ETH zürich

IGP
Institut für Geodäsie und
Photogrammetrie

Master Thesis

Machine learning approaches for estimating tropospheric zenith path delay

Luca Miotti

Autumn Semester 2019

Professorship

Prof. Dr. Markus Rothacher

Supervisors

Endrit Shehaj

Dr. Stefano D'Aronco

Dr. Jan Dirk Wegner

Prof. Dr. Alain Geiger

The accuracy of GNSS measurements depends heavily on the delay its signal experiences on its path from the satellite to the receiver. This delay is caused by various influences. One of those influences is the refractivity of the troposphere which originates from dry gases and water vapour. It can be represented and modeled with three meteorological parameters: pressure, temperature and partial water vapour pressure. The applied models are however highly empiric and can have problems to accurately determine the path delay in some cases. This is especially the case for the part of the delay caused by the water vapour (wet delay).

An alternative approach of estimating the tropospheric path delay could be with the application of machine learning algorithms. During the last two decades machine learning algorithms have become widely used in many fields of science and engineering. Such algorithms allow the extraction of relations in large datasets without the need of specifically modeling said relations in advance.

Thus in this thesis the application of two vastly different machine learning techniques to estimate the tropospheric zenith path delay is examined. Namely these techniques include random forest and a fully connected artificial neural network (ANN). Both algorithms use meteorological parameters as well as the resulting zenith path delay to create a model that should accurately reflect the relation between those variables. It is inspected how well the algorithms perform compared to each other and what the biggest influences on the accuracy of the resulting predictions are.

To further assess the quality of their predictions, the results are finally compared to a more common approach of modeling the zenith tropospheric delay.

Table of content

List of figures

List of tables

.

1. Introduction

1.1. Problem

Global Navigation Satellite Systems (GNSS) base their estimation of a position on microwave signals transmitted from different satellites. By measuring the time the signal took to get from satellite to receiver the distance it traveled can be derived. With signals from at least four satellites the corresponding distances can be used to compute an intersection and therefore determine the position. Thus to be able to accurately determine the position, the path of the signal needs to be represented as accurately as possible. On the path from the satellite to the receiver many factors can influence the travel time and so cause a delay. Such a delay also occurs during the path of the signal through the troposphere.

To determine the distance of the path of the signal, the speed of light in vacuum is used. This speed however changes slightly based on the medium the signal propagates through. The delay can be expressed by the refractivity index of the particular medium. So to determine the delay in the troposphere, the refractivity index along the signal path can be integrated:

$$\Delta\rho_{TROPO} = 10^{-6} * \int_{Slant\ Pat} N(s)\, ds$$

In the troposphere is the refractivity caused by dry gases and water vapour. The delay is therefore split up into a dry and a wet part.

$$\Delta\rho_{TROPO} = 10^{-6} \int_{Slant\ Path} N_{dry}(s)\, ds + 10^{-6} \int_{Slant\ Path} N_{wet}(s)\, ds$$

Although the dry delay usually makes up a large part of the total delay – around 90 % - can it be modeled quite accurately as the causing dry gases stay relatively constant in time. The water vapour in the troposphere however has high temporal and spatial variations. This induces high uncertainties when modeling this part of the delay. At sea level the total tropospheric path delay amounts up to about 2.3 m.

The refractivity index can be determined by empirical models based on three meteorological parameters. These usually consist of the temperature, pressure and water vapour partial pressure. [Troller, GPS based Determination of the Integrated and Spatially Distributed Water Vapor in the Troposphere, 2004]

1.2. Zenith Path Delay

The path delay described so far corresponds to the direction from the receiver to the satellite (slant direction). This delay however depends highly on the position of the satellite. Since it is easier to compute the path delay for a more general case, empirical models usually determine the zenith path delay (ztd). This value covers the delay in the zenith direction of the receiver and is more generally applicable. To get from zenith delay to the slant delay a mapping function is applied. The most simple mapping function looks as follows:

$$m_t(z) = \frac{1}{\cos(z)}$$

Where z corresponds to the angle to the satellite. In the course of this thesis however only the prediction of the zenith total delay is inspected.

1.3. Proposed approach

This thesis investigates the potential of machine learning in general to accurately predict the zenith delay based on tropospheric parameters. In a first step the delay from all available GNSS stations as well as the data from nearby meteorological stations is split up into two different datasets with respect to time. Two different machine learning algorithms then learn the relation between tropospheric parameters and zenith total delay from the first of the two datasets and make their predictions using the meteorological values of the second dataset. These predictions are then compared to the actual delays of the second dataset to evaluate their accuracy. This should provide answers how well both approaches can estimate the zenith delay based on meteorological data for known locations but at unknown times.

The first of the two algorithms is called random forest. Random forest is a popular and capable machine learning technique widely used for both classification and regression. It is used as baseline to evaluate what accuracy is generally possible with powerful machine learning approaches. The second one is a fully-connected neural network. For difficult problems where a large dataset is available, neural networks in some form or another have become the standard approach in the last decade. Both algorithms are trained and make their predictions on the same respective datasets. These individual predictions are then compared and it is evaluated if and by how much neural networks could improve the accuracy of the predicted delay. To have a rough target regarding the accuracy of the predictions, a minimum goal of 2 cm for the overall root mean squared error is set. As can later be seen is this easily achieved.

The accuracy of the predictions of the neural network are then further analyzed with respect to the elevation of the different stations as well as their dependence on the time of year.

Additionally is it tried to evaluate how the results could depend on the distribution of the corresponding meteorological stations.

To further assess the quality of the predictions, a comparison with a standard approach is performed. This standard approach uses the empirical Saastamoinen formula which estimates the zenith path delay based on the same tropospheric parameters. To get reliable meteorological data at the locations of the GNNS stations, a collocation of the corresponding dataset is computed. This is followed by an interpolation at the desired locations. By doing so, the exact same data as for the machine learning approaches is used.

Finally the network's ability to predict at locations which it has never seen before is evaluated. This corresponds to not only a temporal but also a spatial decorrelation between the dataset the network is trained on and the one it is supposed to make its prediction on. Three GNSS stations – all within different distance to their respective nearest GNSS neighbour station – are completely excluded from the dataset the network learns from. The predicted results for these three stations are then analyzed and compared. As will be seen does this spatial decorrelation introduce a systematic error in the predictions. It will finally be analyzed if this systematic error can be prevented by only using a very small sample size of the initially excluded stations in the dataset the network is trained on.

1.4. Motivation

Machine learning algorithms have become widely popular in almost every field of science or engineering during the last two decades. This was on the one hand caused by the refinement or development of new algorithms and on the other hand by ever more powerful and cheaper computational devices. They allow for extracting complex relations in large amounts of data and once the underlying models are built usually do so very time efficient. So in general it can be said that machine learning techniques can be used to both increase accuracy as well as lower the computational time compared to more usual, empirical approaches. This comes however at the cost of the "black box problem". For most techniques – especially the ones based on neural networks – it is very hard to retrace how the algorithm has come to its conclusion. This retraceability issue is currently a topic of many research projects. See for example [Fan et al., On Interpretability of Artificial Neural Networks, 2020]

In the case of satellite geodesy a few similar approaches have already been tested and have shown the successful application of machine learning techniques:

- [Kitprache et al., MACHINE LEARNING BASED PREDICTION OF ATMOSPHERIC ZENITH WET DELAY: A STUDY USING GNSS MEASUREMENTS IN WETTZELL AND CO-LOCATED VLBI OBSERVATIONS, 2019]

  The authors use a Long Short Term Memory (LSTM) neural network approach in combination with the SSA+Copula method to predict the future zenith wet delay based on past time series of temperature, water vapour pressure as well the wet delay itself. They achieve a mean absolute error of around 1 cm for a prediction of the next 24 hours.

- [Shamshiri et al, A machine learning-based regression technique for prediction of tropospheric phase delay on large-scale Sentinel-1 InSAR time-series, 2019]

  This project uses a Gaussian Process (GP) regression model based on the zenith total delay to predict corrections for the tropospheric phase delay in InSAR time series. An improvement of the correction of 81 % as well as a reduction of 50 % of the root mean squared error compared to the usual approach was achieved.

Especially for applications in mobile devices such approaches could prove very impactful as pre-trained models can make accurate predictions without the need for large computational power. Additionally will an increasing amount of data be available from widely distributed low cost receivers in the near future. Fast evaluation of this ever growing amount of data is arguably the main advantage of machine learning approaches. Although as the mentioned cases have shown is there definitely a possibility to increase the accuracy compared to more usual, empirical approaches. The near term applications for purely scientific purposes is although somewhat debatable as time efficiency in post processing is less of a concern and the lack of interpretability of the results can be problematic in certain cases. This discussion is however outside of the scope of this thesis and it is left for the reader to decide.

2. Dataset

2.1. GNSS data

The zenith total delays (ztd) used for this thesis originate from 72 GNSS stations. These stations belong to the AGNES and COGEAR networks and can be seen in Figure 3 marked in orange. These networks are operated by Swisstopo as well as the Mathematical and Physical Geodesy group (MPG) of ETH Zurich respectively. Figure 1 shows the time series of the delay for three individual stations. The first one is the very highest station of the network at almost 3'500 m elevation and is located in the canton of Valais. The second one resides in Ardez in the canton of Grisons at an elevation of 1499 m. Finally the last one is at ETH Zurich at an elevation of 547 m.

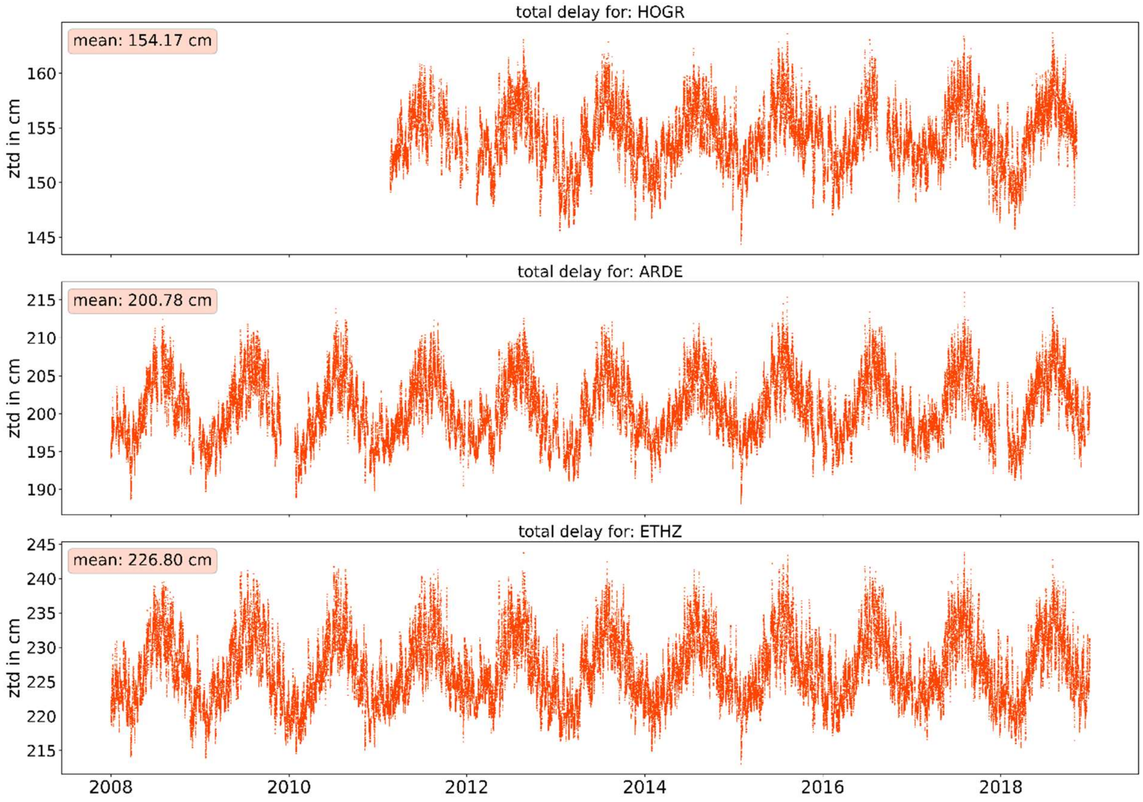


Figure 1: Time series of zenith total delay for three individual stations

The same seasonal oscillations can be identified in all three time series. In the winter months the delay seems to be much shorter. This can be explained with the much smaller wet delay during these time periods. Additionally is the delay for the first time series generally much lower than for the other two, which is also reflected by the respective mean values. This obviously corresponds to the elevation of the particular station, as the less the signal has to propagate through the troposphere, the smaller its effect – and therefore the delay – on it is.

2.2. Origin of zenith total delay

All of the positions of the stations are determined highly accurate as they are part of the mentioned geodetic networks. These accurate positions can be used to extract particular information about the incoming GNSS signals by estimating the desired quantities during the position adjustment of the station. Such information can also include the slant path delay of all incoming signals. They are mapped individually into the zenith direction and so combined into a single parameter. This parameter is then estimated as an unknown during the GNSS adjustment. This results in an accurate zenith total delay which can then be used as a target value for the applied machine learning approaches. This processing of the zenith delay is done with the Bernese software.

[Troller, GPS based Determination of the Integrated and Spatially Distributed Water Vapor in the Troposphere, 2004]

2.3. Meteorological data

As mentioned before is there a need for three different parameters to reliably estimate the refractivity of the troposphere. One of those parameters covers the effects of the temperature, one covers the pressure and the last one the water vapour content. In this case the water vapour content is given as the pressure it causes.

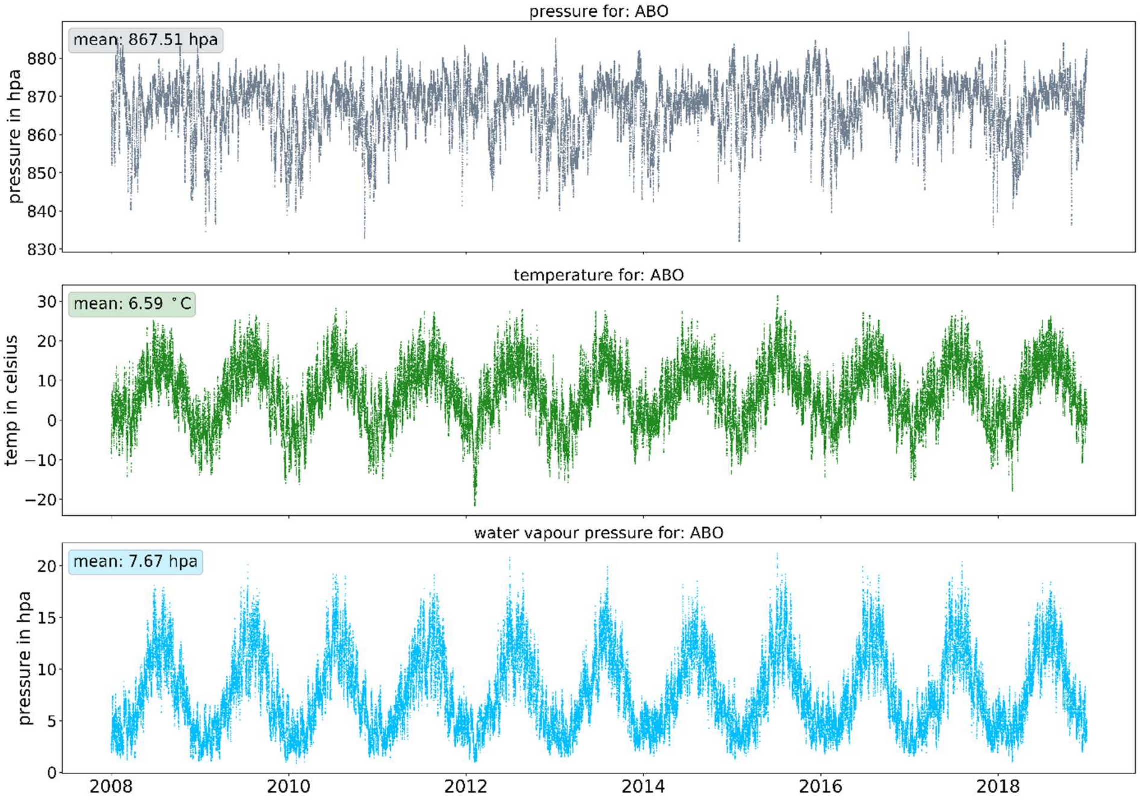


Figure 2: Time series for pressure, temperature and water vapour pressure for ABO

The meteorological stations which provide data are marked blue in figure 3. All stations belong to the SwissMetNet network operated by MeteoSwiss.

