UNIVERSITY OF ICELAND

A Neural Network Approach to Predicting Gust Factors in Complex Landscape

Brynjar Geir Sigurðsson

June 2025

M.Sc. thesis
in Mechanical Engineering

FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

A Neural Network Approach to Predicting Gust Factors in Complex Landscape

Brynjar Geir Sigurðsson

60 ECTS thesis submitted in partial fulfillment of a *Magister Scientiarum* degree in Mechanical Engineering

Supervisor
Kristján Jónasson

M.Sc. Committee
Kristján Jónasson
Ólafur Pétur Pálsson
Guðrún Nína Petersen

Examiner
XXNN3XX

Faculty of Industrial Engineering, Mechanical Engineering and Computer Science
School of Engineering and Natural Sciences
University of Iceland
Reykjavik, June 2025

A Neural Network Approach to Predicting Gust Factors in Complex Landscape

60 ECTS thesis submitted in partial fulfillment of a M.Sc. degree in Mechanical Engineering

Faculty of Industrial Engineering, Mechanical Engineering and Computer Science
School of Engineering and Natural Sciences
University of Iceland
Dunhagi 5
107, Reykjavik Iceland

Telephone: 525 4000

Bibliographic information:
Brynjar Geir Sigurðsson (2025) *A Neural Network Approach to Predicting Gust Factors in Complex Landscape*, M.Sc. thesis, Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland.

Reykjavik, Iceland, June 2025

To all the students who made the wise decision to use LaTeX.

Abstract

In many industries, whether it be windmill farming or transportation, being able to precisely predict the load caused by weather is crucial to prevent harm coming to people and machinery as well as increasing productivity. The efficiency of wind turbines in high wind speed conditions is a vital challenge in optimizing renewable energy systems. Gust predictions can be instrumental in loss prevention. If looking only at wind, the most destructive moments are gusts, by definition. To predict these values there are two ways, traditional numerical weather prediction systems and machine learning or other pattern recognition systems. Computer aided numerical weather prediction systems originated in the 1950's, while the use of machine learning in numerical weather predictions is much more recent, having truly gained traction in the last decade. The goal of this thesis is try to combine these two methods to try and predict wind gust factor, where the gust factor is defined as the ratio between wind gust and average wind speed. This is done by having the input data to the machine learning model be reanalysis variables that were generated by traditional numerical weather prediction systems. In addition to these variables and derived variables, a digital elevation model (DEM) data was used. A deep neural network was constructed using some set of variables and the results are compared to a baseline as well as each other so as to show the impact each feature has. Neural networks can help make wind gust predictions in complex ladscape.

Útdráttur

Hvort sem um er að ræða vindmyllubúgarða eða flutningar er spágetan um álag vegna vinds mikilvæg til að koma í veg fyrir tjón á iðnviðum auk þess er hægt að auka framleiðni. Hviðuspár geta verið stór þáttur í tjónaminnkun. Ef einungis er horft á vind, þá eru hviður hæsti álagspunktur út frá skilgreiningu. Sögulega hefur verið notast við hefðbundin spálíkön sem reiða sig á töluleg eðlisfræðileg líkön til að spá fyrir um niðurstöður. Á síðustu árum hefur gervigreind þróast mikið og líkön verið þjálfuð til þess að giska á veður. Svo miklar framfarir hafa orðið að sum líkön geta keppt við hefðbundin veðurlíkön. Markmið þessa verkefnis er reyna að nýta bæði hefðbundin veðurlíkön og gervigreind til þess að giska á hviðustuðla, þar sem hviðustuðullinn er skilgreindur sem hlutfall hæstu hviðu og meðalvinds. Þetta er gert með því að nota endurgreiningargögn úr hefðbundnu veðurlíkani sem grunngögn í gervigreindarlíkan. Einnig eru notað hæðarlíkan til að lýsa umhverfi. Djúpt tauganet var búið til með því að velja breytur úr endurgreiningargögnunum ásamt hæðarpunktum og afleiddum breytum. Þetta líkan er svo þjálfað og niðurstöður bornar saman við grunnlíkan sem og önnur líkön með öðrum breytum til að skoða áhrif breyta. Djúp tauganet geta búið til hjálplegar spár fyrir vindhviður í flóknu landslagi.

Contents

List of Figures

List of Tables

Listings

Abbreviations

API	Application Programming Interface
ASL	Above Sea Level
AWS	Automatic Weather Stations
AWSL	Average Wind Speed Limit
CARRA	Copernicus Artic Regional ReAnalysis dataset
CNN	Convolutional Neural Networks
DEM	Digital Elevaiton Model
ELI5	Explain Like I am 5
ECMWF	European Centre for Medium-Range Weather Forecast
GCM	General Circulation Model
GeoTIFF	Georeferenced TIFF
GPU	Graphical Processing Unit
HRES	High Resolution forecas
IMO	Icelandic Meteoroligcal Office
IRCA	Icelandic Road and Coastal Administration
JNWPU	Joint Numerical Weather Prediction Unit
LWM	Large AI Weather forecast Model
MAE	Mean Asbolute Error

MAPE Mean Absolute Percentage Error

NN Neural Network

NWP Numerical Weather Prediction

SENS School of Engineering and Natural Sciences

TIFF Tag Image File Format

UoI University of Iceland

Acknowledgments

Special thanks go to my advisor, Kristján Jónasson. He would help me with the actual work of the thesis and sit with go over the code with me. Thanks also go to Ólafur Pétur Pálsson, who would read over the thesis with notes and give helpful suggestions. Guðrún Nína Petersen, also provided valuable insight whenever questioned about meteoroligcal matters as well as giving notes on the thesis.

Finally, I would like to thank my parents without whom I would probably never had any want to finish.

1 Introduction

Wind gusts are brief increases in wind speed (lasting seconds) compared to the mean wind speed over intervals of one or several minutes. The gust factor is defined as the peak gust divided by the mean wind speed over a specified time period. The peak wind gust is often defined as the highest 3-second rolling-average wind speed measured over a 10-minute period, while the mean wind speed is the average of all measurements in the same interval. This thesis uses that definition. However, definitions vary: for example, the US uses a 1-minute interval, leading to approximately 14% higher values [**why_wind_gusts**].

The Navier–Stokes equation (1.1) shows that changes in wind, both in time and space, depend on the pressure gradient, the oscillating force of the Earth (the Coriolis force), and frictional forces [**uncertainties_in_numerical_weather_predictions**].

$$\frac{\delta \mathbf{V}}{\delta t} + \mathbf{V} \cdot \nabla \mathbf{V} = \underbrace{-\frac{1}{\rho}\nabla P}_{\text{pressure}} - \overbrace{f\mathbf{k} \times \mathbf{V}}^{\text{Coriolis}} - g - \underbrace{\frac{\delta(u'\omega')}{\delta z} - \frac{\delta(v'\omega')}{\delta z}}_{\text{resistance}} \tag{1.1}$$

Traditionally, numerical weather prediction (NWP) systems are used to forecast and analyze weather patterns [**medium_range_3d_weather_forecasting_NN**]. These models describe the evolution of discretized atmospheric states using partial differential equations grounded in physics. Forecasts are typically produced every hour, or at coarser intervals for climate simulations. With increasing computational power and efficiency, the trend is to output data more frequently [**GNP_vidtal**]. However, such outputs summarize conditions over each period and may not capture short-term fluctuations well, including variations in wind speed and gusts [**canNNBeatNWP**].

This thesis examines methods to predict the gust factor using various predictors and multiple data sources, including NWP outputs and observational data. Accurate gust predictions are crucial, as it is ofteh the peak wind gust that will cause damaging incidents, such as structural failures, traffic accidents and downed power lines. Moreover, the prevalence of extreme wind events is expected to increase in the future [**nasa_extreme_weather**].

1.1 Background

Numerical weather prediction

The history of numerical weather prediction dates back to the 1920s, when Lewis Fry Richardson pioneered the field and attempted to produce forecasts. His results were flawed due to numerical noise. The ENIAC, built in 1945, was a general-purpose computer used—among other tasks—for weather prediction. These forecasts took 24 hours to compute and predicted 24 hours into the future. While a proof of concept, they were not operationally useful [**TheENIACForecastsARecreation**].

With the advent of electronic computers in the 1950s, the first operational forecasts emerged. In September 1954, Carl-Gustaf Rossby and his Stockholm-based team produced the first real-time barotropic forecasts. The following year, the Joint Numerical Weather Prediction Unit (JNWPU), based in Princeton, New Jersey, released its first 36-hour forecasts at 400, 700, and 900 mb. Although these forecasts were inferior to subjective human analyses, they demonstrated feasibility and spurred further development in the field [**historyNWP**]. Since then, NWP has made tremendous strides in parallel with increases in computational power and efficiency.

AI in weather forecasts

In the last decade, there has been another transformation in weather prediction driven by artificial intelligence (AI). Interest in AI has come in waves: progress is made, then interest diminishes, but over the past 15 years growth has been steady. Notable drivers of this wave include advances in computational power (notably parallel processing on graphics processing units, GPUs), the developement of neural networks (NNs) for processing massive datasets, and the availability of large online datasets. One type of NN is the convolutional neural network (CNN), originally designed for image processing but applicable to any gridded data with spatial structure [**canNNBeatNWP**]. Since 2018, significant work has applied AI to weather prediction.

There are two common approaches to AI-based forecasting: combining NWP outputs with NNs, or using NNs alone. In the former, NWP forecasts can be used as inputs to NN training or integrated in other ways; in the latter, NNs are trained directly on meteorological observations, bypassing NWP entirely.

In 2018, Düben and Bauer showed that an NN could outperform a simple per-

sistence forecast and compete with very coarse-resolution atmospheric models of similar complexity for short lead times [**dueben2018**]. Also in 2018, Scher developed a deep CNN to emulate a general circulation model (GCM)—a numerical model representing physical processes—by training on GCM output. This approach allowed stable emulation of model dynamics for much longer horizons [**scher2018**]. These papers were proofs of concept rather than production-ready replacements for NWP, but they demonstrated that deep-learning–based models could, with further development, compete with standard models in the field.

Large weather models

In the last two years, there have been even more developments with the emergence of Large AI Weather forecast Models (LWMs). In 2024, Ling et al. [**SecondRevolution**] proposed a standardized definition of LWMs in meteorology, outlining three criteria, referred to as the "Three Large Rules":

1. **Large parameter count:** Typically ranging from tens of millions to billions of parameters.

