Virtual Wind Sensors: Improving Wind Forecasting Using Big Data Analytics

K Gray, Dr T van Zyl, Prof. T Celik

September 20, 2016

Abstract

Wind sensors provide very accurate measurements, however it is not feasible to have a network of wind sensors large enough to provide these accurate readings everywhere. A "virtual" wind sensor uses existing weather forecasts, as well as historical weather station data to predict what readings a regular wind sensor would provide. This study attempts to develop a method using Big Data Analytics to predict wind readings for use in "virtual" wind sensors. The study uses Random Forests and linear regression to estimate wind direction and magnitude using various transformations of a Digital Elevation Model, as well as data from the European Centre for Medium-Range Weather Forecasts. The model is evaluated based on its accuracy when compared to existing high resolution weather station data, to show a slight improvement in the estimation of wind direction and magnitude over the forecast data.

Declaration

I, Kevin Alan Gray, hereby declare the content of this dissertation to be my own. This dissertation is submitted for the degree of Master of Science at the University of the Witwatersrand. This work has not been submitted to any other university, or for any other degree.

. . . .

Signature Date

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Background

Accurate weather data is important in many industries and is of particular interest for this study. Wind sensors do provide this data, however wind sensor data is not always available in the location for which the data is required. Building a new wind sensor at a location whenever wind data is required is one solution to the problem of the availability of wind data; however, this solution comes at a high cost, in terms of both time and money. In some cases it may not be possible to build a wind sensor at all because of the terrain or existing infrastructure. This study looks into an alternative solution using existing data to predict wind data that can be used as part of a virtual weather station system.

In order to develop a system that can predict wind data accurately, the available data needs to be processed. There are two elements of the data that need to be considered, namely the amount of data currently available that needs to be processed (the volume of the data), and the rate at which new data that needs to be processed becomes available (the velocity of the data).

When dealing with wind data such as this, there are various approaches that can be considered: a simulation of the wind can be used, equations can be formulated to better define the relationship between the input data and the wind data, and machine learning that can be trained to recognize patterns in the data can be used. Simulations, although accurate, take a long time to calculate wind data for many data points. Physical equations would be the ideal approach, as equations can be calculated very rapidly. However, formulating these equations requires extensive research and the formulated equations may be unique for any particular location. Machine learning algorithms can produce results very rapidly once trained, despite the time required for training.

A machine learning system trained on wind data may be able to determine patterns and correlations between the inputs and wind direction and magnitude, allowing a trained algorithm to make more accurate wind predictions. After the initial training phase, machine learning systems can still be trained further with the data used for making estimations when the weather station wind measurements are available. This may improve future estimations and allow the machine learning system to adapt to potential changes in wind patterns. These more accurate wind predictions can be used as a part of a "virtual" weather station system to provide a much greater coverage of data, without the need to build weather stations.

1.2 Problem Statement

Physical sensor networks are limited by their location. Increasing the number of physical sensors requires sensors to be installed, which leads to unnecessary cost increases. These physical sensors are further limited by the locations in which they can be installed. "Virtual" sensors can replace physical sensors, increasing the availability of sensor data without the requirement of the physical sensors. Before "virtual" sensors can be used however, they need to be able to produce data with an accuracy that has an acceptable margin of error. Machine learning, along with existing sensor data, can be used to develop these "virtual" sensors.

A machine learning system needs to be trained before it can be accurately used as an estimator. Training requires a dataset that is significantly large, to allow the machine learning system to fit the data. The training dataset also needs to be preprocessed to provide various features that the machine learning system can be trained on.

1.3 Research Objectives

This study will determine if wind direction and speed measurements from weather stations can be accurately estimated using machine learning methods and wind forecast data from the European Centre for Medium-Range Weather Forecasts (ECMWF). In addition, the study will evaluate if there is any improvement in the wind estimation when features are added that are extracted from a Digital Elevation Model (DEM). Various topographical transforms and image processing techniques will be applied to the DEM to determine if there is any significant improvement in the accuracy of the estimation when the created features are used. The impact of increasing or decreasing the area that is extracted from the DEM will also be considered. Furthermore, the study will evaluate if scaling the data before training the machine learning system has any impact upon its estimation accuracy. The study must moreover consider which combination of the above methods provides the most accurate wind estimation. Additionally, the capability of the machine learning system to generalize the wind estimations needs to be assessed.

1.4 Thesis Statement

Can machine learning methods be used with low resolution weather forecast data and digital elevation models to accurately estimate high resolution weather station data?

1.5 Delineations and Limitations

1.5.1 Delineations

The potential scope of the study can include many different learning methods and consider many different possible features. Since the aim of the study is not to determine what the best learning method for wind prediction is, but rather to determine if a general learner can be used to predict wind, the learning methods will be limited to Random Forest Regression (for learning non-linear structures) and linear regression (for learning linear structures). For a similar reason, the chosen features with which to perform the tests are also limited to the European Centre for Medium-Range Weather Forecasts (ECMWF) data and a small number of transforms of the DEM, along with the weather station data for validation. The datasets are limited to the Southern African region for data collection purposes, which may also provide limitations for the study as discussed in section 1.5.2.

1.5.2 Limitations

As the entire dataset is limited to the region of Southern Africa, the results from the learning system may be biased towards the Southern African region; however, the system can be trained further using other datasets to overcome this bias. Since the learning system is validated only at the location of weather stations within the data region, the system may be less accurate at predicting wind directions or wind speeds at a different location within the region. This is not the focus of the study though, and the system could be extended in the future to account for other locations. The time frame that is used for the learner is limited to nine months. This may lead to the learning system not performing for the months outside of this range. This limitation can be reduced by training the learner with a dataset with a larger time frame.

1.6 Assumptions

Some assumptions in the study include:

- Data relevance
- DEM accuracy
- Missing values
- Weather data accuracy
- Time difference between datasets

It can be assumed that the relevance of the data being passed to the learning system does not need to be verified before it is passed to the system for learning. Any irrelevant data that will be discovered by the learner has less impact on the results compared to relevant data. The accuracy of the DEM cannot be guaranteed, as discussed in section A.3. There is no feasible way to correct any errors in the DEM, although the errors can also be assumed to be negligible. Another issue with the DEM is that of missing values. These values are usually located over oceans or other large masses of water, and thus can be assumed to be at sea level. The accuracy of the ECMWF and weather station data cannot be guaranteed nor can it be corrected. However, since both datasets come from reputable sources, the data can be assumed to be accurate. In order to estimate the weather station data based on the ECMWF data, the ECMWF forecast needs to be generated for the same time the weather station measurements are taken. Since the timing of some of the weather station readings is not consistent with the forecast times, it is assumed that time differences within a certain window can be considered close enough to the forecast time to be used. For the purposes of this study, any weather station data that has a time difference of less than an hour with the forecast times is considered.

1.7 Significance

Wind data is commonly generated over large areas, using many methods, however very accurate, localised wind estimations are not common. Generating accurate localised wind data can be used to greatly improve wind forecast data, as well as to allow for systems to be implemented that predict the movements of events that are dependent upon wind, such as the spread of fires or air pollution [1].

1.8 Chapter Overview

The first chapter in this dissertation explained the problem that is studied, the background to the problem, the limitations imposed on the problem, and the assumptions made in the study. The second chapter introduces similar research that has been done and research that is related to wind prediction, explaining how the existing research is, or is not, relevant to the study. The third chapter explains what data is required for the study, how this data is processed in preparation for the study, what learning methods are used, how the various features are extracted from the data, and finally how the results are analyzed and compared. The fourth chapter presents the results, explaining the process of obtaining and interpreting these results. The fifth chapter briefly answers the thesis statement, motivating this answer with a summary of the analysis of the results, followed by an explanation of how the study can be extended, as well as any limitations that were encountered.

Chapter 2

Literature Survey

This chapter will briefly focus on what research has been done that relates to the data sources used in this study, namely Digital Elevation Models (DEM), and the European Centre for Medium-Range Weather Forecasts (ECMWF) data [2, 3]. Following this, previous research that has been done on predicting wind speed and direction will be discussed, with a focus on research that used similar variations of the data sources or existing sensors for their predictions. Research on similar applications that use alternative methods to predict the data are also examined.

This study will use the Shuttle Radar Topography Mission (SRTM) DEM as a data source for topographical data. Farr et al. [2] produced a technical document that details the SRTM mission, including mission design, the specifics of how the systems worked, maintenance of the orbit and the data processing that was required. The validation section of Farr et al. [2] is more significant, as it details the performance of the SRTM DEM and explains how areas of extreme errors are handled by returning a void value of -32768, usually caused by steep slopes facing away from the radar, or smooth areas such as smooth water or sand. Rodriguez et al. [4] explain, in more detail, the performance of the DEM with the error meeting the mission goal of a maximum height error of less than $16m$ with a confidence level of 90%. Rodriguez et al. [4] also explain that the errors are affected by the local topography, with larger errors being more prominent in areas with extreme topography.

The wind forecast data that will be used in this study is obtained from the ECMWF[3]. The ECMWF models the weather using historical data to constantly improve weather forecasts. A performance evaluation of the ECMWF model is published annually as a technical memorandum. The performance evaluation by Haiden et al. [5] is the most relevant to the data used in this study.

There are two elements involved in making basic wind estimations for “virtual” wind sensors, namely the direction and the speed of the wind, both of which may be affected differently by the ECMWF and the DEM. Very little research has been done concerning the effect of topographic or forecast data on wind direction estimation, however, a larger amount of research into wind speed estimation is available, although most research in this field is focused more towards wind power generation, as in Abo-Khalil and Lee [6] and Lei et al. [7], or uses existing historical physical sensor data to estimate future data, as in Mohandes et al. [8].

Two studies, Hu et al. [9] and Thompson and Beal [10], use satellite based imagery in an attempt to estimate wind direction and speed. The approach to estimating wind speed used by Hu et al. [9] establishes the relationship between the wave slope variance and the surface wind speed through statistical means, allowing accurate estimations of wind speed to be made using space-based lidar measurements. These lidar measurements are first calibrated using sea surface backscatter to reduce the uncertainty in wave slope variance. The relationship is then determined between the calibrated wave slope data and wind speed data with a resolution of 20km, which is then used to estimate wind speed well. The approach used by Hu et al. [9], however, is only applicable when estimating surface wind speeds for locations that are over the ocean due to the lidar measurements using sea surface backscatter and wave slope variance to calculate the wind speed. Using synthetic aperture radar, Thompson and Beal [10] attempt to develop a method to map wind fields, which estimates both wind direction and wind speed at a high resolution for large areas. Thompson and Beal [10] show that estimating wind speed using measurements from only a single azimuth angle is not possible, and use measurements made from three azimuth angles. These measurements are used with numerical model predictions to estimate both wind speed and wind direction at a higher resolution. However, like the lidar measurements used in Hu et al. [9], the scatterometery approach to estimating wind used by Thompson and Beal [10] is calculated using sea surface backscatter, and thus both studies are limited to making estimations regarding wind over the ocean only, whereas “virtual” wind sensors need to be able to make wind estimations over land as well.

Research into wind simulation methods can provide accurate estimations for both wind speed and direction. This has led to the development of software such as WindNinja [11], the "mass-consistent" model referred to by Forthofer [1], which uses computational fluid dynamics as well certain wind forecast data (ECMWF data or similar) and a DEM to accurately predict wind data over the specified DEM area. Using such software requires the simulation to be run, which may be slow depending on the extent of the area that the simulation's prediction covers. The time to complete large scale ($800km - 1400km$ scale) wind direction and wind speed estimations may restrict the usability of Wind Ninja for real-time wind data estimation since for small scale ($100m - 300m$ scale) simulations, the computation time can be between 8 and 45 seconds, or in the case of the commercial, more accurate WindWizard, 30 minutes to 2 hours [12].

Sensor data is invaluable in providing data that can be used to train systems to accurately estimate wind speed and direction, as can be seen in the research of Mohandes et al. [8] and Allen et al. [13]. Mohandes et al. [8] provide a method that can accurately predict surface wind speed over a small area. The study uses support vector machines and historical wind data from physical sensors to predict future wind speeds at the same location. The support vector machine is trained with a Gaussian kernel to estimate the wind speed of the following day, using the previous x days of wind speed data as inputs, where x ranges from one day to eleven days. A number of multi-layer perceptrons with varying numbers off hidden neurons are also trained using the same sets of input features and implementing the back-propagation algorithm optimized with the Levenberg-Marquardt method. The performance of the support vector machines are shown to be favourable when compared to the performance of the multi-layer perceptrons. Allen et al. [13] present a method that estimates both wind direction and the source characteristics of air pollutants using a genetic algorithm that searches the entire solution space. Searching the entire solution space in this way does have the disadvantage that there are additional calculations for dispersion at each iteration, increasing the computational cost substantially. A more computationally efficient method is used to reduce the impact of this disadvantage. This method evaluates each solution vector based on the predicted concentrations of the air pollutant and a number of receptor data values. These receptors are placed in a grid like arrangement, with each receptor separated by 2000m. Using a uniform crossover scheme that blends each parameter with a small mutation factor, the genetic algorithms were able to use this data to accurately estimate the surface wind direction when a receptor grid of

8×8 receptors were used. However smaller grids provided less accurate results, with a 2×2 receptor grid providing solutions that "were basically random". Although the methods used by Mohandes et al. [8] and Allen et al. [13] provide good ways to estimate wind direction, both rely on physical sensors which "virtual" wind sensors attempt to replace.

Antonić and Legović [14] use a different approach to estimate the direction in which an air pollution source lies, by considering the effect that a DEM may have. Their study focuses more on the effect of the topographic exposure of the area on wind flux to estimate the direction from which a pollutant may have originated. Four estimators are selected from a DEM, the relative terrain aspect, the the terrain exposure to the horizontal component of the wind flux, the horizon angle, and the terrain exposure toward the sloped wind flux. Each of these four estimators are examined for eight directions using the azimuthal step of 45°. Using local linear interpolation with the inverse distance weight method from adjacent pixels, the wind flux can be estimated. Their paper does suggests that topographic exposure is a simple estimator of wind flux and can be an important factor in studies where wind is an important independent variable. Although Antonić and Legović [14] focus on wind flux rather than wind speed, it does indicate that the DEM may be used to extract features that impact on wind estimation.

Robert et al. [15] approach the problem of predicting wind speed maps over complex terrain using general regression neural networks (GRNN). Two types of GRNNs are used, the Isotropic-GRNN and the Adaptive GRNN. The features that the GRNNs use to estimate wind are extracted from a DEM at differing scales, namely, the difference of gaussians, directional derivatives, and slopes. Using only the spatial properties as features (X and Y coordinates and the elevation at that point), both the Isotropic-GRNN and the Adaptive-GRNN performed poorly, however when adding the complete set of features, including those extracted from the DEM, the performance of the Adaptive GRNN showed significant improvement at estimating the monthly wind speed. Although the method proposed by Robert et al. [15] is a good estimator of ,"virtual" wind sensors would be required to estimate the wind more frequently, such as 6-hourly or 3-hourly. Robert et al. [15] do however suggest that there may be limits of spatial predictability at decreasing temporal scales of analysis and that the relationships between wind speed and topography tend to fade out.