Figure 2 shows time series for the three mentioned parameters at the station in Adelboden located in the canton of Bern. This station is located at a height of 1324 m. Again can strong seasonal oscillations be identified in the respective time series.

2.4. Timeframe

Both meteorological data and the ztd time series are available in an hourly interval. The range of the timeframes consists in general of eleven years and lies between the beginning of 2008 and the end of 2018. The delay was available for an even longer time period but was cut to match the mentioned timeframe. Its data alone was of no particular use. As can be seen in figures 1 and 2 does the available data differ in all time series. Especially for the delay there are large periods for some individual stations where no data was available. In the case for the meteorological data this is more a case of singular observations for some points in time missing

2.5. Distribution of stations

Figure 3 shows the locations of all 72 GNNS stations as well as the SwissMetNet meteorological stations. It can be seen that the available meteorological stations are distributed quite evenly all over Switzerland. Furthermore does it show that almost all GNSS stations have multiple meteorological stations in their somewhat close proximity. This can also be seen in table 1. That table shows the average of the distances of each GNSS station to its respective four nearest meteorological stations.

As can be seen are all mean distances for the three nearest stations well below 20 km. Only the fourth nearest meteorological station is on average slightly further away.

	To first station	To second station	To third station	To fourth station
Mean distance [km]	8.2	13.2	17.7	21.5

Table 1: Mean of distances from all GNSS stations to their 4 nearest meteorological stations

Since the goal is to primarily predict the delays on the locations of the GNSS stations, their distances relative to each other and their much more inhomogeneous distribution should not pose a problem for now. A table listing the distance from every GNSS station to each of its four nearest meteorological stations can be found in appendix A.

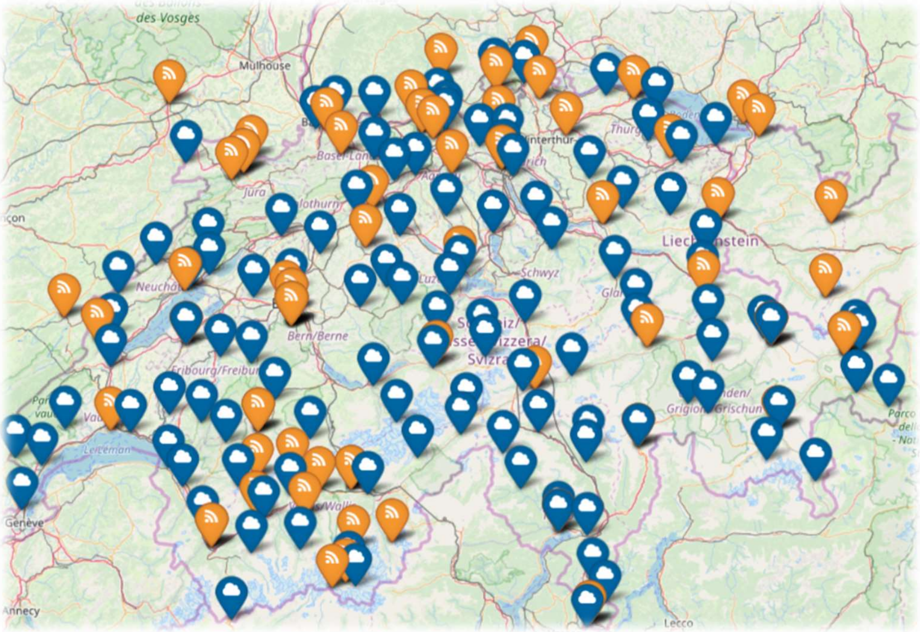


Figure 3: Distribution of GNSS and meteorological stations

2.6. Accuracy of the data

MeteoSwiss and Swisstopo as the providers of the data did remove all outliers according to their own information. They did however not provide any information on the exact accuracy of the data. Following are however some conservative assumptions for one sigma (standard deviation).

- ZTD: ca. 1 cm
- Pressure: 0.15 hPa
- Temperature: 0.2 K
- Water vapour pressure: approx. 0.5 hPa (relative humidity: 3%)

These values originate from [Hurter, GNSS Meteorology in Spatially Dense Networks, 2014], where further details can be found.

2.7. Feature selection

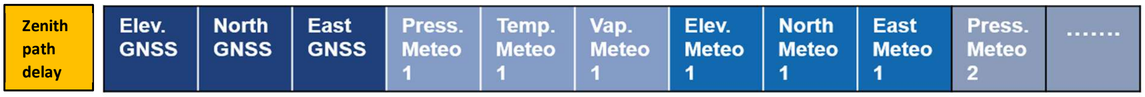
To make the time series data useable as input for the used machine learning algorithms they had to be processed. By doing so the time series were split up into individual samples. Samples consist of a certain amount of features describing different values at a particular point in time as well as a label – the zenith total delay.

The samples used in this thesis contain on the one hand the absolute coordinates of the GNSS station in the LV95 coordinate system, where the delay should be predicted. And on the other hand information of the four closest meteorological stations. These four station are determined by simply calculating the 3-dimensional euclidean distance:

$$distance = \sqrt[2]{(elevation\ gnss - elevation\ meteo)^2 + (north\ gnss - north\ meteo)^2 + (east\ gnss - east\ meteo)^2}$$

The number of four nearest stations was chosen to make sure that the GNNS stations have corresponding meteorological data from multiple directions. Since the distribution of the meteorological stations is quite even, this should hold true for most of the GNSS stations . Some preliminary tests were performed to see if adding additional stations would have improved the performance of the predictions but no evident increase of accuracy was observed. The exact influence the number of meteorological stations have on the accuracy of the result was however not closely evaluated in the course of this thesis and might be a topic for future work.

The data from those four meteorological stations contains their absolute coordinates as well as the three mentioned tropospheric parameters: Pressure, temperature and water vapour pressure. In total each sample then consists of 27 features and the corresponding label.



Zenith path delay	Elev. GNSS	North GNSS	East GNSS	Press. Meteo 1	Temp. Meteo 1	Vap. Meteo 1	Elev. Meteo 1	North Meteo 1	East Meteo 1	Press. Meteo 2	…….

Figure 4: Representation of an individual feature

Other features – derived from the original ones – such as relative distances and weights based on those distances have also been tested without any improvement of the results. It might be argued however that depending on the task to solve, relative distances could be preferred to absolute coordinates.

2.8. Handling missing data

As already mentioned are no of the given time series exactly equivalent. So to extract the described features of a sample only points in time were suitable where data was available in all of the involved 13 time series. This was done in two separate steps:

- Filtering of the meteorological data:
    - Iterating through every individual meteorological station and search for the points in time that exist for all three different time series for that particular station
    - Excluding all other data
- Filter of the GNSS data:
    - Iterating through every individual GNSS station and obtain the previously determined corresponding 4 closest meteorological stations.
    - Computing the intersection of points in time for the GNSS data and the meteorological data of the nearest station
    - Exclude all other data
    - Iteratively compute the intersection of the already existing intersected points in time and the time series of the next meteorological station and exclude all other data points

This reduced the theoretical amount of samples for an hourly time series of 11 years from 6’937’920 to 4’215’193.

2.9. Creating training, validation and test datasets

Once the samples are generated they are furthermore split up into three different datasets:

- Training dataset: The training dataset is the part of the data the algorithms are actually trained on. The model of the algorithm is iteratively feed with samples of this dataset as well as the corresponding labels to extract relations between the input data as well as their label. In this case this corresponds to the described features as well as the resulting zenith total delay. Once the training period is finished, this dataset is not used any further.
- Validation dataset: The validation dataset is used during the training of the neural network to monitor the progress of the training by constantly making predictions on this dataset. By doing so it can be checked that the network does not start to overfit on the training data and is able to generalize. Additionally can an instance of the model be saved when the achieved loss on the validation dataset has improved. This dataset is not used when applying the random forest.
- Test dataset: This dataset is used to evaluate the ability of the trained model to correctly assign values to data it has never seen before. The test dataset is fed to the trained model and the resulting predictions are then compared with the actual labels of the corresponding samples.

To achieve a certain decorrelation of the datasets, they are split in time. The training dataset consist of the years 2008-2016 , the validation dataset of the year 2017 and the test dataset of the year 2018. This results in a sample total of 3’069’950, 566’690 and 578’553 respectively.

2.10. Standardization of the data

For a neural network application the processed data has furthermore also to be standardized. This is performed by subtracting the mean from the data and then dividing it by its standard deviation. This is done for each feature separately:

$$feature_{stand} = \frac{feature - mean(feature)}{\sigma(feature)}$$

By doing so the values of each feature get first centered around zero and then scaled to a standard deviation of one. This is mainly done for numerical reason. As will be described in the next chapter is the training of a neural network based on the gradient descent of a loss function. Highly different values in the range of different features – as for example would be the case for a temperature value compared to the north value of absolute coordinates – could lead to a very steep gradient. This could then prevent the network from finding an optimal solution during the training.

It is important to note that this is applied for training, validation and test dataset but that mean and standard deviation of the features always originate from the training dataset.

Figure xy shows the concept of data standardization.

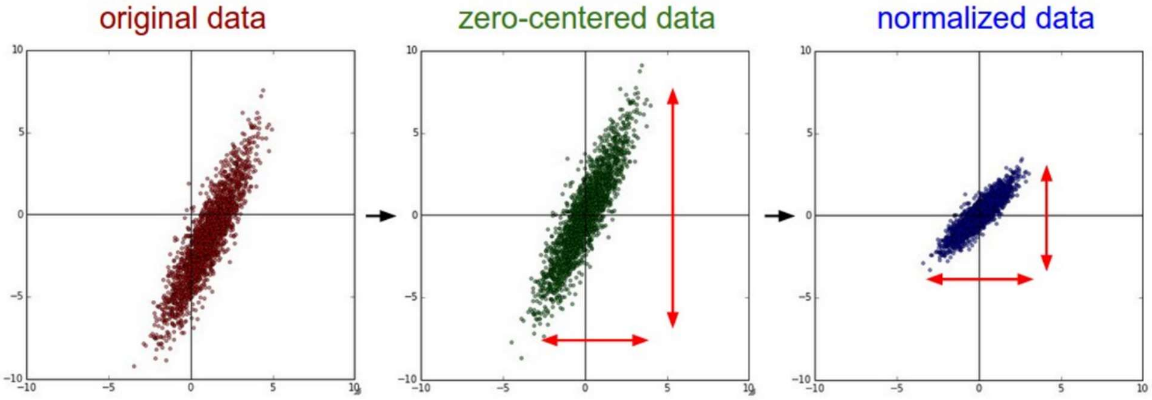


Figure 5: Concept of data standardization

2.11. Computation of the Saastamoinen model

To create a comparison for the results of the approaches investigated in thesis, a Saastmoinen model for the corresponding stations and timeframe was computed. It should however be noted that the computation of this model was not a part of this thesis and that the resulting values were provided.

The computation consists of three steps:

- Firstly a collocation of the meteorological data within a radius of about 50 km in east and north direction is performed. The collocation consists of a functional part, depending on the particular meteorological parameter, a stochastic term and an noise term:

  $$l = f(u, x, t) + s(C_{ss}, x, t) + \varepsilon$$

  For instance would the functional part for the pressure look like this:

  $$p(x, y, z, t) = (p_0 + a(x - x_0) + b(y - y_0) + c(z - z_0)) * e^{\frac{z-z_0}{H}}$$

  Where a,b,c correspond to the coefficients of horizontal and temporal gradients. H to the scale height and p_0 to the pressure at the reference point. The stochastic term is based on a covariance function, which describes the weighting of the individual sampling points to each other.

- The collocated meteorological data is then interpolated at the location of the GNSS stations.
- Finally the computation of the delay is performed with the empirical Saastamoinen formula.

  $$ZTD = 0.002277 f(\emptyset, h)[P_{tot} + (\frac{1255}{T} + 0.05)P_{wet}]$$

  Where f is a small correction term based on geographical latitude as well as the height.

  $$f(\emptyset, h) = 1 + 0.0026\cos(2\emptyset) + 0.00028\, h$$

This model has the advantage that it uses the exact same data as provided to the machine learning algorithms. Thus a fair comparison in terms of accuracy can be made. Finally should it be mentioned that this computation took several hours. Although could this still be optimized, does it also indicate a potential advantage of the proposed machine learning approaches. As especially already trained models are able to make predictions even for very large datasets in a comparatively much shorter amount of time.

For more detailed information please refer to [GPS based Determination of the Integrated and Spatially Distributed Water Vapor in the Troposphere, Troller, 2004].

3. Theory

This chapter will describe some of the theoretical aspects of the algorithms used for this thesis. In a second step each implementation with regards to the specific parameters is detailed. Both applied algorithms belong to the category of supervised learning. Supervised learning algorithms use samples of data combined with the corresponding label and try to adapt their model in such a way that it can describe the relation between the data and its label with as less error as possible.

3.1. Random forest

Random forest is a capable machine learning algorithm based on principal of combining multiple instances of another approach, the decision tree.

This theory part is mostly based on the original paper [Ho, Random Decision Forest, 1995] as well as on the corresponding chapters from [The Elements of Statistical Learning, 2nd edition, 2017] which is freely available online.

3.1.1. Concept of a decision tree

Decision trees are a relatively simply machine learning algorithm that can be used for both regression and classification and allow for highly nonlinear solutions. Built for a regression problem it is simply called regression tree. Figure 6 shows a very simple example of how a regression tree might look representing the given dataset.

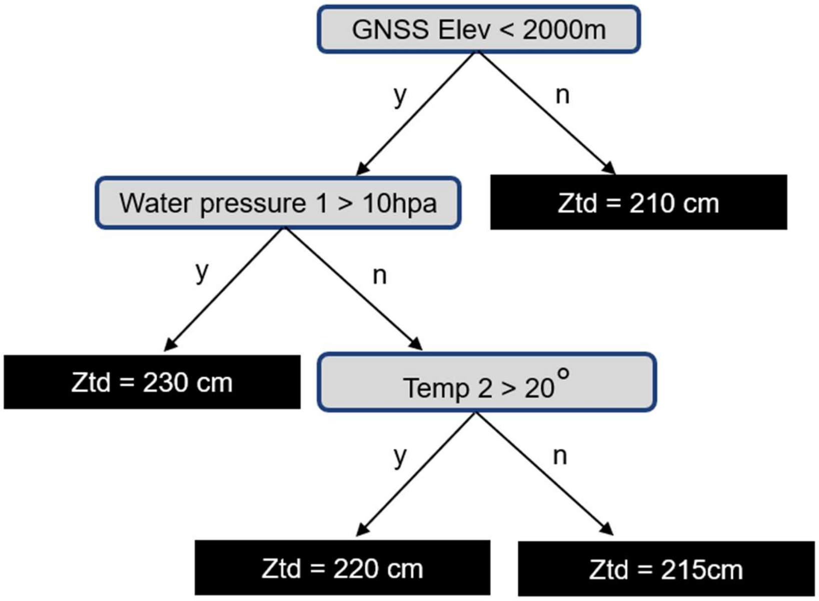


Figure 6: Simple example of a regression tree

Once the tree is built from the training dataset, the delay for another sample can be predicted by simply going through the tree and comparing the value at each node with the value of the feature in the sample. As visible in figure 6 have decision trees the advantage of being much more easy to interpret compared to other machine learning algorithms – especially neural networks.

A regression tree can be built in the following way:

- Find the feature and its specific value for the root node that separates samples the best way. In figure 6 this would correspond to the elevation of the GNNS station and the specific value of 2000m.
  To do so the best possible value for every feature is firstly computed:
  - The values of this particular feature of all samples are ordered numerically and the average of each two neighbouring values is built. This list of averages then contains the candidates for the numerical values for the split. For each candidate all samples are then split accordingly into two groups and the mean of the labels for each group is computed. These two means are then compared to the actual labels of the two groups by calculating an indicator for the size of the resulting error. Such indicators could include the mean squared error (mse) or mean absolute error (mae). The candidate value that causes the smallest error value is then chosen.
- The resulting lowest error indicator for every feature are then compared to each other and the best feature with the corresponding value are then used for the root node.
- At every new node it is now checked if this particular branch of the tree is finished or if it should be split up even further. There are different criteria to achieve this. A very common one is to just set a minimum number of samples that a node should contain – e.g. 20. If a split would cause nodes with a smaller sample size, it is not performed and this branch of the tree is completed. If it indeed should be split up even further it can be done the same way as described above – find the feature that splits up the samples in the node the best way and determine the specific value of the split with regard to a specific resulting error indicator (e.g. mean squared error).
- Once no node can be split up further, the tree is finished. The prediction of the tree now again corresponds to the average of the labels of the samples that ended up in a particular node.

