Utrecht University

Department of Information and Computing Sciences

Master Thesis Computing Science

An Algorithm for Correlation Halving Distance Analysis of Renewable Energy Resources

Daily supervisor:

Laurens P. Stoop

Candidate:

Mustafa Majid

First Examiner:

Ad Feelders

Second Examiner:

Arno Siebes

July 10, 2023

Abstract

Europe's transition from fossil fuel energy to renewable energy sources requires expensive changes to the continent's electricity grid that should hold up for decades. As renewable energy generation methods such as solar and wind are heavily dependent on changes in the weather, there will be increased variability in the power supply. To reduce this energy-meteorological variability, areas of Europe's grid that have low renewable energy generation correlation must be discovered. By using conversion models on climate model output to get relevant energy variables, there is hourly data available for solar and wind energy capacity factors for each grid cell in Europe. Due to the sheer number of grid cells in the data (21,019), calculating correlation between all pairs of grid cells is not feasible without algorithm optimisation. We introduce a novel metric called the "Correlation Halving Distance", which gives the distance value that indicates at what distance the wind and/or solar time series yield 0.5 correlation for any given grid cell. We also explore optimised approaches to calculate the metric efficiently. Here we show that one algorithm based on Active Learning, called Uncertainty Sampling, performed the best on synthetic data and was chosen to be tested on real-world data. In validation, Uncertainty Sampling yields a correlation value of [0.5±0.05] in 87 out of a 100 experiments with random starting grid cells. Additionally, each run calculated only 62 correlations on average, greatly saving on computation cost compared to the brute force approach. We found that the correlation halving distance values varied greatly by geography. Grid cells in land-locked and mountainous eastern Switzerland and western Austria show Correlation Halving Distance values of 105-110 km, while grid cells in the North Sea area show values in the order of 435-440 km. The metric could assist in future-proofing changes to Europe's energy grid as it transitions to renewable energy, given that many types of renewable energy sources rely on specific weather conditions. Additionally, spatial interpolation techniques could be utilised to estimate the Correlation Halving Distance for cells to further reduce the number of computations.

Contents

1. Introduction

Europe is rapidly transitioning from fossil fuel to renewable energy sources to mitigate environmental impacts, especially those caused by climate change [1]. This transition requires costly changes to the electricity grid of Europe that should hold up for multiple decades [2]. The transition from non-renewable energy sources to more renewable resources increases the weather dependence and, therefore, the variability of electricity generation [3], [4].

To ensure a consistent and adequate power supply given the extreme variability of the weather [5], and to prevent issues such as blackouts, this weather-dependent energy generation variability must be reduced [6]. To achieve this, areas of the grid that have a relatively low correlation in their renewable resource must be identified, so that the power supply of these areas may be connected to reduce power supply variability. However, given the high resolution of the underlying climate model data and the extent of the interconnected energy grid across the whole continent of Europe, this can be extremely computationally intensive to compute. Consequently, an alternative, more efficient method should be used [7].

Grams, Beerli, Pfenninger, *et al.* [6] propose the understanding of weather patterns across Europe to optimise deployment of wind and solar energy generation resources to reduce variability in power output. The weather patterns in question are called weather regimes, which are spatial patterns of the weather systems extending over about 1,000 km that last on average five days. If persistent, some of these weather regimes can cause a loss of wind power across neighbouring countries in Europe [8].

As wind and solar power output are heavily dependent on the state of the atmosphere and, thus, weather regimes [3], [6], [9], it is critical to understand inter-regime behaviour across European regions to select the best areas to deploy wind capacity in order to reduce volatility in power generation. For

example, deploying wind capacity in the Balkans instead of the North Sea region was found to dramatically reduce energy output variation, as the Balkans regions have inter-regime behaviour different from the North Sea [6] . Understanding inter-regime behaviour to guide wind power deployment requires understanding of wind patterns all over continental Europe. Weather regimes provide some of the information needed, but they do not tell the whole story as other factors are at play [8].

The weather-dependent variability of energy generation over Europe's future electricity grid needs to be reduced. Therefore, areas of Europe's grid that have low renewable energy generation correlation must be found. For this purpose, a valuable dataset was utilised: using conversion models on climate model output, researchers created a dataset containing wind and solar generation information in hourly data points over 30 years. The dataset contains a grid of 21,019 grid cells over Europe, with each grid cell containing a time series with hourly data of wind and solar energy generation information [10]. To determine areas of low correlation, the time series correlation between all pairs of grid cells will need to be computed, which is a very expensive procedure given the number of grid cells.

In their work on annual flood data of a thousand European rivers, Berghuijs, Allen, Harrigan, *et al.* [11] defined a measure called the flood synchrony scale. The flood synchrony scale is the distance over which rivers flood almost synchronously half the time. Their findings show that the flood synchrony scale extends beyond the size of an individual river's drainage basin, and that years with spatially extensive flooding are serially correlated. By analysing the flood synchrony scale, they show that the typical approach of managing flood risk at the level of individual river basins is inadequate, as the synchronous nature of flooding of river basins necessitates accounting for flood risks beyond an individual basin's borders. In addition, the relevance of specific flood generating mechanisms can be linked to, and analysed by, the flood synchrony scale [12].

While the flood synchrony scale cannot be linked directly to the variability of renewable energy sources, the measure it analyses is very similar in functional form to one used to study the correlation between wind power generation

in Europe: Giebel [13] shows that by plotting cross-correlation in measured wind farm generation over time against distance, an exponential decay curve can be fit on the data. Additionally, Olauson and Bergkvist [14] calculated and analysed correlations of country-wide wind power output time series and found that correlations generally decrease exponentially with distance. For the purposes of making smart changes to Europe's energy grid during the transition to renewable energy resources, it would be helpful to have a measure akin to the flood synchrony scale but one that takes renewable energy correlation instead of flooding events into account. Instead of finding the distance over which rivers flood synchronously 50% of the time, the problem becomes finding the distance at which renewable energy output correlation drops to 0.5, assuming that correlation generally decreases with distance.

In this study we define the Correlation Halving Distance (CHD) as the distance at which the average energy output correlation is 0.5 around a given location [15]. By using the Correlation Halving Distance metric within the analysis of the flood synchrony scale, the spatial scale of synchronous variation of renewable energy generation can be studied to gain better understanding of energy-meteorological variability.

As spatial changes in the Correlation Halving Distance indicate where it could be beneficial to deploy additional renewable energy resources to balance the electricity grid, the CHD measure needs to be calculated for each location within Europe. While the flood synchrony scale was defined using the daily data of 600 locations, the data available for this study contains over 21,019 grid cells, each containing a time series with hourly data points. In addition, this task is very computationally intensive for multiple reasons: the time complexity is quadratic in the number of grid cells; grid cells covering all of Europe are included; and correlation must be calculated between two long time series for each pair of grid cells. Direct calculation of the metric is thus not feasible.

To reduce the number of computations required, we envisage the following two-part strategy. Firstly, we suggest calculating CHD only for a sample of grid cells and use spatial interpolation to obtain CHD values for the remaining grid cells in the dataset. Secondly, to approximate the CHD for a given grid cell,

one must postulate a suitable functional form for correlation against distance, and use smart sampling techniques, including those from the field of Active Learning, to estimate the CHD with a small number of samples. The goal of this thesis is the second part of the strategy.

The rest of this thesis is structured as follows: Chapter 2 introduces the concept of active learning and its relation to the current problem. Chapter 3 introduces the structure of real-world data used for the experiments. Chapter 4 includes a description of the research methodology. Chapter 5 includes the results of synthetic and real-world data experiments. Lastly, Chapter 6 contains a discussion on the study findings and areas for future work.

2. Active Learning

In order to sample grid cells to calculate Correlation Halving Distance efficiently, it is essential to draw insights from the field of Active Learning, a sub-field of machine learning that focuses on optimisation using advanced sampling techniques. By incorporating these techniques into our approach, we can enhance the efficiency and accuracy of the CHD values being computed. This section aims to introduce the definition of active learning with some established techniques in the field, as well as illustrate its relevance to the thesis goal of approximating CHD values with a limited number of samples.

Active learning is an approach utilised in machine learning to reduce the number of data points that require labelling, as data labelling is often required for supervised learning techniques like classification and regression. This typically helps in situations where labelling an entire dataset may be very expensive, so one must ascertain the smallest subset of samples to label that will yield the most accurate model [16]. In this study, the focus is on using sampling to reduce the number of computations rather than sampling labelled data for supervised learning but the approaches presented here can be adapted to the current problem.

One sampling approach that Wu [16] mentions in their work is greedy sampling, a strategy originally proposed by Yu and Kim [17]. This approach involves selecting the next sample that is furthest from the previously selected and labelled sample data point. Greedy sampling considers the diversity of data points, which is a measure of how much of the input space the samples cover.

Other sampling methods mentioned by Wu and Dongrui are Query-By Committee (QBC) and Expected Model Change Maximization (EMCM). In the context of regression and classification, QBC is a pool-based active learning approach that creates a committee of learners from existing pools of labelled

data [18] using bootstrapping, or by utilising different learning algorithms. Then it selects unlabelled samples from the pool to label based on the training points the learner committee disagrees on the most. EMCM is an active learning method that also makes use of bootstrapping [19]. It measures the change between the current learning model and a model that is trained on an expanded training set. Both QBC and EMCM focus on informativeness, a measure of the richness of the information obtained from a selected sample.

Uncertainty sampling is a widely used general active learning approach for assessing informativeness, where a learner selects the instance for which it has the highest "uncertainty" in terms of labelling . The measure of uncertainty can be defined in different ways, with entropy of a model's posteriors over its labels being a common example in the machine learning domain [20].

The concepts from the aforementioned sampling approaches will be adapted to the current problem of sampling time series contained within grid cells. Both greedy and uncertainty sampling approaches will be explored, but will be used to reduce the number of computations by informative grid cell sampling. For uncertainty sampling, a more suitable uncertainty metric will be proposed that is inspired by Query-By-Committee's learner "disagreement" aspect.

3. Data

This chapter provides an overview of the real-world data available on wind and solar energy, as well as discussing the size of the dataset as well as its peculiarities and challenges.

There is spatio-temporal data available of the renewable energy generation potential across the European region. This gridded data has a spatial resolution of approximately 25 to 35 km, with 30 years of hourly time steps per grid cell. For this study, the attention is focused on only the data available for the year of 2020.

The available data is a grid across Europe of time-series data containing the parameters: longitude, latitude, and time, with dimensions $201 \times 445 \times 8784$, respectively. The combination of latitude and longitude indicate the centre of a grid cell. The last dimension for time is the total number of hours in the entire dataset, as the data available cover all hourly timesteps in 2020. Note that all time values are in Coordinated Universal Time (UTC).

The latitude range is 30 to 80 ° North, while the longitude range is -31 to 80 ° East. The dataset contains latitude and longitude values with a resolution of 0.25 degrees. For conciseness, coordinates of this nature will be referred to using the tuple notation: for example, the point 30 ° N, 80 ° E will be written as (30,80).

As each cell in the grid contains a time series and we are calculating time series correlation, we can reduce the sampling problem to spatial sampling on a 2D grid as the time dimension is relevant to the correlation calculation, but not to the sampling of cells in the grid.

Each cell in the grid contains a time series where each hour has three values: the off-shore wind turbine, on-shore wind turbine, and solar photo-voltaic panel capacity factors. The capacity factor (CF) is the potential generation output across a period of time (an hour) for a given grid cell. The ERA5 reanalysis

dataset is used to derive the capacity factors [10]. The calculation of capacity factor is explained in Section 3 of [21].

3.0.1 Onshore Wind Capacity Factor

The full dataset available for Onshore Wind Capacity Factor can be examined visually for preliminary observations. Figure 3.1 shows the full extent of the available dataset. The available area is highlighted on the map, while excluded regions are shown in grey. There is no location in the available dataset where wind CF is above 0.95. The main region of interest is outlined by the red borders, as it holds implications for Europe's electricity grid.

Europe is typically categorised by low mean wind CF in the south but contains pockets of high mean values in the UK, the Netherlands, and Denmark. This effect can be seen clearly in Figure 3.2, which zooms in at the region of interest, where Utrecht, the city where this research was conducted, is marked with a red cross. The trend towards relatively higher mean CF values in the north appears to be broken by Norway, based on this figure.

As the data contains CF values for each hour, the onshore wind CF at different times of the year can be examined. Figure 3.3 shows the onshore wind CF for Europe for different times in 2020 for the 1st of March. It is clear that even at different times on the same day, there is noticeable variation in wind CF, let alone for the whole year. Additionally, there are pockets of high CF surrounded by regions of low CF and vice versa; this is evident in Figure 3.3b showing high wind CF in parts of south Poland and western Ukraine surrounded by low CF regions. Additionally, there is overall extremely low CF around Greece and in North Italy. This noise at individual time points can be obscured by the map showing the mean CF for the entire year. Additionally, the figure also highlights seasonal variation. The 1st of August is in meteorological summer, and it is clear that most of Europe has much lower wind CF values overall compared to the 1st of March, the start of meteorological Spring.

The time series for onshore wind CF for a particular location can be inspected more closely for peculiarities as well. In Figure 3.4, it appears that

the onshore wind CF for Utrecht at coordinates (52,5) oscillate frequently between the maximum of 0.95 and the minimum of 0. The data appears very noisy when inspected for the whole year as one time series graph. Figure 3.5 shows the variation over the hours across two days: the 1st of March, and the 1st of August, respectively. Note that the wind CF values also have different ranges on either day; the 1st of March is the start of meteorological Spring, and shows 0.95 CF most of the day. In contrast, the 1st of August falls in meteorological summer, which is an overall less windy season. The peak CF on this day does not surpass 0.40. With the wind CF time series of just one location, there is a great amount of complexity present that shows high variation at different time scales.

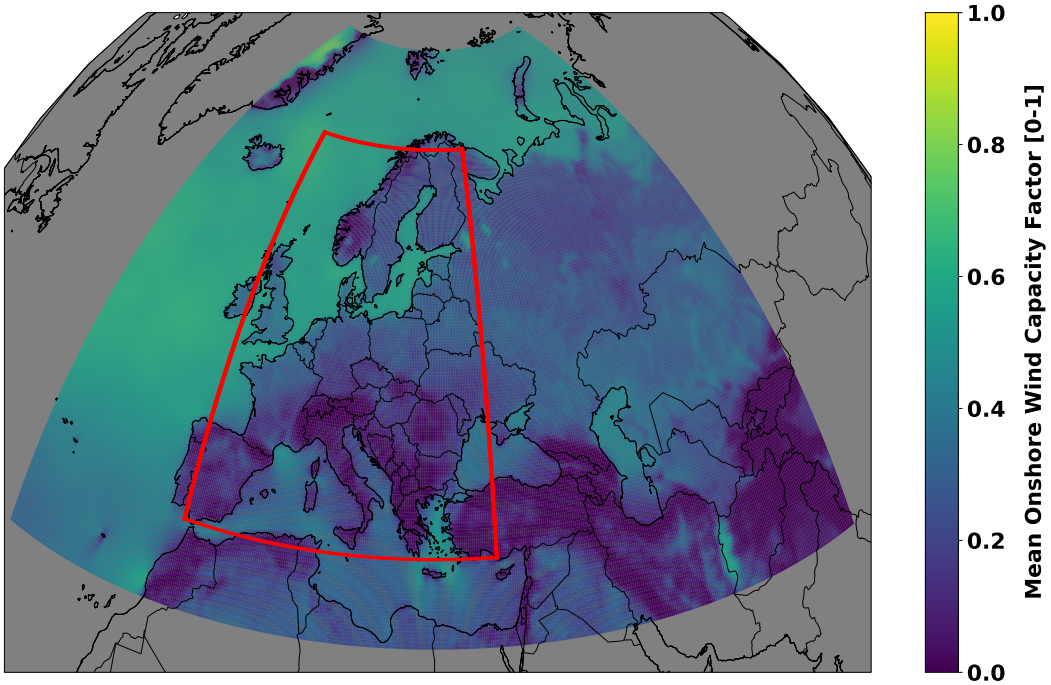


Figure 3.1: Mean capacity factor over the year 2020 in the available data, with red borders showing the region of interest for this study. The grey regions are outside of the dataset. The maximum value is 0.95, while the minimum value is 0.

3.0.2 Solar Capacity Factor

The latitude, longitude, and time dimensions are the same for solar CF as it is derived from the same dataset. Figure 3.6 shows the mean solar CF over the year 2020 over Europe. It is evident that the mean values for the region

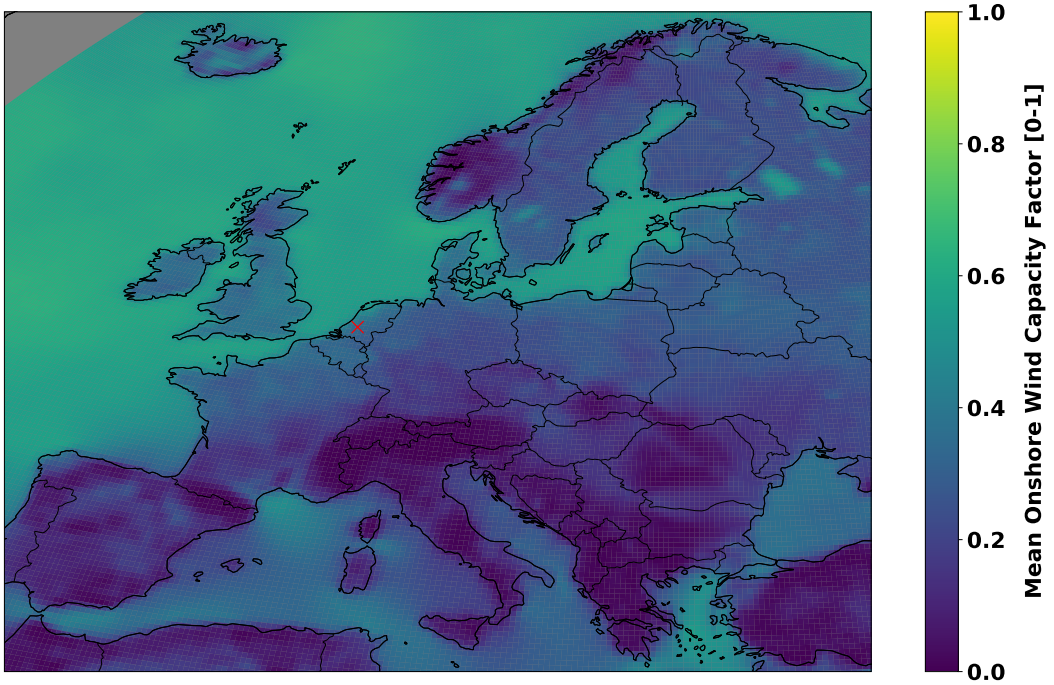
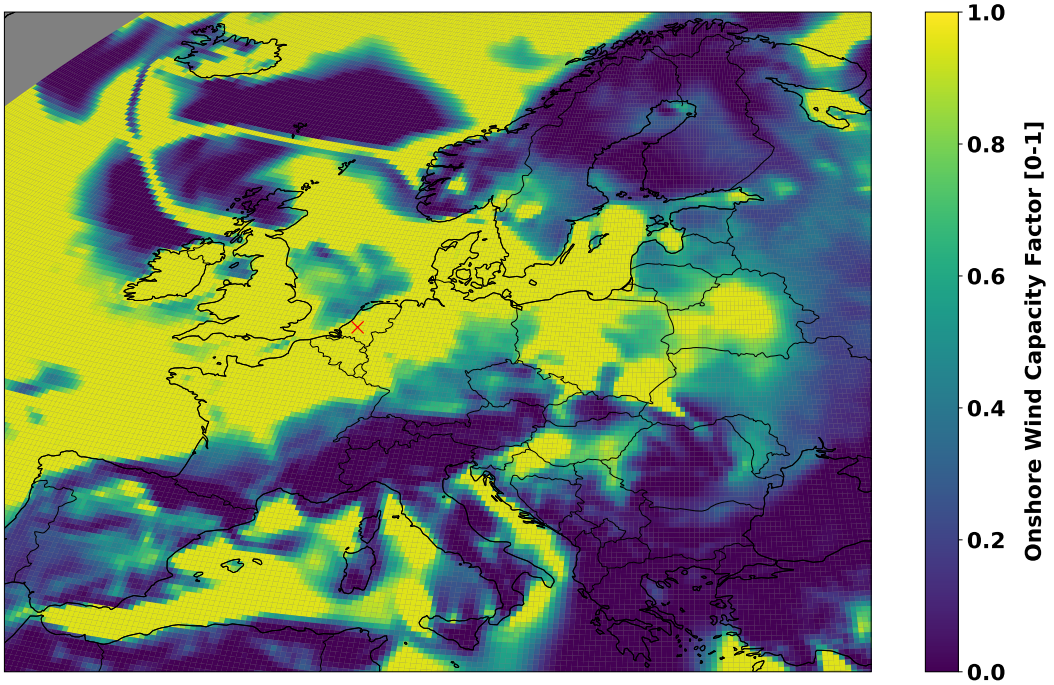
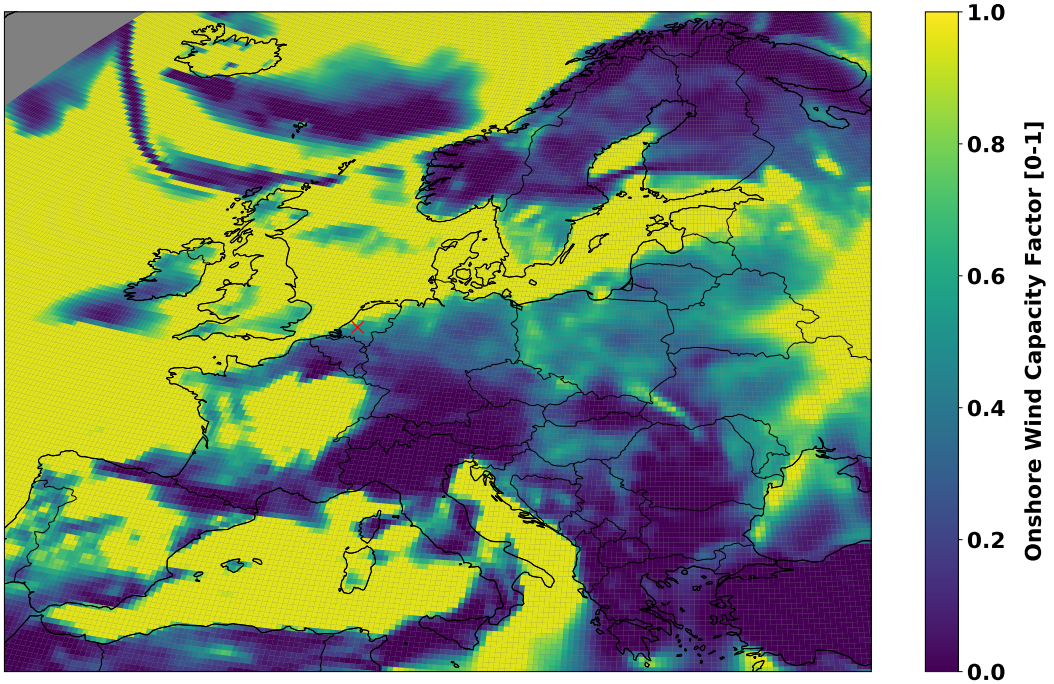


Figure 3.2: The region of interest with Utrecht marked in red, at coordinates (52,5). The colours represent the mean onshore wind capacity factor for the year of 2020, with the black lines representing national borders. It is evident that southern Europe tends to have much smaller onshore wind CF values for 2020.

are quite low, with no grid cells showing a solar CF value above 0.4. Solar CF values are more heavily affected the time of day as compared to wind; this effect is quite evident in Figure 3.7.