2. **Multiple predictands:** Forecasting at various levels (e.g., pressure or height levels) to provide detailed atmospheric vertical structure and surface conditions.

3. **Scalability and downstream applicability:** Demonstrated, for example, by predicting cyclones even when not explicitly trained on cyclone data (e.g., GraphCast) to showcase model versatility [**SecondRevolution**].

Before 2022, LWMs had been shown to compete with traditional NWP in specific cases and to generate forecasts much faster after training, No model had been shown to be able to completely replace tradional NWP systems. In early 2022, Pathak et al. [**FourCastNet**] introduced FourCastNet, which employs an Adaptive Fourier Neural Operator model leveraging transformer architecture instead of convolution. FourCastNet matches the performance of standard forecasting techniques at short lead times for large-scale variables and outperforms them for smaller-scale features. It generates a one-week forecast in under two seconds—orders of magnitude faster than conventional physical methods [**FourCastNet**]. In 2022, several ML-based models demonstrated faster-than-NWP predictions after one-time training, with performance rivaling or exceeding NWP in some cases.

In 2023, Remi Lam and the GraphCast team at Google released GraphCast, which outperformed the European Centre for Medium-Range Weather Forecasts' (ECMWF)

industry-standard High-Resolution Forecast (HRES). GraphCast uses a graph-based representation instead of a regular grid, operating on a global latitude–longitude grid at 0.25° resolution, which introduces nonuniform point spacing near the poles; the graph structure helps mitigate this bias [**GraphCast**].

Conclusion

Substantial progress has been made since 2018 and especially since 2022 [**SecondRevolution**]. The evolution from proof-of-concept ML methods to models competitive with—or surpassing—standard NWP has been remarkably rapid. It is important to note that training data for these large models are drawn from traditional NWP outputs, underscoring the synergy between ML and physics-based approaches. The coming years promise further exciting developments in machine-learning–based weather prediction.

1.2 Methodology and related work

In this study, data from three sources is used:

1. Three-hourly reanalysis data for Iceland (2004–2023)

2. Dense (20 by 20 m) gridded elevation data for Iceland

3. Three-hourly observations from Icelandic meteorological stations

The aim of the study is to improve prediction of gust factors that can be applied at any given place in Iceland. Currently, gust factor predictions can be obtained directly from NWP output, but one may also create a simple regression model (involving, e.g., wind speed and location) to predict them. To accomplish this, an NN is used as a backend to the NWP. The network is trained with gust factors from observations as ground truth, and model quality is measured by comparing predictions with either the NWP or the regression values. Note that only data at the prediction time is used; the time-series aspect of the data is not utilized.

In 2004, Hálfdán Ágústsson and Haraldur Ólafsson [**mean_gust_HA_HO**] investigated gust factor variability in complex landscapes. They used data from automatic weather stations measuring wind at 10 m above ground during 1999–2001. They examined how three parameters affected the gust factor: d_m, D, and H—the wind direction relative to a mountain, the distance to the mountain, and the height

of the mountain above the station. Their main results showed that the gust factor is inversely correlated with distance and directly correlated with mountain height. Ágústsson and Ólafsson considered the effect of a dominant point upwind but did not consider the broader landscape.

1.2.1 Neural networks

To capture patterns in the data, a neural network was constructed. An NN architecture was chosen because NNs handle complex data well and can manage large numbers of parameters. This flexibility is useful when training on different types of data and allows testing various network designs. An NN uses many matrix calculations to weight input parameters and predict an output. A deep neural network (DNN) has an input layer, an output layer, and one or more hidden layers. Hidden layers transform data from the input layer until it reaches the output layer. The input layer width equals the number of input parameters, and the output layer width equals the number of predictands. Models were created with different parameter sets to gauge each parameter's influence. Each model produces a single numerical output. Every layer also has an activation function; common choices include Rectified Linear Unit (ReLU), Exponential Linear Unit (ELU), and hyperbolic tangent (tanh), defined in Equations (1.2), (1.3), and (1.5).

$$r(x) := \max(0, x) \tag{1.2}$$

$$f(x) := x, \ x > 0 \tag{1.3}$$

$$f(x) := \alpha(e^x - 1), \ x \leq 0, \ \alpha > 0 \tag{1.4}$$

$$\tanh(x) := \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{1.5}$$

These activation control neural activation and can help stabilize the network.

1.2.2 Model evaluation

To measure the performance of these models, both in training and testing, mean absolute percentage error (MAPE) as defined in Equation (1.6) was used.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_{\text{predict}} - y_{\text{true}}|}{y_{\text{predict}}} \tag{1.6}$$

This measure was chosen because the target is the gust factor (the wind gust over the average wind). If the target had been the wind gust rather than the gust factor, mean absolute error might be more appropriate.

1.2.3 Model explainability

Neural networks are often considered mysterious black boxes [**nn_black_box**]. To understand model predictions, explainability methods are used. One such method is Shapley values [**shapley_information**]. Shapley values are calculated as the average marginal contribution of a feature value across all possible coalitions. The contribution for a single feature j to a prediction $\hat{f}(X)$ is given by Equation (1.7). In Equation (1.7), x_j is the feature value, β_j its weight, and $\beta_j E[X_j]$ the mean effect estimate for feature j.

$$\phi_j(\hat{f}) = \beta_j x_j - \beta_j E[X_j] \tag{1.7}$$

For any combination of parameters, Shapley values explain the individual prediction by attributing a contribution to each feature. Other methods, like ELI5 (Explain Like I'm 5), randomly shuffle a feature and measure the effect on model performance [**eli5_information**].

2 Data gathering and processing

Data were sourced from several streams. The Icelandic Meteorological Office (IMO) provided measurements from weather stations across Iceland, NWP data were downloaded from the Copernicus Arctic Regional Reanalysis dataset (CARRA), and finally a land-elevation model was also provided by the IMO.

2.1 Automatic Weather Station Data

IMO provided measurements from 327 weather stations across Iceland. The measurements that met the filtering criteria began in 2004 and ended in 2023. Of these 327 stations, 212 were from the IMO, with the anemometer at 10 m above ground, while the remaining 115 stations were from the Icelandic Road and Coastal Administration (IRCA), with the anemometer at 6–7 m above ground [**vegagerdin_postur**]. The locations of these weather stations are shown in Figure 2.1.

Data from these automatic weather stations (AWS) are stored in hourly files, which aggregate the original 10-minute files; measurement errors—unrealistic spikes known as "nails"—have been removed in most cases. Each record contains the following information: date and time; station number (convertible to coordinates using another dataset of Icelandic meteorological stations); average wind speed (f); wind gust (f_g); standard deviation of the wind gust; wind direction (d); and standard deviation of the wind direction.

These measurements began at the end of the 20th century with the installation of the first AWSs, and more stations have been added in subsequent decades.

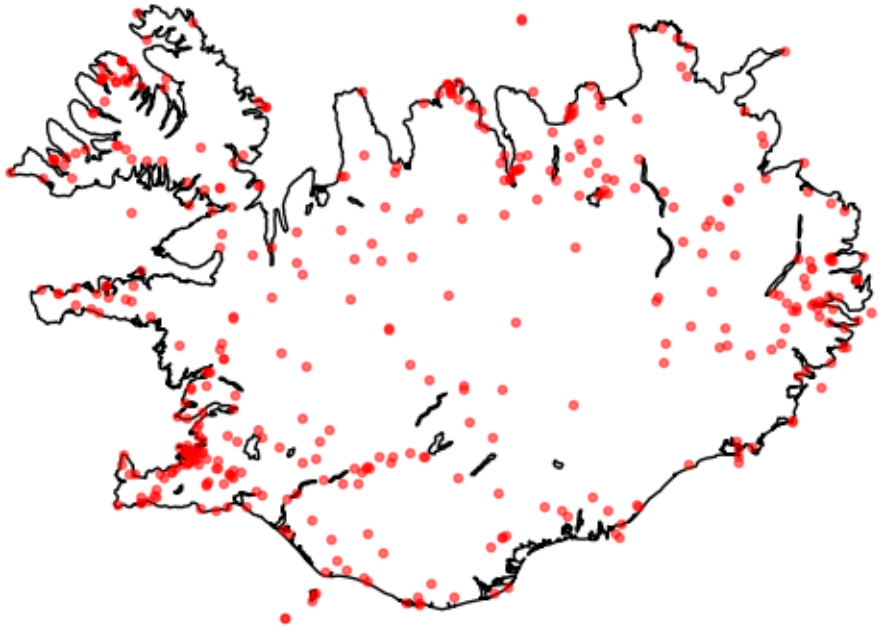

Figure 2.1: Locations of all 412 stations that were looked at in this study. Most of these were from IMO but over a hundred were from IRCA. IMO anemometers are placed at 10 meters above ground, while IRCA ones are placed at 6–7 meters above ground.

2.2 CARRA Data

CARRA is a high-resolution atmospheric reanalysis produced by the Copernicus Climate Change Service and run by ECMWF. It covers two regions, a west region covering Greenland and Iceland and an east region covering the European Arctic. It has a 2.5 km horizontal resolution and dates from 1991 to the present, with monthly updates. CARRA provides three-hourly analysis fields and short-term forecasts (hourly for lead times under 6 h and three-hourly beyond) of surface and near-surface variables—wind, temperature, pressure, precipitation, etc. It is based on the HARMONIE-AROME limited-area NWP model, forced at its boundaries by ERA5 (ECMWF Re-Analysis v. 5) and enhanced by local observations to better represent complex terrain, land–sea contrasts, and sea-ice processes. It is updated monthly, with a latency of 2-3 months[**carra_information**].

The CARRA dataset covers all the IMO observations that fulfill criteria of consistent availability – the oldest observation is from 2004. The CARRA-West region covers a vastly larger area than the area of interest. This leads to having to store a large amount of data. To download CARRA data one has two options, a web interface or using an API client provided by CARRA. Using the API client is the only realistic option here, as there are thousands of requests made for different times. If using the API, it is possible to query a smaller area (such as a rectangular area around

Iceland) given a set of coordinates, but this is not possible with the web interface.

The requests to the API were made at each available CARRA hour ([00, 03, 06, 09, 12, 15, 18, 21]) on a grid covering Iceland, for each available observation time. The downloaded data were interpolated to get values at the weather stations. CARRA contains several types of layers: single levels, model levels, height levels, and pressure levels. The data for this thesis was downloaded from height levels. They were requested at heights of 15, 250 and 500 meters above ground. For each point 4 parameters were requested, wind speed, wind direction, pressure, and temperature.