2.1 Summary

The above-mentioned studies show that wind data can be accurately estimated using various methods, including methods that use topographic data from a DEM. Topographic data extracted from a DEM is relevant when estimating wind, as indicated in Antonić and Legović [14] and Robert et al. [15], which suggests that using features extracted from a DEM is a reasonable starting point for this study.

Chapter 3

Methodology

The datasets that are used for input and training data are explained in this chapter. The chapter also discusses how these datasets are processed in order to prepare the data in such a way that the learning system can utilise it. A short description and explanation of each learning method is presented, followed by a breakdown of all the features that are used in training the learning system. The methods that will be used to compare results are then explained and discussed.

All of the algorithms and scripts in this study are implemented using Python(x,y) version 2.7.5.0 [16]. Further open source projects are mentioned in the relevant sections, and all are developed under the GNU General Public License Version 3 [17].

3.1 Datasets

Three datasets are used to estimate wind direction with the learning system, namely:

- weather station data;
- European Centre for Medium-Range Weather Forecasts (ECMWF) data; and
- a Digital Elevation Model (DEM).

In order to learn and test the wind estimations, wind measurements are needed. For this study, the measurements chosen are from weather stations within the area of interest. These weather stations provide various measurements that could be used; however, for this study the only measurements of interest are:

- the weather station's ID and name;

- the weather station's latitude and longitude;
- the date and time of the measurement;
- the wind direction (in degrees); and
- the wind speed (in $m.s^{-1}$).

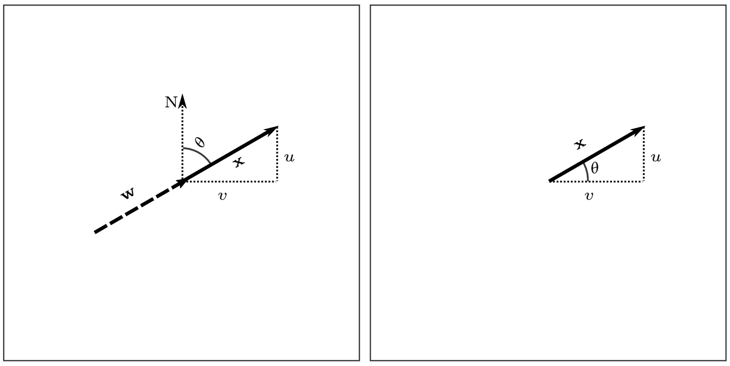
The learning system uses a weather forecast as the basis of the wind estimation. These forecasts come from the ECMWF data. The ECMWF service covers most forms of weather forecasting and thus provides many variables that can be used in the learning system. For the purposes of this study, a number of these forecast features were selected and used with the learning system, namely:

- the time of the forecast (in hours since 00:00:00 1900-01-01);
- the latitude and longitude of the location of the forecast;
- the forecast of the surface pressure (in Pascals);
- the forecast of the average temperature (in Kelvin);
- the forecast of the dewpoint temperature (in Kelvin); and
- the forecast of the u and v components of the wind (in $m.s^{-1}$).

Along with the forecast data, the machine learning system uses topographic data, centred at the location of the forecast data, which is obtained from the DEM, with each pixel or data point representing the elevation of that location in metres above sea level.

The representation of wind from the ECMWF and weather station datasets can be seen in figure 3.1. The weather station's dataset uses a meteorological convention to represent wind (direction in degrees and magnitude in m/s^{-1}), with 0° representing North (positive vertical axis), or rather wind coming from the South, with the angle increasing in a clockwise direction (towards the positive horizontal axis), as shown in figure 3.1a. The ECMWF dataset represents wind in a vector form (u and v components) and when calculating the angle of this vector, 0° would be along the positive horizontal axis (Eastern direction), with the angle increasing towards the positive vertical axis (counter-clockwise), as shown in figure 3.1b. For the purposes of this study, wind will be converted to the vector form.

More specific information about these datasets is discussed in Appendix A.



(a) Weather station dataset (b) ECMWF dataset

Figure 3.1: A representation of the components of the wind vector obtained from the different datasets.

3.2 Data Processing

3.2.1 Raw Data Processing

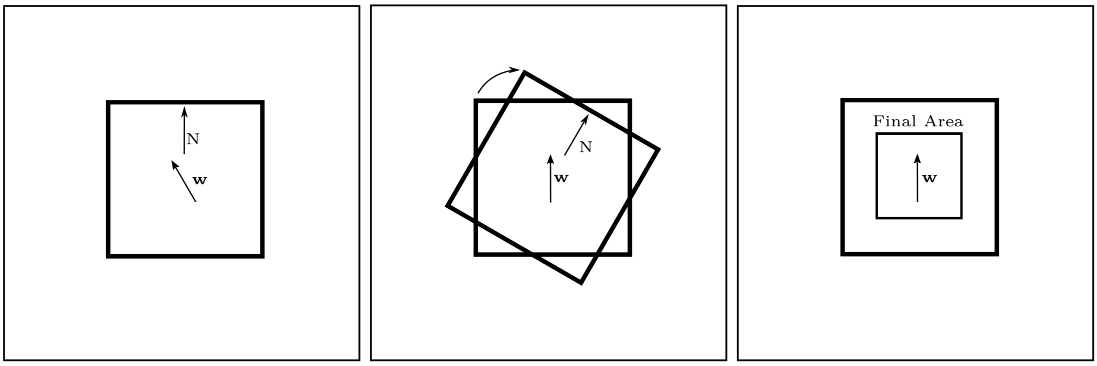
The initial data that is available needs to be processed into a format that the learning system can work with. In order to do this, a feature set needs to be determined. This feature set will also determine how each feature needs to be formatted for the learning system. The learning system used in this study takes as input a number of feature sets from the datasets. The reference variables that link all the data are the location, date and time of each wind measurement.

In order to extract the correct subset from the ECMWF data to create the features for a specific dataset, the ECMWF data needs to have a reference that links it with the weather station data. The references that are used for our study are the location of the weather station and the date and time of the weather station measurements. The ECMWF time data is represented in hours that have passed since 00:00:00 1900-01-01, which is a single floating point value that can easily be matched with a similar number from the weather station data. The location information is stored in two variables, the latitude and the longitude, in a grid-like structure. This makes it simple to use the latitude and longitude of the weather station to reference the correct ECMWF data. The dewpoint temperature, surface pressure and temperature values are extracted from the ECMWF dataset without adjusting the format of the data. However, the wind forecast is adjusted to match the format of the wind speed measurement from the weather station's dataset. The u and v components of the wind forecast are converted into a wind direction value in degrees and a wind magnitude value in $m.s^-1$.

Topographical data that can be used as a part of the feature set is obtained from the DEM data. Since the DEM data is constant for all time steps, it only needs to be referenced with the location of the weather station and ECMWF data. The projection used by the DEM differs from the projection that the weather stations and ECMWF datasets use. The latitude and longitude values from the weather station first need to be converted to the projection of the DEM before the location in the DEM can be determined. Once both the values use the same projection, the pixels of interest can be easily located and extracted. The DEM is not rotationally invariant and will always give the same data for a single location for any time step. By rotating the DEM in such a way that the ECMWF wind forecast is always directed vertically (90°), the data from the DEM for a single location will differ for any two time steps where the ECMWF weather forecast direction differs. This also slightly reduces the number of features that the learning methods require as it combines the ECMWF data with the DEM data. To rotate the DEM, an area of pixels, slightly larger than the required area, is extracted from the DEM, centred at the location of the point of interest. The pixels in this area are then rotated around the point of interest so that the new area has a "false north" in the direction of the wind estimation from the ECMWF data using a rotation matrix (equation 3.1 and figure 3.2b). The new, rotated area is then cropped down to the correct size. The process of rotating the DEM is shown in figure 3.2.

$$R(\theta) = \begin{bmatrix} Cos(\theta) & -Sin(\theta) \\ Sin(\theta) & Cos(\theta) \end{bmatrix} \tag{3.1}$$

Estimating wind speed and direction requires some data that can be tested to determine the accuracy of the estimations. The data that is used to test this comes from the weather station dataset and requires very little processing. The wind direction is modified so that the u and v components follow the same mathematical conventions as the u and v components of the wind from the ECMWF dataset. To convert the wind direction from the meteorological convention (θ_{met}) to the mathematical convention (θ_{math}), the rotational direction in which the angle increases is corrected (by multiplying θ_{met} by -1). The angle is then shifted (by adding 270°) to give the correct value for θ_{math} as shown in equation 3.2. The wind is then reduced into its u and v components, shown in equation 3.3 where θ



(a) A large area of the DEM is extracted (b) This area is then rotated using the wind direction (c) The desired area of the rotated DEM is extracted

Figure 3.2: The process of extracting a certain data point from the DEM.

is the direction of the wind using the mathematical convention, and m is the magnitude of the wind. In order for the weather station wind components to correlate with the rotated DEM, these components are also rotated using the direction of the corresponding ECMWF wind forecast as per equation 3.4, with θ_E being the forecast wind direction and u_R and v_R being the new, rotated components. Other values that are required from this dataset are the weather station's location as well as the date and time that the measurement was taken, as a reference for the other feature sets. The date and time that the wind measurement was taken is processed so that it represents the number of hours that have passed since 00:00:00 1900-01-01. This is to simplify the comparison with times from the ECMWF dataset.

$$\theta_{math} = 270^\circ - \theta_{met} \tag{3.2}$$

$$\begin{aligned} u &= Cos(\theta_{math}) \times m \\ v &= Sin(\theta_{math}) \times m \end{aligned} \tag{3.3}$$

$$\begin{aligned} u_R &= Cos(\theta_E) \times u - Sin(\theta_E) \times v \\ v_R &= Sin(\theta_E) \times u + Cos(\theta_E) \times v \end{aligned} \tag{3.4}$$

3.2.2 Data Cube Generation

Once the data has been processed into the correct formats, it is stored in a number of files on the file system. These files constitute the data cube that the learning system will use to generate training and test sets.

Each file contains data about a single set of features extracted at a single time step from the dataset and is stored as a NumPy array file. The folder in which each file is stored is determined by the properties of the file. At the top level, there are three folders, one for features from each dataset: the DEM, the ECMWF data and the weather station data. The DEM data is further subdivided with folders for each different size that is extracted from the DEM and then for the resolution at which the data was extracted. The weather station data folder is split into a number of folders that contain data for each individual weather station. The ECMWF data and DEM folders are divided in the same way to contain the data that correlates with each weather station. The DEM is then further divided into folders for each transformation that is applied to the DEM. The file name is the time step of the data point of interest, and the file is stored in the appropriate folder. This folder structure can be represented as follows:

```
Root
├── DEM
│   └── Extraction Size
│       └── DEM Resolution
│           └── Station Number
│               └── DEM Transform
│                   └── Time Step (npy)
├── ECMWF
│   └── Station Number
│       └── Time Step (.npy)
└── Weather Stations
    └── Station Number
        └── Time Step (.npy)
```

3.2.3 Data Cube Extraction

The set of data used by any particular learning systems requires all the data to be easily accessible and although the prepared data discussed above is easily accessible, it is slow to access it since each feature is stored in a separate file. To improve performance of the training system, the specific features required by the system are extracted from the file system and encoded into a single `.hdf5` file using `PyTables` [18]. Each basic feature set in the file is stored in a separate table, with the file name indicating the features included within the file. When the system loads a data file as input, the data from each table is matched with corresponding data from other tables, merged into a single list and added to a larger matrix containing all the input data for that experiment. Although for most cases storing all the data in the `.hdf5` file is faster and simpler than storing each feature set in a separate table, there are limitations on the maximum size of the metadata for each table. This causes errors when generating the `.hdf5` file for larger feature sets, thus the decision to store each feature set as a separate table allows the system to be expanded much more easily. Once the matrix of data, containing n entries, has been compiled for an experiment, the system will split the data in one of two possible ways, depending on whether the machine learning system will be trained to generalize the data or not. If the machine learning system will not be trained to generalize the data, the matrix is split by first shuffling the data, then selecting the first p entries of the matrix to be used as the training set, and the remaining $n - -p$ entries of the data are used as the test set, where, given a percentage split $P_{\%}$, p can be calculated with

$$p = n\frac{P_{\%}}{100} \tag{3.5}$$

3.3 Features

The features that the learning system uses are prepared in different ways. Some features are used without any modification, such as the ECMWF data. Three transformations are applied individually to the DEM using operators from `gdaldem` [19], before any values are extracted, to create different feature sets, namely:

- A Topographic Position Index;
- A Terrain Ruggedness Index; and
- A slope map.

The other transformations are applied only to the extracted areas of the DEM using methods from `SciPy` [20] and `scikit-image` [21]. These other transformations are:

- A Fast Fourier Transform;
- Local Binary Patterns; and
- The Sobel operator.

3.3.1 ECMWF Input Data

The European Centre for Medium-Range Weather Forecasts, or ECMWF, works to provide numerical weather forecasts as well as research into methods that can improve these forecasts. The ECMWF also provides various other meteorological services that are less relevant to the research in this study. The ECMWF's numerical weather forecasts include data for various temperature factors, such as wind data, ice data, precipitation data, pressure data, and cloud data. For the purpose of this study, only the wind, temperature and pressure data are used. It is important to note that although the ECMWF data has a temporal element, this element is only used to determine the correct target weather station measurement with which the learning system is trained. This removes any temporal dependency that the feature set may have.

3.3.2 Unaltered DEM

The Digital Elevation Map (DEM) provides elevation information for a vast area and in the case of this study the area that is covered by the DEM is the entirety of South Africa with a resolution of 90 meters. The DEM provides land based elevation values in meters above sea level. Multiple layers of this DEM are generated by aggregating the pixel values to create lower resolution DEMs with resolutions of 180 meters, 360 meters, 720 meters and 1440 meters.

3.3.3 Topographic Positioning Index

The Topographic Positioning Index, or TPI, is similar to the DEM in that it represents elevation data for the location. Unlike the DEM however, the TPI shows the elevation relative to its surroundings, whereas the DEM shows the absolute elevation above sea level at each point. A negative TPI value indicates a valley or a gulley, a near zero TPI value indicates a flat area or a gentle slope, and a positive TPI value is indicative of a peak or a ridge. The TPI of a single point, x_0, is calculated from the difference between the elevation at that point and the average elevation of the neighbouring n points, $(x_1, x_2, \ldots x_n)$ as depicted in figure 3.3, and as can be seen in equation 3.6 used in Weiss [22] and Wilson et al. [23]. The TPI is prepared over the entire extent of the DEM and small areas of the DEM are extracted as input for the system, as follows:

x_1	x_2	x_3
x_8	x_0	x_4
x_7	x_6	x_5

Figure 3.3: A representation of neighbouring points used in equations 3.6 through 3.8.

$$TPI_{x_0} = x_0 - \sum_{i=1}^{n} \frac{x_i}{n} \tag{3.6}$$

3.3.4 Terrain Ruggedness Index

The Terrain Ruggedness Index, or TRI, depicts how rough an area is by comparing a central point to the surrounding points The TRI of a single point is calculated by averaging the absolute value of the difference in elevation between the point and each neighbouring point, $(x_1, x_2, \dots x_n)$, using

$$TRI_{x_0} = \sum_{i=1}^{n} \frac{|x_0 - x_i|}{n} \tag{3.7}$$

from Wilson et al. [23] and Valentine et al. [24]. This is a more straightforward adaption of the TRI calculation that uses the square root of the sum of squares of elevation distances, as in

$$TRI_{x_0} = \sqrt{\sum_{i=1}^{n} \frac{(x_0 - x_i)^2}{n}} \tag{3.8}$$

originally presented by Riley [25]. The TRI is prepared over the entire extent of the DEM and small areas of the DEM are extracted as input for the system.