(a) 00:00 UTC on the 1st of March.



(b) 14:00 UTC on the 1st of March.

Figure 3.3: Onshore wind capacity factor values for Europe at different times on the 1st of March, 2020. Times are in Coordinated Universal Time (UTC).

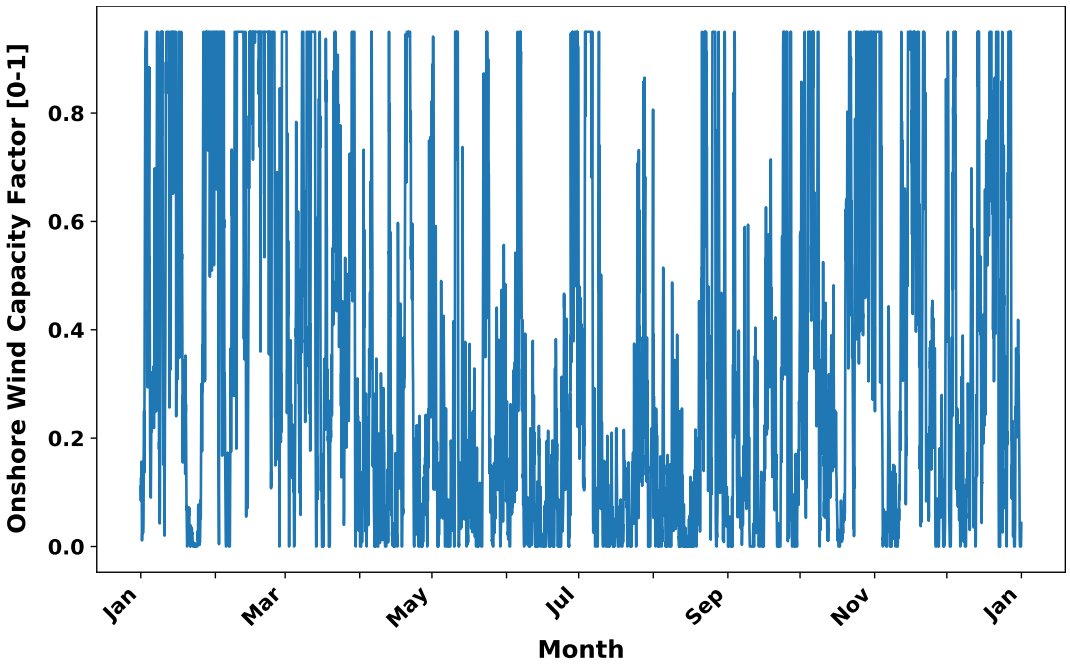
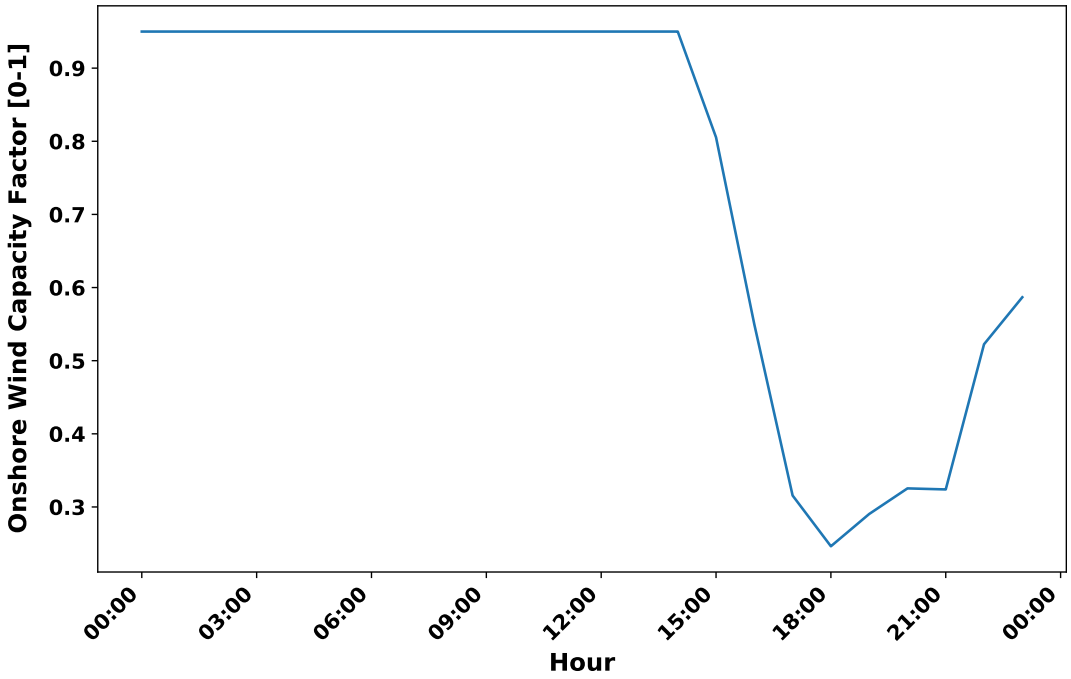
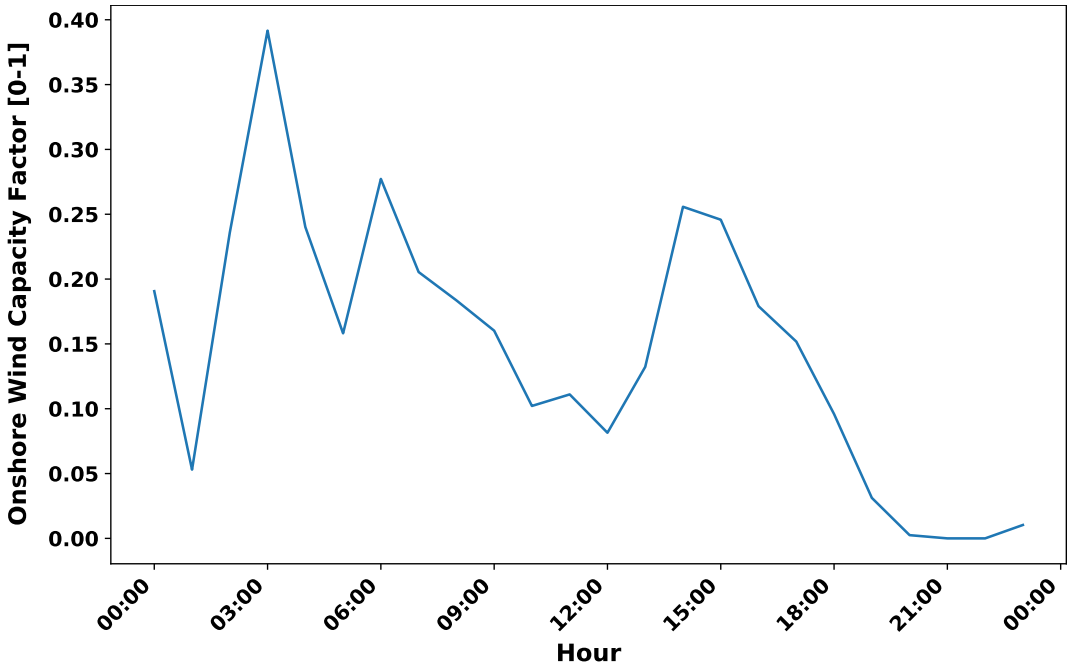


Figure 3.4: Onshore wind capacity factor over 2020 for Utrecht at (52,5). Note that the last x-axis label is for January 2021, but the data stops just short of that, at the 31st of December, 2020.



(a) Wind capacity factor over 1st March.



(b) Wind capacity factor over 1st August.

Figure 3.5: Onshore wind capacity factor of Utrecht at (52,5) shown over the 1st of March and the 1st of August. Note that the capacity factor scale is different for both as the maximum value in (b) is much smaller. All times are in Coordinated Universal Time (UTC).

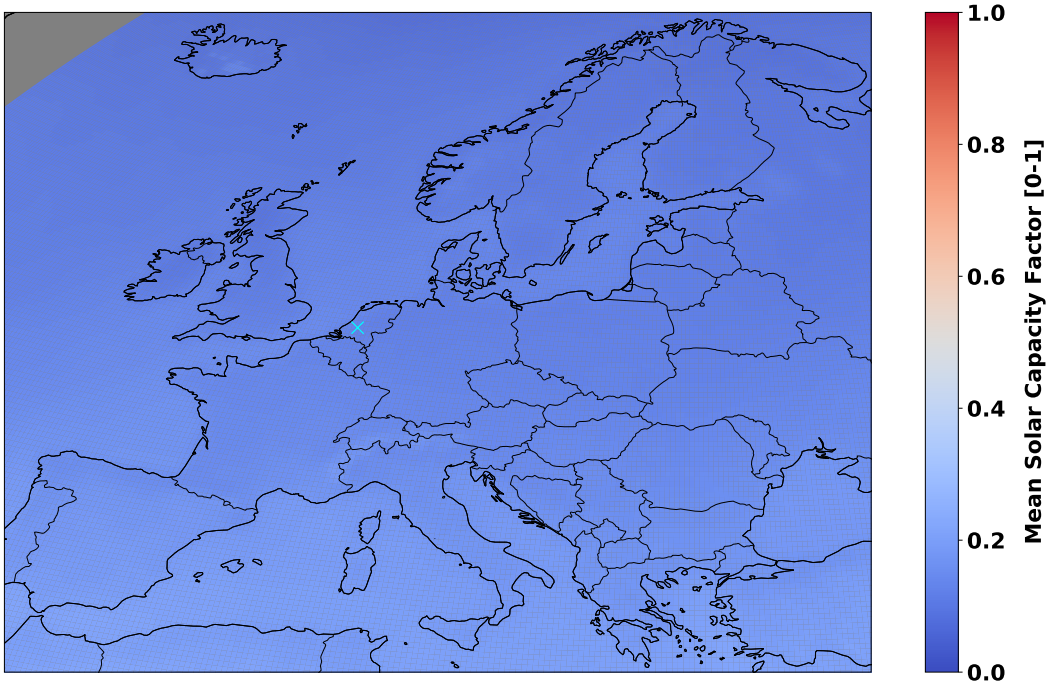
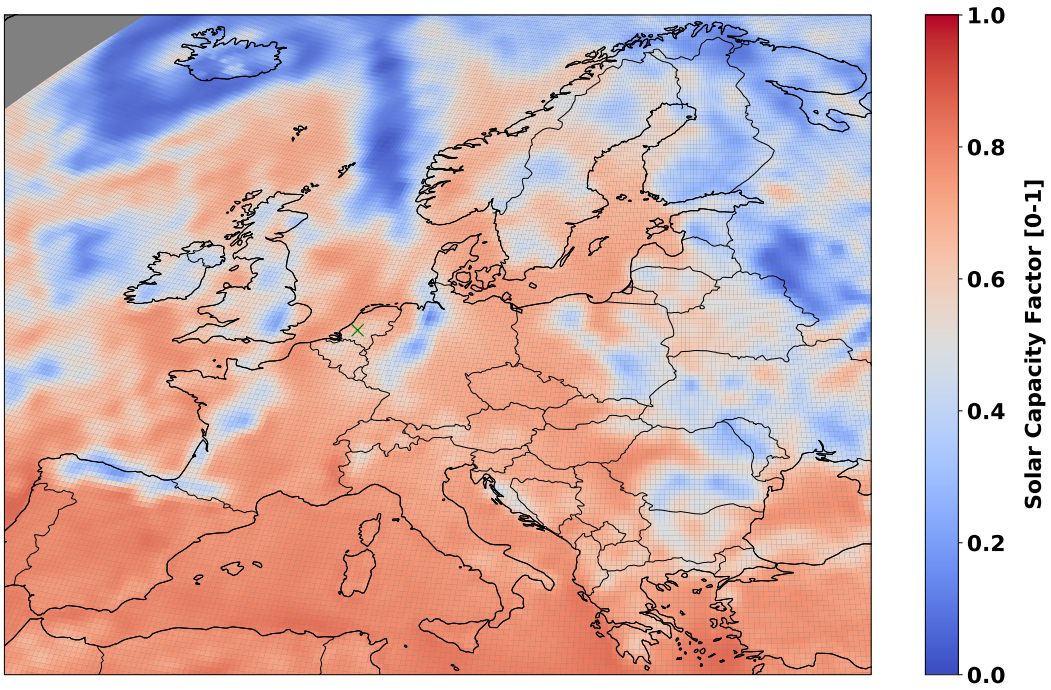
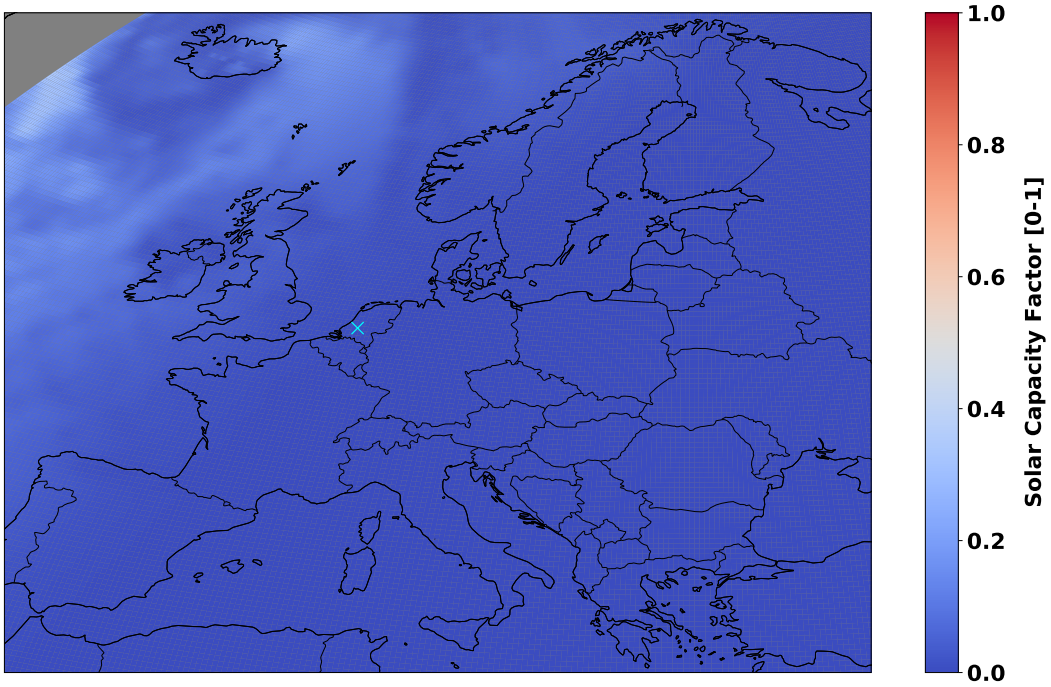


Figure 3.6: Mean solar capacity factor for Europe over 2020, with Utrecht marked with a cyan cross.



(a) Solar capacity factor on 1st August at 12:00 UTC.



(b) Solar capacity factor on 1st August at 20:00 UTC.

Figure 3.7: Mean Solar capacity factor values for Europe at two different times on 1st August, 2020. Utrecht is marked with a green cross in Figure 3.7a and a cyan cross in Figure 3.7b.

4. Method

To determine which regions are highly correlated, this study focuses on calculating the Correlation Halving Distance (CHD) metric efficiently. A higher than average CHD for a given location or origin cell indicates that the surrounding region is more strongly correlated in its potential wind or solar energy generation. By calculating the correlation on the capacity factor, a unitless number between 0 and 1 that represents the potential for renewable energy output, the metric is independent of the allocation of the actual wind and solar energy generators.

To estimate the CHD for any given grid cell taken as the origin cell, the following steps need to be carried out: correlation must be calculated for the time series at the origin with the time series contained in a subset of grid cells in the data. This subset is determined using an iterative sampling algorithm. The sample set of distance values and their corresponding correlation values are then used to fit a function, so that the distance at which correlation is 0.5 can be determined. This procedure is repeated for all desired origin cells to approximation the CHD for each grid cell.

In this chapter, the discussion of the method is organised as follows. Firstly, we present the mathematical definition of CHD, including the functional form that will be employed for the function fit. Subsequently, we discuss mathematical and technical considerations relevant to metric calculation and function fitting. Following this, we shift our focus to the experimental method, where we explore both synthetic and real-world data experiments, along with a discussion on generation of synthetic data. Lastly, the chapter concludes with an overview of iterative distance sampling algorithms, a discussion on stopping criteria for these algorithms, and required adaptations when applying them to real-world two-dimensional data rather than one-dimensional synthetic data.

4.1 Definition of Correlation Halving Distance

As mentioned above, a function must be fitted to estimate the distance at which the correlation is 0.5. The function that models how correlation changes with distance can take a number of forms. [15] lists the very many formulations put forward. One example of such a function is $\alpha e^{(-d/D)}$ where d is distance and α and D are adjustable parameters. This functional form was proposed by Justus and Mikhail [22]. A more recent example from the paper is $e^{(-\alpha d^{\beta})}$ proposed in 2014. The β exponent within the main exponent changes the shape from a negative exponential to a different one entirely. However, the majority of the functional forms presented are exponential decay functions of distance (d) with two parameters (α, D).

In addition to the theoretical backing, there is also empirical evidence for the exponential decay function form to be considered. In terms of real-world evidence, the exponential decay form has been discovered by [13], who plotted time series cross-correlation against distance for grid points found in various northern European countries in Figure 3 of their paper.

When an exponential decay function is used, the α parameter would need to be $\alpha \approx 1,359$ before the parameter D can be interpreted as the distance of 0.5 correlation. Therefore, it would be interesting to consider a function of correlation against distance for this study of the form $\rho = b^{-d/D}$, where ρ is correlation between two grid cells, b is some exponent base, d is the distance between the cells, and D is the Correlation Halving Distance.

Using the formulation above with base-2 means there is only one parameter that needs to be tuned, as well as making it easy to determine a Correlation halving Distance parameter estimate, D. Setting the base to a different number, like Euler's number, e, would mean that the interpretation of the Correlation Halving Distance parameter, D, is not straightforward, as it would correspond to a correlation of a different percentage.

