UNIVERSITY OF ICELAND

A NN approach to predicting gust factors in complex landscape

Brynjar Geir Sigurðsson

June 2024

M.Sc. thesis
in Mechanical Engineering

FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

A NN approach to predicting gust factors in complex landscape

Brynjar Geir Sigurðsson

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

Supervisors
Kristján Jónasson // Ólafur Pétur Pálsson

M.Sc. Committee
Kristján Jónasson
Ólafur Pétur Pálsson

Examiner
XXNN3XX

Faculty of Industrial Engineering, Mechanical Engineering and Computer Science
School of Engineering and Natural Sciences
University of Iceland
Reykjavik, June 2024

A NN approach to predicting gust factors in complex landscape

60 ECTS 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 (2024) *A NN 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 2024

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

Abstract

English abstract (ca. 250 words).

Útdráttur

Hér kemur útdráttur á íslensku sem er að hámarki 250 orð.

Contents

List of Figures

List of Tables

Abbreviations

Í þessum kafla mega koma fram listar yfir skammstafanir og/eða breytuheiti. Gefið kaflanum nafn við hæfi, t.d. Skammstafanir eða Breytuheiti. Þessum kafla má sleppa ef hans er ekki þörf.

The section could be titled: Glossary, Variable Names, etc.

If you use acronyms, it is strongly recommended to use a LaTeX acronyms package. For example, to use the package *acronym*, add in the preamble:

```
\usepackage{acronym}
```

and then here in this chapter (to create the list of all acronyms), use:

```
\begin{acronym}[SENS]
  \acro{SENS}{School of Engineering and Natural Sciences}
  \acro{UoI}{University of Iceland}
\end{acronym}
```

In the square bracket above, you need to put the longest acronym, so that the list gets proper indentation based on the longest acronym. Also note that you need to manually sort here that list.

Then, whereever you use the acronym in your text: `\ac{UoI}`: that will automatically expand the acronym at the first use, but use only the short version after the first use. (Use `\acp` if need the plural form. `\acl` if you explicitly want to have the long form only, `\acf` if you explicitly want to have the full long and short form, i.e. like first use of `\ac`.)

Use `\acresetall` to forget about earlier usage (=expansion) of acronyms, e.g. if despite already used in Abstract or Introduction, but you want to expand them later once again (starting from, e.g., in Foundations chapter): add `\acresetall` at start of Introduction chapter (and maybe also again at start of Foundations chapter).

Acknowledgments

Í þessum kafla koma fram þakkir til þeirra sem hafa styrkt rannsóknina með fjárframlögum, aðstöðu eða vinnu. t.d. styrktarsjóðir, fyrirtæki, leiðbeinendur, og aðrir aðilar sem hafa á einhvern hátt aðstoðað við gerð verkefnisins, þ.m.t. vinir og fjölskylda ef við á. Þakkir byrja á oddatölusíðu (hægri síðu).

1 Introduction

Wind gusts are brief increase in wind speed (lasting seconds) as compared to mean wind speed. The gust factor is defined as the peak gust divided by the mean wind speed over some defined time period. The peak wind gust is often defined as the highest 3 second rolling average measured wind speed over a period of 10 minutes, while the mean wind is the average of all measurements in the 10 minute interval. This varies, with the US using a 1 minute interval, leading to 14% higher results [8]. As the Navier Stokes Equation (1.1) shows the variability of the wind, in time and space, is dependent upon the pressure gradient, the oscillating force of the earth, and resistance. It states that the change in wind is described by the pressure gradient, the oscillation, the acceleration due to gravity and the resistance from the landscape.

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

Equation 1.1 relates the change in wind w.r.t. time ($\frac{\delta \boldsymbol{V}}{\delta t}$) and change in wind w.r.t. position ($\nabla \boldsymbol{V}$) to pressure change w.r.t. position (∇P). The wind is also influenced by the oscillation, gravity and resistance but the main drive of wind is pressure gradient[1].

Traditionally, numerical weather prediction (NWP) systems are used to forecast and analyze weather patterns[2]. These models describe the transition between discretized packages of atmospheric states using partial differential equations based on physical reality. These results are usually published for 3-6 hour intervals.They describe the state over the period and so do not necessarily grasp fluctuations well. These fluctuations would include fluctuations in the wind speed, wind gusts[**canNNBeatNWPw**].

