Final report Jaap de Boer Dieuwer Hondelink Jip Rietveld Rolf de Vries

Estimatic Bachelor End Project: Final report

by



to obtain the degree of Bachelor of Science at the Delft University of Technology, to be defended publicly on Wednesday July 3, 2019 at 10:00 AM.

Student names:J.E. de Boer
D.A. Hondelink
J.A.C. Rietveld
R.B. de VriesProject duration:May 2019 – July 2019Thesis committee:Dr. Ir. A. Bozzon, TU Delft, supervisor
Ir. O.W. Visser, TU Delft
Dr. Ir. H. Wang, TU Delft
Ir. R. Janssen, AerLabs BV

An electronic version of this thesis is available at http://repository.tudelft.nl/.



Preface

This thesis is the end product of four Computer Science and Engineering students, studying together for 4 years. Throughout these years, we have often (not) worked, together, resulting in camaraderie beyond the college benches. To be able to end this Bachelor study with the same group that we started with has been a great pleasure.

We would like to thank Alessandro Bozzon for guiding us during this project. Furthermore we would like to thank Maarten Tielrooj from To70 for his knowledge on SKV and AFOS, to help us establish the success criteria for AFOS and help us validate our findings. We would like to thank MVRDV for allowing us to use their image as the cover for this thesis. We would like to thank the people at the KNMI for their time and for supporting us with their expert knowledge. Finally, we would like to thank Richard Janssen, and AerLabs as a whole, for providing us with guidance, feedback and a great workplace throughout the project.

Jaap de Boer Dieuwer Hondelink Jip Rietveld Rolf de Vries Delft, June 2019

Abstract

Amsterdam Airport Schiphol has 5 runways, each of which can be used for take-off or landing of aeroplanes. The weather heavily influences which runway configuration air traffic control might pick. Airport Forecasting Service (AFOS) predicts which configuration of runways works most efficiently given a set of expected weather conditions and the standard deviations of wind components. These standard deviations give the system an indication of the accuracy of the weather forecasts.

Currently, the KNMI (Royal Netherlands Meteorological Institute) is the only meteorological institute that provides these standard deviations along with the weather forecast. This raises the main research question of this report: *Is it possible to make accurate enough estimations of the standard deviation of wind direction and wind speed using historical data and future weather expectations*? Estimating these standard deviations has been researched with two different approaches: a statistical method approach and a machine learning approach.

Statistical methods Four fitting methods have been researched in search of the best statistical model to estimate the standard deviation of wind direction and speed: the Maximum Likelihood Method (MLM) and three Least Square Method implementations of a Weibull, Minimum Weibull and Double Weibull distribution. The performance of aggregates on the outcome of these four methods was also researched. One case takes the minimum standard deviation of the four, the other takes the mean.

MLM not only performs the best but also performs most consistently of the four fitting methods. Taking into account aggregates, MLM is more consistent than the minimum method but the minimum method outperforms it. Neither of these methods managed to meet the success criteria.

Machine Learning In regards to machine learning, the problem of estimating the standard deviations of wind direction and wind speed is a regression problem. The following machine learning models have been researched for Estimatic: MLPN, LSTM RNN, ERNN and RBFN.

LSTM RNNs outperform MLPNs, RBFNs and ERNNs for both wind direction and speed standard deviation estimation. LSTM RNN performance did not meet the success criteria.

The research concludes that it is not possible to make accurate enough estimations of the standard deviation of wind components using the historical data and future weather expectations available for Amsterdam Airport Schiphol.

Contents

1	Introduction	1
2	Related Works 2.1 Weather forecasting with machine learning 2.2 Time series analysis 2.3 Statistical models for predicting standard deviations 2.3.1 Mardia (1972) 2.3.2 Yamartino (1984) 2.3.3 Linear standard deviation 2.3.4 Weibull distribution 2.4 Other related works	3 3 4 4 4 4 4 4 4
3	Problem Analysis 3.1 Current situation 3.2 Data 3.3 Research questions 3.4 Hypotheses	5 7 8 9
4	Methodology 4.1 Statistical methods 4.2 Machine Learning 4.3 Success criteria 4.4 Software 4.4.1 Quality control and testing 4.4.2 Model training. 4.4.3 SIG	11 12 16 16 17 17
5	Results 5.1 Statistical methods 5.2 Machine Learning 5.3 Data limitations	19 19 23 28
6	Discussion 6.1 Statistical methods 6.1.1 Issues with the Weibull method 6.2 Machine Learning 6.3 Data limitations 6.4 Estimatic 6.5 Ethical implications	31 32 32 34 34 34
7	Conclusion & Recommendations 7.1 Conclusion 7.2 Recommendations 7.2.1 Future studies 7.2.2 Recommendations for To70	37 37 37 37 38
8	Lessons Learned	39
Α	Glossary	41
в	Task Division	43
С	Research Report	45

61

65

67 69

71

D	Machine Learning Results
Е	Specification of Sets of Input
F	SIG Feedback

G	Project Info Sheet	
-		

	1		
RID	liog	jrap	ny

Introduction

Amsterdam Airport Schiphol has 5 runways, each of which can be used for take-off or landing of aeroplanes, and these aeroplanes can land and take off in both directions of the runway. Choosing which configuration of runways to use is an essential part of the job for air traffic control. The weather heavily influences which runway configuration air traffic control might pick. Generally, aeroplanes want to land and take off facing into the wind.

To help air traffic control decide which configuration works best, aviation consultant To70 has created a support system called AFOS (Airport Forecasting Service). AFOS predicts which configuration of runways works most efficiently given a set of expected weather conditions. As input, AFOS requires weather predictions for up to 31 hours into the future. Furthermore, to provide the system with information on the accuracy of the wind direction and wind speed estimations, the standard deviations of these components are provided as well.

KNMI (Royal Netherlands Meteorological Institute) is one of the few meteorological institute that provides these standard deviations along with the weather expectations. To70 wants to be able to sell AFOS to airports outside of the Netherlands. These airports have access to the weather forecasts for their location as well as historical measurements, but not the aforementioned standard deviations. This leads to the main research question of this study: *Is it possible to make accurate enough estimations of the standard deviation of wind direction and wind speed using historical data and future weather expectations?*. This historical data for Schiphol is provided by KNMI databases. Estimating these standard deviations has been done with two different approaches: statistical method approach and a machine learning model approach. The outcome of the research of these approaches has resulted in the product called Estimatic.

This bachelor end project was commissioned by AerLabs, an aviation software company, in collaboration with To70.

The structure of the rest of this report will be as follows: In chapter 2 related works will be discussed. In chapter 3 the problem and available data is analysed, and the methodology is presented in chapter 4. In chapter 5 the found results are listed, and discussed in chapter 6. Chapter 7 presents the conclusion to the research question, after which recommendations for future studies and for To70 are made. The lessons learned during this project are in chapter 8.

\sum

Related Works

There have not been any studies as of yet that research estimating the standard deviations of wind speed and wind direction (these are hereafter called wind components) based on historical data. However, there have been studies on estimating the weather using machine learning, as well as studies using statistical methods that calculate standard deviations based on measurements.

2.1. Weather forecasting with machine learning

A study by Maqsood et al. [9] researched how well different machine learning models performed when forecast weather. To forecast weather, they made use of temperature, wind speed and relative humidity data for forecast 24 hours ahead. The data consisted of hourly measurements, starting from December 1st, 2000, up to November 9th, 2001. The data was split into seasons. In this study, they used ensembles of four different models: multilayer perceptron networks (MLPN), radial basis function network (RBFN), Elman recurrent neural networks (ERNN) and Hopfield models (HFM). They found that RBFNs perform best, both in terms of accuracy and training time. Compared to the other networks, RBFNs correlate better. After RBFNs, ERNNs performed best in terms of accuracy. The dynamic behaviour of the weather was better captured by ERNNs than MLPNs. ERNNs, a type of recurrent neural networks, can show temporal dynamic behaviour of the weather, resulting in a more compact and natural representation of the temporal information contained in the weather profile." MLPNs performed second worst both in terms of accuracy and training speed. HFMs performed worst in terms of accuracy, but performed best in terms of training speed, taking only a few seconds where ERNNs and MLPNs needed up to 30 minutes.

In addition to ensembles of single models, Maqsood et al. created ensembles of combinations of aforementioned models. These combination ensembles were created with two approaches, winner takes-all (WTA) and weighted average (WA). The WTA and WA ensembles were compared with the single model ensembles, and both were found to predict the weather more accurately. In terms of error, the WTA ensemble method performed the best in predicting weather.

2.2. Time series analysis

A study by Kourentzes et al. [7] researched the performance of a neural network ensemble for time series forecasting. They propose a neural network mode operator ensemble and compare its performance with existing fundamental operators; mean and median. They empirically evaluated the performance of the ensemble and the operators using two datasets containing monthly time series. From both datasets, samples that contained 108 or more observations (9 years) were chosen. On choosing the samples, they stated: "Long time series were preferred to allow for adequate training, validation and test sets." The results indicated that the neural network ensemble can accurately forecast time series. They propose a number of applications that can benefit from neural networks ensemble forecasts, such as climate modelling.

2.3. Statistical models for predicting standard deviations

Several methods for calculating the standard deviation of wind speed and direction have been studied in the past. Most of them calculate the standard deviation based on the minutely measurements taken during the previous hour.

2.3.1. Mardia (1972)

In the book "Statistics of Directional Data" Mardia (cited in Mori [10]) discusses a single-pass method to calculate the standard deviation of wind direction. Mori [10] concludes that Mardia's method is one of the best single-pass estimators of σ_d in practical application. So does Weber [13] in his study in which Mardia's method has an RMSE of only 2.9 degrees.

2.3.2. Yamartino (1984)

Yamartino[15] has researched his single-pass method extensively, comparing its results to the classically used two-pass method to determine standard deviation of wind direction. In his research he arrives at three different estimators for the standard deviation. The first two estimators were not sufficient in handling standard deviations larger than 90°. However, the third one was able to accurately estimate a standard deviation with a degree of $\pm 2\%$ (compared with the classical two-pass method).

Even though this method has not been tested on field data it does solve issues presented by Ackermann [1] and Verrall and Williams [12]. Where the former over predicts for values above 40° and the latter under predicts. Yamartino's method does not have these issues.

Weber [13] shows the Yamartino method to have an RMSE of only 1.9 degrees. Turner [2] confirms Yamartino's method as the best method when compared with Verrall and Williams [12] and Ackermann [1].

2.3.3. Linear standard deviation

Multiple papers start off with remarks regarding the usual linear estimation of the standard deviation of an estimate. Turner [2] explains that due to the circular function of wind direction (360° and 0° are the same) this standard statistical method cannot be used. Ackermann [1], Weber [13] and Mori [10] all confirm that this discontinuity means this method will not be of use for determining the standard deviation of wind direction.

Regarding wind speed, however, using this method to determine standard deviation could be of some use. Although this model does not handle real-world phenomena it could be of use for giving a general estimation.

2.3.4. Weibull distribution

Multiple studies have shown that a Weibull Distribution often correctly describes the wind speed distribution [3] [4] [5]. This two-parameter distribution takes a parameter k and C, with k > 0 being the so-called shape-parameter and C > 0 being the scale parameter.

All three of the above studies have determined different methods to determine parameters k and C. In a study conducted by Kaoga et al. [5] the Maximum Likelihood Method (MLM), which uses time series to estimate the values of k and C and the Energy Pattern Factor method (EPF), which relates to the average data of wind speed were researched. These are most relevant to the dataset. EPF was one of the worse performing parameter estimators, but MLM was recommended by Kaoga et al. as an alternative. With an error of 30.10% when estimating standard deviations, it was the second best performing method in the study conducted by Kaoga et al. [5].

2.4. Other related works

Further studies were conducted on the subjects above. A more comprehensive review of multiple studies can be found in the research report in Appendix C. In said report a deeper analysis on the paragraphs lined out above can be found, as well as the formulas necessary to execute these methods.

3

Problem Analysis

Is it possible to estimate the standard deviations of wind speed and wind direction given historical weather measurements and future weather forecasts? Questions that arise are: What is an accurate prediction? What kind of historical data can be expected? The following chapter sheds light on these questions by analysing the problem further.

3.1. Current situation

One of Amsterdam Airport Schiphol's air traffic control's tasks is to choose a runway configuration based on future weather conditions. AFOS (Airport Forecasting Service) is a support system for air traffic control at Amsterdam Airport Schiphol. As input, AFOS requires weather predictions for up to 31 hours into the future. Along with the weather predictions, the standard deviations of the wind direction and speed are provided as well. These parameters are essential to AFOS, as they give the system an indication of the accuracy of the weather forecasts. As output, it suggests air traffic control configurations based on their probability of being able to manage the estimated flight capacity.

According to To70, most meteorological institutes do not offer the standard deviations of the wind direction and speed along with their weather forecasts. The KNMI does offer this information in their weather forecast. The KNMI has a model called EPS (Ensemble Prediction System), which does offer these standard deviations. In the so-called expert-plume in figure 3.1 can be seen how the model creates 52 individual runs. By analysing the spread of these individual runs, the KNMI has an indication of the certainty of their weather forecast¹.

The specifics of the model, as well as the inputs of the different runs, are not known. What is known is that all 52 runs make use of a single model but modify the input slightly. Finally, it is known that the KNMI does not take this model as final truth. A meteorologist analyses the results of the model, and makes adjustments where necessary, based on his or her expertise.