To visualise this, correlation ρ can be set to 0 to examine changes in the formula for both base-2 and base-e. The base-2 formula exhibits the advantage where given that distance $d = D$ meaning that the distance from the origin

grid cell is equal to the CHD, then the expression simplifies to $2^{-1} = \frac{1}{2}$, clearly indicating that correlation is 0.5 at the CHD value. Generally, using an exponential function also provides a useful property in that setting distance as 0, indicating correlation of the origin cell with itself, results in correlation $2^0 = 1$.

In this study, we use the following function, where the correlation ($\rho(d)$) is assumed to be approximated by the negative exponential. It takes the form:

$$\rho(d) = 2^{-d/D} \tag{4.1}$$

where d is distance to a grid cell, and D is Correlation Halving Distance. Additionally, ρ is monotonically decreasing with respect to d. Note that as distance from a grid cell to another grid cell does not contain information about direction, it is assumed that the CHD value is same in all directions. An example of a function of this form fitted on 100 points is shown in Figure 4.1, showing a dashed line to estimate the CHD value, which is also the parameter D within the equation itself.

A drawback of the exponential formula structure is that it will never yield a correlation value that is negative for any distance, even if the actual correlation between the origin grid cell and a cell at that distance actually yields a negative correlation value. If a function is fit on mostly negative correlation values, this can be a problem in terms of accuracy and suitability of function form for function fitting.

Additionally, for the purposes of this experiment, the CHD value is expected to lie between 100 and 4000 km based on domain knowledge, so these will be the bounds used for sampled distances.

4.2 Further Mathematical Considerations

To carry out the CHD computation, the calculation method of pairs of time series as well as the details of the function fitting procedure must also be determined. To carry out the correlation calculation for time series pairs,

Pearson's R correlation will be used [23]. Let x and y represent two time series for which we want to calculate the correlation. The mean values of x and y are denoted as $\bar{x}$ and $\bar{y}$, respectively. The variables x_t and y_t represent the respective function values of x and y at a specific time index t. The time index t ranges from 1 to T, representing the total length of the time series. We assume that all time series are the same length. This is calculated using the following formula:

$$r(x,y) = \frac{\sum_{t=1}^{T}(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum_{t=1}^{T}(x_t - \bar{x})^2 \sum_{t=1}^{T}(y_t - \bar{y})^2}} \tag{4.2}$$

This formula is required for wind or solar time series correlation with time series at various locations from the origin, and a function has to be fitted on the correlation versus distance data. This function will be used to estimate the distance value where correlation is 0.5, and this distance value will be the CHD.

The actual method utilised for function fitting is the SciPy Python library's "curve_fit" function [24]. It takes the distance values and corresponding correlation values as inputs, fits the function, and returns an estimate for the lone parameter, $\hat{D}$, the CHD estimate, as well as the estimate's variance, $\hat{\sigma}^2_{\hat{D}}$. The value for the $\hat{D}$ estimate is obtained by minimising the sum of squared errors function, $S(D)$, which is given as follows:

$$S(D) = \sum_{n=1}^{N}(\rho_n - 2^{\frac{-d_n}{D}})^2 \tag{4.3}$$

The estimated variance of $\hat{D}$ using a first-order approximation can be used to derive the standard deviation of the estimate, $\hat{\sigma}_{\hat{D}}$, by applying a square root operation. The formula for $\hat{\sigma}_{\hat{D}}$ is as follows:

$$\hat{\sigma}_{\hat{D}} = \sqrt{\frac{\sum_{n=1}^{N}(\rho_n - \hat{\rho}_n)^2/N - 1}{\sum_{n=1}^{N}(\frac{ln(2)d_n\hat{\rho}_n}{\hat{D}^2})^2}} \tag{4.4}$$

The standard deviation of the estimate will be of use as a stopping criterion in the result validation phase.

To decide which subset of grid cells will be used for the correlation calculation with the origin cell, a sampling strategy must be utilised. To reduce computational complexity without losing vital information, informative sampling approaches need to be considered. Many possible sampling strategies can be used, ranging from the most uninformed that take random samples, to more adaptive ones that select the next sample based on what has already been sampled. This study involves experimenting with multiple sampling approaches on both theoretical and real-world data. Experiments were carried out on onshore wind CF and solar CF data separately, as both categories can yield separate datasets of time series but with the same region of interest covered in the spatial domain. Off-shore wind CF is ignored as it is structurally quite similar to onshore wind CF.

4.3 Experimental Method

Two sets of experiments are carried out. In the first set, multiple sampling algorithms are tested on synthetic data that is generated using Equation 4.1 with noise added to the result. Optimal stopping criteria is also determined so that the algorithm does not run for more iterations than are necessary.

The second set of experiments involve applying the best-performing algorithm from the first set of experiments to real-world data. Additionally, the results of this algorithm will be analysed using validation checks and examined for improvements, as well as compared with the brute force CHD calculation approach to determine efficiency.

4.3.1 Synthetic Data Generation

Synthetic data is used to perform theoretical comparisons on the algorithms, as the correlation values on the real data are not known in advance. The synthetic correlation value for each distance is generated d using $\rho = 2^{\frac{-d}{D}} + \epsilon$, where ϵ is a simulated noise value that is added to the correlation. The noise is generated by taking a random sample from a normal distribution centred around mean 0 and standard deviation 0.15. An example of the synthetic data generated for the function fitting as well the actual curve fitted on this data is shown in Figure 4.1. Note that for the purposes of the experiment on synthetic data, correlation is allowed to go above 1 as the function fit is being tested and the true meaning of the correlation metric should not affect the fit.

The value of 0.15 for standard deviation in correlation was selected as values smaller than this adhere too closely to the negative exponential line, and values bigger result in spread that is much higher and may cause the negative exponential shape to be lost entirely due to high levels of noise.

Note that noise levels in correlation will vary throughout the real data, but one value was selected for theoretical experiments to maintain a level of consistency.

For each sampling algorithm in the theoretical experiments, a distance value d is generated, and its correlation is calculated using $\rho = 2^{\frac{-d}{D}}$, with the true D set as 400 km. After the ρ value is calculated using the aforementioned function, it has noise added to its value to attempt to mimic the real-world noisiness in correlation values.

To generate noise values in the correlation for each kilometre of distance, a hundred lists of normal distributions centred around mean 0 and standard deviation 0.15 are created. Each list contains 3900 values, corresponding to each kilometre in the range [100,4000]. When an algorithm selects a distance value to sample, it is rounded to the nearest kilometre to retrieve a noise value from the first list. If the same kilometre is selected a second time by the same algorithm, the noise value from the second list is retrieved, and so on. If the same kilometre value has been selected more than one hundred times, the list

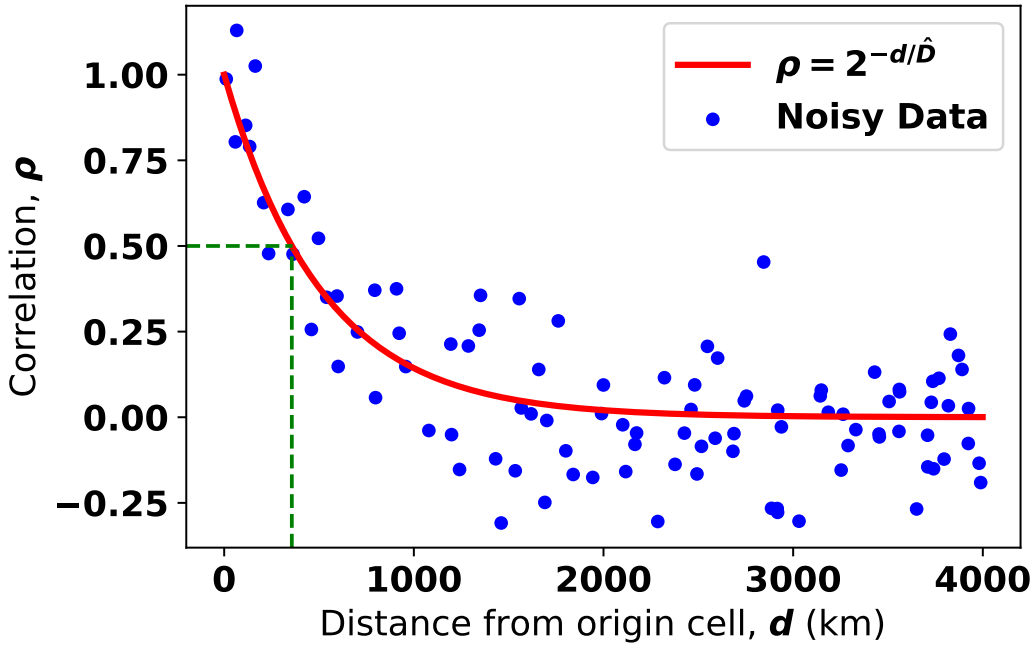


Figure 4.1: A hundred points of data generated using the function $\rho = 2^{\frac{-d}{D}}$ with noise added to the result. The red line shows the function fitted to the data points, with the value for the estimate $\hat{D} = 357$, shown with the green dashed line as the distance value where correlation is 0.5.

counter for that kilometre will cycle back and the first noise list will be used again. A count check will be utilised in each synthetic algorithm test to make sure the same distance value is not being over-sampled very frequently.

A PCG64 random number generator is used to generate the correlation noise lists. The generation of the random noise lists is seeded based on the run number of the algorithm. For example, on run number 5, the seed for the 1000 normal distribution lists will be 5. As each algorithm experiment is receiving the same sequence of seeds, the sequence of noise values generated is kept as consistent as possible.

4.3.2 Synthetic Data Experiments

To account for the potential variation induced by randomness in individual algorithm runs, each algorithm is run with synthetic data one thousand times, with one thousand iterations per run. This approach allows for a comprehensive analysis of multiple aspects of the algorithm's performance, providing a more complete understanding of both average and outlier performance results.

To visually inspect results of these runs across different algorithms, the maximum, minimum, median, first quartile, third quartile, tenth percentile, and ninetieth percentile of $\hat{D}$ across all one thousand runs of each sampling algorithm are plotted to allow for visual comparisons.

A further one thousand runs of each algorithm will be conducted with the synthetic data to test stopping criteria to avoid running for one thousand iterations. The algorithm showing the most favourable results in both of these tests will be applied to the real world data for the next round of experiments.

4.3.3 Real Data Experiments

After the theoretical comparisons have been performed, the best-performing algorithm with the preferred stopping criteria will be selected to test on the real data.

In the dataset, each grid cell on the part of the Earth that is covered is represented by its centre coordinates by latitude and longitude value. Various (latitude,longitude) points on the dataset will be taken as the origin grid cells, and the Correlation Halving Distance will be determined for each. Utrecht's CHD value will be investigated, as well as CHD for cities such as Zurich and Madrid, to allow investigating the robustness of the algorithm in very different geographies. The dataset may not contain the exact latitude and longitude of each city's center, so the nearest point that exists will be used as the representative coordinate.

Utrecht will be the first origin cell to test. For Utrecht specifically, the Correlation Halving Distance of two (latitude, longitude) origin pairs will be investigated. This is to ensure that the Correlation Halving Distance of two cells adjacent to each other should be very similar as well, which serves as a validation check for the behaviour of the algorithm. Other cities will be selected as origin cells, and the sampling algorithm will be run 10 times for each origin, and Correlation Halving Distance as well as other metrics will be recorded.

Additionally, the sampling approach will be run one hundred times with random origin cells each run. The restriction on which coordinates to select as

origin cell are that they must be far from the original bounds of the dataset, focusing mainly on Europe. Figure 3.1 shows the restricted area surrounded by the red bounding box where the random origin cells may be sampled from. The restricted latitude bounds are 36.25 to 69.25 ° North, and the longitude bounds are -6.75 to 32.5 ° East. The restricted range is to ensure that the origin cells are not near the edges of the dataset, as if the sampling algorithm attempts to sample a location out of range, this may lead to less predictable results as the closest location that exists in the dataset will be sampled, which may be at a distance quite far from the actual distance that it was supposed to be sampled at.

In this round of experiments, the direction between grid cells must also be taken into account, as the real-world data is now two-dimensional as opposed to one-dimensional distances with the synthetic data. In the theoretical experiment stage, one could generate synthetic values for correlation by feeding a one-dimensional value for distance between grid cells, but this did not take direction into account. In the real dataset, sampling a grid cell that is x km away means considering an infinite number of directions, and not necessarily only directions in a straight line, as the spatial grid of cells is affected by the curvature of the Earth. This means that a method to choose direction must also be established, and will be discussed following the overview of the iterative sampling algorithms.

4.3.4 Data Type

The dataset contains a time series at each (latitude,longitude) location, and each hourly data point in that series contains three values: solar capacity factor, wind onshore capacity factor, and wind offshore capacity factor. These can be split into three different time series for each location, yielding three separate datasets. This study focuses mostly on using the wind onshore capacity factor and solar capacity factor datasets.

In the event that one of the time series in the correlation calculation contains the value "0" throughout, the correlation calculation will result in the correlation value "NaN" or Not a Number, due to a divide by 0 error generated

by NumPy in the correlation calculation. These values cannot be used to fit the Correlation Halving Distance function. In that case, this location, along with its corresponding distance and invalid correlation value, will be removed from the dataset.

4.3.5 Solar Data Considerations

The time series for solar CF can cause issues with correlation calculations, as the time series curve is rapidly oscillating towards and away from 0 due to the CF being 0 between sunset and sunrise. This is expected to affect the performance of the sampling algorithm. To apply some preprocessing to the data to alleviate this problem, the maximum value of solar capacity factor for each hour is used.

4.3.6 Randomness in Direction

Due to a random direction being taken for each sample distance, running the algorithm with a different randomness seed will potentially generate a different set of sampled locations each time, even if the distances sampled are the same. For example, a point taken 50 km north may have a high correlation with the origin cell, but a point taken 50 km south may have a low correlation with the origin cell. The potential bias the variability in direction necessitates repeating the sampling method a number of times, and taking the mean of the metrics to investigate the "average" behaviour of the algorithm for a given origin cell. To investigate this variability, for each city, the algorithm for determining CHD will be run 10 different times.

4.3.7 Validation Method on Real Data

After the Correlation Halving Distance estimate $\hat{D}$ km for a given centre cell has been determined, 360 cells $\hat{D}$ km from the centre are taken, in a circle around the origin cell. The time series correlation between the origin cell and each cell in the circle is calculated, and the mean value of these correlations is taken. The mean correlation at the Correlation Halving Distance should be 0.5 for a successful approach. This is called the CHD check.

To further investigate potential "bounds" for the Correlation Halving Distance value, the standard deviation value of the $\hat{D}$ estimate, $\hat{\sigma}_{\hat{D}}$, will be utilised. 360 cells that are $(\hat{D}-2\hat{\sigma}_{\hat{D}})$ km from the origin cell will be taken for correlation calculations with the origin cell, and the mean correlation value of these will be calculated. This mean value should be greater than 0.5. This is known as the small circle check.

Additionally, 360 cells $(\hat{D}+2\hat{\sigma}_{\hat{D}})$ km from the centre cell will also be taken and the same procedure will be repeated. The mean correlation value for these cells must be smaller than 0.5. This is known as the big circle check.

4.4 Overview of Sampling Algorithms

To sample grid cells for their respective CHD calculation, there are many different possible sampling strategies. This section includes an overview of the different sampling algorithms that were tested in this study, along with pseudocode and supplementary explanations wherever required. Note that all algorithms are iterative procedures that take four sample points per iteration. As the first algorithm in the list, ABS, is designed around taking four samples, the others were also adjusted for four to keep performance comparisons as fair as possible.

4.4.1 Adjusted Binary Search (ABS)

To evaluate a more deterministic sampling approach, a novel sampling strategy is utilised, called Adjusted Binary Search (ABS). The estimate for Correlation Halving Distance $(\hat{D})$ is assumed to be between 100 km and 4000 km, based on findings in past research [13]. Each iteration of this algorithm will sample two points closer to one of the bounds and one point closer to the other bound. Deciding which bound generates two samples and which produces one is based on the comparison of the latest $\hat{D}$ value with the previous one.

If $\hat{D}_i < \hat{D}_{i-1}$ in the current iteration i, two points are sampled from distances between $\hat{D}_i$ and the lower bound, 100 km, and one point is sampled from between $\hat{D}_i$ and the upper bound, 1000 km. If $\hat{D}_i > \hat{D}_{i-1}$, two points are

sampled from distances between $\hat{D}_i$ and the upper bound, 1000 km, and one point is sampled from between $\hat{D}_i$ and the lower bound, 100 km.

If one point is to be sampled between $\hat{D}_i$ and either bound, the arithmetic mean is calculated to generated a midpoint distance. If two points are to be sampled, the first and third quartiles are taken. The pseudo-code of the complete procedure is given in Algorithm 1.

In the second version of this algorithm, an additional sample point is incorporated: in every iteration, the midpoint between the current $\hat{D}_i$ and previous estimate of $\hat{D}_{i-1}$ is also included. Additionally, for each distance sampled, a small noise value sampled from a normal distribution centred around mean 0 and standard deviation $(1000 - 100)/4$, to provide some noise in the distance sample. This is done as having a very deterministic algorithm introduces some issues in the theoretical experiments, especially if the distances sampled are in a very limited range. Due to the changes in code being very minor, the pseudo-code for the second version is not provided.

The second version is used in the synthetic data experiments as it showed better results in preliminary testing. It takes one extra point per iteration compared to the first version, so there is an increased growth in sample size per iteration.

4.4.2 Random sampling (R)

Each iteration, four distance value samples are randomly taken from the range [100,4000] to fit the function each time. A uniform distribution is used to generate the distance samples so that each possible distance value has an equal chance of being generated. There is no adaptability in this algorithm and previous sample points do not affect what is sampled in the next iteration.

4.4.3 Systematic Random (SR)

Each iteration, four samples are taken at regular intervals, with some noise added as well. To achieve this, the "step size" is calculated using $(4000 - 100)/4 = 975$. Then 4 points are generated using the expression $(100 + i * step)$

Algorithm 1 Adjusted Binary Search (Version 1)

Input: $CurrentGridCell$, $MinDistance$, $MaxDistance$

$\hat{D}, S \leftarrow empty$ ▷ Initialise $\hat{D}$ and std. deviation arrays
$l \leftarrow MinDistance$ ▷ 100
$u \leftarrow MaxDistance$ ▷ 4000

$\rho_l \leftarrow \texttt{correlation}(CurrentGridCell, l)$ ▷ Or sampled as $2^{-l/400} + noise$
$\rho_u \leftarrow \texttt{correlation}(CurrentGridCell, u)$ ▷ Or sampled as $2^{-u/400} + noise$
$X \leftarrow [u, l]$
$Y \leftarrow [\rho_l, \rho_u]$
$\hat{D}, \mathrm{S} \leftarrow \texttt{curve\_fit}(function = 2^{-d/D}, data = (X, Y))$ ▷ Fit function

$m1 \leftarrow 0.5 * (l + \hat{D})$
$m2 \leftarrow 0.5 * (\hat{D} + u)$
$\rho_{m1} \leftarrow \texttt{correlation}(CurrentGridCell, m_1)$
$\rho_{m1} \leftarrow \texttt{correlation}(CurrentGridCell, m_2)$
$[X, Y].\texttt{append}([m1, \rho_{m1}])$
$[X, Y].\texttt{append}([m1, \rho_{m2}])$

for $i \leftarrow 1, 1000$ **do** ▷ Run for max. 1000 iterations

  if $\hat{D}_i < \hat{D}_{i-1}$ **then** ▷ if new $\hat{D}$ estimate is smaller
    $q1 \leftarrow 0.25 * (l + \hat{D}_i)$
    $\rho_{q1} \leftarrow \texttt{correlation}(CurrentGridCell, q1)$
    $q3 \leftarrow 0.75 * (u + \hat{D}_i)$
    $\rho_{q3} \leftarrow \texttt{correlation}(CurrentGridCell, q3)$
    $m1 \leftarrow 0.5 * (u + \hat{D}_i)$
    $\rho_{m1} \leftarrow \texttt{correlation}(CurrentGridCell, m1)$
  else
    $m1 \leftarrow 0.5 * (l + \hat{D}_i)$
    $\rho_{m1} \leftarrow \texttt{correlation}(CurrentGridCell, m1)$
    $q1 \leftarrow 0.25 * (u + \hat{D}_i)$
    $\rho_{q1} \leftarrow \texttt{correlation}(CurrentGridCell, q1)$
    $q3 \leftarrow 0.75 * (l + \hat{D}_i)$
    $\rho_{q3} \leftarrow \texttt{correlation}(CurrentGridCell, q3)$
  end if

  $[X, Y].\texttt{append}([q1, \rho_{q1}])$ ▷ Update arrays
  $[X, Y].\texttt{append}([q3, \rho_{q3}])$
  $[X, Y].\texttt{append}([m1, \rho_{m1}])$
  $\hat{D}, \mathrm{S} \leftarrow \texttt{curve\_fit}(function = 2^{-d/D}, data = (X, Y))$ ▷ Fit function
end for ▷ Algorithm ends

where $i = 0, 1, 2, 3$ to obtain four different distance values. Additionally, each point is given a random error drawn from a uniform distribution where the range of values is equal to the step size. This ensures that unlike random sampling, all 4 points each iteration do not cluster in one region every iteration. It makes sure that generated points cover the whole range between 100 and 4000 km, inclusive.