This thesis looks at how best to predict gust factor based on various factors, using several different data sources, including NWP and observations. NWP forecasts or outputs analysis, traditionally, at intervals of around 3-6 hours and thus, alone, do not capture the variability that is the essence of wind gusts. This is why another model that uses the information produced by the NWP along with observations and landscape information, is needed to try to accurately predict wind gusts. Being able to accurately predict the wind gust is important as it is often the peak wind gusts that will cause failures in structures. A problem that will become increasingly prevalent in the near future[7].

1.1 Background

The history of numerical weather predictions goes all the way back to the 1920's when Lewis Fry Richardson pioneered the field and tried to produce forecasts. The results were flawed and impractical because of the time it took to calculate was much longer than the forecast period. In the 1950's, with the advent of computers we get the first operational forecasts. In September of 1954, Rossby and his Stockholm based team produced the first real-time barotropic forecasts. The next year the Joint Numerical Weather Prediction Unit (JNWPU), based in Princeton New Jersey, released their first forecasts. These forecasts were for 36 hours at 400, 700 and 900 mb. The results were inferior to subjective human-based forecasts but showed that such forecasts were feasible and promoted further development in the area [4]. The field underwent significant transformation over the next 50 years, and predictions got significantly better.

Currently there has been another transformation in the field of weather prediction driven by artificial intelligence. Interest in AI has come in waves. Some progress is made, then interest dwindles. The interest has been increasing steadily since 2010. Notable work that has driven this wave of interest include increase in computational abilities due to parallel processing in graphical processing units (GPU), convolutional neural networks (CNN), which allowed much faster processing of massive (image) datasets and the availability of large datasets online. It is to be noted that images are grid data with some number of channels. Using CNNs could work on any gridded data where there is some spatial features[12]. Since 2018, there has been significant work done in the weather prediction field using AI. In 2018, Dueben and Bauer showed that you can build a NN that can outperform a simple persistence forecast and is competitive with very coarse-resolution atmosphere models of similar complexity for short lead times[3]. Also in 2018, Scher created a deep convolutional neural network (CNN) to emulate a general circulation model (GCM, a numerical model representing the physical processes), training on the GCM which allows it to emulate the dynamics of the model and maintain stability for much longer than Dueben[11]. These two papers were more proof of concept rather than production ready models to replace NWP. They showed that models based on deep learning might, with further development, compete with standard models in the field.

In the last two years there have been even more developments with the emergence of Large AI Weather forecast Models (LWM). In 2024 Ling et al. tried to standardize the definition of LWM in meteorology and came up with 3 rules that need to be met to count as LWM.

Rule 1: Large Parameter Count. The number of parameters can vary wildly but a general range might be from tens of millions to billions of parameters

Rule 2: Large Number of Predictands: predicting on different levels (such as pressure levels or height levels) and offering detailed information on the atmospheric vertical structure and surface conditions

Rule 3: Scalability and downstream applicability. This might crystallize in predicting cyclones. Often, the teams responsible for creating these models try to show their applicability to predict cyclones when not trained specifically on cyclone data (e.g. GraphCast)[6]. This is done to show the versability of the models.

Before 2022, LWM had been shown to be able to compete with traditional NWP for some specifc cases as well as making predictions quicker, after training. No model had shown that it could in any way completely replace the traditional systems. In early 2022, Pathak et al. presented FourCastNet. FourCastNet uses an Adaptive Fourier Neural Operator model that leverages transfomer architecture rather than the popular convolutional model architechture. FourCastNet matches the performance of standard forecasting techniques at short lead times for large-scale variables and outperforms for smaller variables. It generates a week-long forecast in less than 2 seconds, orders of magnitude faster than standard physical methods[9]. In 2022, machine learning methods were presented that made predictions much faster than traditional NWP, after a one time training (or at least training that wouldn't have to be redone often). These were in some cases performing better than NWP. In 2023, Remi Lam and the GraphCast team at Google introduced GraphCast. This model was able to outperform the industry standard High Resolution Forecast (HRES) produced by the European Centre for Medium-Range Weather Forecasts (ECMWF). This model as the name suggests leverages graphing connections rather than traditional grid like data structure. The base data is given in latitude and longitude degrees at a resolution of 0.25 degrees. This means points are closer to each other at the poles. Using the graphing structure is supposed to help with bias incurred as a result of this[5].