In figure 3.2 the so-called 50% and 90% band in which 50% and 90% of the total runs fall in to can be seen. Based on these plumes the meteorologist will assign a value to the standard deviation of the wind direction or wind speed. These values of the standard deviation are then used as input for AFOS, as accuracy metrics for the weather forecasts.

In contrast to the KNMI plume, the weather forecasts made by other meteorological institutes lack a standard deviation. In figure 3.3 a wind forecast made by the KMI (Royal Meteorological Institute) of Belgium can be found. The wind forecast is the only thing presented in their figure, and no plume-like structure is offered.

¹Source: https://www.knmi.nl/kennis-en-datacentrum/achtergrond/over-de-weer-en-klimaatpluim-en-expertpluim



Figure 3.1: Expert plume for wind speed Source: www.knmi.nl



Figure 3.2: Standard plume for wind speed Source: www.knmi.nl

Verwachtingen 14 dagen

Bekijk de verwachtingen in detail (🔿



Figure 3.3: Plume for wind speed and direction offered by the Belgium Royal Meteorological Institute Source: www.meteo.be

Wind direction [°clockwise from north]Dew pointWind direction standard deviation [°]Prob. restrictionWind speed [kt]Prob. restrictionWind speed standard deviation [kt]Prob. restrictionGusts [kt]Issue timeTemperature [°C]Time of	int temperature [°C] egarding visibility conditions egarding cumulonimbus clouds and snow egarding thunderstorms and freezing rain me prediction

Table 3.1: Overview of information contained in the SKV dataset

3.2. Data

SKV With the output of EPS, the KNMI offers a weather forecast along with the aforementioned standard deviations, called SKV (Schiphol Probability Estimation). To70 has offered historical SKV data to train and validate Estimatic. This dataset contains hourly historical weather forecasts up to 31 hours into the future from a given point in time, specifically for Schiphol. Every hour, 15 forecasts are made up to 31 hours into the future, where for each forecast t_x , the time difference $x \in \{1, 2, 3, 4, 5, 6, 7, 10, 13, 16, 19, 22, 25, 28, 31\}$. The dataset ranges from 2010 to 2017. A detailed overview of all the information contained within the SKV dataset can be found in table 3.1.

KNMI Besides having the historical weather forecasts offered by the SKV dataset, there are multiple databases which offer historical *measurements* of weather i.e. the actual weather conditions on a given time and place. Most country specific meteorological institutes offer such databases. Examples of the meteorological institutes are the National Oceanic and Atmospheric Administration (NOAA) of the United States of America², the Meteorological Administration of China³ and the Royal Netherlands Meteorological Institute (KNMI)⁴. Such climatic agencies often offer open-to-public datasets containing e.g. land-based weather station data. These stations often report on the temperature, wind speed, wind direction, humidity and other weather components. Measurements are often available on a monthly, daily, hourly or even minute by minute basis.

The SKV dataset contains hourly weather forecasts between **2010 and 2017 for Schiphol**. A dataset matching the same time range and location of **measured weather conditions** is required. The

²https://www.ncdc.noaa.gov/

³http://www.cma.gov.cn/

⁴https://data.knmi.nl/datasets

Meaning	Units
Temperature	0.1° C
Highest gust measured in the last hour	0.1 m/s
Average wind direction over the last 10 minutes	° clockwise from north
Average wind speed over the last hour	0.1 m/s
Current dew point temperature at 1.50m	0.1° C
Average wind speed over the last 10 minutes	0.1 m/s
Current air pressure reduced to sea level	0.1 hPa
Total solar irradiance over the last hour	J/cm ²
Cloudiness	Expressed in range 1-8
Mist in the last hour	1 if mist occurred, 0 otherwise
Sum of rain in the last hour	in 0.1 mm

Table 3.2: Explanation of the KNMI data labels

KNMI Data Centre offers such a dataset⁵. This dataset contains hourly averaged weather information for Schiphol. An overview of the relevant information and explanation of the abbreviations contained within the KNMI dataset can be found in table 3.2.

3.3. Research questions

The main research question is: *Is it possible to make accurate enough estimations of the standard deviation of wind direction and wind speed using historical data and future weather expectations?* A detailed explanation of "accurate enough" is defined in section 4.3.

In section 2 it was found that statistical methods and machine learning are able to estimate standard deviations of the wind components. These two approaches are used to answer the main research question. Researching how to make the most accurate estimations lead to the formulation of research questions for each topic:

Statistical methods The following sub research questions will aid in the research to find if statistical methods can offer accurate enough estimations of the standard deviation of wind speed. In section 2 no statistical method was found to predict the standard deviation of wind direction. Therefore, this possibility is left outside of the scope of this project.

- RQ₁: What fitting methods work best in the statistical method?
- RQ₂: Which set of inputs work best in the statistical method?

Machine Learning The following sub research questions will aid in the research to find if machine learning can offer accurate enough estimations of the standard deviation of wind direction and wind speed.

- RQ₃: Which machine learning model performs best at estimating the standard deviations of wind components?
- RQ₄: Which data features work best as input for the machine learning models?
- RQ5: How does increasing the amount of lagged observations as input influence the accuracy?
- RQ₆: What happens with the accuracy as an estimation is made further into the future?

Data limitations Another question arose regarding Estimatic and its implementation for To70. In order to find out how much data To70 would need to train Estimatic, the research question below was formulated. As the statistical methods do not train on the data, this question is only relevant for the machine learning approach.

RQ7: What is the influence of the amount of years of training data on the accuracy of predictions made?

⁵https://www.knmi.nl/nederland-nu/klimatologie/uurgegevens

The definition of the best model is the model with the lowest *mean absolute error*. This is further explained in section 4.3.

3.4. Hypotheses

The following hypotheses are proposed for each research questions:

H1 Using a Maximum Likelihood Method (MLM) to fit a Weibull distribution to wind speed data results in the lowest mean absolute error. Kaoga et al. [5] have found that the MLM method outperformed other methods Estimatic could implement.

H2 Meteorologists from the KNMI were contacted for their expertise on filtering input data. Marco van den Berge and Adri Huiskamp suggested sets of inputs, these sets can be found in table 4.1. As Marco van den Berge's set of inputs contains more features, it is possible the filter is more strict than that of Adri. Therefore, Marco van den Berge's set is believed to work best.

H3 Recurrent neural network models work best, because they work especially well on time series data, which fits the data perfectly.

H4 Meteorologists from the KNMI were contacted for their expertise on filtering input data. As Marco van den Berge's set of inputs contains more features, it is possible for the machine learning models to learn more patterns in the data. Therefore, Marco van den Berge's set is believed to work best.

H5 Increasing the amount of lagged observations leading up to a prediction increases the accuracy, up to a sensible amount. After this, there will no longer be a correlation between all measurements and the estimation.

H6 The accuracy of estimations decreases as they are made further into the future, as the weather forecast used as input becomes more uncertain the further it is made into the future. This uncertainty will affect the accuracy.

H7 More years of training data increases the accuracy of predictions, as having more data allows models to learn patterns that happen over a greater period of time. Some years can have events that did not happen in other years, such as heatwaves. Therefore, more data will increase the accuracy.



Methodology

The following section will explain which method is used to implement the statistical and machine learning approaches. This chapter has been written instead of a "design and implementation" chapter as the main implementation of the research was decided here.

4.1. Statistical methods

Weibull In section 2.3.4 multiple sources [3] [4] [5] confirm that a Weibull distribution represents the distribution of wind speed best. Fitting a distribution to the entire historical data will result in only one calculated standard deviation, regardless of the weather forecast. Filtering on similar weather conditions of the weather forecast provides better insight into the wind speed distributions of those conditions.

Figure 4.1 shows the method taken to make a prediction using the Weibull method. First, the filter takes historic data and filters it on a window around the weather forecast. The next step fits a Weibull distribution on that data. With the Weibull distribution the standard deviation is calculated. An example illustrates this process:

Example The dataset consists of 8 years of hourly measured weather data. Estimatic wants to know what standard deviation it can expect for the current weather forecast. That weather forecast contains the following: a wind direction of 270 degrees, a wind speed of 7 knots and a temperature of 20 degrees Celsius. It takes all historic data points and filters them on similar measured values. So it takes all historic data which had a wind direction between 260 and 280 degrees, as well as 17 and 23 degrees Celsius. Fitting a Weibull distribution to the filtered data it then calculates the standard deviation of said distribution.



Figure 4.1: Methodology to make a prediction using Weibull

Marco van den Berge	Adri Huiskamp
Wind direction Avg. wind speed (hourly) Avg. wind speed (last 10 minutes) Temperature Solar irradiation	Wind direction Avg. wind speed (hourly) Avg. wind speed (last 10 minutes) Difference temp. at 150 cm and 10 cm Solar irradiation
Air pressure Cloudiness Mint	Cloudiness Ratio between gusts and avg. wind speed
MIST	

Table 4.1: Set of inputs as suggested by KNMI meteorologists

The research questions regarding statistical models concern the decisions of the example above. RQ_1 researches multiple fitting methods. RQ_2 researches different methods of filtering data. Below is the methodology of the approach taken to answer these individual research questions.

Fitting In section 2.3.4 it has been found that two different methods to fit a Weibull distribution to a dataset could be used: Maximum Likelihood Method (MLM) and Energy Pattern Factor (EPF). Estimatic implements the MLM method because Kaoga et. al [5] did not find that EPF improved their results and suggested MLM as a better alternative.

SciPy¹ is a scientific Python library. SciPy offers three different implementations of a Weibull distribution: Weibull, Double Weibull and Minimum Weibull. SciPy implements a Least Square Method to fit these distributions to the data. These fitting methods are used besides the MLM method to research the performance of the Weibull method. Finally the research explores the performance of aggregates on the outcome of these four methods. One case takes the minimum standard deviation of the four, the other takes the mean.

The best performing methods have been selected to conduct further research on the input space. The mean absolute error between the standard deviation returned by the fitting method and the actual standard deviation has been used to approach the best fitting method.

Filtering Choosing the right parameters to filter on, as well as the window to filter on, is something that requires expert knowledge. This expertise was not available in-house, so meteorologists were contacted to give insight regarding these filtering options. Table 4.1 shows the two methods. The Weibull method uses all parameters but wind speed itself as input to filter on. The exact window sizes can be found in appendix E.

The wind speed is not used to filter the data as that would always skew the distribution to the size of the window it was filtered on.

The mean absolute error is used as an indicator of performance of the various configurations of methods.

4.2. Machine Learning

The problem of estimating the standard deviations of wind direction and wind speed is a regression problem and can therefore also be approached as a machine learning task. In regression problems one tries to approximate the function F(x) = Y, where x is the input and Y the required output. With respect to Estimatic, Y is the standard deviation of either the wind direction or wind speed. As the regression model is trained using historical weather data the problem is a supervised learning task, and therefore Y is the label to be trained on.

There are different machine learning frameworks available that suit the needs of Estimatic. There are also possibilities to create custom made frameworks using environments like MATLAB or Python. In the research report (see appendix C) the use of Keras in combination with the popular Tensorflow framework was discussed. This abstracts most of the complexity of creating and training machine learning models. This allows for fast and easy prototyping and model creation. However, as Keras did not

¹https://www.scipy.org/

Parameter	Value
Epochs	50
K-Fold splits	5
Learning rate	0.001
Optimiser	Adam
Regularisation term (L1)	0.001
Loss function	mean absolute error

Table 4.2: Constant values for machine learning models

have all the models available that Estimatic implements, it was decided to also use MATLAB together with its built-in machine learning toolbox, thus making comparisons of these frameworks possible.

Models Machine learning models come in varying shapes and sizes, each with their own advantages and disadvantages. As can be found in section 2, Maqsood et al. [9] used four different machine learning models in order to forecast weather in southern Canada, namely multilayered perceptron network (MLPN), Elman recurrent neural network (ERNN), radial basis function network (RBFN) and Hopfield model (HFM). Inspired by their results and findings, the following machine learning models have been researched for Estimatic: MLPN, long short-term memory recurrent neural network (LSTM RNN), ERNN and RBFN. Below is a brief discussion of the characteristics of these models:

- MLPNs are one of the basic neural networks and they are easy to implement. This makes them a good base case to compare other models with. They are flexible and lend themselves well to learn the mapping between an input and an output.
- ERNNs are thin networks, having only one hidden layer in addition to context units. ERNNs are also known as simple RNNs, capable of time series prediction. Their input is in the form of sequences, which they can process because of their internal state (memory).
- LSTM RNNs are a more complex type of recurrent neural network, capable of dealing with the vanishing gradient problem². Like ERNNs, they are known to work well with time series data.
- RBFNs are also thin networks as they have only one hidden layer. Their hidden neurons use radial basis functions³ as activation function, and each hidden neuron stores its own 'prototype' of the dataset. For every new input, each neuron compares that input to its own prototype, and the closer its resemblance, the higher the output value. RBFNs have many uses, among which are function approximation and time series prediction.

For all models, the learning rate, regularisation term, amount of splits for K-Fold cross validation, number of epochs and optimiser were kept constant. For all models except ERNNs, the loss function was also kept constant. ERNNs had mean squared error as loss function, as MATLAB did not offer mean absolute error as an option. Keras also provided the results as expressed in mean squared error, allowing for comparison between all models. These values are listed in table 4.2.

