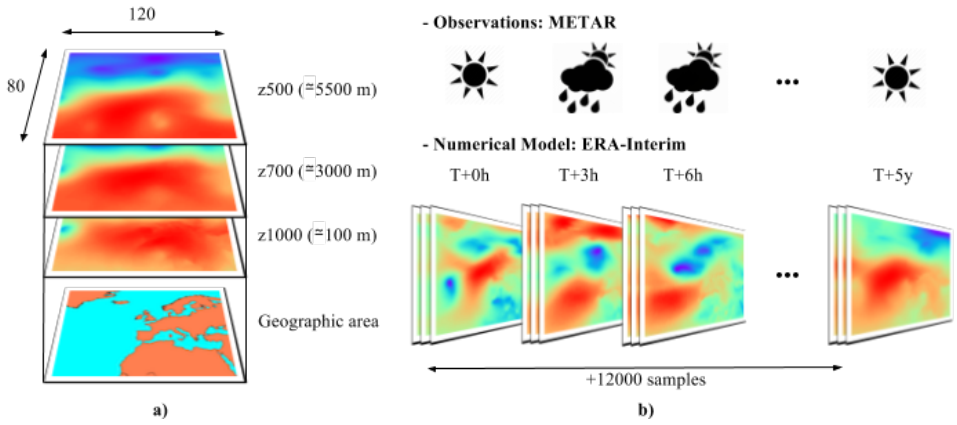


*Figure 1.* a) represents the 3 geopotential subsets extracted from ERA-Interim, corresponding to different heights of the atmosphere, stacked over a map to represent the spatial extent. b) Represents the whole extracted time series and the alignment of both datasets.

particular location, using numerical weather model data as input. In this section, we describe how these datasets have been generated.

ERA-Interim (Dee et al., 2011) is a publicly available meteorological reanalysis dataset from the European Centre for Medium-Range Weather Forecasts (ECMWF). This dataset is generated using a numerical weather model which simulates the state of the atmosphere for the whole planet, with a spatial resolution of approximately 80 km. There is data available since the year 1979, with a temporal resolution of 3 hours. The output is presented in the form of regular numerical grids and there is a large number of physical parameters available, representing variables such as temperature, wind speed and relative vorticity.

Aviation Routine Weather Reports (METARs) are operational aviation weather text reports that encode observed meteorological variables for every commercial airport in the world. METARs are produced with an hourly or half-hourly frequency and are also made publicly available through the World Meteorological Organisation (WMO) communications system. Each report is uniquely identified by its header, which contains the International Civil Aviation Organization (ICAO) airport code and a UTC time stamp.

We considered 5 main airports located in different cities across Europe and a period of 5 years (2012-2017) to perform our experiments. The airports and their corresponding ICAO codes are: Helsinki-Vanta (EFHK), Amsterdam-Schiphol (EHAM), Dublin (EIDW), Rome-Fiumicino (LIRF) and Vienna (LOWW).

We extracted an extended area over Europe from ERA-Interim, creating a 3 hourly series of images composed by 3 bands, corresponding to the geopotential height **z** at the 1000, 700 and 500 pressure levels of the atmosphere. This parameter represents the height in the atmosphere at which a certain pressure value is reached and the levels correspond typically to 100, 3000 and 5500 metres above the mean sea level respectively.

The reason for selecting these fields is that weather forecasters normally base their predictions on these. They contain information about the shape, location and evolution of the pressure systems in the atmosphere.

Using the METAR data, the precipitation conditions [*rain*, *dry*] were extracted for each airport for the same time period and frequency. The resulting dataset time series contains over 12000 samples. Figure 1 represents a sample of the considered ERA-Interim fields with their size and geographical extent on the left. The right side, shows how the ERA-Interim data aligns with the observed precipitation for a sample location.

## 3. Experiments and Results

The objective of the proposed experiment is to predict precipitation events for the considered airports using ERA-Interim geopotential data as the input and METAR observations to annotate the samples [*rain*, *dry*]. Two different CNNs are used. The first model performs 2D convolutions and the second incorporates the temporal dimension based on 3D convolutions (Ji et al., 2013). We aim to prove that these models can capture part of the mental and intuitive process that human forecasters follow when interpreting numerical weather data.

### 3.1. CNN architecture

To perform the experiments, a 2 layer CNN is used. Each convolution layer uses a 3x3 kernel followed by a 2x2 max-pooling layer. After the convolution operations, a fully connected layer is used to connect the output [*rain*, *dry*] using a 'softmax' activation function.

*Table 1.* Rain forecasting accuracies for the different locations comparing 2D and 3D CNNs with the reference accuracy of climatology.

| AIRPORT | RAIN CLIM. | 2D CNN | 3D CNN |
|---|---|---|---|
| EFHK | 60.8 | 73.6 | 75.4 |
| EHAM | 74.2 | 77.8 | 79.3 |
| EIDW | 61.2 | 70.7 | 72.6 |
| LIRF | 83.1 | 87.3 | 88.2 |
| LOWW | 75.7 | 77.1 | 78.8 |

For the 3D CNN, the configuration is similar to the previous version, but the kernels in both the convolution and max-pooling layers have an extra dimension, with sizes 3x3x3 and 2x2x2 respectively. The 3D CNN, is trained by aggregating the input dataset in groups of 8 consecutive images. This aggregation represents the evolution of the atmosphere over 24 hours. The neural network can then extract information out of the temporal dimension, using the observation corresponding to the last image of the series as output.

The 2D and 3D CNNs were implemented in TensorFlow (Abadi et al., 2016) and trained per airport over 300 epochs using 80% of the data. The remaining 20% was used as validation to test the accuracy of the models.

### 3.2. Results

Table 1 contains the results produced by the different models using the validation dataset. The accuracy values represent the success rate of the model when predicting either *rain* or *dry* conditions for each location. The climatology for each location, number of rain observations over the total number of observations, is also included in the results as a reference. A model whose output is always '*dry*' would have that success rate.

Figure 2 represents the results using a stacked bar chart. The relative improvement over climatology achieved with the 2D and 3D convolutional models is represented by the green and red fractions of the bar.

### 3.3. Class Activation Mapping

Class Activation Mapping (CAM) (Zhou et al., 2016) is a technique that localises class-specific image regions in a trained CNN.

This technique uses the last layer of a CNN to create a graphical representation for a particular output based on its weights. The resulting image is a heat-map representing which parts of the image have a higher influence in the output.

For example, Figure 3 depicts two different CAM representations for the 2D CNN models trained using the precipitation data of Helsinki and Rome. Warmer colours in the image represent higher weight values, so the network makes its decisions mostly based on the features located in those areas. The images in Figure 3 corroborate the intuitive idea that local weather patterns have a higher influence than distant ones when forecasting the weather of a particular location. The images have been overlaid with a coast map to serve as a reference for the relative position of the structures in the heat-maps.

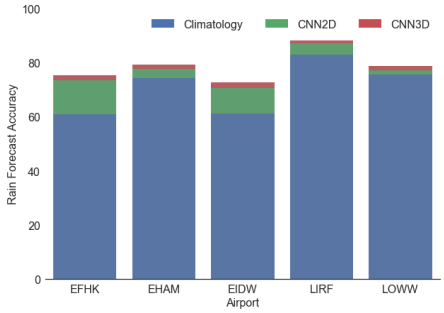


*Figure 2.* Accuracy results obtained for the different airports and methodologies.

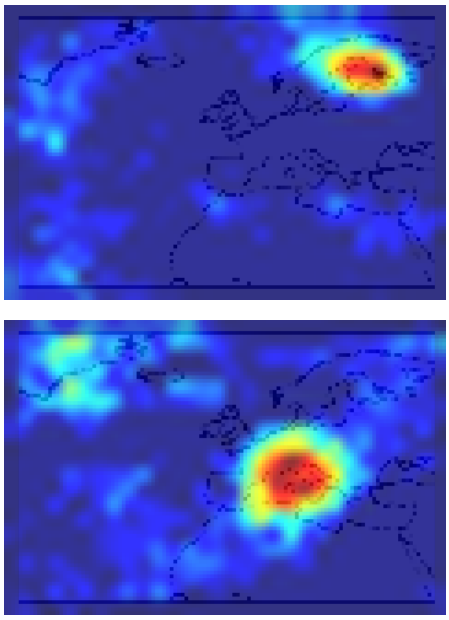

*Figure 3.* Example of the resulting Class Activation Maps for an ERA-Interim CNN trained using the observed precipitation at Helsinki-Vantaa airport, EFHK, (top) and Rome Fiumicino airport, LIRF, (bottom). Coastlines have been overlaid as a reference for readers.

This technique has been proven very useful for visually assessing the soundness of a CNN model. Another use could be for input variable selection, identifying NWP parameters which show a higher correlation with respect to the

class to be predicted.

### 3.4. Software and Data

The code used to run all the experiments included in this work and instructions on how to access the corresponding datasets are available at the following repository: http://github.com/prl900/DeepWeather

## 4. Conclusions and future work

This work demonstrates how CNNs can be directly applied to the output of numerical weather models by using observed data to annotate the samples. The design of the CNNs used in our experiments is very simple compared to some of the state-of-the-art architectures (Simonyan & Zisserman, 2014; Szegedy et al., 2015). Despite their simplicity, results show that convolutional layers can be used to interpret the output of weather models.

The NWP parameters used in the experiments are not directly correlated to the precipitation output variable. NWPs have many other variables, such as humidity, vorticity or even total precipitation, that could be used to forecast precipitation patterns with better accuracy. The purpose of this initial experiment was to demonstrate that CNNs can learn certain configurations of the atmospheric pressure systems and associate them with precipitation events (fronts, convection, etc).

Apart from weather model interpretation, these techniques open a new research pathway for the automatic generation of derived products. Some of the variables contained in NWPs are computed based on parameterisations or statistical models instead of physical equations. We think that these variables can be computed using CNN based models, potentially offering better results.

## Acknowledgements

The authors wish to acknowledge funding from the Australian Government Department of Education, through the National Collaboration Research Infrastructure Strategy (NCRIS) and the Education Investment Fund (EIF) Super Science Initiatives through the National Computational Infrastructure (NCI), Research Data Storage Infrastructure (RDSI) and Research Data Services Projects.

## References

Abadi, Martín, Agarwal, Ashish, Barham, Paul, Brevdo, Eugene, Chen, Zhifeng, Citro, Craig, Corrado, Greg S, Davis, Andy, Dean, Jeffrey, Devin, Matthieu, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Dee, DP, Uppala, SM, Simmons, AJ, Berrisford, Paul, Poli, P, Kobayashi, S, Andrae, U, Balmaseda, MA, Balsamo, G, Bauer, P, et al. The era-interim reanalysis: Configuration and performance of the data assimilation system. *Quarterly Journal of the royal meteorological society*, 137(656):553–597, 2011.

Gravelle, Chad M, Runk, Kim J, Crandall, Katie L, and Snyder, Derrick W. Forecaster evaluations of high temporal satellite imagery for the goes-r era at the nws operations proving ground. *Weather and Forecasting*, 31(4): 1157–1177, 2016.

Ji, Shuiwang, Xu, Wei, Yang, Ming, and Yu, Kai. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

Tripathi, Shivam, Srinivas, VV, and Nanjundiah, Ravi S. Downscaling of precipitation for climate change scenarios: a support vector machine approach. *Journal of Hydrology*, 330(3):621–640, 2006.

Wilson, Katie A, Heinselman, Pamela L, Kuster, Charles M, Kingfield, Darrel M, and Kang, Ziho. Forecaster performance and workload: Does radar update time matter? *Weather and Forecasting*, 32(1):253–274, 2017.

Xingjian, SHI, Chen, Zhourong, Wang, Hao, Yeung, Dit-Yan, Wong, Wai-Kin, and Woo, Wang-chun. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pp. 802–810, 2015.

Zhou, Bolei, Khosla, Aditya, Lapedriza, Agata, Oliva, Aude, and Torralba, Antonio. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, 2016.

## 2.4 Convolutional encoder-decoders for image to image regression

### 2.4.1 Introduction

The work presented at the 2017 ICML conference led and motivated us to continue researching the field and applications of Convolutional Neural Networks (CNN). Being aware of the demonstrated capacity that CNNs have to extract the spatial structure from input images, we decided to explore the possibilities that these techniques had in the field of meteorology.

For this new research work we focused on studying NWP parameterisations. There are physical processes in the atmosphere that cannot be represented by NWP regardless of its resolution. For these physical processes that cannot be directly resolved, NWP uses approximate models, which are known as parameterisations.

To perform the experiments in this work, we use the ERA-Interim global climate reanalysis dataset produced by the European Centre for Medium-Range Weather Forecasts (ECMWF). ERA-Interim makes available a large number of parameters, from which we choose geopotential height and total precipitation. We crop an extended area over Europe and extracted the selected parameters over the 1980-2016 period with a temporal resolution of 6 hours, resulting in dataset with more than 50,000 samples.

