UNIVERSITY OF ICELAND

Prediction of Gust Factor in Complex Landscape with Neural Networks

Brynjar Geir Sigurðsson

June 2025

M.Sc. thesis
in Mechanical Engineering

FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

Prediction of Gust Factor in Complex Landscape with Neural Networks

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
Halldór Björnsson

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

Prediction of Gust Factor in Complex Landscape with Neural Networks

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) *Prediction of Gust Factor in Complex Landscape with Neural Networks*, M.Sc. thesis, Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland.

Reykjavik, Iceland, June 2025

Abstract

Whether it is wind farms, traffic accidents, high-voltage towers, or other things affected by wind, accurately predicting wind loads is crucial for reducing damage, improving operational safety, and boosting productivity. Wind gusts are multi-second spikes in speed, and wind gust factor is defined as the ratio of the highest three-second gust to the ten-minute mean wind speed. It is important to be able to predict both wind and gust factor accurately. Historically, numerical weather prediction models based on physical equations have been used to predict such variables. In recent years, artificial intelligence has advanced rapidly, and AI-based models have begun to rival conventional model performance. This project combines both approaches by using atmospheric model reanalysis output and digital elevation data as inputs to a deep neural network. Selected features from the reanalysis fields, terrain elevation, and derived variables are fed into several network architectures, whose performance is then compared against a simple linear regression baseline. The results are very promising: this hybrid methodology can substantially improve both instantaneous estimation and short-term forecasting of gust factors, particularly in areas of complex terrain.

Útdráttur

Hvort sem um er að ræða vindmyllugarða, umferðarslys, háspennumöstur, eða aðra hluti sem vindur getur haft áhrif á er spágeta um álag vegna vindsins mikilvæg, til að minnka eða koma í veg fyrir tjón, auka rekstraröryggi, eða bæta framleiðni. Vindhviður eru nokkurra sekúndna aukning á vindi og hviðustuðull mælir hæsta hlutfall milli þriggja sekúndna hviðu og 10 mínútna meðalvinds. Góðar hviðuspár eru mikilvægar. Sögulega hefur verið notast við hefðbundin spálíkön sem reiða sig á töluleg eðlisfræðileg líkön til að spá fyrir um veður, þar með talið hviðustuðul. Á síðustu árum hefur gervigreind þróast mikið og líkön verið þjálfuð til þess að spá fyrir um 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 meta hviðustuðla. Þetta er gert með því að nota endurgreiningargögn úr hefðbundnu veðurlíkani sem grunngögn í gervigreindarlíkan. Einnig eru notað hæðarlíkan sem lýsir landslagi. Djúpt tauganet er búið til með því að velja breytur úr endurgreiningargögnunum og hæðarlíkaninu og afleiddum breytum. Nokkur líkön af þessu tagi eru svo þjálfuð og niðurstöður bornar saman við einfalt grunnlíkan sem fengið er með línulega aðhvarfi. Niðurstöðurnar lofa mjög góðu um að þessi aðferðafræði geti bætt bæði samtímamat á hviðustuðli og spár um hann umtalsvert. Þetta á ekki síst við í flóknu landslagi.

Contents

Abbreviations xi

Acknowledgments 1

1 Introduction 2

1.1 Background . . . 2

1.1.1 Numerical weather prediction . . . 2

1.1.2 AI in weather forecasts . . . 3

1.1.3 Large weather models . . . 4

1.1.4 Conclusion . . . 4

1.2 Methodology and related work . . . 5

1.2.1 Neural networks . . . 5

1.2.2 Model evaluation . . . 7

1.2.3 Model explainability . . . 7

2 Data gathering and processing 9

2.1 Automatic Weather Station Data . . . 9

2.2 CARRA Data . . . 11

2.3 Elevation data . . . 12

2.4 Combining data sources . . . 12

2.5 Comparison of observed and reanalysis wind speed . . . 13

2.6 Layout of combined data . . . 15

2.7 Distributions of observed and reanalysis data . . . 17

2.8 Model architecture . . . 20

3 Results 22

3.1 Baseline models . . . 22

3.2 Modeling for $\mathrm{ws}_{\mathbf{15}}$ above 10 m/s . . . 22

3.3 Modeling for various $\mathrm{ws}_{\mathbf{15}}$ lower bounds . . . 23

3.4 Results for specific locations . . . 24

3.5 Results for wind speed intervals . . . 27

3.6 Feature importance with Shapley values . . . 27

3.6.1 Shapley summary plots for all stations . . . 27

3.6.2 Shapley summary plots for individual stations . . . 30

3.7 CARRA wind gust . . . 32

4 Discussion and conclusions 34

4.1 Main findings . . . 34

4.2 Limitations and future work . . . 34

4.3 Possible applications . . . 35

4.4 Conclusions . . . 35

List of Figures

1.1 General structure of a fully connected neural network 6

2.1 AWS at Stórhöfði in Vestmannaeyjar, south of Iceland. 10
2.2 Locations of automatic weather stations in Iceland 11
2.3 Flow chart illustrating the data-combination workflow 13
2.4 MAE distribution by station 15
2.5 Comparison of observed and reanalysis wind-speed distributions. CARRA values are spatially interpolated by linear weighting of surrounding grid points. 18
2.6 Distribution of weather station altitudes above sea level. One outlier near 2000 m was excluded due to limited and inconsistent data. . . . 19
2.7 Histogram of gust factors. By definition, gust factor ≥ 1; most values lie between 1.2 and 2. The gust factor declines as wind speed increases. 19

3.1 MAPE distribution by station 25
3.2 Shapley contributions for a single observation 28
3.3 Shapley summary plots 29
3.4 Summary feature importance of a neural network only looking at AWS at Keflavíkurflugvöllur. 30
3.5 Summary feature importance of a neural network only looking at AWS at Almannaskarð. 31
3.6 Summary feature importance of a neural network only looking at AWS at Akrafjall. 31

A.1 Summary feature importance of a neural network using entire dataset. 37
A.2 Summary feature importance of a neural network only looking at AWS at Ásgarðsfjall. 37
A.3 Summary feature importance of a neural network only looking at AWS at Háahlíð. 38

List of Tables

2.1 Measured vs. reanalysis wind-speed errors 14
2.2 Station MAE extremes 14
2.3 Example of combined data structure 16
2.4 Hyperparameter search with Hyperband 20

3.1 MAPE for baseline models 22
3.2 Model results for different sets of parameters. 23
3.3 Model results for different wind speed limits 24
3.4 Primary model MAPE by selected stations 25
3.5 Model result by station 26
3.6 Model result looking at closed wind speed intervals 27
3.7 Results adding CARRA gust factor 32
3.8 Regression MAPE with and without CARRA gust factor 33

Listings

2.1 Generation of elevation points in the upwind sector 15

Abbreviations

API	Application Programming Interface
ASL	Above Sea Level
AWS	Automatic Weather Stations
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, who helped me with the thesis work and sat with me to go over the code. Thanks also to Ólafur Pétur Pálsson who provided detailed notes and offered very helpful suggestions. I am also grateful to Guðrún Nína Petersen, who provided valuable insight into meteorological matters and gave constructive feedback 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**].

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

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 often 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

1.1.1 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.

1.1.2 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 development 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 persistence 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.

1.1.3 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 traditional 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**].

1.1.4 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 how ML and physics-based approaches can complement each other. 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 are used:

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

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

3. 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.

Structure of neural networks

The general structure of a neural network is shown in Figure 1.1. The input layer consists of n neurons, each representing an input parameter. The output layer has m neurons, each representing a predictand. Between these layers are one or more hidden layers, each with h_i neurons, where i is the layer index. Each neuron in a layer is connected to every neuron in the previous and next layers. The connections between neurons have weights that are adjusted during training to minimize prediction error.