These parameters were kept constant to not increase the search space of the hyperparameter optimisation. The optimiser was kept constant at its value because "the method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters" [6]. The learning rate was kept constant at its value because it was the default value for the optimiser. The regularisation term was kept at 0.001 because it worked better than the tried values of 0.01 and 0.0001. In order to be able to compare results of models, the number of epochs and K-Fold splits were kept constant. The loss function was set to mean absolute error because of the success criteria mentioned in section 4.3.

²https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

³https://towardsdatascience.com/radial-basis-functions-neural-networks-all-we-need-to-know-9a88cc053448



Figure 4.2: Visualisation of the input

Input data and parameters Each machine learning model is trained on a set of input features. The weather consists of different components, allowing for models to be trained on different sets of weather components. Each set possibly yields different results. Furthermore, the input data consists of time series data. Parameters such as how many lagged observations of the time series to take into account may also affect the accuracy. Estimating the standard deviation of wind components further into the future could also have a negative effect on the accuracy. Finally, there are also training parameters such as the learning rate, number of hidden neurons and number of epochs. This then, is a parameter optimisation problem.

In order for the machine learning models to train on the available data, it must first be processed into the correct format. For example, to create an estimation of the standard deviation of a weather forecast n hours into the future, the model is fed the past x hours of actual weather conditions (lagged observations) and the future weather forecasts up to n hours into the future. Figure 4.2 elaborates on how this affects the size of the input of these models. As the input size differs depending on the amount of hours into the future the estimation is made, a separate model to estimate for each hour into the future has been created. Furthermore, as the lagged hours and the future forecasts are in different units, for example m/s for the KNMI data and knots for the SKV data, the former has been converted to the latter.

Preprocessing the data is reliant on the amount of lagged observations leading up to now, as well as the time difference of the estimation and now. These factors can increase the size of the input data, which leads to longer training times. However, the additional data can also help the models better recognise patterns in the temporal data. Regardless of the size of the input data, it is usual for the data to be normalised before being offered to the models [14] after all data preprocessing has been completed.

Representing the wind direction in a way that makes sense to the machine learning model is an important challenge when preprocessing the data. Represented in degrees, the difference between 359° and 1° is very small, but a machine learning model has no concept of degrees, so it will see the difference as 358 rather than 2. As per advice of To70, the wind direction is represented as its sine and cosine components. This does mean that the original one-dimensional value is now represented with two dimensions. However, the dimensionality of the input can remain the same by removing the wind speed as a separate input and multiplying each of the direction components by the wind speed.

Optimisation The accuracy of a model is reliant on the type of parameters it uses, for example the amount of lagged observations to take into account or the number of neurons, during model training. Optimising these parameters is not an easy task. For the inputs, meteorologists were contacted for their expertise, as mentioned in table 4.1. Their suggestions formed the basis for an exhaustive grid search algorithm. In other words, different sets of parameters were tried on a trial-and-error basis, fine-tuning the search space as certain combinations proved to yield better results.

To validate a model, k-fold cross validation was used. K-fold cross validation splits the dataset in k sections, this allows for an ensemble of k models to be trained, where each section will be used as validation set, and the rest of the sections as training set. Figure 4.3 shows a visualisation where k = 5. The mean of the k performances is taken as the final accuracy of this ensemble. For all models except Elman recurrent neural networks (ERNN), the performance is measured by it mean absolute error. For ERNNs, it is measured by its mean squared error.

The total search space for each parameter that was not kept constant is listed in table 4.3.

Parameter	Search space
Number of hidden neurons Number of hidden layers Number of lagged observations Inputs	10 to 100, in steps of 10 1 to 6, in steps of 1 1 to 12, in steps of 1 The set of inputs as suggested by Marco van den Berge, random subsets of that suggestion and continuing with variations of the subsets that showed promise.

Table 4.3: Search space for machine learning parameters



Figure 4.3: K-fold cross validation with k = 5

Size of the dataset The available SKV (Schiphol Probability Estimation) data ranges from 2010 to 2017. The effect of smaller datasets on the accuracy of Estimatic was researched for To70 to analyse how much data they would need to acquire from other countries' meteorological institutes in order to train Estimatic. With the parameters that were found to be performing best after the optimisation phase, the effect of the following datasets were researched:

- 2 years of data, ranging from 2011-2012
- 4 years of data, ranging from 2011-2014
- 8 years of data, ranging from 2010-2017

The smaller datasets do not start from 2010 because the data in 2010 starts in February, meaning there is one month of data missing. In order to capture two whole years of weather data in the datasets, they start in January of 2011.

4.3. Success criteria

The success criteria of Estimatic differ between predicting the standard deviations of wind direction and wind speed. These criteria have been established by To70 to outline how accurate the estimations must be and are listed below.

- Wind direction standard deviation: The mean absolute error of estimations must be lower than 5° for all the time differences listed in section 3.2.
- Wind speed standard deviation: The mean absolute error of estimations must be lower than 0.28 knots for all the time differences listed in section 3.2.

Amsterdam Airport Schiphol has 5 runways with varying orientations. Whether or not an aeroplane can take off depends on the direction and speed of the wind. To70 stated that because of this, it is unlikely for air traffic control to choose a different runway configuration when the wind direction changes up to 15° with respect to the runway's orientation. Three standard deviations from the mean cover 99.73% of a distribution, which led to a one standard deviation error of 5°. Standard limits are a maximum tailwind of 7 knots, and a maximum crosswind of 25 knots. To70 stated that a 1 knot error for the tailwind can influence the decision, but a 1 knot error for the crosswind does not. Therefore, the mean absolute error has been established at $\frac{7}{25} = 0.28$ knots.

4.4. Software

To support the research as described in this chapter, the following Python project and programming environments was created. The predictions made with Estimatic were made available through an API. The Flask⁴ framework was used for this purpose. To create all but one of the machine learning models, the python deep learning library Keras⁵ was used, in combination with Google's machine learning library Tensorflow⁶. For ERNN models, MATLAB with the Deep Learning toolbox ⁷ was used. A MySQL database was setup to store the historical SKV data, the corresponding KNMI measurements and results and metadata of the trained models. To facilitate a connection with this database, SqlAlchemy⁸, a Python SQL toolkit and ORM was used. The Python visualisation library Matplotlib⁹ was used for visualising results and parameters of models. Finally, Gitlab¹⁰ was used for code version control and continuous integration.

⁴https://www.palletsprojects.com/p/flask/

⁵https://keras.io/

⁶https://www.tensorflow.org/

⁷https://nl.mathworks.com/products/deep-learning.html

⁸https://www.sqlalchemy.org/

⁹https://matplotlib.org/

¹⁰https://gitlab.com/gitlab-org/gitlab-ce/

4.4.1. Quality control and testing

To ensure the quality of the code written, general coding practices were enforced. No code changes could be pushed to the master branch directly. Instead, merge requests had to be created which had to be reviewed before being allowed to merge with the master branch. During these reviews, apart from functionality checks, it was made sure the code adhered to the PEP8¹¹ guidelines and if clear and concise documentation was written for it. To make sure the code functioned as expected, tests were written for the different modules. These tests were written with the pytest¹² framework. A continuous integration pipeline was setup to run these tests. All tests had to succeed for a new addition to the project to be allowed to merge with the master branch.

4.4.2. Model training

Training machine learning models was not feasible on the personal laptops that were used to write code. Instead, Amazon Elastic Compute Cloud (Amazon EC2) was used to host virtual private servers (VPS). Different tasks, such as creating models with varying amounts of training data, were setup before hand and distributed to the different VPSs. This significantly sped up the training process.

4.4.3. SIG

Halfway through the project, the software improvement group (SIG)¹³ reviewed and commented on the project. The project got 3.3 out of 5 stars on the maintainability scale. This means the project is market average maintainable. Things to improve upon were Unit Interfacing and Unit Size. Unit Interfacing relates to functions written with a lot of input parameters. This could indicate a lack of abstraction. Furthermore, calling methods with a lot of input parameters can be confusing. To solve this problem, related parameters were grouped together in parameter classes and dictionaries. It was made sure no method took more than four input parameters. Unit Size is related to pieces of code that are longer than average. Most often, this is because a method contains too much functionality. Resolving this issue was done by creating separate methods to abstract functionality. Finally, it was deemed promising that test were written. The amount was something to be worked on. As such, more tests were written. The full feedback by SIG (in Dutch) can be found in appendix F.

¹¹https://www.python.org/dev/peps/pep-0008/

¹²https://docs.pytest.org/en/latest/

¹³https://www.softwareimprovementgroup.com/



Results

In this chapter, the results are presented. Interpretation of the results is done in chapter 6.

5.1. Statistical methods

This section will discuss the results of using the Weibull method to estimate the standard deviation of wind speed. Figure 5.1 shows an example of the four different probability density functions of the four different fitting methods used. Each prediction Estimatic makes using the Weibull method creates such a graph. The figure also contains the histogram of the filtered data and the calculated standard deviations of the different fitting methods. The visualisation made use of Marco van den Berge's input on a weather forecast of the 11th of February 2016.

RQ₁: What fitting methods work best in the statistical method? Four different methods to fit a Weibull distribution to the eventual dataset were used. Weibull Least Squares Method (LSM), Maximum Likelihood Method, Minimum Weibull LSM and a Double Weibull LSM. All other parameters were kept constant, so as to test the difference between these fitting methods. The set of inputs as suggested by Marco van den Berge was used. The performance of the fitting methods on fifteen different validation sets can be seen in figure 5.2.

Table 5.1 shows the results of a subset of one of the validation sets. All methods estimate a standard deviation above the actual standard deviation. As a result, choosing the minimum of the four fitting methods seemed fitting. A different combination of these four fitting methods is taking the average of the four methods. Figure 5.3 shows these results beside the results of the initial fitting methods.

A boxplot with only the two best performing methods is shown in figure 5.4. This makes the comparison between the two methods clearer.

 \mathbf{RQ}_2 : Which set of inputs work best in the statistical method? Two different sets of inputs as to filter the data were suggested by Marco van den Berge and Adri Huiskamp respectively. Using the two best performing fitting methods, Maximum Likelihood Method and the minimum method, these two sets of inputs as were compared with one and other in figure 5.5.



Figure 5.1: Visualisation of the probability density functions of the different fitting methods



Figure 5.2: Boxplot of the performance of different fitting methods

	Predicted values	
Weibull	[3.75, 7.55, 6.2, 4.18, 4.58, 4.13, 12.06, 3.64, 5.58, 6.26]	
MLM	[3.22, 2.94, 6.17, 3.53, 5.14, 3.97, 3.81, 2.44, 5.52, 6.18]	
Minimum Weibull	[3.65, 5.13, 6.18, 5.04, 6.00, 5.32, 5.32, 4.70, 11.1, 6.31]	
Double Weibull	[3.63, 5.19, 6.22, 4.88, 6.11, 5.98, 4.24, 3.11, 5.74, 6.36]	
Minimum of the four	[3.22, 2.94, 6.17, 3.53, 4.58, 3.97, 3.81, 2.44, 5.52, 6.18]	
Mean of the four	[3.56, 5.20, 6.19, 4.40, 5.45, 4.85, 6.35, 3.47, 6.98, 6.27]	
Actual values.	[1.00, 2.00, 2.00, 2.00, 2.00, 2.00, 1.00, 1.00, 2.00, 3.00]	
Actual values.	Difference between prediction and actual	Mean Absolute Error
Weibull	Difference between prediction and actual [2.75, 5.55, 4.2, 2.18, 2.58, 2.13, 11.06, 2.64, 3.58, 3.26]	Mean Absolute Error 3.993
Weibull MLM	Difference between prediction and actual [2.75, 5.55, 4.2, 2.18, 2.58, 2.13, 11.06, 2.64, 3.58, 3.26] [2.22, 0.94, 4.17, 1.53, 3.14, 1.97, 2.81, 1.44, 3.52, 3.18]	Mean Absolute Error 3.993 2.492
Weibull MLM Minimum Weibull	Difference between prediction and actual [2.75, 5.55, 4.2, 2.18, 2.58, 2.13, 11.06, 2.64, 3.58, 3.26] [2.22, 0.94, 4.17, 1.53, 3.14, 1.97, 2.81, 1.44, 3.52, 3.18] [1.65, 3.13, 4.18, 3.04, 4.00, 3.32, 4.32, 3.70, 9.10, 3.31]	Mean Absolute Error 3.993 2.492 4.075
Weibull MLM Minimum Weibull Double Weibull	Difference between prediction and actual [2.75, 5.55, 4.2, 2.18, 2.58, 2.13, 11.06, 2.64, 3.58, 3.26] [2.22, 0.94, 4.17, 1.53, 3.14, 1.97, 2.81, 1.44, 3.52, 3.18] [1.65, 3.13, 4.18, 3.04, 4.00, 3.32, 4.32, 3.70, 9.10, 3.31] [2.63, 3.19, 4.22, 2.88, 4.11, 3.98, 3.24, 2.11, 3.74, 3.36]	Mean Absolute Error 3.993 2.492 4.075 3.346
Weibull MLM Minimum Weibull Double Weibull Minimum of the four	Difference between prediction and actual [2.75, 5.55, 4.2, 2.18, 2.58, 2.13, 11.06, 2.64, 3.58, 3.26] [2.22, 0.94, 4.17, 1.53, 3.14, 1.97, 2.81, 1.44, 3.52, 3.18] [1.65, 3.13, 4.18, 3.04, 4.00, 3.32, 4.32, 3.70, 9.10, 3.31] [2.63, 3.19, 4.22, 2.88, 4.11, 3.98, 3.24, 2.11, 3.74, 3.36] [2.22, 0.94, 4.17, 1.53, 2.58, 1.97, 2.81, 1.44, 3.52, 3.18]	Mean Absolute Error 3.993 2.492 4.075 3.346 2.436