### 2.4.2 Research contribution

Autoencoders (Hinton and Salakhutdinov, 2006) are generic neural networks that recreate the input by performing a dimensionality reduction and subsequent expansion of the input space. This technique allows learning compressed representations of the data in an unsupervised manner. Convolutional autoencoders combine CNNs with autoencoders to efficiently learn compressed representations of images. A latter evolution of convolutional autoencoders demonstrates that similar networks offer an efficient way of performing image segmentation by training the model with samples of segmented images. Figure 2.4 contains a representation of the transformations performed by an encoder-decoder network to the geopotential field transforming it into a precipitation field for the same region.

We consider three different state-of-the-art convolutional encoder-decoder networks in the field of image segmentation: VGG-16 (Long, Shelhamer, and Darrell, 2015), Segnet (Badrinarayanan, Kendall, and Cipolla, 2017) and U-net (Ronneberger, Fischer, and Brox, 2015). These networks are modified to perform image regression tasks instead of segmentation by changing the loss function to MAE and tested with the task of predicting precipitation.

This work demonstrates that convolutional encoder-decoder neural networks can be used, as an alternative to parameterisations, to learn complex atmospheric processes using basic NWP fields as input. As far as we know, this is the first attempt to provide an end-to-end automated learning approach to derive new parameterisations from basic NWP fields using deep convolutional encoder-decoder networks. This work also demonstrates how popular deep learning networks in the field of image segmentation can be adapted to interpret and derive new weather parameters.

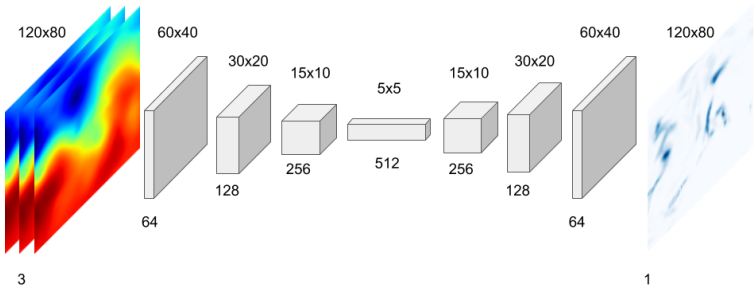


FIGURE 2.4: Representation of the transformations performed by the encoder-decoder network to the geopotential height field and its transformation into a field representing the total precipitation field for the same region.

### 2.4.3 Publication: Learning parameterisations with encoder-decoder convolutional neural networks

This work was first submitted in June 2018 to the "Monthly Weather Review", which is a peer-reviewed scientific journal published by the American Meteorological Society. It covers research related to analysis and prediction of observed and modeled circulations of the atmosphere, including technique development, data assimilation, model validation, and relevant case studies.

The editor of the journal wrote us a few weeks after our submission expressing his interest in the techniques presented in our work and proposing major changes in the contents of the manuscript. The main concern was about the language and references presented in our work, which where not suitable to the public of this journal.

The paper was adapted, following the indications received from the editor of the journal. The new version contains an introduction to machine learning and CNNs in the context of weather forecasting and NWP parameterisations. This new version was submitted in October 2018 to the same journal for its consideration.

MONTHLY WEATHER REVIEW

# Off-the-shelf deep neural networks learn state-of-the-art precipitation forecasting models

PABLO ROZAS LARRAONDO*, JIAN GUO†, IÑAKI INZA‡ AND JOSE A. LOZANO‡§

*Fenner School of Environment & Society, The Australian National University, Canberra, Australia*

ABSTRACT

Precipitation is one of the meteorological phenomena that has a great impact on human activities. The ability to accurately forecast precipitation is, therefore, an important task for current Numerical Weather Prediction (NWP) models. NWP precipitation is usually modelled using simplified or parameterised physical and chemical processes. The models representing these processes are often quite complex and require a high level of expertise in its design and tuning process. In this work, we devise a simple new methodology to derive precipitation from NWP geopotential fields using deep learning, a current hot-topic area of machine learning. Particularly, we consider a pipeline that first selects a set of geopotential height levels from the NWP and secondly uses these levels as the input to train three Convolutional Neural Network (CNN) models learning the precipitation field generated by the NWP. We include different experiments to compare the performance of each neural network as well as a comparison with alternative baseline machine learning methodologies. As far as we know, this paper covers the first attempt to model NWP parameterisations using generic machine learning methodologies.

## 1. Introduction

Numerical Weather Prediction (NWP) is the foundation of most weather forecasting products nowadays. Many organisations across the planet dedicate significant amounts of compute power to simulate the evolution of the atmosphere by solving the primitive equations that govern its dynamics. However, there are physical processes occurring in the atmosphere, such as convection, friction or radiation, that cannot be represented succinctly by NWP, regardless of its resolution (Stensrud 2009). For these physical processes, NWP uses approximate models called parameterisations.

NWP models resolve mid- and large-scale atmospheric processes under the assumptions of an adiabatic, frictionless atmosphere. Although these equations provide good approximations to the synoptic scale (i.e., $\sim$ 1000km) evolution of the atmosphere, exchanges of momentum, heat and moisture become important when simulating mid-range (36-72 hours) and sub-grid scale weather forecasts (Coiffier 2011). NWP parameterisations make it possible to correctly account for the various dynamic and radiative processes in the atmosphere that influence the weather.

Parameterisations, representing different atmospheric processes, are usually defined together inside "physical packages" that NWP models run (Louis et al. 1982). These parameterised processes interact with each other defining complex relationships. These relationships can be represented in the form of graphs, that often include feedback loops and nested dependencies. Often, a small modification in one of its parameters can lead to instabilities in the NWP output. The process of designing and maintaining these "physical packages", which define the different parameterisations and their relationships, is laborious and requires a high level of expertise (i.e. human intervention).

Although parameterisations based on statistical or probabilistic approaches – mainly used for modelling turbulence – can be found in the literature (Berner et al. 2017), most of them are based on determistic model equations representing simplifications of the physical processes they try to encapsulate. In this paper, we propose a novel approach for defining NWP parameterisations in which the underlying physics, determining the relationships between atmospheric variables, can be learned as an optimisation problem using machine learning techniques.

NWP models describe the state and evolution of the atmosphere, using a grid system that represents the spatial

*Corresponding author address: Pablo Rozas Larraondo, Fenner School of Environment & Society, ANU Building 141, Linnaeus Way, Canberra, ACT, 2601

E-mail: pablo.larraondo@anu.edu.au

†National Computational Infrastructure, ANU Building 143, Ward Road, ACT, 2601, Australia

‡Intelligent Systems Group, Computer Science Faculty, University of the Basque Country UPV/EHU, Paseo de Manuel Lardizabal, Donostia, 20018, Spain

§Basque Center for Applied Mathematics (BCAM), Mazarredo 14, Bilbao, 48009, Spain

and temporal components of the output data, defining a highly structured space. The field of machine learning that studies structured domains is commonly known as "structured prediction" (Taskar et al. 2005). Learning features on temporal and spatial dimensions are two common cases that structured prediction addresses (Gupta et al. 2010; Tran and Yuan 2012).

One of the main limitations in applying traditional machine learning methods to NWP datasets has been the difficulty to accurately represent the spatial and temporal structure in the data. There are many examples in the literature applying supervised machine learning methods to improve the quality of NWP. These examples typically use observational data to downscale or correct bias and systematic errors in NWP (Loridan et al. 2017; Gagne et al. 2014; Foley et al. 2012; Rozas-Larraondo et al. 2014). The problem with these methods is that they are generally applied in isolation to individual, or small clusters of grid cells, without accounting for the surrounding spatial and temporal information present in the data. These approaches also often suffer from a lack of capacity to represent complex non-linear relationships in the data.

Spatial data analysis has received significant attention in machine learning. The problem of learning and analysing datasets that contain spatial features is commonly studied in the field of computer vision. For a long time, methods based on engineered features, such as SIFT (Lowe 2004) and HOG (Dalal and Triggs 2005) have been the standard approach for applying machine learning to images and other high-dimensional regular gridded data. In the last decade, new methodologies based on deep neural networks have been introduced, offering substantial advantages over previous approaches.

Deep Learning (LeCun et al. 2015) methods have recently achieved unprecedented results in different supervised classification and regression tasks performed on different high dimensional datasets. These methods, based on the use of deep artificial neural networks, have surpassed human-level performance at different complex tasks such as image classification (Krizhevsky et al. 2012) or semantic description (Karpathy and Fei-Fei 2015). These architectures make use of Convolutional Neural Networks (CNN) (Krizhevsky et al. 2012), which have been proven to be very effective at capturing intrinsic features represented at different scales of an image. CNNs arrange its neurons in three dimensions (width, height, depth) and establish local connections to capture the spatial features in the input image. These structures extract the spatial relationships on the data learning to represent features represented at different scale levels on images.

Convolutional encoder-decoder networks are a type of CNN that provides state-of-the-art results at tasks such as image segmentation (Badrinarayanan et al. 2017), image denoising (Mao et al. 2016) or image-to-image regression (Isola et al. 2017). These networks are based on autoencoders (Hinton and Salakhutdinov 2006), which use CNNs to learn reduced but accurate representations of images, generalising its use to find relationships between different images. Convolutions in the encoder half of the network perform a feature selection process by reducing the dimensionality of the data. The decoder part enlarges the feature space mapping it to the output space. Encoder-decoder networks offer an effective method for learning the relationship between high dimensional input and output spaces, such as the ones defined by images or video.

Convolutional encoder-decoder networks have recently opened an active and promising field of research in areas such as medicine (Greenspan et al. 2016), astronomy (Shallue and Vanderburg 2018) or high-energy physics (Baldi et al. 2014). In the field of weather and climate sciences there is also an incipient interest in the introduction of convolutional networks to perform analysis and interpretation of NWP weather and climate datasets (Liu et al. 2016; Xingjian et al. 2015; Rozas Larraondo et al. 2017).

In this work, we demonstrate how existing deep learning encoder-decoder networks, proposed in the field of image segmentation, can be adapted to perform regression tasks and learn NWP parameterisations using basic fields as input. Using the ERA-Interim geopotential height field as input, we demonstrate how encoder-decoder networks can learn to simulate non-trivial physical processes that relate this field to precipitation. The experimental section contains a comparison between the results obtained with deep encoder-decoder networks and traditional methodologies, such as random forest. The simplicity and effectiveness of this method constitutes an interesting alternative to the more complex parameterisation models currently used in NWP.

## 2. Dataset and methodology

### *a. Dataset*

For this work, we use the NWP ERA-Interim (Dee et al. 2011) global climate reanalysis dataset produced by the European Centre for Medium-Range Weather Forecasts (ECMWF). This dataset contains data from 1979 every 6 hours to present. The spatial resolution of the data set is approximately 80 km (reduced Gaussian grid N128) on 60 vertical levels from the surface up to 0.1 hPa. ERA-Interim data is publicly accessible from ECMWF's Public Datasets web interface (Berrisford et al. 2011).

ERA-Interim makes available a large number of output parameters, from which we choose geopotential height ***(z)*** and total precipitation ***(tp)***. The experiments in the next section train models that can learn to predict total precipitation using geopotential heigh fields as the only input.

We crop an extended area over Europe (*latitude: [75, 15], longitude = [-50, 40]*), and select a subset of geopotential heights ***(z)*** at the following pressure levels of the

atmosphere: [1000, 900, 800, 700, 600, 500, 400, 300, 200, 100] hPa. The resulting geopotential height data are stored as a 4-dimensional numerical array with shape [54.023, 10, 80, 120] for the corresponding dimensions [time, height, latitude, longitude].

The ERA-Interim ***(tp)*** parameter represents the total amount of precipitation accumulated at each grid point during a 3-hour period measured in metres (1000 litres/squared metre). This field is aggregated to match the 6-hour frequency of the geopotential height field by adding 2 consecutive lead-time accumulations and scaled by 1.000 to represent millimetres (1 litre/squared metre) of rain. The result is a 3-dimensional numerical array with shape [54.023, 80, 120] for the corresponding dimensions [time, latitude, longitude]. Figure 1 represents the geographic area as well as the correspondence between the geopotential height and the total precipitation field time series.

### *b. Methodology*

NWP parameterisations of cloud and precipitation processes make simplifying assumptions, either for the purpose of computational efficiency or due to the uncertainty associated with these individual processes, in particular micro-physics and sub-grid scale interactions (Lopez 2007). These parameterisations, depending on their nature, are classified in two groups: deterministic or probabilistic.

Deterministic parameterisations are commonly found in NWP (Kain 2004; Tiedtke 1989), based on the definition of new physical models or approximations to describe atmospheric processes. Both linear and non-linear regression are used to define parameterisations (Crawford and Duchon 1999; Feng 2007).

Probabilistic parameterisations use statistical methods by means of stochastic dynamic equations (Berner et al. 2017). This area represents a novel and promising field of research for improving the quantification of NWP uncertainty or generating variability in ensemble based NWP.

An alternative approach is to consider parameterisations as a learning process using supervised machine learning methodologies to find the relationships between different variables, given a large enough dataset of historical records. Machine learning offers a broad collection of methodologies, such as random forests (Breiman 2001), support vector regression or neural networks that can solve generic regression problems. However, the application of these techniques is rarely found in the NWP parameterisation literature, apart from a few examples (Lipponen et al. 2013).