4.4.4 Greedy (G)

Greedy sampling will take the midpoint of the largest distance between a pair of points already in the sample set, breaking ties randomly if more than one distance qualifies as the largest. For example, if we start by including the bounds [100,4000] as the first two distances in the sample set, the next point to sample will be 2050 km. Now with the list containing [100,2050,4000], there are two pairs (100,2050) and (2050,4000) with the maximum distance between them. One of these pairs will be selected randomly for the next sample, and the other pair will be selected for the sample after that one. This procedure repeats four times to obtain four points per iteration. It is prone to slowing down especially as the number of maximal distances to check increases as more and more samples are taken.

4.4.5 Uncertainty Sampling (U)

This method is an approach similar to query-by-committee, where the next point to be sampled is the one the query of learners disagree the most on [18]. In this case, the shape of the function to be fitted is already known to be negative exponential of the form $\rho = 2^{\frac{-d}{D}}$, and fitting it using $curve\_fit$ gives a $\hat{D}$ estimate and its standard deviation, $\hat{\sigma}_{\hat{D}}$. These estimates can be used to generate a confidence interval [a,b] for $\hat{D}$. The values for the interval bounds are given by $[a = \hat{D} - sf \times \hat{\sigma}_{\hat{D}}, b = \hat{D} + sf \times \hat{\sigma}_{\hat{D}}]$, where sf is scale factor. The committee of learners can be modelled by two functions given by substituting the interval bounds in the correlation function, giving $\rho = 2^{\frac{-d}{a}}$ and $\rho = 2^{\frac{-d}{b}}$. The difference between these two equations is $2^{\frac{-d}{b}} - 2^{\frac{-d}{a}}$, which is a third function on its own. The "disagreement" between a committee can

be represented by the maximum difference between the curves, given by the derivative of the difference with respect to d, yielding the formula:

$$d = -\frac{a \times b \times \ln(\frac{a}{b})}{(\ln 2) \times (b - a)} \tag{4.5}$$

As an example to visualise how this procedure works in generating a d value to sample, we introduce the example of two curves, $\rho = 2^{-d/a}$, where $a = 2$, and $\rho = 2^{-d/b}$, where $b = 5$. Plugging the a and b values into Equation 4.5, the result $d = 4.404$ is computed. Figure 4.2 shows the respective decay curves of a and b as well as their difference curve. The gray dashed line is the line where the x-value is 4.404, as computed earlier.

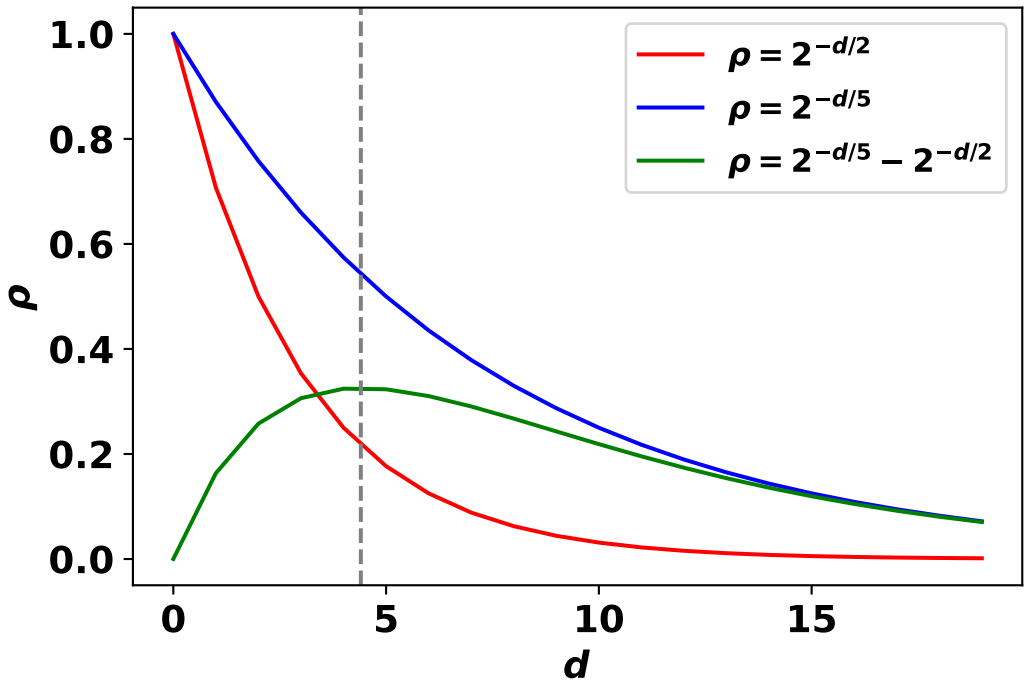


Figure 4.2: Figure showing two exponential decay curves plotted in red and blue, with their difference curve plotted in green. The gray dashed line is the maximal difference between the red and blue curves, as well as the global maximum of the difference curve.

As described, Equation 4.5 yields the next distance value d to sample. As four sample points are to be taken per iteration, the confidence interval [a,b] can be varied by varying the value of the scale factor, sf, so four different

distances can be computed per iteration. For this study, the scale factor values are 1.5,2,2.5, and 3.

Note that the logarithm of a negative number is undefined, so in the case that a is negative, a run-time error will be generated. This can occur if the $\hat{\sigma}$ value is too large, for example. In preliminary testing when determining choice of scale factor, this often occurred when the scale factor is 4 or above, and/or the standard deviation value is significantly larger than the $\hat{D}$ estimate. In this case, to prevent an algorithm crash as well as take a sample point, a random distance value is drawn from the range [100,4000] instead, and this value is used as the next sample to be included. Preliminary testing showed the switch to a random point did not occur very frequently in experimental runs.

Before the iterative part of the algorithm, the pseudo-code for this procedure is shown as Algorithm 2. In lines 1 to 11, the lower bound, 100, and upper bound, 4000, as well as their midpoint, 2050, are used as starting distance values for the sample set, with corresponding calculated correlations values. This data is used to fit the function once before the iterative portion of the algorithm begins, so that initial estimates for $\hat{D}$ and $\hat{\sigma}$ are available to calculate a and b.

In line 16, the iterative procedure starts with the i loop: a scale factor is initialised as 1.5, and will be incremented by 0.5 each iteration, so that b and a can be calculated four times with four different scale factors, and yield four distance samples.

Another inner j loop begins in line 19 to generate four points where the scale factor increases by 0.5 each iteration. Lines 20 and 21 are where a and b are initialised. Following these is a check on the sign of a: if a is negative, take a random distance in the range [100,4000] instead as the next sample. The random distance sampling procedure is also repeated to take more random points in case a is negative on the first, second, or third point: it follows that increasing the scale factor value will yield the following a in to be negative in the following iteration of the j loop as well. For example, if the 1st a calculation with scale factor 1.5 yields a negative a, it is assumed that a will be negative even for bigger scale factor values, as the second term in $\hat{D}_{n-1} - sf * S_{n-1}$ will

only get larger. In this case, all four distances will be random samples. In contrast, if only the fourth distance to sample yields a negative a, only one random sample is taken that iteration.

After the a sign check has been performed, the value for d is calculated in line 30, using Equation 4.5, and its corresponding correlation is calculated. The scale factor is incremented by 0.5 in line 33, and the next iteration of the i loop follows.

Uncertainty Sampling with Systematic Random (USR)

This algorithm performs uncertainty sampling until 25 samples have been taken. This switching criterion value was determined partially by preliminary experiments with U sampling, as it seemed that the values 562 to 597 km are sampled over a hundred times over one thousand runs. This range is 35 km, but a smaller and neater value was chosen as the switching criteria.

Uncertainty Four Out Of Five (U5)

This algorithm switches from U sampling to SR every fifth iteration, therefore still using the U strategy the majority of the time.

Uncertainty with Adjusted Binary Search (UA)

This uses a similar approach to USR, but instead of switching to SR at 25 samples, the algorithm switches to ABS instead.

4.4.6 Overview of Stopping Criteria

To avoid having to run the algorithms for a full one thousand iterations and attempt to minimise total number of samples taken, some stopping criteria have to be tested for each algorithm.

As an estimate of $\hat{D}$ and the standard deviation of that estimate are available with the function fit each iteration, it is useful to consider both of these parameters when designing stopping criteria to test.

Algorithm 2 Uncertainty Sampling

Input: $CurrentGridCell$, $MinDistance$, $MaxDistance$

$\hat{D}, S \leftarrow empty$ ▷ Initialise $\hat{D}$ and std. deviation arrays
$l \leftarrow MinDistance$
$u \leftarrow MaxDistance$

$\rho_l \leftarrow \texttt{correlation}(CurrentGridCell, l)$ ▷ Or sampled as $2^{-l/400} + noise$
$\rho_u \leftarrow \texttt{correlation}(CurrentGridCell, u)$ ▷ Or sampled as $2^{-u/400} + noise$
$X \leftarrow [u, l]$
$Y \leftarrow [\rho_l, \rho_u]$
$q2 \leftarrow 0.5 * (l + u)$
$\rho_{m1} \leftarrow \texttt{correlation}(CurrentGridCell, q_2)$
$[X, Y].\texttt{append}([q2, \rho_{q2}])$
$\hat{D}, \mathrm{S} \leftarrow \texttt{curve\_fit}(function = 2^{-d/D}, data = (X, Y))$

for $i \leftarrow 1, 1000$ **do**

▷ (This blank line is where switching criteria may be checked for USR, UA, and U5 algorithms.)

$sf \leftarrow 1.5$ ▷ Set scale factor
for $j \leftarrow 0, 4$ **do** ▷ Calculate bounds with different s.f. per point
$b \leftarrow \hat{D}_{i-1} + sf * S_{i-1}$
$a \leftarrow \hat{D}_{i-1} - sf * S_{i-1}$

if $a < 0$ **then** ▷ if a is negative, take random point instead
for $k \leftarrow 0, 4 - j$ **do** ▷ Take as many random points as needed.
$d \leftarrow \texttt{randGenerator.uniform}(l, u)$
$\rho_d \leftarrow \texttt{correlation}(CurrentGridCell, d)$
$[X, Y].\texttt{append}([d, \rho_d])$
end for
`break` ▷ End j's loop after 4 points are taken
end if

$d \leftarrow -(a * b * log(a/b))/(log(2) * (b - a))$ ▷ a is non-negative
$\rho_d \leftarrow \texttt{correlation}(CurrentGridCell, d)$
$[X, Y].\texttt{append}([d, \rho_d])$
$sf \leftarrow sf + 0.5$ ▷ increment scale factor for next sampled point
end for
$\hat{D}, \mathrm{S} \leftarrow \texttt{curve\_fit}(function = 2^{-d/D}, data = (X, Y))$

end for

A hundred iterations

As a starting point, one hundred iterations were used as the stopping criterion for each algorithm. However, this means around four hundred samples will be taken per algorithm, an already large amount. The criterion itself is also rather arbitrary and not adaptive to the actual behaviour of any algorithm.

Standard Deviation, $\hat{\sigma}_{\hat{D}}$

To use this as a stopping criterion, a threshold can be decided to stop the algorithm iteration when the $\hat{\sigma}_{\hat{D}}$ value generated from a function fit is below the threshold. For this research, we have considered the standard deviation thresholds 25 km, 17 km and 10 km, called "std25", "std17" and "std10", respectively. All three will be tested separately as three different stopping criteria.

Difference in $\hat{D}$

This is the absolute difference of the $\hat{D}$ estimate from its previous iteration estimate. For the threshold, 3 km was decided. However, from some preliminary algorithm runs, the value of $\hat{D}$ is expected to vary erratically until enough samples are taken. Therefore, the expression $\mid \hat{D}_{current} - \hat{D}_{previous} \mid < 3$ is checked for at least three consecutive iterations before stopping the algorithm run. Throughout this paper, it will be referred to as "absdiff" for conciseness.

Combination

For the final criterion, the algorithm checks if 10 iterations have passed. This is to set a minimum threshold of at least 40 samples before the algorithm stops, due to the assumption that the algorithm stopping when taking too few samples may generate a very bad $\hat{D}$ estimate. After this is checked, the algorithm may stop if $\hat{\sigma}_{\hat{D}}$ is less than 25 and the $\hat{D}$ estimate difference is smaller than 3 for at least three consective iterations. These combined criteria will be referred to as "triple" for conciseness.

4.4.7 Direction Considerations

For any of these algorithms applied on the real data, a method to choose direction as well as distance must be determined, as simply having a distance value is not enough to sample a single grid cell on the real-world data. The GeoPy Python library allows for geodesic distances between two (latitude,longitude) coordinates on the Earth to be calculated [25]. For a given sampling algorithm, a random angle is chosen each time a distance is sampled. If this results in a (latitude,longitude) coordinate that already exists in the sample set, the random angle is resampled and a new coordinate is found. If a coordinate is found that does not exist in the dataset, the nearest point that does exist in the dataset will be taken as the point to be sampled, with nearest being determined by XArray's built-in library function that utilises Euclidean distance.

Additionally, if all possible coordinates at a given distance are already included in the sample set, a counter will ensure that an infinite loop does not result by regenerating angles when all 360 angles have been used up to include corresponding locations in the dataset. If the same distance is attempted to generate a location not already in the sample set more than 360 times, it is assumed there are no more possible coordinates at that distance that are not already sampled, so a random distance sample in the range (100,4000) is taken in that case, and a random angle for that distance as well. After the actual location has been sampled successfully, the distance d is updated by calculating the distance between the actual location and the origin to ensure accuracy, before d is added to the sample set for function fitting. The pseudocode for this subroutine is shown in Algorithm 3.

Algorithm 3 Subroutine to Determine Location

Input: $distance\_value, origin\_cell\_latitude, origin\_cell\_longitude, locs$

```
d ← distance_value
lat ← origin_cell_latitude
lon ← origin_cell_longitude
angle ← random.uniform(0, 360)                          ▷ take random direction
loc_d ← getLoc(lat, lon, d, angle)   ▷ get coordinates for location sample
counter ← 0
while loc_d is in locs do          ▷ if location has already been sampled
    angle ← random.uniform(0, 360)
    loc_d ← getLoc(lat, lon, d, angle)                  ▷ Re-sample with new angle
    counter ← counter + 1
    if counter > 360 then   ▷ take random distance if stuck in while loop
        d ← random.uniform(LowerBound, UpperBound)
        loc_d ← getLoc(lat, lon, d, angle)  ▷ Re-sample with new distance
    end if
end while
distance_value ← getDistance(loc_d,(lat,lon))
 ▷ Re-calculate the distance between origin and the actual location found
    return
```

5. Results

The following sections show results of the sampling algorithm comparison experiments on synthetic data, followed by the results of the application of Uncertainty sampling experiments on real-world data.

5.1 Synthetic Data Experimental Results

Here, we present the results of comparison of the sampling algorithms presented in Section 4.4 applied to the synthetic data described in Section 4.3.1.

5.1.1 Comparison of CHD Estimate

After each algorithm was run 1000 times for 1000 iterations each run, the final $\hat{D}$ estimate for each run was recorded. Figure 5.1 shows various metrics regarding how the $\hat{D}$ estimate changes every iteration across 1000 runs of ABS and U, including the median value as well as the maximum and minimum values. For results regarding the other algorithms, those are available in Section A.2 of the appendix.

The median value for all algorithms appears to converge to the true value fairly quickly, and the maximum and values are extremes as the 90th percentile lines are much closer to the median. The same goes for the minimum being extreme and the 10th percentile being closer to the median as well.

There are noticeable differences in the other metrics. U appears to have a smaller range of values as shown by the min and max lines in Figure 5.1b. The figure also shows that the maximum and minimum $\hat{D}$ estimates of U converge and stabilise faster.

Most algorithms completed execution in around the same time, but one algorithm stood out in this regard. Greedy Sampling additionally takes considerably more time to execute, as it took about 25 minutes to complete 1000

iterations, and every other algorithm took around 10 minutes.

5.1.2 Standard Deviation Comparison

Each algorithm's median standard deviation for its $\hat{D}$ estimates, $\hat{\sigma}_{\hat{D}}$, was recorded after 1000 runs of each algorithm. Figure 5.3 shows the results, excluding the U-based hybrid algorithms USR, U5 and UA. The baseline algorithm for this comparison is taken as SR, and it is clear that both G and R are doing worse in terms of the speed of $\hat{\sigma}_{\hat{D}}$ decrease over time. For this reason, R and G will no longer be considered for stopping criteria testing.

Figure 5.4 shows the median standard deviation over time for algorithms that are better than baseline SR, with SR also included. All Uncertainty-based algorithms are shown in purple with different line styles, and all perform better, or just as good as, SR in terms of speed of median $\hat{\sigma}_{\hat{D}}$ drop. Figure 5.5 provides a closer look to how the Uncertainty-based algorithms start off with median standard deviation dropping quite fast. The hybrid approaches USR and UA start approaching their corresponding SR and ABS deviation lines over time.

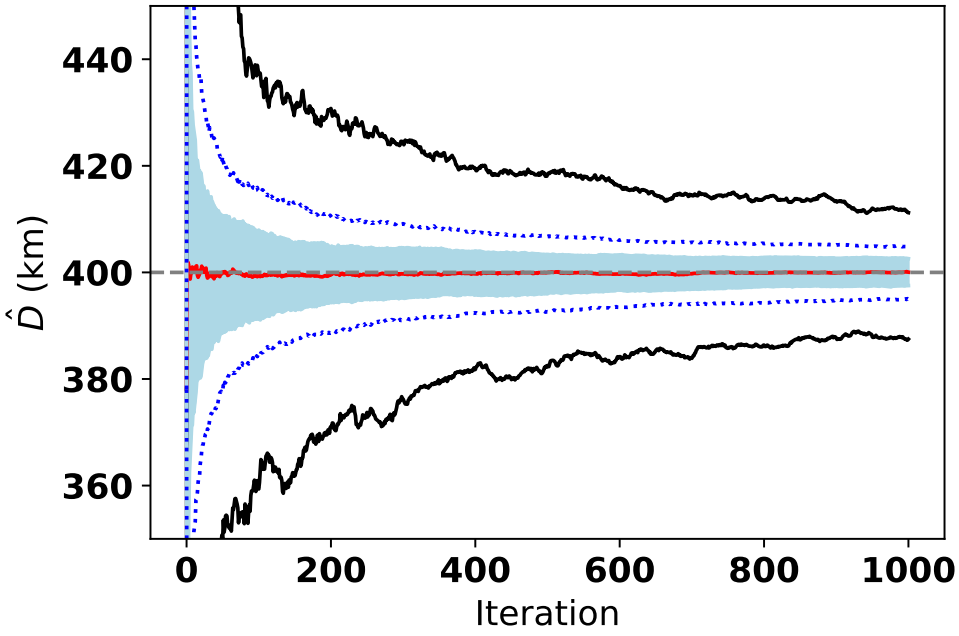
5.1.3 Algorithm Accuracy Comparison

The performance of different algorithms in determining a $\hat{D}$ estimate close to the assumed true value of 400 km was also recorded for an accuracy comparison. Figure 5.2 shows that U has the smallest maximum $\hat{D}$ value over 1000 runs up to 600 iterations, but gets surpassed by ABS after that point. However, at 600 iterations, ABS has taken at least 2400 samples, which is already a sample size far too high.

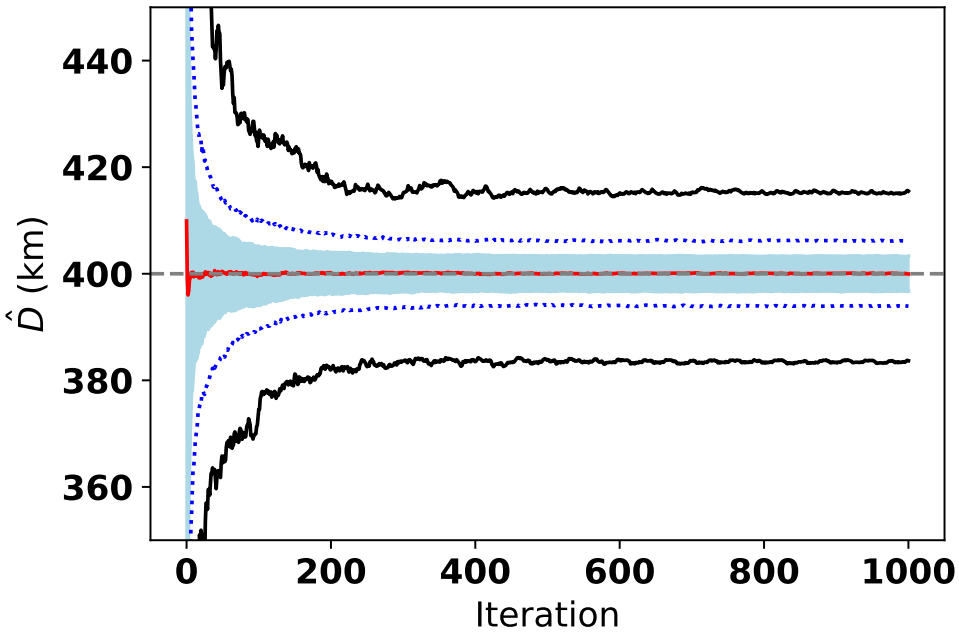
5.1.4 Comparison of Stopping Criteria

The main metrics used as stopping criteria were measures that required little to no extra computations within the algorithm itself. These include the total number of iterations, difference in $\hat{D}$ estimate over iterations, and standard deviation of the $\hat{D}$ estimate, $\hat{\sigma}_{\hat{D}}$.