3.1.2 From decision tree to random forest

The biggest problem with singular decision trees is that they tend to overfit rather highly: they adapt very strongly to the given dataset and fail to generalize on a weakly correlated dataset. Additionally is the capability of the underlying model not that sophisticated. They can fail to extract more complex relations in the dataset and tend to be rather noisy. To solve this problem [Ho, Random Decision Forest, 1995] introduced a concept called bagging. Bagging uses a multitude of individual decision trees and combines them to a random forest. This concept applies to both classification and regression trees in the same way. The general process consists of the following steps:

- Bootstrapping

  As mentioned in the previous subchapter is an individual regression tree created with regards to usually all of the training data. For the trees in a random forest however this is done slightly different. For every tree a bootstrapped dataset is generated. To do so a certain amount of samples are randomly picked from the complete training dataset. It is important to note that samples may be chosen repeatedly. This results in a different dataset for every trained tree.

  - Building decision trees

  The trees are built almost the same as already described with only a small difference: As opposed to taking all features into consideration when splitting the tree, a randomly chosen subset of the features is generated and considered for the split. Note that at every step of the tree this subset is regenerated so that each split is based on a slightly different selection of features. This further increases the difference between the trees.

  - Aggregating

  To make a prediction with the forest, every decision tree makes an individual prediction. By calculating the mean (mode respectively for a classification problem) the complete model comes to its conclusion. This process is also called aggregating.

As can be seen is the name bagging also a composite of bootstrapping and aggregating. By applying these two steps, the resulting model does overfit much less on the training data and is able to capture more complex relations. This is mainly caused by the fact that in theory averaging the large individual errors of mostly uncorrelated trees should deliver a zero mean.

For more details on the creation of decision trees, their combination to random forest as well as the underlying statistical assumptions please refer to the stated textbook.

3.1.3. Implementation

The random forest used in the course of this thesis consists of 300 individual trees. Adding additional trees did not seem to have any improvement on the accuracy of the result, while simultaneously increasing the computational time noticeably. Using significantly less trees did let the quality of the predictions decrease however. To determine the suitability of a feature or threshold to split up the samples, the mean squared error was used.

As opposed to the neural network approach, are not all training samples used to train the random forest. 500’000 samples are chosen randomly from the training dataset. Although is more data generally favourable for every machine learning algorithm, is the random forest algorithm not that dependent on a very large number of samples (especially compared to neural networks). Increasing the number of training samples did not seem to improve the accuracy of the prediction at some point, but did significantly increase the computation time of the training.

The model used during this thesis was implemented with the Python library Scikit-klearn [https://scikit-learn.org/, 2020].

3.2. Neural network

In the past decade artificial neural networks (ANN) have become increasingly popular in the field of machine learning. Especially for problems where a large dataset is available they are applied widely in some form or another. This breakthrough was largely made possible due to much more widely spread powerful hardware such as graphics processing units (GPU) as they allow for highly parallelised computing.

The theoretical part of this subchapter is again based on [The Elements of Statistical Learning, 2nd edition,2017] as well as on the Stanford course [CS231n: Convolutional Neural Networks for Visual Recognition, 2019]. Most of the shown figures also originate from this course if not mentioned otherwise.

3.2.1. Concept of a neural network

The idea of an ANN is loosely based on the structure of biological neural networks such as found in the human brain. Individual neurons receive an input signal and transform it into a different output signal. Such a neuron can be seen in figure 7. By combining a multitude of different individual neurons, a network is generated that can transform a certain input into a vastly different output. ANN also apply this concept in a simplified manner.

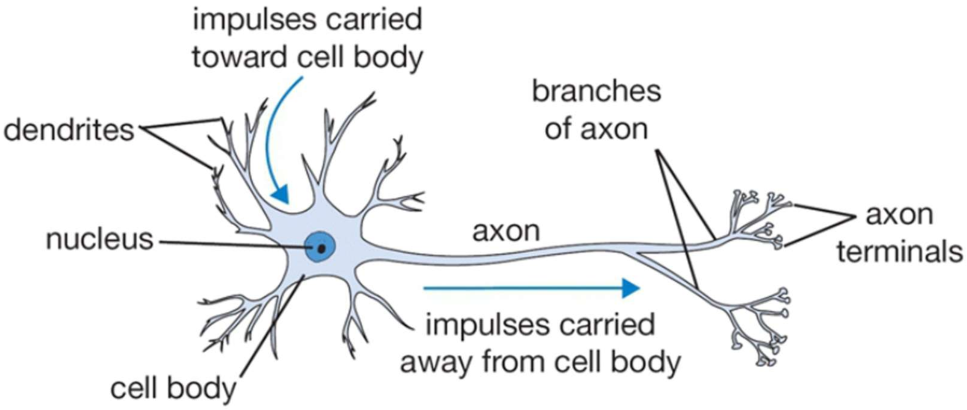

impulses carried toward cell body
dendrites
branches of axon
nucleus
axon
axon terminals
impulses carried away from cell body
cell body

Figure 7: An individual biological neuron

The most simple and general applicable conversion of the mentioned principle is called a fully-connected neural network. Such an architecture is also used as one of the two algorithms in this thesis. These networks consist of an input layer, a certain amount of hidden layers and an output layer, where each hidden layer contains a certain amount of neurons. The size of the output and input layer corresponds to the amount of features of a sample of the data as well as the dimension of the output the network is supposed to predict. As can be seen in chapter 2.7. does the number of features amount to 27.

Each neuron of a layer is then connected with all of the neurons of the previous layer, hence the name fully connected. Figure 8 shows this principle for two hidden layers. Often are many of such layers combined and so create a deep network. This is where the frequently used term “deep learning” originates.

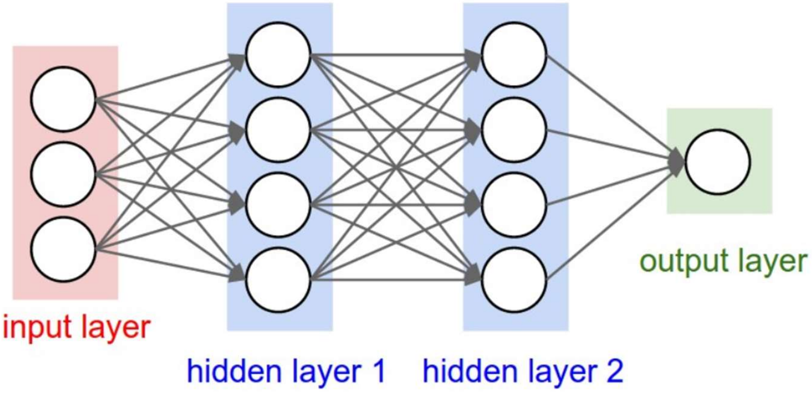


Figure 8: A fully connected neural network

The value of an individual neuron can be calculated in the following way:

$$neuron = f(\sum_{i=1}^{n} input_i * weigth_i + bias)$$

Where n corresponds to the amount of neurons in the previous layer. As can be seen does each connection of the network get a certain weight. An additional bias for this particular neuron is then added. These weights and biases make up all the parameters of a fully connected neural network.

Since neural networks are used for complicated classification or regression tasks, a set of linear equations would often prove not to be powerful enough to capture complex relations between dataset and label. This is why the activation function f is introduced. That function establishes the nonlinearity of the network and allows for much more accurate predictions.

The most common used activation function is called rectified linear unit (ReLU):

$$f(x) = \max{(0, x)}$$

It basically just suppresses neurons with negative values. Figure 9 shows ReLU.

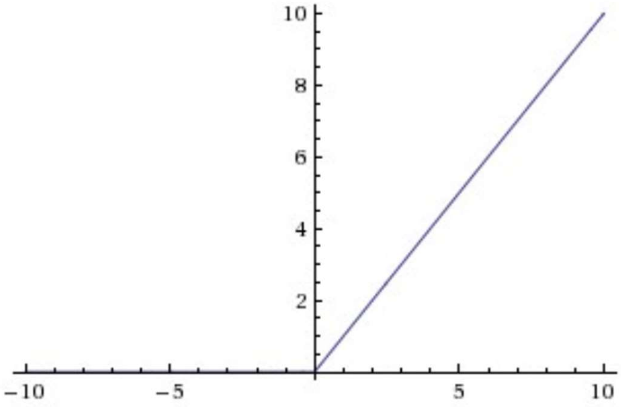


Figure 9: ReLU activation function

3.2.2 Training of the network

Once the architecture or model of the network is set up, the training phase can start. The goal of the training phase is to iteratively feed in samples of the training data and adapt all parameters of the network in such a way that it can make accurate predictions on data samples it has not seen before. To do so the prediction of the network – i.e. the neuron in the output layer – is expressed as a function of the values in the previous layer. These values are then again expressed as a function of the neurons of the previous layer. If this is continued until the input layer, the result is a function describing the desired prediction of the network as a function of all the features of a sample.

$$pred = F(feature_1, \dots \dots feature_n)$$

By comparing this prediction with the label of that particular sample, a loss function of the network can be created. In the case of regression problems, the used loss often corresponds to the mean squared error.

$$mse = \frac{1}{n}\sum_{i=1}^{n}(pred_i - label_i)^2$$

The unknown parameters of the loss function are exactly described by the weights and biases of the network mentioned in subchapter 3.2.1. Thus is the loss a function in n-dimensional space where n corresponds to the amount of parameters of the network. By now searching for local minima of this function, a set of parameters can be determined that describe the relation between samples and prediction with a minimal loss.

This search of a local minimum is performed with an approach called stochastic gradient descent. Stochastic gradient descent uses a small subset of all available samples – also known as batch size – to determine the direction of largest gradient values of the loss function. Once all the samples of the dataset are used, an epoch of the training is finished. Usually a training period consists of a multitude of epochs. Figure 10 shows an example of a gradient descent around a local minimum.

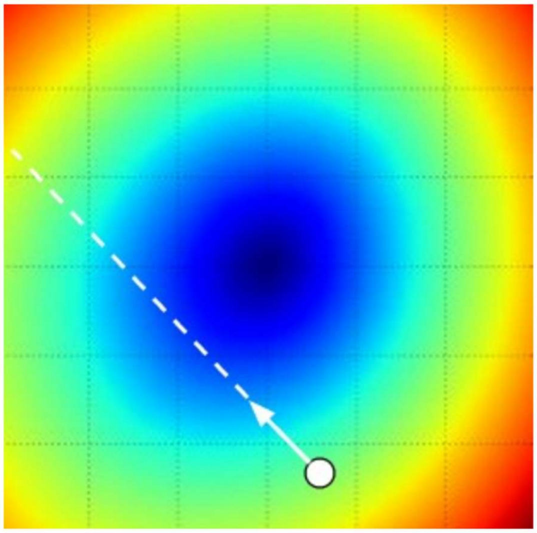


Figure 10: Visualisation of a gradient descent

The size of the step into that direction is called the learning rate and is of vital importance. If it is chosen too large local minima might be missed repeatedly and the network would not find an optimal solution during the training. If it is too small the descent might get stuck or might just not reach an a minimum during the whole training time. This whole problem is illustrated in Figure 11.

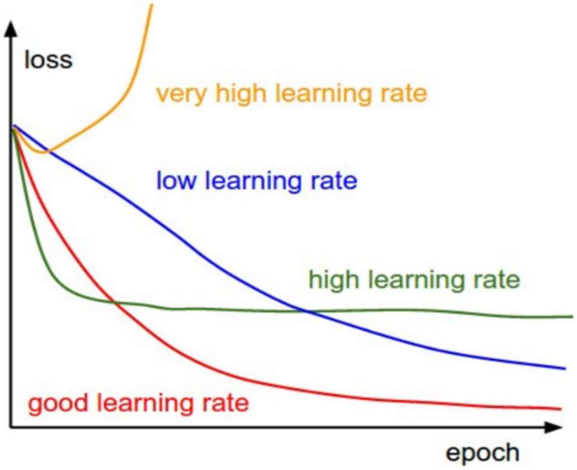


Figure 11: Different learning rates

Learning rate, batch size and number of epochs belong to the what are known as hyperparameter of the network. The optimization of the hyperparameter is often done – as also in the course of this thesis – by manual search.

For further theoretical details and the mathematical background please refer to the sources stated at the beginning of this subchapter.

3.2.3. Implementation

The architecture of the chosen network consists of four hidden layers and can in detail be seen in table 2.

Layer	Number of neurons	Parameters
Input layer	27	-
1. Hidden layer	512	14’336
2. Hidden layer	128	65’664
3. Hidden layer	128	16’512
4. Hidden layer	128	16’512
5. Hidden layer	128	16’512
Output layer	1	129
Total	-	129’665

Table 2: Architecture of used network

Table 3 summarizes the selection of the three hyperparameters mentioned in the previous subchapter.

Hyperparameter	Value
Learning rate	0.0001
Batch size	1000
Epochs	250

Table 3: Hyperparameters

Many different combination of learning rate and batch size have been tried manually but this one seemed to do the best – although only by a very slight margin – for the given particular problem. The amount of epochs could in principal have been chosen significantly lower. Both training and validation loss decreased very rapidly during the first two epochs and then only improved very slightly about once every tenth epoch. This can also be seen in Fig b when for example looking at the mean absolute error.

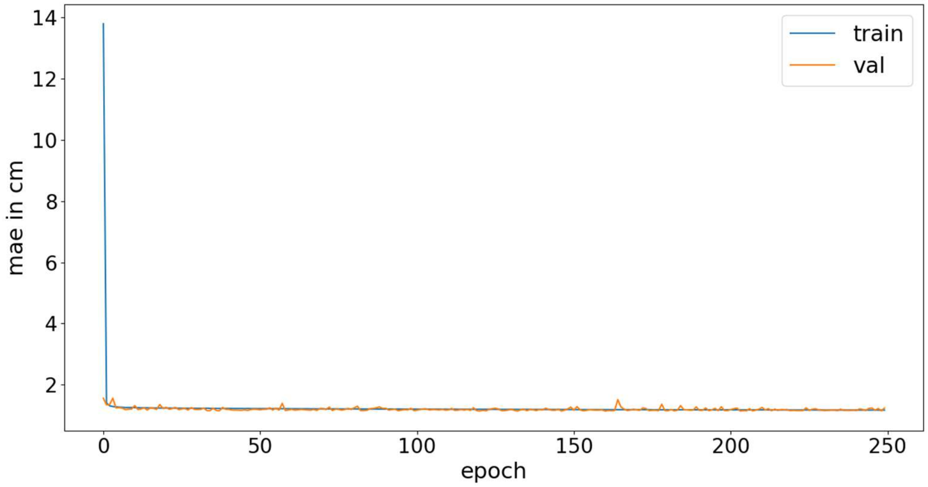


Figure 12: Mean absolute error during training

The described network was implemented with the Python library Keras [https://keras.io/, 2020].

4. Results

4.1. Comparison between neural network and random forest results

Once the random forest as well as neural network models are trained with the dataset described in chapter 2.7. they can predict the zenith path delay for the year 2018 based on the corresponding tropospheric parameters. The residuals for their predictions for all stations combined can be seen in figures 13 and 14 respectively.