As mentioned in the introduction, the output of NWP defines an structured data set along the temporal and spatial dimensions. NWP datasets present a challenge for traditional machine learning methodologies because of their high dimensionality and volume.

Computer vision is a field of machine learning that applies algorithms to perform different tasks on high dimensional datasets, such as digital images. Common problems addressed in this field are image denoising, segmentation, upsampling or detection of objects. Algorithms in this area use a broad range of approaches based on the application of geometry, statistical or physical models.

Image classification, used in applications such as face detection or object detection, maps the high dimensional space of an image into a category (Haralick et al. 1973) or value (Takeda et al. 2007) that identifies the contents of the image. A more challenging problem is to map one image to another, because the dimensionality of the output space makes it very difficult for machine learning algorithms to find relationships between both spaces. Image segmentation is a common image-to-image classification problem that assigns a label to each pixel in an image to determine the boundaries of objects in the image. Image segmentation algorithms apply different methods such as logistic regression, support vector machines or random forests to images for classifying its pixels (Haralick and Shapiro 1985; Pal and Pal 1993; Pal 2005). Image denoising and upsampling are examples of image-to-image regression, in which the task consists in learning to predict the numerical values of each pixel in the output image, as opposed to a categorical value in image segmentation problems. This is a much harder problem than image-to-image classification and algorithms in this area normally include different forms of neighbour embedding to provide context for each pixel, using principal component analysis or random forests (Chang et al. 2004; Schulter et al. 2015).

Convolutional Neural Networks (CNN) (LeCun et al. 2010), are a particular type of feed-forward artificial neural networks specifically designed to analyse images. The application of fully-connected neural networks to high-dimensional data, such as images, has failed because of the large number of connections required to map the input space into the network. CNNs propose a simplified model in which neurons in one layer connect only to a local region of the next layer. This region is called kernel and its weights are shared across the image for each layer of the CNN, which results in translation invariance characteristics (Scherer et al. 2010). These kernels define a limited area in the width and height dimensions of an image but comprise the total depth dimension of images, which usually corresponds to the Red, Green and Blue (RGB) channels of a colour image. The convolution operation performs a cross-correlation operation by sliding the kernel across the image. The weights of the kernel are updated during training using backpropagation (Widrow and Lehr 1990) to minimise the error of a given loss function.

Autoencoders (Hinton and Salakhutdinov 2006) are generic neural networks that reproduce the input by per-

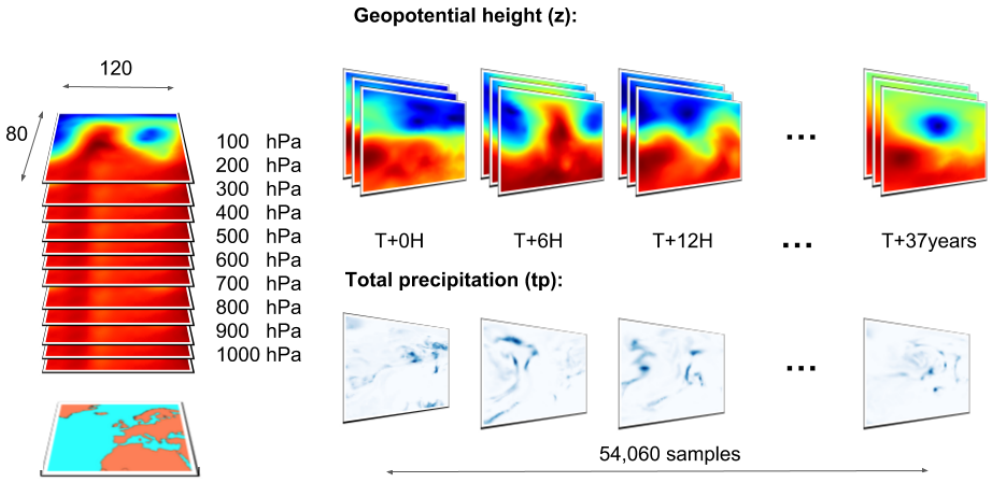


FIG. 1. Representation of the geographic area covered by the dataset and the geopotential height and total precipitation temporal series.

forming a dimensionality reduction and subsequent expansion of the input space. This technique allows learning compressed representations of the data in an unsupervised manner. Convolutional autoencoders (Masci et al. 2011) use CNNs to learn compressed representations of images. Convolutional encoder-decoder networks are a latter evolution of convolutional autoencoders which can perform image segmentation tasks, by mapping an input image to an output segmented image (Long et al. 2015).

Image segmentation using convolutional encoder-decoder networks has been an active area of research in recent years (Krizhevsky et al. 2012; Chen et al. 2018). Several techniques and network architectures have been proposed to improve the accuracy of the image segmenting process. The main constraint of convolutional encoder-decoder networks is the loss of spatial information, caused by dimensionality reduction pooling operations (Scherer et al. 2010). A consequence of this is that output images lack resolution and features often become blurry or ill defined. Many convolutional encoder-decoder network architectures have been proposed to mitigate the loss of spatial information: Segnet (Badrinarayanan et al. 2017) uses the transferred pool indices from its encoder to communicate the exact pixel position to the decoder. U-net (Ronneberger et al. 2015), on the other hand, enables precise localization by creating connections between the symmetric convolution and deconvolution operations to capture spatial context.

A generalisation of convolutional encoder-decoder networks is found in pixel-to-pixel regression or image-to-image translation networks (Isola et al. 2017). Existing segmentation networks can be easily modified to perform image-to-image translation tasks by substituting the classification loss function by a regression one, such as Root Mean Square Error (RMSE) or Mean Absolute Error (MAE).

Although most of the research on CNNs has been performed using colour images as input, the same networks have been proven to be effective with other types of image data, such as multispectral images in the fields of remote sensing (Hu et al. 2015) or volumetric medical imaging (Milletari et al. 2016). In weather forecasting, NWPs produce gridded outputs representing physical parameters for different vertical levels of the atmosphere. In this paper we propose the use of NWP fields as input channels for CNNs, combining them to analyse and interpret weather data.

The experimental section in this paper explores the application of convolutional encoder-decoder segmentation networks adapted to perform grid to grid regression (image to image regression) and learn NWP parameterisations directly from basic atmospheric fields. Specifically, we demonstrate how the total precipitation field can be learned, per grid point, using geopotential height as the only input.

## 3. Experimental design

The objective of the experiments presented in this work is to demonstrate that convolutional encoder-decoder neural networks can be used, as an alternative to NWP parameterisations, to learn complex atmospheric processes, such as precipitation, using basic NWP fields as input.

Geopotential height is closely related with air pressure and is one of the most fundamental physical variables used in weather forecasting. This field is simulated by NWP

models by solving the primitive equations of the atmosphere. Geopotential height has been traditionally used by weather forecasters to detect fronts, which separate masses of air of different properties, and ultimately forecast the occurrence and intensity of precipitation (Renard and Clarke 1965; Hope et al. 2014).

Existing NWP precipitation parameterisations normally use geopotential height in combination with other basic variables, such as temperature, humidity or vorticity as inputs. For this work we propose the use of geopotential height at different levels as the only input variable to forecast total precipitation for the following reasons: 1. Using a low number of inputs simplifies the encoder-decoder model and results in faster training process. 2. It demonstrates the ability of neural networks to find complex non-linear relationships between input and output grid fields that are not obviously correlated. 3. It serves as a nod to the hundreds of skilled human weather forecasters that are able to provide an accurate analysis using this field exclusively.

To perform the comparison between the different encoder-decoder models, we propose the use of a pipeline comprising two steps. The first step performs a variable selection process to determine the geopotential height levels that minimise the error at forecasting the total precipitation field. The second step compares the results of three state-of-the-art convolutional encoder-decoder architectures, in the field of image segmentation, at learning to predict the ERA-Interim total precipitation field.

The input dataset comprises ten levels of the geopotential height; however, due to the linear increase in the number of trainable parameters and the hardware requirements to train these convolutional encoder-decoder networks, we limit the number of input levels to three. Different methods for performing variable selection have been proposed (Saeys et al. 2007). These methods generally reduce the search space of input variables and optimise the construction of accurate predictors.

For our experiment, we build a simplified convolutional encoder-decoder network and perform an exhaustive search using all the different combinations of 1, 2 and 3 elements, out of the 10 geopotential field levels. The combination that reaches the lower error is chosen.

This simplified network has a similar architecture to the deeper encoder-decoder networks, but its complexity is reduced by limiting the number of layers and depth of the convolution operations. Performing a similar exhaustive search of input variables with the deeper, more complex networks, like the ones introduced in the next part of the experimental process, would be computationally infeasible. However, this simplified network allows a quick iteration across the whole feature space to identify the subset of geopotential heights that minimises the error of the precipitation field in the training set.

To identify the levels of geopotential height that produce the best precipitation results, the simple convolutional encoder-decoder network is trained with all the different combinations of the ten levels, without repetition. Therefore, the number of models that need to be trained is:

$$C_1(10)+C_2(10)+C_3(10)=\binom{10}{1}+\binom{10}{2}+\binom{10}{3}=175 \quad (1)$$

This process requires training the same network 175 times using the different combinations of the input data. The results of this variable selection process are used to compare the accuracy of the three state-of-the-art deep encoder-decoder convolutional networks in the second step of the pipeline. Therefore, the final validation comparing the accuracy of the different networks must be performed using a different dataset partition that remains unseen during the variable selection process (Reunanen 2003).

The whole pipeline can be seen as a variable selection process followed by the neural network model comparison process. The initial dataset, which contains 54,060 samples, is randomly split into the training and validation datasets, containing 80% and 20% of the data respectively, so that there is an even proportion of the different meteorological situations in both splits. These partitions are used to evaluate the differences in accuracy between the compared deep architectures. The variable selection process is performed internally using the training dataset, which is further split into 80% and 20% partitions to train and validate the different subsets of input variables. The final comparison between the architectures is performed using the initial 20% validation split, which does not intervene at neither the variable selection process nor the training of the different architectures. Figure 3 represents the proposed experiment and the relationship between the variable selection the model evaluation processes.

The network used for to perform the variable selection process is a simplification of the VGG-16 convolutional encoder-decoder network (Long et al. 2015), which has demonstrated state-of-the-art results in image segmentation tasks. Figure 2 represents the dimensionality transformations performed by the VGG-16 network to the input space using 3x3 convolutions using stride value of 2 to perform the spatial dimension reduction. The simplification proposed for this part of the experiment is to remove the last convolution operation in the encoder section. The information gets compressed to a depth of 256 channels instead of the 512 in the full VGG-16 network. This simplified network reduces significantly the total number of parameters when compared to the full VGG-16 which results in a significant reduction in the amount of compute resources required to train it. Each network is trained

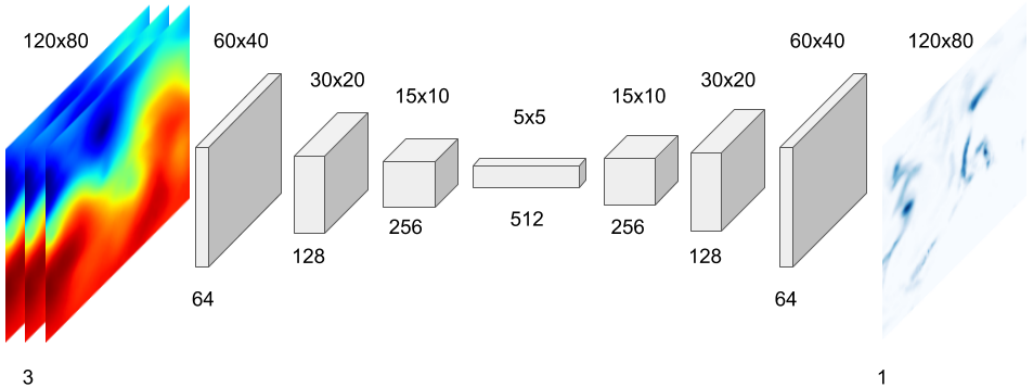


FIG. 2. Transformations in the dimensionality of the data performed by a VGG-16 encoder-decoder to map between the input and output spaces.

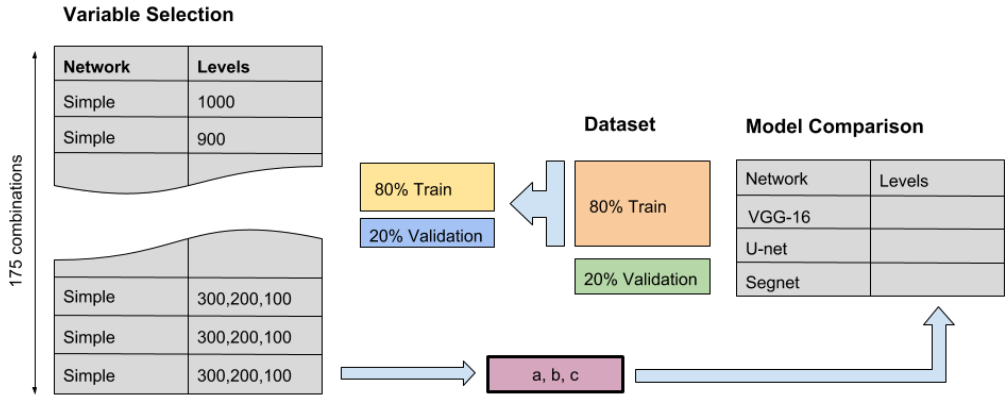


FIG. 3. Graphical representation of the experiment pipeline comprising the variable selection and the encoder-decoder network comparison processes.

during 20 epochs (iterations over the internal input training split) and their results are honestly computed using the MAE metric comparing the predicted outputs to the total precipitation field in the internal validation dataset. The results of this first experiment are therefore used to determine the geopotential height levels that minimise the error at forecasting total precipitation.