2.3 Elevation data

A GeoTIFF file containing Iceland's elevation on a 20 m by 20 m grid was provided by the Icelandic Meteorological Office (IMO). The entire country is covered by this file, and its size is approximately 685 MB.

The Python package `rasterio` is used, enabling rapid elevation lookup via its spatial indexing and affine-transform capabilities. Elevation at specified geographic coordinates can be retrieved directly, and grid indices may be used for efficient access. Elevation for any exact location can be interpolated by fetching neighboring grid points.

2.4 Combining data sources

Three main data sources were used, each requiring querying, filtering, and merging to prepare the combined dataset. When handling hundreds of thousands of rows, code efficiency is essential: row-by-row iteration can increase execution time dramatically compared to vectorized operations.

The sources were provided in different formats: IMO measurement data in text files, elevation data in GeoTIFF, and CARRA reanalysis data in GRIB. To train the models, these datasets were combined into a single file using the IMO measurements as a reference. CARRA data are supplied on a rectangular grid with approximately 2.5 km spacing, while IMO observations are tied to specific station locations. Elevation data are on a 20 m by 20 m grid covering Iceland. Linear interpolation was applied to merge the sources.

The merging procedure, shown in Figure 2.2, was as follows: for each AWS observa-

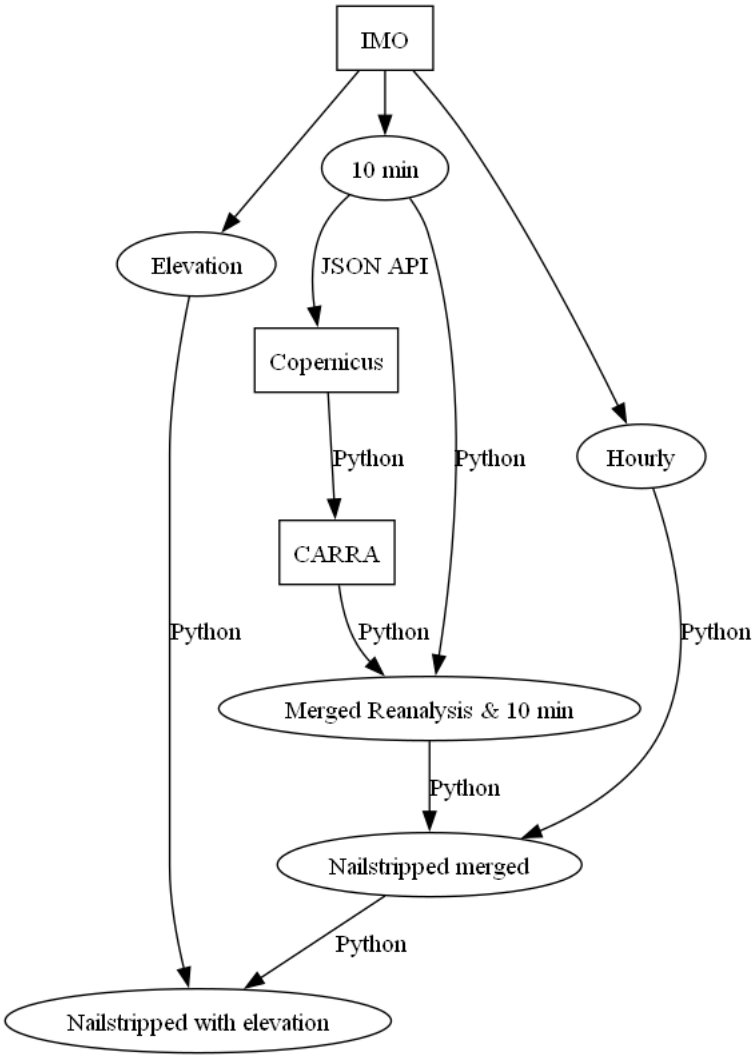

Figure 2.2: Flow chart illustrating the data-combination workflow

tion, a query was constructed for the CARRA API by specifying year, month, day, hour, and spatial extent. Because the API returns all specified days when queried by hour, and all specified months when queried by day, monthly queries were issued for only the required days, retrieving all eight three-hourly time points (00, 03, 06, 09, 12, 15, 18, and 21 UTC). After downloading the requested variables at the desired pressure levels, point values were interpolated and appended to a pandas dataframe. The monthly GRIB files were then discarded before proceeding to the next month. This strategy reduced storage needs from several terabytes to under one gigabyte.

Elevation values from the GeoTIFF were interpolated in the same manner: the four surrounding grid points were used in a linear interpolation to estimate the elevation at each station location. These interpolated values were included in the dataframe, since topography influences both average wind speed and gustiness [**GNP_vidtal**].

2.5 Comparison of observed and reanalysis wind speed

The error in reanalysis wind speed and measured wind speed can be significant. The absolute error increases as the measured wind speed increases, while the percentage wind speed decreases. A grouping of these errors by wind speed can be seen in Table 2.1

Table 2.1: Comparison of measured and reanalysis wind speed using mean absolute error (MAE) and mean absolute percentage error (MAPE). Note that for the computation of MAPE for ranges that otherwise include 0, 0 values have been excluded so as to prevent division by zero and exploding values. The comparisons are done using measured wind speed (at 10 meters above ground for IMO and 6-7 meters above ground for IRCA) and reanalysis wind speed at 15 meters above ground.

f	n	MAE
$[0; 5[$	6.2e6	2.1
$[5; 10[$	4.2e6	2.2
$[10; 15[$	1.5e6	2.5
$[15; 20[$	3.9e5	3.0
$[20; 25[$	8.4e4	4.0
$[25; \infty[$	2.0e4	6.6
$[0; \infty[$	1.2e7	2.2

Another thing to look at is the distribution of error by station, both in terms of their coordinates and number of measurements. Looking at Figure 2.3, this distribution can be seen.

Table 2.2 shows the 5 best and worst stations in terms of MAE.

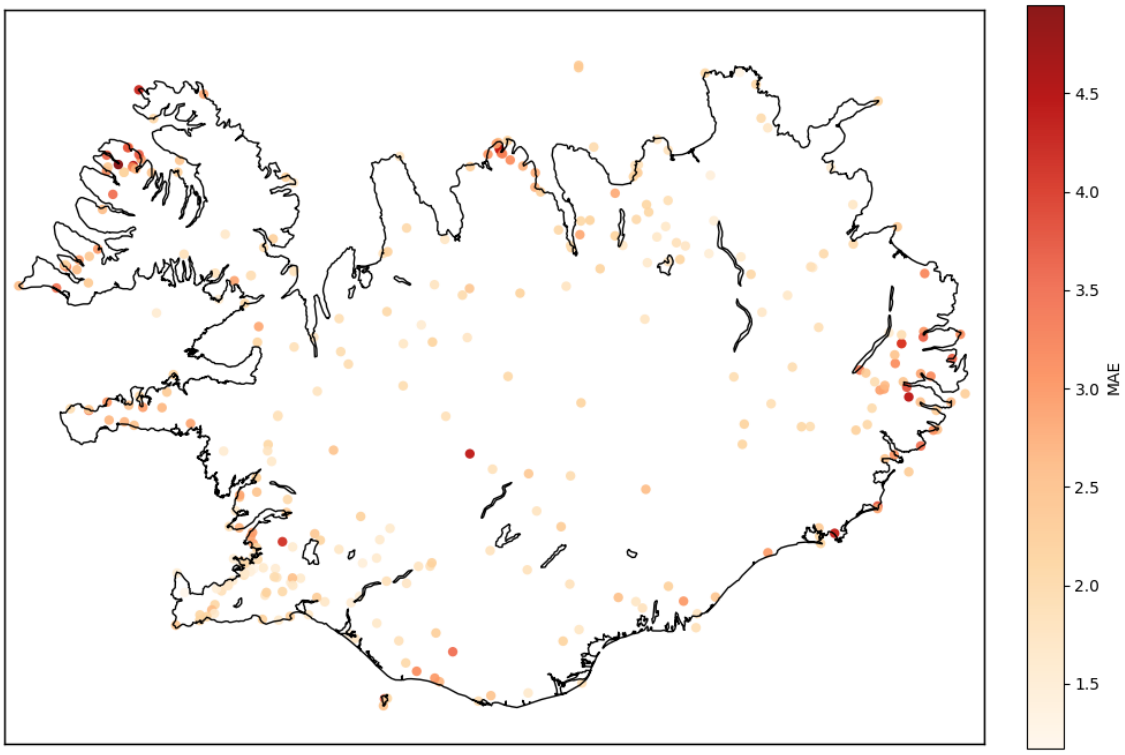

Figure 2.3: The distribution of mean absolute errors by station. Using mean absolute error instead of mean absolute percentage error allows for all points to be used. Mean absolute percentage error can only be used if 0 values are ignored.

Table 2.2: Mean absolute error for reanalysis wind speed as compared to measured wind speed, for the five stations with the highest difference and the five stations with the lowest difference.

Station	Number of measurements	MAE	Location
1470	6.8e3	1.17	Reykjavík Háahlíð
1350	5.2e4	1.18	Keflavíkurflugvöllur
1482	1.4e4	1.23	Reykjavík Víðidalur
4921	1.3e4	1.29	Rif á Melrakkasléttu
1477	5.6e4	1.29	Reykjavíkurflugvöllur
35553	4.0e3	4.30	Almannaskarð - göng
6745	1.5e4	4.36	Kerlingarfjöll - Ásgarðsfjall
35978	7.9e3	4.40	Fáskrúðsfjarðargöng suður
2640	1.6e3	4.51	Seljalandsdalur
32635	3.2e4	4.95	Botn í Súgandafirði

2.6 Data Structure

Once data has been retrieved for all three sources and processed, including interpolating values, it needs to be made ready to use by the model, for both training, validation and test. The starting point is a dataframe that contains measured information from AWS. This includes the average wind, the wind gust, wind direction along with the station number and coordinates. When selecting the CARRA data certain height levels are chosen. These present as separate lines in the CARRA dataframe. Information for one observation is represented in as many lines as height levels requested in the reanalysis data. These rows need to be combined on the position (the weather station). When this is done it is possible to combine the AWS IMO data and CARRA reanalysis data on the location and time columns. The last data source is the elevation. A circle sector upwind is looked at. In any case the points, that represent these sections, were selected as shown in Code Listing 2.1. A range of angles are defined based on the wind direction d at some distance from the given point. This means that the resultant points (equal in number to the length of angleRange by k) from arcs at several distances from the given weather station.