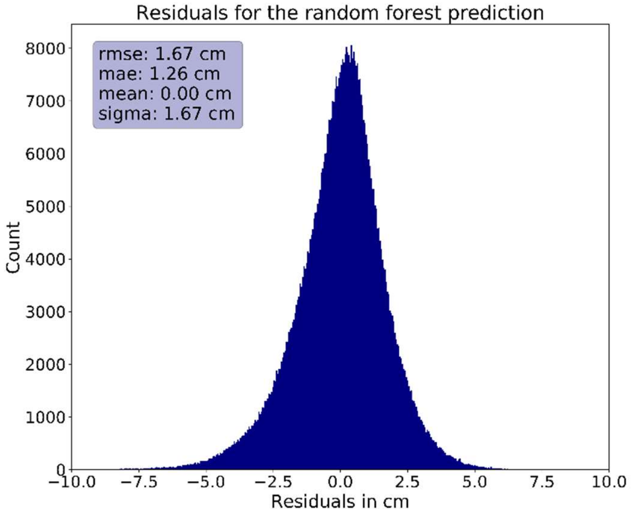


Figure 13: Residuals for the random forest prediction

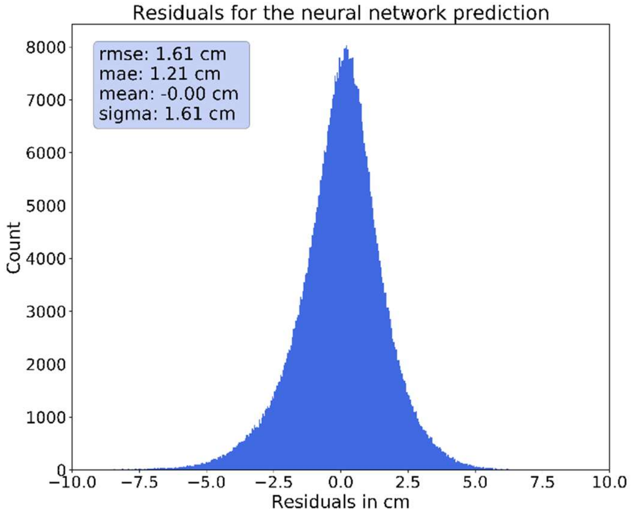


Figure 14: Residuals for the neural network prediction

As can be seen is the achieved root mean squared error as well as the mean absolute error at a pretty even level. Although are the predictions from the neural network slightly more accurate, is the difference smaller than one millimeter in both error statistics. Fortunately though is the achieved rmse in both cases well below the minimum goal of 2 cm. Additionally is it also interesting to note that neither algorithm tends to systematically over- or underestimate the delay. Both residual distributions are centered at almost exactly zero with a mean value of less than a tenth of a millimeter. Because of this distribution the standard deviation of the residuals corresponds highly to the determined root mean squared error.

Since both approaches seem to achieve very similar results and random forest was mainly used to establish a baseline only the neural network predictions are analyzed in detail.

4.2. Correlation between station height and error

Because of the individual circumstances of each GNSS station it is to be expected that the achieved accuracy would vary quite a lot depending on the station. The difference in station height as well as the distribution of available meteorological stations are after all quite vast. This can be seen in figure 15 where the root mean squared error, mean relative error (mre) and station height are displayed for each individual station.

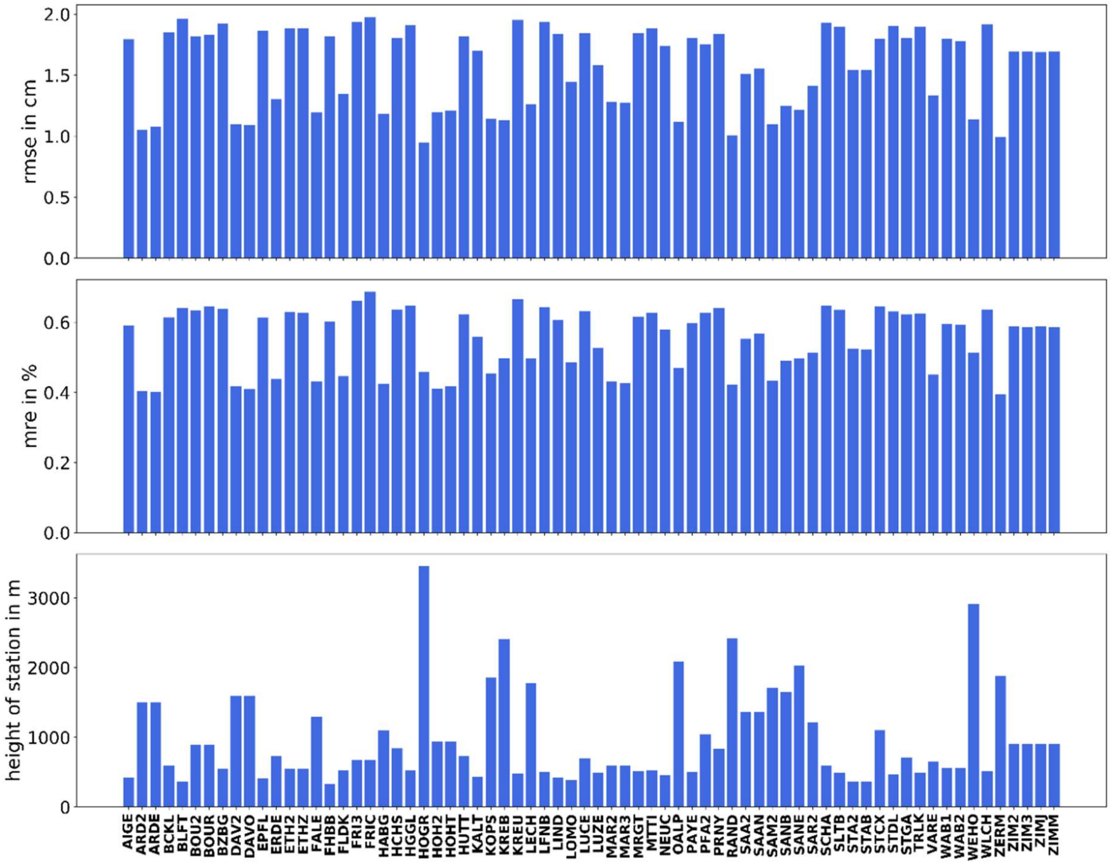


Figure 15: Root mean squared error, mean relative error and height for every station

This figure shows quite a range of individual errors, where the highest (Station: FRIC, rmse: 1.98 cm) is more than double the size of the lowest (Station: HOGR, rmse: 0.95 cm). It is important to note however that no station has an rmse higher than the minimum goal of 2 cm. Also is the average residual for no station higher than 0.69 % (Station: FRIC) of the respective delays.

These results also seem to indicate that the predictions for stations above a certain elevation generally result in a much lower error. This assumption seems to be confirmed when looking at the relation between station height and rmse.

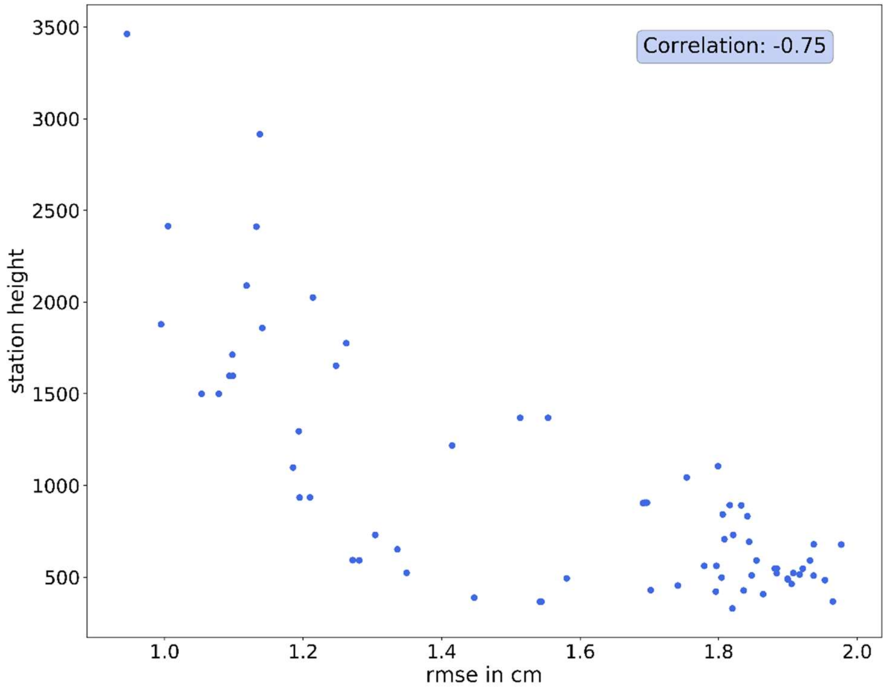


Figure 16: Relation between rmse and station height

The correlation is quite high at negative 0.75. As already assumed is the error especially for very high stations considerably lower. This was somewhat to be expected as the higher the stations are, the smaller the influence of the wet part of the troposphere is. And since the wet delay is much more difficult to predict, the smaller it is, the generally lower the error of the prediction is. Below a station height of 1000 m however the elevation does not seem to have as much of an impact anymore.

As figure 17 shows is the mre – with a correlation of negative 0.54 - less influenced by the station height.

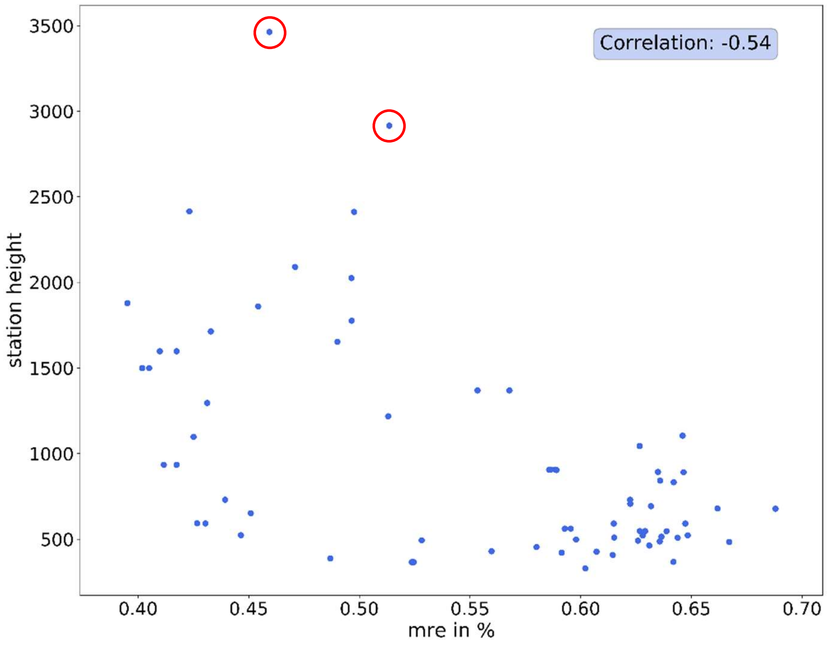


Figure 17: Relation between mre and station height

Especially for the highest stations of the network can a clearly lower correlation be identified (marked in red). Compared to the scatter plot of the rmse these stations are shifted considerably to the right. This is again not that surprising. The elevation of the station correlates highly to the values of the delay as also mentioned in chapter 2.1. It now seems that the influence the elevation has on decreasing the overall delay is larger than one it has on decreasing the error of the prediction. Therefore the mre increases compared to the rmse in relation to other stations.

The distribution of the stations below the 1000 m elevation line does however look very similar when compared to figure 16. This seems to suggest that below a certain threshold the weight of the elevation becomes much less significant compared to other influences. Such influences could include the distances to the nearest meteorological stations or the particular distribution of these stations around the respective GNSS station.

4.3. Feature importances

As already mentioned in chapter 3.1.1. do decision trees, and therefore also random forests to some extent, have the advantage of being more interpretable than other approaches. One way to do so is to look at the importance of each feature during the creation of the forest. These values are in the range of 0 to 1 and add up to 1. They indicate the importance of each feature for splitting up the individual trees. The following table shows the 10 most important features as well as their weight.

Elev. GNSS	**Elev. Meteo 1**	**Press. Meteo 1**	**East Meteo 1**	**Elev. Meteo 3**	**Elev. Meteo 4**	**Press. Meteo 3**	**East Meteo 2**	**East GNSS**	**North Meteo 3**
0.30	0.10	0.08	0.06	0.05	0.04	0.04	0.03	0.03	0.03

Table 4: Feature Importances

As can be seen are the elevations – especially the ones of the GNNS stations – of great importance to divide the samples of the dataset. This corresponds nicely to the findings of the last subchapter and should in general confirm the significance of the station height for both models. Additionally are in general a lot of coordinate features in this list. This could suggest that the random forest model builds different tree branches for the individual stations.

4.4. Comparison with Saastamoinen

To further assess the achieved accuracy of the machine learning predictions, it is now compared to the already described Saastamoinen model. These computations incorporate the exact same timeframe and can therefore serve as a direct comparison to a more common approach. Table 5 shows several properties of the residuals of all three methods.

	Random forest	Neural network	Saastamoinen
root mean squared error [cm]	1.67	1.61	2.09
mean absolute error [cm]	1.26	1.21	1.59
mean relative error [%]	0.58	0.55	0.74
amount of residuals over 2 cm [%]	20.01	18.57	30.06
mean of residuals [cm]	0.00	0.00	0.34

Table 5: Comparison between random forest, neural network and Saastamoinen

As can be seen result both random forest and neural network predictions in a generally lower error than the Saastamoinen approach. For rmse and mae the resulting error is roughly 4 mm lower, which corresponds to a reduction of the error of about 19 %. The improvement of the mre is with a relative reduction of about 27 % - ca. 0.2 % absolute – even higher. Additionally is the amount of residuals that lie above the threshold of 2 cm considerably lower for the machine learning approaches. This is especially the case for the neural network predictions, where the amount of residuals above that threshold – in relative terms – is almost 40 % lower compared to Saastamoinen.

Interestingly enough seems there also to be a bias in the estimation of the delay produced by the Saastamoinen approach. As already mentioned is the mean of the residuals for random forest and neural network almost equal to zero. The Saastamoinen mean however amounts to over 3 mm. This indicates that this approach usually tends to overpredict the delay while the machine learning algorithms have no bias for the complete set of stations. Since the computation of the Saastamoinen delay however was not a part of this thesis and serves only as a comparison, this bias is not further investigated here.

To analyze the result with respect to separate times of the year does figure 18 show a monthly comparison of the rmse for all three approaches. Generally can it be said that – regardless of the approach – the error is much lower during the winter and spring months while the highest errors occur during August, September and October. This is likely again caused by the increased wet delay during that time period. As already mentioned is the wet delay much harder to exactly determine compared to the dry part. Thus would increased wet delay also increase the size of the error. It seems however though that both machine learning predictions deal better with this problem than Saastamoinen as the difference in rmse between them is also larger during the summer and autumn months as opposed to the winter period.

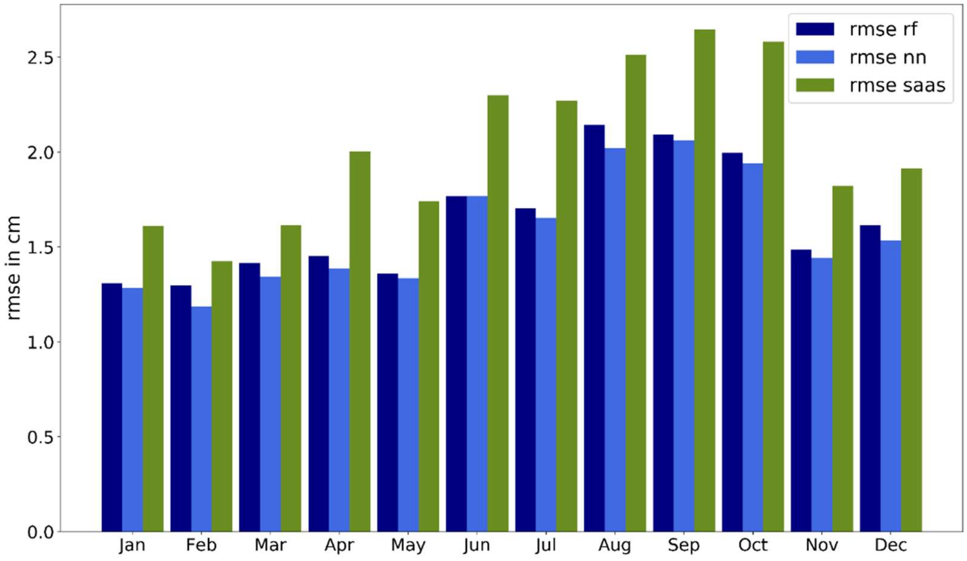


Figure 18: Monthly comparison of rmse

The statistical values for all individual stations can be found in appendix B,C and D.