3.3.5 Slope Map

A slope is an indication of the steepness of an inclination represented in degrees from the horizontal plane. A slope is always a positive value as it does not consider the direction in which the slope is facing. This direction is represented by the aspect value, which is not used in this study. The slope is calculated using a 3×3 area around a point in the DEM, as in figure 3.3. This leads to boundary points in the DEM, which result in a slope with a value of zero, since the slope map is calculated for the entire DEM using

$$\begin{aligned} p &= \frac{((x_3 + 2x_4 + x_5) - (x_1 + 2x_8 + x_7))}{8\Delta x} \\ q &= \frac{((x_3 + 2x_2 + x_1) - (x_5 + 2x_6 + x_7))}{8\Delta y} \\ s &= \sqrt{p^2 + q^2} \end{aligned} \tag{3.9}$$

where Δx and Δy is the resolution of the DEM [26]. As the slope operator requires all values in the 3×3 area surrounding the central pixel, the slope value for boundary locations cannot be calculated; however, only small (non-boundary) portions are extracted by the system, and so this is not a problem for the study [27, 28].

3.3.6 Fast Fourier Transform

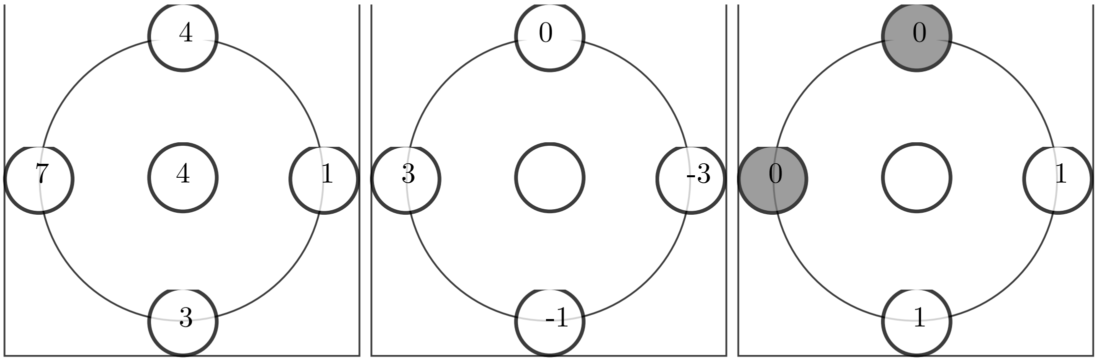
A Fourier Transform reduces a signal into its base frequencies, producing a complex function that represents the magnitude of a frequency in the original function as well as the phase angle of a basic sinusoid in that frequency. A Fast Fourier Transform (FFT) is an algorithm that computes a discrete form of the Fourier Transform for a sequence or signal. If an image can be considered as a 2-dimensional signal, by extending the FFT to 2-dimensions, it can be used to extract the frequencies and phase angles from the image. The Discrete Fourier Transform of a point is a complex number which may be represented in magnitude and phase angle form and can be calculated for a point of the DEM using

$$\mathcal{F}(u, v) = \frac{1}{n} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} F(j, k) e^{\left(\frac{-2\pi i}{n}(uj+vk)\right)}, [29]. \tag{3.10}$$

The FFT of the DEM is generated using methods from `SciPy` [20]. The FFT is shifted so that the zero-frequency component is centred, then the resulting FFT matrix is used to generate matrices representing the phase angle and the amplitude of the FFT. These matrices are then flattened and used as input data for the learning system.

3.3.7 Local Binary Patterns

Local Binary Patterns, or LBPs, are a simple texture operator that labels using thresholding on the neighbourhood of each pixel. The LBP bilinearly interpolates a number of sample points in a circular neighbourhood around the central pixel. The difference between each sample point and the central pixel is calculated and used to determine the



(a) Sampling (b) Difference (c) Thresholding

Figure 3.4: The steps involved in determining Local Binary Patterns.

pattern using a threshold shown in equation 3.11 where n represents the number of points to be sampled from the neighbourhood, g_c is the central pixel and g_i is the i^{th} point in the neighbourhood [30].

$$LBP_n = \sum_{i=0}^{n-1} s(g_i - g_c)2^i \qquad s(x) = \begin{cases} 1, \text{if } x \geq 0; \\ 0, \text{otherwise.} \end{cases} \tag{3.11}$$

An example of this process can be seen in figure 3.4 with the corresponding pattern calculation shown in equation 3.12. LBPs are implemented employing `scikit-image` [21], using three circularly symmetric neighbour set points with a circular radius of 0.1.

$$LBP_4 = 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 = 3 \tag{3.12}$$

3.3.8 Sobel Operator

The Sobel operator is similar to the slope operator in the sense that it calculates the gradient of an inclination; however, unlike the slope operator, values from the Sobel operator can be negative. More formally, the Sobel is a slightly modified version of the Prewitt operator, a 3×3 pixel edge gradient operator, with a similar modification resulting in the Frei-Chen operator [29]. The most notable difference between these operators is whether the operator is more sensitive to horizontal and vertical edges or diagonal edges. Each of these operators are calculated using the same set of equations with a different value for one constant multiplier, which, in the case of the Sobel operator, is 2. The Sobel operator considers a 3×3 area, as in figure 3.3, and is calculated using

$$S(x,y) = \sqrt{(S_R(x_0))^2 + (S_C(x_0))^2} \tag{3.13}$$

which requires two intermediate calculations, shown by

$$\begin{aligned} S_R(x,y) &= \frac{1}{4}\left((x_3 + 2x_4 + x_5) - (x_1 + 2x_8 + x_7)\right), \text{ and} \\ S_R(x,y) &= \frac{1}{4}\left((x_1 + 2x_2 + x_3) - (x_7 + 2x_6 + x_5)\right), \end{aligned} \tag{3.14}$$

from Pratt [29].

Although both intermediate calculations are used for the Sobel operator, each can provide slightly different information, resulting in data from three operators, namely:

- the partial Sobel operator over only the horizontal coordinate ($S_C(x_0)$);
- the partial Sobel operator over only the vertical coordinate ($S_R(x_0)$); and
- the Sobel operator across all coordinate directions ($S(x_0)$).

The values from these operators are then aggregated by summing the values across both the horizontal and vertical axes, as follows

$$\begin{aligned} S_x(y) &= \sum_{i=0}^{X} S(i,y) \\ S_y(x) &= \sum_{i=0}^{Y} S(x,i) \end{aligned} \tag{3.15}$$

to reduce the size of the feature set.

The Sobel operator used in this study is calculated using methods from `scikit-image` [21].

3.4 Learning Methods

For this study, the chosen learning methods are Random Forests and linear regression, both of which are implemented through the `scikit-learn` module, a machine learning module for Python [31]. The learning methods in this section take as an input a vector $\mathbf{D} = \mathbf{x}_i, y_i$ where $\mathbf{x}_i \in \mathbb{R}^n$, $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n}]$ contains the input features and $y_i \in \mathbb{R}$ is the target value, and n is the dimensionality of the feature vectors.

3.4.1 Linear Regression

Linear regression attempts to approximate a linear relationship within the data. Linear regression achieves this by fitting the data to a regression model, which in the case of this study will be in the form of equation 3.16. Using the input vector, $\mathbf{x}_i$, and the target value, y_i, the model will estimate an unknown vector, $\mathbf{a} \in \mathbb{R}^n$, to allow for estimations of a different, unknown value, $y_{i_E} \in \mathbb{R}$, using known values for $\mathbf{x}_i$. The method used to fit the linear regression for this study is known as the ordinary least squares method, which calculates the regression coefficients, $\mathbf{a}$, in such a way that it minimizes the sum of the squares of the difference between the target value, $y_{i_E} - y_i$, and the line generated by using the input vector, $\mathbf{x}_i$, and the regression coefficients; $\mathbf{a}$ is represented with equation 3.16 [32].

$$y_{i_E} = a_1 x_{i,1} + a_2 x_{i,2} + \ldots + a_n x_{i,n} \tag{3.16}$$

The ordinary least squares method calculates the values for $\mathbf{a}$ using

$$\mathbf{a} = \left(\frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i \right)^{-1} \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i y_i, \tag{3.17}$$

where m is the number of data points in the training dataset [33].

3.4.2 Random Forests

Random Forests is a machine learning method that can be parallelized easily to improve performance with larger data systems. Random Forests work in a similar manner to decision trees, but rather than forming a single tree from all the data, many trees are formed from independent random samples of the data. These systems perform with an accuracy that is similar to other ensemble algorithms, while being more robust when dealing with noisy data [34]. Different ways to implement Random Forests exist, using different classification methods, with the Binary Hierarchical Classification (BHC) method giving higher accuracies and lower standard deviations than the Classification and Regression Trees (CART) method as shown in the study by Ham et al. [35].

Gislason et al. [36] have shown that Random Forests, when compared to other ensemble algorithms, give similar accuracies. While the accuracy of Random Forests may not always be better than other ensemble algorithms, training Random Forests is much faster [36]. Additionally, Random Forests do not overfit the data, and can accurately classify outliers where other algorithms may incorrectly classify these [34, 36]. The research of both Ham et al. [35] and Gislason et al. [36] is of particular interest, as these studies were conducted using spatial datasets, with the dataset for the latter using remote sensing data and geographic data.

Random Forests for regression form a collection of a large number of tree-structured classifiers $\{h(\mathbf{x}, \Theta_k), k = 1, \ldots\}$, where $\{\Theta_k\}$ are independent identically distributed random vectors so that the tree predictor $h(\mathbf{x}, \Theta_k)$ takes on numerical values. The Random Forests predictor is then formed by taking the average over k of the trees $\{h(\mathbf{x}, \Theta_k)\}$ [34, 36].

Parameter Validation for Random Forests

Random Forests have a number of parameters that can be adjusted to improve accuracy. The parameters that were adjusted in this study include:

- The maximum depth of the trees in the forest (Depth);
- The maximum number of features a tree may use from a feature set with size n (Features);
- The total number of trees per forest (Trees);
- The minimum number of samples required to split an internal node (Split); and
- The minimum number of samples in newly created leaves (Leaves).

For this study, these parameters are validated using k-fold cross validation on 10% of the data, and a grid search method to determine which parameter has the most accurate results. The results of the cross validation of over 1568 combinations of parameters showed that the mean absolute error falls in the range of $61° - 64°$, with a standard deviation on this error of $1.6° - 2.8°$. Only minor patterns emerged in the top parameter sets with the least mean error, as can be seen in table 3.1. The noticeable pattern is that a lower number of trees, a minimum of one leaf node, a maximum depth of seven, and limiting each tree to use a maximum of $\sqrt{n}$ features provide marginally more accurate results.

Rank	Mean	Std Dev	Features	Split	Leaves	Depth	Trees
1	61.209	2.347	$\sqrt{n}$	5	1	7	30
2	61.225	2.230	$\sqrt{n}$	5	1	9	80
3	61.285	2.418	$log_2(n)$	5	5	7	30
4	61.319	2.340	$\sqrt{n}$	2	1	7	115
5	61.332	2.446	$\sqrt{n}$	2	1	7	85

Table 3.1: Validation scores and parameters for the top five parameter sets.

3.5 Comparative Metrics

In order to compare the performance of multiple learning setups, the results need to be standardized and analyzed using some form of metric. The metrics that are considered to be used for comparison are:

- percentile values;
- inter-quartile range;
- mean angle; and
- error tolerance.

3.5.1 Percentiles

The n^{th} percentile of a set of results is a measure that gives a value p of the data so that n% of the results are less than or equal to p and $(100 - n)$% of the results are above p. Percentiles can be used to extract various information from the data, such as the median, which is the 50^{th} percentile. The minimum and maximum values of the data are the 0^{th} and 100^{th} percentiles respectively. Percentiles can be useful in finding the maximum value of the lowest n% of the data, or alternatively the minimum value of the $(100 - n)$% of the data. For absolute error, as is used in this study, a percentile can be used to give the bottom n% of the data, removing the higher, possibly outlying, values.

3.5.2 Interquartile Range

A quartile is a type of percentile where each consecutive pair of quartiles contains exactly 25% of the data. This results in five different values, 0%, 25%, 50%, 75% and 100%, which are usually named Q_0, Q_1, Q_2, Q_3 and Q_4 respectively. The interquartile range covers the centre 50% of the data, namely all the data greater than Q_1 that is also less than

Q_3. This is a very good measure for comparing the extent of the majority of the data in such a way that outlying data does not make the data seem to cover a larger area than it actually does. In the case of our experiment, the interquartile range does provide an adequate comparison, although the 75^{th} percentile may be able to give either similar or equivalent information.

3.5.3 Mean

The mean is similar to the median in the sense that it is a measurement of central tendency. Unlike the median, which is calculated by finding the central point of the data, the mean uses the values of the data to calculate the average of the values. The mean of the data gives the expected value around which the results are centred and, given a set of results, $\mathbf{y} = [y_1, y_2, \ldots, y_n]$, the mean value, $\bar{y}$, can be calculated using

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{3.18}$$

If the data is not symmetric around the mean, the data can be considered to be skewed. The direction in which the data is skewed can be indicated by comparing the mean to the median. The direction in which the data is skewed can be confirmed or contradicted by comparing the median to the 75^{th} and the 25^{th} percentiles. Extremely large values relative to the data, either negative or positive, can have a significant impact on the value of the mean, making the value of the mean insufficient to provide information about the results. The impact of large outliers can be reduced by calculating the mean on only a subset of the data that excludes these outliers, although the use of other metrics in cases like this may provide better information.

3.5.4 Error Tolerance

Reasonable, or tolerable, error is when an erroneous result can still provide accurate information, despite the error, through further simplification of the result b y means such as categorization. Wind direction, for example, does not need an estimation to be precise, but can be categorized into a range of angles, such as the cardinal points on a compass, which allows the estimation to tolerate an error of a few degrees without affecting the significance of the result. Given the error from a set of results, $\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n]$, and an error tolerance value, t, the percentage of the results that contain errors that are greater than the error tolerance value, $\epsilon_{\%}(\boldsymbol{\varepsilon}, t)$ can be calculated with:

$$\epsilon_{\%}(\boldsymbol{\varepsilon}, t) = \frac{100}{n}\sum_{i=1}^{n} f(\varepsilon_i) \qquad \text{where } f(\varepsilon_i) = \begin{cases} 1, \text{if } \varepsilon_i \geq t; \\ 0, \text{otherwise.} \end{cases} \tag{3.19}$$

Since $\epsilon(\boldsymbol{\varepsilon}, t)$ calculates the percentage of results that have an error larger than t, any outlying error values would be classified as greater than t, and will not affect the value of $\epsilon(\boldsymbol{\varepsilon}, t)$ any more than another error value greater than t, resulting in very little impact on outliers for the error tolerance value.