There has been a lot of progress made over the last 6 years (since 2018) and especially in the last 2 years (since 2022). The progression from machine learning methods being an interesting idea in the field of numerical weather predictions, to outperforming the standard NWP has been remarkably quick. Two years ago, machine learning methods were able to predict quickly and in some niche cases outperform traditional models. They were not generally competitive with standard weather models. Now they are becoming competitive. It will be very interesting to watch what the next few years will have in store for the development of machine learning in weather predictions.

1.2 Methodology and related work

This study looks at data from three sources and tries to predict the gust factor in a given place in Iceland. It tries to use reanalysis data, along with elevation data and predict the gust factor. It looks at the data at the point of interest and does not look at the data as a time series. This thesis tries to improve on the baseline model of always predicting the mean and show that some structure can be learned from reanalysis data about gusts. To do this a neural network was created. The goal of this neural network was to predict the gust factor and improve upon a baseline of always guessing the mean gust factor. This baseline model gave an average error of around 10%. Any significant improvement on this would indicate that the final model has something to contribute.

In 2004, H.Ágústsson and H. Ólafsson looked at the variability of gust factor in complex landscapes. They looked at data from automatic weather stations that measure wind at 10 meters above ground. A common definition of wind gust is the average 1 second measured wind over three seconds. This is a rolling average. A common definition of average wind is the average of the measured wind over 10 minutes. These are the definitions used both by H. Ágústsson and H. Ólafsson in 2004 as well as this study. The data that was studied in 2004 comes from the same source as used in this study, but limits itself to a smaller section. They only looked at the years 1999-2001. They looked at three factors and how these effected the gust factor. These were d_m, D, H, that is direction of wind blowing off a mountain, distance to the mountain and the height of the mountain above the weather station. Their main results were these, gust factor is inversely correlated to the distance from a mountain and correlated to the height of the mountain. The study in 2004 looked at the effect of a dominant point upwind. It did not look at the effects of the landscape more broadly. In this study, landscape upwind is looked at along with landscape downwind. Both the upwind and downwind landscape can influence the gustiness at a certain point [10].

To be able to capture the patterns in the data we construct a neural network. The choice to select a neural network is one that is made because they are known to capture well patterns in complex data and can be applied to varied problems. This comes in handy when training on different types of data. It is also easy to construct different types of neural networks and see how they fit well with parts of the dataset. To measure the performance of these models, both to train and test, mean absolute percentage error was used. This measures the average percentage error. This was chosen because the target is the gust factor (the wind gust over the average wind). If the target would have been the wind gust rather than the gust factor then something like mean absolute error might be more appropriate.

2 Data gathering and preprocessing

Data was sourced from several streams. Veðurstofan provided recordings from weather stations all around the Iceland. Data was downloaded from Copernicus Arctic Regional Reanalysis dataset (CARRA). A land elevation model was also provided by Veðurstofan.

2.1 Automatic Weather Station Data

Veðurstofan provided recordings from weather stations all around Iceland. These automatic weather stations (AWS) are at a height of 10 meters above ground level. The information that is provided by these AWS is presented in two different type of documents, hourly and 10 minute documents. The hourly documents are summations of the 10 minute documents, with the exception that any errors (so called nails) still in the 10 minute documents should have been removed from the hourly documents. Each type of document contain the following information: the date and time, the station number (that can be converted to the coordinates using another document), the average wind speed, the wind gust, the standard deviation of the wind gust, the direction of the wind and the standard deviation for the wind direction. These features for each of the stations go back at least two decades, but do not have the same start date. This paper does not look at the data as a time series, it tries to make predictions using only the information at a given point in time.

2.2 Carra Data

The CARRA dataset goes back to September 1990 and is currently updated monthly, with a latency of 2-3 months. The oldest Vedurstofu data example that fulfills given criteria is from 2004 . This is covered by Carra. A few of the newer points from AWS were not available from Carra when data was gathered. The Carra dataset is available for two regions, west and east. Each of these covers a vastly larger area than the area of interest. It is possible to request a subsection of the area, a rectangular subsection of the area. This leads to having to store a large amount of data, because we cannot simply ask for the specific points of interest. To get the data one has two options. Their web interface or using their API client. Using the API client is the only realistic option here, as we are making thousands of requests for different times.