Table 5.1: Representation of predicted, estimated and error values



Performance of fitting methods and their aggregates

Figure 5.3: Boxplot of the performance of combining different fitting methods



Figure 5.4: Boxplot of the performance of the two best performing methods



Figure 5.5: Boxplot of the performance of the two different sets of inputs

Model	Predicts	no. of layers	no. neurons per layer
Multilayered Perceptron Networks (MLPN)	Direction	2	10
	Speed	2	20
Radial Basis Function Networks (RBFN)	Direction	1	20
	Speed	1	20
Long short-term memory	Direction	1	30
Recurrent Neural Networks (LSTM RNN)	Speed	1	30
Elman Recurrent Neural Networks (ERNN)	Direction	2	20
· · · ·	Speed	2	20

Table 5.2: Best found hyperparameters per model

Abbreviation	Meaning	Units
Т	Temperature	0.1° C
FX	Highest gust measured in the last hour	0.1 m/s
DD	Average wind direction over the last 10 minutes	° clockwise from north
FH	Average wind speed over the last hour	0.1 m/s
TD	Current dew point temperature at 1.50m	0.1° C
FF	Average wind speed over the last 10 minutes	0.1 m/s
Р	Current air pressure reduced to sea level	0.1 hPa
Q	Total solar irradiance over the last hour	J/cm ²
Ν	Cloudiness	Expressed in range 1-8
Μ	Mist in the last hour	1 if mist occured, 0 otherwise
RH	Sum of rain in the last hour	in 0.1 mm

Table 5.3: Explanation of the KNMI data labels

5.2. Machine Learning

The results of each machine learning research questions mentioned in section 3.3 are listed below and in appendix D for readability. For each model, the learning rate, regularisation parameter and number of training epochs were kept constant, as explained in table 4.2. However, as each machine learning model can contain different numbers of hidden layers and neurons per layer, the (sub) optimal set of these parameters first needed to be found. Table 5.2 provides an overview of the best recorded sets of parameters for each machine learning model. These parameters have been kept constant when answering the research questions below, to not grow the search space further.

While the success criteria stated in section 4.3 refer to the mean absolute error, the visualisation of the results in figures 5.6 and 5.7 is done using the mean squared error. This is because the ERNN models were created in MATLAB, and did not provide the model's mean absolute error. The relative performance of a model does not change, a lower mean squared error means the model also had a lower mean absolute error. Due to the difference in loss metric for ERNNs and the other models, and to make visualisation with regards to the success criteria possible, the remaining results for LSTM RNNs, MLPNs and RBFNs will be visualised using mean absolute error, and will not regard ERNNs. The ERNN results can be found in appendix D

RQ₃: Which machine learning model performs best at estimating the standard deviations of wind components? The machine learning models used while researching were multilayered perceptron networks (MLPN), radial basis function networks (RBFN), long short-term memory recurrent neural networks (LSTM RNN) and Elman recurrent neural networks (ERNN). Figure 5.6 and 5.7 visualise the performance of the models when estimating the standard deviations of wind direction and speed, respectively.

RQ₄: Which data features work best as input for the machine learning models? The sets of inputs as suggested by the meteorologists in table 4.1 in section 4.2 were used as a starting point to find the best working set of input parameters for the machine learning models. Their suggestions defined the



Figure 5.6: Wind direction stdev performance per machine learning model



Figure 5.7: Wind speed stdev performance per machine learning model



Figure 5.8: Wind direction stdev performance with regards to inputs, for LSTM RNNs, MLPNs and RBFNs

search space, which was then explored via the exhaustive grid search algorithm mentioned in section 4.2. The sets of inputs as defined in table 4.1 could only be tested for estimations made one hour into the future. This is because the SKV estimations did not have all the features suggested by the meteorologists. To70 explained that a one hour time difference is small enough that an estimation is very close to its true measurement, which allowed for the measurement one hour into the future to be used as a 'forecast'. The set of inputs suggested by Adri Huiskamp could not be used for machine learning, as some values for the temperature at 10cm were missing. The inputs for machine learning models required several hours of lagged observations, which resulted in many unusable sequences because of the missing values. thus leaving too few data to train the models on. For example, using the last 5 hours of lagged observations, half the usual amount of the data was unusable. There were no missing values for the features in Marco van den Berge's set of inputs. The abbreviations of the inputs are explained in table 5.3. Figure 5.8 and 5.9 show the input performance of wind direction and wind speed respectively for LSTM RNNs, MLPNs and RBFNs models. The figures for ERNNs can be found in appendix D, figure D.1 and D.2.

RQ₅: How does increasing the amount of lagged observations as input influence the accuracy? Each model's performance was tested using a different amount of lagged observations as input, that is, taking additional past observations into account when training the models. The results for LTSM RNNs, MLPNs and RBFNs have been visualised in figure 5.10 and 5.11 for wind direction and speed respectively. The remaining figures for ERNNs can be found in appendix D, figure D.3 to D.4.

RQ₆: What happens with the accuracy as an estimation is made further into the future? Each model was first trained to estimate the standard deviations of the wind components one hour into the future. To find out what the effect is on the accuracy of an estimation when predicting the standard deviation of the wind components further into the future, multiple models were trained and evaluated accordingly. Only LSTM RNN models were created due to limited time and computing power, and because of their performance displayed in figure 5.6 and 5.7. The results can be found in figure 5.12 and 5.13 for wind direction and speed respectively.

Best performing parameters After creating many different machine learning models, using exhaustive grid search on the parameters listed in 4.3, the best parameters per model type were found. These result are listed in tables 5.4 and 5.5 for wind direction and speed standard deviation estimation, respectively.



Figure 5.9: Wind spd stdev performance with regards to inputs, for LSTM RNNs, MLPNs and RBFNs



Figure 5.10: Wind direction stdev performance with regards to lagged observations, for LSTM RNNs, MLPNs and RBFNs

Parameter	number of hidden neurons	number of hidden layers	number of lagged observations	inputs
LSTM RNN	35	1	8	DD, FH, FX, T TD
MLPN	10	2	1	DD, FH
RBFN	20	1	6	DD, FH, FX
ERNN	20	2	1	DD, FH, FX

Table 5.4: Best performing parameters for wind direction stdev estimation, per model


Figure 5.11: Wind speed stdev performance with regards to lagged observations, for LSTM RNNs, MLPNs and RBFNs

Parameter	number of hidden neurons	number of hidden layers	number of lagged observations	inputs
LSTM RNN	35	1	2	DD, FH, FX, T, TD
MLPN	20	1	1	DD, FH, FX, TD, RH, N, T, P
RBFN	20	1	9	DD, FH, FX
ERNN	20	2	1	DD, FH, T, P

Table 5.5: Best performing parameters for wind speed stdev estimation, per model



Figure 5.12: LSTM RNNs wind direction stdev performance with regards to delta time estimation



Figure 5.13: LSTM RNNs wind speed stdev performance with regards to delta time estimation

5.3. Data limitations

RQ₇: What is the influence of the amount of years of training data on the accuracy of predictions made? With the smaller dataset sizes as mentioned in 4.2, the performance of the machine learning models was compared when the dataset became smaller. Only LSTM RNN models were created due to limited time and computing power, and because of their performance displayed in figure 5.6 and 5.7. The results for wind direction and speed standard deviation estimation can be found in figures 5.14 and 5.15 respectively.



Figure 5.14: LSTM RNN performance with regard to dataset size



Figure 5.15: LSTM RNN performance with regard to dataset size

\bigcirc

Discussion

In this chapter, the results per research question will be interpreted and discussed. At the end of this chapter the ethical implications of Estimatic will be discussed.

6.1. Statistical methods

There are two method features that have been researched regarding the implementation of the Weibull method to estimate standard deviations of wind speed, fitting methods and sets of inputs. Figure 4.1 shows how Estimatic uses historical measurements to create a prediction. Fitting and choosing a set of inputs are the two most prominent method features in this methodology and the research questions offer an overview on the performance of using the Weibull distribution to predict the standard deviation of wind speed.

RQ₁: What fitting methods work best in the statistical method? The four methods initially used to fit a Weibull distribution to the filtered data were Weibull Least Squares Method (LSM), Maximum Likelihood Method (MLM), Minimum Weibull LSM and a Double Weibull LSM. Figure 5.2 shows a boxplot of the performance of these different models. After measuring performance with the mean absolute error it is clear that MLM not only performs the best but also performs most consistently of the four methods.

The biggest inconsistencies are seen with the Double Weibull LSM. Although the median is lower than other methods, its outliers are what makes this method of fitting unreliable. Minimum Weibull LSM and Weibull LSM do not approach the performance of the MLM method. Although the Maximum Likelihood Method performs best, its mean absolute error is larger than 2. A mean absolute error of 2 is too large for predicting wind speed as the success criteria of To70 state the prediction of wind speed must have only a mean absolute error of 0.28 knots as per section 4.3.

Figure 5.3 shows the performance of the mean and minimum aggregates of the initial four fitting methods. The method that takes the mean of the four estimated values performs the worst of any method previously seen. The high value of some outliers skew that data to very high values. The minimum of the four fitting methods does seem promising. Figure 5.4 shows the difference between MLM and the minimum method. MLM is more consistent than the minimum method but the minimum method does outperform it. Both of the methods do not come close to the success criteria.

No Weibull method managed to approach the success criteria discussed in section 4.3. The eventual decision of the best performing method is a choice between consistency and best possible performance but neither is deemed good enough to offer to To70 for use in Estimatic. In the hypothesis it was stated that the Maximum Likelihood Method (MLM) would work best. The fact that MLM outperformed all other fitting methods beside an aggregate is positive for this case. But this method is sometimes outperformed by the minimum method. The inconsistency of that method suggests there could be some underlying issues with the chosen methodology, these are discussed in section 6.1.1. The findings support **H1**. \mathbf{RQ}_2 : Which set of inputs work best in the statistical method? The results in figure 5.5 show that the set of inputs as suggested by Adri Huiskamp work best. Both the MLM method and minimum method outperform the set of inputs of Marco van den Berge. Although the Adri Huiskamp set of inputs outperforms the set of inputs of Marco van den Berge, there is a difference in consistency between the two. The MLM method on Marco van den Berge's set of inputs performs significantly more consistent than all other methods. The same MLM method on Adri Huiskamp's set of inputs is less consistent than the minimum method.

The inconsistency in the search of the best and most consistent method raises an suspicion: the MLM method has a worse fit to a smaller dataset. This would explain why MLM loses consistency on the more strict data of the Adri set of inputs. Another reason to think this is given by Kaoga et al. [5]. *"MLM is a mathematical expression known as a likelihood function of the wind speed data in time series format."* The time series that MLM uses could be negatively influenced by the stricter data of the Adri set of inputs. Finally, the issues discussed in section 6.1.1 could still explain why the Adri set of inputs outperform the Marco set of inputs even though the MLM fitting method loses consistency.

No Weibull method managed to approach the success criteria discussed in section 4.3. The eventual decision of the best performing method is a choice between consistency and best possible performance but neither is deemed good enough to offer to To70 for use in AFOS. The hypothesis stated that the Marco van den Berge input would perform best. But it was outperformed by the Adri Huiskamp filtering method. Adri Huiskamp's set of inputs is more strict than that of Marco van den Berge. That results in distributions that are closer together and this correlates with smaller calculated standard deviations. A distribution closer together results in smaller standard deviations. As can be seen in table 5.1 the smaller Weibull calculates them better. This conclusion suggests some underlying issues with the chosen methodology, these are discussed in section 6.1.1. These findings fail to support **H2**.

6.1.1. Issues with the Weibull method

Figure 5.1 shows the histogram of wind speed after the filter presented by Marco van den Berge. The histogram after the filter of Adri Huiskamp is similar and both show a looming issue with the Weibull method. Intuitively, filtering on similar weather conditions could show that a specific wind speed is to be expected. A range between 3 and 5 knots, for instance, or maybe very high speeds between 7 and 10 knots. The histograms show otherwise, the histograms suggest that many weather conditions still offer a wide distribution of wind speed data, generally ranging from 0 to 25.

Due to this wide distribution, the standard deviations calculated by the fitting methods are large. The fact that these are larger than the SKV show that SKV takes extra steps to make the standard deviation smaller. This could be the meteorologist that checks the calculated standard deviation afterwards, but could also be a stricter filter for the KNMI model. Constructing an even stricter set of inputs is statistically irresponsible, as filtering even more strictly would result in a distribution built on too few measurements.