3.6 Summary

This chapter discussed the method through which the study produces results, beginning with what data sources are used and what data is extracted from these data sources. How the data is prepared was then discussed, explaining how the weather station data is transformed to conform to the same convention that the ECMWF data uses, as well as how the DEM is rotated to align with the ECMWF wind forecast direction, which increases the variation in features extracted from the DEM. Once the data was prepared, transforms that are performed on the DEM to produce different sets of features were discussed, covering the methods used to calculate:

- the TPI of the entire DEM;
- the TRI of the entire DEM;
- a slope map for the entire DEM;
- LBPs for the extracted portion of the DEM;
- an FFT of the extracted portion of the DEM; and
- an aggregation of the Sobel operator performed on an extracted portion of the DEM.

The selection and workings of the two machine learning methods used for this study, Random Forests and linear regression, were explained. Finally, the metrics that are used to compare the various results were discussed. The following chapter, Chapter 4, implements the above methods to produce and analyze the results.

Chapter 4

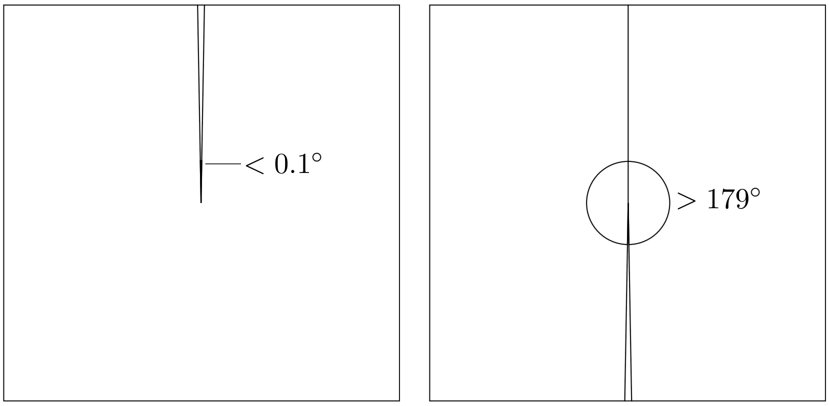
Results and Analysis

This chapter presents the results of the various tests using a combination of the following features:

- various transformations of the Digital Elevation Map (DEM);
- changing the size of the extracted area of the DEM;
- using multiple layers of differing resolution DEMs;
- training the system using different learning methods; and
- scaling the data in different ways.

For wind direction estimation, the absolute error is considered and represented across three graphs, shown in figures 4.2 to 4.9. The first graph contains four different percentile ranges, 25^{th}, 50^{th}, 75^{th}, and 85^{th} percentiles. The second graph shows the mean absolute error, whilst the third graph shows the percentage of the results that are greater than what may be a tolerable level of accuracy, which for the case of this study is 15°. For wind speed estimation, only the relative error is considered and represented on a single graph, representing the four percentile ranges, shown in figures 4.10 to 4.17. The minimum and maximum range of error angles, $\Delta\theta$, for wind direction are shown in figure 4.1, indicating the angle of error can extend both clockwise and counter-clockwise, with an error of 180° as the largest potential error.

When comparing results, the analysis can vary when comparing different metrics. For this reason the 85^{th} percentile is the primary metric that is considered when analysing the results. If the 85^{th} percentile of two result sets are similar, the percentage of the results



(a) Minimum error angle (b) Maximum error angle

Figure 4.1: A representation of the range of minimum and maximum angles.

that have an error greater than the tolerable error range of 15° is considered as a secondary metric. An initial comparison is included in each set of results, comparing the European Centre for Medium-Range Weather Forecast (ECMWF) data directly with the weather station data as a benchmark result. The numeric values used to create all these graphs are included in a tabular form in Appendix B. All graphs in this section were generated using `matplotlib` [37].

This chapter discusses the best approach to estimate wind direction, explaining how this approach is used to generalize the wind direction estimation. Estimating wind speed and the generalization of the wind speed estimation are then discussed using a similar approach. A breakdown of the approach used can be seen in table 4.1

	Forecast	Transforms	DEM Area	DEM Layers	Learners	Scaling
Transforms	Benchmark	Comparison	3×3	90m only	Random Forests	No scaling
DEM Area	Benchmark	Top 3	Comparison	90m only	Random Forests	No scaling
DEM Layers	Benchmark	Top 3	Best	Comparison	Random Forests	No scaling
Learners	Benchmark	Best	Best	Best	Comparison	No scaling
Scaling	Benchmark	Best	Best	Comparison	Best	Comparison
Sanity	Benchmark	Best	Comparison	Comparison	Best	Best

Table 4.1: Table representing the test structure through the features selected for each experiemnt to produce the optimal results. Each experiment may consider previous results when selecting features.

4.1 Transformations of the DEM

Beginning with no indication of what feature set may produce the best results, the feature set with the largest number of different types of features is chosen as a starting point. The largest feature set that is used contains all the various transformations that are applied to the DEM, namely:

- the Topographical Positioning Index (TPI) calculated from the entire DEM;
- the Terrain Ruggedness Index (TRI) calculated from the entire DEM;
- a slope map calculated from the entire DEM;
- Local Binary Patterns (LBP) calculated from the extracted segment of the DEM;
- a Fast Fourier Transform (FFT) calculated from the extracted segment of the DEM; and
- the Sobel operator applied to the extracted segment of the DEM.

In addition to these transforms, the machine learning system is trained with two additional feature sets: a feature set where an unaltered version of the DEM is used and a feature set that excludes all DEM data.

For training the system with the various transformations, the area of the DEM that is extracted is a 3×3 area, extracted from only the 90m of the DEM. The data is not scaled in any way and is trained using Random forests.

The results of training this system with the specified features can be seen in figure 4.2. As can be seen, the forecast data produces the poorest results and the results of using the unaltered DEM provide the best results. Using the TRI or slope shading also provide similar results. Not including any data from the DEM does show an improvement over the forecast data, however using any transformation on the DEM will provide a further improvement to the estimation. The FFT performs the poorest of all the DEM transformations, whilst the TPI, LBPs and the Sobel operator all perform similarly, although slightly worse than the TRI, slope shading and the unaltered DEM. For this reason, subsequent systems will be trained using either the unaltered DEM, the TRI or the slope map.

4.2 Areas of the DEM

Following from the previous set of results, the next set of results can be generated by comparing the effects of adjusting the area that is extracted from the DEM with each chosen transform. The area extracted from the DEM is a square, measured by the pixel

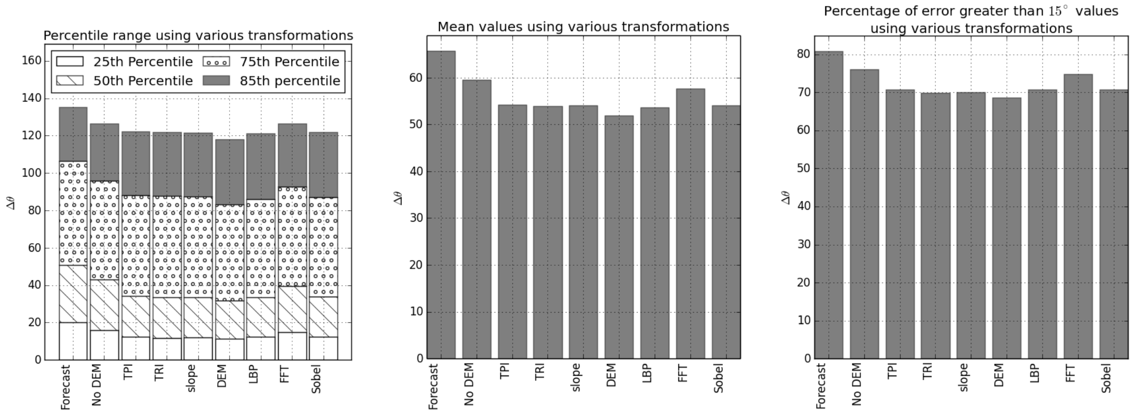


Figure 4.2: A comparison of the absolute error of wind direction estimation results, $\Delta\theta$, of each transformation applied to the DEM for all stations.

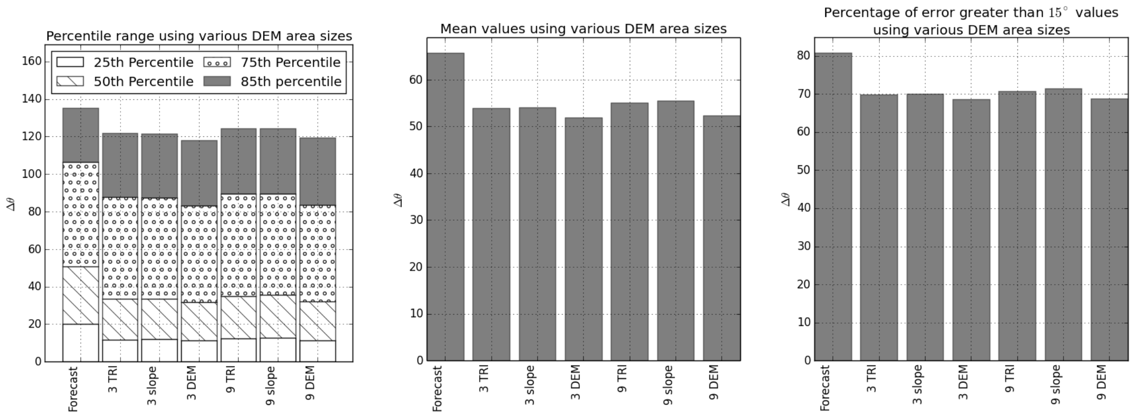


Figure 4.3: A comparison of the absolute error of wind direction estimation results, $\Delta\theta$, when using different sized areas of the DEM for all stations.

length of the area. Only two different sizes are considered, a 3×3 and a 9×9 area. As is to be expected based on the results of the the transformations in figure 4.2, the unaltered DEM performs better than the TRI and slope shading transformations for both the 3×3 and the 9×9 areas, although the 3×3 area does produce better results than the 9×9 area, even though there is a larger input set with the 9×9 area. This may be due to a larger portion of the input being interdependent on each other than with the 3×3 area. Using this information, the following experiments will use a 3×3 area of the DEM, along with the previously selected transformations

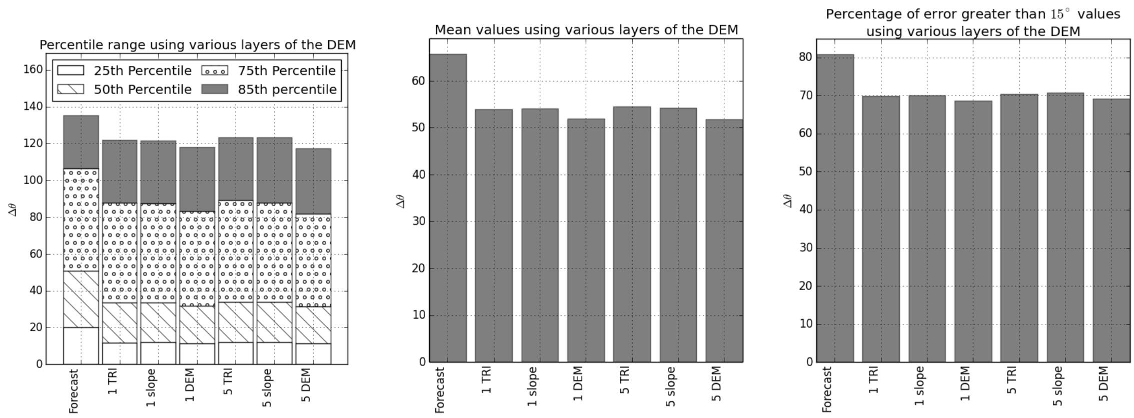


Figure 4.4: A comparison of the absolute error of wind direction estimation results, $\Delta\theta$, when adding extra layers of the DEM as additional inputs for all stations.

4.3 Layering the DEM

Rather than using extremely large areas extracted from the DEM to increase the range of data extracted from the DEM, multiple layers of varying resolutions of the DEM can be extracted and used to produce a similar effect. Using this idea, we can compare the effect of introducing multiple layers of the DEM to the learning process. Previous results were all produced using only the version of the DEM with the finest resolution — the 90m version of the DEM. This is the same as using only a single layer of the DEM at that resolution. Using a further four layers of the DEM, results can be used to confirm or deny whether or not there is any improvement gained by adding extra layers of the DEM. The four layers added are the 180m, 360m, 720m and 1440m versions of the DEM, which are used alongside the 90m version of the DEM to produce a total of 5 layers. These results are compared to the results of using only the 90m version of the DEM in figure 4.4. Similar to the effect of increasing the size of the area extracted from the DEM, the unaltered DEM produces better results than using either the TRI or the slope shading. Although the difference between using the unaltered DEM is very similar to using only the 90m version of the DEM or using all five layers of the DEM, when comparing the percentage of the results that are outside the tolerable error range, the 90m version of the DEM provides better results than when using all five layers of the DEM. Because of this, only the 90m DEM will be used to produce the next set of results.

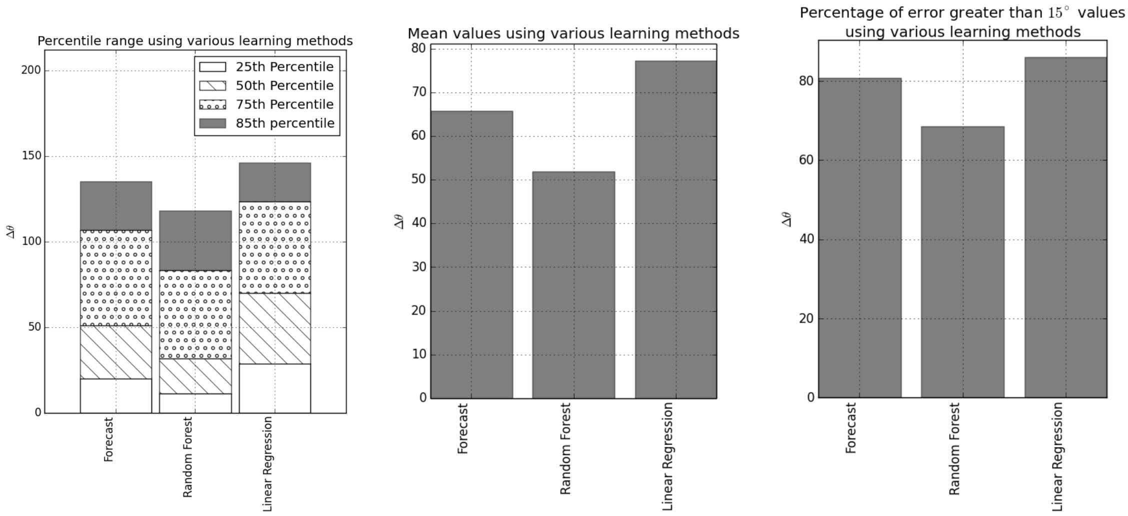


Figure 4.5: A comparison of the absolute error of wind direction estimation results, $\Delta\theta$, when using different learning methods for all stations.