Each request is made by going through all instances of the Veðurstofu data were able to fulfill the requirements (of mean wind speed reaching 25 m/s over a 10 minute interval). For each such observation in the Veðurstofa gögn, we need to make two API (calls excluding redundancy). One for the 3 hour interval before and one for the 3 hour interval after. That is if the observation was at 1 pm, we would need Carra data for noon and 3 pm. Date and times were generated automatically from the AWS and requested by calling the client. Using these datapoints, we should be able to bridge and get an estimation for the point of the given weather station. The Carra data contains several types of layers. These are single levels, model levels, height levels, pressure levels. The data for this observation was downloaded from height levels. That is, data was requested at heights of 15, 150, 250 and 500 meters above ground.

For each point 4 features were requested, wind speed, wind direction, pressure and temperature. Each of these features needed to be bridged to create data for model to be trained on.

After using this method to request terabytes of data, it was discovered that it is apparently possible to query a specific area. This decreased the size of each file created after a request from around 50 MB to around 2 MB. Downloading and processing this data would take a lot less time. The largest bottleneck in terms of retrieving the data is the request time. That is the time the request is queued and the time it is running before it begins downloading. This could range from almost immediate to 15 minutes or more. This problem was exacerbated by the fact that the climate data store (CDS) was undergoing updates during the winter and spring, which increased the wait time and sometimes resulted in queries not being responded to. This meant that the time it would take to retrieve the remaining information went from around a day, when the requests were at their quickest, to many months, something that would not be possible given the time frame of the project.

2.3 Elevation data

Veðurstofa provided a tif file containing the elevation of Iceland on a 20 meter by 20 meter grid. This file encompasses Iceland and is around 685 Mb. At first I thought

I might have to split the file into subsections and read for each one but that doesn't seem to be the necessary. A good amount of time was spent finding a data structure that was best for lookup when trying to find points within a certain area. The country was divided into 10 parts (as only around 13% of the file was able to be read into memory at each time as a part of dictionary object) with boundary boxes. A quadtree was constructed for each of the sections, so as to simplify the lookup. A quadtree allows for efficient lookup by searching for points that intersect a given boundary. A quadtree, like the name indicates, is a tree structure. To initialize it you give it a boundary box. This boundary box is then the parent of exactly four nodes and so on, dividing the area in four at each level and allowing quick lookup. This was not necessary as the rasterio package allowed for quick lookup with it's index and the affine transform.

3 Data processing and structure

3.1 Combining data sources

This project used three main data sources, which need to be queried, filtered and combined to prepare the data for use in the models. When working with hundred of thousands of rows, the efficiency of the code is very important. Iterating through those rows might be necessary at times but will increase the time exponentially as compared to using vectorizing methods were possible. The three data sources were all in a different format. Measurement data from Veðurstofa was in text files, elevation data was in GeoTiff and reanalysis data from Carra was in a GRIB format. To use the data to train, these three data sources needed to be combined into one file. This was done based on the measurement data from the Veðurstofa. A limit was set on the average wind speed and it was used to select measurement points. Along with the average wind speed having to be above a certain limit, to make sure that we were not essentially getting duplicates, we would select only the top wind speed in any given 48 hour period. That is, for a data point to be included it must have been the highest wind speed in the previous and following 24 hours, so as to not describe the same weather multiple times. The data from Veðurstofan was supplied for 10 minute increments, while Carra data is in 3 hour intervals. This means that to use the Carra data to predict the measured values from Veðurstofa, temporal bridging would need to be done. Along with the temporal bridging, we note that the Carra data is given in a rectangular grid where the distance between each point is around 2.5km while the the information from the Veðurstofa is given at specific locations. The elevation information was given by a 20 by 20 rectangular grid that covers Iceland. When combining these data sources a selection of bridging needs to occur. The bridging was linear, both temporally and spatially. This might influence the results but was not considered in this study.

The procedure of combining these sources was as follows. We start with the measured data from the AWS, we filter this by using a limit on the average wind speed. The gust factor generally drops with increased wind speed (although not always dependent on the factors such as the landscape [10]). Even so being able to predict the gust factor is more important for higher average wind speed as there we have the highest wind gusts. After this stripped dataset over every AWS has been created it is used to query the Carra data by using their API. When querying the Carra API we will have to query in such a way that we query for given hours, days, months, years and a given area. That is, if we ask for a given hour, we get that hour for every day that we ask for. Similarly if we ask for a given day, we get that day for every month we ask for. In light of these restraints, it was decided to query month by month. Querying only the days needed (both the days included in the 10 minute measurements and the days needed for bridging if close to midnight) but every hour of the day (midnight, 3 AM, 6 AM, 9 AM, midday, 3 PM, 6 PM and 9 PM) and then take these values and bridge for the 10 minute values. After querying and downloading the data for the height levels and variables requested, bridging is done for the points of interest and values stored in a pandas dataframe. After this is done the downloaded data is discarded and we look to the next month. This drastically decreases the amount of data that needs to be stored as compared to downloading the entire area and keeping all the data points (goes from several terabytes to less than a gigabyte).