6.2. Machine Learning

RQ3: Which machine learning model performs best at estimating the standard deviations of wind components? As can be seen in figures 5.6 and 5.7, long short-term memory recurrent neural networks (LSTM RNN) outperform multilayered perceptron networks (MLPN), radial basis function networks (RBFN) and Elman recurrent neural networks (ERNN) for both wind direction and speed standard deviation estimation. The performance of LSTM RNNs support **H3**, however it can also be seen that ERNNs perform the worst out of all models. This could be because ERNNs are simple recurrent neural networks, and could not handle the complexity of the data as well as LSTM RNNs can, and even MLPNs and RBFNs. Overall, these findings fail to support **H3**.

RQ₄: Which data features work best as input for the machine learning models? As can be seen in figure 5.8, the set of inputs suggested by Marco van den Berge (the set of inputs at the bottom) performed second best for LSTM RNNs and MLPNs, and second worst for RBFNs when estimating the wind direction standard deviaton. For ERNNs, it can be seen in D.1 that the set performed sixth best. For wind speed standard deviations, it performed best for MLPNs and fourth best for LSTM RNNs and RBFNs. Figure D.2 shows that for ERNNs, the set performed sixth worst. The abbreviations of the inputs are explained in table 5.3. The best performing inputs per machine learning model are

	Machine learning model	wind direction stdev inputs	wind speed stdev inputs
-	MLPN	DD, FH	DD, FH, FX, N, P, RH, T, TD
	RBFN	DD, FH, FX	DD, FH, FX, N, P, RH, T, TD
	ERNN	DD, FH, FX, T, TD	DD, FH
	LSTM RNN	DD, FH, FX, T, TD	DD, FH, FX, N, P, RH, T, TD

Table 6.1: Best performing list of inputs per machine learning model type

listed in table 6.1. As can be seen, the suggested set of inputs never resulted in the lowest mean absolute/squared error, thus failing to support **H4**. A reason for this can be that in general, the suggested set performs well for any location, while the found best sets work especially well for the last 8 years of weather data for Schiphol. Further research would be required to conclude if the found sets are generalisable to other locations.

RQ₅: How does increasing the amount of lagged observations as input influence the accuracy? Figure 5.10 and 5.11 show the effect of increasing the amount of lagged observations on the mean absolute error, for LSTM RNNs, MLPNs and RBFNs. The results for ERNNs are listed in D.3 and D.4. As can be seen in these figures, varying the amount of lagged observations has an impact on the mean absolute error of the predicted standard deviations.

For wind direction, MLPNs overall performed worse the more lagged observations were used. RBFNs slightly improved as more observations were used, but like MLPNs, they converged after 4 hours of lagged observations. LSTM RNNs and ERNNS show odd oscillating behaviour, unlike MLPNs and RBFNs. The overall trend for LSTM RNNs was downwards up to its lowest value at 8 lagged observations, after which the trend changed upwards, the models did not show convergence like MLPNs and RBFNs. ERNNs failed to improve the performance of 1 lagged observations. It can be seen that the best performance is by LSTM RNNs, with 8 hours of lagged observations. Their better performance with regard to the other models could be because LSTM RNNs can handle time series data better than MLPNs, RBFNs and ERNNs. While ERNNs are also capable at handling time series data, they could be too simple recurrent neural networks and be unable to handle the complexity of the weather data.

For wind speed, both MLPNs and LSTM RNNs perform best at 2 hours of lagged observations. Both models' performance drops after 2 hours, showing an upward trend. RBFNs show some improvement for 3 and 4 hours, but can not outperform LSTM RNNs. ERNNs show the same behaviour as before, performing best at 1 lagged observation. It can be seen that the best performance is by LSTM RNNs, closely followed by MLPNs at 2 hours of lagged observations. The difference in the number of lagged observations for wind direction and speed standard deviation estimation might indicate that wind speed is less predictable over time, as only 2 lagged observations perform best for estimating its standard deviation. The results fail to support **H5**. Increasing the amount of lagged hours doesn't necessarily increase the accuracy of a prediction. The mean absolute error goes both up and down for a larger amount of lagged hours. However, there is an optimal amount of lagged hours, after which the mean absolute error continues to grow.

RQ₆: What happens with the accuracy as an estimation is made further into the future? As stated in the results for RQ_6 in 5.2, the effect on the accuracy of an estimation when it is made further into the future was only researched for LSTM RNNs,due to limited time and computing power. Figures 5.12 and 5.13 show that overall for both wind direction and speed standard deviation estimation, the accuracy drops as estimations are made further into the future. However, regarding wind direction, estimating standard deviations two hours ahead has a small improvement in accuracy. This also happens when estimating the standard deviation of wind speed two and five hours ahead. These improvements might be because of the models recognising trends in those parts of the data, or because of the randomness in their initialisation. Further research would be needed to understand these events. Overall, the results are inconclusive with regards to **H6**.



Figure 6.1: Estimatic's wind direction stdev estimations versus real values

6.3. Data limitations

RQ₇: What is the influence of the amount of years of training data on the accuracy of predictions made? As stated in the results for RQ_7 in 5.2, the effect on the accuracy of an estimation when the available dataset is smaller was only researched for LSTM RNNs, due to limited time and computing power. Figures 5.14 and 5.15 show that, for both wind direction and speed standard deviation estimation, the mean absolute error decreases as the size of the dataset increases. The performance of the models is proportional to the size of the dataset. These findings support **H7**.

6.4. Estimatic

The methods discussed above result from the research focused on making the best estimation of the standard deviation of wind direction and wind speed. Statistical methods did not offer solutions for wind direction and the performance of Weibull to estimate the standard deviation of wind speed was poor. The machine learning approach offered better results, with LSTM RNNs performing the best. Although the LSTM RNNs near the success criteria as stated in 4.3, they fail to consistently achieve them. Figures 6.1 and 6.2 show the performance of the best LSTM RNN models. These figures visualise the large outliers Estimatic still offers. The effect of these outliers is discussed in the next section.

6.5. Ethical implications

It is important for any research to consider the ethical implications of its result. What would happen if Estimatic produces a completely wrong prediction of the wind components? After all, these predictions are used as an indication of the accuracy of a given weather forecast. AFOS (Airport Forecasting Service) might suggest terrible runway configurations. However, AFOS is a decision-making support tool. At the end of the day, human interaction is needed to choose a runway configuration. The people operating it are able to recognise strange behaviour and choose to discard a suggested configuration. Furthermore, there are no ethical implications of having an accuracy metric of wind components, even if it is wrong, as long as this information is accompanied by information on the accuracy and flaws of



Figure 6.2: Estimatic's wind speed stdev estimations versus real values

the system predicting it. It is up to the end user to correctly interpret the provided information.

Conclusion & Recommendations

7.1. Conclusion

Using the Maximum Likelihood Method (MLM) to fit the Weibull distribution not only performs the best but also performs most consistently of the initial four methods. Introducing aggregates, the method that takes the mean of the four estimated values performs the worst of any method previously seen. Although MLM is more consistent than the minimum method, the minimum method does outperform it. The eventual decision of the best performing fitting method is a choice between consistency and best possible performance but neither is deemed good enough to offer to To70 for use in AFOS.

Adri Huiskamp's suggested set of inputs outperforms that of Marco van den Berge, yet there is a difference in consistency between the two. The MLM method on Marco van den Berge input performs significantly more consistent than all other methods. This being said, no Weibull method managed to near the success criteria discussed in section 4.3. Therefore neither is deemed accurate enough for use in Estimatic.

The long short-term memory recurrent neural networks (LSTM RNNs) outperformed the other machine learning models. For the standard deviation of the wind direction, the lowest mean absolute error of 4.99° was achieved when estimating five hours into the future, taking the last eight hours of weather measurements into account. For the wind speed, the lowest mean absolute error of 0.27 knots was achieved when estimating two hours into the future, taking the last two hours of weather measurements into account.

The success criteria as stated in section 4.3 have not been achieved for all estimation times as listed in section 3.2. Therefore, it is concluded that it is not possible to make accurate enough estimations of the standard deviation of wind components using the historical data and future weather expectations available for Amsterdam Airport Schiphol.

7.2. Recommendations

7.2.1. Future studies

Other locations Within the scope of this study, the standard deviations for wind speed and direction were estimated for Amsterdam Airport Schiphol. While the results did not meet the success criteria stated in section 4.3, these findings cannot be generalised to other airports. Further research must be done using the weather data for different airports in order to find out if Estimatic can be applied there.

Different machine learning models The machine learning models for Estimatic, namely multilayered perceptron networks (MLPN), long short-term memory recurrent neural networks (LSTM RNN), Elman recurrent neural networks (ERNN) and radial basis function network (RBFN), have been tried for estimating the wind component standard deviations. The study by Maqsood et al. [9] showed promising results with Hopfield models, a different type of RNN. In their paper on state-of-the-art techniques for time series pattern recognition, Lin et al. [8] listed several model construction techniques, such as Bayesion Networks and Support Vector Machines, in addition to neural networks. The use of such techniques could pose interesting for future studies.

Classification problem As stated in section 4.2, the problem of estimating the wind component standard deviations is a regression problem. However, the limited variation in the data labels could mean that this problem can be stated as a classification problem. For wind speed standard deviation estimation, there would be 10 classes, ranging from 0 to 9 knots in steps of 1 knot. For wind direction standard deviation, there would be 25 classes, ranging from 0 to 125 degrees in steps of 5 degrees. This only applies for the data available for this study, and is not necessarily generalisable. A study by Salman and Kecman [11] showed promise when stating a regression problem as a classification one. While outperformed by Support Vector Machines in 3 out of 5 regression problems, the results shows promise for this approach.

7.2.2. Recommendations for To70

Frequency of the data The available hourly SKV data did not make it possible to estimate the wind component standard deviations with some of the statistical methods described in section 4.2 of the research report in appendix C. If To70 is able to obtain SKV data that is more frequent than hourly, it could be possible to use those statistical methods to research their accuracy.

Impact of the errors The highest mean absolute errors for wind direction and speed direction standard deviation estimation were obtained when estimating 31 hours into the future, as can be seen in figure 5.12 and 5.13 respectively. These mean absolute errors were the highest among all time differences for estimations listed in section 3.2. This means that the mean absolute error of Estimatic on the validation data is upper bounded by 7.24° and 0.45 knots. However, as can be seen in figure 6.1, some estimation have errors up to 30° for wind direction standard deviation estimation. It is not known what impact such an error would have on AFOS' performance, and would require further looking into if Estimatic is to be implemented in the future.

8

Lessons Learned

One of the main lessons we have learnt this project is to prepare questions for the client to the last detail. Although we went into the first meeting with the client well prepared there were many questions we forgot or did not know we should have asked. Questions as: What are the exact success criteria? Or, how do we find the exact data? These were questions we only asked our client two or three weeks into the project. These questions should have been clear from week one and would have made some early issues clear to us.

The initial research we did on ways to estimate the standard deviation of wind direction and speed were not as effective as they could have been. It was good we did not focus on our data and implementation specifically, as that keeps the research open to new possibilities. But some analysis and more focus would have come in handy in deciding our approach.

For example, simple analysis on the SKV data in week 1 could have offered insights that we could have taken into account. Information as the high number of occurrences of 2 as wind speed standard deviation could have resulted in research regarding training machine learning with fewer outliers or differences. The analysis on how time in the future does not matter as much as initially thought could've resulted in only a single model. Issues with the data, as duplicates and incorrect measurements were also only filtered out in week 3 or 4. With good analysis before starting research these issues could have also been avoided, or handled earlier.

For machine learning we quickly discovered that finding the correct data is quite challenging. Even if you find a suitable dataset, it is most likely not in the required format and would still need to process or validate it before it can be used. This is exactly what happened with the used dataset from the KNMI and the SKV dataset. The SKV dataset was delivered in different file formats, namely: text, html, docx and pdf, which is not ideal as each format requires its own parser. Furthermore, KNMI dataset contained NULL values for needed features, which then could not be used to train on. Another problem was that the units of both datasets were not similar and had to be converted. Finally, different machine learning models require different input dimensions. For example, multilayer perceptron models require a flattened input whilst recurrent neural network models require a multidimensional array as input, simulating timesteps in time-series data.

Another lesson learned with respect to the parameter grid search algorithm used for machine learning, is that a small mistake in the dataset or code invalidates all previously tried combinations of parameters. For example, as mentioned in the previous paragraph, small mistakes in data validation like NULL values for needed features causes the machine learning models to be trained on corrupt data, automatically invalidating the trained models. It is also important to systematically work through all parameters and store the results accordingly, which we did in a MySQL database.

Finally, we learned that an exhaustive search algorithm for machine learning problems can become computationally expensive very fast. For our use case, it was still a viable option, even though training certain models could take up to 5 hours. It therefore does not seem like the right choice for parameter optimisation if the dataset would be larger. Instead, other hyper-parameter optimisation approaches could be explored, like Bayesian or gradient based optimisation.