4.5. Comparison of individual stations

To further analyze the properties of the predictions of the neural network a few stations are investigated more closely here and also compared to the Saastamoinen result.

4.5.1 WEHO station

This station is the second highest station of the network at 2'916 m elevation. It is located in the canton of Bern right next to border to Valais. Figure 19 shows the monthly comparison of rmse for this station.

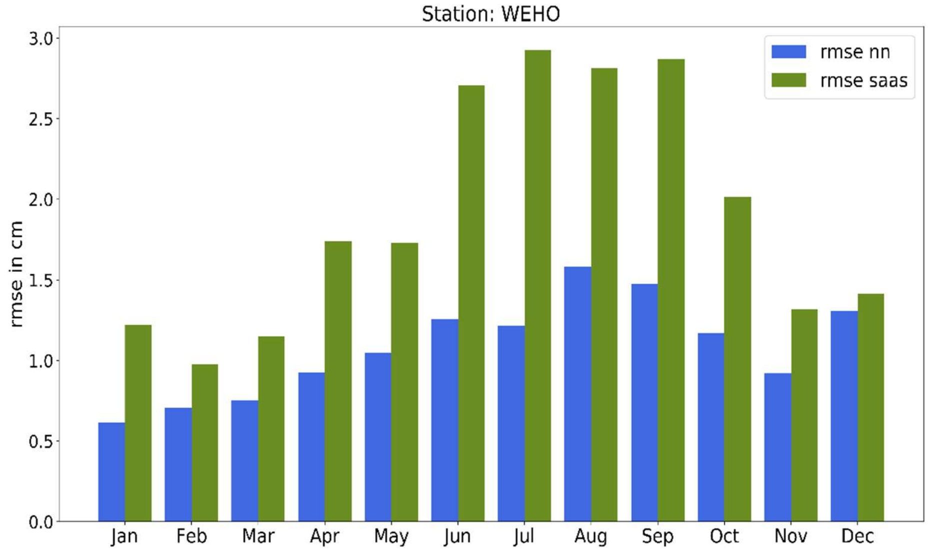


Figure 19: Monthly rmse for WEHO

As can be seen does the monthly distribution of the rmse show very similar properties when compared to the error for all stations. Again is the error generally much lower during the winter. Especially noticeable is the vast difference during the summer months where in some cases the rmse of the neural network predictions is less than half than its Saastamoinen counterpart. This is also visible when looking at the time series of the predictions and their corresponding residuals.

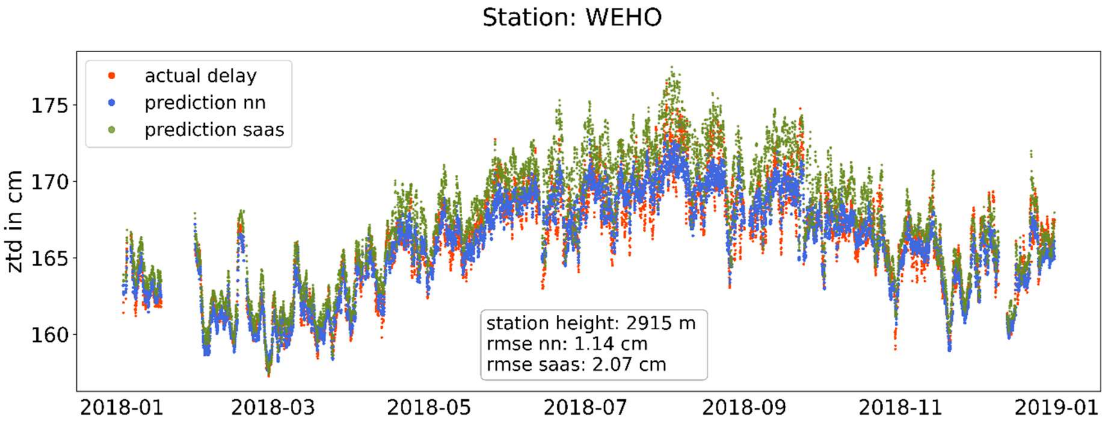


Figure 20: Time series of predictions for WEHO

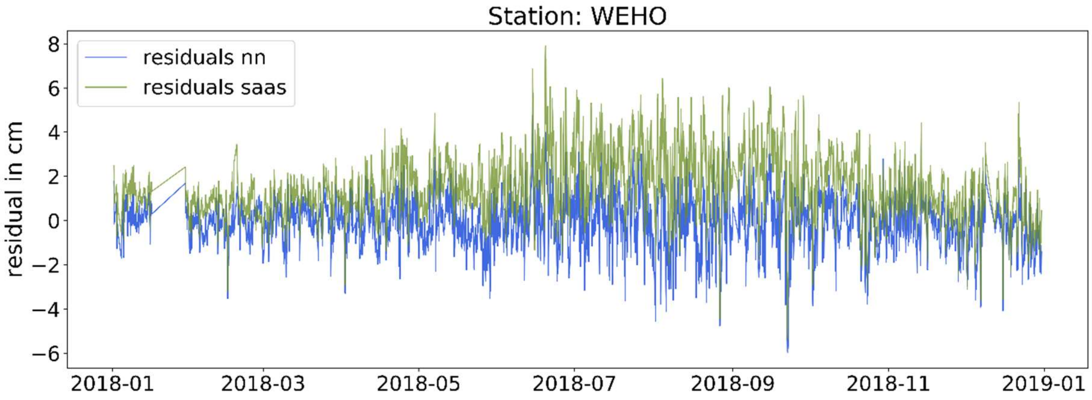


Figure 21: Residuals of predictions for WEHO

As already mentioned does the Saastamoinen model tend to overpredict the delay. This is especially visible here during the summer months as can be seen in the residuals. The neural network on the other hand seems to fail to capture the extreme spikes of the delay and tends to smooth its predictions.

4.5.2. BLFT station

To analyze how the neural network deals with unidirectional distributed meteorological stations, this station is shown here. Its location in France near the swiss border as well as the distribution of available meteorological stations can be seen in Figure 22.

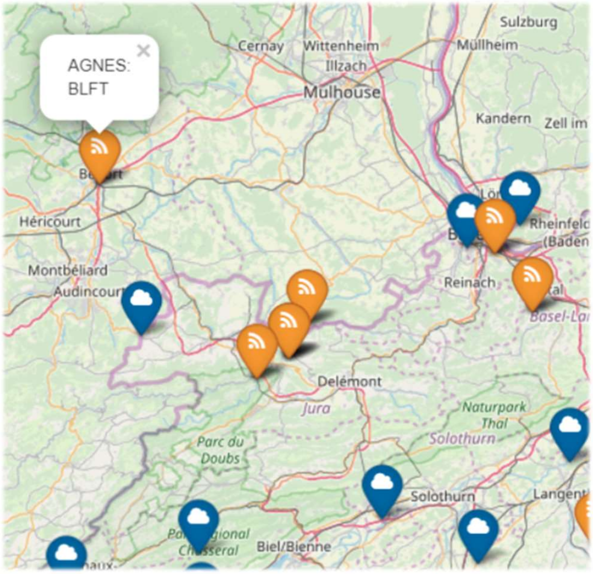


Figure 22: Location of the BLFT station

Since this station lies at an elevation of 368 m, should its station height not have any significant influence on the quality of the resulting predictions. Figure 23 shows again the monthly rmse.

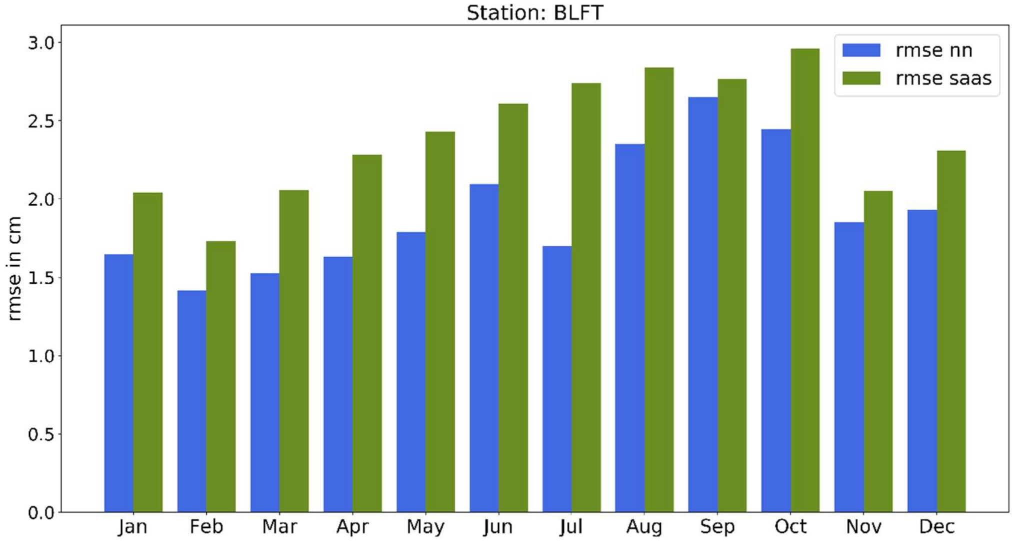


Figure 23: Monthly rmse for BLFT

Both approaches seem to cope well with the poor distribution of meteorological stations. It is noticeable that during the summer months the neural network predictions are not as vastly more accurate compared to Saastamoinen as before. The error range of the Saastamoinen predictions resides at pretty much the same level as before with the highest rmse slightly below 3 cm, while the neural network predictions are considerably worse compared to the previous highlighted station. It is however possible that this increment of the rmse is caused by the lower elevation of the station and not by the poorly distributed meteorological stations.

4.5.3. ERDE station

Finally the ERDE station is highlighted. With an elevation of 730 m, is this station neither at an extreme height nor does it rely on poorly distributed meteorological station as can be seen in Figure 24.

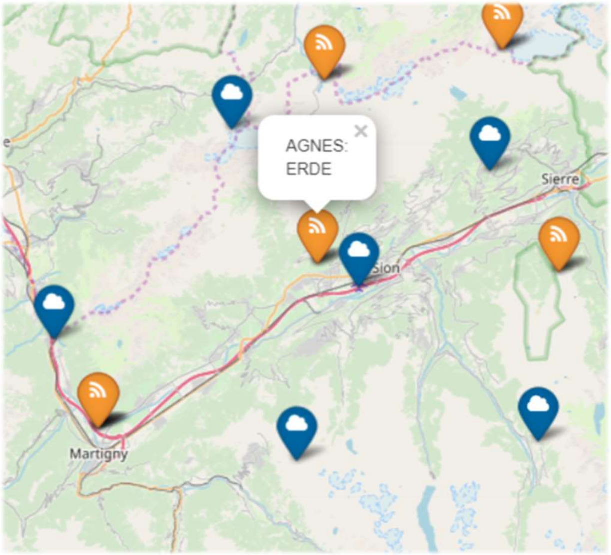


Figure 24: Location of the ERDE station

When looking at the monthly rmse, a similar pattern as before emerge.

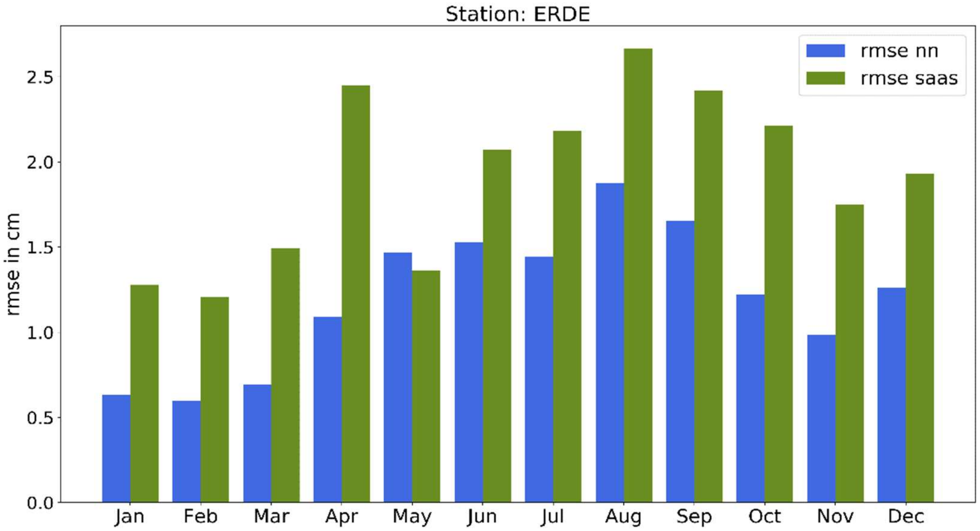


Figure 25: Monthly rmse for ERDE

Especially at the beginning of the year is the rmse of the neural network predictions extremely low. But even during the summer period does it never exceed the level of 2 cm. The improvement compared to Saastamoinen – with the exception of May – is again quite large. This large improvement is although distributed quite well over the year – unlike in other cases – and not concentrated around the summer period.

When looking at the time series and residuals, the trend of the Saastamoinen model to overpredict can again be identified.

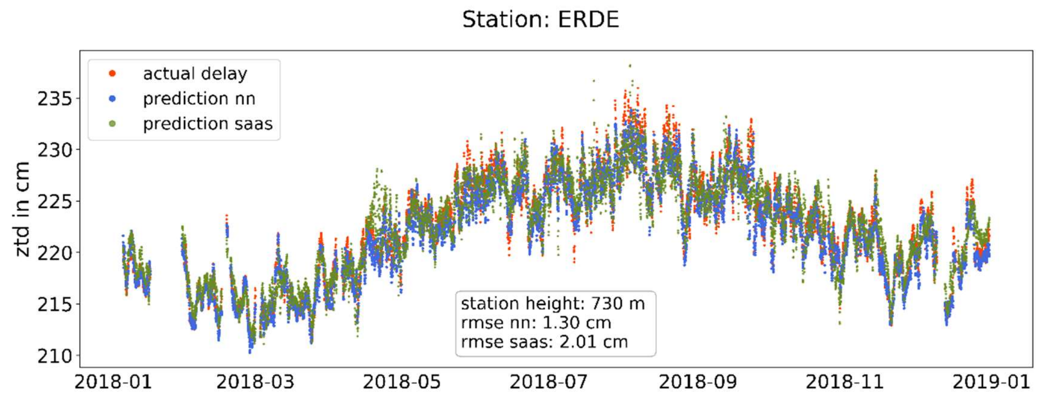


Figure 26: Time series of predictions for ERDE

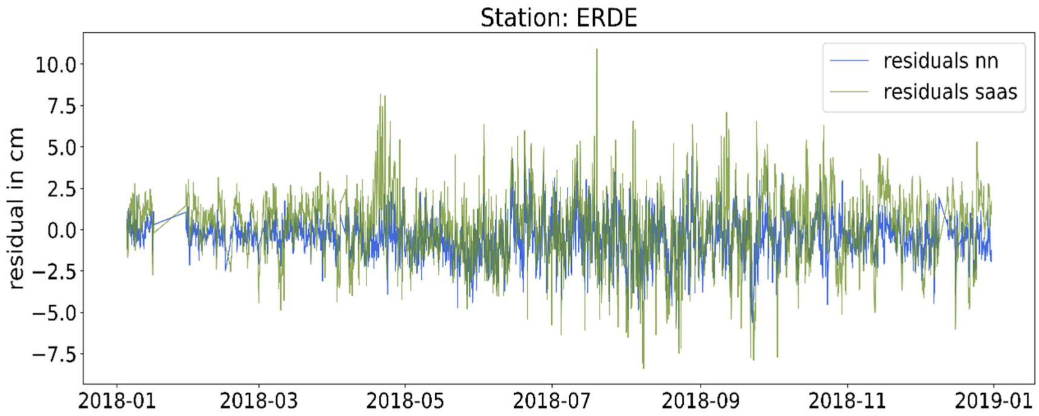


Figure 27: Residuals of predictions for ERDE

So it seems that the improvement of the neural network predictions over the Saastamoinen model predictions increases with higher elevation of the station as well as a good distribution of corresponding meteorological stations. These results are however more of an indicatory nature and further tests would need to be performed to more accurately assess the correlation between the distribution of meteorological stations and the resulting accuracy of the predictions. Especially the height difference between GNSS station and the surrounding meteorological stations should be analyzed further.

