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*

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.

Keywords: Circular Variables, Weather Forecasting, Meteorology, Regression Trees, Machine Learning

Software Availability

Name of software: AeroCirTree
Developer: Pablo Rozas Larraondo

*Corresponding author: Tel.: +61 (02) 6125 3211;
Email address: pablo.larraondo@anu.edu.au (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 [1]. 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 [2], interpolation [3] or machine learning models [4, 5].

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 [6]. 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.

There are a number of tools that facilitate the process of generating airport weather forecasts [7, 8], 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 suboptimal results in most cases. The model presented in this article builds upon the concept of circular regression trees introduced by Lund [9]. 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) [10] 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.

Figure 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.

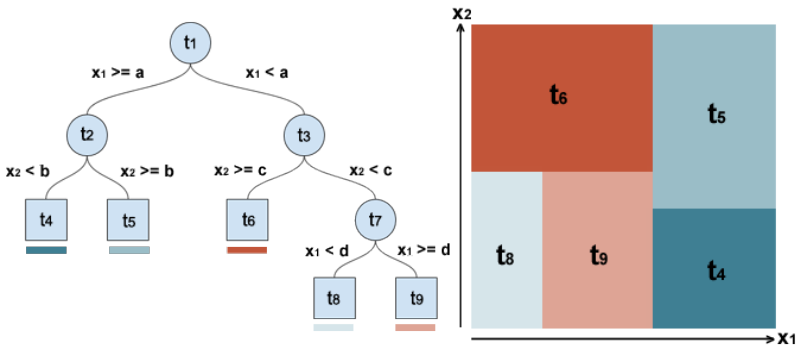

Figure 1: Example of a classic linear regression tree and a representation of how the space is divided.

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 2π, 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 [9].

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 [9] proposes a methodology that allows circular variables to be incorporated into regression trees. Figure 2 contains a similar representation to the previous example, but considering one circular variable α and a linear one x_1. On the right side, there is a chart representing how the space is partitioned using polar coordinates.

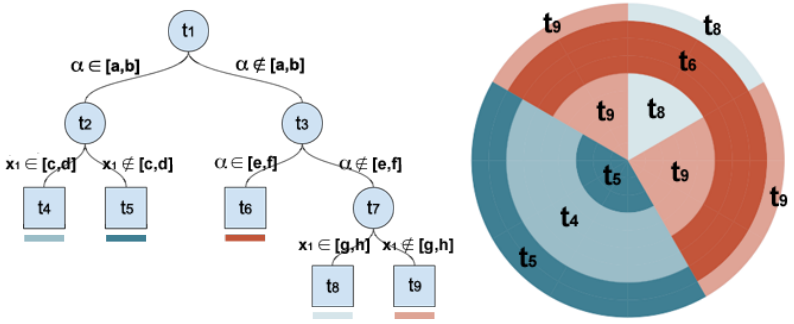

Figure 2: Example of Lund's original proposal of circular regression tree and a representation of how the space is divided.

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. Figure 3 shows how the space is partitioned using the proposed methodology.

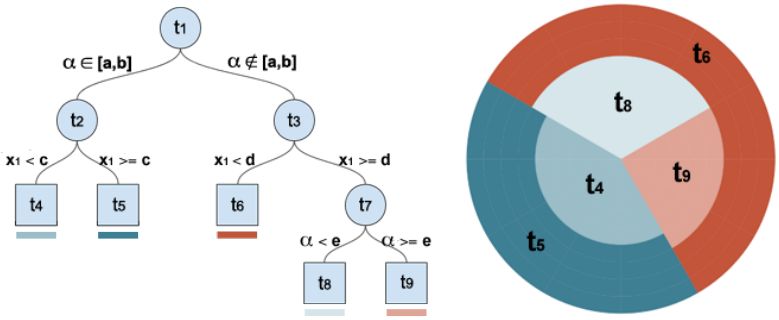

Figure 3: Example of the proposed circular regression tree and a representation of how the space is divided.

Visually comparing Figure 2 and Figure 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 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) [11], which is run operationally by the National Oceanic and Atmospheric Administration (NOAA). The second uses Meteorological Aerodrome Reports (METARs) [6], 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 hours. 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) [12] 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.

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 to 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.

Figure 4 and Figure 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

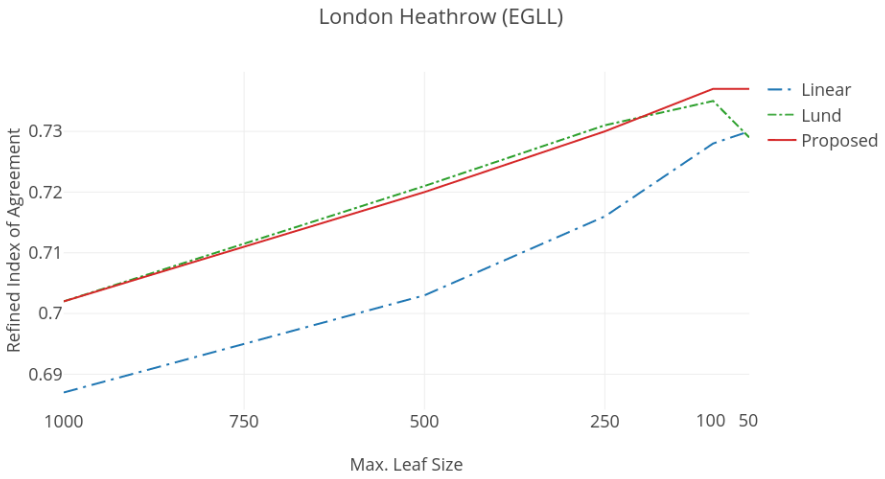

Figure 4: RIA values for the airport of London Heathrow (EGLL), comparing the accuracy of the output for different maximum leaf sizes.