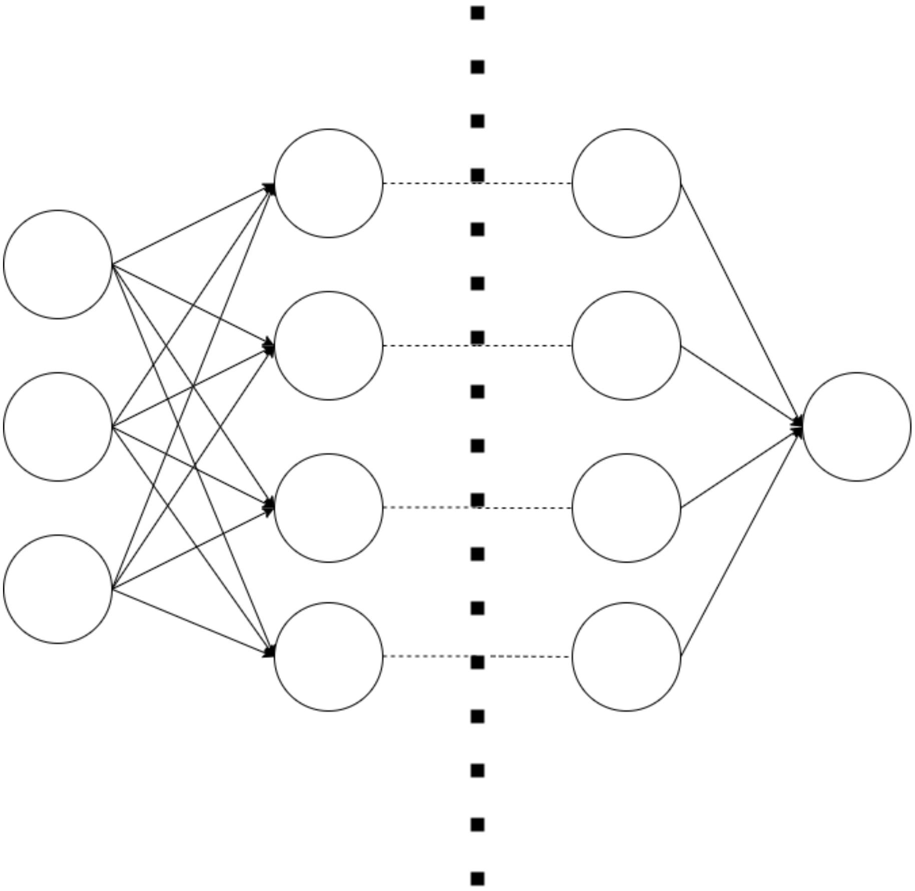


Figure 1.1: General structure of a fully connected neural network. The figure shows a network with three input variables and one output variable. It also shows one hidden layer (and others are implied). This hidden layer has four neurons that each connect to all input neurons and all connect to all neurons in the next hidden layer (not shown). This continues for all hidden layers until the output layer, when there is only one neuron.

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 functions 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**].

Apart from aiding in understanding final constructed models and the role of individual features, such explainability measures can also be used to select features to use in the modelling. For example, if the computed Shapley distribution of a feature is clustered at 0, then the corresponding feature has little or no influence, and can be safely removed from the model.

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

Automatic weather stations (AWS) have been set up all around the world, including Iceland. AWSs might include a variety of sensors, such as barometers, thermometers, and anemometer along with solar panels (or other power source) and some kind of data logger. IMO provided measurements from AWSs across Iceland. One of the stations used in this study can be seen in Figure 2.1. Not every station had consistently available data. Data from 327 AWSs were included looking at a period from 2004 and ended in 2023. Of these 327 stations provided by IMO, 212 were IMO stations, 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 2.2.

Data from these AWSs were 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.

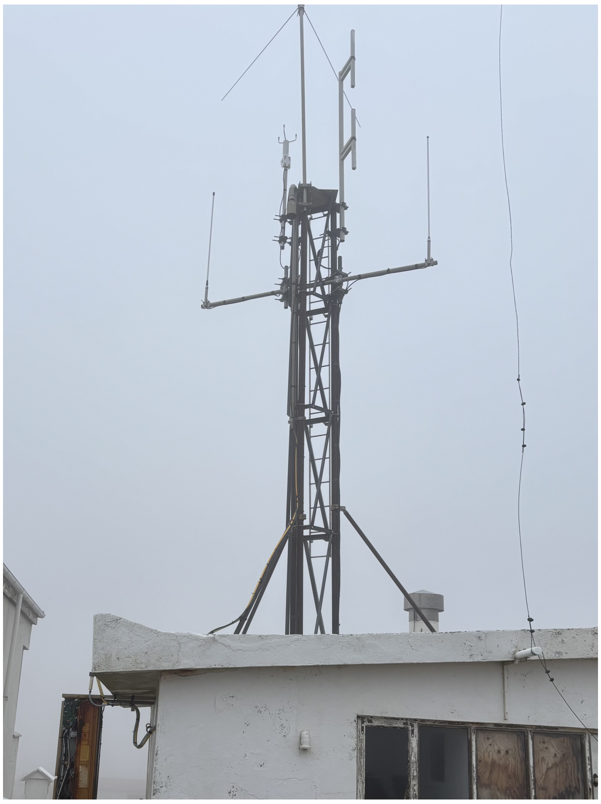


Figure 2.1: AWS at Stórhöfði in Vestmannaeyjar, south of Iceland. The anemometer can be seen as three pins in the upper-central area of the figure just slightly to the left.

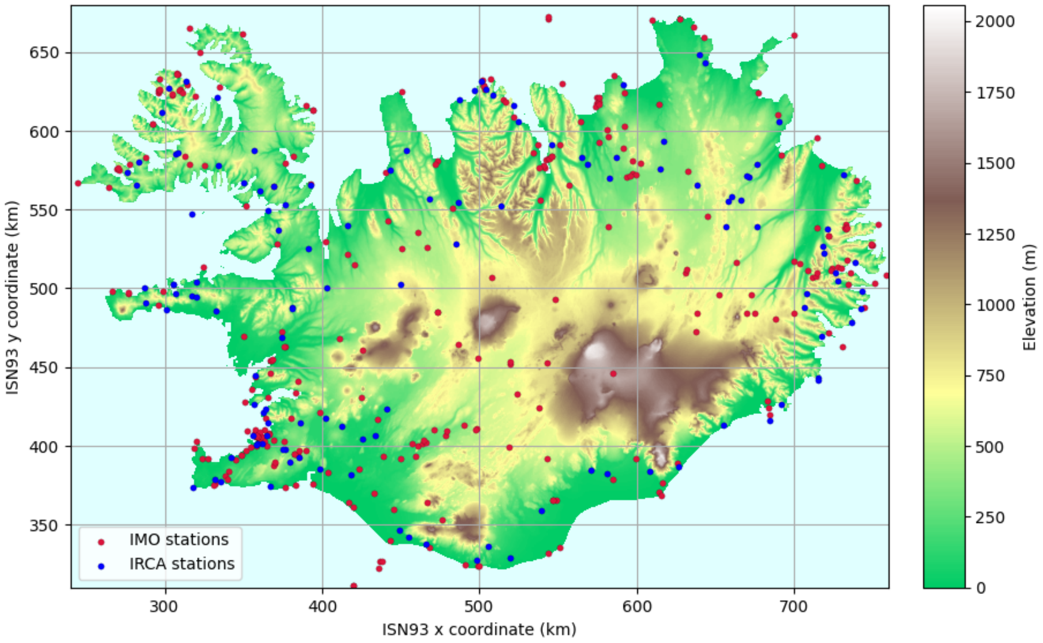


Figure 2.2: Locations of all 412 stations that were looked at in this study. Most of these were from IMO but over a hundred were from IRCA. IMO anemometers are placed at 10 meters above ground, while IRCA ones are placed at 6–7 meters above ground.

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 a digital elevation model (DEM) for Iceland 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 2.3, was as follows: for each AWS observation, 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.

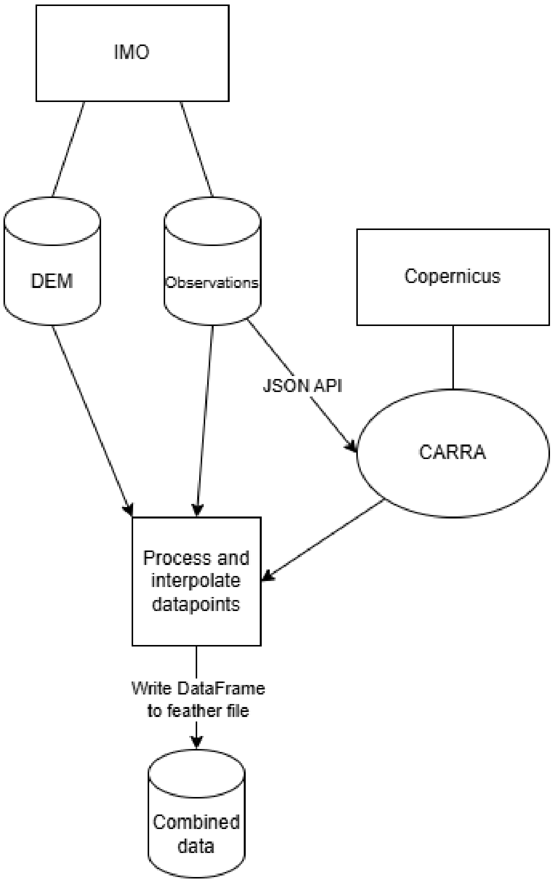

IMO
DEM
Observations
Copernicus
JSON API
CARRA
Process and interpolate datapoints
Write DataFrame to feather file
Combined data

Figure 2.3: Flow chart illustrating the data-combination workflow

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

Differences between reanalysis and measured wind speeds can be substantial. Absolute error increases with wind speed, while percentage error decreases. An overview of these errors by wind-speed range is shown in Table 2.1.