4.4 Learning Methods

The results thus far have all been produced using Random Forests, however this is not the only learning method available. The other learning method that is considered is linear regression. By using the previous results, the next set of results can be generated using a 3×3 area of only the unaltered 90m DEM, using both Random Forests and linear regression, the results of which can be seen in figure 4.5. It can be seen that Random Forests produce a significant improvement over both linear regression and the forecast data, with linear regression performing worse than the forecast. This could indicate a non-linear relationship between the wind direction and the input data. Further experiments will only consider the performance of Random Forests.

4.5 Scaling the Data

The only remaining way considered in this study by which the input data is manipulated is by scaling the data. Scaling the data brings the range of possible values into a more bounded range, as well as shaping the data to better fit the normal curve. This reduces the effect of large outliers in the data. Three different approaches are used to scale the data, namely:

- No scaling — No scaling is applied to the data, as in previous result sets;

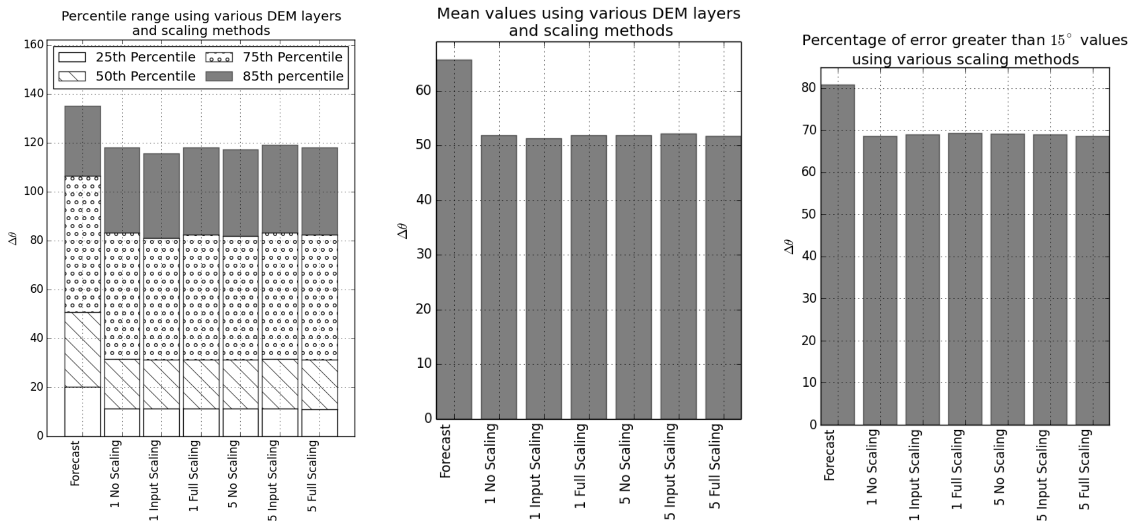


Figure 4.6: A comparison of the absolute error of wind direction estimation results, $\Delta\theta$, when the data is scaled differently for all stations.

- Input scaling — Only the input data is scaled; and
- Full scaling — All data is scaled. Both input data and target data used for training.

Using these approaches to scale the data, the results are produced using a 3×3 area of the unaltered DEM. Both the 90m version of the DEM and the five layers of the DEM are considered due to the previous similarity in results between these inputs in figure 4.4. In figure 4.6 it can be seen that there is a slight improvement in using input scaling with only the unaltered 90m DEM. However this improvement is not replicated when using five layers of the unaltered DEM, with an actual decrease in accuracy when scaling the data. Full data scaling shows a decrease in performance for both the 90m unaltered DEM and the five layers of unaltered DEM.

A final set of results is generated using combinations of features to ensure that there are no improvements in the results when deviating from the approach to select optimal features. All these results are produced using Random Forests without scaling the data, and use the following feature sets:

- a 3×3 area extracted from only the unaltered 90m DEM;
- a 9×9 area extracted from only the unaltered 90m DEM;
- a 3×3 area extracted from all five layers of the unaltered DEM; and
- a 9×9 area extracted from all five layers of the unaltered DEM.

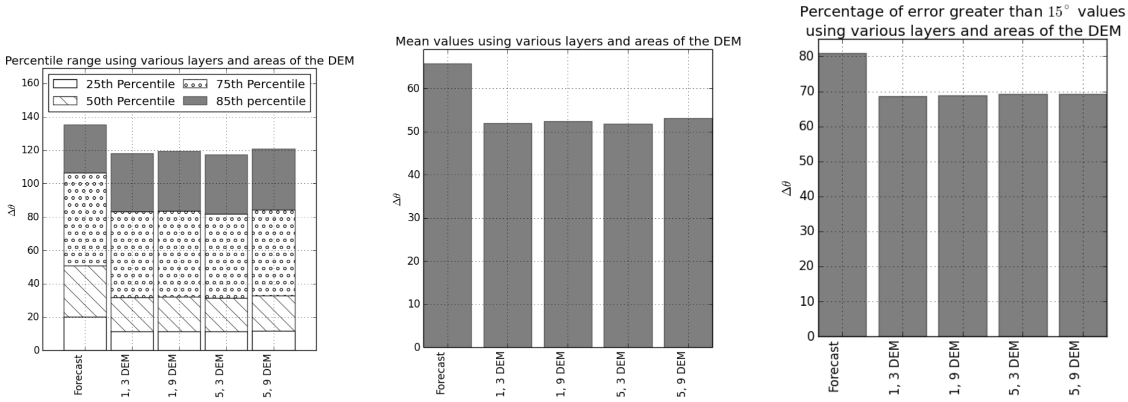


Figure 4.7: A comparison of the absolute error of wind direction estimation results, $\Delta\theta$, when various implementations for all stations.

As can be seen in figure 4.7, extracting a 3×3 area from the DEM provides the best results and there is with very little difference between using only the unaltered 90m version of the DEM and using five layers of the DEM as before. Although both of these results are slightly worse than a previous result, using input scaling on a 3×3 area extracted from only the unaltered 90m DEM, making this result the best performing result.

4.6 Generalization

Although the results above show a definite improvement upon the forecast data, it does not accurately represent how well the system can be trained to estimate wind direction for arbitrary locations, as the system is trained with all the weather station data. In this section, the ability of the system to generalize for unseen locations is tested and analyzed using the same features as above.

Considering all the transformations that are applied to the DEM, figure 4.8 shows that there is a more noticeable difference between each different transform than what is shown in figure 4.2. Figure 4.8 also shows that the difference in error between the forecast data and the results are significantly smaller, indicating a large decrease in estimation accuracy; in particular the results from the TRI show a larger error than even the forecast data. The most accurate result is produced by LBPs, with similar results from both FFTs and the unaltered DEM.

Figure 4.9 compares the results when the system is trained to generalize with the best performing features from previous tests. It can be seen that training the system with five

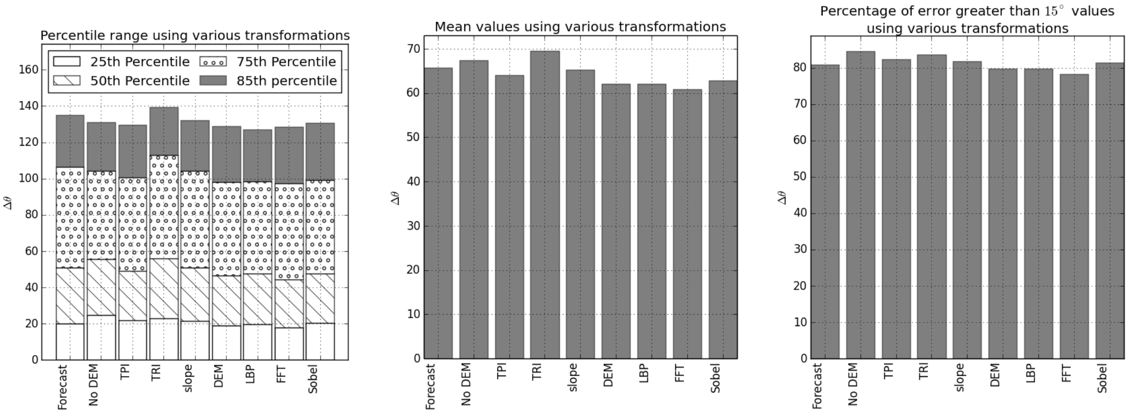


Figure 4.8: A comparison of the absolute error of wind direction estimation results, $\Delta\theta$, of each transformation applied to the DEM for certain untrained stations.

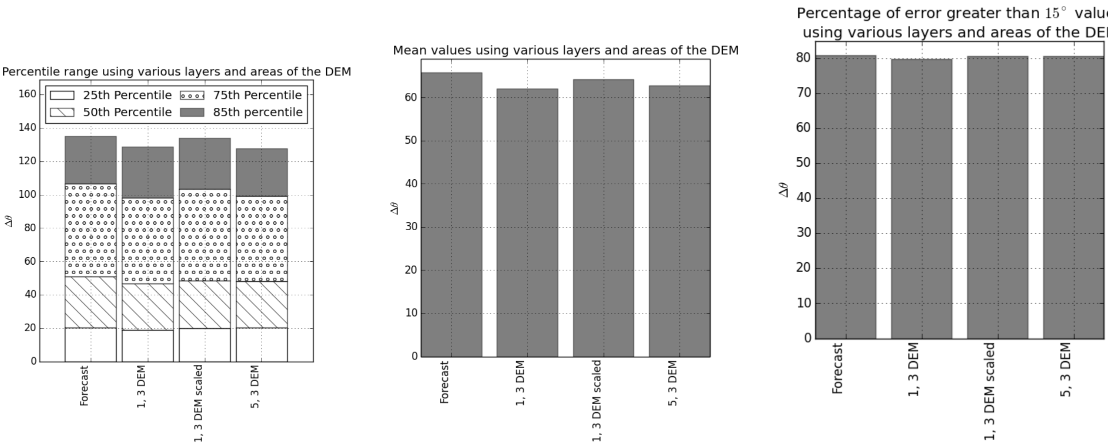


Figure 4.9: A comparison of the absolute error of wind direction estimation results, $\Delta\theta$, when adding extra layers of the DEM as additional inputs for certain untrained stations.

layers of the unaltered DEM provides no significant improvement and scaling the input before training the data shows a noticeable decrease in accuracy. This indicates that the best approach to generalizing the data is to train the system using LBPs on a 3×3 area of only the unaltered 90m DEM using Random Forests without scaling any of the data.

4.7 Wind Speed Estimation

A similar approach to the estimation of wind direction can be used to produce results for the estimation of the magnitude of the wind. A major difference is that only the 85^{th} percentile value is considered when comparing results, as no tolerable level of error is considered for wind speed error. The other alternative metric, the mean, could also be used as a comparison. However the large values of the mean, with the mean values

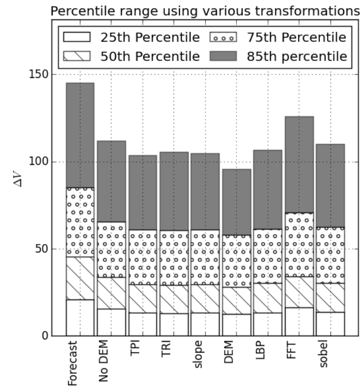


Figure 4.10: Comparison of the relative error of wind magnitude estimation results, $\mathbf{\Delta V}$, of each transformation applied to the DEM for all stations.

ranging between 4000% and 20000% error, whilst the median values range between 55% and 65% error, indicate that the mean is affected by outliers in the results rather than a representation of the overall data. For this reason the graphs representing the means of the results are excluded.

Beginning with the various transformations, figure 4.10 shows a marked improvement in accuracy for all transformations. The transformations showing the best improvements are the unaltered DEM, the TPI and slope shading, with most results showing significant differences between most transforms.

Using these three transforms, along with varying sizes of the extracted area of the DEM, the results depicted in figure 4.11 can be compared. As before, the unaltered DEM performs significantly better than both the TPI and slope shading for any size of the area extracted from the DEM. The 3×3 area of the DEM does provide better results than the 9×9 area of the DEM for each of these transforms, although it is interesting to note how increasing the size of the area extracted from the DEM affects the accuracy using the TPI transform.

As the 3×3 area of the DEM performs better for each of the three transforms, the next feature to consider is the number of layers that are used when training the data. Figure 4.12 shows that using the unaltered 90m DEM still provides the best accuracy. Slope shading does improve when using all five layers of the DEM, however it still provides lower accuracy than when using the unaltered 90m DEM.

Changing the learning method can have a significant effect on the accuracy of the system. Figure 4.13 shows that using Random Forests significantly improve the accuracy

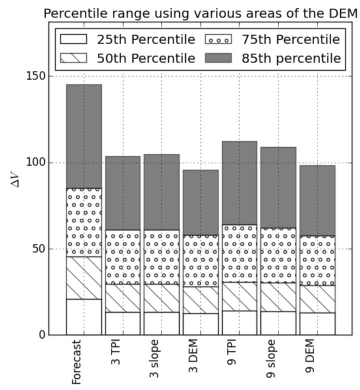


Figure 4.11: A comparison of the relative error of wind magnitude estimation results, $\mathbf{\Delta V}$, when using different sized areas of the DEM for all stations.

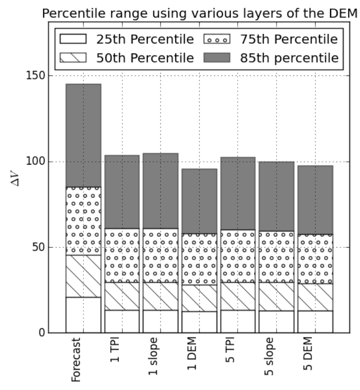


Figure 4.12: A comparison of the relative error of wind magnitude estimation results, $\mathbf{\Delta V}$, when adding extra layers of the DEM as additional inputs for all stations.

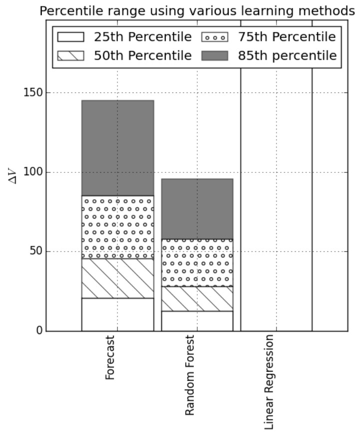


Figure 4.13: A comparison of the relative error of wind magnitude estimation results, $\mathbf{\Delta V}$, when using different learning methods for all stations. The linear regression data extends beyond the bounds of this graph.

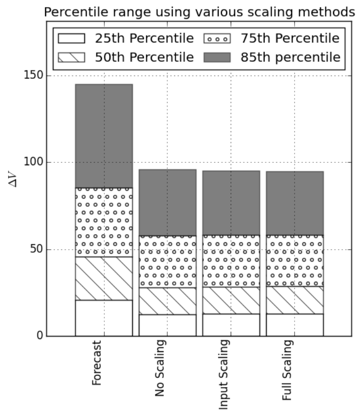


Figure 4.14: A comparison of the relative error of wind magnitude estimation results, $\mathbf{\Delta V}$, when the data is scaled differently for all stations.

of the system. Linear regression, however, indicates that there is no linear relation between the input data and the estimation of wind magnitude, with even the 25^{th} percentile extending far past the upper bounds of the graph.