Listing 2.1: Sector elevation points generated

```
angles = [(angle + (90 - d)) * pi/180 for angle in angleRange]
length_rng = [(exp(i * log(n + 1)/ k) - 1) * 1000
                  for i in  range(1, k + 1)]
points = np.array([[(X + l * cos(angle), Y + l * sin(angle))
                      for angle in angles] for l in length_rng])
```

The result is a dataframe that has measured data from AWS, which gives us our target, reanalysis data from CARRA, which gives us weather variables to train on, and finally elevation points in the landscape to include in our training data. An example of what the data looks like can be seen in Table 2.3.

Table 2.3: An example of data structure used to train model. Data points include the derived variables Richardson number (Ri) and Brunt-Väisäla frequency (N) (defined below), the elevation of the station, direction of wind and relative direction of the wind (twd, that is the direction of the wind relative to center of Iceland), along with some combination of wind speed, pressure and temperature at the different height levels. Finally there are the elevation points around a given station, where the elevation is relative to the station.

Ri	N^2	station elevation	twd	ws_{15}	wd_{15}	t_{15}	p_{15}	$elevation_0$	...
-1.18e+00	2.67e+04	100	1.5	10	5	0	100	2	...

Looking at Table 2.3 note that the first two columns represent two variables that describe the stability of the air. These are the Richardson number (Ri)[**richardson_number_skybr**
and Brunt–Väisälä frequency (N)[**brunt_vaisala_freq_eumtrain**], and are calculated using Equations (2.1) and (2.2)[**mean_gust_HA_HO**]. These values are calculated using reanalysis data at two different height levels. Thus Ri refers to the Richardson number calculated between height levels 15m and 500m. Exactly the same notation is used with the Brunt–Väisälä frequency, except the square is used.

$$Ri = \frac{g \cdot dT \cdot dz}{T_{ave} \cdot dU^2} \, [] \tag{2.1}$$

$$N = \sqrt{\frac{g \cdot dT}{T_{ave} \cdot dz}} \, [\text{Hz}] \tag{2.2}$$

Here, g is the acceleration due to gravity, dT is the temperature difference between the two height levels, dz is the elevation difference, T_{ave} is the average temperature (that is the average of the two temperatures in the height levels) and dU is the wind speed difference between the two height levels. Both of these numbers provide some insight about the stability of the air. A lower value for the Richardson number indicates a higher turbulence. A typical range of values could be between 0.1 and 10, with values below 1 indicating significant turbulence[**richardson_number_skybrary**]. When the square of the Brunt-Väisälä frequency is negative, then the air is unstable (an air parcel will move away from its original position)[**brunt_vaisala_freq_eumtrain**]. These are derived factors from the reanalysis data and as such there shouldn't be a significant information gain using Ri and N as opposed to having the raw data. However, including these factors instead of every reanalysis variable requested

might speed up training as well as making the model more easily explainable with the use of Shapley values or other tools for explainability. Using Shapley a feature importance value is attributed to a given feature by creating all possible permutations of any possible length (up to number of features) and seeing how the predictions are skewed when the given parameter is included or excluded. This needs to be done for all parameters. The time complexity of this is very high (2^n coalitions)[**shapley_information**]. Most implementations use some approximations, which still can take a considerable amount of time for models with a high parameter count and many examples. The Richardson number includes the difference in wind speeds in the denominator. In certain cases, where the difference in wind speed between two levels is very, can blow up to infinity. This can cause problems and distort the predicitons.

2.7 Data distribution

The CARRA data is reanalysis and as such might have a bias or some systemic distortion when compared to the measured data. The distribution of the observed and reanalysis wind speed can be seen in Figure 2.4. Looking at the figure, the reanalysis wind tends to be higher even though the distribution is similar.

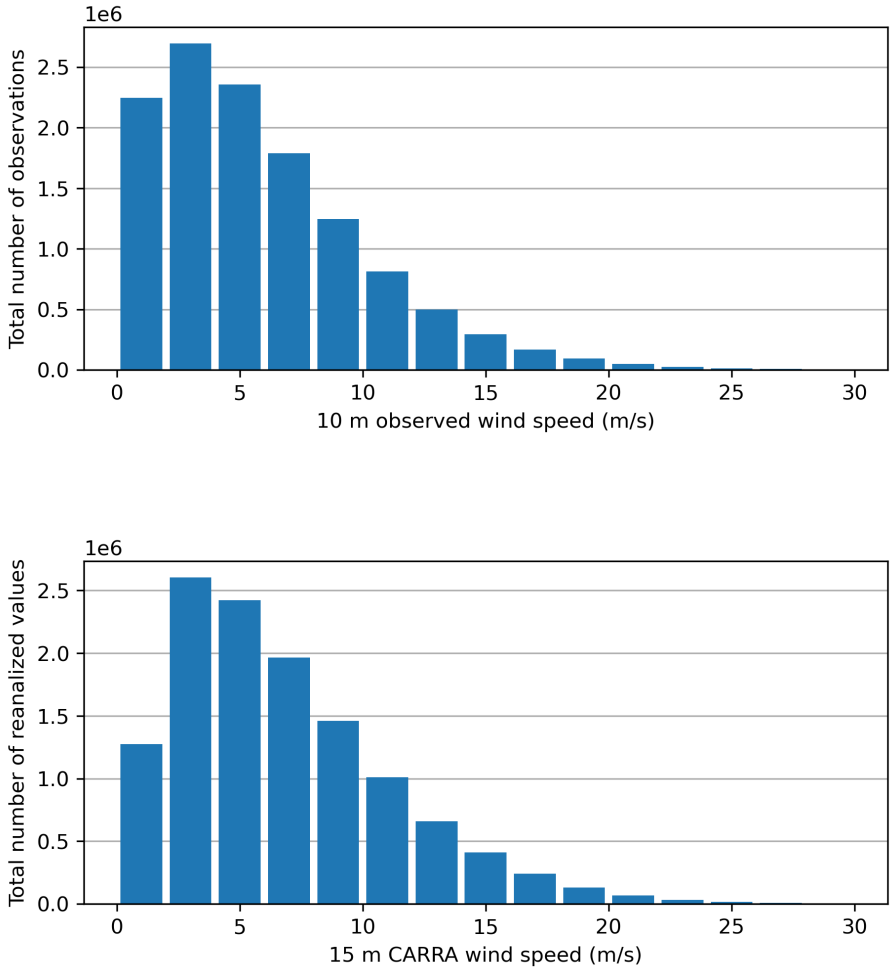

Figure 2.4: Upper figure shows a histogram of observed wind speeds provided by IMO and IRCA for all used weather stations. Lower figure shows the interpolated CARRA reanalysis values at weather stations. The reanalysis data is only available at 3 hour intervals and 2.5km grid. As such the observed values are only chosen at these specific times (00, 03, 06, ..., 21). Spatial interpolation is needed and is linearly weighted based on distance of CARRA grid points from stations.

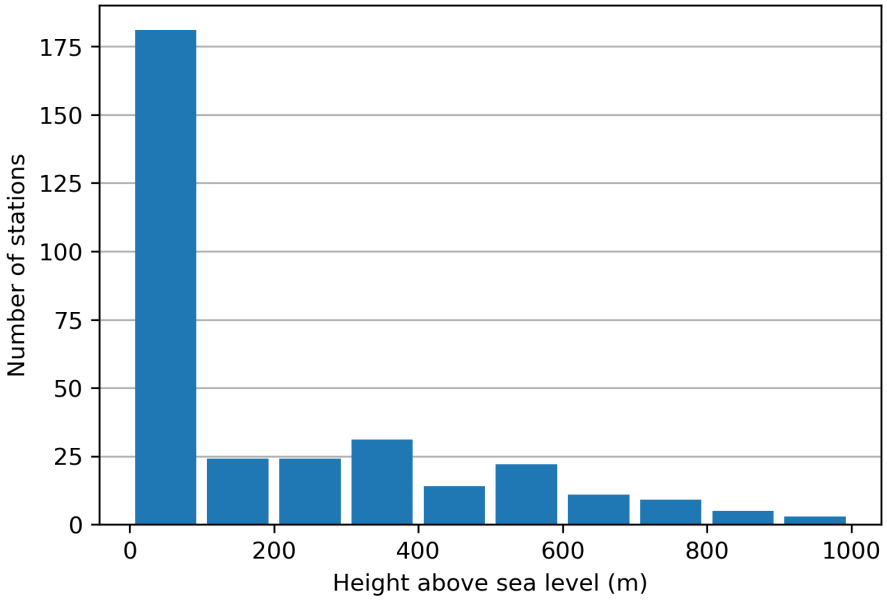

Figure 2.5: Distribution of heights of weather stations above sea level. One station has been excluded as an outlier of well above 1000 meters, having few and inconsistent datapoints.

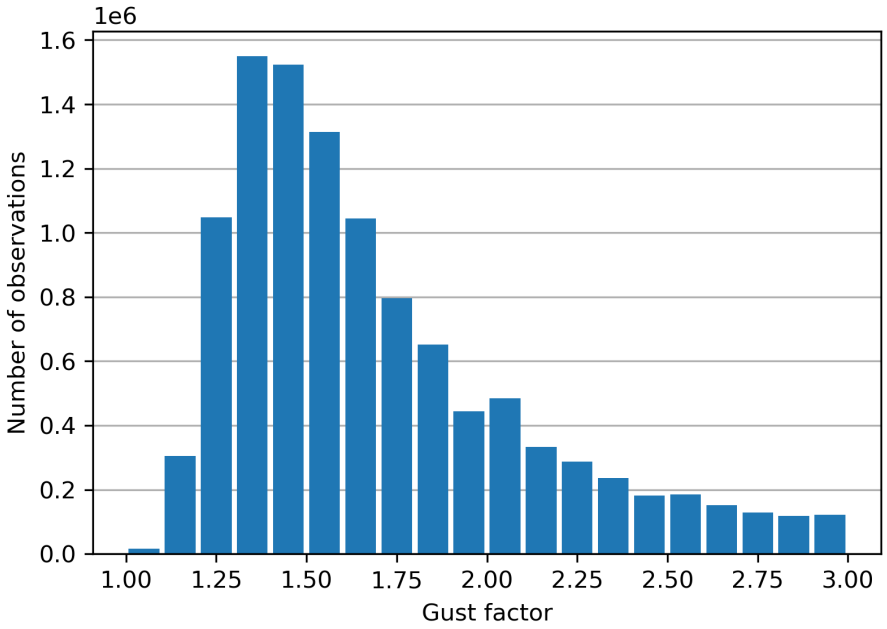

Figure 2.6: Histogram of gust factors. By definition the lower bound of gust factor is 1. The majority of observed gust factors fall in the range 1.2 to 2. Gust factor decreases with increasing wind speed.