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 [13] 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 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

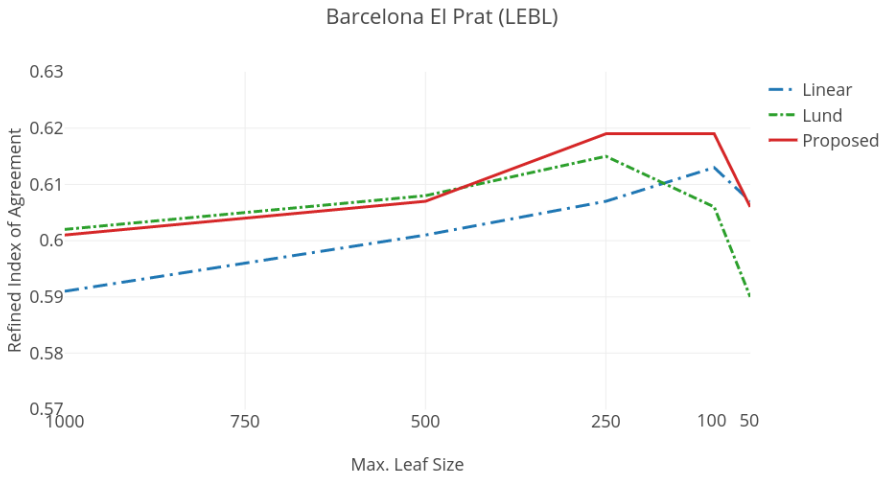

Figure 5: RIA values for the airport of Barcelona El Prat (LEBL) comparing the accuracy of the output for different maximum leaf sizes.

critical difference.

Figure 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) [14].

As can be seen in the CD diagrams in Figure 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

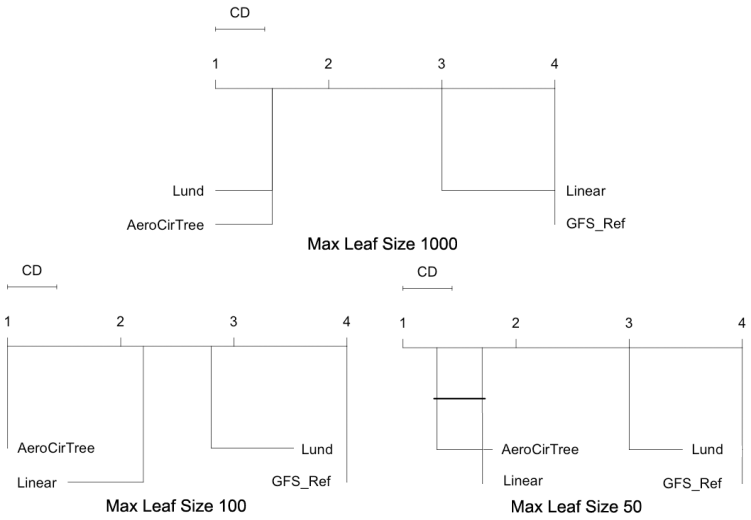

Figure 6: Critical Differences comparing the three methodologies for shallow and deep trees. $\alpha = 0.05$

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 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 hours 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 previouly 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 [10, 15] or random forests [16] and ensembles [17] 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

[1] A. C. Carvalho, A. Carvalho, H. Martins, C. Marques, A. Rocha, C. Borrego, D. X. Viegas, A. I. Miranda, Fire weather risk assessment under climate change using a dynamical downscaling approach, Environmental Modelling & Software 26 (2011) 1123-1133.

[2] M. Bannayan, G. Hoogenboom, Weather analogue: A tool for real-time prediction of daily weather data realizations based on a modified k-nearest neighbor approach, Environmental Modelling & Software 23 (2008) 703-713.

[3] C. C. F. Plouffe, C. Robertson, L. Chandrapala, Comparing interpolation techniques for monthly rainfall mapping using multiple evaluation criteria and auxiliary data sources: A case study of sri lanka, Environmental Modelling & Software 67 (2015) 57-71.

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

[5] T. Salameh, P. Drobinski, M. Vrac, P. Naveau, Statistical downscaling of near-surface wind over complex terrain in southern france, Meteorology and Atmospheric Physics 103 (2009) 253-265.

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

[7] J. E. Ghirardelli, B. Glahn, The meteorological development laboratorys aviation weather prediction system, Weather and Forecasting 25 (2010) 1027-1051.

[8] A. J. M. Jacobs, N. Maat, Numerical guidance methods for decision support in aviation meteorological forecasting, Weather and Forecasting 20 (2005) 82-100.

[9] U. J. Lund, Tree-based regression for a circular response., Communications in Statistics - Theory and Methods 31 (2002) 1549-1560.

[10] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and regression trees, Wadsworth Books, 1984.

[11] K. Campana, P. Caplan, Technical procedure bulletin for t382 global forecast system., 2005.

[12] C. J. Willmott, S. M. Robeson, K. Matsuura, A refined index of model performance, International Journal of Climatology 32 (2012) 2088–2094.

[13] J. Demsar, Statistical comparisons of classifiers over multiple data sets., Journal of Machine Learning Research 7 (2006) 1–30.

[14] B. Calvo, G. Santafe, scmamp: Statistical comparison of multiple algorithms in multiple problems., The R Journal (2016).

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

[16] L. Breiman, Random forests., Machine Learning 45.1 (2001) 5–32.

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

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