Glossary

Term	Definition
API	Application Programming Interface
AFOS	Airport Forecasting Service A product from aviation consultancy company To70, which supports decision making for Schiphol.
EPF	Energy Pattern Factor - used to estimate Weibull parameters
EPS	Ensemble Prediction System - predictions based on a combination of models
Keras	A Python machine learning framework
KNMI	Koninklijk Nederlands Meteorologisch Instituut or Royal Netherlands Meteorological Institute
MATLAB	Mathematical computing environment
MLM	Maximum Likelihood Method - used to estimate Weibull parameters
ORM	Object Relational Mapper - used to convert data into usable format
RBFN	Radial Basis Function Network
RNN	Recurrent Neural Network
SKV	Schiphol Kans Verwachting or Schiphol Probability Estimation Weather forecast with standard deviation, provided by the KNMI for Amsterdam Airport Schiphol.
SQL	Structured Query Language
VPS	Virtual Private Server

B

Task Division

	Tasks
Jaap de Boer	Keras machine learning Data preprocessing Visualisations
Dieuwer Hondelink	Data preprocessing API Statistical models DevOps VPS management
Jip Rietveld	Weibull statistical models MATLAB machine learning Visualisations
Rolf de Vries	Keras machine learning Data preprocessing MySQL database VPS management

\bigcirc

Research Report

Estimatic: Research Report Delft University of Technology

Jaap de Boer j.e.deboer@student.tudelft.nl Jip Rietveld j.a.c.rietveld@student.tudelft.nl

Dieuwer Hondelink d.a.hondelink@student.tudelft.nl Rolf de Vries r.b.devries@student.tudelft.nl

Tuesday $7^{\rm th}$ May, 2019

Contents

1	Intr	Introduction		
2	The	Prob	lem	3
	2.1	Introd	luction to the problem	3
	2.2	Defini	tion of the problem	3
3	Inp	ut & (Dutput	3
	3.1	Input		3
	3.2	Outpu	ıt	4
	3.3	SKV o	data	4
4	Res	earch	Questions	5
	4.1	How a	are machine learning models used for weather prediction?	5
		4.1.1	Which machine learning models are used for weather prediction?	5
		4.1.2	Which machine learning models work well for weather prediction? \ldots .	7
		4.1.3	Conclusion	7
	4.2	How a weath	are mathematical models used for the estimation of standard deviation of er?	8
		4.2.1	Input	8
		4.2.2	Mardia (1972)	9
		4.2.3	Yamartino (1984)	9
		4.2.4	Linear standard deviation	10
		4.2.5	Weibull Distribution	10
		4.2.6	Kernel Density Estimation	11
		4.2.7	Conclusion	12
5	Tec	hnical	Decisions	12
6	Cor	nclusio	n and Approach	13
7	Ap	pendix		14
	7.1	Glossa	ary	14

1 Introduction

Weather affects our lives on a daily basis. Transport, agriculture, choosing which clothes to wear, all these things are affected by the expected weather. Meteorological institutes work hard to estimate the upcoming weather as accurately as possible, but as Abhishek et al. state: "the chaotic nature of the atmosphere, the massive computational power required to solve the equations that describe the atmosphere, error involved in measuring the initial conditions, and an incomplete understanding of atmospheric processes mean that forecasts become less accurate as the difference in current time and the time for which the forecast is being made increases" [1]. The **KNMI** (Royal Netherlands Meteorological Institute) seems to be the only meteorological institute that, along with a core weather estimation, also provides the standard deviation of weather components, such as wind speed and direction.

The standard deviation of these weather components allows tools and mathematical models to be implemented. Currently this data is not available in every country. That is why we aim to find a solution to estimate the standard deviation based on a core expectation. In the following report we research various methods on how to estimate the standard deviation of wind speed and direction. Both classical statistical methods and machine learning models have been researched.

2 The Problem

2.1 Introduction to the problem

AFOS (Airport Forecasting Service) is a tool by aviation consultancy company **To70**. It supports air traffic control at Schiphol airport. Given the current weather conditions AFOS estimates which runway configuration works best. Schiphol has 108 different combinations, each with up- and downsides depending on the weather. AFOS relies on the core weather forecast, along with the standard deviations of wind speed and wind direction. The KNMI provides this data in a Schiphol Probability Estimation (**SKV**).

2.2 Definition of the problem

AFOS uses weather estimates and their standard deviations to create a runway capacity forecast and its probability. This information is offered by the KNMI. It turns out, however, that this is a unique situation. By default, other international meteorological institutes only offer a core estimation of the weather, rather than a core estimation and standard deviation. This means that AFOS cannot be implemented at airports outside of The Netherlands without significant investment. To70 wants us to come up with a tool that can estimate the standard deviations of wind speed and wind direction, given a core weather expectation.

The core expectation of the weather should be accompanied with its standard deviations.

3 Input & Output

3.1 Input

As input for the models, we will receive the following parameters:

- Temperature $T \ [^\circ \mathrm{C}]$
- Wind speed V [kt]
- Wind direction θ [°]

This climate data comes from country specific meteorological institutes such as the National Oceanic and Atmospheric Administration¹ (NOAA) of the United States of America or the Meteorological Administration² of China. The European Meteorological Society³ provides an overview of these meteorological services on a per country basis. Such climatic agencies often include an open-to-public database containing e.g. land-based weather station data. These stations often report on the temperature, wind speed, wind direction, humidity and other core weather components. They usually do this on a monthly, daily, hourly or even on a minute by minute basis.

Furthermore, for training and validation purposes, we have the SKV data. The SKV data contains the parameters named above, but also contains the output defined below. In section 3.3 we clarify this data.

3.2 Output

The output of Estimatic will need to be usable as the input data of AFOS. The only data missing from the current input offered by international meteorological institutes are the standard deviation of wind direction and wind speed.

- Standard deviation of the wind speed σ_s [kt]
- Standard deviation of the wind direction σ_d [°]

The output of Estimatic should be validated to ensure that the system provides accurate estimations of the standard deviation. This can be done in a number of ways. The easiest way to validate the output is by using weather forecasts from countries that provide the standard deviation by default, such as the SKV forecast.

An alternative method is to estimate the standard deviation of a forecast, followed by afterwards comparing the estimated distribution with the actual weather to check if the real value lies within the bounds estimated. Especially the second method can be performed in a lot of countries, allowing us to validate our model in different environments across the world.

3.3 SKV data

To train and validate Estimatic To70 has offered SKV data. Not all data in the SKV is of use for Estimatic. Below you will find all useful parameters.

- Wind direction [°clockwise from north]
- Wind direction standard deviation [°] (seems to be rounded to nearest quintet)
- Wind speed [kt]
- Wind speed standard deviation [kt]

¹https://www.ncdc.noaa.gov/

²http://www.cma.gov.cn/

³https://www.emetsoc.org/

- Gusts [kt]
- Temperature [°C]
- Probabilities regarding visibility conditions (further classification is confidential)

SKV data is created based on models of various meteorological institutes. Institutes on bordering locations consult each other regarding their predictions.

How we will use this data to validate and train Estimatic can be found in section 5.

4 Research Questions

To help us structure our search process, we formulated two research questions. The first is regarding potential machine learning models. To70's initial thoughts were to use machine learning models to estimate the standard deviation of wind speed and direction. But because companies often enjoy using buzzwords, we thought it would be wise to also research more classical approaches to making these estimations.

 \mathbf{RQ}_1 : How are machine learning models used for weather prediction?

 \mathbf{RQ}_2 : How are mathematical models used for the estimation of standard deviation of weather?

To answer \mathbf{RQ}_1 we proposed the following sub questions:

- Which machine learning models are used for weather prediction?
- Which machine learning models work well for weather prediction?

And to answer \mathbf{RQ}_2 we proposed the following sub question:

• What are standard statistical methods?

After answering these questions, we will make a decision regarding the models or methods we will implement for Estimatic.

4.1 How are machine learning models used for weather prediction?

4.1.1 Which machine learning models are used for weather prediction?

To determine how and which machine learning models are used for weather forecasting, we have researched several papers to establish which methods or models have been used. An overview of these findings can be found in table 1. Models used by papers [6] and [5] were not specifically used for weather prediction, but application of the models showed that they can be used to predict weather.

The input space for the machine learning models listed in table 1 is quite large, and not all components are needed in order to estimate the weather. Mohandes et al. [4] only used wind speed data in order to predict wind speed. Maqsood et al. [3] used multiple inputs, i.e. temperature,

Machine Learning Model	Paper
Elman Recurrent Neural Network (ERNN)	[2] $[3]$
Hopfield Recurrent Neural Network (HRNN)	[3]
Multilayer Feed-Forward Neural Network	[1] [4]
Radial Basis Function Network	[3]
Multilayer Perceptron Network (MLP)	[3] [5] [6]
Fuzzy time series model	[7]
Ensemble of Neural Networks	[3] [5]

Table 1: Overview of Machine Learning Models used in previous studies

Input	Paper
Oscillation index	[2]
Sea surface temp.	[2]
Outgoing long wave radiation	[2]
Temperature	[1] $[7]$ $[3]$
Cloudiness	[7]
Wind	[1] $[2]$
Wind speed	[1] $[3]$ $[4]$
Humidity	[1] [3]
Not specific	[5] [6]

Table 2: Machine learning model inputs

Predicts	Paper
Temperature Wind speed Humidity Other	$ \begin{bmatrix} 1 & [2] & [7] & [3] \\ [3] & [4] \\ [3] \\ [5] & [6] $

Table 3: Machine learning model predictions

Name	Function	Paper
Mean squared error Root mean squared error Mean absolute percentage error Median absolute deviation	$MSE = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{X}_i)^2$ $RMSE = \sqrt{MSE}$ $MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \frac{X_i - \hat{X}_i}{X_i} $ $MAD = median(X_i - median(X))$	[1] [5] [3] [4] [3] [3]
Correlation coefficient	$\rho = \frac{Cov(X,Y)}{\sigma_x \sigma_y}$	[3]

Table 4: Error measurements

wind speed and humidity, but they also predicted these weather components. However, this does not imply a one-to-one relation between the input and output of machine learning models, as Chen and Hwang [7] used both temperature and cloudiness to only predict temperature. The inputs used by the studies is given in table 2, and their predicted outputs in table 3.

4.1.2 Which machine learning models work well for weather prediction?

Different error measurements can be used to determine the performance of machine learning models. Table 4 contains an overview of such error measurements linked to the specific papers that use them. Most papers use a single metric for evaluation, but Maqsood et al.[3] use a combination of different metrics to determine the performance of their model. Overall it seems like a good idea to use different metrics to evaluate the accuracy of the output of a machine learning model.

The study by Maqsood et al. [3] used compared ensembles of four different neural networks to forecast weather. In this study, RBFN showed the best results, both in terms of accuracy as well as training time. After RBFN, MLP and ERNN showed similar results in accuracy and training time, but it was stated that: "The ERNN model, compared to MLP, could efficiently capture the dynamic behaviour of the weather, resulting in a more compact and natural representation of the temporal information contained in the weather profile." HFM performed the worst of the four models, in terms of accuracy. However, in terms of training speed, it was the second best. In addition to ensembles of single models, Maqsood et al. also created ensembles of combinations of aforementioned models. These combination ensembles were created with two approaches, winnertakes-all (WTA) and weighted average (WA). The WTA and WA ensembles were compared with the single model ensembles, and both were found to predict the weather more accurately. It is worth noting that the WTA ensemble method performed the best in predicting weather, in terms of error.

In their study, Magdon-Ismail and Atiya [6] tried to approximate the probability density function of a random variable using machine learning methods. It is similar to kernel density estimation (KDE) but uses a neural network for its prediction. Compared with KDE, the method used in the paper performs faster estimations but lacks details on the accuracy of the methods. Furthermore, the method is still highly experimental and not much research has been done into approximating probability density functions using neural networks.

4.1.3 Conclusion

The researched studies listed in table 1 have shown that it is possible to accurately predict weather with machine learning models. From the results we have found, we conclude that a Radial Basis Function Network is the best choice for a single model weather estimator, in terms of accuracy and training time. Maqsood et al. [3] have shown that it outperforms MLP, ERNN and HFM in either accuracy or training time, or both. Though the numbers regarding performance seem sufficient, we will try out different models to see which works best, this is further discussed in section 5.

While the study by Magdon-Ismail and Atiya [6] shows promise with regards to estimation time, the lack of details on the accuracy of the methods are discouraging. Therefore, we are probably better of using traditional methods to approximate the probability density function if needed.

Inspired by the findings of Maqsood et al. [3], we will apply a similar approach for Estimatic, where we create ensembles of single model networks to compare. After comparing these single model ensembles, we will create combined model ensembles as well, with a winner-takes-all and weighted average approach.

The performance of all ensembles will then be compared, to provide an overview of their accuracy. Multiple error measurements (e.g. RMSE, MAD) will be used for the error measurement, as inspired by Maqsood et al. [3]. Regarding the inputs we will use for the machine learning models, we can not yet give a definite decision. We believe that the best approach is to try sensible subsets of the input space, to see which components are relevant in estimating wind speed and direction. In doing so, we believe this will find our best machine learning method to estimate the standard deviation of wind speed and direction.

4.2 How are mathematical models used for the estimation of standard deviation of weather?

Part of our research focused on current methods used to calculate the standard deviation of wind speed and wind direction. The KNMI currently calculates these with their model. We will also explore classic statistical options for these estimations. For these different methods we will offer an overview of how the method works, as well as the specific input it might need. Eventually we will choose the methods we will implement for Estimatic, where we hope to compare these different methods to find the best method (or combination of methods) suited for our needs.

During our research we encountered many papers discussing multi-pass or single-pass methods to calculate the standard deviation of wind speed or direction. These papers date back all the way to the 1960s, where computational power might not have been so strong as it is now. Besides that, the main disadvantage of a multi-pass method is that it cannot be computed as soon as a measurement is made by an instrument. But because we are calculating standard deviation afterwards, that is a non-issue. Therefore, we do not believe choosing between a single or multipass method is relevant for us.