4.6. Spatial decorrelation of training and test data

So far the networks ability to predict the zenith path delay has only been tested on locations it has already seen during the training phase. This corresponds to only a temporal decorrelation between the training and test dataset. To analyze the ability of the network to predict the delay for unknown locations, three stations have been excluded from the training dataset. These three stations include:

- ARD2 station
    - Height: 1499 m
    - Nearest GNSS station: 0.0 km
- BZBG station
    - Height: 547 m
    - Nearest GNSS station: 4.8 km
- HGGL station
    - Height: 522 m
    - Nearest GNSS station: 14.5 km

Each of those stations lies at a different distance to its respective nearest GNSS station neighbour. The first station – as a twin station – has its nearest neighbour right next to each and the other two at a range of 4.8 km and 14.5 km respectively. It is expected that the shorter this distance is, the less problems the network should have to accurately predict the delay at that location as it has seen very similar locations during the training phase. Additionally are no stations at extreme heights excluded.

4.6.1. Excluded station: ARD2

When looking at the time series as well as the residuals of the predictions for the first station it can be seen that it is not really distinguishable from the case where this station was included during the training. This is also reflected in the total rmse which changed from 1.05 cm to 1.10 cm. A difference so small that is much more likely to be caused by slight variations of the solutions that the two different networks found than by the fact that this station was not included during the training of the second network. These results are not that surprising as was the network trained on a station in the immediate vicinity.

Figures 28 and 29 show the respective time series of predictions and residuals.

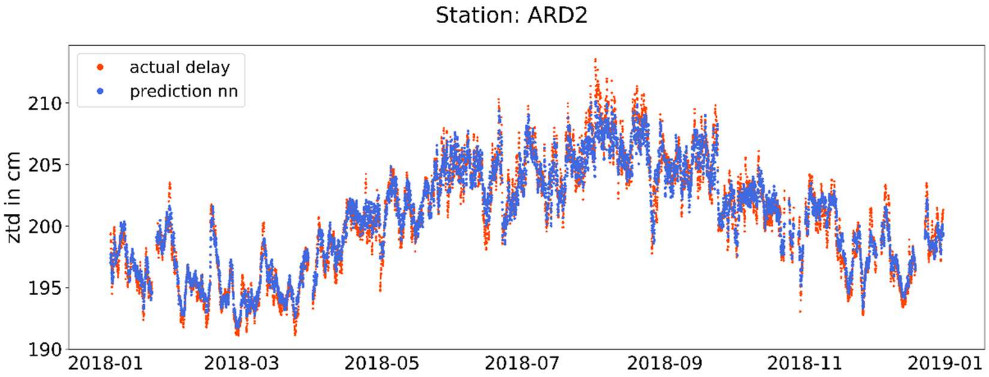


Figure 28: Time series of predictions for excluded station ARD2

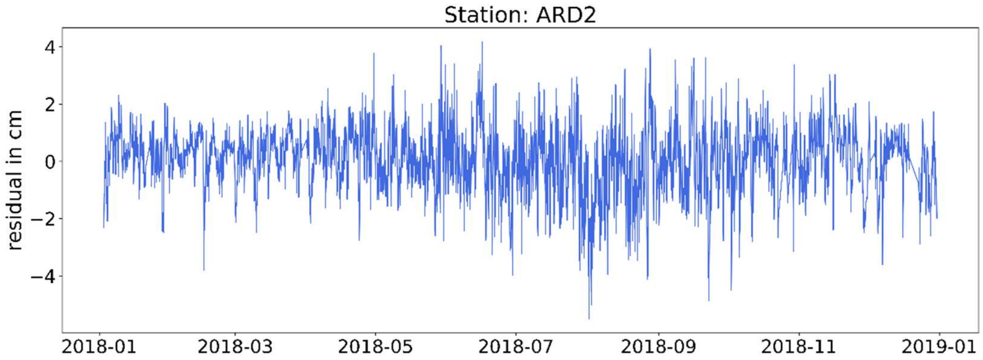


Figure 29: Residuals for excluded station ARD2

4.6.2. Excluded station: BZBG

The time series for the second not included station show a different picture however. While the shape of the curve of the predictions seems to reflect the curve of the actual delay quite accurately, the network also introduces a clear bias in its prediction. The prediction seems to be shifted along the y-axis.

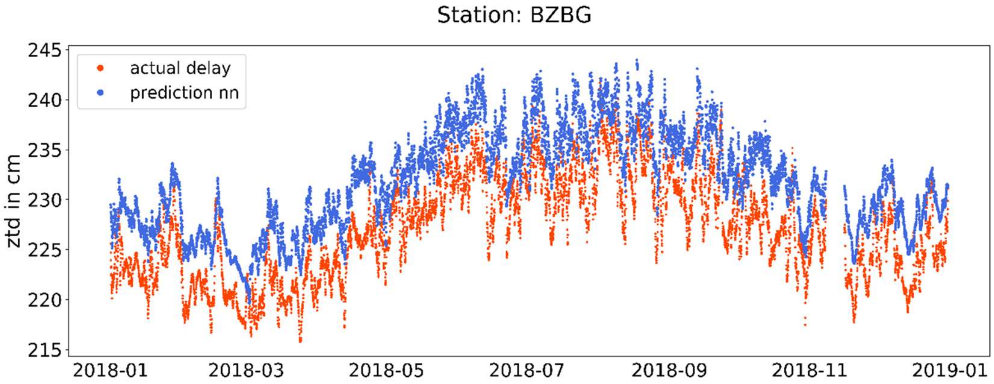


Figure 30: Time series of predictions for excluded station BZBG

This shift can also be identified in the residuals as they are now centered around about 5 cm with a mean of 4.87 cm. Otherwise they show very similar properties as before.

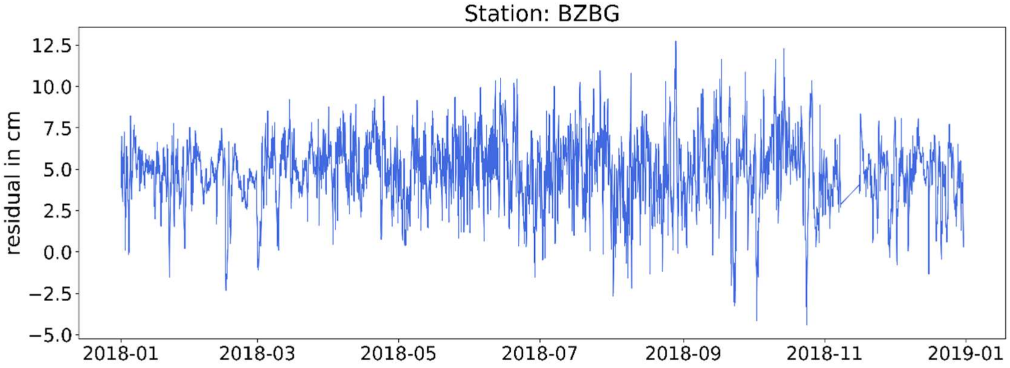


Figure 31: Residuals for excluded station BZBG

4.6.3. Excluded station: HGGL

For the third station this shift in the predictions gets even more extreme as the distance from the location of the station to its nearest neighbour station increases.

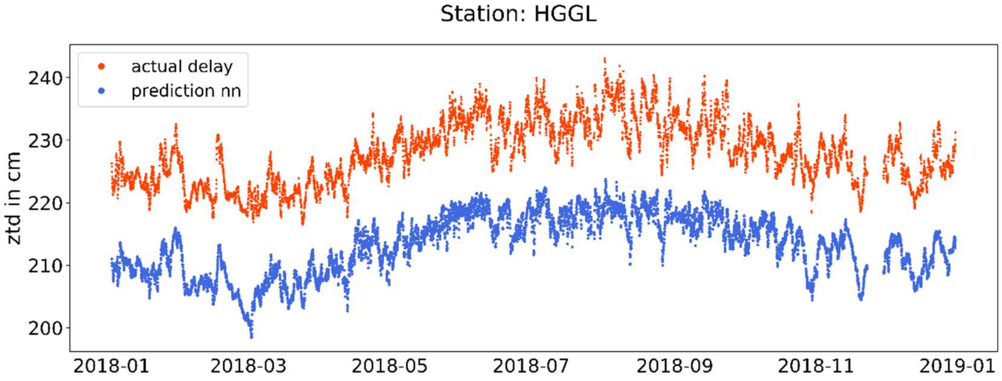


Figure 32: Time series of predictions for excluded station HGGL

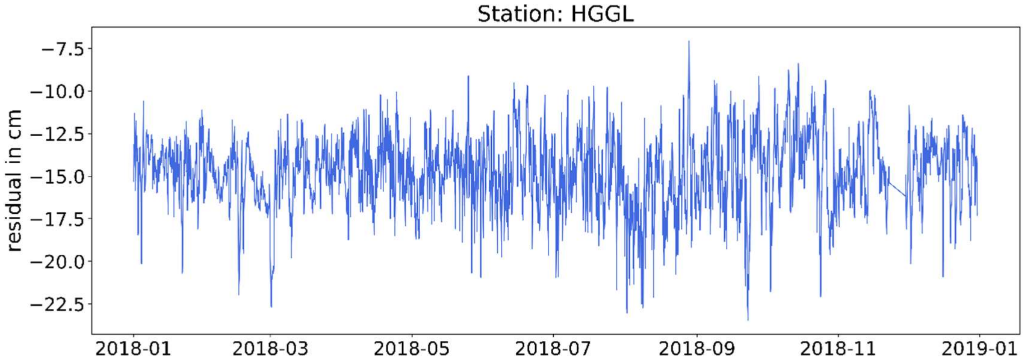


Figure 33: Residuals for excluded station HGGL

Once again seems the predicted curve of the delay to fit quite nicely to the actual delay. As can be seen in the residuals, is the curve now shifted by around 15 cm. The mean of the residuals amounts to -14.76 cm. Unlike for the last station, the prediction is now shifted in the negative direction of the y-axis. Strangely enough is the factor between distance to the nearest neighbour and absolute value of the resulting shift almost identical for both stations. For the BZBG station with a distance of 4.8 km to its nearest neighbour the mean value of the residuals consist of 4.87 cm. While in the case of the HGGL station with a distance of 14.8 km the absolute mean values amounts to 14.76 cm. If this is however actually correlated or just a coincidence is difficult to assess without performing further tests.

It generally can be said that the network can predict the shape of the delay curve somewhat accurately even for locations it has not seen during the training. Tough it does also introduce this mentioned bias, which might depend on the distance of the specific location to the nearest station that was included in the training dataset.

One simple solution for this problem would be to use much more spatially densely distributed training data. This could cause the bias to vastly decrease for any possible location as the network would have seen a large amount of reference points during the training. It is however not easy to obtain large datasets of all the needed features at that many points in space. It might be more feasible to obtain the corresponding widely distributed data for a much smaller timeframe. Especially could this be the case in the near term future with ever more increasing amounts of available data.

So it would be interesting to know if the network can accurately predict the delay for stations it has only seen a couple of times during the training. To analyze this, the same latter two stations as in the last subchapters are now not excluded completely, but just highly underrepresented in the training dataset. To do so two tests have been performed. For the first one does the training dataset include 100 completely randomly chosen samples for each of the two stations. For the second one is this sample size increased to 1’000. In the original training dataset these stations are represented with 44’652 (HGGL) and 54’210 (BZBG) respectively.

4.6.4. Underrepresented station BZBG

Figures 34 and 35 show the times series for this station with 100 samples included in the training data.

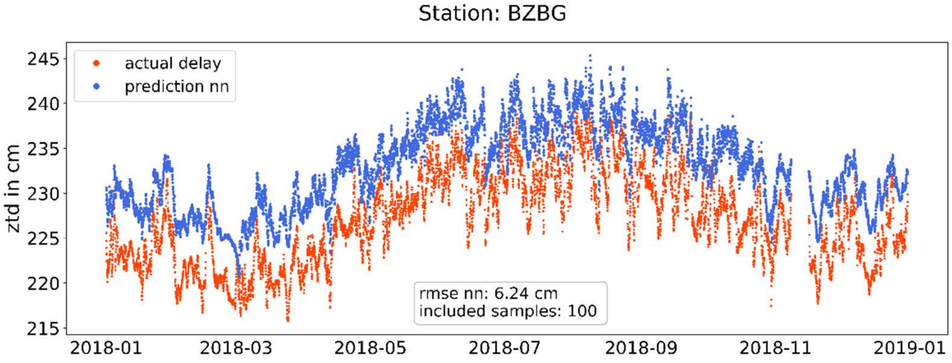


Figure 34: Time series of predictions for underrepresented station BZBG (100 samples)

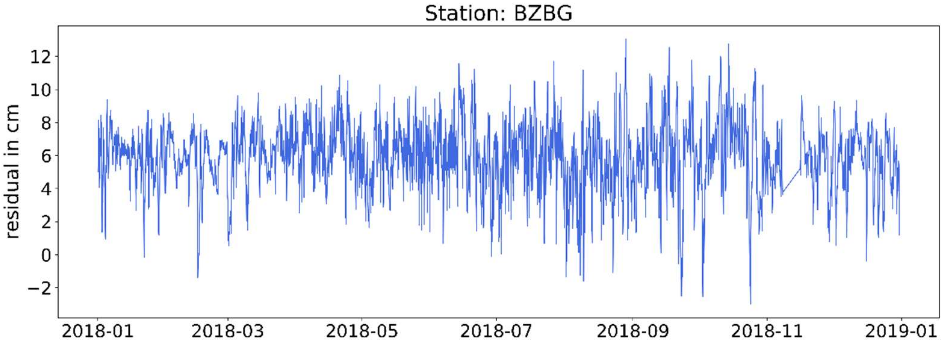


Figure 35: Residuals for underrepresented station BZBG (100 samples)

As can be seen did including the 100 samples into the training dataset not help at all. The shift of the curve did even increase as the residuals are now centered around approximately 6 cm with a mean of 5.91 cm. Including 1'000 samples however improves the result dramatically.

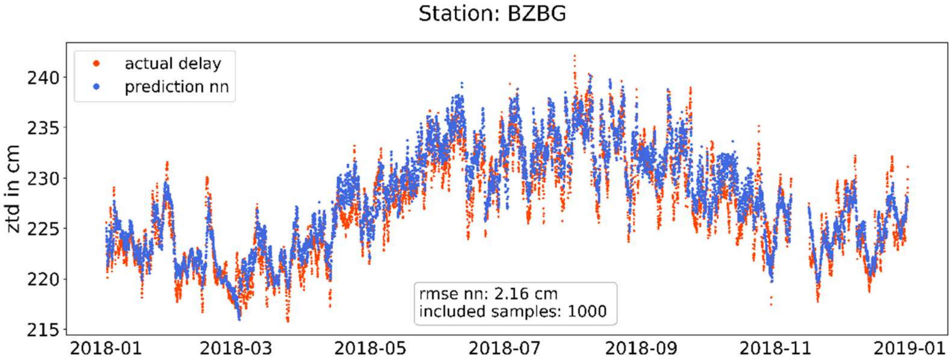


Figure 36: Time series of predictions for underrepresented station BZBG (1'000 samples)

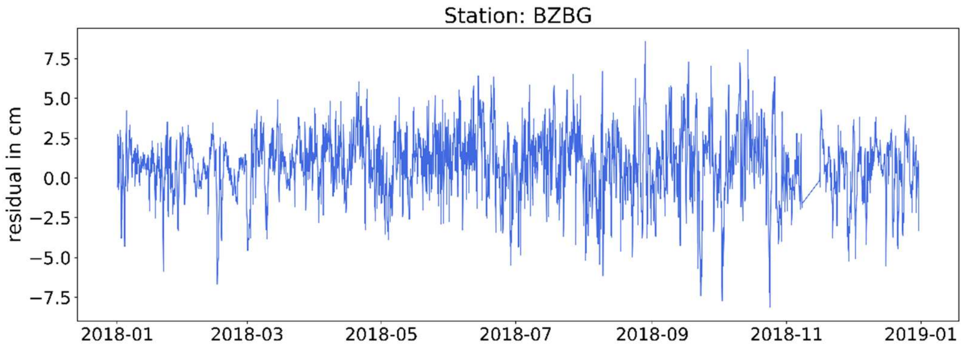


Figure 37: Residuals for underrepresented station BZBG (1'000 samples)

The curve is now only shifted very slightly with a residual mean of 0.8 cm. A certain bias however is still existing and the rmse of the prediction is still higher – although not by a large margin – compared to when all of the existing samples for this station where included in the training dataset (2.16 cm vs. 1.92 cm).