The second step of the pipeline focuses on finding which deep encoder-decoder convolutional network is more accurate at forecasting ERA-Interim's total precipitation field. For this part, we use the levels of geopotential height, computed previously, to compare the three different state-of-the-art segmentation networks.

We consider three different state-of-the-art convolutional encoder-decoder networks in the field of image segmentation: VGG-16 (Long et al. 2015), Segnet (Badrinarayanan et al. 2017) and U-net (Ronneberger et al. 2015). These networks are modified to perform regression tasks instead of classification by changing the loss function to MAE. To accomplish an honest comparison between these three networks, we build them using the same number of layers and depth of the convolution operations. The basic structure for all three networks is represented in Figure 2. Therefore, all three networks perform the same dimensionality transformations when estimating total precipitation from the geopotential height input. The difference between these networks resides in the number of convolution operations performed at each layer and in the configuration of the connections between layers. VGG-16 presents a linear architecture, in which each layer is only connected to the adjacent ones. Segnet computes the index of the max pooling operation at each of the encoder layers and communicates this value to its symmetric in the decoder, so they can be used in the up-sampling stage. The U-net decoder, on the other hand, concatenates the weights used in the encoder part to reconstruct the spatial information. The objective of this second half of the experimental process is to determine the network that provides the best accuracy when forecasting total precipitation.

All three networks are trained using the initial 80/20 split defined at the beginning of the experiment. This way, the validation split used to compare the accuracy of the different networks has remained unseen during at the vari-

able selection and training of the different networks. This method assures the independence and fairness of the results between both experiments.

The networks are trained during 50 epochs using the subset of geopotential heights selected in the first part of the experiment as input and the total precipitation field as output. The same optimiser (stochastic gradient descent), learning rate (0.01) and loss function (MAE) as in the variable selection process are used to train the three networks. These networks are then compared with the total precipitation field in the validation split dataset produced by ERA-Interim, to determine the error.

The models are implemented using the Keras (Chollet et al. 2017) framework and the TensorFlow (Abadi et al. 2016) back-end. These models, as well as a copy of the dataset used in the experiments, are available at this repository: `https://github.com/prl900/precip-encoder-decoders`.

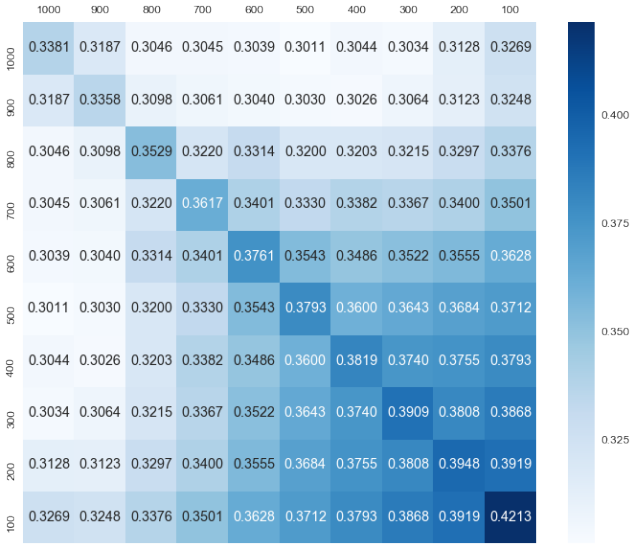


FIG. 4. Matrix representing the average validation MAE results for each simple encoder-decoder network trained with each possible combination of two geopotential levels.

## 4. Results and discussion

### *a. Variable selection process*

In the first part of the experiment we identify the levels of the geopotential height that produce a better estimate in the training set of the ERA-Interim total precipitation field. We train the simple encoder-decoder network 175 times as described in the previous section using the training split.

The resulting models are then compared using the internal validation split, which is formed with the remaining 20% of the initial training split. The MAE metric is used to compare the error in predicting total precipitation at each point of the grid. Figure 4 contains the MAE scores produced for each combination of any two geopotential height levels compared to the total precipitation field produced by the NWP. The results indicate that combinations of lower levels of the atmosphere produce better estimates of the precipitation field than the higher levels. The main diagonal of the matrix in Figure 4 represents the resulting errors when using a single geopotential level to train the network. Individually, the lower levels of the atmosphere present lower MAE values when forecasting precipitation, being 900 hPa the one with the lowest error. In the case of using two inputs, the lowest errors are found when combining low and mid levels of geopotential, being the combination of 1000 and 500 hPa levels the one with the lowest error.

Training the encoder-decoder network with three geopotential levels, results in a significant improvement in performance. Table 1 contains the five lowest error results and their corresponding atmospheric levels. Unfortunately, the results for three levels cannot be easily represented graphically. Compared to the previous results, there is a significant improvement in performance when a third level is added as input to the encoder-decoder network. This implies that the neural network is capable of finding internal relationships between the different levels of the atmosphere and relate them with precipitation events. The combination of 1000, 800 and 400 hPa geopotential heights results in the lowest error of the total precipitation field in the training partition. This result is surprisingly similar to the traditional practice in weather forecasting of using the sea-level, 850 and 500 hPa geopotential fields to determine the location of weather fronts and therefore, precipitation.

TABLE 1. Top 5 average MAE results when training the simple encoder-decoder network with every combination of three geopotential height levels to predict the ERA-Interim total precipitation field.

| *z levels* | *MAE* |
|---|---|
| 1000, 800, 400 | 0.2895 |
| 1000, 800, 500 | 0.2897 |
| 1000, 900, 500 | 0.2897 |
| 1000, 900, 400 | 0.2901 |
| 1000, 700, 400 | 0.2927 |

The MAE values represented in Table 1 are calculated using the average of the MAE results over the 120 by 80 grid area and for all the temporal entries in the validation dataset. Considering that the total precipitation field is expressed in millimetres, the error of these networks when forecasting total precipitation is, on average, less than 1/3th of litre per square metre in a 6-hour period, when compared to the values produced by the NWP.

*b. Deep convolutional networks' comparison*

For this part of the experimental process, we choose the subset of 1000, 800 and 400 hPa geopotential heights to evaluate the performance of deeper, state-of-the-art segmentation encoder-decoder networks adapted to perform regression tasks. The number of parameters and depth of these networks is significantly higher than the simplified network previously used to perform the selection of the geopotential levels. Training these deeper networks demands therefore significantly higher compute and memory resources, we use a compute node equipped with a Tesla P100-PCIE-16GB accelerator provided by the Australian National Computational Infrastructure.

Table 2 represents the total number of trainable parameters for each of these networks and the total amount of time required to train the different networks during 50 iterations (epochs) over the initial training split. The total number of trainable parameters provides an indication of the size of each network and the time value in this table gives an indication of the time required to train each network using TensorFlow models running on a P-100 NVIDIA GPU node.

TABLE 2. Number of parameters for each encoder-decoder network architectures and resulting training time for each network (50 epochs).

| *Network* | *Parameters* | *Time* [*hours*] |
|---|---|---|
| Simple (ref.) | 745,000 | 0.6 |
| VGG-16 | 16,467,469 | 4.7 |
| U-net | 7,858,445 | 2.4 |
| Segnet | 29,458,957 | 8.6 |

Figure 5 shows the learning process of the four different encoder-decoder networks during 50 iterations (epochs) over the training dataset. At the end of each epoch during training, the validation dataset is used to assess the error of the model and the improvement of the different models can be honestly compared using unseen data. At the beginning of the training process the network learns fast and it slows down as the training progresses. The reduction in the validation error is different for each network which flattens at different points and rates.

TABLE 3. Accuracy of the different networks using the validation split at the end of the training process.

| *Network* | *MAE* [mm] |
|---|---|
| Simple | 0.2893 |
| VGG16 | 0.2630 |
| Segnet | 0.2618 |
| U-net | 0.2386 |

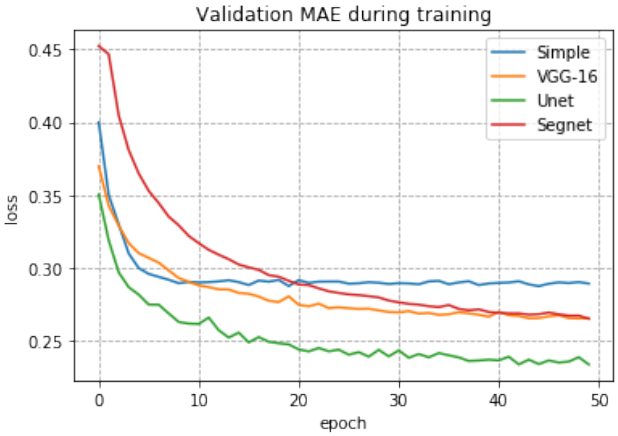


FIG. 5. Comparison of the evolution of the validation error during training for the four convolutional encoder-decoder networks over 50 epochs.

Considering the validation results in Figure 5 and the size of each network in Table 2, we highlight the behavior of U-net which shows the lowest validation error and is also the one with the lowest number of parameters of the three deep learning networks considered.

Figure 6 offers a visual comparison between the outputs generated by each model for an atypical atmospheric situation around the 15th of June of 1983. The first two columns from the left represent the 1000 hPa geopotential height and total precipitation, as produced by the ERA-Interim model. Total precipitation represents the total precipitation accumulated over the 6-hour period following the indicated time. In a similar way, and using the same colour scale, the next 4 columns represent the precipitation generated by the different encoder-decoder networks.

The spatial structure and intensity of the precipitation field is represented differently by each network, with slight variations in respect of the ERA-Interim reference output. Different convolutional encoder-decoder networks use different methods to reconstruct the spatial information lost during the encoding phase. Apart from capturing the spatial structure of the precipitation field, the different networks have to provide accurate results for the precipitation intensity at each grid point.

*c. Statistical analysis of the results*

To compare the performance of the different convolutional encoder-decoder architectures we use the validation split to extract the total precipitation at the closest grid point to nine different cities produced by each network. Figure 7 represents the geographical location of these nine cities within the region of our dataset. These cities are located in different climatic zones and present distinct precipitation patterns.

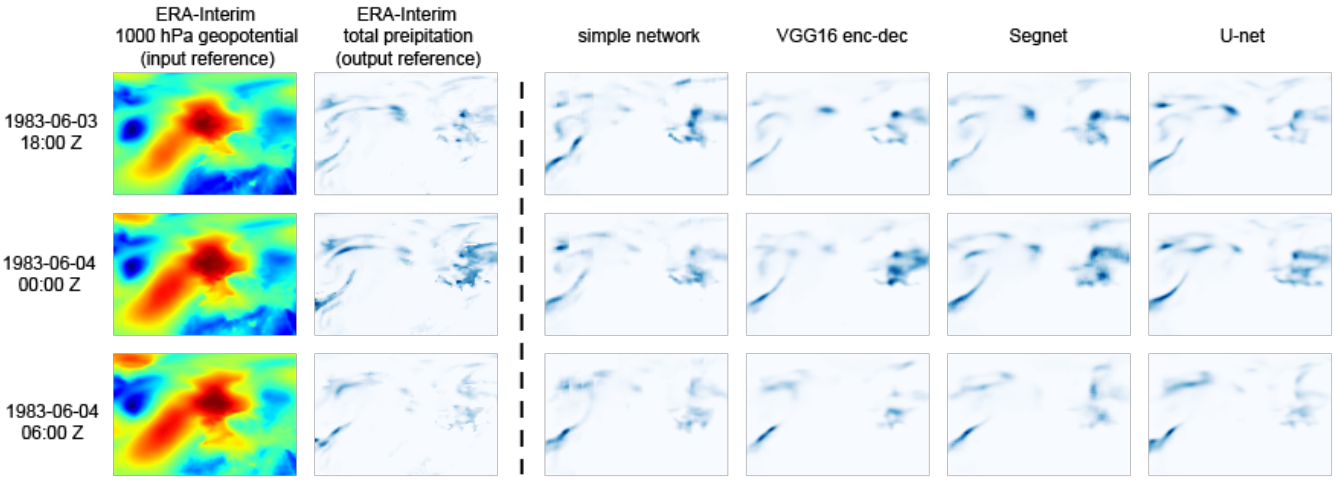


FIG. 6. Visual comparison between the total precipitation field generated by the different networks. ERA-Interim 1000 hPa geopotential height and total precipitation fields are included for reference.