3 Model Architecture and Training

3.1 Model structure

The structure of the neural network is such that it contains some number n of fully connected layers and batch normalization for each layer, along with regularization. All of these layers have the same number of units. The last layer has a dropout of 50%. In addition to these layers there is one more output layer. This is simply a dense layer with 1 unit. A grid search was performed to determine the hyperparameters that minimize the loss. Hyperparameters are parameters that are set before the training begins[**hyperparameters_definition**]. These hyperparameters include number of units in each layer, number of epochs to train for, number of layers, batch size, optimizer and penalty to enforce in the regularization. The possible combinations tried can be seen in Table 3.1.

Table 3.1: Hyperparamter search with best performing combination shown. Hyperparameter search was done using hyperband algorithm that initially searches randomly for the best parameters but then hones in on what is working and as such is neither exhaustive nor completely random. This means that a hard upper limit will not be set on the number of combinations to try like with randomsearch.

Parameter	Range of values	Selected
Layers	min_value = 4, max_value = 15, step = 1	**10**
Units	min_value = 32, max_value = 512, step = 32	**64**
Penalties	min_value = 1e-5, max_value = 1, sampling = log	**1e-4**
Epochs	min_value = 10, max_value = 1000, step = 10	**250**
Optimizers	Adam, RMSprop, Adamax	**Adamax**
Activation	ReLU, ELU, tanh	**ReLU**

As mentioned, hyperband doesn't set a hard upper limit to the number of epochs it will train in total. When using the hyperband class several factors can be set. One of interest here is the $hyperband_iterations$ argument. This determines how often the hyperband algorithm is run and defaults to 1. For each iteration the epochs are distributed between tries (that is each set of hyperparameters) with the total amount of epochs approximately $n_{epochs} = max_{epochs} * log^2(max_{epochs}) \approx 10^4$,

where $max_{epochs} = 1000$ gives the maximum number of epochs that one set of hyperparameters can be trained for. Searching a space generally takes a lot of time but this drastically improves on gridsearch. If each epoch takes around 10 seconds to run then the total search would take around 28 hours on a shared resource. This is resource intensive and cannot be repeated often. Another question that remains is whether the ranges given are optimal.

3.2 Model Training

To determine the required number of epochs a high number of epochs can be selected and the loss plotted. These plots can be seen in Figures 3.1 and 3.2. Looking at these two plots, training loss decreases for over 1900 epochs in both plots, while the validation loss stops decreasing at around 200 epochs for DEM model and 1800 epochs for model without DEM. This is unexpected. The DEM model has vastly more parameters and as such should contain more useful information to be learned, which one might expect to take longer to learn. This does not seem to be the case. The DEM model only gets trivially better after 1000 epochs while the non DEM model keeps meaningfully improving until around 1800 epochs. With respect to time constraints of shared resources models were trained for 250 epochs.

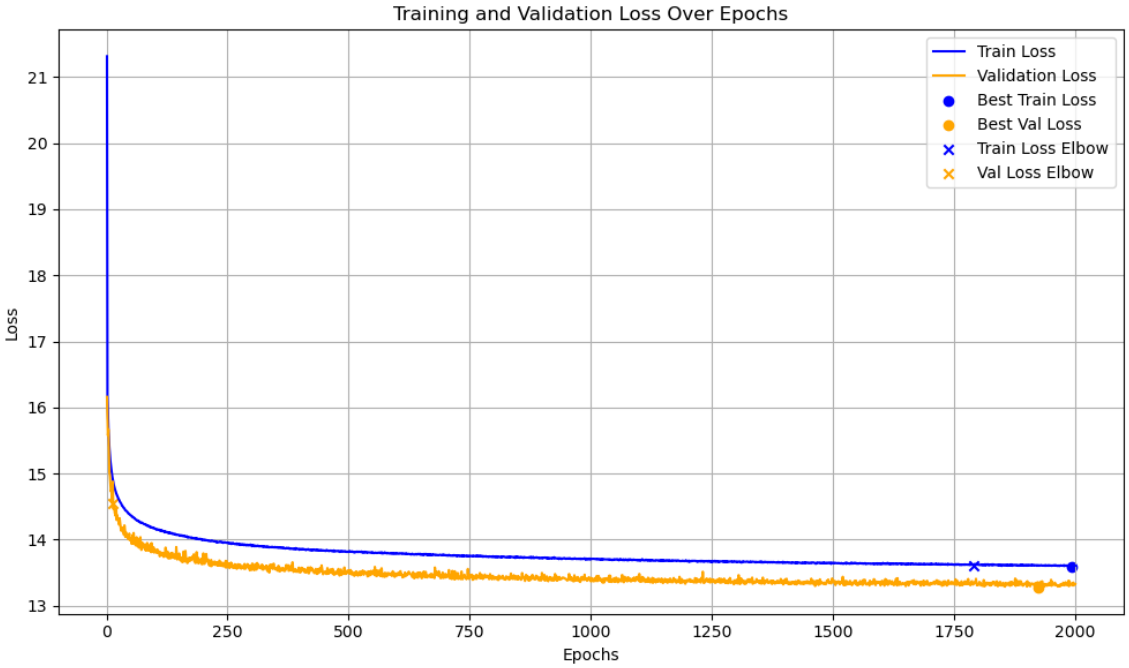

Figure 3.1: Loss plot of model trained for 2000 without DEM.

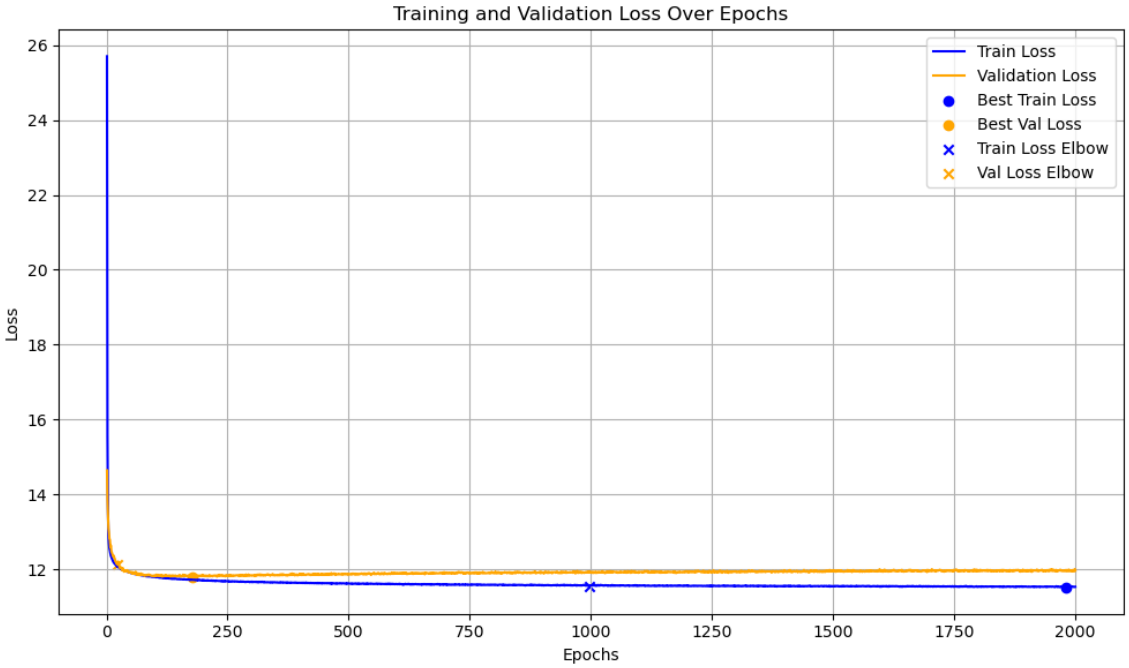

Figure 3.2: Loss plot of model trained for 2000 with DEM.

4 Results

4.1 Results

A baseline model was constructed. This model looked at the gust factor for some training data and took the average of this and predicted this average everytime. Several baseline guesses were created based on a lower limit assigned to the average wind speed limit (AWSL). As the average wind speed increases, then the variability in gust as a percentage decreases[**mean_gust_HA_HO**]. This means looking at a subset of the data where the AWSL is higher, a better result can be expected. The results show this. Several different AWSL were used for the baseline model, as can be seen in table 4.1. This sets a goal. A model that does not significantly improve on this baseline suggests either failure to capture essential patterns in the data or that the data itself may lack the necessary information for substantial improvements upon the baseline. Using the previously described neural network architecture setups for each AWSL, with and without landscape elevation information, MAPE was determined. The results can be seen in Table 4.1.

Table 4.1: MAPE for each average wind speed limit with and without landscape elevation in a 30° sector around the point of interest into the direction of the reanalysis wind. The influence of adding elevation data seems to reduce the error. The percentage error is higher for lower wind speeds and thus observing the error for different lower bound of wind speed will produce different results. This lower bound is determined using the reanalysis wind speed at 15 meters (ws_{15}).

AWSL	**MAPE**		
$[m/s]$	*Baseline*	*Without DEM*	*With DEM*
≥ 0	39.2%	19.2%	18.9%
≥ 5	28.1%	15.3%	14.9%
≥ 10	23.9%	13.3%	12.5%
≥ 15	23.2%	14.4%	11.9%
≥ 20	24.7%	15.7%	13.3%
≥ 25	27.7%	19.4%	17.3%

This is some improvement upon the baseline error, with a decrease in error from 23.9% to 13.6% and 10.6% for the baseline, model without DEM and with model

with DEM for 10 m/s cutoff. The power generated by wind mill increases with wind speed cubed[**wind_power**]. The highest wind gusts in Iceland are around 70 m/s. Knowing the gust factor with half as much error as before can allow better anticipation and thus spare turbines for high wind gusts. Another way to look at the error improvement is by station. No location data was directly included in the training data. In Table 2.2 the mean absolute error of predicted average wind speed and measured average wind speed can be seen for the extreme values. A question to ask is it possible to achieve better results when only looking at a single station?