4.6.5. Underrepresented station: HGGL

In the case of the second station adding the 100 samples already improved the result dramatically.

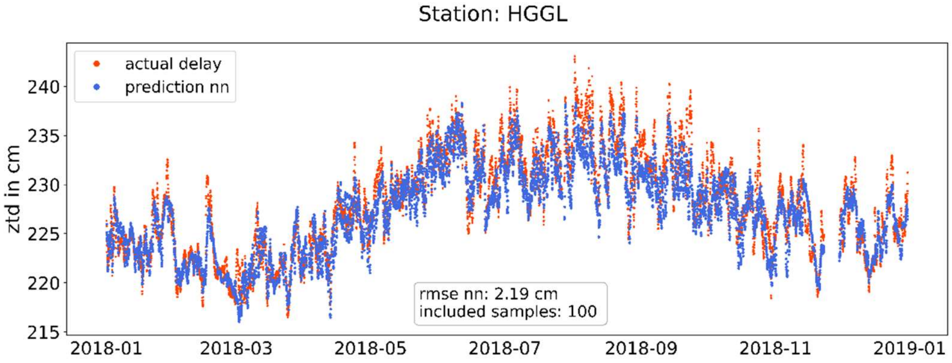


Figure 38: Time series of predictions for underrepresented station HGGL (100 samples)

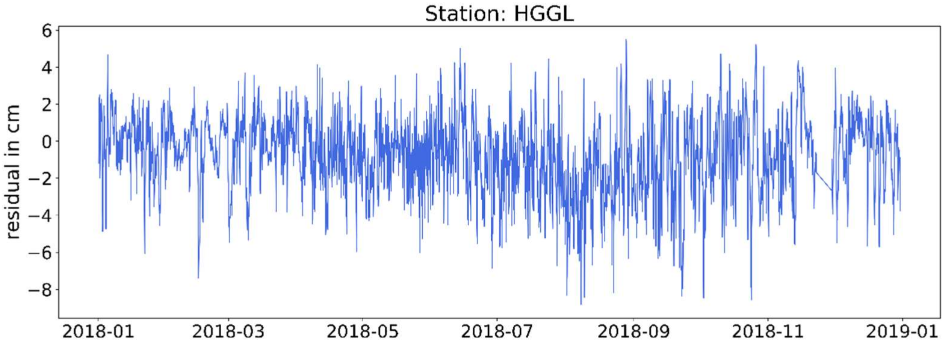


Figure 39: Residuals for underrepresented station HGGL (100 samples)

The resulting bias of the prediction is very small. In fact amounts the mean of the residuals only up to -0.79 cm. Additionally is the rmse only slightly higher than it was before (1.91 cm).

By adding another 900 samples into the training dataset, the bias disappears almost completely.

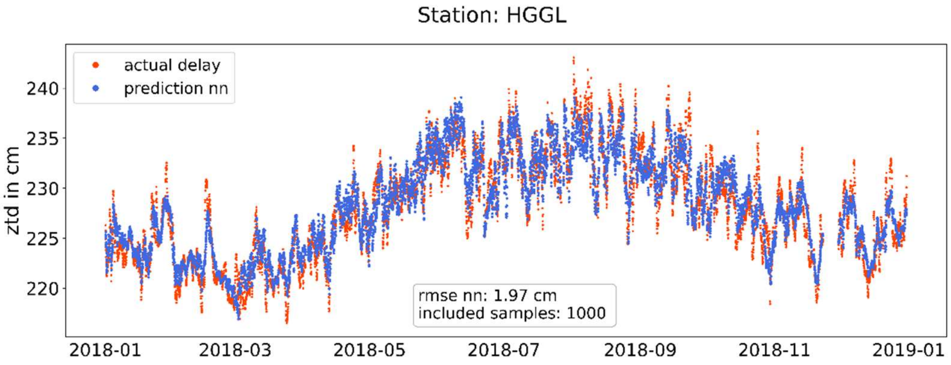


Figure 40: Time series of predictions for underrepresented station HGGL (1'000 samples)

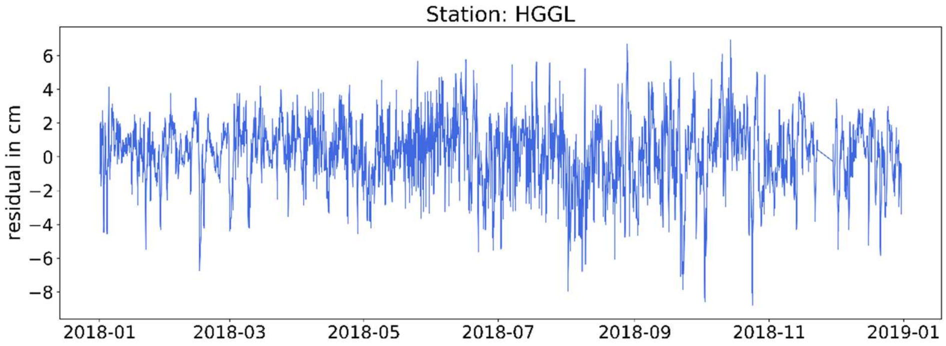


Figure 41: Residuals for underrepresented station HGGL (1'000 samples)

The rmse is – with a difference of 0.06 cm – now at almost the same level where it was when this station was fully included during the training. This is also reflected in the almost inexistent bias. The mean of the residuals for this case is only 0.21 cm. Interestingly enough is this even lower than for the original prediction for this station where the mean of the residuals adds up to 0.28 cm. This also shows that the original network can absolutely be biased with respect to individual stations. Only the mean of the residuals of the entire network is approximately zero.

So it seems that at least for the HGGL station, that the bias can be almost prevented by only adding 100 samples into the training dataset. With 1’000 samples the accuracy of the prediction is almost exactly as good as for the original dataset. Which after all consisted of over 44’000 samples for this particular station. In the case of the BZBG station however the results do look less promising. While adding the 100 samples basically did not improve anything, did adding 1’000 samples of this station get at least close to the original accuracy. These differences in the results maybe could relate to a poor distribution of the samples with respect to for example the time of year. It is however strange that the results for the BZBG station are clearly worse for both cases. The 100 and 1’000 samples were chosen completely randomly and independent from each other after all. This would suggest that problem lies somewhere else and would need to be investigated further.

Finally it should be noted that a reliable prediction would not be possible if every station would only be represented in the training dataset with 1’000 samples. These tests should merely indicate how many samples the network needs to not include the mentioned bias in its prediction. Assuming of course that all the other stations of the network are included with all the available samples.

5. Conclusion

Both random forest and neural network can predict the zenith path delay for stations they have seen during the training with an overall network root mean squared error of slightly above 1.6 cm. For individual stations however the size of this error varies quite a lot. The mean error for some stations is at about double the size compared to lowest ones. For no station in the network however amounts the root mean squared error of the predictions up to more than 2 cm. The complete underlying causes of these differences in accuracy could not be completely determined. Although seems there to be a clear correlation with the height of the particular stations. Highly elevated stations tend to perform much better with error values in the range of 1 cm. This is likely caused by the smaller size of the wet path delay in extreme altitudes. Additionally do the results indicate that the accuracy of the predictions benefit from an even distribution of the corresponding meteorological stations as well as the distance to those stations. Furthermore could a clear dependence of the error of the predictions on the time of year be identified. Once again could this likely be linked to the amount of water vapour in the troposphere as this factor is influenced by strong seasonal variations.

When compared to the predictions of the more commonly used Saastamoinen model, a relatively high improvement in accuracy can be determined. The overall root mean squared error is reduced from slightly above 2 cm to the mentioned 1.6 cm. This corresponds to a relative reduction in error of about 20 %. By looking at the monthly resulting error values it can be seen that this reduction turns out generally higher during the summer and autumn months. As mentioned before, is the amount of water vapour and therefore the resulting wet path delay considerably higher during these season. This might suggest that a large part of the reduction in error the machine learning approaches show is based on their capability of dealing more successful with the wet path delay. Furthermore do both random forest and neural network tend to smooth out their predictions. This is somewhat expected as in case of the random forest the predictions results from averaging the predictions of many individual trees. The neural network on the other hand minimizes the root mean squared error of the predictions which also tends to penalize large residuals. Saastamoinen however shows an evident bias of about 0.3 cm and thus tends to overpredict the delay.

The current neural network approach was however not fully capable to make reliable predictions at locations it has not seen during its training. While the overall curve of the prediction seems to fit quit accurately to the corresponding actual delay, the network also introduced a clear bias in its prediction. This bias amounts up to several centimeters, seemingly depending on the distance between the location of the prediction and the nearest reference point in the training data. By not completely excluding the corresponding locations, but just severely underrepresenting them during the training phase, the bias decreased considerably. This would suggest that the network learns to predict the actual curve of the delay reliably from the tropospheric information in the training data. The shift of this curve however is only learned by heart depending on the available locations in the training data.

All in all can it be concluded that the potential of applying machine learning approaches to predict the tropospheric zenith delay has been shown.

6. Outlook

There are several possibilities to continue with the results obtained during the course of this thesis. The first one would be to fully eliminate the occurring bias when predicting for unknown locations. As already addressed would the easiest solution be to simply use a larger dataset during the training. This dataset would not necessarily need to consist of as many samples over such a long period of time as the dataset used for this thesis since using 1'000 samples during the training already reduced the bias drastically. It would however need a much more densely distributed set of locations. This solution could also be combined with another machine learning model that only learns and predicts the explicit bias depending on the distance to the nearest reference point. This resulting prediction of the bias could then be applied to the prediction of the original network. Furthermore could a variation of a Convolutional Neural Network (CNN) be used to extract information about the spatial correlation that a densely distributed dataset could provide. These proposed options could also provide additional insight into the dependence of the error of the prediction on the distribution of the meteorological stations as well as the corresponding distance. This dependence would need to be analyzed further in any case as the results obtained in this thesis could not provide a clear conclusion.

To extract the temporal correlation in the available time series and use that information to increase the accuracy of the prediction recurrent neural networks (RNN) could be used. Such an approach could also be used to learn on solely the time series of the delay and then make prediction for the following hours when given a time series as input. These advantages come however at the cost of drastically increased computational time.

Another possibility would be to further investigate the application of algorithms not based on neural networks such as random forests or gradient boosting machines. As seen in this thesis can a large random forest model almost achieve the accuracy of the used neural network. They are generally faster to train and need less data to make reliable predictions. Deep learning usually is only preferable in cases where very large datasets are available or when the data consists of a very specialized format (images, audio, etc.).

With ever more increasing available data, the application of machine learning approaches within the topic of tropospheric delay in particular and the field of satellite geodesy in general will certainly become more common in the near term future. Especially in cases where the resulting predictions do not need to be retraceable they can provide advantages in terms of both accuracy and computational time.

7. Acknowledgment

I would like to thank the following people.

- Endrit Shehaj, for the numerous inputs and the continuous help during the course of this thesis.
- Dr. Stefano D'Aronco and Dr. Jan Dirk Wegner, for their helpful input and expertise in any machine learning related questions.
- Prof. Dr. Markus Rothacher and Prof. Dr. Alain Geiger, for making this thesis possible.
- Marcelina Los, for providing the map of the distribution of GNNS and meteorological stations shown in this report.

8. References

GPS based Determination of the Integrated and Spatially Distributed Water Vapor in the Troposphere [Troller, 2004]

On Interpretability of Artificial Neural Networks [Fan et al., 2020]

MACHINE LEARNING BASED PREDICTION OF ATMOSPHERIC ZENITH WET DELAY: A STUDY USING GNSS MEASUREMENTS IN WETTZELL AND CO-LOCATED VLBI OBSERVATIONS [Kitprache et al., 2019]

A machine learning-based regression technique for prediction of tropospheric phase delay on large-scale Sentinel-1 InSAR time-series [Shamshiri et al., 2019]

GNSS Meteorology in Spatially Dense Networks [Hurter, 2014]

Random Decision Forest [Ho, 1995]

The Elements of Statistical Learning 2nd edition [Stanford, 2017]

https://web.stanford.edu/~hastie/ElemStatLearn/

Scikit-learn documentation [Scikit-learn, 2020]

https://scikit-learn.org/

CS231n: Convolutional Neural Networks for Visual Recognition [Stanford, 2019]

http://cs231n.github.io/

Keras documentation [Keras, 2020]

https://keras.io/

Swisstopo

https://www.swisstopo.admin.ch/en/home.html

MeteoSwiss

https://www.meteoswiss.admin.ch/home.html?tab=overview

Appendix A: Distances from each GNSS station to its four nearest meteorological stations

GNSS	Meteo 1	Dist 1	Meteo 2	Dist 2	Meteo 3	Dist 3	Meteo 4	Dist 4
AIGE	GVE	0.06	CGI	18.68	DOL	19.84	BIE	34.93
ARD2	SCU	6.28	NAS	6.33	BUF	15.04	SMM	25.73
ARDE	SCU	6.28	NAS	6.33	BUF	15.04	SMM	25.74
BCKL	TAE	7.28	HOE	18.46	HAI	21.14	KLO	22.19
BLFT	FAH	23.32	DEM	47.95	COY	52.5	BAS	55.35
BOU2	DEM	10.16	FAH	22.09	COY	25.96	GRE	27.69
BOUR	DEM	10.16	FAH	22.1	COY	25.97	GRE	27.69
BZBG	PSI	6.07	BEZ	7.73	LEI	10.43	BUS	14.89
CRDM	MVE	9.39	MTE	11.66	EVO	13.15	SIO	15.53
DAV2	DAV	0.02	WFJ	3.79	ARO	12.78	LAT	21.76
DAVO	DAV	0.01	WFJ	3.79	ARO	12.76	LAT	21.75
EPFL	PUY	7.71	VIT	15.6	BIE	17.31	VEV	19.78
ERDE	SIO	3.6	DIA	12.38	MVE	14.98	ATT	15.32
ETH2	REH	2.35	SMA	5.28	UEB	6.41	KLO	8.27
ETHZ	REH	2.36	SMA	5.28	UEB	6.4	KLO	8.29
EXWI	BER	4.77	BAN	7.46	MSK	11.86	MUB	12.46
FALE	ILZ	3.52	CMA	5.81	ELM	13.91	VLS	19.9
FHBB	BAS	4.22	STC	5.59	MOE	18.51	RUE	21.24
FLDK	VAD	12.5	OBR	16.33	SAE	18.18	RAG	24.6
FRI3	PSI	8.96	LEI	9.68	BEZ	9.75	BUS	16.1
FRIC	PSI	8.94	LEI	9.67	BEZ	9.73	BUS	16.1
HABG	MER	2.05	GIH	11.37	TIT	18.85	ENG	19.24
HCHS	LEI	13.01	BEZ	17.54	PSI	19.58	HLL	20.07
HGGL	BUS	13.6	LAE	14.51	MOA	16.49	PSI	16.54
HOGR	ZER	8.46	GOR	8.9	MRP	11.47	MTE	18.99
HOH2	VIS	6.45	BLA	12.17	GRC	14.94	MTE	21.87
HOHT	VIS	6.45	BLA	12.17	GRC	14.95	MTE	21.87
HUTT	WYN	13.17	EGO	13.57	NAP	17.16	KOP	17.58
KALT	EBK	9.76	LAC	12.14	HOE	17.74	GLA	20.86
KOPS	NAS	20.68	SCU	23.82	DAV	27.38	WFJ	28.28
KREB	SIM	9.68	GRC	11.74	ZER	20.4	VIS	20.63
KREU	GUT	10	HAI	10.29	BIZ	16.96	TAE	26.29
LECH	OBR	43.29	SRS	45.2	NAS	46.19	VAD	48.35
LFNB	LEI	9.92	MOE	13.62	BEZ	13.83	PSI	14.28
LIND	ARH	13.53	OBR	21.5	STG	27.65	GUT	32.6
LOMO	OTL	0.04	CIM	3.37	MAG	11.38	CEV	21.73
LUCE	DEM	11.43	FAH	24.73	BAS	26.37	GRE	30.89
LUZE	LUZ	3.53	PIL	10.71	CHZ	18.25	GES	18.73
MAR2	MAR	4.22	EVI	7.56	MOB	13.24	ATT	15.67
MAR3	MAR	4.22	EVI	7.56	MOB	13.24	ATT	15.67
MRGT	WYN	6.35	GOE	12.98	EGO	14.54	RUE	18.19
MTTI	DEM	13.85	FAH	18.21	COY	21.57	CHA	27.62
NEUC	NEU	1.18	CHM	6.84	CRM	10.82	CDF	15.01
OALP	GUE	4.57	GOS	7.1	ANT	7.87	DIS	14.72
PAYE	PAY	0.13	GRA	13.73	MAS	19.79	NEU	20.89
PFA2	ARH	16.8	OBR	20.16	STG	30.78	GUT	39.23
PRNY	FRE	19.5	BRL	22.13	MAH	25.61	CHB	26.18