Next, the distribution of MAE by station is examined, considering station location and number of observations. Figure 2.4 shows this distribution, and Table 2.2 lists

Table 2.1: Comparison of measured and reanalysis wind speeds using mean absolute error (MAE) and mean absolute percentage error (MAPE). Values for observed wind speeds below 1 m/s are excluded to avoid inflated MAPE values. Measured speeds are at 10 m above ground (IMO) or 6–7 m (IRCA); reanalysis speeds are at 15 m.

f	n	MAE	MAPE
$[1; 5[$	5,260,814	2.0	83.9%
$[5; 10[$	4,150,923	2.2	31.3%
$[10; 15[$	1,480,487	2.5	21.0%
$[15; 20[$	388,905	3.0	17.8%
$[20; 25[$	84,099	4.0	18.4%
$[25; \infty[$	20,288	6.6	23.0%
$[1; \infty[$	11,385,516	2.2	53.7%

the five stations with the lowest MAE and the five with the highest MAE.

Table 2.2: Mean absolute error (MAE) between reanalysis and observed wind speeds at the five stations with lowest and highest errors. The observation counts have been rounded.

Station	n	MAE
Reykjavík Háahlíð	6,800	1.17
Keflavíkurflugvöllur	52,000	1.18
Reykjavík Víðidalur	14,000	1.23
Rif á Melrakkasléttu	13,000	1.29
Reykjavíkurflugvöllur	56,000	1.29
Almannaskarð	4,000	4.30
Kerlingarfjöll	15,000	4.36
Fáskrúðsfjarðargöng	7,900	4.40
Seljalandsdalur	1,600	4.51
Botn í Súgandafirði	32,000	4.95

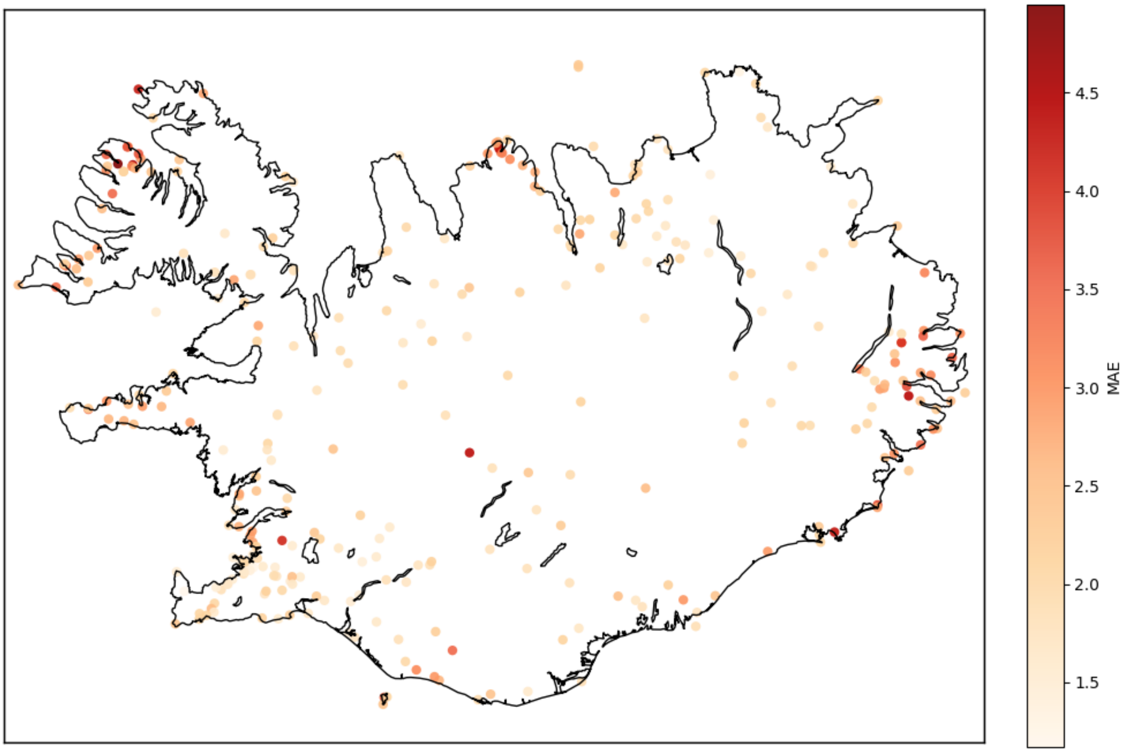


Figure 2.4: Distribution of mean absolute error (MAE) by station for all observed wind speeds.

2.6 Layout of combined data

Once data from all three sources have been retrieved and processed—including interpolation—they must be combined and formatted for modeling (training, validation, and testing). The starting point is a DataFrame containing AWS observations: average wind speed, wind gust, wind direction, station number, and coordinates. CARRA data are provided at selected height levels, each as a separate row in the CARRA DataFrame. Thus a single observation will span multiple rows, one per level. These rows are merged by time and station location, enabling AWS and CARRA data to be joined on station and time fields.

The DEM elevation data are handled by defining an upwind sector: a range of angles relative to the wind direction *d* and radial distances from each station. Points within this sector are generated as shown in Code Listing 2.1.