The elevation data comes in a GeoTif file that covers Iceland. It is a rectangular grid of resolution 20 meters. For every point of interest (every weather station), the elevation of that given point along with other points surrounding the weather station is retrieved. For each point retrieved bridging needs to be done. This is done in a similar manner to the briding of the Carra data. Weights are assigned to each of the four points bounding the point of interest. These weights are then used to calculate the weighted average that represents the bridged value. This information is included in the training data as the landscape is known to influence both the average wind and the gustiness [10].

3.2 Data structure

Once data has been retrieved for all three sources and processed, including bridging values, it needs to be made ready to use by the model, for both training, validation and test. We start with a dataframe that contains the 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 we choose certain height levels. That is at which heights we want the reanalysis data. These present as separate lines in the Carra dataframe. We need to combine them so that each line indicates a single observation. When this is done we can combine the AWS Veðurstofu data and Carra reanalysis data on the location and time columns. Lastly we look at the elevation data. A couple of different sections of land around the weather stations have been looked at. We tried a sector of a circle looking upwind, two sectors looking upwind and downwind and a circle around the point. In any case the points, that represent these sections, was selected like as shown in code listing 3.1. That is we look at the wind direction d and define a range of angles around that direction at some distance from the given point. This means that we get some number of points (equal to the length of angleRange) at a distance from given weather station as defined by length_rng. This then gives us some sectors at some distances away from the weather station.