For 100 iterations as stopping criterion for ABS and U, the difference from



(a) ABS



(b) Uncertainty

Figure 5.1: Results of the $\hat{D}$ estimate from each algorithm over 1000 runs. The black lines show the maximum and minimum values; the red line shows the median value; and the dark blue dotted lines show the 90th and 10th percentiles. The light blue shaded region is the area between the first and third quartiles.

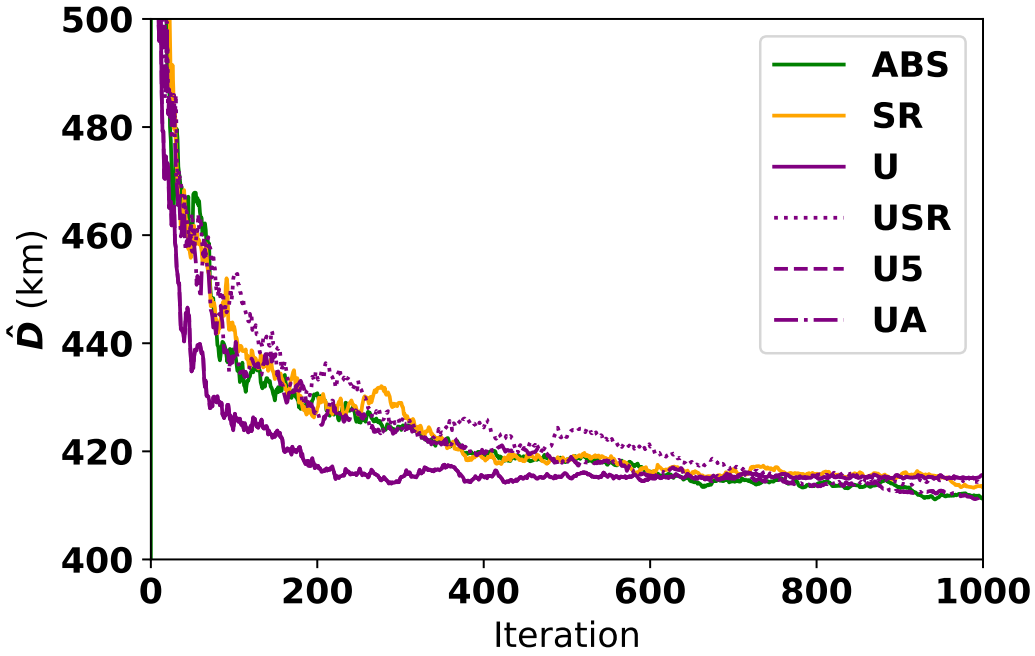


Figure 5.2: Comparing the maximum $\hat{D}$ value at every iteration of the 6 algorithms, taken from 1000 runs of each. Even though maximum $\hat{D}$ values in early iterations can be very high, the y-axis upper limit of the graph is set at 500 km to ensure readability. The minimum value is set as the assumed true value of D, 400 km.

the true value of D, 400, was plotted in histograms in Figure 5.8. Results for the remaining algorithms can be found in Section A.2 in the appendix. The difference appears to be normally distributed for each algorithm. U shows a smaller spread in the difference compared to the other algorithms. However, at the 100th iteration, each algorithm will have taken at least 400 samples, which is a very large sample size.

The remaining stopping criteria were also tested, and the difference from the assumed true value of 400 km along with sample size was recorded. The figures in this section focus on ABS and U as the results for these two algorithms are the most relevant. The remaining results for the other algorithms can be found in the Appendix in Section A.2.

In terms of the $\hat{D}$ difference from 400 km, the results are shown for ABS and U in Figure 5.6. It is clear from all subfigures that absdiff produces a much larger difference overall. The remaining stopping criteria are overlapping in the figures and appear to be centred around 0, with std25 showing the largest

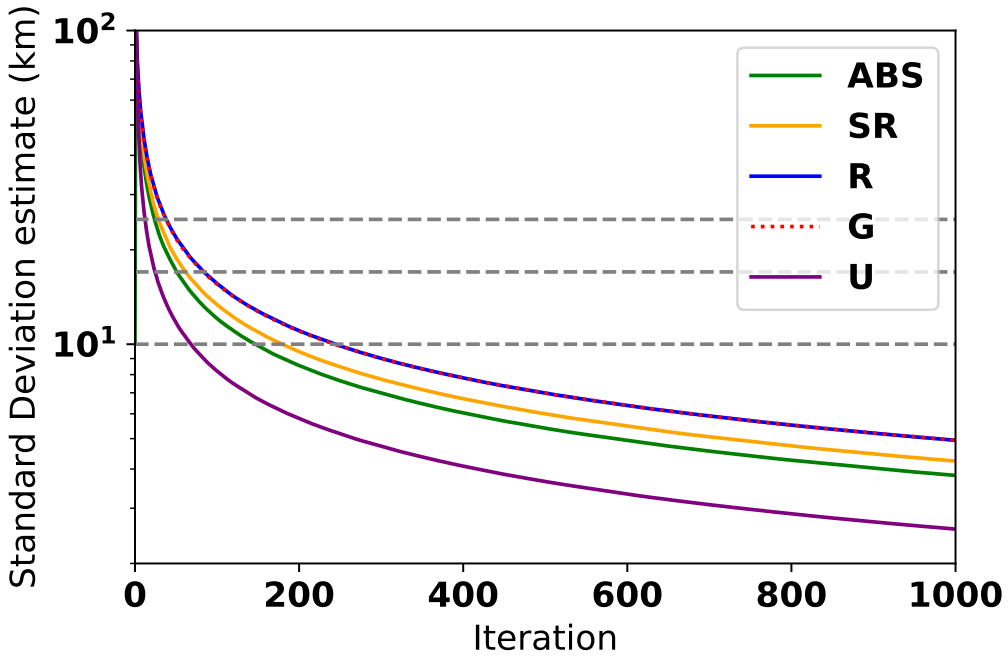


Figure 5.3: Median $\hat{\sigma}_{\hat{D}}$ for five of the sampling algorithms over 1000 runs of 1000 iterations. Greedy and Random have overlapping lines due to very close median values of standard deviation decrease over 1000 iterations of 1000 runs of each. The three dashed gray lines are at standard deviations 10, 17, and 25 km.

spread.

Additionally, it can be observed that std25 can sometimes lead to outliers where the difference from 400 is -300 km, meaning that this criterion for stopping is causing the algorithm to stop too earlier and, therefore, produce a less accurate result. To visualise this for U specifically, Figure 5.9 shows the individual runs of the U algorithm with their difference values. It can be observed that many of the algorithm runs stop at iteration 0, and produce a $\hat{D}$ value very far from 400. This difference is alleviated when the triple criteria is utilised, as shown in Figure 5.10.

These results should be considered with the corresponding sample sizes at stopping, shown in Figure 5.7. Here, it is clear that std10 takes a far greater number of samples overall compared to the other stopping criteria, making it a poor choice in terms of efficiency. The best option appears to be triple, due to it taking smaller sample sizes overall as well as having a small spread in difference from 400 km for all algorithms.

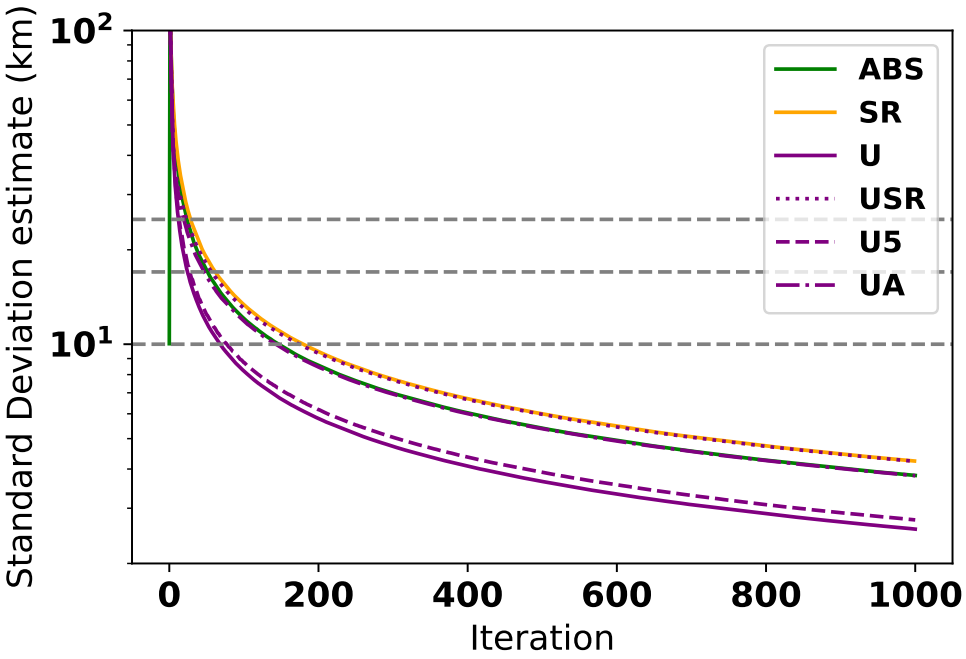


Figure 5.4: Median $\hat{\sigma}_{\hat{D}}$ change over the iterations in 1000 runs of each algorithm, excluding R and G. The three dashed gray lines are at standard deviations 10, 17, and 25.

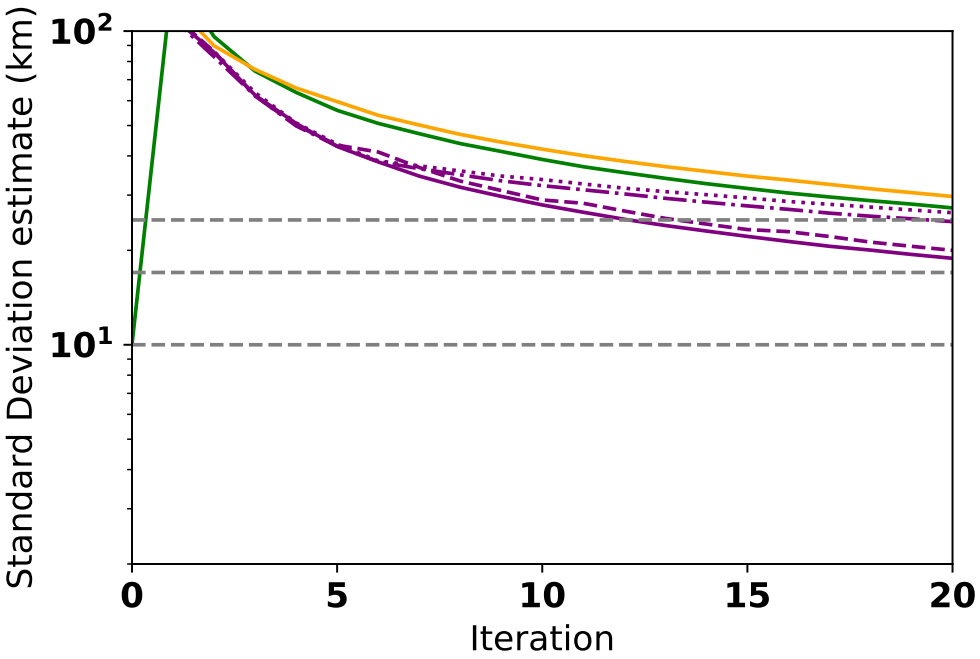
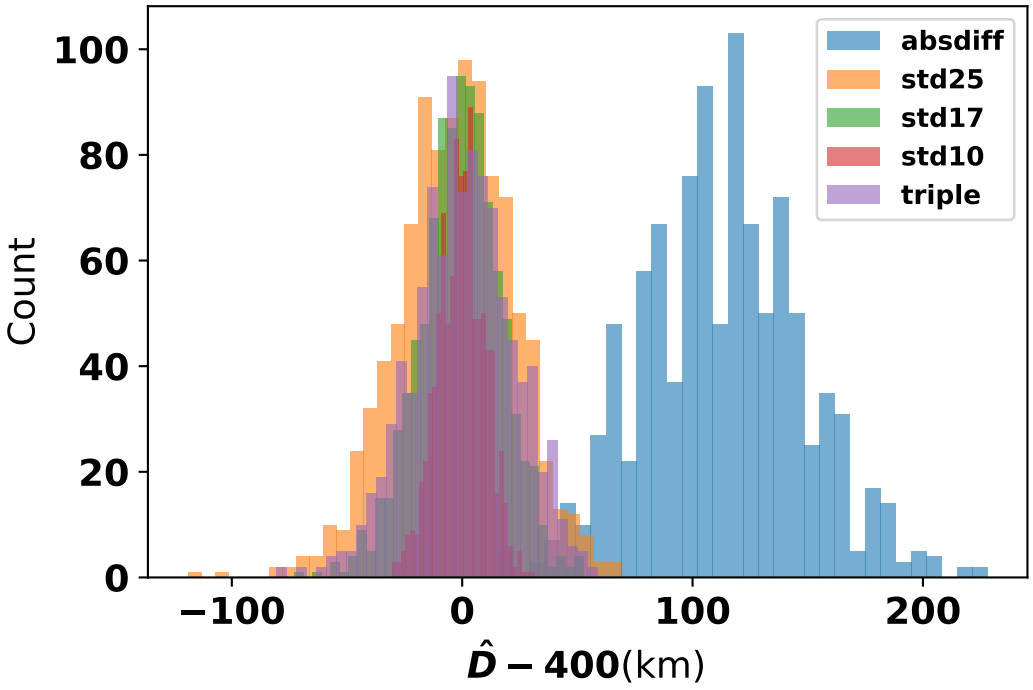
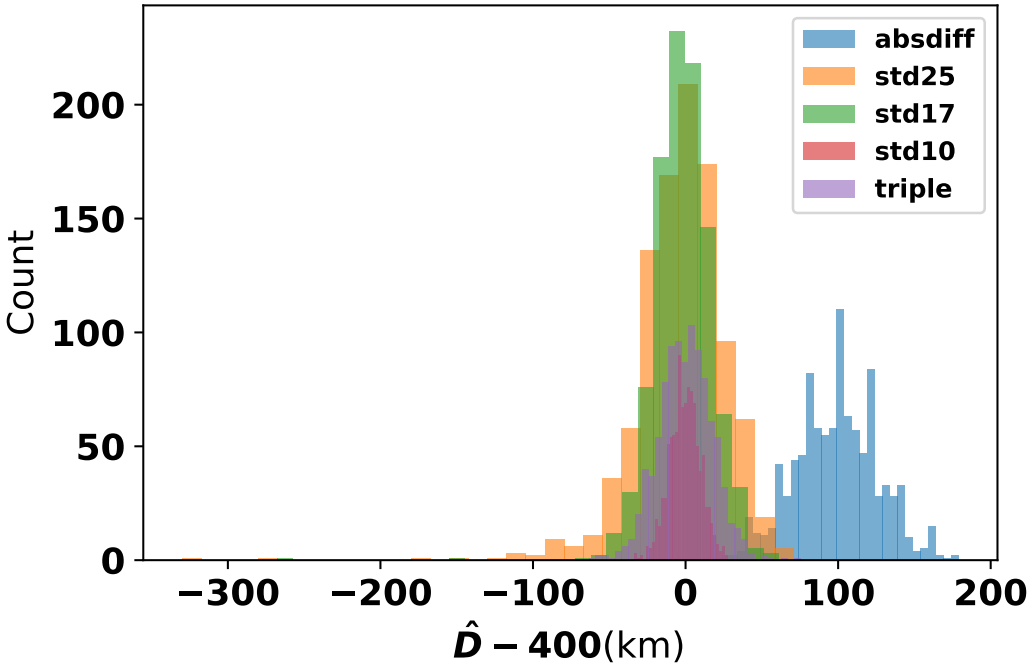


Figure 5.5: Zoomed in and focused on the first 20 iterations of median $\hat{\sigma}_{\hat{D}}$, with the same colouring scheme as in Figure 5.4. The three dashed grey lines are at standard deviations 10, 17, and 25.

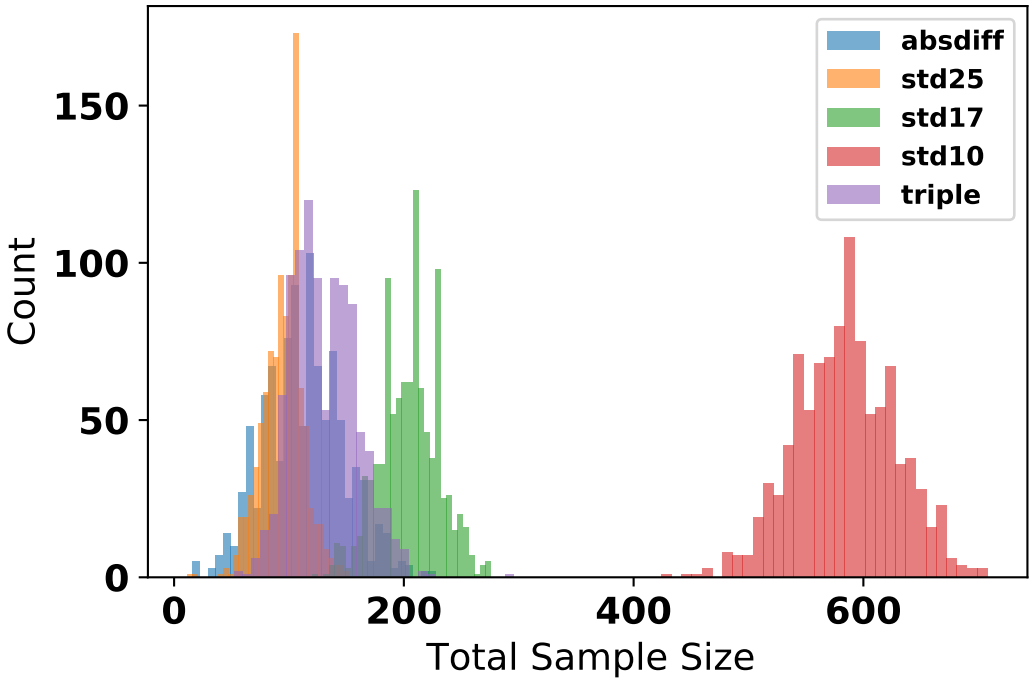


(a) ABS

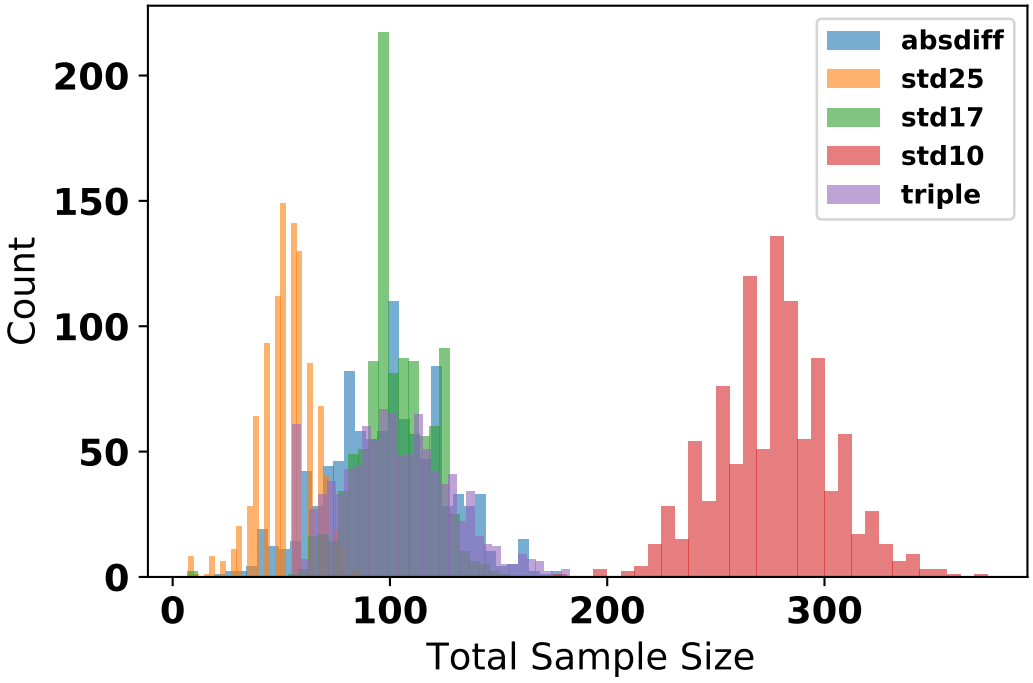


(b) U

Figure 5.6: Difference of $\hat{D}$ from the assumed true value of 400 km, across 1000 runs of ABS and U algorithms with the relevant stopping criteria applied. The count represents the number of runs where this difference occurs when the algorithm run stops. Note that this is not the absolute value of the difference.