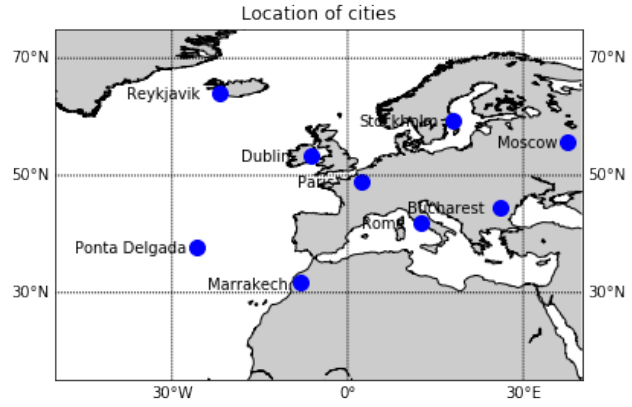


FIG. 7. Location of the nine different cities within the comprised region.

Results are assessed using the ERA-Interim total precipitation field as reference for the same grid points using the signed error – or bias – metric. This metric provides information about possible biases and distribution of the error as opposed to the previously used MAE, which does not provide information about the sign of the error. For each city and point in time the error in predicting total precipitation is calculated. These results are then aggregated by city and type of network. Figure 8 uses a violin plot (Hintze and Nelson 1998) to represent the error results at each location for the different architectures. A violin plot proposes a modification to box plots adding the density distribution information to the basic summary statistics inherent in box plots. The horizontal blue bar towards the centre of each of the violins in Figure 8, represents the mean. The lower part in each plot, shows the mean $(\mu)$ and standard deviation $(\sigma)$ of the error values for each network and location. The shape of the violin gives a visual indication of each model's performance. Wider and sharper violin shapes around the 0 value provide an indication of good network performance.

In order to statistically compare the results, we use the methodology proposed by Demsar (Demsar 2006) to assess the statistical significance of the differences between the error results of each network in the nine locations. The initial Friedman test rejects the null hypothesis of similarity among the 4 convolutional encoder-decoder networks. This justifies the use of post-hoc bivariate tests, Nemenyi (Pohlert 2014) our case, to assess the significance of the differences between the different pairs of encoder-decoder networks.

The results of these tests are graphically expressed using Critical Difference (CD) diagrams. The Nemenyi test pairwisely compares the error results between any two architectures. Differences are considered significant if the corresponding average rank differs by at least one critical difference.

Figure 9 shows a CD diagram representing the results of the Nemenyi test $(\alpha = 0.05)$ using the error values at the nine locations for each convolutional encoder-decoder network.

CD diagrams make a pairwise comparison between methods, connecting the architectures for which no significant statistical differences are found, or in other words, those whose distance is less than the fixed critical difference, shown at the top of Figure 9. Networks ranked with lower values in CD diagrams imply higher error values. These tests have been performed using the `scmamp` R package, which is publicly available at the Comprehensive R Archive Network (CRAN) (Calvo and Santafe 2016).

Statistical differences are found between all pairs of networks. As can be seen in the CD diagram, the performance of U-net is significantly better at forecasting total precip-

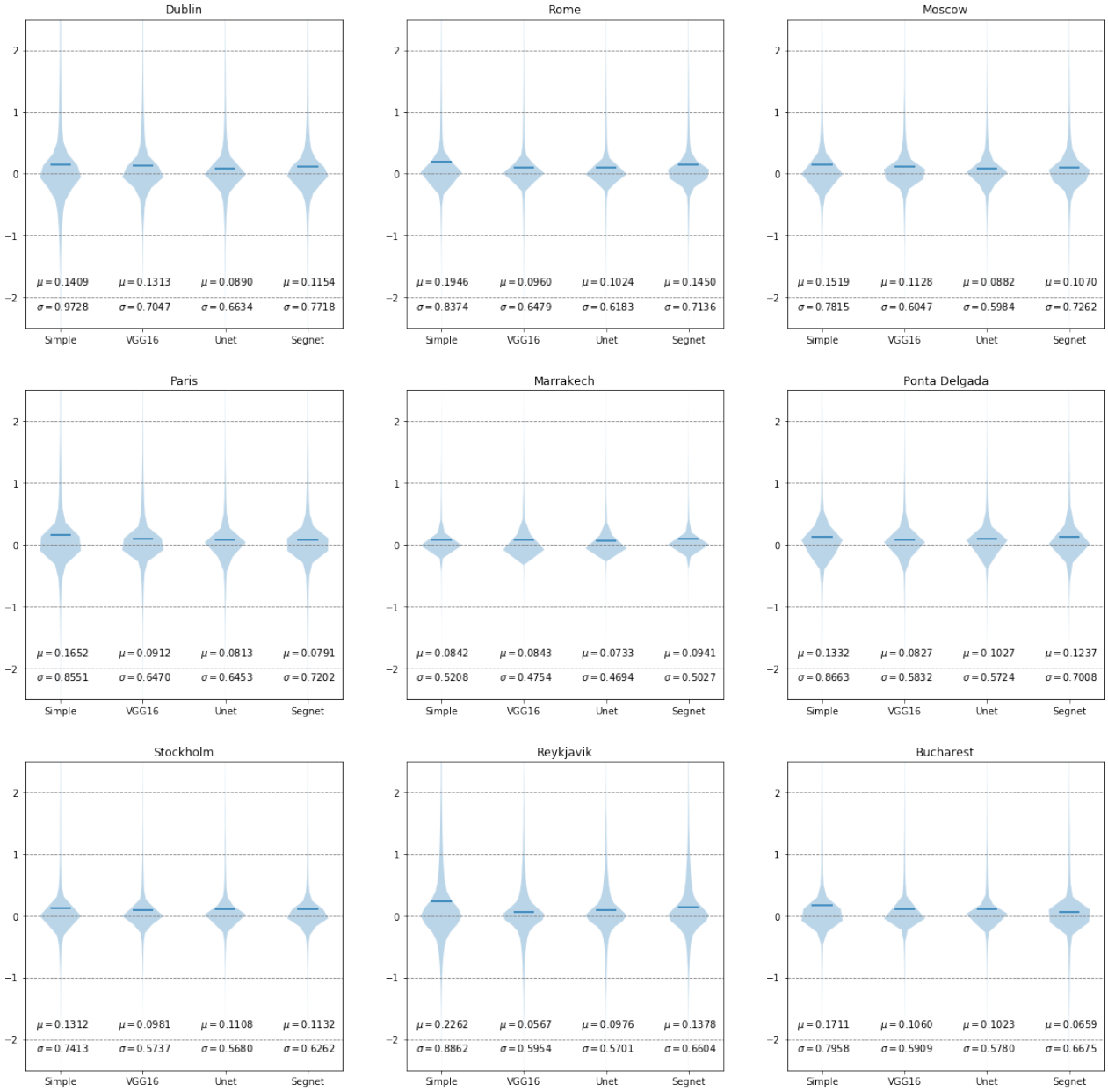


FIG. 8. Representation of the error density function and the mean ($\mu$) and standard deviation ($\sigma$) values for the different architectures at each city.

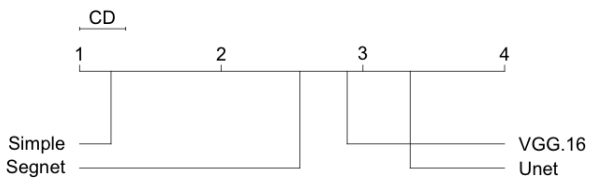


FIG. 9. Critical Differences comparing the 4 convolutional encoder-decoder architectures. $\alpha = 0.05$

itation than the other 3 networks. VGG-16 and Segnet have a significantly lower performance but they are still considerably better than the simple convolutional encoder-decoder described in the first part of the experimental process. These results imply that U-net based architectures provide better results when forecasting total precipitation, using geopotential height as input. Considering the results presented in Table 2, U-net requires approximately half the GPU and memory resources than VGG-16 or a quarter than Segnet equivalent networks.

*d. Comparison with traditional methods*

This section is intended to provide readers with an understanding of the qualitative improvement that deep convolutional architectures offer when compared to previous machine learning methodologies.

First of all, we provide a baseline comparison of precipitation forecast using persistence. We consider two different constant rain fields using zero and the average precipitation over the area of study. This two situations represent the cases where we always predict that there is no precipitation or the average precipitation based on the climatology at each grid point. The MAE results of comparing these two scenarios to the ERA-Interim precipitation values over the validation dataset are represented in Table 4.

TABLE 4. Baseline comparison of precipitation forecast using constant values over the whole area.

| *Constant value* [*mm*] | *MAE* [mm] |
|---|---|
| 0 (No precipitation) | 0.3417 |
| 0.45 (Mean precipitation) | 0.4845 |

The results in Table 4 indicate that forecasting no precipitation provides a substantially better forecast than using the average value. The mean in this case results in a poor estimate for the precipitation field. The distribution of precipitation has a high variability, precipitations concentrates around well defined clusters and in most of the grid points there is no precipitation. This is why using zero precipitation performs better than the mean value for the MAE metric.

A common technique in computer vision is to train model that learn to predict the value of a pixel using a patch containing the surrounding pixels in the input space (Pal 2005; Mueller et al. 2016). We approach the problem of learning the total precipitation field from the 3 levels of geopotential height determined previously, but using traditional regression methodologies. We choose three common regression algorithms: linear regression, Least Absolute Shrinkage and Selection Operator (LASSO) and random forest regressor. Due to the high dimensionality of the data, we train the different algorithms using increasingly larger patches (1,3,5,7 and 9) comprising the 3 levels in the input to predict the central pixel in the output.

As the size of the patch increases the overlap area between two adjacent patches is larger and the size of the dataset increases and the resulting dataset cannot fit in memory of high-end machines. To train the different models we randomly sample 100.000 patches of each size.

Table 5 contain the MAE results of the different regression models for each patch size. Looking at the results, it can be concluded that none of these techniques is capable of learning the relationships between the geopotential and total precipitation fields of a NWP. Also, because these models are trained using a narrow patch or window of the input field, they cannot even match the accuracy of the naive approaches proposed at the beginning of this section.

Figure 10 shows the output generated by each regression algorithm for the same meteorological situation presented in Figure 6. The models are not able to provide the sharpness necessary to represent the precipitation field. I can be seen that the output generated by Random forest provides a light improvement in detecting the position of the precipitation regions, possibly because is the only nonlinear method. However, the capacity of this methodology is not enough to resolve this problem.

TABLE 5. Comparison of the accuracy level for the different regression models.

| *method* | patch size | | | | |
|---|---|---|---|---|---|
| | 1 | 3 | 5 | 7 | 9 |
| Lin. Reg. | 0.5281 | 0.5061 | 0.5055 | 0.5054 | 0.5105 |
| LASSO | 0.5281 | 0.5056 | 0.5049 | 0.5034 | 0.5034 |
| RF | 0.5437 | 0.4924 | 0.4903 | 0.4862 | 0.4851 |

## 5. Conclusions and future work

This work demonstrates the suitability of convolutional encoder-decoder networks in learning NWP parameterisations using only the geopotential height field to predict total precipitation. Considering the results presented in this manuscript, it is noticeable that the geopotential height at different levels of the atmosphere contains enough information to infer the precipitation field, as shown in Figures 6 and 8. There are many other physical variables simulated by NWP, such as temperature or humidity, that contain valuable information to determine the location and intensity of precipitation. Although adding these variables as inputs to the encoder-decoder convolutional networks improves significantly the accuracy of the precipitation results, this paper focuses on demonstrating the capacity of these networks to find relationships between different variables. The spatial structures at different levels of the atmosphere of the geopotential field contains enough information to estimate a precipitation field with reasonable accuracy.

The networks presented in the experimental section of this paper uses a loss function to train the models based on the mean absolute error metric. Verification of NWP precipitation, can be based on a wide variety of metrics. For example, there are effects such as the 'Double Penalty'


FIG. 10. Output of the three regression methods (patch size = 7) using the same dates shown in Figure 6.

(Mass et al. 2002; Bougeault 2003) that become important when atmospheric structures are correctly represented in terms of their shape and intensity but not in their position. We consider that further research on defining new loss functions based on existing verification methods, that can account for errors in the spatial structure (Rossa et al. 2008), would lead to more accurate parameterisation models.

In this work we present a model that is trained to output precipitation values from the geopotential field. The quality of our model is therefore limited by the quality of the underlying NWP parameterisation used to simulate precipitation. The same encoder-decoder network could ideally be trained using observed precipitation data, resulting in a better model. Unfortunately, the research community has currently no access to observational datasets that match the spatial and temporal resolution of NWP. However, the rapid evolution of satellite and earth observation technologies open the possibility of having new high quality observations at a global scale in the future.

Lastly, another promising evolution of the methodology presented here, is to modify the convolutional encoder-decoder networks introducing recurrent structures. Recurrent neural networks (Mikolov et al. 2010) have demonstrated remarkable results in the area of time-series analysis and speech recognition and they open an interesting new line of research for models that can learn from both the spatial and temporal components of NWP data.

*Acknowledgments.* We would like to thank the National Computational Infrastructure (NCI) at the Australian National University and the University of the Basque Country for their support and advice in carrying out this research work.