Listing 2.1: Generation of elevation points in the upwind sector

```
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 resulting DataFrame includes AWS measurements (our target), CARRA weather variables, and elevation points. An example of the combined data structure is shown in Table 2.3.

Table 2.3: Example of the combined data structure of features used for modeling. Data include derived variables Richardson number (Ri) and squared Brunt–Väisälä frequency (N^2), station altitude (meters above sea level), transformed wind direction (twd), wind speed (ws_{15}), wind direction (wd_{15}), temperature (t_{15}), pressure (p_{15}) (CARRA values at 15 m height), and elevation points (from DEM) in a sector pointing upwind.

Ri	N^2	station altitude	twd	ws_{15}	wd_{15}	t_{15}	p_{15}	$\mathrm{elevation}_0$	$\mathrm{elevation}_1$	...
-1.18	26700	100	1.5	10	5	0	100	2	4	...
...										

The transformed wind direction is a proxy showing whether the wind is coming from land or sea. It is computed as the angle between the wind direction and a vector from the center of Iceland to the station.

The Richardson number and the Brunt–Väisälä frequency describe atmospheric stability, see Equations (2.1) and (2.2) [**richardson_number_skybrary**, **brunt_vaisala_freq_eu** **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.

$$Ri = \frac{g \cdot d\theta \cdot dz}{\theta_{\text{ave}} \cdot dU^2} \, [] \tag{2.1}$$

$$N = \sqrt{\frac{g \cdot d\theta}{\theta_{\text{ave}} \cdot dz}} \, [\text{Hz}] \tag{2.2}$$

Here, g is gravitational acceleration; $d\theta$ the potential temperature difference between levels; dz the height difference; θ_{ave} the average potential temperature; and dU the wind-speed difference. Lower Ri indicates greater turbulence, with typical values between 0.1 and 10, and values below 1 signifying significant turbulence [**richardson_number_skybrary**]. Negative N^2 denotes instability, as an air parcel will accelerate away from its original position [**brunt_vaisala_freq_eumtrain**]. The potential temperature is computed as:

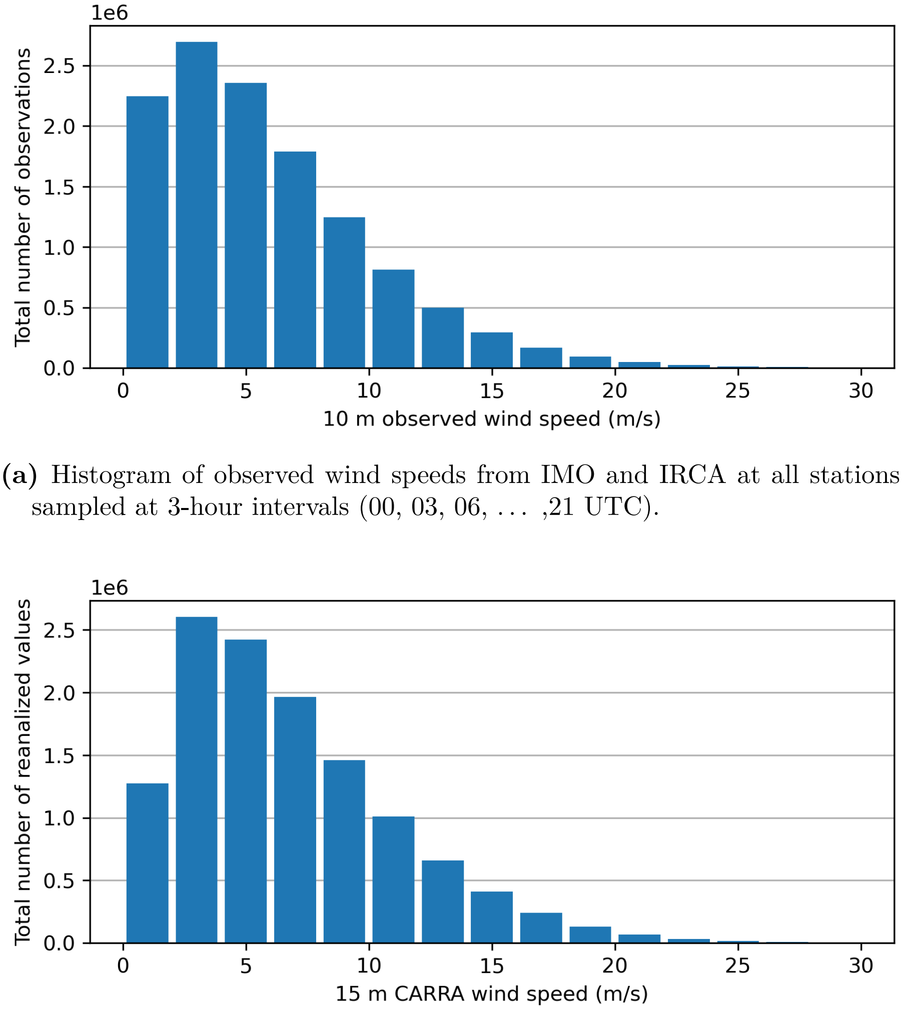
$$\theta = T \left(\frac{P0}{P} \right)^{R/c_p} \tag{2.3}$$

where P is the pressure and $R = 287$ and $c_p = 1004$ are the specifc gas constant and the specific heat capacity of air.

Since Ri and N^2 are derived from reanalysis variables, they add limited new information compared to the raw data, but they can reduce input dimensionality and improve model explainability via Shapley-value analysis [**shapley_information**]. Shapley values assess feature importance by evaluating all possible feature subsets—an operation of computational complexity 2^n. In practice, approximations are used but can still be costly for large models. Because dU appears squared in the denominator of Ri, if the wind-speed difference between levels is very small, Ri can become arbitrarily large, potentially distorting predictions.

2.7 Distributions of observed and reanalysis data

CARRA reanalysis data may exhibit bias or systematic differences when compared to measurements. Figure 2.5 shows the distributions of observed and reanalysis wind speeds. Although the shapes are similar, reanalysis values tend to be higher.



(a) Histogram of observed wind speeds from IMO and IRCA at all stations, sampled at 3-hour intervals (00, 03, 06, . . . ,21 UTC).

(b) Histogram of interpolated CARRA wind speeds at station locations on a 2.5 km grid and 3-hour intervals.

Figure 2.5: Comparison of observed and reanalysis wind-speed distributions. CARRA values are spatially interpolated by linear weighting of surrounding grid points.

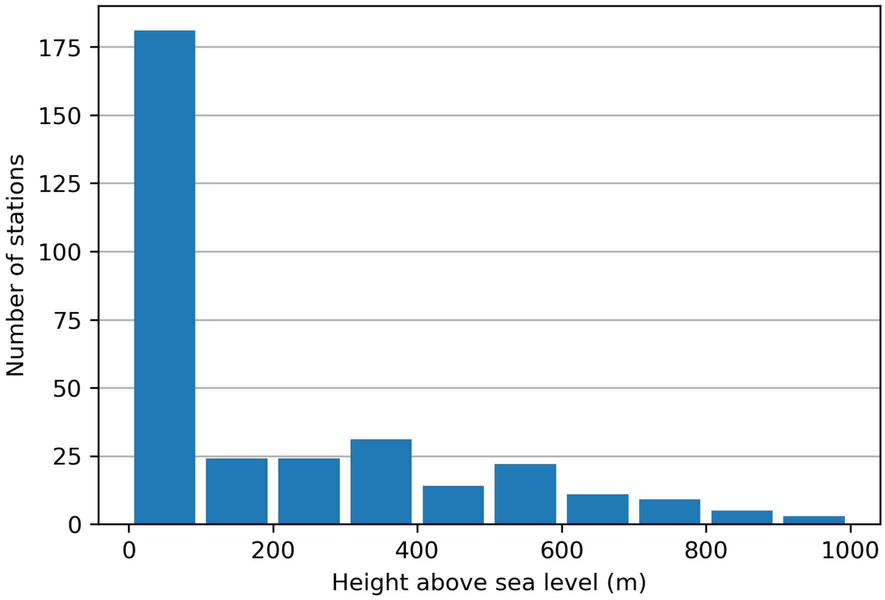


Figure 2.6: Distribution of weather station altitudes above sea level. One outlier near 2000 m was excluded due to limited and inconsistent data.

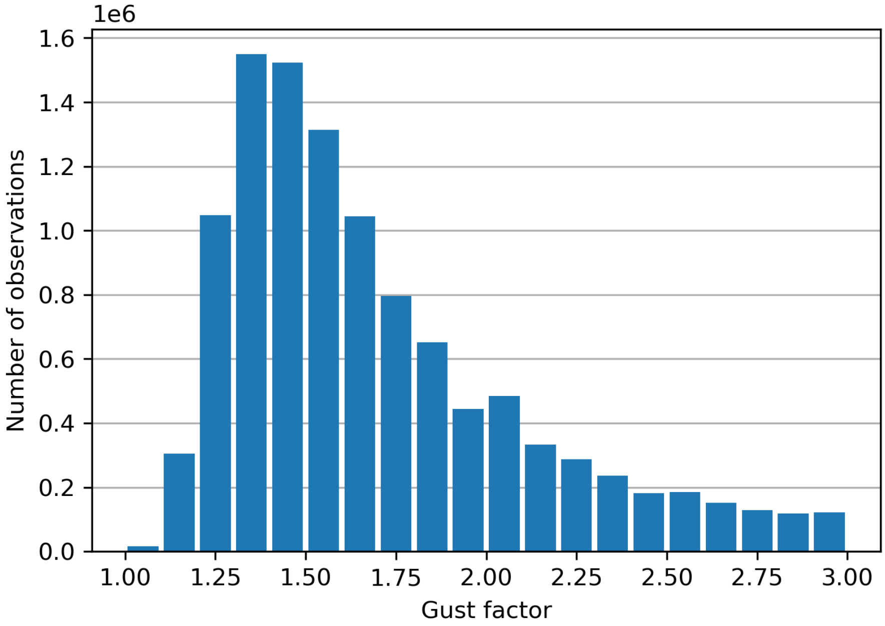


Figure 2.7: Histogram of gust factors. By definition, gust factor ≥ 1; most values lie between 1.2 and 2. The gust factor declines as wind speed increases.

2.8 Model architecture

The network consists of n fully connected layers, with n around 10. Each hidden layer has the same number of units and is followed by batch normalization. A dropout layer with rate 50% is applied after the final hidden layer. A dense output layer with one unit produces the predicted gust factor. A grid search was used to find the hyperparameters that minimize the loss. Hyperparameters are settings defined before training begins [**hyperparameters_definition**]. They include the number of units per layer, the number of layers, the number of training epochs, batch size, optimizer choice, and regularization penalty. The ranges tested are listed in Table 2.4.

Table 2.4: Hyperparameter search was performed using the Hyperband algorithm, which begins by sampling configurations at random and then focuses on the most promising ones. The table shows the ranges tested and the best-performing combination. Unlike random search, Hyperband does not impose a fixed limit on the number of trials.

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 grid search. 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.

As noted, Hyperband does not impose a fixed limit on the total number of training epochs. When using the Hyperband tuner, the Hyperband iterations parameter controls how many times Hyperband is run and defaults to 1. On each run, the total number of epochs is distributed across the hyperparameter trials. The total number of epochs is approximately

$$n_{\text{epochs}} = \text{max\_epochs} \times \big(\log(\text{max\_epochs})\big)^2 \approx 10^4$$