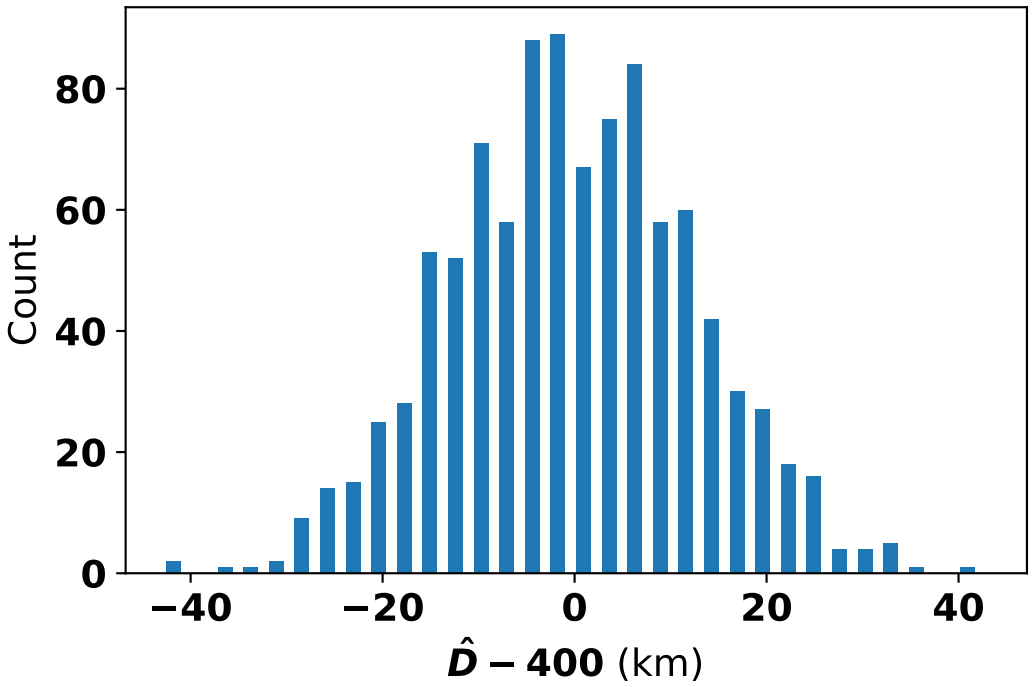


(a) ABS

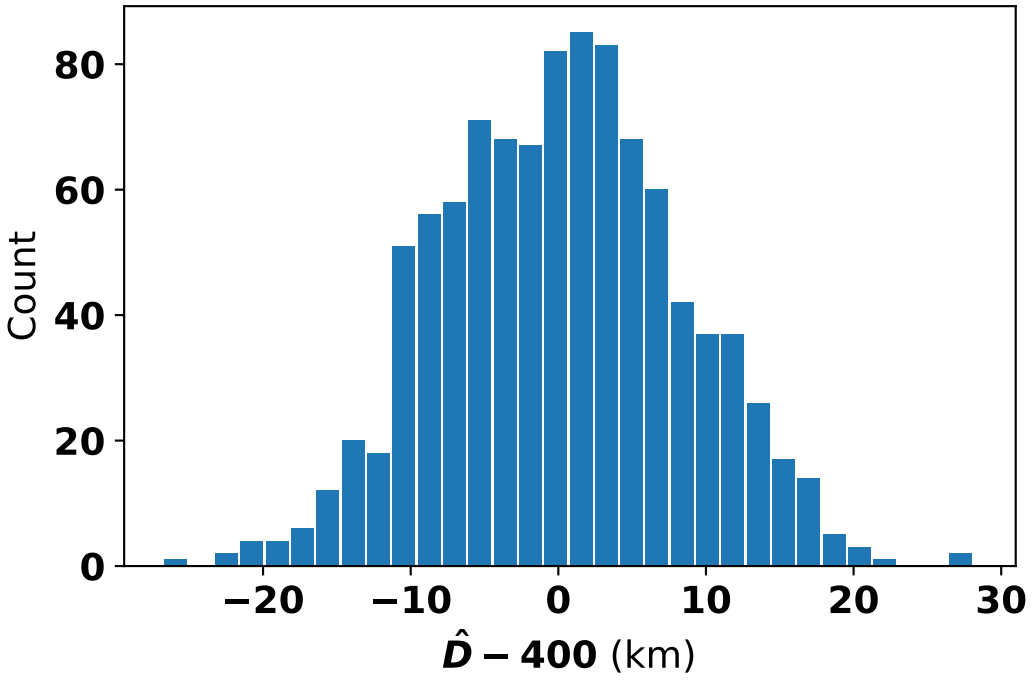


(b) U

Figure 5.7: Total sample size at the stopping iteration when the stopping criteria is reached, across 1000 runs of the ABS and U algorithms. The count represents the number of runs that ended with a given sample size.



(a) ABS



(b) U

Figure 5.8: Histograms for the value of $\hat{D}$-400 at the 100th iteration for 1000 runs of ABS and U, rounded to the nearest km. The count shows how many times runs the difference appeared in when the algorithm was stopped at the 100th iteration.

5.1.5 Best-performing Algorithm

We observe that U shows the fastest decrease in terms of standard deviation in 1000 runs for 1000 iterations, as well as the smallest maximum value of $\hat{D}$ for 600 iterations. Before this algorithm is applied on the dataset of renewable energy potential presented in Chapter 3, the relevant stopping criteria must be tested for suitability.

As mentioned before, the "triple" stopping criteria shows the most favourable results and is a good contender for use with U sampling on the real data. However, there is an accuracy concern regarding the use of std25 with this stopping criteria. As seen from the stopping criteria result figures, using std25 as a stopping criterion sometimes produces a $\hat{D}$ estimate that is very different from the assumed true value of 400 km. On the other hand, using a stricter stopping criterion of std10 results in a much larger sample size being required to stop. To strike a finer balance between this trade-off, the stopping criteria of std17, or stopping after the standard deviation drops below 17 iterations, is selected to be used in the triple criteria on the real data, instead of std25 as was tested on synthetic data. The other two criteria present in triple of having at least ten iterations and checking the difference in $\hat{D}$ between iterations remain the same.

5.2 Real Data Results

Uncertainty Sampling was applied to the real data as it showed the best overall performance and results in the synthetic data experiments. Both wind onshore CF and solar CF time series datasets were investigated separately.

5.2.1 Solar Capacity Factor Results

Unfortunately, the algorithm did not converge when tested on solar CF, even after preprocessing was applied to only take the maximum solar CF values for each location. As solar CF is heavily affected by time of day, there is a lot more variability in the data as opposed to wind CF values, which are not necessarily heavily affected as much by sunrise or sunset.

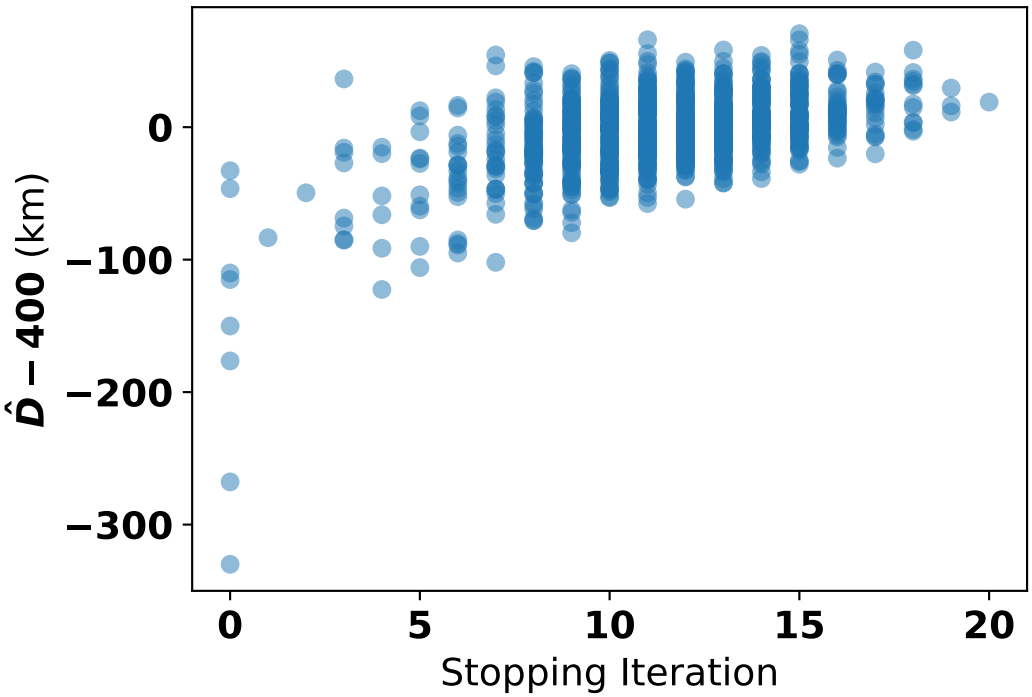


Figure 5.9: The difference of the $\hat{D}$ estimate from the true value in U sampling plotted against the stopping iteration for stopping criterion std25.

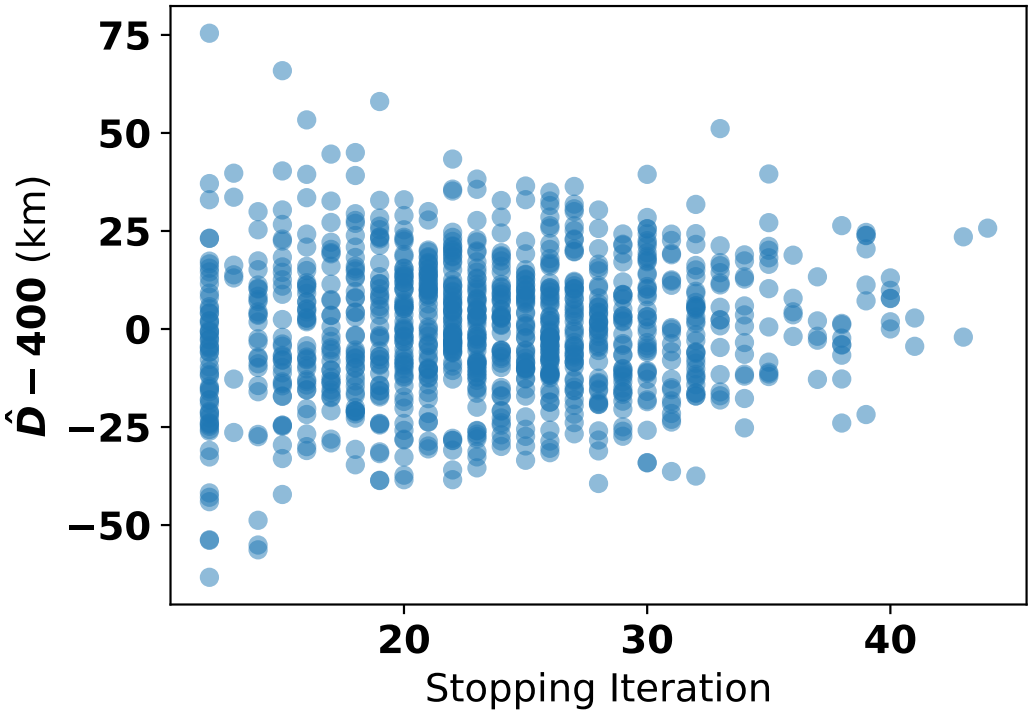


Figure 5.10: The difference of the $\hat{D}$ estimate from the true value in U sampling plotted against the stopping iteration for the triple stopping criterion.

5.2.2 Wind Onshore Capacity Factor Results

For the city of Utrecht in the Netherlands, the Correlation Halving Distance of two locations was investigated: (52,5) and (52,5.25). Close locations to Zurich, Madrid and Kiev were chosen from the dataset as origin cells as well. The results are shown in Table 5.1 and Table 5.2. Both locations close to (or in) Utrecht show similar results in every metric shown.

For some cities, the mean correlation at the mean CHD value is more than 10% away from 0.5, as shown for both points close to Utrecht. The three other cities show the converse, however. Passing the small and big circle check is also not consistent throughout, but it is clear that the mean correlation at the big circle distance is smaller than the mean correlation at the small circle distance, which indicates that correlation does generally decrease with distance even when it may be different for individual locations in different directions.

Coordinates (latitude, longitude)	**Nearest City**	**Mean CHD (km)**	**σ of CHD**	**Mean Total Sample Size**
(52,5)	Utrecht	448	8.338	112
(52,5.25)	Utrecht	442	7.003	114
(47.5,8.5)	Zurich	172	28.000	63
(40.5,-3.75)	Madrid	306	6.479	61
(50.5,30.5)	Kiev	427	12.956	65

Table 5.1: Correlation Halving Distance results after 10 runs on each origin cell. The standard deviations of the 10 CHD values for each city are shown in the fourth column.

Coordinates (latitude, longitude)	**Mean ρ at CHD**	**Mean ρ (small circle check)**	**Mean ρ (big circle check)**
(52,5)	0.550	0.572	0.523
(52,5.25)	0.552	0.575	0.552
(47.5,8.5)	0.447	0.480	0.420
(40.5,-3.75)	0.489	0.508	0.472
(50.5,30.5)	0.537	0.562	0.518

Table 5.2: Correlation Halving Distance results after 10 runs on each origin cell.

The results for 100 runs with random centres in the restricted dataset

bounds are as follows: the mean correlation at CHD is 0.494; the small circle check yields a mean correlation of 0.515; and the big circle check yields a mean correlation of 0.474. While the mean results show that the small circle correlation is greater than 0.5 and vice versa for the big circle, this check indicates that the algorithm has a bias where it over or underestimates the Correlation Halving Distance, and, on average, over 100 runs, the bias cancels itself out.

For a more visual inspection of the results of 100 runs, the 100 different origins with their corresponding correlation halving distance values are plotted in Figure 5.11 with colouring determined by mean onshore wind CF over 2020. In the landlocked regions of eastern Switzerland and western Austria, the CHD values are relatively low, at around 105 km and 110 km, respectively. However, at more northern regions bordering the North Sea, such as the grid cell in Northern France, the CHD value is 437 km instead. The Northern locations overall show a trend of higher CHD values compared to the south of the continent. This may be due to stronger wind currents in the North Sea region compared to the Mediterranean sea. However, there are outliers even within these trends. A point off the coast of Norway has a CHD value of 176 km, which is quite small compared to its surrounding CHD values. Similarly, the point in North Portugal also has a relatively high CHD value of 312 km when compared to South France and the other sampled cells in Southern Europe.

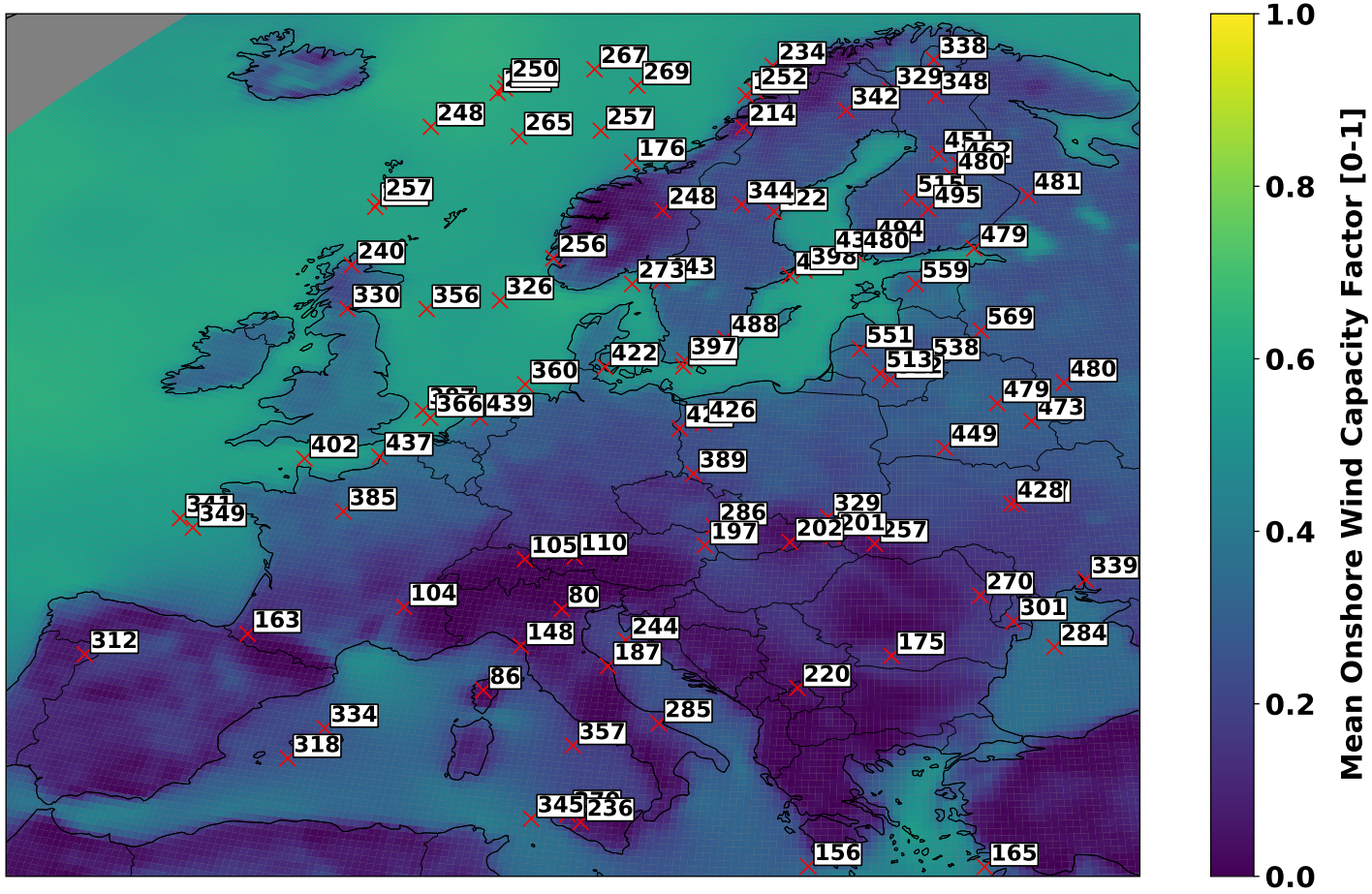


Figure 5.11: A hundred locations with their corresponding CHD values plotted. Each value is rounded to the nearest kilometre. The background shows the mean onshore wind capacity factor over 2020.

Additionally, 87 out 100 runs resulted in a correlation value that is in the range $[0.5 \pm 0.05]$. This means that the majority of the time, the algorithm has about a 10% error in producing an accurate Correlation Halving Distance value.

In terms of efficiency, each run of the 100 took 62 samples on average, meaning that 62 time series correlations were calculated on an average run. The brute force approach relied doing correlation between every pair of grid cells to determine the CHD for a single origin cell. Given that there are 21,019 grid cells in total, taking only 62 correlations greatly saves on computation cost.

6. Conclusion

6.1 Summary

This study examined various spatial sampling algorithms to efficiently calculate Correlation Halving Distance (CHD), a distance value, where the correlation between the time series at the CHD and the time series at the origin cell is 0.5. Experiments were first conducted on synthetic data, and the best-performing algorithm, Uncertainty sampling, was applied to real data. This algorithm showed good performance on average with onshore wind CF in terms of accuracy and efficiency. More specifically, the correlation at the CHD was within 10% of 0.5 for most random runs of Uncertainty Sampling. On average, 62 samples were taken for 100 random runs of Uncertainty Sampling, representing a significant improvement in the number of correlations calculated, considering the original dataset contains 21,019 time series. However, validation checks using the standard deviation of the estimate did not yield optimal results.

Two validation checks were performed to validate the obtained CHD values for wind onshore CF. The "small circle check" required the mean correlation over ten algorithm runs at a distance of CHD minus 2 standard deviations to be greater than 0.5, while the 'big circle check' required that the mean correlation over ten algorithm runs at a distance of CHD plus 2 standard deviations be smaller than 0.5. These two checks did not consistently hold up for all grid cells whose CHD was being calculated, indicating that the CHD value may be over or underestimated for certain locations.

Additionally, the algorithm did not produce results for $\hat{D}$ using the solar CF dataset. This happened despite data pre-processing measures applied to account for possible complications resulting from how heavily solar CF is affected by time of day.