Table 4.2: The MAPE results for selected stations of interest, both when training for the specific site and when the stations are a part of the general data. For every station the AWSL is set at 10 m/s. In training for a single station at a time, some site specific information can be gauged. This does not mean that the a better result can be reached for that site. Factors such as the number of datapoints at given location can significantly impact the result. This table uses the measured wind speed to determine the cut off for data points. This leads to some data leakage and an increased performance compared to using the reanalysis CARRA speed for cut off.

Station name	**# points**	**MAPE**	
		General training	Site training
Akrafjall	42,791	18.6%	93.7%
Almannaskarð	4,014	12.2%	86.7%
Ásgarðsfjall	15,121	9.1%	9.4%
Jökulheimar	17,176	7.7%	7.7%
Sandbúðir	18,718	6.8%	6.4%
Stórholt	35,126	7.1%	29.2%
Þúfuver	19,538	6.4%	6.8%

Instead of looking at the exact values of MAPE at select stations, a plot of the error distribution can be created. This can be seen in Figure (4.1). Looking at Figure (4.1), the worst performing stations can be seen at Vestfirðir and around the coastline. Stations further inland seem to have lower error. The worst performing station, at Seljalandsdalur, is at Vestfirðir while the best performing station Garðskagaviti is at the South-Western tip of Iceland in Reykjanes.

Table (4.3) shows no improvement over the values in the last column in Table (4.1). As previously mentioned, the gust factor decreases with increasing wind speed and thus, training on intervals and lessening this effect might be expected to give better results. This does not seem to be the case. Some interesting sites to look closer at for drivers might include places such as Ingólfsfjall, Kjalarnes and others. These can be seen in Table (4.4).

In Figure (4.2) the contribution of each feature, excluding the elevation points, can be seen for the model in general (a significant subset of data is used). Looking

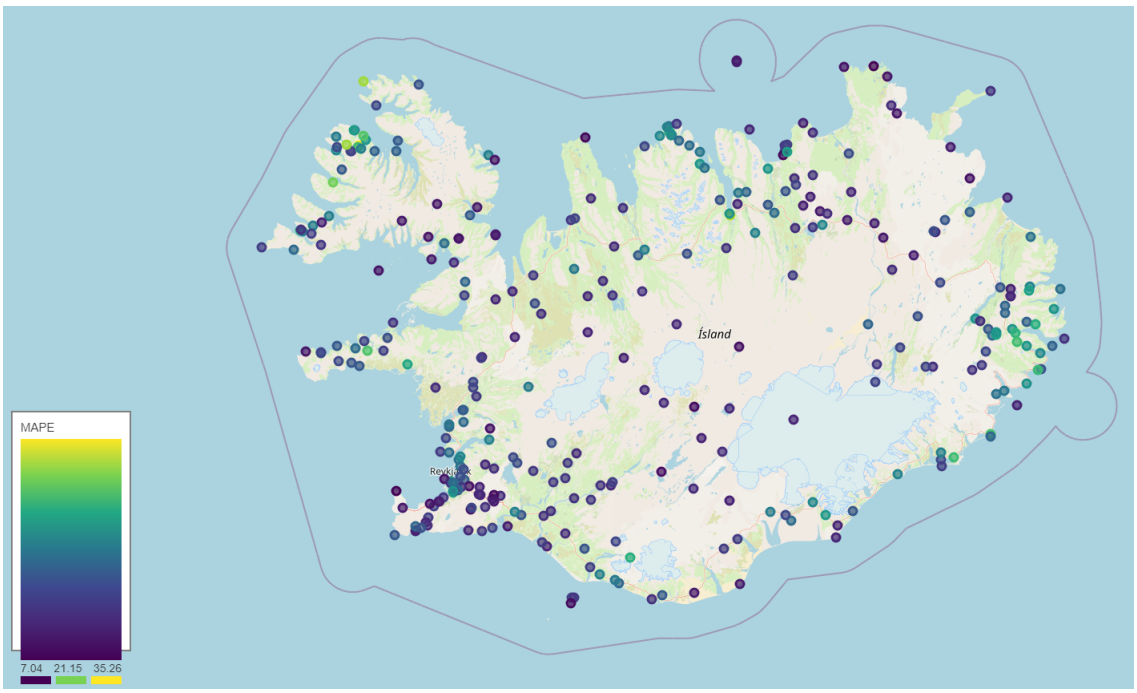

Figure 4.1: The MAPE error of each station in data shown as color gradient circles. That is each station is represented by one circle, with the error value represented as a color gradient from dark blue to yellow. The lowest error at a single station was around 7% at Garðskagaviti and the highest around 35% at Seljalandsdalur. It is important to note that the model is trained using a cutoff of 10 m/s and this cutoff point is determined using the reanalysis wind at 15 meters above ground.

at Figure (4.2), there is an outlier. Exactly calculating the Shapley values is time intensive, in the subset shown there is an outlier that skews the figure and makes it so that viewing the importance distribution excluding the outlier is difficult. For this reason, another shapley summary vas created that looked at different, and larger distribution. This can be seen in Figure (4.3). The Figures (4.2) and (4.3) show the summary for a model trained only on the features shown and not on landscape elevation of the surrounding area. This is done as the number of points there is too high to display in one figure (70 total points). Looking specifically at Figure (4.3), the station elevation is most influential and the Richardson number's influence is very low. Most of the feature values bunch up in the middle, while the station elevation is elongated compared to other features. Station elevation seems to have two bunches on either side of 0. Overall, the values seem to be skewed to the right of 0. This would be expected as the predicted values are expected to be in the range of 1.2-2, or at least always above 1 by definition. Finally, for the Shapley values, looking that all the data there are again outliers that skew the data so that spotting general distribution is difficult. This can be seen in Figure (A.4).

Further Shapley figures can be seen in Appendix 5. In each of these plots, even

Table 4.3: The MAPE results for different AWSL intervals. Here instead of training for all data above a certain threshold put of the wind, training is done only on data between two wind speeds. The percentage variance in gust factor as a function of wind speed increases with decreasing wind speed. Measured wind speed is used for the cutoff and thus have data leakage. This results should thus be somewhat comparable to the last column in Table (4.1)

Interval	**MAPE**	
$[m/s]$	Without Elevation	With Elevation
$[5, 10[$	17.1%	16.4%
$[10, 15[$	14.5%	13.0%
$[15, 20[$	15.0%	12.0%
$[20, 25[$	15.6 %	13.1%
$[25, 30[$	18.4%	19.0%

without the feature labels, the station elevation is easily noticed as the value is constant for a station. This is not noteworthy. What is noteworthy is the range of impact from this single value. For simpler models, this would not happen. It is important to note that SHAP assumes feature independece[**Salih_2024**]. This might explain why the impact of the Richardson number is so low. Both the squared Brunt–Väisälä frequency and the Richardson number are derived features from reanalysis data. They carry with them some extra information over the other features in the dataset. This is because both are variables over elevation ranges. That is, as seen in Equations (2.1, 2.2), both are dependent on values at lower and upper elevations and try to describe the stability of that range. Shapley tries to assign contribution values for each feature for each observation. SHAP assumes that the features are independent, but this is not the case. It is clearly not the case for the derived variables, but how the contribution should be distributed between the features is not clear. Seemingly the SHAP python package is giving all the impact to the Brunt–Väisälä squared frequency and none to Richardson number. If the Brunt–Väisälä would be excluded from the data, the impact of the Richardson number would likely increase. Another point to note is that the features are ordered by their impact. This means that the station elevation is the most impactful for each plot, but the ordering of other variables changes. Looking at Figure (A.4), which shows the Shapley summary plot for all data, the wind speed is the second most important feature. This is reversed in Figure (A.5). This is not unexpected as Akrafjall station was specifically selected as the MAE for reanalysis wind speed was very high as can be seen in Table (A.5), where you will also find the stations whose summary plot is shown in Figures (A.6, A.7) and these also fall into the category of very high MAE for wind speed. What is interesting is that the reanalysis wind speed is also of low impact at stations like Háahlíð and Keflavíkurflugvöllur, as shown in Figures (A.8, A.9). These stations had the lowest of MAE for difference between measured wind speed and reanalysis wind speed. As the summary plot over

Table 4.4: The MAPE results of different stations for several stations of interest, both when training for the specific site and when the stations are a part of the general data. For every station the AWSL is set at 10 m/s. In training for a single station at a time, some site specific information can be gauged. This does not mean that the a better result can be reached for that site. Factors such as the number of datapoints at given location can significantly impact the result. This table uses the measured wind speed to determine the cut off for data points. This leads to some data leakage and an increased performance compared to using the reanalysis CARRA speed for cut off.

Station name	**MAPE**	
	Baseline	Model
Fáskrúðsfjörður	28.2%	21.8%
Ingólfsfjall	30.0%	19.6%
Kjalarnes	20.7%	13.5%
Sandskeið	13.0%	10.2%
Seyðisfjörður	32.1%	23.0%
Þjórsárdalur	12.2%	11.4%
Þrengsli	13.6%	11.3%

all stations (Figure (A.4)) shows that reanalysis wind speed is impactful, might lead to the conclusion that the reanalysis wind speed is not a good predictor at these locations or something else is skewing the data. A simpler way to look at feature importance is to create models that are trained on and use to predict different sets of parameters. The results of such a comparison can be seen in Table 4.5.

Table 4.5: Showing the results of different models. Models defined by different sets of parameters, with increasing complexity. This table shows the feature importance as the loss decreases as more variables are added. Adding all height levels decreases the loss but very little.