for max_epochs = 1000. Although this method is more efficient than grid search, if each epoch takes about 10 seconds, a complete Hyperband search can still require about 28 hours on shared resources, making repeated searches impractical. It remains unclear whether the current hyperparameter ranges are optimal.

3 Results

3.1 Baseline models

The simplest model for the gust factor is to guess the average gust factor for all available observations, i.e. use no variables from the training data. A slightly better model is obtained by using the wind speed as a training feature, either through regression or NN training. The MAPEs for the constructed base models are shown in Table 3.1. The ws_{15} model was obtained with NN training but regression produces an almost identical result.

Table 3.1: MAPE for several lower limits on the reanalysis 15 m wind speed (ws_{15}). Note that the ws_{15} model was only evaluated for the 10 m/s lower bound.

Wind speed	MAPE	
limit $[m/s]$	mean model	ws_{15} model
≥ 0	39.2%	
≥ 5	28.1%	
≥ 10	23.9%	17.2%
≥ 15	23.2%	
≥ 20	24.7%	
≥ 25	27.7%	

As the average wind speed increases, then the variability in gust as a percentage decreases [**mean_gust_HA_HO**]. Therefore lower MAPE would be expected if the training data is restricted to higher wind speeds. This fact explains why several models were created for different lower bounds.

3.2 Modeling for ws_{15} above 10 m/s

As noted in Chapter 1, it is more important to predict accurate gust factor when it is windy than when the weather is calm. Therefore, at first, the data is restricted to $\mathrm{ws}_{15} \geq 10$ m/s. The first constructed model incorporates the features temperature, pressure and wind direction at 15 m height. This gives a moderately reduced MAPE of 16.7% compared with 17.2% for the baseline ws_{15} model. When the next pair of features, station altitude and transformed wind direction, is included, the MAPE is reduced considerably, down to 13.8%. The atmospheric stability indicators, Ri and N^2, add little to the model performance, contrary to what might have been expected. When the atmosphere is unstable, wind gusts from higher up might be expected. However, another considerable performance increase is observed when digital elevation model (DEM) information is included, down to a MAPE of 12.0%.

This represents over 30% reduction in the MAPE from the ws$_{15}$ baseline model. Adding all height levels decreases the loss but very little, as seen in the last line of the table.

The second last model in the Table 3.2 will be referred to as the *primary model* and the corresponding parameters as the *primary parameter set*. The model immediately before that is the *secondary model*. In some cases the terminology "primary parameter set with and without DEM" will be used to refer to the parameter sets of these models.

Table 3.2: Models defined by a selection of parameter sets, with increasing complexity. The table shows decreasing MAPE as more variables are added.

Model variables	MAPE
[ ] Baseline	23.9%
$[ws_{15}]$ Baseline	17.2%
$[ws_{15}, t_{15}, p_{15}, wd_{15}]$	16.7%
$[ws_{15}, t_{15}, p_{15}, wd_{15}, \text{altitude}, twd]$	13.8%
$[ws_{15}, t_{15}, p_{15}, wd_{15}, \text{altitude}, twd, N^2, Ri]$*	13.7%
$[ws_{15}, t_{15}, p_{15}, wd_{15}, \text{altitude}, twd, N^2, Ri]$ + DEM**	12.0%
$[ws_{15,250,500}, t_{15,250,500}, p_{15,250,500}, wd_{15,250,500}, \text{altitude}, twd, N^2, Ri]$	12.0%

*Secondary model **Primary model

3.3 Modeling for various ws$_{15}$ lower bounds

To assess how model performance depends on ws$_{15}$ lower bound, several models were trained using the primary variable set, with and without DEM. The results can be seen in Table 3.3. For this investigation comparison was only made with the constant baseline model. In all cases adding the DEM information is seen to increase the model performance. The reason for the difference between the 10 m/s lower bound results seen in Tables 3.2 and 3.3 is that they were obtained with different training setups.

Table 3.3: MAPE for each ws_{15} lower bound without and with landscape elevation data (DEM) in a 30° sector into the direction of the reanalysis wind. Adding DEM data reduces the error noticably. Note that the MAPE is worse for both the lower lower bounds and the higher ones, but better for the intermediate ones.

Wind Speed		MAPE	
limit $[m/s]$	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%

3.4 Results for specific locations

In this section, the distribution of MAPE by station will be considered. For each station the primary model MAPE was computed using just the $\mathrm{ws}_{15} \geq 10$ data for that station was calculated and the results are shown in Figure 3.1. The stations with low performance are all near mountains and often in fjords. The worst cases are in Vestfirðir and Austfirðir, and several bad stations are e.g. in Tröllaskagi and Snæfellsnes, and near Öræfajökull, Esja, Eyjafjallajökull. Table 3.4 shows the primary model MAPE together with the constant baseline reference for a few selected stations.

Inland stations seem to have lower error. The worst performing station is Seljalandsdalur in Vestfirðir while the best performing station is Garðskagaviti at the tip of the Reykjanes peninsula. Evidently rugged landscape makes it difficult to estimate the gust factor accurately.

After examining the MAPE at each station individually, it is logical to try accounting for the station (or its location) in the model's design. There are two ways to do this: To include the station location in the training data, and to train on subsets of the data, e.g. by station or groups of stations. Here the latter approach is considered. Results of an investigation for a few stations can be seen in Table 3.5. For many of the stations, the results are worse than expected, sometimes much worse, calling for further investigation.