Scaling the data, as the results show in figure 4.14, provides extremely little benefit when estimating the magnitude of the wind, with scaling the input providing the best results, albeit marginally.

Figure 4.15 indicates that the selection of features has been correct, with the best performing features for wind speed estimation resulting from a 3×3 area extracted from the unaltered 90m DEM, trained using Random Forests with the input data scaled before training is performed.

When considering the ability of the system to generalize data, figure 4.16 shows that

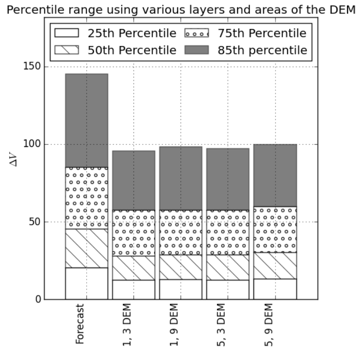


Figure 4.15: A comparison of the relative error of wind magnitude estimation results, $\mathbf{\Delta V}$, when several of the above are implemented for all stations.

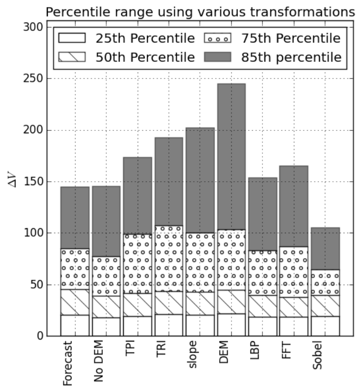


Figure 4.16: A comparison of the relative error of wind magnitude estimation results, $\mathbf{\Delta V}$, of each transformation applied to the DEM for certain untrained stations.

the system is poor at generalizing the wind speed data for most transforms and extending the estimation of the magnitude of the wind. The Sobel operator does outperform all other transforms by a significant margin, and is the only transform that provides better accuracy when compared to the forecast data.

Figure 4.17 indicates that scaling the input of the data may improve the performance when using the unaltered 90m DEM in such a way that it is more accurate than the forecast data. However, the Sobel operator is still more accurate, showing that the most accurate of the above set of features when generalizing wind speed estimation is using a 3×3 area extracted from the unaltered 90m DEM, trained with Random Forests without any data scaling.

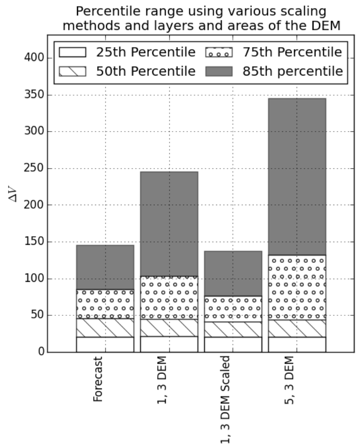


Figure 4.17: A comparison of the relative error of wind magnitude estimation results, $\Delta\mathbf{V}$, when adding extra layers of the DEM as additional inputs for certain untrained stations.

4.8 Summary

From the results above it can be seen that the accuracy of the forecast data is a problem, with the 85^{th} percentile of the forecast data having an error of as much as 135° and over 80% of the forecast data has an error that is greater than the 15° tolerable error range. This error range may be attributed to the fact that the ECMWF data provides a large area forecast, whilst the weather station data provides a very localized, small area measurement. Despite this error margin, minor improvements in accuracy can be achieved by using a 3×3 area extracted from various transformations of the 90m DEM and using Random Forests training. For wind direction estimation, scaling the input data and using the unaltered DEM provides the best results for an overall estimation, whilst for a more generalized estimation – where that system has never seen a similar location – not scaling the data and using LBPs provide more accurate results. For wind magnitude estimation, scaling the input data and using the unaltered DEM provide the most accurate results, whilst the Sobel operator without scaling any data provides more accurate results when the system is trained to generalize the estimation of the magnitude of the wind.

Chapter 5

Conclusion and Future Work

This study shows that it is possible to use machine learning with weather forecast data to provide a more accurate wind estimation of weather station data than the forecast would provide alone. The estimation provided, however, is not sufficiently accurate to be used in a "virtual" wind sensor system, and further significant improvements are needed before estimations with a sufficient accuracy can be attained. The large error likely emanates from the error when comparing the forecast data with the weather station data, as the weather station data is a location specific wind measurement, whereas the forecast data is a forecast for the wind over a large area, suggesting that the two values will vary based on the topography of the area surrounding the weather station. Random Forests clearly provide an improvement in the accuracy of the estimation, and, as could be expected, linear regression is a very poor estimator of both wind speed and wind direction, likely due to the non-linear nature of wind. It is clear that the Digital Elevation Model (DEM) does provide a significant improvement in the accuracy of the estimation, however, it is interesting that the 3×3 area of only the $90m$ DEM provides the most accurate estimation rather than the estimation that the 9×9 area, or multiple layers of different resolutions can provide, which suggests that a higher resolution DEM may provide further slight improvements to the accuracy of the estimation. Additionally, the best performing transform for both wind speed and wind direction when trained for all the weather stations is the regular DEM rather than the transforms of the DEM, such as the Topographic Positioning Index (TPI), which transform absolute height data into relative height information. Interestingly, when generalizing wind direction, Local Binary Patterns (LBPs) show the largest improvement in accuracy, and the Sobel operator provides the best accuracy when generalizing wind speed. Scaling the data only shows a significant improvement in the estimation accuracy when generalizing wind speed, it shows no significant change in accuracy when estimating wind speed and wind direction, with a small decrease in accuracy when generalizing wind direction.

It is possible that the similarity of results for estimating wind direction and wind speed for all weather stations is due to Random Forests overfitting the data rather than learning relations between the transformed topographic data and the weather station data. This is made slightly more evident in the large decrease in performance when attempting to generalize the wind data, especially when using the untransformed DEM to estimate wind speed, which previously provided the most accurate results when trained using all the data, provides significantly poorer results when generalizing wind speed. Unlike the use of DEM transforms, the effect of changing the area that is extracted from the DEM is similar when generalizing and estimating both wind speed and direction. Increasing the area that is extracted from the DEM shows a slight decrease in performance when estimating wind direction, with a slightly larger decrease when estimating wind speed, which may be related to the inaccuracies of the forecast data that is used to rotate the DEM, leading to topographic features with the incorrect orientation relative to the wind, where the number of these topographic features increases the larger the area that is extracted from the DEM.

Although this study shows the improvements gained by using weather forecast data with machine learning to estimate and generalize weather station wind measurements, the improvements provided by the machine learning system are not sufficient, especially in the case of generalizing wind data, and further research is needed before the estimations have the accuracy that would be required of a “virtual” wind sensor. The error in estimation and generalization may be improved by implementing alternative learning methods, or testing various parameters of the learning methods with a focus on optimizing the parameters. Incorporating historical data into the training system may allow the system better to learn the behaviour of wind over a period of time, although for “virtual” wind sensors, the historical data must be limited to non-weather station data. Obtaining higher resolution DEMs, such as a $30m$ DEM, may provide a further improvement with wind estimations. Combining multiple transforms or features, such as training the system using the unaltered DEM, the Sobel operator, LBPs and TPI for a 3×3 and a 9×9 from each of the 90m, 180m, 360m, 720m, and 1440m DEM, may allow the machine learning system to learn a relationship between different features and the wind measurements. Adding additional features may provide the greatest improvement in accuracy, although it potentially has the largest research scope when compared with the methods mentioned above to improve upon this study. The scope of generating additional features could include the following:

- adjusting parameters of existing features, by optimizing the parameters that the LBP uses to generate feature sets;
- comparing similar algorithms and determining which of the Prewitt, the Sobel and the Frei-Chen operators most improves wind estimation;
- implementing alternative image processing methods or topographical transformations; and even
- creating new methods to extract features from the data.

Creating a method to extract features from the data may increase the dimensionality of the data by incorporating the shape of the topography of the data into the feature, or a method that can reduce the surrounding topographic features into values, which can be applied to multiple layers of the DEM to provide a small collection of features. The most important thing to note when creating a new method to generate features is that given two similar inputs, the resulting sets of features should be similar in most aspects, whilst if two wildly different inputs are provided, the two resulting sets of features should not be similar.

This study does not provide any useful result in terms of applications for “virtual” wind sensors, however, it can serve as a base for future research to improve the availability of wind data through “virtual” wind sensors.

Appendix A

Dataset

A.1 ECMWF Dataset

The European Centre for Medium-Range Weather Forecasts (ECMWF) dataset is a low resolution weather prediction dataset. ECMWF data is generated from a model to forecast weather in a grid based system. Each forecast in the resulting grid will give a general weather forecast that covers a large area. Each forecast can contain any number of different variables; however, for the dataset, only the u and v components of the wind vector, average temperature, dewpoint temperature and surface pressure are used.

The data is collected through a continuous, paid subscription from EUMETCast. The range of data used covers a period of nine months, from 01-09-2013 to 02-06-2014. Forecast sets are generated at 12:00am and 12:00pm every day using the model. Each forecast set contains many forecasts at varying time intervals, including a forecast at zero hours from the time the forecast is generated. These zero hour forecasts are the only forecasts that are used from the forecast sets.

The data is forecast using an imperfect forecast model and as such some discrepancies in the data do occur. The model is constantly updated by the ECMWF using accurate measurements from weather stations to ensure the model improves and reduces these discrepancies. Although the accuracy of the data is not guaranteed, it can still be used as a basis for weather predictions [3].

The ECMWF dataset covers a large section of Southern Africa, ranging from 16°E to 33°E longitude and 22°S to 35°S latitude. The dataset is formed from a grid in this range with each point in the grid 0.125° apart along the longitudinal or latitudinal axis. This

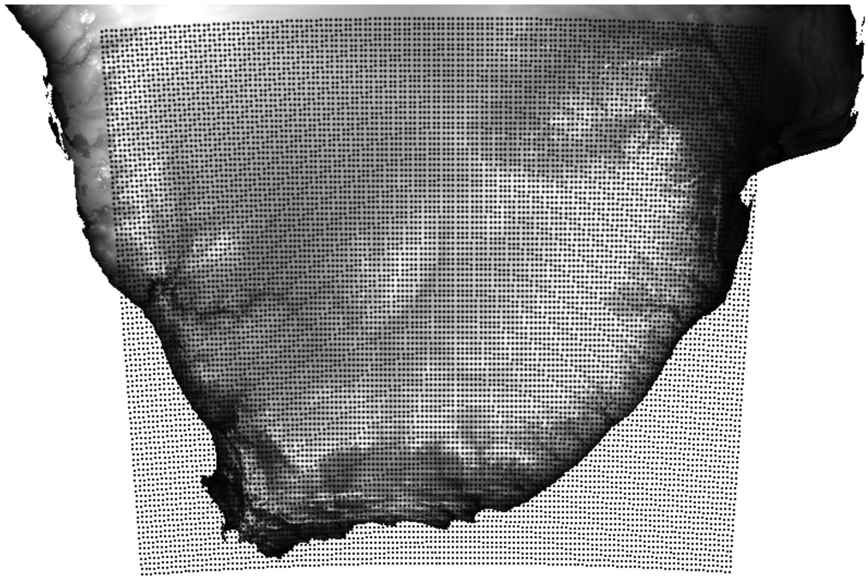


Figure A.1: A representation of the location of the ECMWF forecast location grid overlaid on the DEM.

results in a grid of 138×105 data points, over 542 time steps as can be seen overlaid on the DEM in figure A.1.

A.2 Weather Station Dataset

The weather station dataset is a sparse, high resolution dataset containing weather measurements from approximately 200 weather stations around South Africa. These measurements are accurate for the weather stations' locations. The dataset measures a few different weather variables, but only wind speed and wind direction are extracted from the dataset.

The measurements are measured hourly and collected from each of the stations using services provided by South African Weather Service (SAWS). The data that is extracted ranges from 01-09-2013 to 02-06-2014 as with the ECMWF data. The time periods of interest in this range are the time steps that correspond with the ECMWF dataset and so only measurements at 12:00am and 12:00pm for each day are considered.

Although the weather stations are scheduled to take measurements at certain times each day, each set of measurement data comes from independent stations and is not always received, or may be received with missing values. The distribution of data entries for each station and time point can be seen in figure A.2. Only the data entries with all the

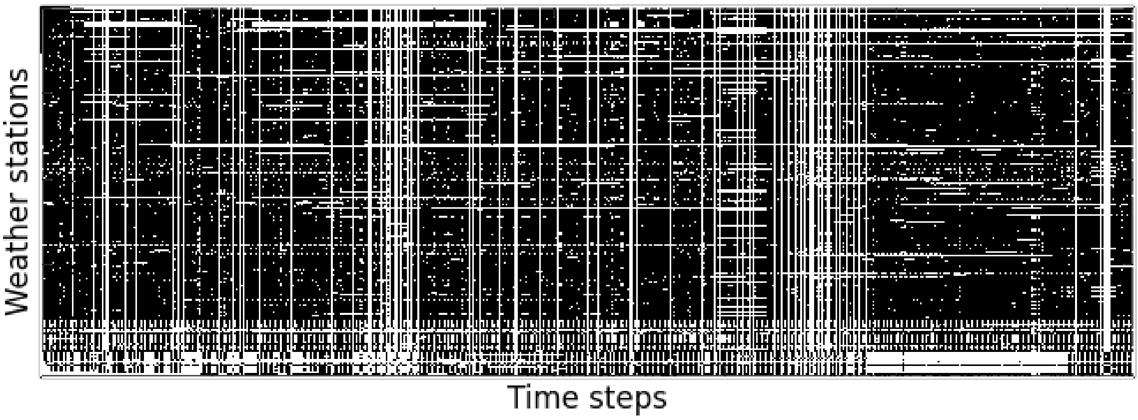


Figure A.2: A representation of the distribution of the weather station data points in relation to the measurement times. Each row of pixels represents the data for a single weather station, whilst each column of pixels represents the data for each unique time of measurement.

relevant values are included in the final dataset. In spite of the missing values, there are approximately 100000 weather station data entries in the final dataset.

The weather station dataset contains data from weather stations spread around South Africa, all of which are in the area covered by both the ECMWF dataset and the Digital Elevation Map. Figure A.3 shows the sparse distribution of weather stations over the region of interest.

A.3 Digital Elevation Map

A Digital Elevation Map (DEM) is a dataset that contains elevation data for the region of interest. The DEM is collected from the Shuttle Radar Topography Mission (SRTM) conducted by NASA, the National Geospatial-Intelligence Agency, and the German and Italian Space Agencies [2]. The DEM can be used to obtain various topographical features in the region and has support for various transforms for further features to be extracted.