4.2.1 Input

We have found common input parameters for the methods used. Below we will list them and briefly explain where they come from and what they consist of.

Wind speed and wind direction are represented with different symbols. For this paper we will use θ to represent wind direction, and V to represent wind speed. When subscript is used to further define a certain variable, d represents wind direction and s represent wind speed.

 σ will be used to represent the standard deviation we are going to calculate. As stated above σ_s represent the standard deviation of wind speed and σ_d represent the standard deviation of wind direction.

Other subscript annotations are $_a$ and $_i$. $_a$ stands for average and $_i$ stands for a individual data point. For example, θ_a will be the average wind direction.

Besides θ_a some methods use s_a and c_a . These are defined as follows: $n^{-1} \sum_{i=1}^{n} sin(\theta_i) = s_a$. Idem for cosine to define c_a .

Given average wind direction θ_a then $n^{-1} \sum_i^n \sin(\theta_i) = s_a \neq \sin(\theta_a)$. The same holds for the cosine. Because Estimatic is only offered the core expectation of the wind direction (see section 3.1), θ_a , we are unable to reverse engineer the values of s_a and c_a and cannot use methods that utilise these variables.

Symbol	Description
$\overline{\theta}$	Wind direction [°]
V	Wind speed [kt]
σ	Standard Deviation
d	Wind direction
8	Wind speed
a	Average
i	Individual data point
s_a	Average of individual sines of direction
c_a	Average of individual cosines of direction

Table 5: Often used symbols in classical methods

4.2.2 Mardia (1972)

In the book "Statistics of Directional Data" Mardia discusses a single-pass method to calculate the standard deviation of wind direction. Even though access to the book was restricted, the paper by Mori [8] did offer enough information for us to be able to discuss the method.

The method offered is as follows:

$$\sigma_d = (-2\ln(R))^{1/2} \tag{1}$$

with R defined as follows: $R = (s_a^2 + c_a^2)^{1/2}$.

Mori [8] concludes that Mardia's method is one of the best single-pass estimators of σ_d in practical application. So does Weber [9] in his study in which Mardia's method has an RMSE of only 2.9 degrees. However Estimatic will not be able to use it due to the input restrictions described in Section 4.2.1.

4.2.3 Yamartino (1984)

Yamartino[10] has researched his single-pass method extensively, comparing its results to the classically used two-pass method to determine standard deviation of wind direction. In his research he arrives at three different estimators for the standard deviation. The first two estimators were not sufficient in handling standard deviations larger than 90°. However, the third one was able to accurately estimate a standard deviation with a degree of $\pm 2\%$ (compared with the classical two-pass method).

The method is as follows:

$$\sigma_d = \sin^{-1}(\epsilon)[1.0 + b\epsilon^3] \tag{2}$$

Given:
$$b = (2/\sqrt{3}) - 1 = 0.1547$$
 and $\epsilon^2 = 1 - (s_a^2 + s_a^2)$

Even though this method has not been tested on field data it does solve issues presented by Ackermann [11] and Verrall and Williams [12]. Where the former over predicts for values above 40° and the latter under predicts. Yamartinos method does not have these issues.

Weber [9] shows the Yamartino method to have an RMSE of only 1.9 degrees. Turner [13] confirms Yamartino's method as the best method when compared with Verrall and Williams [12] and Ackermann [11]. But unfortunately because s_a and c_a are needed we cannot use the method described by Yamartino in Estimatic (Section 4.2.1).

4.2.4 Linear standard deviation

Understanding standard deviation calculations regarding wind direction and speed should start off by understanding the core concept of standard deviation. The usual method to calculate standard deviation is as follows:

$$\sigma^2 = E(X^2) - E(X)^2$$
(3)

With E being a function returning the expected value of variable X.

Multiple papers start off with remarks regarding this usual linear estimation of the standard deviation of an estimate. Turner [13] explains that due to the circular function of wind direction, e.g. 360° and 0°, are the same, that this standard statistical method cannot be used. Ackermann [11], Weber [9] and Mori [8] all confirm that this discontinuity means this method will not be of use for determining the standard deviation of wind direction.

Regarding wind speed however, using this method to determine standard deviation could be of some use. Although this model does not handle real-world phenomena it could be of use for giving a general estimation. So although we do not expect this method to be used in the final version of Estimatic, we will implement and compare it to the other models.

4.2.5 Weibull Distribution

Multiple studies have shown that a Weibull Distribution often correctly describes the wind speed distribution [14] [15] [16]. This two-parameter distribution takes a parameter k and C, with k > 0 being the so-called shape-parameter and C > 0 being the scale parameter.

All three of the above studies have determined different methods to determine parameters k and C. Below you will find an overview of these methods that are most relevant to our data set. Once these parameters are known, the standard deviation can be determined by the following equation.

$$\sigma_s = C^2 \left[\Gamma \left(1 + \frac{2}{k} \right) - \left(\Gamma \left(1 + \frac{1}{k} \right) \right)^2 \right] \tag{4}$$

With Γ defined as the gamma function.

Maximum likelihood method (Kaoga 2014[14])

The Maximum Likelihood Estimation method (MLM) uses time series to estimate the values of parameters k and C. The shape factor k and scale factor C are estimated as follows.

$$k = \left[\frac{\left(\sum_{i=1}^{n} V_{i}^{k} ln(V_{i})\right)}{\left(\sum_{i=1}^{n} V_{i}^{k}\right)} - \frac{\left(\sum_{i=1}^{n} ln(V_{i})\right)}{n}\right]^{-1}$$
(5)

$$C = \left(\frac{1}{n}\sum_{i=1}^{n}V_{i}^{k}\right)^{1/k}\tag{6}$$

Where:

n = number of none zero data values

i = measurement interval

 V_i = wind speed measured at the interval i[m/s]

Energy pattern factor method (Kaoga 2014[14])

The energy pattern factor (EPF) method is related to the averaged data of wind speed. It defines shape factor k and scale factor C as follows:

$$k = 1 + \frac{3.69}{(E_{pf})^2} \tag{7}$$

$$C = \left(\frac{1}{n}\sum_{i}^{n} = 1\vec{V}_{i}^{k}\right)^{1/k} \tag{8}$$

With $E_p f$ defined as the energy pattern factor. $E_p f = \vec{V^3} / \vec{V^3} = \frac{\left(\frac{1}{n} \sum_{i=1}^n \vec{V_i^3}\right)}{\left(\frac{1}{n} \sum_{i=1}^n \vec{V_i}\right)^3}$

Both of the above methods were tested on the same data sets by Kaoga et. al. Even though EPF is one of the worse performing parameter estimators it is one of the few we can use with the available data set. MLM however is recommended by Kaoga as an alternative to the EPF method. Luckily our data set does offer the necessary data to estimate Weibull distribution parameters using the MLM method. However the other methods described by Kaoga et. al [14] cannot be applied to the available data set.

4.2.6 Kernel Density Estimation

In statistics, kernel density estimation (KDE) is a non-parametric method for estimating the probability density function of a random variable. Non-parametric methods do not require the data to follow a normal distribution and hence do not rely on data such as the mean and variance.

A kernel density estimation requires two predefined parameters, namely a chosen kernel function and the smoothing parameter, also called the bandwidth. Kernel functions are basically alternative versions of probability density function. Different kernels produce different estimates. The smoothing parameter determines the shape of the kernel. Examples of kernel functions are:

- Epanechnikov
- Uniform
- Triangular
- Gaussian

KDE, also called the Parzen–Rosenblatt method, works as follows. First we generate a sample of the data. Then, around each data point of the sample, we create a kernel using the chosen kernel function. Finally, we combine the different kernel functions to a single density function. After normalisation, the resulting function is the estimation of the probability distribution function.

The mathematical description of kernel density estimation is as follows:

$$f(x) = \frac{1}{n} \sum_{n=1}^{n} K(\frac{x - x_i}{h})$$
(9)

Here, K is the kernel function and h is the smoothing parameter.

4.2.7 Conclusion

The methods described by Mardia (4.2.2) and Yamartino (4.2.3) cannot be used by Estimatic. Both of those methods require inputs s_a and c_a and as described in section 4.2.1 these are not available to us.

Even though s_a and c_a cannot be calculated we can try and approach these methods anyway with our given input. With the given θ_a of the previous hours we could calculate s_a and c_a by taking the sine of the direction angle (instead of the average of individual sines), and then try and calculate a standard deviation anyway.

We want to try and make these methods work because Turner [13] is content with the results offered by these methods. Also Weber [9] has concluded that Yamartino's method performed best compared to other methods.

We realise an incomplete implementation of these methods is far from optimal. But because the implementation is relatively trivial, we hope to still be able conclude something from these implementations.

In addition to these weather methods, we will also implement a number of different statistical methods. The Weibull Distribution as well as the Kernel Density Estimation are both used by multiple papers to determine an estimation of the standard deviation. Because these distributions can be built with the entire data set or only seasonal data, we hope to be able to find the most optimal estimation of the standard deviation of wind speed and direction. For example, Abhishek et al.[1] describes using a 0, 2 or 4 season approach and Wieringa and Rijkoort[17] discuss seasonal-average data. Therefore we will also try out different configurations of the data.

Other methods described by Ackermann [11] and Verrall and Williams [12] were discarded because the data used to execute the methods is not available to us.

5 Technical Decisions

For the machine learning aspect of Estimatic we compared different machine learning frameworks, based on their popularity, namely:

- TensorFlow ⁴
- $\bullet\,$ Keras 5
- PyTorch ⁶
- Caffe ⁷
- Deeplearning4j ⁸

TensorFlow, Keras and PyTorch are all Python ⁹ frameworks. Tensorflow is the most popular machine learning framework and is developed by Google. It has a large community and is well documented, but has a steep learning curve. Keras is a high-level API on top of frameworks such as TensorFlow. It allows users to quickly setup both basic and complex neural networks. Therefore,

⁴https://www.tensorflow.org/

⁵https://keras.io/

⁶https://pytorch.org/

⁷https://caffe.berkeleyvision.org/

⁸https://deeplearning4j.org/

 $^{^9 {\}tt https://www.python.org/}$

Keras offers most of the functionality of Tensorflow, but is easier to use for newer users. PyTorch is a different Python machine learning framework, developed by Facebook. It is more intuitive to use than Tensorflow. Caffe is a C++ framework, which works well with image data. However, as the data we will be working with is not image data, we believe this is not the framework we should go with.

Taking all this into account, we choose to use Keras as machine learning framework, as we are new to machine learning and it seems that new users should start with Keras because of its ease of use ¹⁰. As a result of choosing Keras, we will be using Python, as that is the language for Keras.

For traditional methods, the choice of programming language is trivial, as they can be implemented in any language. If computation speed proves to be an issue, we might have to try different languages.

6 Conclusion and Approach

Even though the required data to execute the classic statistical methods is not available in most cases, we wish to implement them experimentally. The implementation of these methods are quite trivial, so we do not expect to lose much when implementing them. Even though this is far from optimal we still hope to be able to make some findings based on their results. Besides that, we believe having these methods implemented in Estimatic paves the way to be able to easily use these methods if the data does ever allow it.

The classic statistical methods of which we do have the necessary data will be implemented and compared to the other models or methods we implement. By comparing these results to the machine learning models we hope to prove or disprove either method as being the most effective in estimating the standard deviation of wind speed and direction.

Hopefully machine learning will be able to offer the estimations. We do not believe there exists a set path for the machine learning approach that we can already decide on. Instead, we will explore the large space of possibilities, in order to find which model, inputs, and other parameters will best fit our problem. Here, we will quickly review our envisioned approach.

We will use the Python machine learning framework Keras, as we are new to machine learning, and Keras is stated to be easy to use and learn. As a result of choosing Keras, Python will be the programming language for the machine learning approach. As for the details of the machine learning, this is where the exploratory part comes in. In order to accurately estimate the standard deviations of wind speed and direction, we will have to experiment with:

- The choice of machine learning model
- The design of the model (no. of hidden layers, no. of neurons/layer)
- Parameters of the model (e.g. learning rate)
- Whether we use ensembles of the model (or multiple models)
- Which inputs are best for our estimations

¹⁰"Start with Keras if you are new to deep learning." https://towardsdatascience.com/ deep-learning-framework-power-scores-2018-23607ddf297a

7 Appendix

7.1 Glossary

Term	Definition
KNMI	Koninklijk Nederlands Meteorologisch Instituut or Royal Netherlands Meteorological Institute
AFOS	Airport Forecasting Service A product from aviation consultancy company To70, which supports decision making for Schiphol.
SKV	Schiphol Kans Verwachting or Schiphol Probability Estimation Weather forecast with standard deviation, provided by the KNMI for Amsterdam Airport Schiphol.