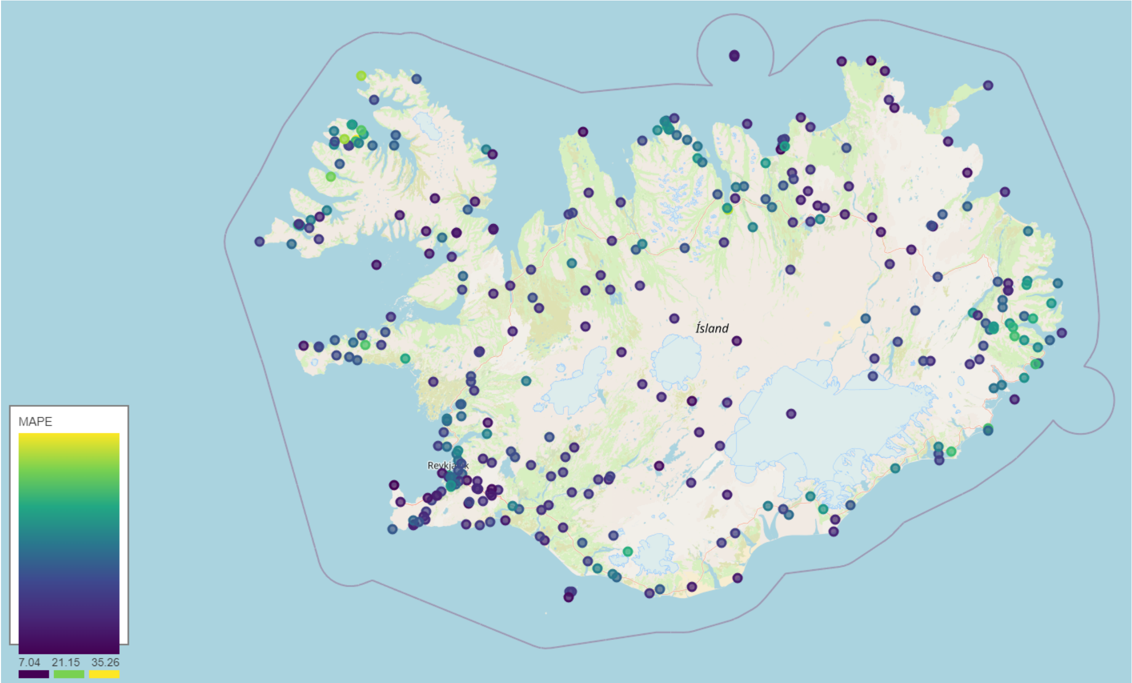


Figure 3.1: Mean absolute percentage error (MAPE) at each station, shown as colored circles (dark blue = low error; yellow = high error). The lowest station error is about 7% at Garðskagaviti and the highest about 35% at Seljalandsdalur. The model uses a 10 m/s lower bound based on ws_{15}.

Table 3.4: Constant baseline and primary model MAPE for a few selected stations. High MAPE values is seen in fjords near tall mountains in Austfirðir (Fáskrúðsfjörður and Seyðisfjörður), intermediate values at the foothills of mountains (Ingólfsfjall and Kjalarnes), and low values in flatter places (Sandskeið, Þrengsli).

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%

Table 3.5: MAPE for a few selected stations, for a model trained with the whole data set (general training), and for a model trained only with the data of the station itself (site training), with a lower bound of 10 m/s. This investigation was done in an earlier phase of the study, when f instead of ws_{15} was being used as the lower bound cut off, and that leads to some data leakage and a somewhat lower MAPE.

Station name	n	MAPE	
		General training	Site training
Akrafjall	43,000	18.6%	93.7%
Almannaskarð	4,000	12.2%	86.7%
Ásgarðsfjall	15,000	9.1%	9.4%
Jökulheimar	17,000	7.7%	7.7%
Sandbúðir	19,000	6.8%	6.4%
Stórholt	35,000	7.1%	29.2%
Þúfuver	20,000	6.4%	6.8%

3.5 Results for wind speed intervals

In addition to training for individual stations one can also try to train for individual wind speed intervals. Table 3.6 shows the results of training with the primary parameter set, with and without DEM. The table does not show noticeable improvement over the values in the last column in Table 3.3, in fact the two tables show quite similar results.

Table 3.6: The MAPE results for different wind speed limit intervals. Here instead of training for all data above a certain threshold of wind speed, 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. These results should thus be somewhat comparable to the last column in Table 3.3

Interval	MAPE	
$[m/s]$	Without DEM	With DEM
$[5, 10[$	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%

3.6 Feature importance with Shapley values

In this section Shapley values (see section 1.2 and 2.6) for the features of the secondary parameter set will be considered. Each feature value in each data row will contribute some share in the final prediction of an NN model, that can be computed by shifting the respective value and observe the change in the model output. This procedure is illustrated in Figure 3.2.

3.6.1 Shapley summary plots for all stations

Figure 3.3 provides a Shapley summary plot showing the contribution of each feature in the set. Computation of these values is expensive so subset of data is used (∼1000 randomly selected data rows). There is an obvious outlier in the figure, making it difficult to assess the distribution. A second Shapley summary was computed with more data rows (∼10000) and outliers excluded, see the lower panel of the figure. Note that it is not feasible to include the DEM data in this analysis, as they contain far too many variables.

Looking specifically at the lower panel in Figure 3.3, the station elevation (altitude) is the most influential and the Richardson number's influence is very low, with the

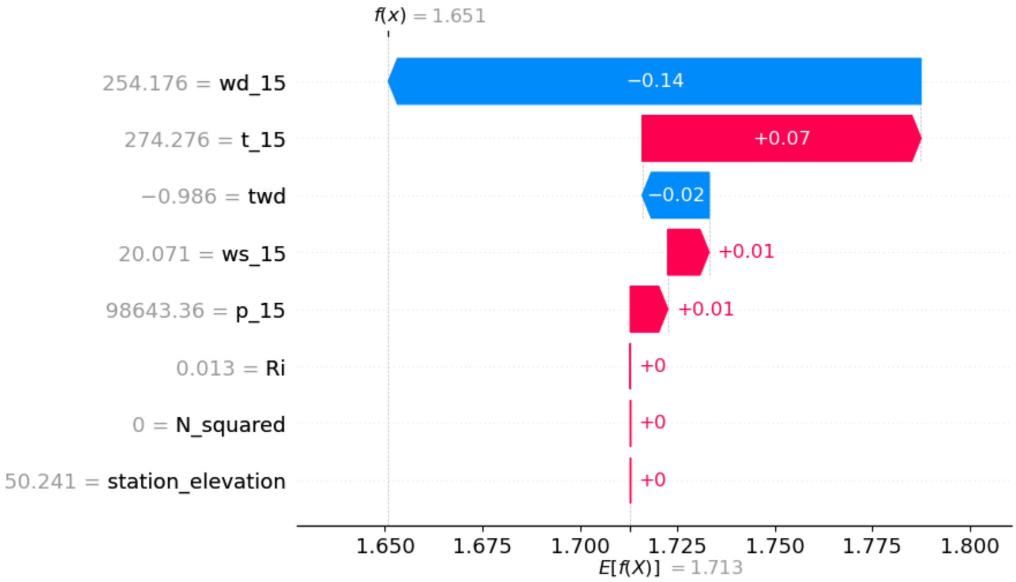


Figure 3.2: Shapley contributions for a single data row. The figure shows an example of the individual values contributing to a Shapley summary plot. Each point in a summary plot corresponds to one such instance. Several values with the same SHAP value will result in a thicker bulge in the summary. In this specific instance the wind direction (wd_{15}) has the highest negative contribution and the temperature has the highest positive contribution. The last three features contribute nothing.

other features falling in between. For most features the Shapley values are bunch up around 0, while the station elevation, and to some extent ws_{15} have a wider distribution. Station altitude/elevation seems to have a wide flat part between –0.1 and 0.1. Overall, the values seem to have a longer right tail. This is expected as gust factors are positive (~1.2 to 2). Notice also that Ri has virtually no contribution and could safely be removed from the model.

The coloring of the points in the displayed distributions show the corresponding feature values. For example, in the lower panel of Figure **??**, high station altitude/elevation corresponds to negative contribution and vice versa, and same for p_{15}. For ws_{15} and t_{15} the relationship is opposite, with high feature values corresponding to positive contribution. When the coloring is more mixed the contribution of the feature is more mixed. For example, high twd might have a positve contribution at some locations but negative elsewhere.

Figure A.1 in Appendix A shows a Shapley summary plot for all the data rows. However this plot is detrimentally affected outliers.

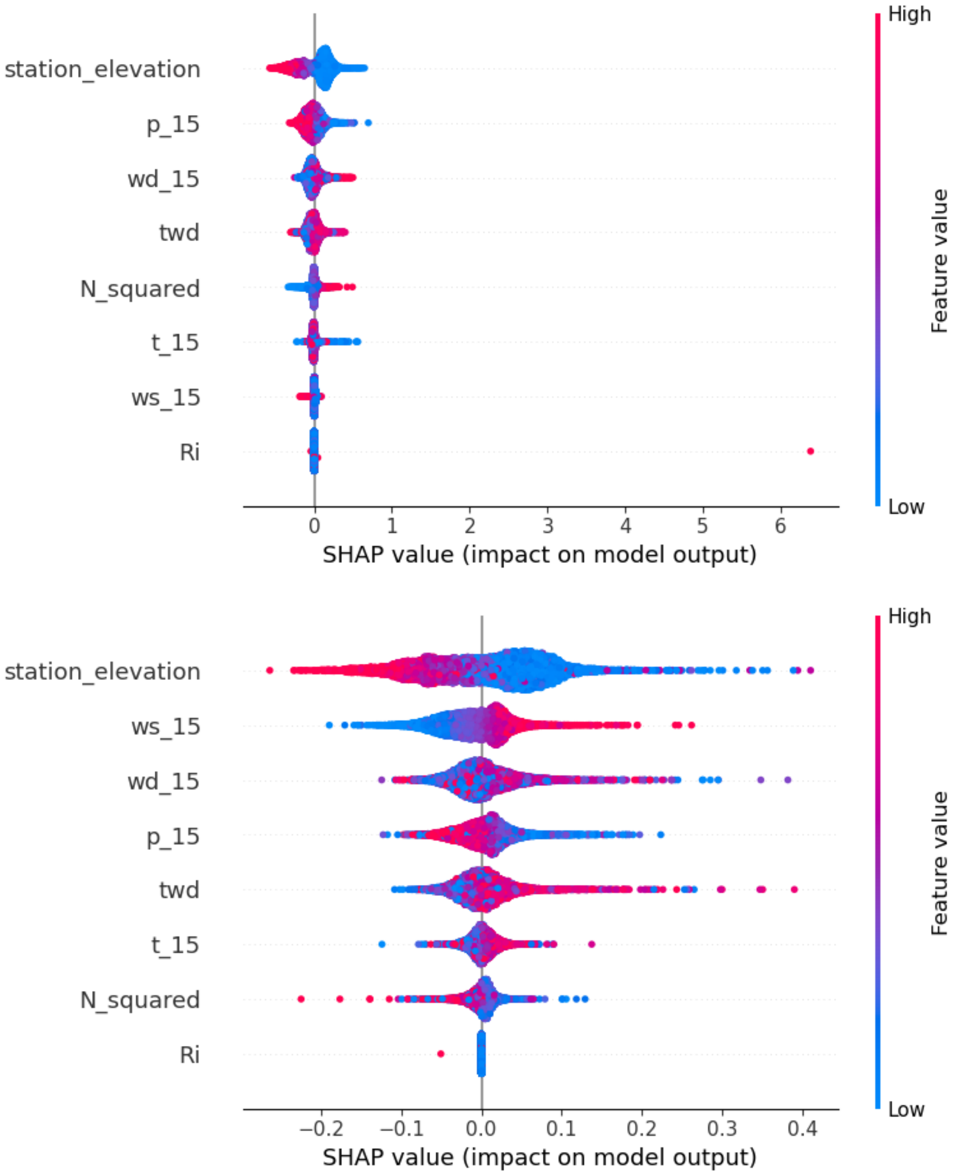