There are varying resolutions available for DEMs. The resolution refers to the length and width of each pixel in the DEM. The DEM that was used has a resolution of 90 metres, however there are DEMs available with a higher resolution. The DEM is subject to some inaccuracies, although the data is still reliable as discussed in the study by Rodriguez et al. [4]. Despite the inaccuracies, the absolute height values are not important in this application, with the relative height changes being more meaningful.

The DEM covers the entirety of Southern Africa, covering the area ranging from 7°E

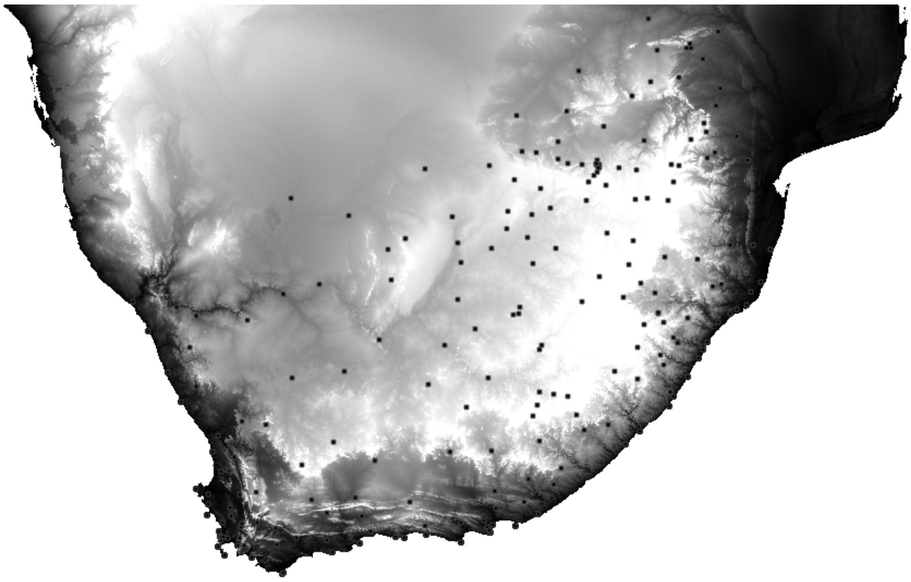


Figure A.3: The location of weather stations from which the dataset is acquired.

through 43°E longitude and 14°S through 35°S latitude as shown in figure A.4.

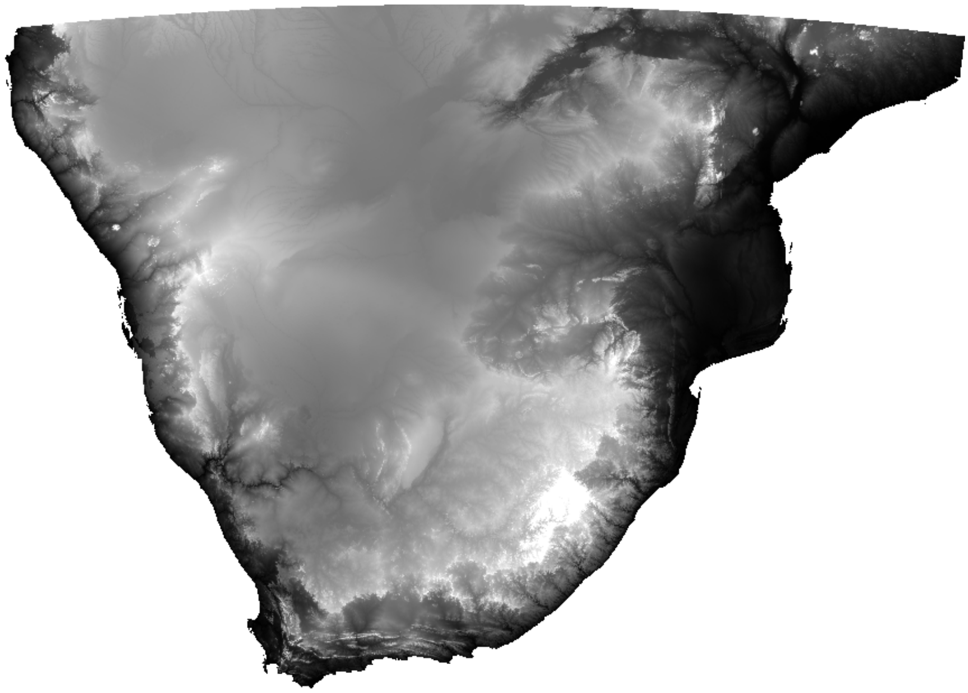


Figure A.4: An image generated from the DEM. Lighter areas have a higher elevation.

Appendix B

Resulting Data

The tables below contain the data that is represented in the graphs in section 4. Due to the large variety of experiments performed, many combinations of features have no results, leading to sparse tables. In the interest of readability, some sets of rows or columns were not included if no data is available for those rows or columns. The titles of the columns and rows have also been shortened, with the meanings of the shortened titles as follows:

- Unscaled – represents data which has not been scaled at all;
- Scaled – represents data where both the input and output are scaled;
- X-scaled – represents data where only the input data is scaled;
- RF – Random Forests were used;
- LR – linear regression was used;
- Layers – The number of layers used, with 1 layer representing that only the 90m version of the Digital Elevation Map (DEM) is used, and 5 layers representing that all of the 90m, 180m, 360m, 720m and 1440m versions of the DEM were used;
- Area – The size of the area extracted from the DEM;
- Format – Which transformations were performed on the DEM before the system is trained:
  - Forecast – Only the forecast data from the European Centre for Medium-Range Weather Forecasts (ECMWF) data is used. This represents the benchmark data for the results, and no training is performed on this data;

- No DEM – The system is trained only using the ECMWF data, with no data from the DEM;
- TPI – The DEM is transformed to provide the Topographical Positioning Index (TPI) of the data;
- TRI – The DEM is transformed to provide the Terrain Ruggedness Index (TRI) of the data;
- slope – slope shading is extracted from the DEM;
- DEM – No transform is made on the DEM;
- LBP – Local Binary Patterns (LBP) are extracted from the DEM;
- FFT – A Fast Fourier Transform (FFT) is performed on the DEM. Both the phase angle and the magnitude data is used;
- Sobel – The Sobel operator is applied to the DEM, and then this data is aggregated as explained in section 3.3.8.

B.1 Wind Direction Estimation Results

This section contains resulting data for all systems trained to estimate wind direction.

		Scaling Method	Unscaled		Scaled	X-Scaled
		Learning Method	RF	LR	RF	RF
Layers	Area	Format				
		Forecast	20.178			
		No DEM	15.761			
		TPI	12.306			
		TRI	11.859			
	3 × 3	slope	11.968			
		DEM	11.309	28.652	11.352	11.287
		LBP	12.266			
		FFT	14.885			
1		Sobel	12.335			
		No DEM				
		TPI				
		TRI	12.266			
		slope	12.673			
	9 × 9	DEM	11.245			
		LBP				
		FFT				
		Sobel				
		No DEM				
		TPI				
		TRI	11.951			
		slope	12.167			
	3 × 3	DEM	11.462		11.185	11.249
		LBP				
		FFT				
		Sobel				
5		No DEM				
		TPI				
		TRI				
		slope				
	9 × 9	DEM	11.624			
		LBP				
		FFT				
		Sobel				

Table B.1: Resulting absolute 25^{th} percentile data for wind direction estimation for all stations.

	Scaling Method		Unscaled		Scaled	X-Scaled
	Learning Method		RF	LR	RF	RF
Layers	Area	Format				
		Forecast	50.860			
		No DEM	42.942			
		TPI	34.217			
		TRI	33.571			
	3×3	slope	33.532			
		DEM	31.716	69.918	31.276	31.368
		LBP	33.652			
		FFT	39.475			
1		Sobel	33.786			
		No DEM				
		TPI				
		TRI	34.972			
		slope	35.559			
	9×9	DEM	32.140			
		LBP				
		FFT				
		Sobel				
		No DEM				
		TPI				
		TRI	33.774			
		slope	33.719			
	3×3	DEM	31.376		31.311	31.610
		LBP				
		FFT				
		Sobel				
5		No DEM				
		TPI				
		TRI				
		slope				
	9×9	DEM	32.822			
		LBP				
		FFT				
		Sobel				

Table B.2: Resulting absolute 50^{th} percentile data for wind direction estimation for all stations.

Layers	Area	Format	Unscaled RF	Unscaled LR	Scaled RF	X-Scaled RF
1	3×3	Forecast	106.524			
		No DEM	95.740			
		TPI	88.037			
		TRI	87.840			
		slope	87.380			
		DEM	83.192	123.319	82.508	81.097
		LBP	85.896			
		FFT	92.544			
		Sobel	87.145			
	9×9	No DEM				
		TPI				
		TRI	89.347			
		slope	89.639			
		DEM	83.577			
		LBP				
		FFT				
		Sobel				
5	3×3	No DEM				
		TPI				
		TRI	89.307			
		slope	87.764			
		DEM	81.845		82.522	83.198
		LBP				
		FFT				
		Sobel				
	9×9	No DEM				
		TPI				
		TRI				
		slope				
		DEM	84.408			
		LBP				
		FFT				
		Sobel				

Table B.3: Resulting absolute 75^{th} percentile data for wind direction estimation for all stations.

		Scaling Method	Unscaled		Scaled	X-Scaled
		Learning Method	RF	LR	RF	RF
Layers	Area	Format				
	3 × 3	Forecast	135.195			
		No DEM	126.525			
		TPI	122.280			
		TRI	121.716			
	3 × 3	slope	121.450			
		DEM	118.088	146.221	118.250	115.612
		LBP	121.311			
		FFT	126.395			
1		Sobel	121.871			
	9 × 9	No DEM				
		TPI				
		TRI	124.524			
		slope	124.456			
	9 × 9	DEM	119.461			
		LBP				
		FFT				
		Sobel				
	3 × 3	No DEM				
		TPI				
		TRI	123.399			
		slope	123.275			
	3 × 3	DEM	117.360		118.277	119.140
		LBP				
		FFT				
		Sobel				
5	9 × 9	No DEM				
		TPI				
		TRI				
		slope				
	9 × 9	DEM	120.827			
		LBP				
		FFT				
		Sobel				

Table B.4: Resulting absolute 85^{th} percentile data for wind direction estimation for all stations.

	Scaling Method		Unscaled		Scaled	X-Scaled
	Learning Method		RF	LR	RF	RF
Layers	Area	Format				
		Forecast	65.709			
		No DEM	59.509			
		TPI	54.271			
		TRI	53.994			
	3×3	slope	54.012			
		DEM	51.951	77.325	51.857	51.244
		LBP	53.647			
		FFT	57.675			
1		Sobel	54.005			
		No DEM				
		TPI				
		TRI	55.090			
		slope	55.500			
	9×9	DEM	52.404			
		LBP				
		FFT				
		Sobel				
		No DEM				
		TPI				
		TRI	54.475			
		slope	54.278			
	3×3	DEM	51.842		51.738	52.200
		LBP				
		FFT				
		Sobel				
5		No DEM				
		TPI				
		TRI				
		slope				
	9×9	DEM	53.093			
		LBP				
		FFT				
		Sobel				

Table B.5: Resulting absolute mean data for wind direction estimation for all stations.

		Scaling Method	Unscaled		Scaled	X-Scaled
		Learning Method	RF	LR	RF	RF
Layers	Area	Format				
		Forecast	80.851			
		No DEM	76.021			
		TPI	70.736			
		TRI	69.881			
	3 × 3	slope	70.114			
		DEM	68.642	86.152	69.254	69.047
		LBP	70.781			
		FFT	74.843			
1		Sobel	70.852			
		No DEM				
		TPI				
		TRI	70.786			
		slope	71.454			
	9 × 9	DEM	68.895			
		LBP				
		FFT				
		Sobel				
		No DEM				
		TPI				
		TRI	70.402			
		slope	70.771			
	3 × 3	DEM	69.168		68.677	68.925
		LBP				
		FFT				
		Sobel				
5		No DEM				
		TPI				
		TRI				
		slope				
	9 × 9	DEM	69.360			
		LBP				
		FFT				
		Sobel				

Table B.6: Resulting absolute tolerance data for wind direction estimation for all stations.

		Scaling Method Learning Method	Unscaled RF	Scaled RF	X-Scaled RF
Layers	Area	Format			
1	3×3	Forecast	20.178		19.938
		No DEM	24.602		
		TPI	21.726		
		TRI	22.927		
		slope	21.321		
		DEM	18.941		
		LBP	19.723		
		FFT	17.722		
		Sobel	20.303		
5	3×3	No DEM	20.123		
		TPI			
		TRI			
		slope			
		DEM			
		LBP			
		FFT			
		Sobel			

Table B.7: Resulting absolute 25^{th} percentile data for wind direction estimation for certain untrained stations.

		Scaling Method Learning Method	Unscaled RF	Scaled RF	X-Scaled RF
Layers	Area	Format			
1	3×3	Forecast	50.860		48.271
		No DEM	55.595		
		TPI	49.115		
		TRI	56.093		
		slope	50.858		
		DEM	46.527		
		LBP	47.638		
		FFT	44.290		
		Sobel	47.672		
5	3×3	No DEM	48.219		
		TPI			
		TRI			
		slope			
		DEM			
		LBP			
		FFT			
		Sobel			

Table B.8: Resulting absolute 50^{th} percentile data for wind direction estimation for certain untrained stations.

Scaling Method Learning Method			Unscaled RF	Scaled RF	X-Scaled RF
Layers	Area	Format			
1	3×3	Forecast	106.524		
		No DEM	104.251		
		TPI	100.725		
		TRI	113.055		
		slope	104.173		
		DEM	97.992		103.421
		LBP	98.433		
		FFT	97.236		
		Sobel	99.025		
5	3×3	No DEM	99.179		
		TPI			
		TRI			
		slope			
		DEM			
		LBP			
		FFT			
		Sobel			

Table B.9: Resulting absolute 75^{th} percentile data for wind direction estimation for certain untrained stations.

Scaling Method Learning Method			Unscaled RF	Scaled RF	X-Scaled RF
Layers	Area	Format			
1	3×3	Forecast	135.195		
		No DEM	131.207		
		TPI	129.680		
		TRI	139.345		
		slope	132.100		
		DEM	128.870		134.096
		LBP	126.959		
		FFT	128.480		
		Sobel	130.547		
5	3×3	No DEM	127.622		
		TPI			
		TRI			
		slope			
		DEM			
		LBP			
		FFT			
		Sobel			

Table B.10: Resulting absolute 85^{th} percentile data for wind direction estimation for certain untrained stations.

Scaling Method / Learning Method			Unscaled RF	Scaled RF	X-Scaled RF
Layers	Area	Format			
1	3×3	Forecast	65.709		
		No DEM	67.318		
		TPI	63.992		
		TRI	69.505		
		slope	65.241		
		DEM	62.068		64.243
		LBP	62.122		
		FFT	60.812		
		Sobel	62.874		
5	3×3	No DEM	62.768		
		TPI			
		TRI			
		slope			
		DEM			
		LBP			
		FFT			
		Sobel			

Table B.11: Resulting absolute mean data for wind direction estimation for certain untrained stations.