Model parameters	**MAPE**
Baseline constant	23.9%
ws_{15}	17.2%
$[ws_{15}, t_{15}, p_{15}, wd_{15}]$	16.7%
$[ws_{15}, t_{15}, p_{15}, wd_{15}, ASL, twd]$	13.8%
$[ws_{15}, t_{15}, p_{15}, wd_{15}, ASL, twd, N, Ri]$	13.7%
$[ws_{15}, t_{15}, p_{15}, wd_{15}, ASL, twd, N, Ri]$ + DEM	12.0%
$[ws_{15,250,500}, t_{15,250,500}, p_{15,250,500}, wd_{15,250,500}, ASL, twd, N, Ri]$	12.0%

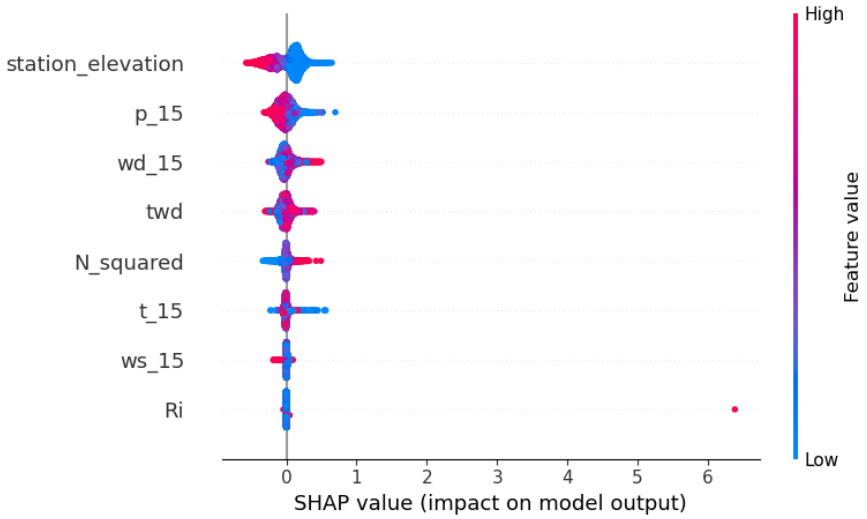

Figure 4.2: Feature importance of a neural network with model architecture as described in Table 3.1 and data as described in Table 2.3. We can see that generally multiple factors influence the prediction, with the station elevation being highly influential. There is seemingly one outlier for the Richardson number, which usually has very little influence. Elevation data is excluded when working with Shapley values, as the contribution of each elevation point is very low and there are very many of them. To see their influence on the model output see Table 4.1.

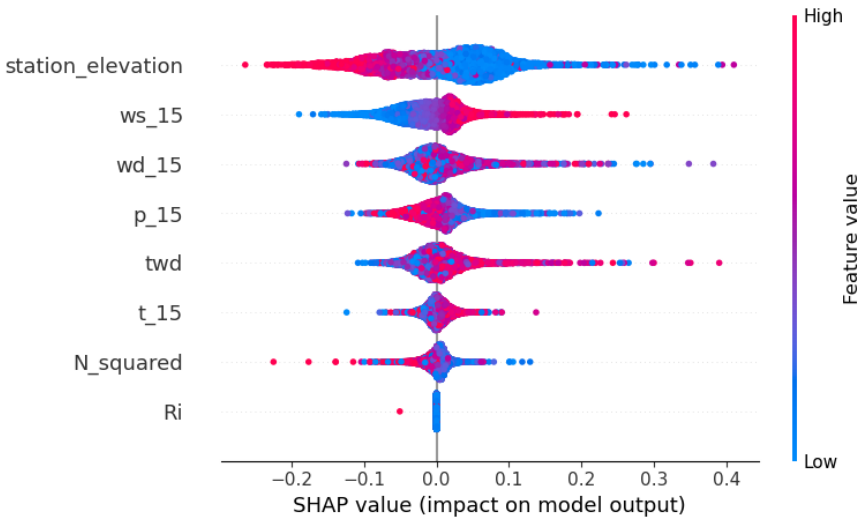

Figure 4.3: Feature importance of a neural network with model architecture as described in Table 3.1 and data as described in Table 2.3. Generally multiple factors influence the prediction, with the station elevation being highly influential. Elevation data is excluded when working with Shapley values, as the contribution of each elevation. In contrast to Figure (4.2), the distribution doesn't have as extreme outliers. This means that more details can be seen in the figure. The X-axis shows the influence of feature values on the model. The color gradient shows the value of each feature. As an example, there is a very red value for station elevation (top line, all the way to the left). This means that in this instance, the station elevation contributed around -0.25 to the final output and that the station had an elevation significantly above average.

5 Discussion

The goal of this tesis was to research whether it was possible to use reanalysis data to increase the predictability of wind gusts and see what influence including the landscape elevation would have. A quick statistical analysis will give a p-value of 0, that is these results are statistically significant. The improvement shown in Table (4.1) is not chance. There is some pattern to be learned in the interval that leads to a higher predictability of wind gusts over baseline. There is also a large variability in the predictability based on weather stations. A large part of that can be attributed to the differing amount of data points for each station. It would be interesting to know how much of the predictability difference can be explained away with more data for these stations that have few data points or if there is some inherent difference. If there is inherent less predictability at those stations, what could be contributing factors in that?

Appendix A: Feature importance on Shapley plots

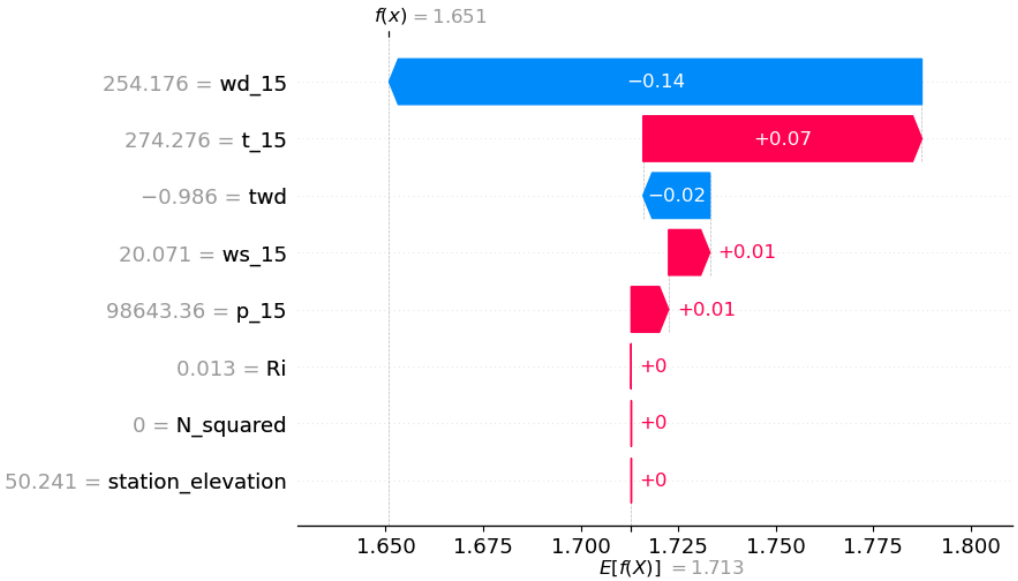

Figure A.1: Feature importance of a neural network with model architecture as described in Table 3.1 and data as described in Table 2.3. In this specific instance the wind direction (wd_{15}) has the highest negative influence and the temperature has the highest positive influence. Elevation data is excluded when working with Shapley values, as the contribution of each elevation point is very low and there are very many of them. To see their influence on the model output see Table 4.1.

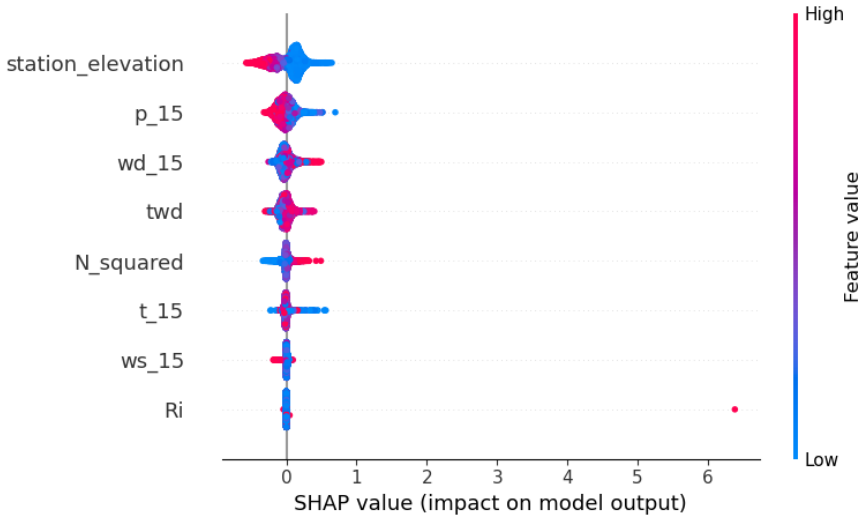

Figure A.2: Feature importance of a neural network with model architecture as described in Table 3.1 and data as described in Table 2.3. We can see that generally multiple factors influence the prediction, with the station elevation being highly influential. There is seemingly one outlier for the Richardson number, which usually has very little influence. Elevation data is excluded when working with Shapley values, as the contribution of each elevation point is very low and there are very many of them. To see their influence on the model output see Table 4.1.

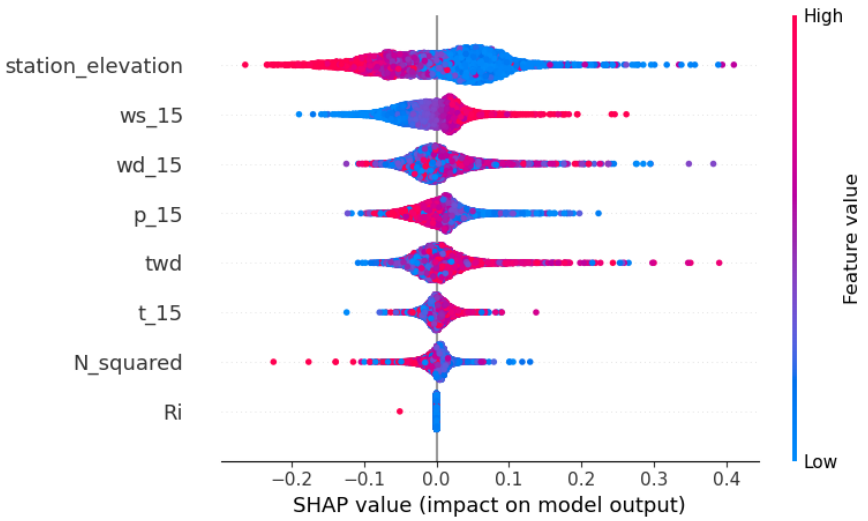

Figure A.3: Feature importance of a neural network with model architecture as described in Table 3.1 and data as described in Table 2.3. Generally multiple factors influence the prediction, with the station elevation being highly influential. Elevation data is excluded when working with Shapley values, as the contribution of each elevation. In contrast to Figure (4.2), the distribution doesn't have as extreme outliers. This means that more details can be seen in the figure. The X-axis shows the influence of feature values on the model. The color gradient shows the value of each feature. As an example, there is a very red value for station elevation (top line, all the way to the left). This means that in this instance, the station elevation contributed around -0.25 to the final output and that the station had an elevation significantly above average.