Figure 3.3: Shapley summary plots showing feature importance for the secondary model described in Table 3.2, with randomly chosen subsets of the data. SHAP values with high absolute value indicate high contribution of the corresponding feature. Coloring shows the respective feature value, blue for low and red for high. For example high station altitude/elevation corresponds to negative contribution and vice versa, and for ws_{15} the relationship is opposite, with high feature values corresponding to positive contribution. It can be seen that generally multiple factors influence the prediction, with the station elevation being highly influential. There is one outlier in the upper panel, for the Richardson number, which otherwise has very little influence. The upper panel is based on ∼1000 data rows. The lower panel is based on more data rows (∼10000), and furthermore, outliers have been removed there to make the plot more legible.

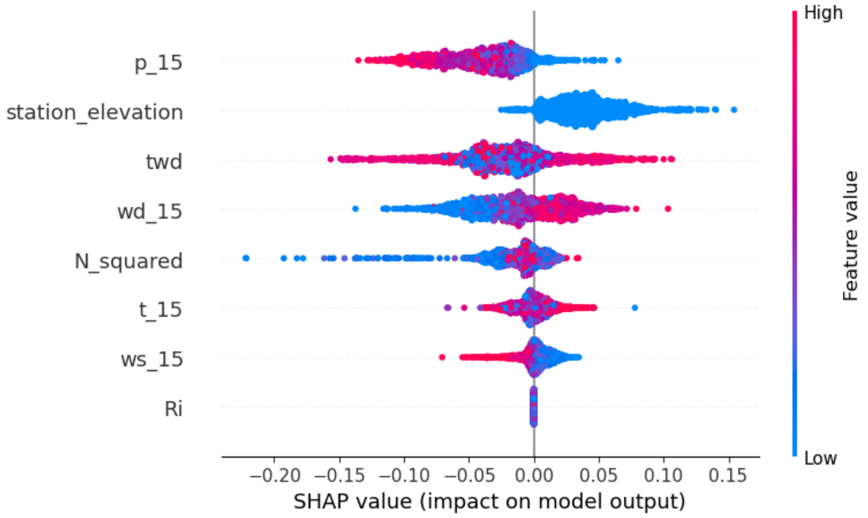


Figure 3.4: Feature importance of a neural network with model architecture as described in Table 2.4 and data as described in Table 2.3. This plot only looks at datapoints from Keflavíkurflugvöllur. This seems to show the same distribution as previous summary plots. Station elevation is influential and Richardson number has no impact.

3.6.2 Shapley summary plots for individual stations

It can also be informative to look at summary plots for individual stations. In this subsection two such examples are considered, for Akrafjall and Keflavíkurflugvöllur. It is noteworthy to see that the constant value of a stations altitude produces quite different contributions to the predictions, presumable because of diffent interactions with other features. For (very) simple models, such as linear regression without cross terms, this would not happen.

Further examples of Shapley summary plots for individual stations may be found in Appendix A.

It is important to note that Shapley summary plot analysis assumes feature independence, because only one feature at a time is changed [**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 measures of atmospheric stability and partly depend on the same variables, i.e. pressure and temperature at different height levels, as seen in Equations (2.1) and (2.2). Shapley tries to assign contribution values for each feature for each observation, but how the contribution should be distributed between the two features is not clear. 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. The station

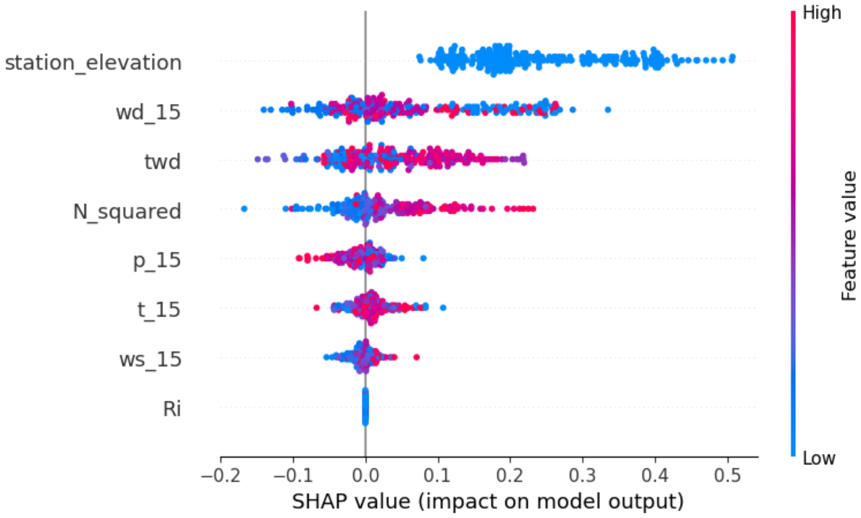


Figure 3.5: Feature importance of a neural network with model architecture as described in Table 2.4 and data as described in Table 2.3. This plot only looks at datapoints from Almannaskarð. This seems to show the same distribution as previous summary plots. Station elevation is influential and Richardson number has no impact.

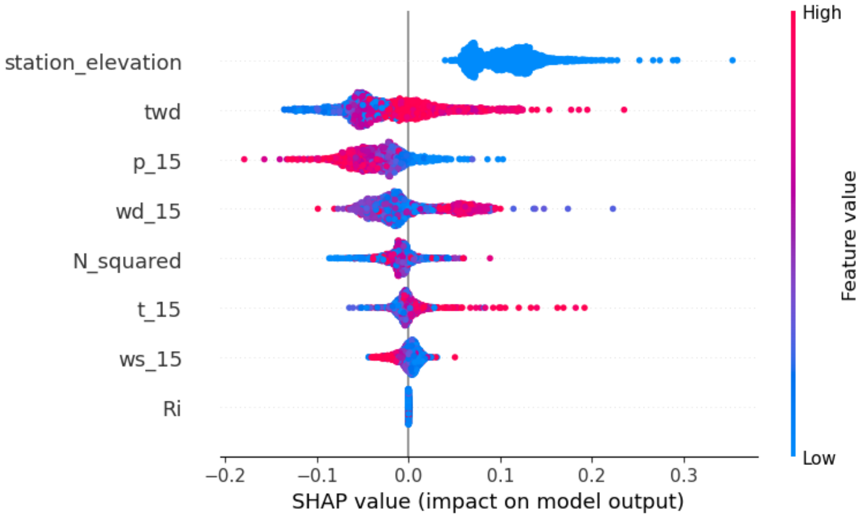


Figure 3.6: Feature importance of a neural network with model architecture as described in Table 2.4 and data as described in Table 2.3. This plot only looks at datapoints from Akrafjall. This seems to show the same distribution as previous summary plots. Station elevation is influential and Richardson number has no impact.

altitude is the most impactful for all the plots except Keflavíkurflugvöllur (Figure 3.4), but the ordering of other variables changes. Looking at the lower panel of Figure 3.3, the wind speed is the second most important feature. This is reversed in Figure 3.5. This is not unexpected as the Almannaskarð station was specifically selected as a station with high MAE between observed and reanalysis wind speed (Table 2.2. Another station with high MAE is Akrafjall, with a Shapley summary plot in Figure 3.6).

What is interesting is that the reanalysis wind speed is also of low impact at stations like Keflavíkurflugvöllur, as shown in Figure 3.4 which had the second lowest MAE difference between measured wind speed and reanalysis wind speed (see Table 2.2). Another such station, in fact the one with the lowest MAE, is Háahlíð in Reykjavík. The Shapley summary plot for this station is shown in Appendix A (A.3).

3.7 CARRA wind gust

The CARRA dataset provides no gust values in its analysis component but does include 10 m wind gust in the forecast. With a shortest lead time of one hour and given that a model of the gust factor is to be constructed, it stands to reason to incorporate the one-hour forecast CARRA gust factor, computed as:

$$\text{CARRA gust factor} = \frac{\text{1 hr forecast 10 m wind gust}}{\text{1 hr forecast 10 m wind speed}}$$

into the modelling, as a proxy for a reanalysis wind gust.

One issue with the CARRA gust factor is that it is not available for several time points including the period after the middle of 2022, whereas the CARRA reanalysis data used in this study extends through 2023, making it somewhat involved to assess its predictive power. Two investigations were carried out. First, the reduction in MAPE resulting from the addition of the CARRA gust factor to the baseline ws_{15}, secondary, and primary NN models of Section 4.2 was assessed, using only data with CARRA gust available. The results of this comparison are shown in Table 3.7

Table 3.7: Results of adding the CARRA gust factor as a variable in selected models from Sections 4.2 and 4.3.

	MAPE	
Model	Unaltered	With CARRA gust factor
Baseline ws_{15} model	17.0%	16.7%
Secondary model	13.3%	13.2%
Primary model	12.5%	12.4%

The causes of the differences between the numbers in Table 3.7 and those of Tables 4.2 and 4.3 could be different time period and/or unlike training setup.

Second, three simple linear regression models were constructed involving ws_{15} and the CARRA gust factor, using available data with $\mathrm{ws}_{15} \geq 10$ m/s. The results are shown in Table 3.8.