Scaling Method / Learning Method			Unscaled RF	Scaled RF	X-Scaled RF
Layers	Area	Format			
1	3×3	Forecast	80.851		
		No DEM	84.578		
		TPI	82.299		
		TRI	83.739		
		slope	81.761		
		DEM	79.705		80.546
		LBP	79.858		
		FFT	78.229		
		Sobel	81.355		
5	3×3	No DEM	80.481		
		TPI			
		TRI			
		slope			
		DEM			
		LBP			
		FFT			
		Sobel			

Table B.12: Resulting absolute tolerance data for wind direction estimation for certain untrained stations.

B.2 Wind Speed Estimation Results

This section contains the resulting data for all systems trained to estimate the magnitude of the wind.

		Scaling Method	Unscaled		Scaled	X-Scaled
		Learning Method	RF	LR	RF	RF
Layers	Area	Format				
1	3×3	Forecast	20.683			
		No DEM	15.614			
		TPI	13.188			
		TRI	13.038			
		slope	13.317			
		DEM	12.542	4795.2	12.63	12.713
		LBP	13.494			
		FFT	16.201			
		Sobel	13.705			
	9×9	No DEM				
		TPI	13.883			
		TRI				
		slope	13.522			
		DEM	12.826			
		LBP				
		FFT				
		Sobel				
5	3×3	No DEM				
		TPI	13.396			
		TRI				
		slope	13.110			
		DEM	12.803			
		LBP				
		FFT				
		Sobel				
	9×9	No DEM				
		TPI				
		TRI				
		slope				
		DEM	13.460			
		LBP				
		FFT				
		Sobel				

Table B.13: Resulting relative 25^{th} percentile data for wind magnitude estimation for all stations.

Layers	Area	Format	Unscaled RF	Unscaled LR	Scaled RF	X-Scaled RF
		Scaling Method	Unscaled		Scaled	X-Scaled
		Learning Method	RF	LR	RF	RF
1	3 × 3	Forecast	45.601			
		No DEM	33.613			
		TPI	29.388			
		TRI	29.255			
		slope	29.437			
		DEM	28.072	11175.7	28.193	28.521
		LBP	30.145			
		FFT	34.065			
		Sobel	30.346			
	9 × 9	No DEM				
		TPI	30.684			
		TRI				
		slope	30.210			
		DEM	28.798			
		LBP				
		FFT				
		Sobel				
5	3 × 3	No DEM				
		TPI	29.413			
		TRI				
		slope	29.537			
		DEM	28.709			
		LBP				
		FFT				
		Sobel				
	9 × 9	No DEM				
		TPI				
		TRI				
		slope				
		DEM	30.345			
		LBP				
		FFT				
		Sobel				

Table B.14: Resulting relative 50^{th} percentile data for wind magnitude estimation for all stations.

		Scaling Method	Unscaled		Scaled	X-Scaled
		Learning Method	RF	LR	RF	RF
Layers	Area	Format				
1	3×3	Forecast	85.262			
		No DEM	65.585			
		TPI	60.812			
		TRI	60.694			
		slope	60.907			
		DEM	57.792	22216.1	58.035	58.208
		LBP	61.289			
		FFT	70.924			
		Sobel	62.282			
	9×9	No DEM				
		TPI	63.870			
		TRI				
		slope	61.974			
		DEM	57.679			
		LBP				
		FFT				
		Sobel				
5	3×3	No DEM				
		TPI	60.345			
		TRI				
		slope	59.503			
		DEM	57.677			
		LBP				
		FFT				
		Sobel				
	9×9	No DEM				
		TPI				
		TRI				
		slope				
		DEM	59.987			
		LBP				
		FFT				
		Sobel				

Table B.15: Resulting relative 75^{th} percentile data for wind magnitude estimation for all stations.

		Scaling Method	Unscaled		Scaled	X-Scaled
		Learning Method	RF	LR	RF	RF
Layers	Area	Format				
1	3×3	Forecast	145.177			
		No DEM	111.931			
		TPI	103.614			
		TRI	105.383			
		slope	104.823			
		DEM	95.829	32836.6	95.290	94.767
		LBP	106.719			
		FFT	125.825			
		Sobel	110.099			
	9×9	No DEM				
		TPI	112.530			
		TRI				
		slope	108.849			
		DEM	98.354			
		LBP				
		FFT				
		Sobel				
5	3×3	No DEM				
		TPI	102.445			
		TRI				
		slope	99.999			
		DEM	97.470			
		LBP				
		FFT				
		Sobel				
	9×9	No DEM				
		TPI				
		TRI				
		slope				
		DEM	99.819			
		LBP				
		FFT				
		Sobel				

Table B.16: Resulting relative 85^{th} percentile data for wind magnitude estimation for all stations.

Layers	Area	Format	Unscaled RF	Unscaled LR	Scaled RF	X-Scaled RF
1	3 × 3	Forecast	91.208			
		No DEM	4692.8			
		TPI	5272.5			
		TRI	8620.7			
		slope	8644.1			
		DEM	6099.0	19244.5	7908.1	4161.4
		LBP	7584.6			
		FFT	9902.1			
		Sobel	9823.6			
	9 × 9	No DEM				
		TPI	11451.7			
		TRI				
		slope	9357.1			
		DEM	4742.4			
		LBP				
		FFT				
		Sobel				
5	3 × 3	No DEM				
		TPI	5853.1			
		TRI				
		slope	5891.4			
		DEM	9269.0			
		LBP				
		FFT				
		Sobel				
	9 × 9	No DEM				
		TPI				
		TRI				
		slope				
		DEM	4368.5			
		LBP				
		FFT				
		Sobel				

Table B.17: Resulting relative mean data for wind magnitude estimation for all stations.

		Scaling Method / Learning Method	Unscaled RF	X-Scaled RF
Layers	Area	Format		
1	3×3	Forecast	20.683	
		No DEM	18.007	
		TPI	19.004	
		TRI	21.033	
		slope	20.745	
		DEM	21.589	20.173
		LBP	18.701	
		FFT	18.460	
		Sobel	19.391	
5	3×3	No DEM		
		TPI		
		TRI		
		slope		
		DEM	20.191	
		LBP		
		FFT		
		Sobel		

Table B.18: Resulting relative 25^{th} percentile data for wind magnitude estimation for certain untrained stations.

		Scaling Method / Learning Method	Unscaled RF	X-Scaled RF
Layers	Area	Format		
1	3×3	Forecast	45.601	
		No DEM	38.753	
		TPI	41.398	
		TRI	43.662	
		slope	43.032	
		DEM	44.566	40.837
		LBP	39.810	
		FFT	37.985	
		Sobel	39.492	
5	3×3	No DEM		
		TPI		
		TRI		
		slope		
		DEM	44.045	
		LBP		
		FFT		
		Sobel		

Table B.19: Resulting relative 50^{th} percentile data for wind magnitude estimation for certain untrained stations.

		Scaling Method / Learning Method	Unscaled RF	X-Scaled RF
Layers	Area	Format		
1	3×3	Forecast	85.262	76.309
		No DEM	77.623	
		TPI	99.129	
		TRI	107.543	
		slope	100.379	
		DEM	103.218	
		LBP	82.920	
		FFT	86.939	
		Sobel	64.699	
5	3×3	No DEM	132.371	
		TPI		
		TRI		
		slope		
		DEM		
		LBP		
		FFT		
		Sobel		

Table B.20: Resulting relative 75^{th} percentile data for wind magnitude estimation for certain untrained stations.

		Scaling Method / Learning Method	Unscaled RF	X-Scaled RF
Layers	Area	Format		
1	3×3	Forecast	145.177	137.506
		No DEM	145.749	
		TPI	173.527	
		TRI	192.518	
		slope	202.375	
		DEM	244.982	
		LBP	154.051	
		FFT	165.623	
		Sobel	105.415	
5	3×3	No DEM	345.297	
		TPI		
		TRI		
		slope		
		DEM		
		LBP		
		FFT		
		Sobel		

Table B.21: Resulting relative 85^{th} percentile data for wind magnitude estimation for certain untrained stations.

	Scaling Method Learning Method		Unscaled RF	X-Scaled RF
Layers	Area	Format		
1	3×3	Forecast	91.208	19645.2
		No DEM	31453.9	
		TPI	9426.9	
		TRI	20721.4	
		slope	18025.5	
		DEM	41174.2	
		LBP	16599.0	
		FFT	18009.4	
		Sobel	3084.5	
5	3×3	No DEM	83565.5	
		TPI		
		TRI		
		slope		
		DEM		
		LBP		
		FFT		
		Sobel		

Table B.22: Resulting relative mean data for wind magnitude estimation for certain untrained stations.

Bibliography

[1] J. M. Forthofer, "Modeling wind in complex terrain for use in fire spread prediction," Ph.D. dissertation, Colorado State University, 2007.

[2] T. G. Farr, P. A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, L. Roth *et al.*, "The shuttle radar topography mission," *Reviews of geophysics*, vol. 45, no. 2, 2007.

[3] European Centre for Medium-Range Weather Forecasts. (1975) European centre for medium-range weather forecasts. [Online]. Available: http://www.ecmwf.int/

[4] E. Rodriguez, C. S. Morris, and J. E. Belz, "A global assessment of the srtm performance," *Photogrammetric Engineering and Remote Sensing*, vol. 72, no. 3, pp. 249–260, 2006.

[5] T. Haiden, M. Janousek, P. Bauer, J. Bidlot, L. Ferranti, T. Hewson, F. Prates, D. Richardson, and F. Vitart, "Evaluation of ecmwf forecasts, including 2013-2014 upgrades," 2014, not published.

[6] A. G. Abo-Khalil and D.-C. Lee, "Svr-based wind speed estimation for power control of wind energy generation system," in *Power Conversion Conference-Nagoya, 2007. PCC'07.* IEEE, 2007, pp. 1431–1436.

[7] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, and Z. Yan, "A review on the forecasting of wind speed and generated power," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 4, pp. 915–920, 2009.

[8] M. Mohandes, T. Halawani, S. Rehman, and A. A. Hussain, "Support vector machines for wind speed prediction," *Renewable Energy*, vol. 29, no. 6, pp. 939–947, 2004.

[9] Y. Hu, K. Stamnes, M. Vaughan, J. Pelon, C. Weimer, D. Wu, M. Cisewski, W. Sun, P. Yang, B. Lin *et al.*, "Sea surface wind speed estimation from space-based lidar measurements," *Atmospheric Chemistry and Physics*, vol. 8, no. 13, pp. 3593–3601, 2008.

[10] D. R. Thompson and R. C. Beal, "Mapping high-resolution wind fields using synthetic aperture radar," *Johns Hopkins APL Technical Digest*, vol. 21, no. 1, pp. 58–67, 2000.

[11] U. F. Service, "Windninja: A computer program for spatially computing wind fields for wildland fore applications." [Online]. Available: http://www.firelab.org/project/windninja

[12] M. Brown, "A gust changes everything: Local wind information in unprecedented detail with windwizard and windninja," *JFSP Briefs*, 2010.

[13] C. T. Allen, G. S. Young, and S. E. Haupt, "Improving pollutant source characterization by better estimating wind direction with a genetic algorithm," *Atmospheric Environment*, vol. 41, no. 11, pp. 2283–2289, 2007.

[14] O. Antonić and T. Legović, "Estimating the direction of an unknown air pollution source using a digital elevation model and a sample of deposition," *Ecological modelling*, vol. 124, no. 1, pp. 85–95, 1999.

[15] S. Robert, L. Foresti, and M. Kanevski, "Spatial prediction of monthly wind speeds in complex terrain with adaptive general regression neural networks," *International Journal of Climatology*, vol. 33, no. 7, pp. 1793–1804, 2013.

[16] P. Raybaut. (2008) Python(x,y). [Online]. Available: https://code.google.com/p/pythonxy/

[17] Free Software Foundation, Inc. (2007) Gnu general public license version 3. [Online]. Available: http://www.gnu.org/licenses/gpl.html

[18] F. Alted, I. Vilata *et al.*, "PyTables: Hierarchical datasets in Python," 2002–. [Online]. Available: http://www.pytables.org/

[19] GDAL Development Team, *GDAL - Geospatial Data Abstraction Library, Version x.x.x*, Open Source Geospatial Foundation, 201x. [Online]. Available: âĂŃhttp://www.gdal.org

[20] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online]. Available: http://www.scipy.org/

[21] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 6 2014. [Online]. Available: http://dx.doi.org/10.7717/peerj.453

[22] A. Weiss, "Topographic position and landforms analysis," in *Poster presentation, ESRI User Conference, San Diego, CA*, 2001, pp. 200–200.

[23] M. F. Wilson, B. O'Connell, C. Brown, J. C. Guinan, and A. J. Grehan, "Multiscale terrain analysis of multibeam bathymetry data for habitat mapping on the continental slope," *Marine Geodesy*, vol. 30, no. 1-2, pp. 3–35, 2007.

[24] P. Valentine, S. Fuller, and L. Scully, "Terrain ruggedness analysis and distribution of boulder ridges and bedrock outcrops in the stellwagen bank national marine sanctuary region-seabed ruggedness: Us geological survey poster," *URL: http://woodshole. er. usgs. gov/project-pages/stellwagen/posters/rugged. html*, 2004.

[25] S. J. Riley, "Index that quantifies topographic heterogeneity," *Intermountain Journal of Sciences*, vol. 5, no. 1-4, pp. 23–27, 1999.

[26] B. K. Horn, "Hill shading and the reflectance map," *Proceedings of the IEEE*, vol. 69, no. 1, pp. 14–47, 1981.

[27] M. Shapiro and O. Waupotitsch, "GRASS GIS manual: r.slope.aspect," 2011–. [Online]. Available: http://grass.osgeo.org/grass64/manuals/r.slope.aspect.html

[28] S. Warren, M. Hohmann, K. Auerswald, and H. Mitasova, "An evaluation of methods to determine slope using digital elevation data," *Catena*, vol. 58, no. 3, pp. 215–233, 2004.

[29] W. K. Pratt, *Digital image processing: PIKS Inside*, 3rd ed. John Wiley and Sons Inc, 2001.

[30] M. Pietikäinen, "Local binary patterns," *Scholarpedia*, vol. 5, no. 3, p. 9775, 2010.

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

[32] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Li, *Applied linear statistical models*. McGraw-Hill Irwin, 2005, vol. 5.

[33] F. Hayashi, "Econometrics. 2000," *Princeton University Press. Section*, vol. 1, pp. 60–69, 2000.

[34] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[35] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 43, no. 3, pp. 492–501, 2005.

[36] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, "Random forests for land cover classification," *Pattern Recognition Letters*, vol. 27, no. 4, pp. 294–300, 2006.

[37] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[38] D. Maturana, D. Mery, and A. Soto, "Face recognition with local binary patterns, spatial pyramid histograms and naive bayes nearest neighbor classification," in *Chilean Computer Science Society (SCCC), 2009 International Conference of the*. IEEE, 2009, pp. 125–132.

[39] N. Cheney, J. Gould, and W. Catchpole, "The influence of fuel, weather and fire shape variables on fire-spread in grasslands," *International Journal of Wildland Fire*, vol. 3, no. 1, pp. 31–44, 1993.