We are grateful for the support of the Basque Government (IT609-13), the Spanish Ministry of Economy and Competitiveness (TIN2016-78365-R).

Jose A. Lozano is also supported by BERC program 2014-2017 (Basque Gov.) and Severo Ochoa Program SEV-2013-0323 (Spanish Ministry of Economy and Competitiveness).

## References

Abadi, M., and Coauthors, 2016: Tensorflow: A system for large-scale machine learning. *OSDI*, Vol. 16, 265–283.

Badrinarayanan, V., A. Kendall, and R. Cipolla, 2017: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39 (12)**, 2481–2495.

Baldi, P., P. Sadowski, and D. Whiteson, 2014: Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, **5**, 4308.

Berner, J., and Coauthors, 2017: Stochastic parameterization: Toward a new view of weather and climate models. *Bulletin of the American Meteorological Society*, **98 (3)**, 565–588.

Berrisford, P., and Coauthors, 2011: The era-interim archive version 2.0. Shinfield Park, Reading, 23 pp.

Bougeault, P., 2003: The wgne survey of verification methods for numerical prediction of weather elements and severe weather events. *Toulouse: Météo-France*.

Breiman, L., 2001: Random forests. *Machine learning*, **45 (1)**, 5–32.

Calvo, B., and G. Santafe, 2016: scmamp: Statistical comparison of multiple algorithms in multiple problems. *The R Journal*.

Chang, H., D.-Y. Yeung, and Y. Xiong, 2004: Super-resolution through neighbor embedding. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, IEEE, Vol. 1, I–I.

Chen, L.-C., G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, 2018: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, **40 (4)**, 834–848.

Chollet, F., and Coauthors, 2017: Keras (2015).

Coiffier, J., 2011: *Fundamentals of numerical weather prediction*. Cambridge University Press.

Crawford, T. M., and C. E. Duchon, 1999: An improved parameterization for estimating effective atmospheric emissivity for use in calculating daytime downwelling longwave radiation. *Journal of Applied Meteorology*, **38 (4)**, 474–480.

Dalal, N., and B. Triggs, 2005: Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, IEEE, Vol. 1, 886–893.

Dee, D. P., and Coauthors, 2011: The era-interim reanalysis: Configuration and performance of the data assimilation system. *Quarterly Journal of the royal meteorological society*, **137 (656)**, 553–597.

Demsar, J., 2006: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, **7**, 1–30.

Feng, J., 2007: A 3-mode parameterization of below-cloud scavenging of aerosols for use in atmospheric dispersion models. *Atmospheric Environment*, **41 (32)**, 6808–6822.

Foley, A. M., P. G. Leahy, A. Marvuglia, and E. J. McKeogh, 2012: Current methods and advances in forecasting of wind power generation. *Renewable Energy*, **37 (1)**, 1–8.

Gagne, D. J., A. McGovern, and M. Xue, 2014: Machine learning enhancement of storm-scale ensemble probabilistic quantitative precipitation forecasts. *Weather and Forecasting*, **29 (4)**, 1024–1043.

Greenspan, H., B. van Ginneken, and R. M. Summers, 2016: Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, **35 (5)**, 1153–1159.

Gupta, A., M. Hebert, T. Kanade, and D. M. Blei, 2010: Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. *Advances in neural information processing systems*, 1288–1296.

Haralick, R. M., K. Shanmugam, and Coauthors, 1973: Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, **(6)**, 610–621.

Haralick, R. M., and L. G. Shapiro, 1985: Image segmentation techniques. *Computer vision, graphics, and image processing*, **29 (1)**, 100–132.

Hinton, G. E., and R. R. Salakhutdinov, 2006: Reducing the dimensionality of data with neural networks. *science*, **313 (5786)**, 504–507.

Hintze, J. L., and R. D. Nelson, 1998: Violin plots: a box plot-density trace synergism. *The American Statistician*, **52 (2)**, 181–184.

Hope, P., and Coauthors, 2014: A comparison of automated methods of front recognition for climate studies: A case study in southwest western australia. *Monthly Weather Review*, **142 (1)**, 343–363.

Hu, F., G.-S. Xia, J. Hu, and L. Zhang, 2015: Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, **7 (11)**, 14 680–14 707.

Isola, P., J.-Y. Zhu, T. Zhou, and A. A. Efros, 2017: Image-to-image translation with conditional adversarial networks. *arXiv preprint*.

Kain, J. S., 2004: The kain–fritsch convective parameterization: an update. *Journal of applied meteorology*, **43 (1)**, 170–181.

Karpathy, A., and L. Fei-Fei, 2015: Deep visual-semantic alignments for generating image descriptions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3128–3137.

Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012: Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1097–1105.

LeCun, Y., Y. Bengio, and G. Hinton, 2015: Deep learning. *nature*, **521 (7553)**, 436.

LeCun, Y., K. Kavukcuoglu, and C. Farabet, 2010: Convolutional networks and applications in vision. *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, IEEE, 253–256.

Lipponen, A., V. Kolehmainen, S. Romakkaniemi, and H. Kokkola, 2013: Correction of approximation errors with random forests applied to modelling of cloud droplet formation. *Geoscientific Model Development*, **6 (6)**, 2087–2098.

Liu, Y., and Coauthors, 2016: Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *arXiv preprint arXiv:1605.01156*.

Long, J., E. Shelhamer, and T. Darrell, 2015: Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Lopez, P., 2007: Cloud and precipitation parameterizations in modeling and variational data assimilation: A review. *Journal of the Atmospheric Sciences*, **64 (11)**, 3766–3784.

Loridan, T., R. P. Crompton, and E. Dubossarsky, 2017: A machine learning approach to modeling tropical cyclone wind field uncertainty. *Monthly Weather Review*, **145 (8)**, 3203–3221.

Louis, J.-F., M. Tiedtke, and J.-F. Geleyn, 1982: A short history of the pbl parameterization at ecmwf. *Workshop on Planetary Boundary Layer parameterization, 25-27 November 1981*, ECMWF, Shinfield Park, Reading, ECMWF, 59-79.

Lowe, D. G., 2004: Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, **60 (2)**, 91–110.

Mao, X., C. Shen, and Y.-B. Yang, 2016: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *Advances in Neural Information Processing Systems*, 2802–2810.

Masci, J., U. Meier, D. Cireşan, and J. Schmidhuber, 2011: Stacked convolutional auto-encoders for hierarchical feature extraction. *International Conference on Artificial Neural Networks*, Springer, 52–59.

Mass, C. F., D. Ovens, K. Westrick, and B. A. Colle, 2002: Does increasing horizontal resolution produce more skillful forecasts? *Bulletin of the American Meteorological Society*, **83 (3)**, 407–430.

Mikolov, T., M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, 2010: Recurrent neural network based language model. *Eleventh Annual Conference of the International Speech Communication Association*.

Milletari, F., N. Navab, and S.-A. Ahmadi, 2016: V-net: Fully convolutional neural networks for volumetric medical image segmentation. *3D Vision (3DV), 2016 Fourth International Conference on*, IEEE, 565–571.

Mueller, N., and Coauthors, 2016: Water observations from space: Mapping surface water from 25 years of landsat imagery across australia. *Remote Sensing of Environment*, **174**, 341–352.

Pal, M., 2005: Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, **26 (1)**, 217–222.

Pal, N. R., and S. K. Pal, 1993: A review on image segmentation techniques. *Pattern recognition*, **26 (9)**, 1277–1294.

Pohlert, T., 2014: The pairwise multiple comparison of mean ranks package (pmcmr). *R package*, 2004–2006.

Renard, R. J., and L. C. Clarke, 1965: Experiments in numerical objective frontal analysis.

Reunanen, J., 2003: Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, **3 (Mar)**, 1371–1382.

Ronneberger, O., P. Fischer, and T. Brox, 2015: U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, Springer, 234–241.

Rossa, A., P. Nurmi, and E. Ebert, 2008: Overview of methods for the verification of quantitative precipitation forecasts. *Precipitation: Advances in measurement, estimation and prediction*, Springer, 419–452.

Rozas-Larraondo, P., I. Inza, and J. A. Lozano, 2014: A method for wind speed forecasting in airports based on nonparametric regression. *Weather and Forecasting*, **29 (6)**, 1332–1342.

Rozas Larraondo, P., I. Inza, and J. A. Lozano, 2017: Automating weather forecasts based on convolutional networks. *ICML 17 Workshop on Deep Structured Prediction*.

Saeys, Y., I. Inza, and P. Larrañaga, 2007: A review of feature selection techniques in bioinformatics. *bioinformatics*, **23 (19)**, 2507–2517.

Scherer, D., A. Müller, and S. Behnke, 2010: Evaluation of pooling operations in convolutional architectures for object recognition. *International Conference on Artificial Neural Networks*, Springer, 92–101.

Schulter, S., C. Leistner, and H. Bischof, 2015: Fast and accurate image upscaling with super-resolution forests. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3791–3799.

Shallue, C. J., and A. Vanderburg, 2018: Identifying exoplanets with deep learning: A five-planet resonant chain around kepler-80 and an eighth planet around kepler-90. *The Astronomical Journal*, **155 (2)**, 94.

Stensrud, D. J., 2009: *Parameterization schemes: keys to understanding numerical weather prediction models*. Cambridge University Press.

Takeda, H., S. Farsiu, and P. Milanfar, 2007: Kernel regression for image processing and reconstruction. *IEEE Transactions on image processing*, **16 (2)**, 349–366.

Taskar, B., V. Chatalbashev, D. Koller, and C. Guestrin, 2005: Learning structured prediction models: A large margin approach. *Proceedings of the 22nd international conference on Machine learning*, ACM, 896–903.

Tiedtke, M., 1989: A comprehensive mass flux scheme for cumulus parameterization in large-scale models. *Monthly Weather Review*, **117 (8)**, 1779–1800.

Tran, D., and J. Yuan, 2012: Max-margin structured output regression for spatio-temporal action localization. *Advances in neural information processing systems*, 350–358.

Widrow, B., and M. A. Lehr, 1990: 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, **78 (9)**, 1415–1442.

Xingjian, S., Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, 2015: Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, 802–810.

# Chapter 3

# Conclusions and future work

This doctoral thesis has been focused on the study of the methodologies and applications of machine learning within the context of weather forecasting. Weather forecasting produces large amounts of data from both observations and numerical simulations. The interpretation of the information contained in these data sets presents interesting challenges in terms of the volume and its high dimensionality. The idea behind our work has been to explore the application of models that can learn to interpret and extract patterns out of these data. In particular, we have explored methodologies that use observed data to enhance the output of numerical simulated weather, and methods for deriving precipitation from basic weather fields.

### 3.0.1 Conclusions

The first part of this doctoral thesis addresses the application of regression methods using circular variables. Most of the machine learning methodologies available for performing regression are designed to work with linear variables. Circular variables cannot be naturally represented by these algorithms and, in most cases, they are treated as linear variables, which often leads to unsatisfactory results. The first approach that we explored was the idea of using "cyclic kernels" to perform a non-parametric regression of wind speed. This method allows selecting observed data points around an specific directional component to perform a regression that improves wind speed forecasted by NWP. The second idea that we explored was a method for incorporating circular variables in regression trees. This method builds upon the already existing concept of circular trees proposing a new way of training the trees to improve their accuracy and computational performance.

For our third contribution, we changed subject to explore the application of convolutional neural networks into a rain classification or detection problem. Using NWP data as input, treated as an image, we applied convolutional networks to interpret the spatial information contained in these fields and predict the existence of precipitation using observed data for specific collections. This work demonstrates that neural networks can be used to interpret the spatial structure and find correlations with other atmospheric processes, such as precipitation.

Our fourth and last contribution continues exploring the application of convolutional neural networks but with a more challenging task, inferring parameterised weather fields from basic variables. In this case, we explore the use of encoder-decoder networks to derive precipitation using atmospherical pressure fields at several heights. This work demonstrates how convolutional encoder-decoder networks, originally designed to perform segmentation tasks, can be modified to solve difficult problems in the field of weather forecasting, such as learning the physics that relates complex atmospherical processes with each other.

The main contributions in this doctoral thesis can be summarised as follows:

- Non-parametric regression can significantly improve NWP fields by eliminating biases.
- Cyclic kernels enable non-parametric regression accounting for non resolved topographic effects.
- Circular regression trees with contiguous partitions significantly improve the accuracy of regression for circular variables compared to non-contiguous ones.
- Our proposed implementation of circular regression trees provides a significant improvement in computational performance and accuracy over the previous version.
- Convolutional neural networks (CNN) can extract the underlying spatial and temporal structure from basic NWP fields.
- A methodology to visualise the areas in a map that have higher influence in CNN models is proposed.
- Convolutional encoder-decoder networks have the capacity to learn the relationships between atmospheric variables.
- This method can be used as an alternative to NWP parameterisations, being U-net the one that offers the best accuracy from all the compared networks.