References

- K. Abhishek, M. Singh, S. Ghosh, and A. Anand, "Weather forecasting model using artificial neural network," *Proceedia Technology*, vol. 4, pp. 311–318, 2012.
- [2] A. G. Salman, B. Kanigoro, and Y. Heryadi, "Weather forecasting using deep learning techniques," in 2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 281–285, IEEE, 2015.
- [3] I. Maqsood, M. R. Khan, and A. Abraham, "An ensemble of neural networks for weather forecasting," *Neural Computing & Applications*, vol. 13, no. 2, pp. 112–122, 2004.
- [4] M. A. Mohandes, S. Rehman, and T. O. Halawani, "A neural networks approach for wind speed prediction," *Renewable Energy*, vol. 13, no. 3, pp. 345–354, 1998.
- [5] N. Kourentzes, D. K. Barrow, and S. F. Crone, "Neural network ensemble operators for time series forecasting," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4235–4244, 2014.
- [6] M. Magdon-Ismail and A. F. Atiya, "Neural networks for density estimation," in Advances in Neural Information Processing Systems, pp. 522–528, 1999.
- [7] S.-M. Chen and J.-R. Hwang, "Temperature prediction using fuzzy time series," *IEEE Trans*actions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 30, no. 2, pp. 263–275, 2000.
- [8] Y. Mori, "Evaluation of several "single-pass" estimators of the mean and the standard deviation of wind direction," *Journal of climate and applied meteorology*, vol. 25, no. 10, pp. 1387– 1397, 1986.
- [9] R. O. Weber, "Estimators for the standard deviation of horizontal wind direction," Journal of Applied Meteorology, vol. 36, no. 10, pp. 1403–1415, 1997.
- [10] R. J. Yamartino, "A comparison of several "single-pass" estimators of the standard deviation of wind direction," *Journal of Climate and Applied Meteorology*, vol. 23, no. 9, pp. 1362–1366, 1984.
- [11] G. Ackermann, "Means and standard deviations of horizontal wind components," Journal of Climate and Applied Meteorology, vol. 22, no. 5, pp. 959–961, 1983.
- [12] K. Verrall and R. Williams, "A method for estimating the standard deviation of wind directions," *Journal of Applied Meteorology*, vol. 21, no. 12, pp. 1922–1925, 1982.

- [13] D. Bruce Turner, "Comparison of three methods for calculating the standard deviation of the wind direction," *Journal of climate and applied meteorology*, vol. 25, no. 5, pp. 703–707, 1986.
- [14] D. K. Kaoga, S. D. Yamigno, D. Raidandi, and N. Djongyang, "Performance analysis of weibull methods for estimation of wind speed distributions in the adamaoua region of cameroon," *International Journal of Basic and Applied Sciences*, vol. 3, no. 3, p. 298, 2014.
- [15] A. Garcia, J. Torres, E. Prieto, and A. De Francisco, "Fitting wind speed distributions: a case study," *Solar energy*, vol. 62, no. 2, pp. 139–144, 1998.
- [16] C. Justus, W. Hargraves, A. Mikhail, and D. Graber, "Methods for estimating wind speed frequency distributions," *Journal of applied meteorology*, vol. 17, no. 3, pp. 350–353, 1978.
- [17] J. Wieringa and P. J. Rijkoort, "Windklimaat van nederland," ISBN 9012044669, 1983.

Machine Learning Results



Figure D.1: ERNNs wind direction stdev performance with regards to inputs



Figure D.2: ERNNs wind speed stdev performance with regards to inputs


Figure D.3: ERNNs wind direction stdev performance with regards to lagged observations



Figure D.4: ERNNs wind speed stdev performance with regards to lagged observations

Specification of Sets of Input

Marco van den Berge

Input	Window size
Wind direction	20 [°]
Temperature	10 [° C]
Solar irradiation	75 [//cm ²]
Air pressure	200 [0.1 hPa]
Cloudiness	5 [coverage rate of the sky in eighths]
Mist	0 [0 or 1]

Adri Huiskamp

Input	Window size
Wind direction Difference temp. at 150 cm and 10 cm Solar irradiation Cloudiness	20 [°] 5 [° C] 75 [//cm ²] 5 [coverage rate of the sky in eighths]
Ratio between gusts and avg. wind speed	15%

SIG Feedback

Cc: o.w.visser@tudelft.nl

Beste,

Hierbij ontvang je onze evaluatie van de door jou opgestuurde code. De evaluatie bevat een aantal aanbevelingen die meegenomen kunnen worden tijdens het vervolg van het project. Bij de evaluatie van de tweede upload kijken we in hoeverre de onderhoudbaarheid is verbeterd, en of de feedback is geaddresseerd. Deze evaluatie heeft als doel om studenten bewuster te maken van de onderhoudbaarheid van hun code, en dient niet gebruikt te worden voor andere doeleinden.

Let tijdens het bekijken van de feedback op het volgende:

Het is belangrijk om de feedback in de context van de huidige onderhoudbaarheid te zien. Als een project al relatief hoog scoort zijn de genoemde punten niet 'fout', maar aankopingspunten om een nog hogere score te behalen. In veel gevallen zullen dit marginale verbeteringen zijn, grote verbeteringen zijn immers moeilijk te behalen als de code al goed onderhoudbaar is.
Voor de herkenning van testcode maken we gebruik van geautomatiseerde detectie. Dit werkt voor de gangbare technologieën en frameworks, maar het zou kunnen dat we jullie testcode hebben gemist. Laat het in dat geval vooral weten, maar we vragen hier ook om begrip dat het voor ons niet te doen is om voor elk groepje handmatig te kijken of er nog ergens testcode zit.
Hetzelfde geldt voor libraries: als er voldaan wordt aan gangbare conventies worden die automatisch niet meegenomen tijdens de analyse, maar ook hier is het mogelijk dat we iets gemist hebben.

Mochten er nog vragen of opmerkingen zijn dan horen we dat graag.

Met vriendelijke groet, Dennis Bijlsma

[Feedback]

De code van het systeem scoort 3.3 sterren op ons onderhoudbaarheidsmodel, wat betekent dat de code marktgemiddeld onderhoudbaar is. We zien Unit Interfacing en Unit Size vanwege de lagere deelscores als mogelijke verbeterpunten.

Voor Unit Interfacing wordt er gekeken naar het percentage code in units met een bovengemiddeld aantal parameters. Doorgaans duidt een bovengemiddeld aantal parameters op een gebrek aan abstractie. Daarnaast leidt een groot aantal parameters nogal eens tot verwarring in het aanroepen van de methode en in de meeste gevallen ook tot langere en complexere methoden. Dit kan worden opgelost door parameter-objecten te introduceren, waarbij een aantal logischerwijs bij elkaar horende parameters in een nieuw object wordt ondergebracht. Dit geldt ook voor constructors met een groot aantal parameters, dit kan een reden zijn om de datastructuur op te splitsen in een aantal datastructuren. Als een constructor bijvoorbeeld acht parameters heeft die logischerwijs in twee groepen van vier parameters bestaan, is het logisch om twee nieuwe objecten te introduceren.

Voorbeelden in jullie project:

- DarkSkySqlAlchemyModel.__init_(...)
- MLVisualiser._plot_minimum_loss(predicts,axis_y,axis_x,to_plot,model,max_y,max_x)
- CoreExpectation.__init__(wind_dir,wind_spd,temp,dt,issue_dt)

Bij Unit Size wordt er gekeken naar het percentage code dat bovengemiddeld lang is. Dit kan verschillende redenen hebben, maar de meest voorkomende is dat een methode te veel functionaliteit bevat. Vaak was de methode oorspronkelijk kleiner, maar is deze in de loop van tijd steeds verder uitgebreid. De aanwezigheid van commentaar die stukken code van elkaar scheiden is meestal een indicator dat de methode meerdere verantwoordelijkheden bevat. Het opsplitsen van dit soort methodes zorgt er voor dat elke methode een duidelijke en specifieke functionele scope heeft. Daarnaast wordt de functionaliteit op deze manier vanzelf gedocumenteerd via methodenamen.

Voorbeelden in jullie project:

- mlp.py:main()
- rbfn.py:main()
- DarkSkySqlAlchemyModel.hourly_forecast_to_sql_alchemy_model(hourly_forecast)
- data_helper.py:get_data(columns_to_select,prediction_time_difference,max_hours_back)

De aanwezigheid van testcode is in ieder geval veelbelovend. De hoeveelheid tests blijft nog wel wat achter bij de hoeveelheid productiecode, hopelijk lukt het nog om dat tijdens het vervolg van het project te laten stijgen. Op lange termijn maakt de aanwezigheid van unit tests je code flexibeler, omdat aanpassingen kunnen worden doorgevoerd zonder de stabiliteit in gevaar te brengen.

Over het algemeen is er dus nog wat verbetering mogelijk, hopelijk lukt het om dit tijdens de rest van de ontwikkelfase te realiseren.

Dennis Bijlsma | Senior Consultant

+31 6 45 172 816 l d.bijlsma@sig.eu

Software Improvement Group I www.sig.eu

DB

\bigcirc

Project Info Sheet

General information:

Title of the project: Estimatic Name of the client organisation: AerLabs Date of the final presentation: 03-07-2019 10:00 Final Report: Delft University of Technology

Description:

This project was created to estimate the standard deviations of wind direction and wind speed, based on the historical weather and forecast data for Amsterdam Airport Schiphol. AerLabs commissioned this project in collaboration with To70 to explore the possibility of using AFOS (Airport Forecasting Service) at international airports.

The main challenge of this research was that there have not been any studies as of yet that research estimating the standard deviations of wind speed and wind direction based on historical data.

Our research offered two possible solutions to estimate the standard deviation of wind direction and wind speed: a statistical method approach and a machine learning model approach. The outcome of the research of these approaches has resulted in the product called Estimatic.

Multiple approaches and optimisations were outlined and implemented. These results were recorded and discussed in the research.

Estimatic will take a weather forecast and accompany it with the standard deviation of wind direction and wind speed. The mean absolute error between the estimated value and actual value was taken to measure the performance of Estimatic.

Eventually it was concluded that it is not possible to make accurate enough estimations of the standard deviation of wind components using the historical data and future weather expectations available for Amsterdam Airport Schiphol.

If To70 is able to obtain SKV data that is more frequent than hourly, it could be possible to use new statistical methods to research their accuracy. Besides that, if Estimatic is to be implemented in the future, To70 should research what impact an outlier would have on AFOS' performance.

Members of the project team:

Name: Jaap de Boer

Interests: Machine Learning, Data Science, Back-end development

Contribution and role: Keras machine learning, data preprocessing and visualisations

Name: Dieuwer Hondelink

Interests: DevOps, Machine Learning, Back-end development

Contribution and role: Data prepocessing, API, DevOps and VPS management

Name: Jip Rietveld

Interests: Algorithm Design, Data Science, Design Patterns

Contribution and role: Weibull statistical model, MATLAB machine learning, presentation and visualisations

Name: Rolf de Vries

Interests: Machine Learning, Data Science, Back-end development

Contribution and role: Keras machine learning, data prepocessing, MySQL, presentation and VPS management

All team members contributed to the final report.

Client:

AerLabs BV, aviation software company, in collaboration with aviation consultancy company To70.

Coach:

Alessandro Bozzon, department of software technology of faculty of EEMCS, web information systems group.

Contact:

Rolf de Vries: rolf_d_v@hotmail.com.

The final report for this project can be found at: https://repository.tudelft.nl/

Bibliography

- GR Ackermann. Means and standard deviations of horizontal wind components. Journal of Climate and Applied Meteorology, 22(5):959–961, 1983.
- [2] D Bruce Turner. Comparison of three methods for calculating the standard deviation of the wind direction. Journal of climate and applied meteorology, 25(5):703–707, 1986.
- [3] A Garcia, JL Torres, E Prieto, and A De Francisco. Fitting wind speed distributions: a case study. Solar energy, 62(2):139–144, 1998.
- [4] CG Justus, WR Hargraves, Amir Mikhail, and Denise Graber. Methods for estimating wind speed frequency distributions. *Journal of applied meteorology*, 17(3):350–353, 1978.
- [5] Dieudonné Kidmo Kaoga, Serge Doka Yamigno, Danwe Raidandi, and Noël Djongyang. Performance analysis of weibull methods for estimation of wind speed distributions in the adamaoua region of cameroon. *International Journal of Basic and Applied Sciences*, 3(3):298, 2014.
- [6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [7] Nikolaos Kourentzes, Devon K Barrow, and Sven F Crone. Neural network ensemble operators for time series forecasting. *Expert Systems with Applications*, 41(9):4235–4244, 2014.
- [8] Jessica Lin, Sheri Williamson, Kirk Borne, and David DeBarr. Pattern recognition in time series. *Advances in Machine Learning and Data Mining for Astronomy*, 1:617–645, 2012.
- [9] Imran Maqsood, Muhammad Riaz Khan, and Ajith Abraham. An ensemble of neural networks for weather forecasting. *Neural Computing & Applications*, 13(2):112–122, 2004.
- [10] Yukihiro Mori. Evaluation of several "single-pass" estimators of the mean and the standard deviation of wind direction. *Journal of climate and applied meteorology*, 25(10):1387–1397, 1986.
- [11] Raied Salman and Vojislav Kecman. Regression as classification. In 2012 Proceedings of IEEE Southeastcon, pages 1–6. IEEE, 2012.
- [12] KA Verrall and RL Williams. A method for estimating the standard deviation of wind directions. Journal of Applied Meteorology, 21(12):1922–1925, 1982.
- [13] Rudolf O Weber. Estimators for the standard deviation of horizontal wind direction. Journal of Applied Meteorology, 36(10):1403–1415, 1997.
- [14] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*, page 135. Morgan Kaufmann, 2016.
- [15] Robert J Yamartino. A comparison of several "single-pass" estimators of the standard deviation of wind direction. *Journal of Climate and Applied Meteorology*, 23(9):1362–1366, 1984.