Table 3.8: Mean absolute percentage error (MAPE) for regression models using the CARRA gust factor, ws_{15}, and both combined.

Model	MAPE
ws_{15}	17.03%
CARRA–gf	16.81%
CARRA–gf + ws_{15}	16.80%

Finally, using the raw CARRA gust factor without any modelling gave MAPE = 18.14%.

The conclusion that can be drawn from this investigation is that there is a modest information in the CARRA wind gust forecast, potentiially reducing model MAPE by 0.1 to 0.3 percentage points. This reduction is however quite small compared with the overall MAPE reduction of ∼5 percentage points seen in Table 4.2 when all available variables are added.

4 Discussion and conclusions

4.1 Main findings

As shown in Table 3.3, the neural-network model achieves a significant reduction in mean absolute percentage error (MAPE) compared to a constant baseline. Table 3.2 further demonstrates an improvement over a simple regression model. A statistical test yields $p < 0.01$, confirming that these improvements are unlikely to be due to chance. The influence of individual features on model performance is illustrated by Shapley plots (Figure 3.4 and Appendix 4.4). Although one might expect stability indicators—such as the Richardson number and the Brunt–Väisälä frequency—to be among the strongest predictors of gust variability, they contributed less than other variables. For reanalysis wind speeds above 10 m/s, the network reduces error by about 30% relative to the regression baseline. The largest gains appear in the 10–25 m/s range (Table 3.6). At lower speeds, percentage errors inflate even when absolute gust–wind differences remain small; at higher speeds, sample sizes become sparse.

Predictive skill varies significantly from station to station. This variability can be partly attributed to discrepancies between reanalysis and observations (Table 2.2), differences in local terrain, and the number of measurements per site. Training separate models for individual stations did not improve performance (Tables 3.5 and 3.4); in fact, some well-sampled stations performed worse when modeled in isolation. Future work should investigate whether these anomalies arise from systematic measurement errors, inherent atmospheric variability at those sites, or other factors.

4.2 Limitations and future work

This study did not include station coordinates (XY) as direct inputs. Adding location information could improve site-specific accuracy without losing the benefits of training on the full dataset. Similarly, instead of applying a fixed linear interpolation to CARRA and elevation grids, the network could receive raw neighboring grid values and their relative positions, allowing it to learn an optimal interpolation scheme.

Finally, treating the data as a time series may offer further gains. Here, only instantaneous inputs at forecast time were used, partly because CARRA fields are available every three hours and may not capture rapid stability changes. Access to higher-frequency data or the inclusion of temporal history could enable models to exploit evolving atmospheric conditions and further enhance gust-prediction skill.

Many papers cited in this thesis suggest that adding a machine learning model as a backend to more traditional neural networks models could lead to improved performance. The results of this study support this.

4.3 Possible applications

Wind gusts are displayed on road signs in Iceland, if wind gusts exceed expected values. The model could be used to allow predicitons of wind gusts to be displayed. In 2023, outside of Reykjavik harbor, a cruise ship nearly beached due to an unexpected wind gust. This wind gust of ~26 m/s, pushed the ship into a buoy near Viðey. It was only luck that prevented the ship's screws from hitting the buoy. The ship was only 10 meters from the shoals of Viðey, where the water is only 0.4 meters deep. A better prediction of wind gust could have prevented this incident. Similarly, the model could be used to prevent damage to transportation vehicles, that take on a lot of wind, such as trucks and buses, or to prevent damage to buildings.

4.4 Conclusions

The neural network model presented in this study demonstrates a significant improvement in predicting wind gusts over a simple regression model, particularly for reanalysis wind speeds above 10 m/s. The model's performance varies by station, and while some stations benefit from the full dataset, others may require more localized training. The power generated by wind turbines increases with the cube of the wind speed [**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.

Appendix A: Feature importance on Shapley plots

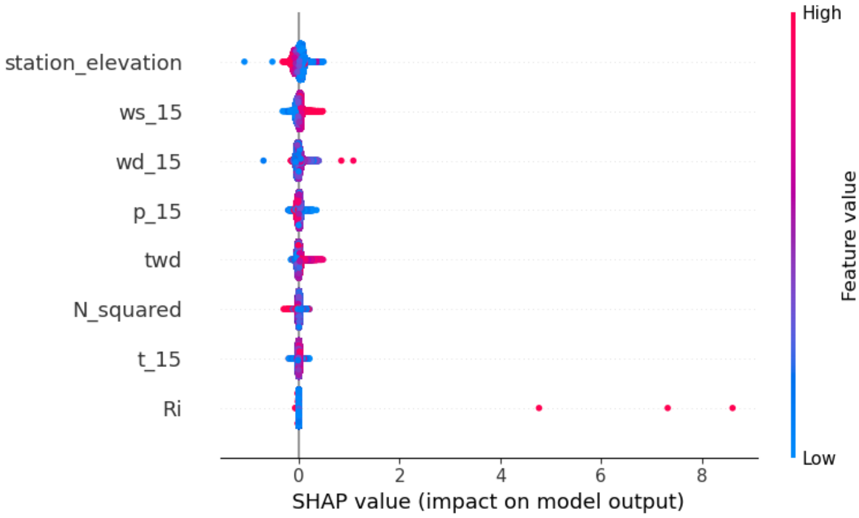


Figure A.1: Feature importance of a neural network with model architecture as described in Table 2.4 and data as described in Table 2.3. The distribution seems to be the same as before, discounting the outliers.

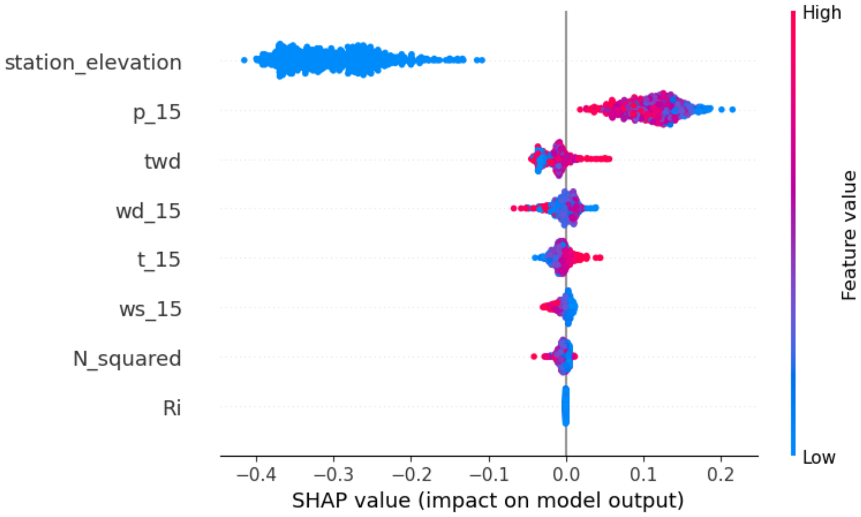


Figure A.2: Feature importance of a neural network with model architecture as described in Table 2.4 and data as described in Table 2.3. This plot only looks at datapoints from Ásgarðsfjall. This seems to show the same distribution as previous summary plots. Station elevation is influential and Richardson number has no impact.

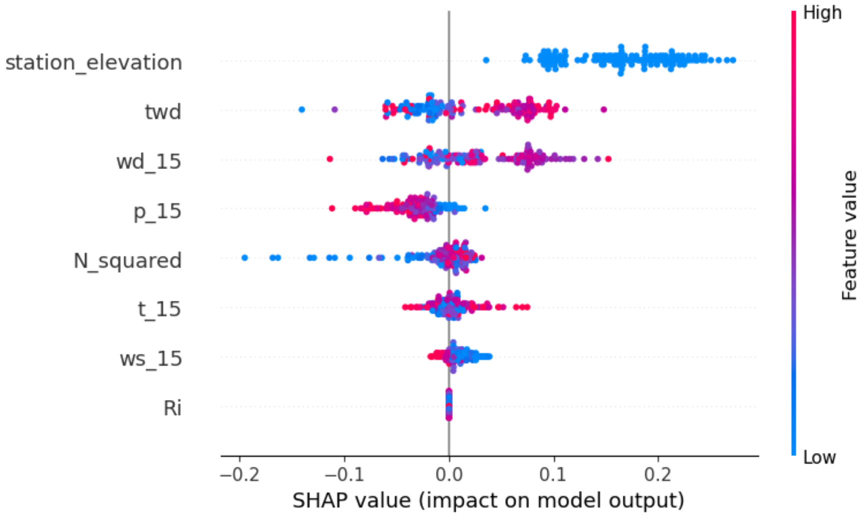


Figure A.3: Feature importance of a neural network with model architecture as described in Table 2.4 and data as described in Table 2.3. This plot only looks at datapoints from Háahlíð. This seems to show the same distribution as previous summary plots. Station elevation is influential and Richardson number has no impact.