RAND	ZER	9.73	GRC	10.27	MTE	11.88	GOR	14.74
SAA2	CHD	13.04	BOL	13.59	ABO	20.09	CDM	20.81
SAAN	CHD	13.04	BOL	13.59	ABO	20.09	CDM	20.82
SAM2	SAM	0.01	COV	12.92	SIA	13.77	LAT	14.81
SANB	SBE	0.03	VLS	18.23	COM	19.17	MTR	20.85
SANE	DIA	8.15	MVE	14.45	CDM	15.62	SIO	16.13
SAR2	RAG	3.85	SRS	12.27	CHU	12.62	VAD	16.13
SCHA	SHA	5.95	HLL	14.61	HAI	29.25	KLO	30.07
SLTB	RUE	12.89	STC	13.06	BAS	13.54	MOE	18.06
STA2	SBO	1.56	GEN	10.01	LUG	16.52	MAG	33.82
STAB	SBO	1.56	GEN	10.01	LUG	16.51	MAG	33.82
STCX	FRE	6.08	MAH	10.8	BRL	19.31	CHB	22.24
STDL	KLO	7.48	LAE	9.88	REH	12.71	HLL	17.48
STGA	STG	4.36	BIZ	10.42	ARH	17.29	GUT	18.48
TRLK	SHA	8.04	HLL	18.24	KLO	21.67	TAE	23.68
VARE	MVE	11.1	VIS	18.56	MTE	18.65	ABO	19.94
WAB1	BER	7.45	BAN	7.78	MSK	14.16	MUB	15.19
WAB2	BER	7.45	BAN	7.77	MSK	14.16	MUB	15.2
WEHO	MVE	9.42	ABO	14.03	SIO	21.37	DIA	21.6
WLCH	HLL	3.35	SHA	10.84	KLO	21.36	LAE	21.58
ZERM	ZER	3.48	GOR	4.78	MRP	8.17	MTE	18.31
ZIM2	BAN	12.2	BER	12.64	MSK	16.18	THU	16.87
ZIM3	BAN	12.2	BER	12.64	MSK	16.18	THU	16.87
ZIMJ	BAN	12.19	BER	12.63	MSK	16.19	THU	16.87
ZIMM	BAN	12.19	BER	12.64	MSK	16.2	THU	16.86

Appendix B: Statistics of all random forest predictions

GNSS	Rmse [cm[	Mae [cm]	Mre [%]	Res over 2 cm [%]	Mean res [cm]
AIGE	1.86	1.45	0.63	25.87	0.12
ARD2	1.09	0.85	0.42	6.92	0.05
ARDE	1.13	0.87	0.43	7.67	-0.08
BCKL	1.91	1.46	0.64	26.68	-0.05
BLFT	1.98	1.51	0.65	28.26	-0.1
BOU2	1.85	1.42	0.65	26.12	-0.08
BOUR	1.89	1.46	0.67	27.01	-0.1
BZBG	2.01	1.53	0.67	28.43	-0.05
DAV2	1.14	0.87	0.44	8.61	-0.02
DAVO	1.14	0.87	0.43	8.7	-0.12
EPFL	1.9	1.46	0.63	25.27	-0.13
ERDE	1.31	1.02	0.46	10.85	-0.07
ETH2	1.93	1.48	0.65	27.31	0.04
ETHZ	1.94	1.47	0.65	27.07	-0.08
FALE	1.26	0.95	0.46	10.73	-0.08
FHBB	1.91	1.48	0.63	26.9	-0.09
FLDK	1.49	1.12	0.49	15.66	-0.13
FRI3	2.04	1.56	0.7	29.54	-0.01
FRIC	2.03	1.57	0.7	29.7	0.2
HABG	1.28	0.98	0.46	11.1	0.1
HCHS	1.85	1.43	0.66	24.96	-0.03
HGGL	1.94	1.49	0.65	27.23	0.06
HOGR	0.98	0.74	0.48	5.2	-0.02
HOH2	1.28	0.98	0.45	10.56	0.07
HOHT	1.28	0.97	0.45	10.75	-0.1
HUTT	1.85	1.41	0.64	24.89	0.05
KALT	1.72	1.31	0.57	22.23	-0.14
KOPS	1.19	0.92	0.48	9.86	0.1
KREB	1.28	1.01	0.57	11.87	0.25
KREU	2.04	1.6	0.7	30.58	0.32
LECH	1.3	0.99	0.51	11.64	-0.06
LFNB	2.03	1.56	0.68	29.1	-0.1
LIND	1.94	1.48	0.64	27.07	0.16
LOMO	1.52	1.19	0.51	16.9	0.08
LUCE	1.93	1.48	0.66	27.26	-0.13
LUZE	1.66	1.28	0.56	20.39	0.16
MAR2	1.37	1.06	0.47	13.38	-0.15
MAR3	1.37	1.06	0.47	13.22	-0.07
MRGT	1.9	1.46	0.64	26.13	-0.05
MTTI	1.92	1.48	0.65	27.49	-0.25
NEUC	1.78	1.38	0.6	23.66	0.06
OALP	1.14	0.88	0.47	8.19	0.07
PAYE	1.85	1.42	0.62	24.6	-0.01
PFA2	1.83	1.41	0.66	25.1	0.04
PRNY	1.91	1.48	0.67	27.54	0.14
RAND	1.02	0.78	0.44	5.45	0
SAA2	1.51	1.14	0.56	16.6	-0.02

SAAN	1.53	1.16	0.56	16.68	-0.08
SAM2	1.21	0.95	0.48	10.13	0.17
SANB	1.3	1.02	0.52	11.55	0.03
SANE	1.29	1.02	0.54	11.08	0.18
SAR2	1.5	1.14	0.54	16.43	-0.17
SCHA	1.96	1.51	0.67	27.77	-0.05
SLTB	1.96	1.51	0.66	27.93	-0.03
STA2	1.58	1.25	0.53	18.92	0.17
STAB	1.57	1.25	0.53	18.79	0.11
STCX	1.84	1.41	0.67	25.05	-0.07
STDL	1.97	1.51	0.66	27.85	-0.06
STGA	1.88	1.45	0.65	26.24	0.21
TRLK	1.95	1.48	0.65	26.76	-0.14
VARE	1.43	1.1	0.49	14.46	-0.1
WAB1	1.83	1.38	0.61	23.63	-0.11
WAB2	1.84	1.42	0.62	25.24	0.3
WEHO	1.19	0.93	0.56	8.6	0.12
WLCH	1.99	1.53	0.67	28.49	-0.1
ZERM	1.03	0.79	0.41	5.88	-0.05
ZIM2	1.73	1.32	0.61	22.3	-0.02
ZIM3	1.73	1.32	0.61	22.36	-0.01
ZIMJ	1.73	1.33	0.61	22.45	0.08
ZIMM	1.73	1.32	0.61	22.37	-0.04

Appendix C: Statistics of all neural network predictions

GNSS	Rmse [cm[	Mae [cm]	Mre [%]	Res over 2 cm [%]	Mean res [cm]
AIGE	1.8	1.37	0.59	23.36	0.03
ARD2	1.05	0.82	0.4	6.05	0.15
ARDE	1.08	0.81	0.4	7.33	-0.17
BCKL	1.85	1.4	0.61	24.33	-0.04
BLFT	1.96	1.5	0.64	27.26	-0.46
BOU2	1.82	1.39	0.63	24.28	-0.16
BOUR	1.83	1.41	0.65	25.04	-0.08
BZBG	1.92	1.45	0.64	25.77	0.03
DAV2	1.1	0.83	0.42	7.21	0.09
DAVO	1.09	0.82	0.41	7.35	-0.08
EPFL	1.86	1.43	0.61	25.09	-0.16
ERDE	1.3	0.98	0.44	12	-0.48
ETH2	1.88	1.43	0.63	25.99	0.05
ETHZ	1.88	1.43	0.63	25.75	0
FALE	1.19	0.9	0.43	9.46	-0.11
FHBB	1.82	1.41	0.6	25.67	-0.04
FLDK	1.35	1.02	0.45	13.28	0
FRI3	1.94	1.48	0.66	26.91	0.12
FRIC	1.98	1.54	0.69	28.68	0.41
HABG	1.19	0.9	0.43	9.24	0.13
HCHS	1.81	1.38	0.64	23.56	0.1
HGGL	1.91	1.48	0.65	26.8	0.28
HOGR	0.95	0.71	0.46	4.73	-0.07
HOH2	1.19	0.89	0.41	9.37	0.03
HOHT	1.21	0.91	0.42	9.45	-0.15
HUTT	1.82	1.38	0.62	24.1	0
KALT	1.7	1.3	0.56	21.15	-0.08
KOPS	1.14	0.87	0.45	8.45	0.06
KREB	1.13	0.89	0.5	7.28	0.28
KREU	1.95	1.53	0.67	28.27	0.33
LECH	1.26	0.96	0.5	11.42	0.18
LFNB	1.94	1.47	0.64	26.61	-0.07
LIND	1.84	1.41	0.61	25.48	0.14
LOMO	1.45	1.14	0.49	15.46	0.21
LUCE	1.84	1.41	0.63	24.97	-0.18
LUZE	1.58	1.21	0.53	18.05	0.19
MAR2	1.28	0.98	0.43	11.08	-0.32
MAR3	1.27	0.97	0.43	10.86	-0.28
MRGT	1.85	1.41	0.62	24.46	0.11
MTTI	1.88	1.44	0.63	25.5	-0.38
NEUC	1.74	1.34	0.58	22.58	-0.01
OALP	1.12	0.87	0.47	7.34	0.3
PAYE	1.8	1.37	0.6	23.5	-0.07
PFA2	1.75	1.34	0.63	23.36	0.06
PRNY	1.84	1.41	0.64	24.25	0.05
RAND	1	0.75	0.42	5.58	0.13
SAA2	1.51	1.14	0.55	16.66	-0.19

SAAN	1.55	1.17	0.57	17.46	-0.39
SAM2	1.1	0.85	0.43	7.46	0.11
SANB	1.25	0.97	0.49	10.52	0.12
SANE	1.21	0.93	0.5	9.24	0.07
SAR2	1.42	1.07	0.51	14.49	-0.12
SCHA	1.93	1.47	0.65	25.9	-0.18
SLTB	1.9	1.46	0.64	26.23	0.03
STA2	1.54	1.22	0.52	18.17	0.28
STAB	1.54	1.22	0.52	18.17	0.28
STCX	1.8	1.37	0.65	23.33	-0.13
STDL	1.91	1.45	0.63	25.96	-0.05
STGA	1.81	1.39	0.62	24.93	0.17
TRLK	1.9	1.44	0.63	25.21	-0.13
VARE	1.34	1.02	0.45	12.69	-0.17
WAB1	1.8	1.36	0.6	23.67	0.08
WAB2	1.78	1.35	0.59	23.09	0.13
WEHO	1.14	0.86	0.51	8.06	-0.13
WLCH	1.92	1.46	0.64	26.34	-0.06
ZERM	0.99	0.76	0.4	5.19	0.15
ZIM2	1.7	1.28	0.59	21.04	-0.03
ZIM3	1.69	1.28	0.59	20.94	-0.02
ZIMJ	1.69	1.28	0.59	20.98	0.04
ZIMM	1.69	1.27	0.59	20.96	-0.1
WEHO	1.14	0.86	0.51	8.06	-0.13

Appendix D: Statistics of all Saastamoinen predictions

GNSS	Rmse [cm[	Mae [cm]	Mre [%]	Res over 2 cm [%]	Mean res [cm]
AIGE	1.98	1.55	0.67	29.32	0.11
ARD2	1.27	0.97	0.48	11.95	-0.17
ARDE	1.37	1.04	0.52	14.46	-0.49
BCKL	2.01	1.55	0.68	29.64	-0.01
BLFT	2.43	1.93	0.83	40.01	0.46
BOU2	2.89	2.17	1	42.55	0.63
BOUR	2.95	2.23	1.02	43.68	0.75
BZBG	2.06	1.6	0.7	30.74	0.15
DAV2	1.27	0.99	0.5	11.87	0.19
DAVO	1.26	0.97	0.49	11.72	0.02
EPFL	2.2	1.71	0.74	33.89	0.61
ERDE	2.01	1.56	0.7	28.68	0.13
ETH2	2.01	1.55	0.68	29.81	0.16
ETHZ	2	1.55	0.68	29.59	0.1
FALE	1.97	1.57	0.76	30.17	0.57
FHBB	1.94	1.51	0.64	27.57	-0.29
FLDK	2.04	1.6	0.7	30.78	0.32
FRI3	2.31	1.79	0.8	35.08	-0.34
FRIC	2.28	1.77	0.79	34.97	-0.06
HABG	1.65	1.3	0.61	22.05	0.14
HCHS	3.21	2.57	1.18	54.7	1.62
HGGL	2.07	1.61	0.7	30.71	0.31
HOGR	1.83	1.43	0.92	27.01	1.2
HOH2	1.96	1.54	0.71	29.14	0.7
HOHT	1.91	1.49	0.69	27.42	0.51
HUTT	2.24	1.77	0.8	36.5	0.89
KALT	2	1.55	0.67	29.71	-0.27
KOPS	1.44	1.09	0.57	15.54	-0.16
KREB	1.92	1.45	0.81	25.41	0.93
KREU	2.56	2.06	0.9	44.34	1.3
LECH	2.04	1.54	0.79	28.25	-0.03
LFNB	2.23	1.75	0.77	35.44	0.57
LIND	2.15	1.67	0.72	33.21	0.13
LOMO	1.78	1.38	0.59	24.1	-0.43
LUCE	2.47	1.91	0.85	38.75	0
LUZE	1.92	1.5	0.65	27.67	0.31
MAR2	2.63	2.06	0.91	42.76	0.4
MAR3	2.63	2.06	0.91	42.81	0.43
MRGT	2.05	1.61	0.7	31.42	0.5
MTTI	2.22	1.7	0.74	33.29	-0.02
NEUC	2	1.57	0.68	29.84	0.15
OALP	2.01	1.66	0.9	32.6	1.48
PAYE	1.9	1.47	0.64	27.15	0.11
PFA2	3.43	2.61	1.22	49.91	0.46
PRNY	2.49	1.92	0.87	39.26	0.29
RAND	2.25	1.78	1	34.74	1.48
SAA2	1.95	1.52	0.74	28.23	0.69

SAAN	1.89	1.47	0.72	26.94	0.51
SAM2	1.22	0.95	0.48	10.26	0.21
SANB	1.35	1.05	0.53	13.95	-0.08
SANE	2.07	1.56	0.83	28.99	0.83
SAR2	1.61	1.23	0.59	19.39	-0.13
SCHA	2.25	1.78	0.78	34.91	-0.23
SLTB	2.15	1.7	0.74	34.76	0.66
STA2	1.75	1.37	0.59	24.16	0.16
STAB	1.75	1.37	0.59	24.09	0.15
STCX	2.06	1.61	0.76	31.24	0.34
STDL	2.02	1.56	0.67	29.76	-0.22
STGA	2.01	1.57	0.71	29.89	0.56
TRLK	2.04	1.58	0.69	30.23	-0.29
VARE	1.72	1.33	0.59	22.1	-0.27
WAB1	1.96	1.51	0.67	29.02	0.28
WAB2	1.94	1.5	0.66	28.42	0.34
WEHO	2.07	1.66	1	32.76	1.48
WLCH	2.06	1.61	0.7	31.06	-0.29
ZERM	1.38	1.06	0.55	14.51	-0.12
ZIM2	2.16	1.7	0.78	33.98	0.96
ZIM3	2.16	1.7	0.78	33.84	0.97
ZIMJ	2.22	1.76	0.81	35.6	1.08
ZIMM	2.14	1.67	0.77	33.4	0.89