6.2 Discussion

It is not fully clear whether the sampling algorithm itself needs refinement, or if the data needs to be further pre-processed to reduce the level of noise. There is likely to be a trade-off where sampling algorithms that produce more accurate CHD values may consume more resources. Additionally, a location at the Correlation Halving Distance for an origin cell may produce a correlation value that is very different from another location at the same distance but in a different direction, due to geographical variation in a circle around the same distance. It is not fully clear how to accommodate this fact, and the definition of Correlation Halving Distance may need to be modified to account for this variation. Further investigation is also required as to why the solar dataset requires a modified approach.

In terms of understanding weather regimes to aid Europe's power supply transition to renewables, it is helpful to review Figure 5.11 as it shows many CHD values throughout the continent with mean onshore wind CF in the background. It appears that, generally, CHD values for onshore wind CF are higher at higher latitudes, with proximity to the North Sea also contributing to a boost in the value. However, more information may be needed to determine more complete trends as well as peculiarities, given that these locations are all random, and more than 100 of them may be required. Overall, it is difficult to state anything determinate about CHD and weather regimes given the limited scope of this research. However, the findings do provide a good foundation for future studies to build upon by using the sampling techniques here and perform more comprehensive analyses on weather regimes.

6.3 Future Work

The Correlation Halving Distance value by itself may need to be further refined, as the distance at which correlation is 0.5 is an arbitrary value used in this study. It may be the case that discovering a value for distance from the origin where correlation is 0.3 or 0.7 could also be valuable depending on the specific use case. To this end, it would also be helpful to investigate whether the base

exponent should be Euler's number e or a different base, so that the functional form can be fine-tuned to yield the most useful results. Additionally, as CHD is a novel measure, research could be conducted on better function forms to fit that may not be negative exponential in the input of distance. Additionally, the range of the current functional form is strictly positive, meaning that negative correlation values will adversely affect the accuracy of the resulting Correlation Halving Distance estimate.

With respect to Uncertainty Sampling, it is important to note that taking four sample points per iteration even on the real data was done to maintain the algorithm form that was used in the theoretical experiments, as all other algorithms were also taking four samples at a time. An avenue to explore would be to take one sample at at time for each function fit, and observe how that affects performance. The choice of scale factors was also non-trivial; choosing one scale factor value at each iteration as opposed to four different ones may noticeably affect the algorithm's performance, and it is not apparent which scale factor value should be chosen.

As it can still be very computationally intensive to compute CHD for each coordinate in the dataset of Europe, a spatial interpolation approach must be discovered that may estimate the CHD value for a given cell based on the calculated CHD value for a neighbouring cell. A successful interpolation approach will further reduce the computational resources required to understand how the CHD values vary for all of Europe.

The Correlation Halving Distance measure, as well as the algorithms proposed in this study, may be applied to different spatio-temporal datasets as well. The flood synchrony scale [11] is a similar measure using a similar data structure: essentially, a two-dimensional array where each element is a time series, and the elements cannot be modified or moved around in the array. There are possibly other domains where such approaches would be useful and should be explored for similar analyses.

Bibliography

[1] European Commission and Directorate-General for Communication, *European Green Deal—Delivering on Our Targets*. Publications Office of the European Union Luxembourg, 2021. DOI: doi/10.2775/373022.

[2] ENTSO-E, *European resource adequacy assessment - 2021 edition*, 2021. [Online]. Available: https://www.entsoe.eu/outlooks/eraa/2021/.

[3] H. C. Bloomfield, D. J. Brayshaw, and A. J. Charlton-Perez, "Characterizing the winter meteorological drivers of the european electricity system using targeted circulation types," *Meteorological Applications*, vol. 27, no. 1, Dec. 2019. DOI: 10.1002/met.1858.

[4] S. Sundar, M. Craig, A. Payne, D. Brayshaw, and F. Lehner, "Meteorological drivers of resource adequacy failures in current and high renewable western u.s. power systems," *California Digital Library (CDL)*, Sep. 2022. DOI: 10.31223/x57d2g.

[5] K. van der Wiel, L. Stoop, B. van Zuijlen, R. Blackport, M. van den Broek, and F. Selten, "Meteorological conditions leading to extreme low variable renewable energy production and extreme high energy shortfall," *Renewable and Sustainable Energy Reviews*, vol. 111, pp. 261–275, 2019, ISSN: 1364-0321. DOI: 10.1016/j.rser.2019.04.065.

[6] C. M. Grams, R. Beerli, S. Pfenninger, I. Staffell, and H. Wernli, "Balancing europe's wind-power output through spatial deployment informed by weather regimes," *Nature Climate Change*, vol. 7, no. 8, pp. 557–562, Jul. 2017. DOI: 10.1038/nclimate3338.

[7] M. T. Craig, J. Wohland, L. P. Stoop, *et al.*, "Overcoming the disconnect between energy system and climate modeling," *Joule*, vol. 6, no. 7, pp. 1405–1417, 2022, ISSN: 2542-4351. DOI: 10.1016/j.joule.2022.05.010.

[8] R. Wuijts, L. Stoop, J.Hu, *et al.*, "Linking unserved energy to weather regimes," *Earth's Future*, 2023, In submission.

[9] K. van der Wiel, H. C. Bloomfield, R. W. Lee, *et al.*, "The influence of weather regimes on european renewable energy production and demand," *Environmental Research Letters*, 2019. DOI: 10.1088/1748-9326/ab38d3.

[10] H. Hersbach, B. Bell, P. Berrisford, *et al.*, "The ERA5 global reanalysis," *Quarterly Journal of the Royal Meteorological Society*, vol. 146, no. 730, pp. 1999–2049, Jun. 2020. DOI: 10.1002/qj.3803.

[11] W. R. Berghuijs, S. T. Allen, S. Harrigan, and J. W. Kirchner, "Growing spatial scales of synchronous river flooding in europe," *Geophysical*

Research Letters, vol. 46, no. 3, pp. 1423–1428, 2019. DOI: h10.1029/2018GL081883.

[12] M. Kemter, B. Merz, N. Marwan, S. Vorogushyn, and G. Blöschl, "Joint trends in flood magnitudes and spatial extents across europe," *Geophysical Research Letters*, vol. 47, no. 7, Apr. 2020. DOI: 10.1029/2020gl087464.

[13] G. Giebel, *Equalizing effects of the wind energy production in northern europe determined from reanalysis data*, 2000. [Online]. Available: https://orbit.dtu.dk/en/publications/equalizing-effects-of-the-wind-energy-production-in-northern-euro.

[14] J. Olauson and M. Bergkvist, "Correlation between wind power generation in the european countries," *Energy*, vol. 114, pp. 663–670, Nov. 2016. DOI: 10.1016/j.energy.2016.08.036.

[15] C. M. S. Martin, J. K. Lundquist, and M. A. Handschy, "Variability of interconnected wind plants: Correlation length and its dependence on variability time scale," *Environmental Research Letters*, vol. 10, no. 4, p. 044 004, Apr. 2015. DOI: 10.1088/1748-9326/10/4/044004.

[16] D. Wu, "Pool-based sequential active learning for regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1348–1359, 2019. DOI: 10.1109/TNNLS.2018.2868649.

[17] H. Yu and S. Kim, "Passive sampling for regression," in *2010 IEEE International Conference on Data Mining*, 2010, pp. 1151–1156. DOI: 10.1109/ICDM.2010.9.

[18] N. Abe and H. Mamitsuka, "Query learning strategies using boosting and bagging," in *Proceedings of the Fifteenth International Conference on Machine Learning*, ser. ICML '98, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 1–9, ISBN: 1558605568. DOI: 10.5555/645527.657478.

[19] W. Cai, Y. Zhang, and J. Zhou, "Maximizing expected model change for active learning in regression," in *2013 IEEE 13th International Conference on Data Mining*, 2013, pp. 51–60. DOI: 10.1109/ICDM.2013.104.

[20] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii: Association for Computational Linguistics, Oct. 2008, pp. 1070–1079. [Online]. Available: https://aclanthology.org/D08-1112.

[21] L. P. Stoop, E. Duijm, A. Feelders, and M. van den Broek, "Detection of critical events in renewable energy production time series," in *Advanced Analytics and Learning on Temporal Data*, Springer International Publishing, 2021, pp. 104–119. DOI: 10.1007/978-3-030-91445-5_7.

[22] C. Justus and A. Mikhail, *Energy statistics for large wind turbine arrays*, 1978. [Online]. Available: https://www.jstor.org/stable/43749121.

[23] *scipy.stats.pearsonr*, SciPy Documentation, https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html.

[24] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[25] *Geopy: Python Geocoding Toolbox*, https://github.com/geopy/geopy, 2023.

[26] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020. DOI: 10.1038/s41586-020-2649-2.

[27] *Cartopy: A Cartographic Python Library*, https://scitools.org.uk/cartopy, 2023.

[28] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

[29] Unidata, *netCDF Software*, https://www.unidata.ucar.edu/software/netcdf, 2012.

[30] S. Hoyer and H. Joseph, "xarray: N-D labeled Arrays and Datasets in Python," *Journal of Open Research Software*, vol. 5, no. 1, Apr. 2017. DOI: 10.5334/jors.148.

A. Appendix A

The technical specifications and environment for the experiments in this study are outlined in the following section. Additionally, the appendix includes diagrams for algorithm comparison results for algorithms that were not Uncertainty (U) or Adjusted Binary Search (ABS).

A.1 Setup and Environment

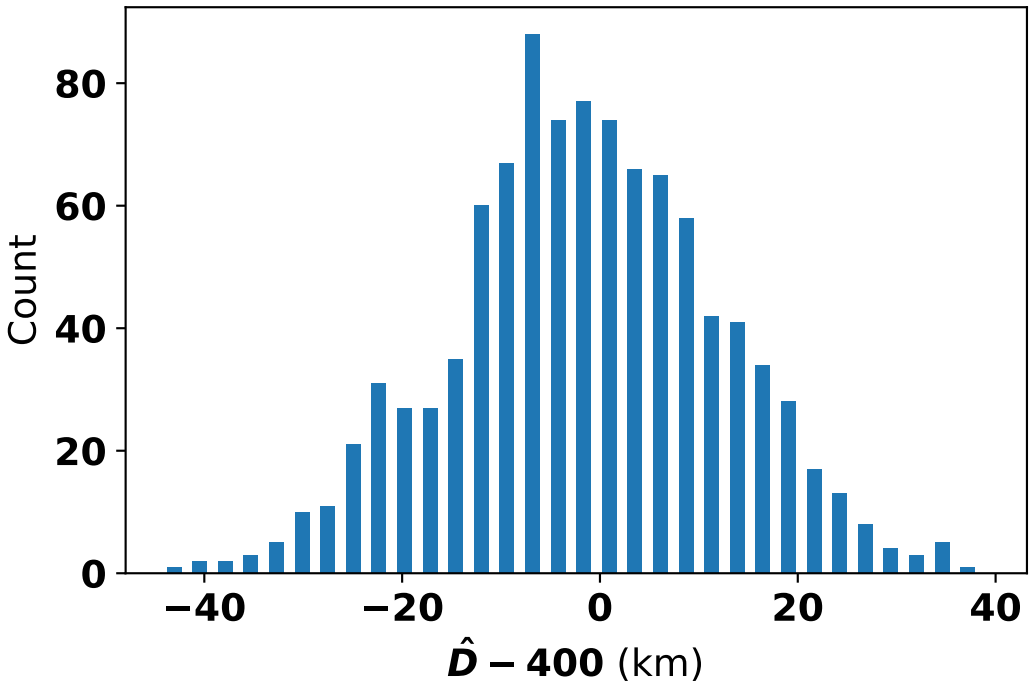
The environment used to test is a Jupyter Notebook with a conda Python 3.8.8 interpreter. Separate Jupyter notebooks were created to test individual algorithms as well as visualisations. These are also stored on a github repository for version control. The computer used for testing is a Macbook Pro 2020 with an M1 chip and 8GB of RAM.

Various Python libraries are utilised to provide relevant functionality. NumPy [26] is used for multiple functions such as providing a PCG64 generator for random value generation, and calculating metrics such as arithmetic mean and standard deviation of a list or array. SciPy [24] is used to fit the negative exponential curve function in Equation 4.1 onto the given data, and yield a CHD estimate along with the standard deviation in the estimate. GeoPy [25] is used to calculate the geodesic distance between two coordinates on the Earth's surface, where geodesic assumes the Earth is of an ellipsoid shape. Additionally, GeoPy allows for determining a coordinate that is a given input distance away from an input coordinate, as long as the angle is specified. Cartopy [27] and MatPlotLib [28] are used to plot various graphs and map projections.

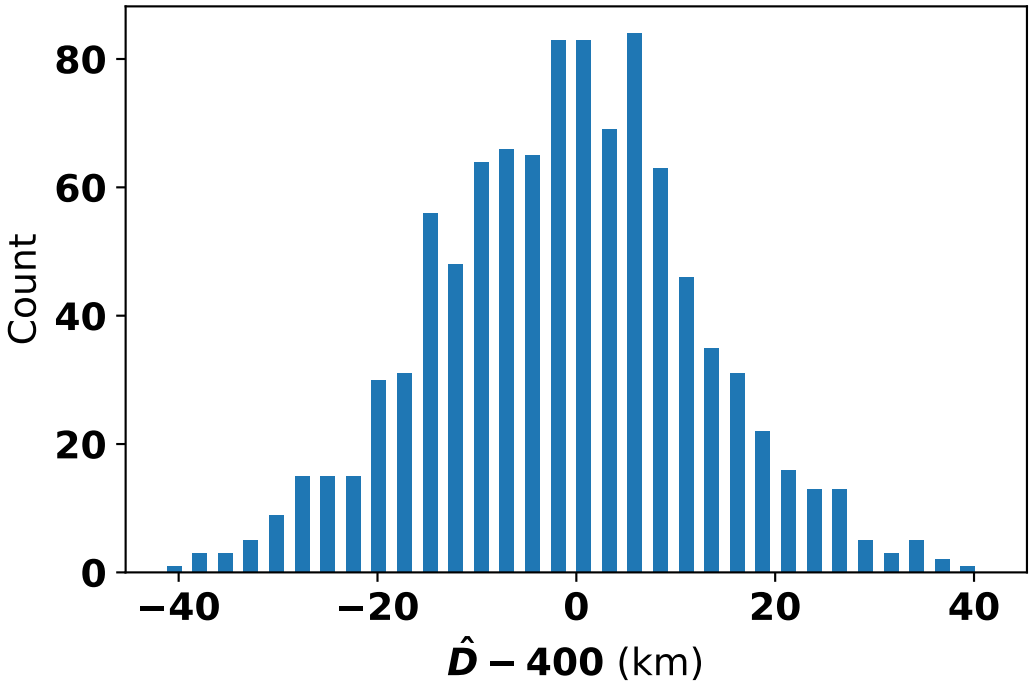
The dataset is available in netCHDF format files, a format commonly used for array-oriented scientific data [29]. To open and manipulate these for use in Python, the XArray [30] library is used. Additionally, XArray provides a convenient function that "snaps" to the nearest coordinate that actually exists in the dataset. For example, if a location at coordinate (35.01,55.01) must be sampled, XArray allows for the closest coordinate in terms of Euclidean distance to be selected, which is (35,55).

A.2 Algorithm Comparisons

The appendix includes additional figures showcasing experimental results for various algorithms that were not included in the main section. While the results for ABS and U were covered in the main text, the results and comparisons for all other algorithms are presented below.

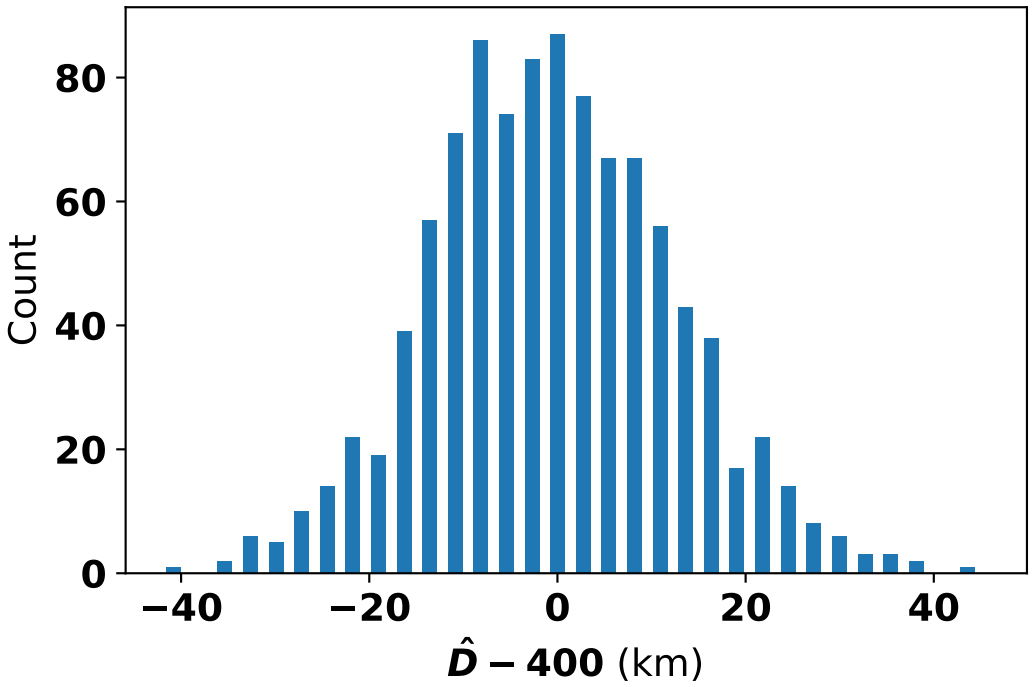


(a) SR

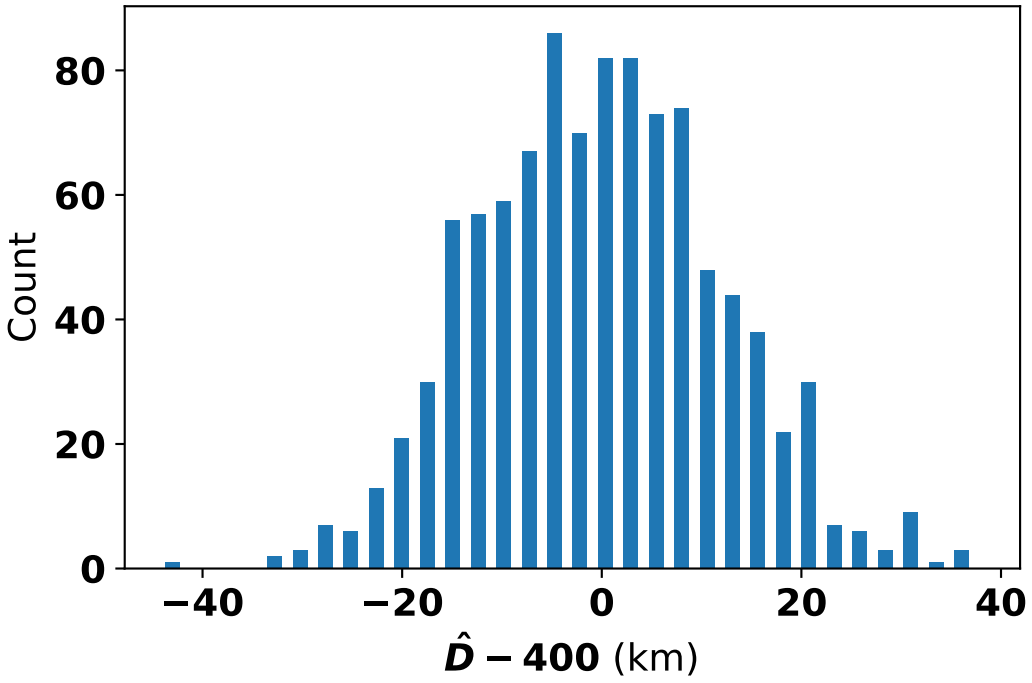


(b) USR

Figure A.1: Histograms for the value of $\hat{D}$-400 at the 100th iteration for 1000 runs of SR and USR, rounded to the nearest km. The count shows how many times runs the difference appeared in when the algorithm was stopped at the 100th iteration.