Listing 3.1: 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])
```

We end up with 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 3.1.

Ri_01	Ri_12	N_01	N_12	station_elevation	relative_corner	PC1	...
-1.18e+00	2.67e+04	-8.57e-06	6.78e-05	3.34e+01	2.73e+00	1.36e+01	...

Table 3.1: An example of data structure used with model

Looking at table 3.1 we note that the last 10 columns represent the principle component analysis of the elevation data. We also note that the first four columns represent two variables that describe the stability of the air. These are the Richardson number (Ri) and Brunt–Väisälä frequency (N). They are calculated using equations 3.1 and 3.2. They are calculated using reanalysis data about the the weather at two different height levels. Thus Ri_{01} refers to the Richardson number calculated between height levels 0 and 1. Exactly the same notation is used with the Brunt–Väisälä frequency. The height levels used were 15, 250 and 500 meters above the ground.

$$Ri = \frac{g \cdot dT \cdot dz}{T_{ave} \cdot dU^2} \tag{3.1}$$

$$N = \sqrt{\frac{g \cdot dT}{T_{ave} \cdot dz}} \tag{3.2}$$

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 tell us something about the stability of the air. These are derived factors from the reanalysis data and as such there shouldn't be a significant information gain in calculating the as opposed to having the raw data. Including these factors instead of the every variable might speed up training as well as making the model more easily explainable with the use of Shapley values or other tools for explainability.

4 Model architecture

4.1 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 figure out the best hyper parameters. 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 4.1.

Layers	4, 5, 6, 7, 8, 9
Units	128, 256, 512
Epochs	100, 200, 500
Batch size	32, 64, 128, 256
Optimizers	Adam, RMSprop, Adamax
Penalties	1, 0.1, 0.01, 0

Table 4.1: Hyperparamter search with best performing combination colored red

4.2 Feature selection and importance

Using the three datasources, we train a model. The data from Veðurstofa only represents the ground truth and is not used as part of the training data. We use the gust (f_g) and wind speed (f) to calculate the target gust factor. From the Carra reanalysis data we get all of our weather information, with variables wind speed, wind direction, temperature, pressure. All of these variables are queried in three height levels (15, 250, 500 meters above ground) at the location of a given weather station. In addition we have the elevation above sea level over Iceland. We select the elevation in a couple of different shapes. A sector upwind, a sector upwind and downwind as well as a circle surrounding a weather station.

To add to these observed values and reanalysis, we calculate two derived variables, the Richardson number and the Brunt-Väisälä frequency. To begin with we simply use all the features we have to train our model along with the derived variables to see. We do this to be able to then use tools such as Shapley to see which features are impacting our predictions. Using Shapley values to see which features, will then allow us to exclude features that don't affect our model prediction and simplify the training data. We thus start with much higher dimensionality of our training data and reduce it as much as we can without impacting our results. This crystallizes in the landscape points. We start with n points that describe the elevation of the landscape around our weather station. This n might be in the hundreds. If we look at concentric circles outward from the weather station, we might have 10 circles (largest with radius 20 km), each with 72 points (5°spacing) and end up with 720 points for the landscape. This might slow down training. In addition, we might imagine that we are more interested in certain important landscape features. There might be a large mountain up- or downwind of our weather station, that might be the most important feature, in addition to some hills and such. To address this we use principal component analysis (PCA) to try to reduce the dimensionality of our landscape data. Another thing that might serve the same purpose is to select some number of m points that have the most increase or decrease in elevation from the weather station along with the coordinates of those points in relation to the weather station and the direction of the wind.

Using all the variables from Carra, along with the two derived factors (Ri, N) and two sectors (upwind and downwind) downscaled to 10 features using PCA we can see the feature importance in figure 4.1. Looking at the waterfall graph we can see that Richardson number for higher levels is most important.

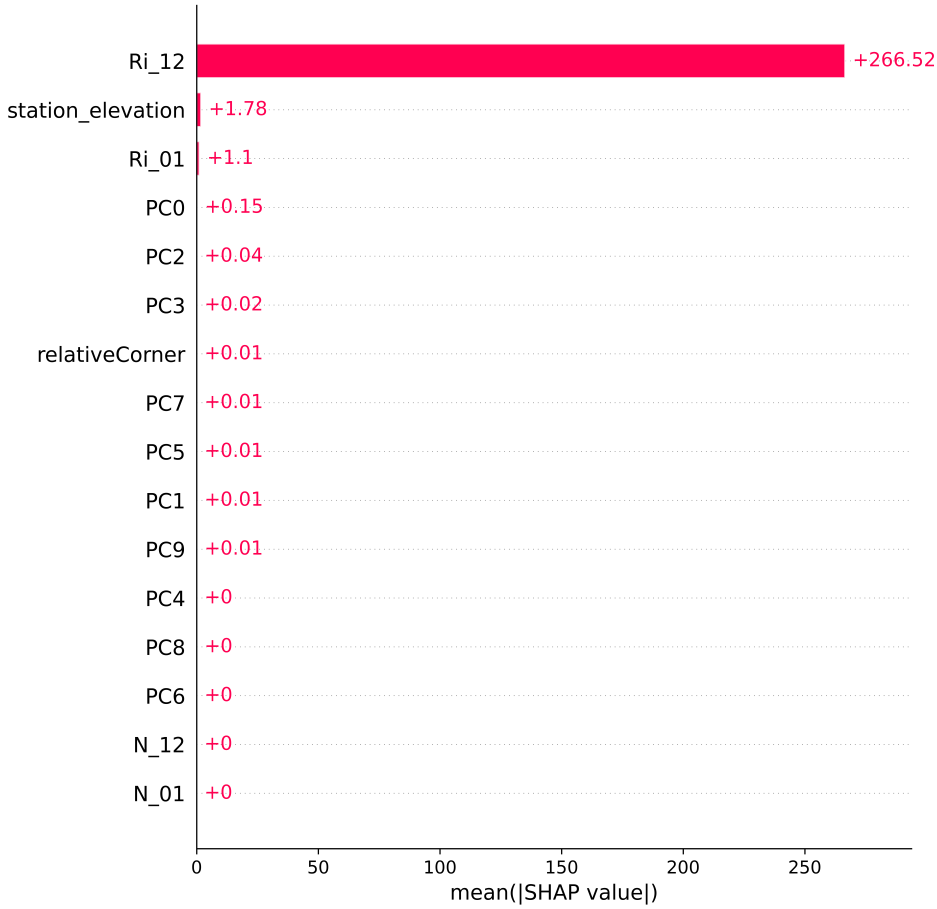

Figure 4.1: Feature importance of a neural network as described with 5 hidden layers and 256 units in each