This doctoral work was set up and conceived as an exploratory work on the application of machine learning in the field of weather forecasting. In this context, one of the main challenges has been understanding and communicating concepts across both domains and communities. The machine learning community has grown out of the more generic computer science and statistics fields. The weather forecasting community, in spite of having strong ties to the numerical simulation and high performance aspects of computer science, it has been mainly driven along the physics field. For example, some concepts, such as stochastic modeling methods, have been developed in concurrently following different paths, which has resulted in the development of specific terminology to refer to often similar concepts.

This challenge has become patent when presenting our ideas to journals belonging to both fields. Although NWP data sets provide an excellent resource for experimenting with generic machine learning algorithms, they require an understanding about the structure and significance of the different variables they represent. Communicating our ideas and contributions to the machine learning community, using a methodological approach, became difficult for the amount of context and domain-specific knowledge required to present our work.

The experience has also been similar in the other direction, when presenting our work to weather forecasting audiences. Introducing new machine learning approaches and methodologies to the weather forecasting community has required us to dedicate an special effort in communicating and translating machine learning concepts in the domain-specific language.

Personally, the experience of working in this doctorate during these years, has led me to the appreciation of the amount of work required to make contributions in

science and a much better understanding of the process. Although my level of understanding about the different machine learning methodologies and techniques has significantly improved through these years, the number of questions and unknowns in my head has only increased. Each of the papers that we published concludes presenting new interesting questions and challenges waiting to be explored.

The outcomes of the research we have carried out, have also opened a window for optimism. Realising about the new perspective that machine learning brings to weather forecasting problems and the potential to improve our understanding about the atmosphere has been an encouraging experience. I personally hope to continue learning from and contributing to the development of new links between the machine learning and meteorological communities.

### 3.0.2 Future Work

The work carried out on the first two publications, on the use of circular variables into regression models, has not been explored in the case of ANN models. Exploring the possibilities that neural networks and deep learning models offer in the space of circular variables remains an interesting topic of research. Our literature research about previous works in this field, makes us think that is mostly an unexplored area in neural networks and this is something that I would like to explore.

The software for generating circular trees, published in the Environmental Modelling Software journal, provides a basic implementation of regression tree. Research on classification and regression trees has demonstrated that ensemble methods greatly outperform individual trees. Also, there are techniques such as pruning, balancing or smoothing (Breiman et al., 1984; Quinlan, 1993), that significantly improve the performance of trees. There are packages such as Python scikit-learn (Pedregosa et al., 2011), that offer a large collection of machine learning models and tools to facilitate the training and testing of these models. Offering our implementation of circular regression tree, as a module in one of these generic machine learning packages, would considerable contribute to the diffusion and expansion of this methodology. Another benefit would be that functionalities available for the generic binary tree class, such as ensemble methods, could then be applied to circular trees.

In the area of deep learning, we have limited our study to the application of CNNs to NWP fields. We have demonstrated that this kind of networks are able to extract the spatial information contained in gridded weather fields. The temporal component of weather forecasting, which describes the evolution of the different atmospheric parameters, has not been explored. Recurrent Neural Networks (RNN) (Williams and Zipser, 1989), is a class of neural network where the output of the network is feed back as input to the next step, providing a model to simulate sequential data. The combination of CNNs and RNNs has been already proposed to forecast precipitation from radar data (Xingjian et al., 2015) but the representation of the temporal component of weather forecasting can be explored using other kind of networks. For example, temporal convolution (Bai, Kolter, and Koltun, 2018) has been recently proposed as an alternative to RNN methods demonstrating to perform better in different applications.

Lastly, another idea that I would like to explore is the automatic generation of Terminal Aerodrome Forecast (TAF) messages. TAFs complement and use similar encoding to METAR reports but describe the evolution of the meteorological conditions at a specific airport. Nowadays they are produced by human forecasters based

on the information provided by NWP and observations. The idea at the start of this thesis was to explore the possibility of creating such system. TAFs contain information relative to different variables, such as the wind, clouds, visibility conditions or pressure. Soon, at the beginning of this work, I realised that each of these variables contains enough complexities which need to be solved as independent pieces of research. The idea of implementing such system is still relevant and proposes a challenging and interesting piece of work that I would like to consider in the future.

# Bibliography


Adrianto, Indra, Theodore B Trafalis, and Valliappa Lakshmanan (2009). "Support vector machines for spatiotemporal tornado prediction". In: *International Journal of General Systems* 38.7, pp. 759–776.

Ahijevych, David et al. (2009). "Application of spatial verification methods to idealized and NWP-gridded precipitation forecasts". In: *Weather and Forecasting* 24.6, pp. 1485–1497.

Ahijevych, David et al. (2016). "Probabilistic forecasts of mesoscale convective system initiation using the random forest data mining technique". In: *Weather and Forecasting* 31.2, pp. 581–599.

Anagnostou, Emmanouil N (2004). "A convective/stratiform precipitation classification algorithm for volume scanning weather radar observations". In: *Meteorological Applications* 11.4, pp. 291–300.

*AWS Machine Learning*. https://aws.amazon.com/machine-learning/. Accessed: 2018-05-01.

Aznarte, José L and Nils Siebert (2017). "Dynamic line rating using numerical weather predictions and machine learning: a case study". In: *IEEE Transactions on Power Delivery* 32.1, pp. 335–343.

Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla (2017). "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12, pp. 2481–2495.

Bai, Shaojie, J Zico Kolter, and Vladlen Koltun (2018). "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling". In: *arXiv preprint arXiv:1803.01271*.

Baldi, Pierre, Peter Sadowski, and Daniel Whiteson (2014). "Searching for exotic particles in high-energy physics with deep learning". In: *Nature communications* 5, p. 4308.

Baldwin M E, Kain J S and S Lakshmivarahan (2005). "Development of an automated classification procedure for rainfall systems". In: *Monthly weather review* 133.4, pp. 844–862.

Bankert, Richard L (1994). "Cloud classification of AVHRR imagery in maritime regions using a probabilistic neural network". In: *Journal of Applied Meteorology* 33.8, pp. 909–918.

Bengio, Yoshua, Aaron Courville, and Pascal Vincent (2013). "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8, pp. 1798–1828.

Berner, Judith et al. (2017). "Stochastic parameterization: Toward a new view of weather and climate models". In: *Bulletin of the American Meteorological Society* 98.3, pp. 565–588.

Billet, John et al. (1997). "Use of regression techniques to predict hail size and the probability of large hail". In: *Weather and Forecasting* 12.1, pp. 154–164.

Bowler, Neill E, Clive E Pierce, and Alan W Seed (2006). "STEPS: A probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with

downscaled NWP". In: *Quarterly Journal of the Royal Meteorological Society* 132.620, pp. 2127–2155.
Breiman, Leo (1996). "Bagging predictors". In: *Machine learning* 24.2, pp. 123–140.
— (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32.
Breiman, Leo et al. (1984). "Classification and regression trees". In:
Buehner, Mark et al. (2010). "Intercomparison of variational data assimilation and the ensemble Kalman filter for global deterministic NWP. Part II: One-month experiments with real observations". In: *Monthly Weather Review* 138.5, pp. 1567–1586.
Buizza, Roberto et al. (2005). "A comparison of the ECMWF, MSC, and NCEP global ensemble prediction systems". In: *Monthly Weather Review* 133.5, pp. 1076–1097.
Burgers, Gerrit, Peter Jan van Leeuwen, and Geir Evensen (1998). "Analysis scheme in the ensemble Kalman filter". In: *Monthly weather review* 126.6, pp. 1719–1724.
Camargo S J, Robertson A W Gaffney S J Smyth P and M Ghil (2007). "Cluster analysis of typhoon tracks. Part I: General properties". In: *Journal of Climate* 20.14, pp. 3635–3653.
Chapelle, Olivier, Bernhard Scholkopf, and Alexander Zien (2009). "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]". In: *IEEE Transactions on Neural Networks* 20.3, pp. 542–542.
Charney, Jules G, Ragnar Fjörtoft, and J von Neumann (1950). "Numerical integration of the barotropic vorticity equation". In: *Tellus* 2.4, pp. 237–254.
Chisholm, Donald A et al. (1968). "The diagnosis of upper-level humidity". In: *Journal of Applied Meteorology* 7.4, pp. 613–619.
*Cloud AI*. https://cloud.google.com/products/machine-learning/. Accessed: 2018-05-01.
*Copernicus Europe's eyes on Earth*. http://www.copernicus.eu/. Accessed: 2018-05-01.
Courtier, PHILIPPE, J-N Thépaut, and Anthony Hollingsworth (1994). "A strategy for operational implementation of 4D-Var, using an incremental approach". In: *Quarterly Journal of the Royal Meteorological Society* 120.519, pp. 1367–1387.
Dagum, Leonardo and Ramesh Menon (1998). "OpenMP: an industry standard API for shared-memory programming". In: *IEEE computational science and engineering* 5.1, pp. 46–55.
Dalal, Navneet and Bill Triggs (2005). "Histograms of oriented gradients for human detection". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, pp. 886–893.
Day, Neil E (1969). "Estimating the components of a mixture of normal distributions". In: *Biometrika* 56.3, pp. 463–474.
Dee, Dick P et al. (2011). "The ERA-Interim reanalysis: Configuration and performance of the data assimilation system". In: *Quarterly Journal of the royal meteorological society* 137.656, pp. 553–597.
Delage, Yves (1997). "Parameterising sub-grid scale vertical transport in atmospheric models under statically stable conditions". In: *Boundary-Layer Meteorology* 82.1, pp. 23–48.
DelSole, Timothy and Arindam Banerjee (2017). "Statistical seasonal prediction based on regularized regression". In: *Journal of Climate* 30.4, pp. 1345–1361.
DelSole, Timothy, Liwei Jia, and Michael K Tippett (2013). "Scale-selective ridge regression for multimodel forecasting". In: *Journal of Climate* 26.20, pp. 7957–7965.
Deng, Jia et al. (2009). "Imagenet: A large-scale hierarchical image database". In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, pp. 248–255.

Dietterich, Thomas G (2000). "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization". In: *Machine learning* 40.2, pp. 139–157.

*Earth on AWS*. https://aws.amazon.com/earth/. Accessed: 2018-05-01.

Eldali, Fathalla A et al. (2016). "Employing ARIMA models to improve wind power forecasts: A case study in ERCOT". In: *North American Power Symposium (NAPS), 2016*. IEEE, pp. 1–6.

Evans, Ben et al. (2015). "The NCI high performance computing and high performance data platform to support the analysis of petascale environmental data collections". In: *International Symposium on Environmental Software Systems*. Springer, pp. 569–577.

Fan, Wei and Albert Bifet (2013). "Mining big data: current status, and forecast to the future". In: *ACM sIGKDD Explorations Newsletter* 14.2, pp. 1–5.

Foley, Aoife M et al. (2012). "Current methods and advances in forecasting of wind power generation". In: *Renewable Energy* 37.1, pp. 1–8.

Folk, Mike et al. (2011). "An overview of the HDF5 technology suite and its applications". In: *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*. ACM, pp. 36–47.

Forgy, Edward W (1965). "Cluster analysis of multivariate data: efficiency versus interpretability of classifications". In: *biometrics* 21, pp. 768–769.

Freund, Yoav and Llew Mason (1999). "The alternating decision tree learning algorithm". In: *icml*. Vol. 99, pp. 124–133.

Friederichs, P and A Hense (2007). "Statistical downscaling of extreme precipitation events using censored quantile regression". In: *Monthly weather review* 135.6, pp. 2365–2378.

Gentine, Pierre et al. (2018). "Could machine learning break the convection parameterization deadlock?" In: *Geophysical Research Letters*.

Giebel, Gregor et al. (2011). "The state-of-the-art in short-term prediction of wind power: A literature overview". In: *ANEMOS. plus*.

Glahn, Harry R and Dale A Lowry (1972). "The use of model output statistics (MOS) in objective weather forecasting". In: *Journal of applied meteorology* 11.8, pp. 1203–1211.

Godfrey, Christopher M and David J Stensrud (2010). "An empirical latent heat flux parameterization for the Noah land surface model". In: *Journal of Applied Meteorology and Climatology* 49.8, pp. 1696–1713.

Goldston, David (2008). "Big data: Data wrangling". In: *Nature News* 455.7209, pp. 15–15.

Goodfellow, Ian et al. (2014). "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Gorelick, Noel et al. (2017). "Google Earth Engine: Planetary-scale geospatial analysis for everyone". In: *Remote Sensing of Environment* 202, pp. 18–27.

Greenspan, Hayit, Bram van Ginneken, and Ronald M Summers (2016). "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique". In: *IEEE Transactions on Medical Imaging* 35.5, pp. 1153–1159.

Gropp, William D et al. (1999). *Using MPI: portable parallel programming with the message-passing interface*. Vol. 1. MIT press.

Gupta, Abhinav et al. (2010). "Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces". In: *Advances in neural information processing systems*, pp. 1288–1296.

Hamill, Thomas M and Jeffrey S Whitaker (2006). "Probabilistic quantitative precipitation forecasts based on reforecast analogs: Theory and application". In: *Monthly Weather Review* 134.11, pp. 3209–3229.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). "Unsupervised learning". In: *The elements of statistical learning*. Springer, pp. 485–585.

Hawkins, Douglas M (2004). "The problem of overfitting". In: *Journal of chemical information and computer sciences* 44.1, pp. 1–12.