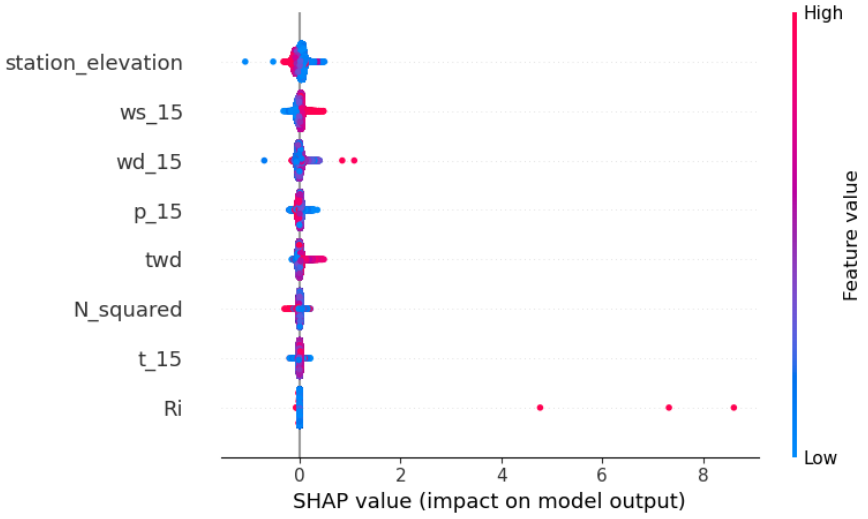

Figure A.4: Feature importance of a neural network with model architecture as described in Table 3.1 and data as described in Table 2.3. The distribution seems to be the same as before, discounting the outliers.

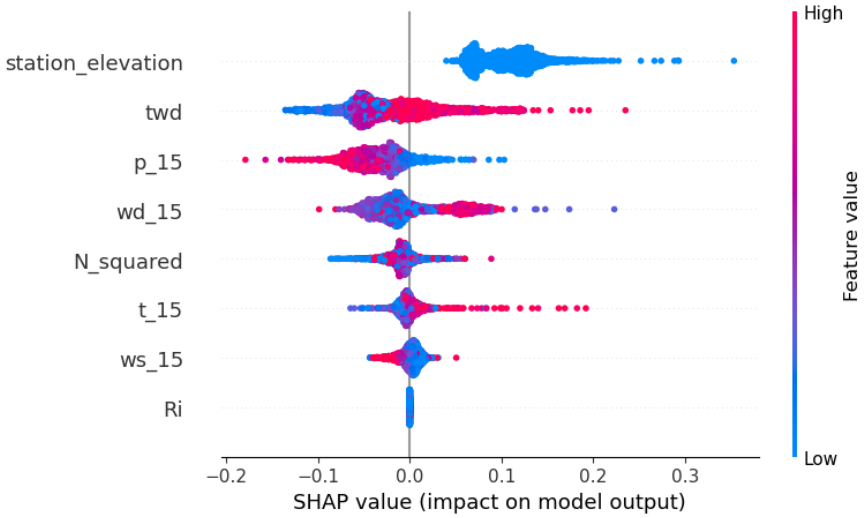

Figure A.5: Feature importance of a neural network with model architecture as described in Table 3.1 and data as described in Table 2.3. This plot only looks at datapoints from Akrafjall. This seems to show the same distribution as previous summary plots. Station elevation is influential and Richardson number has no impact.

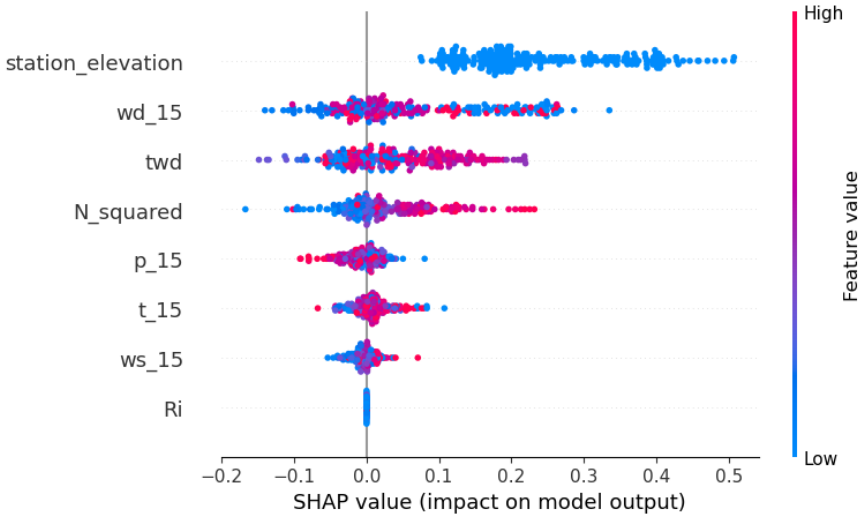

Figure A.6: Feature importance of a neural network with model architecture as described in Table 3.1 and data as described in Table 2.3. This plot only looks at datapoints from Almannaskarð. This seems to show the same distribution as previous summary plots. Station elevation is influential and Richardson number has no impact.

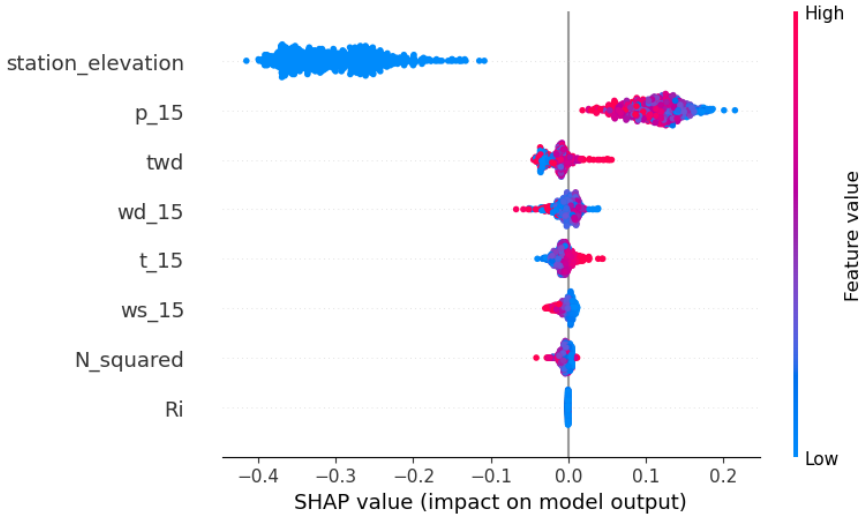

Figure A.7: Feature importance of a neural network with model architecture as described in Table 3.1 and data as described in Table 2.3. This plot only looks at datapoints from Ásgarðsfjall. This seems to show the same distribution as previous summary plots. Station elevation is influential and Richardson number has no impact.

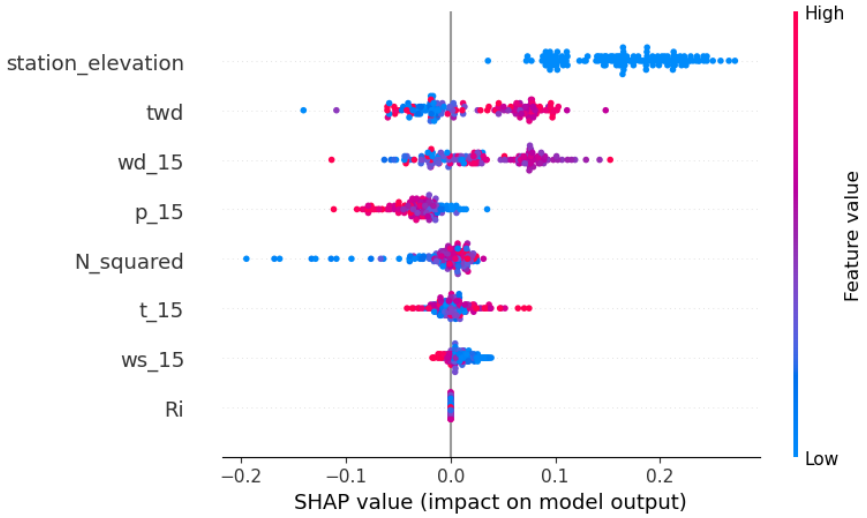

Figure A.8: Feature importance of a neural network with model architecture as described in Table 3.1 and data as described in Table 2.3. This plot only looks at datapoints from Háahlíð. This seems to show the same distribution as previous summary plots. Station elevation is influential and Richardson number has no impact.

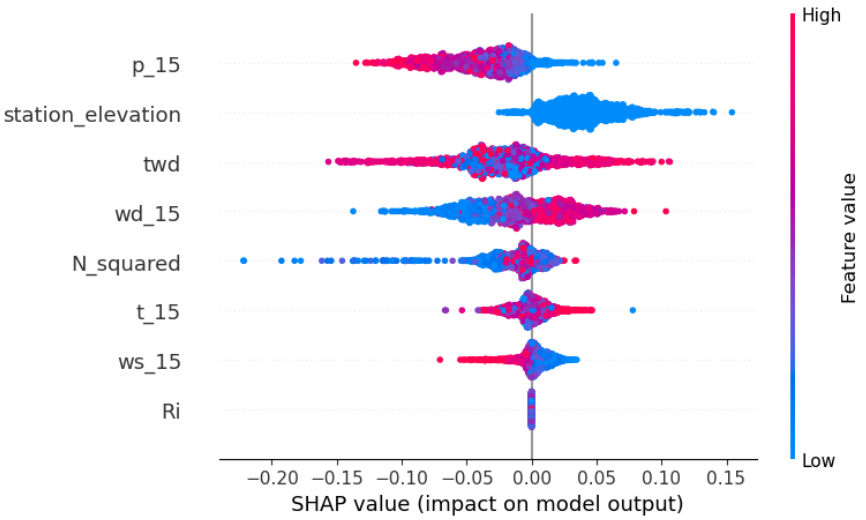

Figure A.9: Feature importance of a neural network with model architecture as described in Table 3.1 and data as described in Table 2.3. This plot only looks at datapoints from Keflavíkurflugvöllur. This seems to show the same distribution as previous summary plots. Station elevation is influential and Richardson number has no impact.