Hearst, Marti A. et al. (1998). "Support vector machines". In: *IEEE Intelligent Systems and their applications* 13.4, pp. 18–28.

Herman, Gregory R and Russ S Schumacher (2018). "Money Doesn't Grow on Trees, but Forecasts Do: Forecasting Extreme Precipitation with Random Forests". In: *Monthly Weather Review* 146.5, pp. 1571–1600.

Hernández-González, Jerónimo, Inaki Inza, and Jose A Lozano (2016). "Weak supervision and other non-standard classification problems: a taxonomy". In: *Pattern Recognition Letters* 69, pp. 49–55.

Hinton, Geoffrey E (2009). "Deep belief networks". In: *Scholarpedia* 4.5, p. 5947.

Hinton, Geoffrey E and Ruslan R Salakhutdinov (2006). "Reducing the dimensionality of data with neural networks". In: *science* 313.5786, pp. 504–507.

Hodge, Bri-Mathias et al. (2011). "Improved wind power forecasting with ARIMA models". In: *Computer Aided Chemical Engineering*. Vol. 29. Elsevier, pp. 1789–1793.

Hodyss, Daniel and William F Campbell (2013). "Square root and perturbed observation ensemble generation techniques in Kalman and quadratic ensemble filtering algorithms". In: *Monthly Weather Review* 141.7, pp. 2561–2573.

Hofer, Marlis et al. (2017). "Evaluating predictor strategies for regression-based downscaling with a focus on glacierized mountain environments". In: *Journal of Applied Meteorology and Climatology* 56.6, pp. 1707–1729.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1989). "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5, pp. 359–366.

Isola, Phillip et al. (2017). "Image-to-image translation with conditional adversarial networks". In: *arXiv preprint*.

Johnson, N C (2013). "How many ENSO flavors can we distinguish?" In: *Journal of Climate* 26.13, pp. 4816–4827.

Jones, N (2017). "How machine learning could help to improve climate forecasts." In: *Nature News*.

Kalkstein, Laurence S, Guanri Tan, and Jon A Skindlov (1987). "An evaluation of three clustering procedures for use in synoptic climatological classification". In: *Journal of climate and applied meteorology* 26.6, pp. 717–730.

Kalnay, Eugenia (2003). *Atmospheric modeling, data assimilation and predictability*. Cambridge university press.

Karpathy, Andrej and Li Fei-Fei (2015). "Deep visual-semantic alignments for generating image descriptions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3128–3137.

Key, J, JA Maslanik, and AJ Schweiger (1989). "Classification of merged AVHRR and SMMR Arctic data with neural networks". In:

Kimura, Ryuji (2002). "Numerical weather prediction". In: *Journal of Wind Engineering and Industrial Aerodynamics* 90.12-15, pp. 1403–1414.

Kingma, D P and M Welling (2013). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114*.

Kitzmiller, David H, Wayne E McGovern, and Robert F Saffle (1995). "The WSR-88D severe weather potential algorithm". In: *Weather and forecasting* 10.1, pp. 141–159.

Klein, William H, Billy M Lewis, and Isadore Enger (1959). "Objective prediction of five-day mean temperatures during winter". In: *Journal of Meteorology* 16.6, pp. 672–682.

Krasin, Ivan et al. (2017). "OpenImages: A public dataset for large-scale multi-label and multi-class image classification." In: *Dataset available from https://storage.googleapis.com/openimages/we*

Kravtsov, S, D Kondrashov, and M Ghil (2005). "Multilevel regression modeling of nonlinear processes: Derivation and applications to climatic variability". In: *Journal of Climate* 18.21, pp. 4404–4424.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105.

Larraondo, Pablo Rozas, Iñaki Inza, and Jose A Lozano (2018). "A system for airport weather forecasting based on circular regression trees". In: *Environmental Modelling & Software* 100, pp. 24–32.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *nature* 521.7553, p. 436.

Liu, Yunjie et al. (2016). "Application of deep convolutional neural networks for detecting extreme weather in climate datasets". In: *arXiv preprint arXiv:1605.01156*.

Liu Z, Zhou P and Y Zhang (2015). "A Probabilistic Wavelet–Support Vector Regression Model for Streamflow Forecasting with Rainfall and Climate Information Input". In: *Journal of Hydrometeorology* 16.5, pp. 2209–2229.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Lorenz, EN (1982). "Atmospheric predictability experiments with a large numerical model". In: *Tellus* 34.6, pp. 505–513.

Lowe, David G (2004). "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2, pp. 91–110.

Lund, Ulric J (2002). "Tree-based regression for a circular response". In: *Communications in Statistics-Theory and Methods* 31.9, pp. 1549–1560.

Mallet, Vivien, Gilles Stoltz, and Boris Mauricette (2009). "Ozone ensemble forecast with machine learning algorithms". In: *Journal of Geophysical Research: Atmospheres* 114.D5.

Malone, Thomas F (1955). "Application of statistical methods in weather prediction". In: *Proceedings of the National Academy of Sciences* 41.11, pp. 806–815.

Mao, Xiaojiao, Chunhua Shen, and Yu-Bin Yang (2016). "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections". In: *Advances in Neural Information Processing Systems*, pp. 2802–2810.

Marzban, Caren, Stephen Leyton, and Brad Colman (2007). "Ceiling and visibility forecasts via neural networks". In: *Weather and forecasting* 22.3, pp. 466–479.

Marzban, Caren, Scott Sandgathe, and Eugenia Kalnay (2006). "MOS, perfect prog, and reanalysis". In: *Monthly weather review* 134.2, pp. 657–663.

McGovern, Amy et al. (2017). "Using artificial intelligence to improve real-time decision-making for high-impact weather". In: *Bulletin of the American Meteorological Society* 98.10, pp. 2073–2090.

Mecikalski, John R et al. (2015). "Probabilistic 0–1-h convective initiation nowcasts that combine geostationary satellite observations and numerical weather prediction model data". In: *Journal of Applied Meteorology and Climatology* 54.5, pp. 1039–1059.

Messner, Jakob W et al. (2014). "Heteroscedastic extended logistic regression for postprocessing of ensemble guidance". In: *Monthly Weather Review* 142.1, pp. 448–456.

Mika, Sebastian et al. (1999). "Fisher discriminant analysis with kernels". In: *Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop.* Ieee, pp. 41–48.

Milton, SF and CA Wilson (1996). "The impact of parameterized subgrid-scale orographic forcing on systematic errors in a global NWP model". In: *Monthly weather review* 124.9, pp. 2023–2045.

Mo, Kingtse and Michael Ghil (1988). "Cluster analysis of multiple planetary flow regimes". In: *Journal of Geophysical Research: Atmospheres* 93.D9, pp. 10927–10952.

Moghim, Sanaz and Rafael L Bras (2017). "Bias correction of climate modeled temperature and precipitation using artificial neural networks". In: *Journal of Hydrometeorology* 18.7, pp. 1867–1884.

Molteni, Franco, Stefano Tibaldi, and TN Palmer (1990). "Regimes in the wintertime circulation over northern extratropics. I: Observational evidence". In: *Quarterly Journal of the Royal Meteorological Society* 116.491, pp. 31–67.

Molteni, Franco et al. (1996). "The ECMWF ensemble prediction system: Methodology and validation". In: *Quarterly journal of the royal meteorological society* 122.529, pp. 73–119.

Neter, John, William Wasserman, and Michael H Kutner (1989). "Applied linear regression models". In:

O'Gorman, Paul A and John G Dwyer (2018). "Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change and extreme events". In: *arXiv preprint arXiv:1806.11037*.

Palmer, Tim and Renate Hagedorn (2006). *Predictability of weather and climate*. Cambridge University Press.

Pedregosa, Fabian et al. (2011). "Scikit-learn: Machine learning in Python". In: *Journal of machine learning research* 12.Oct, pp. 2825–2830.

Peng, Jian et al. (2017). "A review of spatial downscaling of satellite remotely sensed soil moisture". In: *Reviews of Geophysics* 55.2, pp. 341–366.

Quinlan, J Ross (1993). *C4. 5: programs for machine learning*. Elsevier.

Raible, Christoph C et al. (1999). "Statistical single-station short-term forecasting of temperature and probability of precipitation: Area interpolation and NWP combination". In: *Weather and forecasting* 14.2, pp. 203–214.

Ramos, M C (2001). "Divisive and hierarchical clustering techniques to analyse variability of rainfall distribution patterns in a Mediterranean region". In: *Atmospheric Research* 57.2, pp. 123–138.

Rasp, Stephan, Michael S Pritchard, and Pierre Gentine (2018). "Deep learning to represent sub-grid processes in climate models". In: *arXiv preprint arXiv:1806.04731*.

Renzullo, Luigi J, Edwin H Sutanudjaja, and Marc FP Bierkens (2016). "Improved large-scale hydrological modelling through the assimilation of streamflow and downscaled satellite soil moisture observations". In: *Hydrology and Earth System Sciences* 20.7, p. 3059.

Rew, R, E Hartnett, J Caron, et al. (2006). "NetCDF-4: Software implementing an enhanced data model for the geosciences". In: *22nd International Conference on*

*Interactive Information Processing Systems for Meteorology, Oceanograph, and Hydrology*.

Richardson, LF (1922). "Weather Prediction by Numerical Process". In:

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.

Rozas-Larraondo, Pablo, Iñaki Inza, and Jose A Lozano (2014). "A method for wind speed forecasting in airports based on nonparametric regression". In: *Weather and Forecasting* 29.6, pp. 1332–1342.

Russell, Stuart J and Peter Norvig (1995). *Artificial intelligence: a modern approach*. Prentice-Hall,

Samuel, Arthur L (1959). "Some studies in machine learning using the game of checkers". In: *IBM Journal of research and development* 3.3, pp. 210–229.

Shallue, Christopher J and Andrew Vanderburg (2018). "Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90". In: *The Astronomical Journal* 155.2, p. 94.

Stram, Daniel O and William WS Wei (1986). "Temporal aggregation in the ARIMA process". In: *Journal of Time Series Analysis* 7.4, pp. 279–292.

Szturc, Jan, K Osrodka, and Anna Jurczyk (2007). "Parameterisation of radar precipitation quality index scheme on raingauge data". In: *Proc. 33rd International Conference on Radar Meteorology, Cairns*.

Taskar, Ben et al. (2005). "Learning structured prediction models: A large margin approach". In: *Proceedings of the 22nd international conference on Machine learning*. ACM, pp. 896–903.

Taylor, Karl E, Ronald J Stouffer, and Gerald A Meehl (2012). "An overview of CMIP5 and the experiment design". In: *Bulletin of the American Meteorological Society* 93.4, pp. 485–498.

Tebaldi, Claudia and Reto Knutti (2007). "The use of the multi-model ensemble in probabilistic climate projections". In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 365.1857, pp. 2053–2075.

Tracton, M Steven and Eugenia Kalnay (1993). "Operational ensemble prediction at the National Meteorological Center: Practical aspects". In: *Weather and Forecasting* 8.3, pp. 379–398.

Tran, Du and Junsong Yuan (2012). "Max-margin structured output regression for spatio-temporal action localization". In: *Advances in neural information processing systems*, pp. 350–358.

Turing, Alan M (1950). "Computing machinery and intelligence". In: *Mind* 59.236, p. 433.

Vasquez, Tim (2009). *Weather Forecasting Red Book: Forecasting Techniques for Meteorology*. Weather Graphics Technologies.

Vislocky, Robert L and George S Young (1989). "The use of perfect prog forecasts to improve model output statistics forecasts of precipitation probability". In: *Weather and forecasting* 4.2, pp. 202–209.

Wald, Lucien (1999). "Some terms of reference in data fusion". In: *IEEE Transactions on geoscience and remote sensing* 37.3, pp. 1190–1193.

Wei, Chih-Chiang (2012). "Wavelet support vector machines for forecasting precipitation in tropical cyclones: comparisons with GSVM, regression, and MM5". In: *Weather and forecasting* 27.2, pp. 438–450.

Wilks, Daniel S (2002). "Smoothing forecast ensembles with fitted probability distributions". In: *Quarterly Journal of the Royal Meteorological Society* 128.586, pp. 2821–2836.

Wilks, DS (1995). "Forecast verification". In: *Statistical methods in the atmospheric sciences*.

Williams, John K et al. (2008). "Remote detection and diagnosis of thunderstorm turbulence". In: *Remote sensing applications for aviation weather hazard detection and decision support*. Vol. 7088. International Society for Optics and Photonics, p. 708804.

Williams, Ronald J and David Zipser (1989). "A learning algorithm for continually running fully recurrent neural networks". In: *Neural computation* 1.2, pp. 270–280.

Xingjian, SHI et al. (2015). "Convolutional LSTM network: A machine learning approach for precipitation nowcasting". In: *Advances in neural information processing systems*, pp. 802–810.

Zhang, G Peter (2003). "Time series forecasting using a hybrid ARIMA and neural network model". In: *Neurocomputing* 50, pp. 159–175.

Zhou, Bolei et al. (2016). "Learning deep features for discriminative localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929.