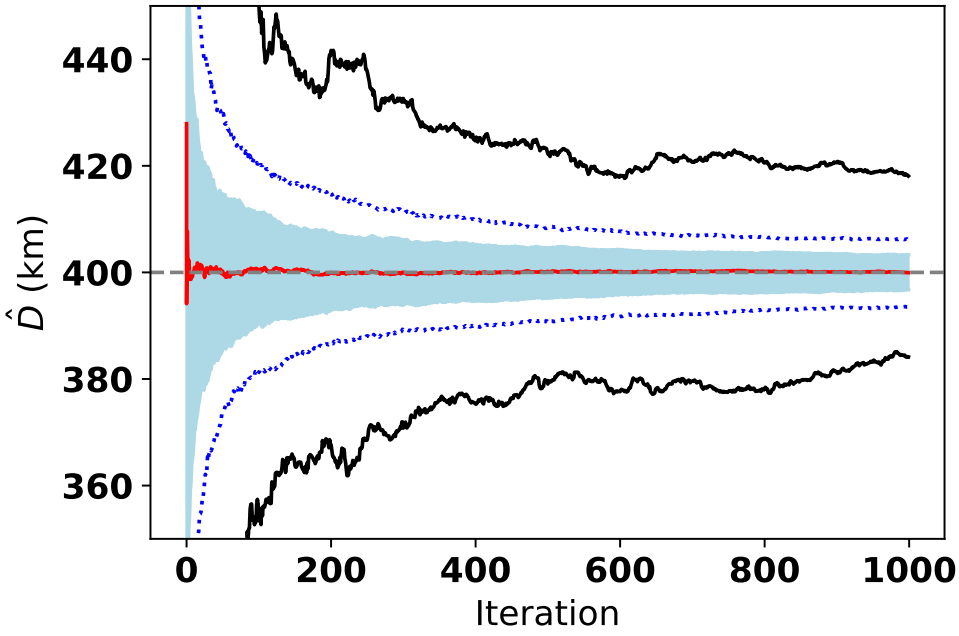


(a) U5

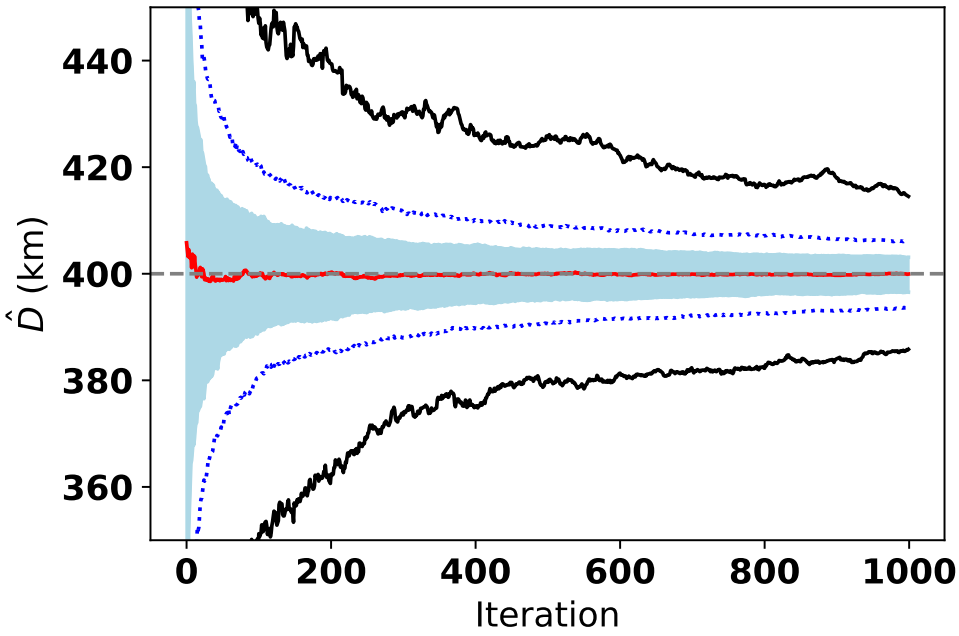


(b) UA

Figure A.2: Histograms for the value of $\hat{D}$-400 at the 100th iteration for 1000 runs of UA and U5, rounded to the nearest km. The count shows how many times runs the difference appeared in when the algorithm was stopped at the 100th iteration.

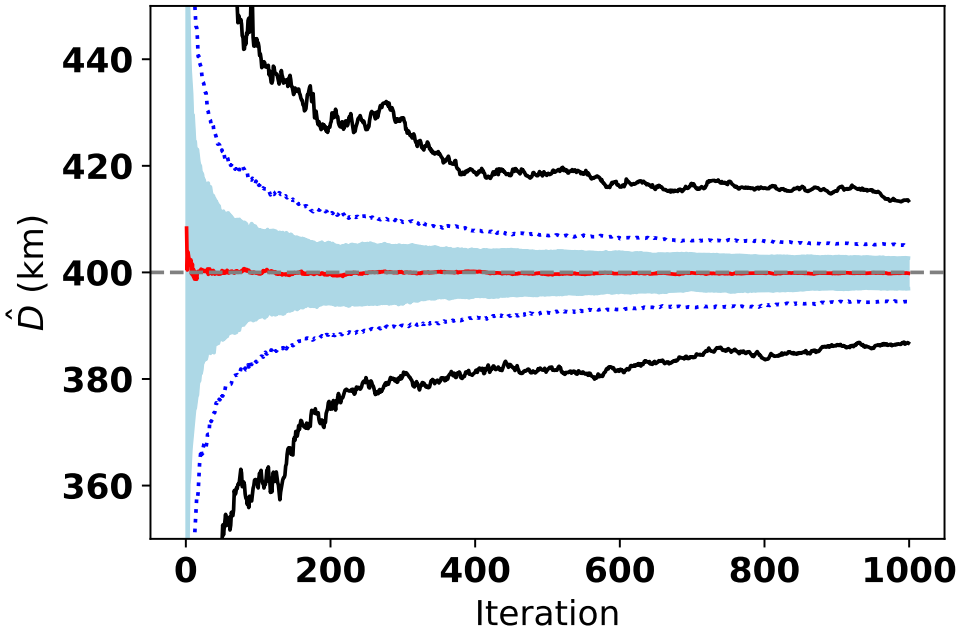


(a) G

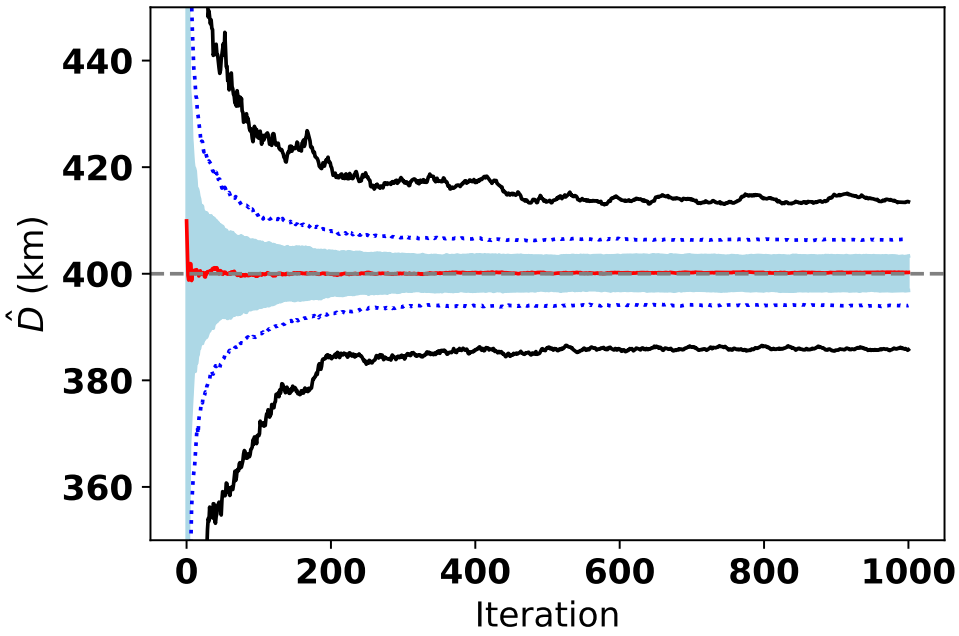


(b) R

Figure A.3: Results of the $\hat{D}$ estimate from the G and R algorithms over 1000 runs. The black lines show the maximum and minimum values; the red line shows the median value; and the dark blue dotted lines show the 90th and 10th percentiles. The light blue shaded region is the area between the first and third quartiles.

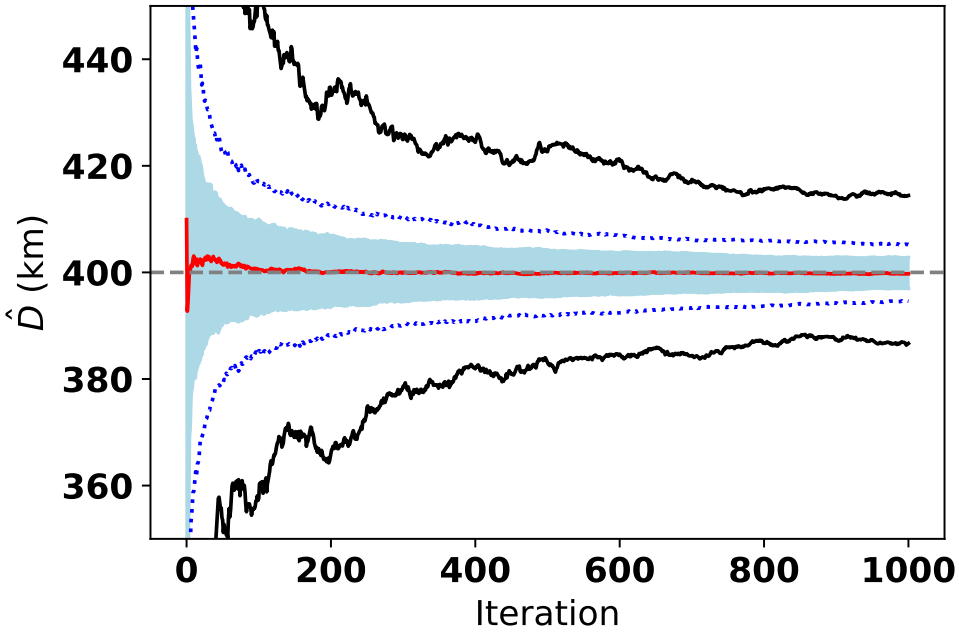


(a) SR

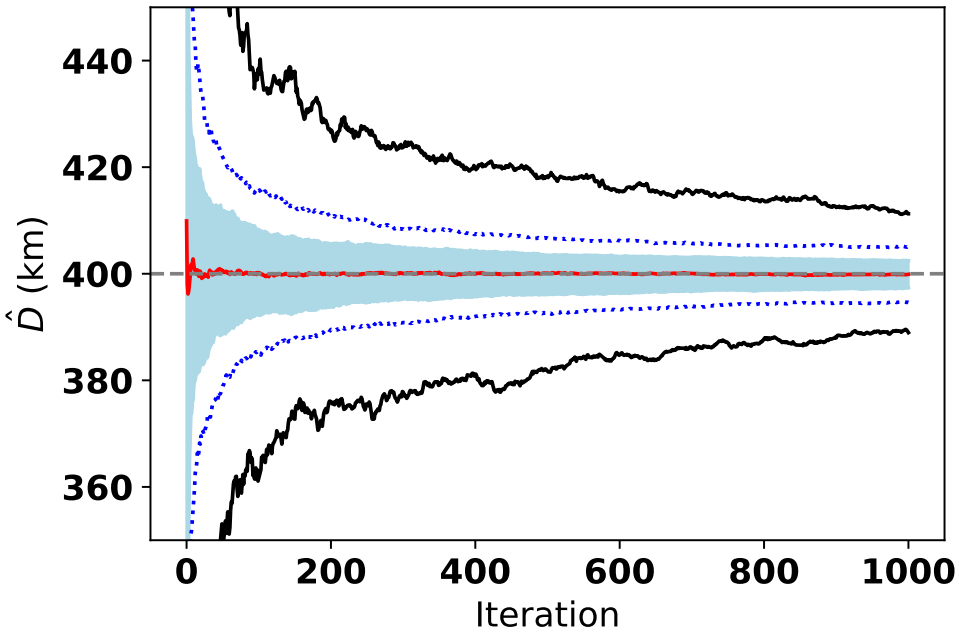


(b) U5

Figure A.4: Results of the $\hat{D}$ estimate for the SR and U5 algorithms over 1000 runs. The black lines show the maximum and minimum values; the red line shows the median value; and the dark blue dotted lines show the 90th and 10th percentiles. The light blue shaded region is the area between the first and third quartiles.

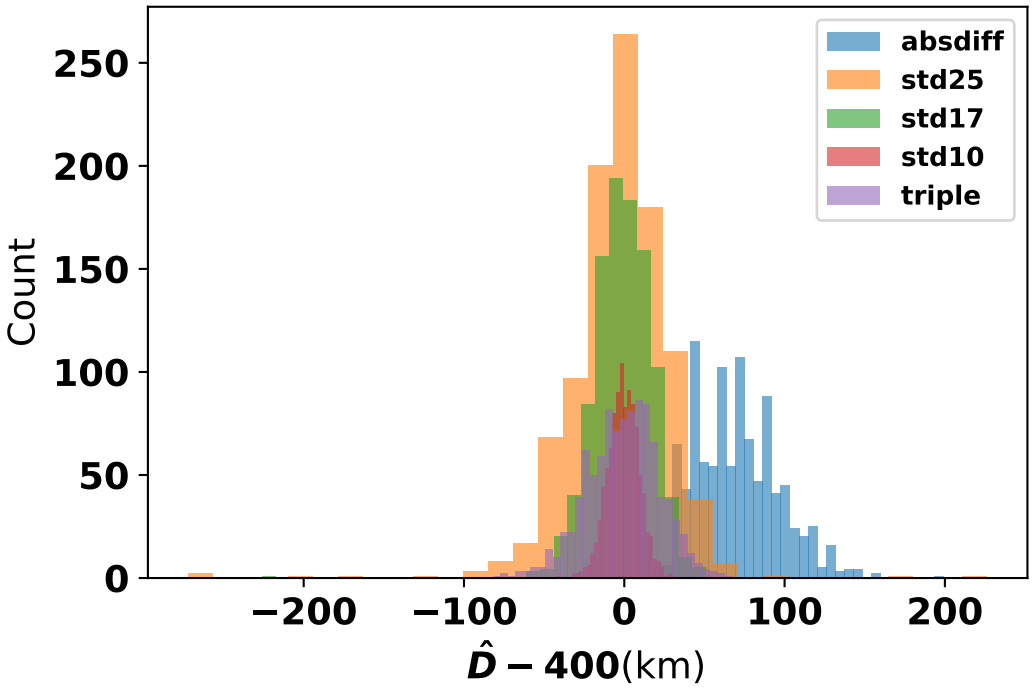


(a) USR

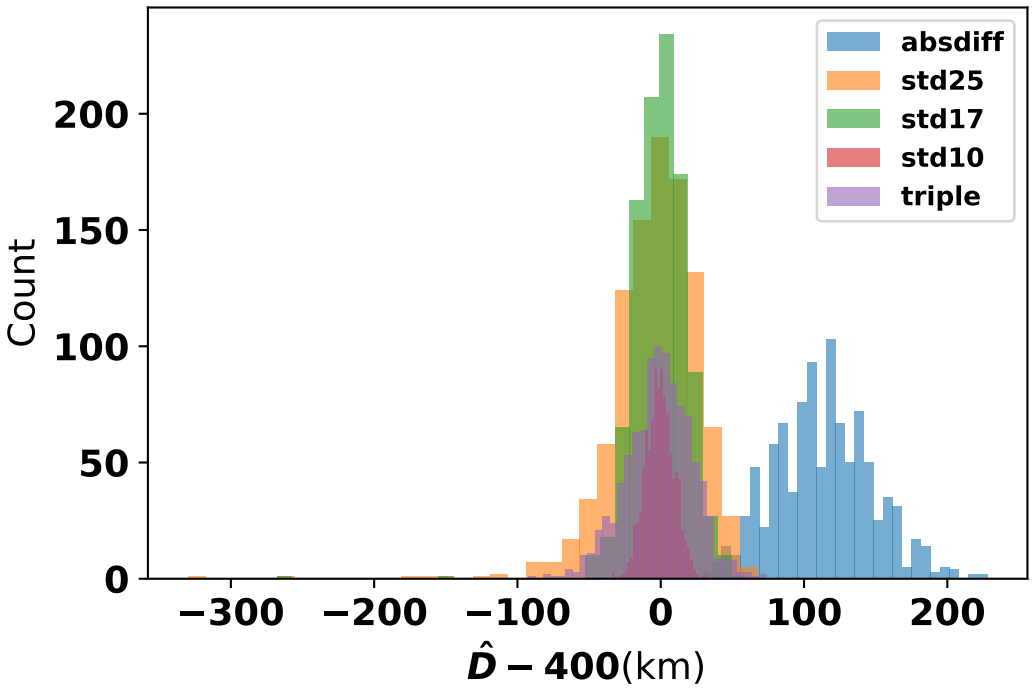


(b) UA

Figure A.5: Results of the $\hat{D}$ estimate for USR and UA algorithms over 1000 runs. The black lines show the maximum and minimum values; the red line shows the median value; and the dark blue dotted lines show the 90th and 10th percentiles. The light blue shaded region is the area between the first and third quartiles.

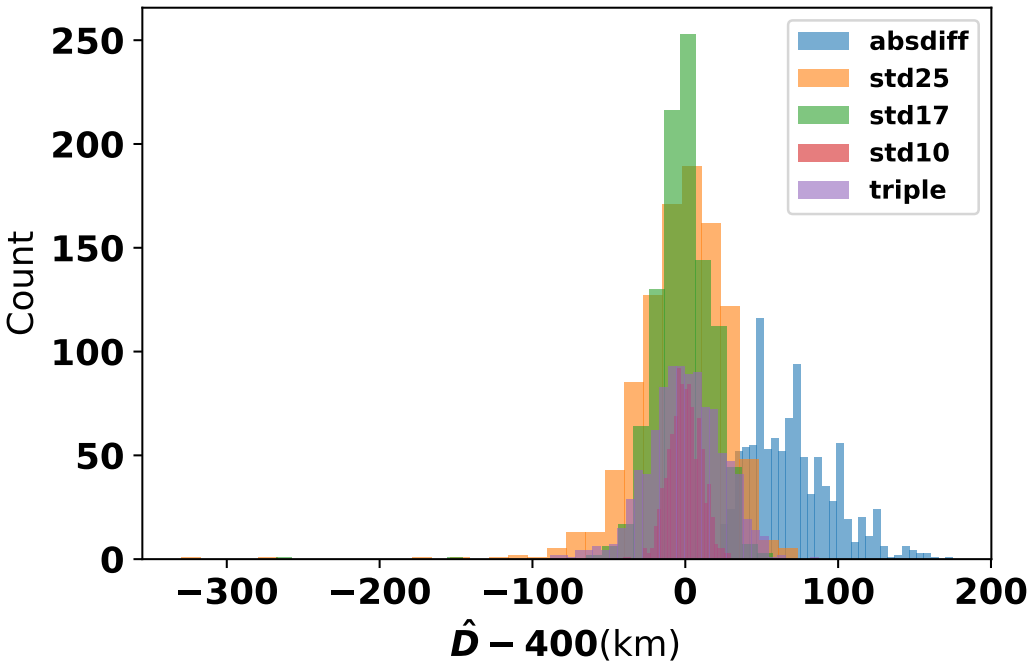


(a) SR

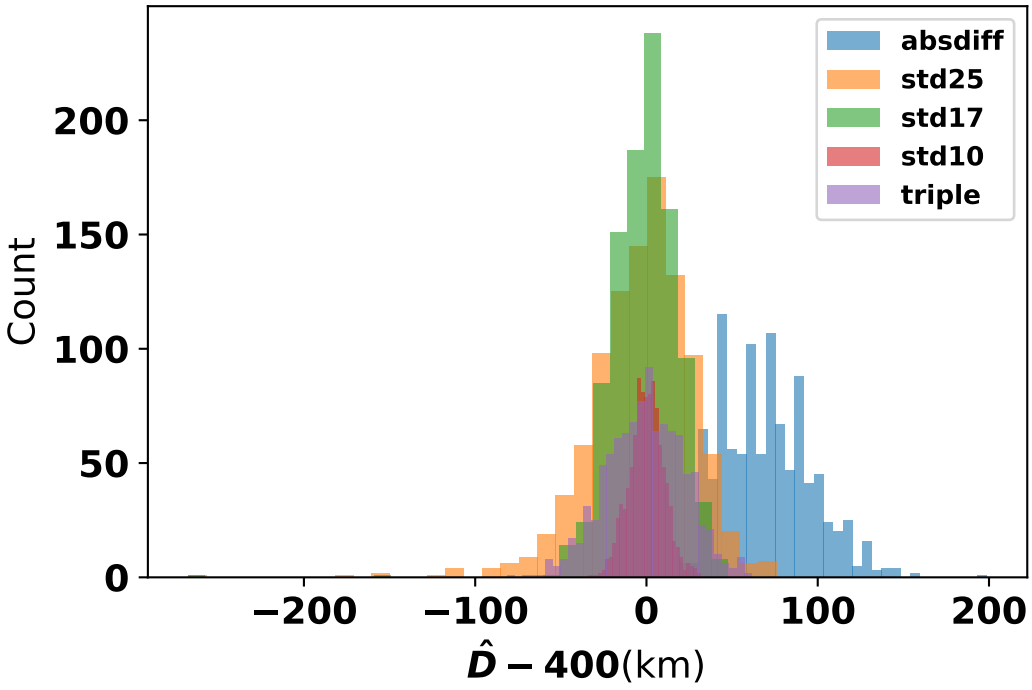


(b) USR

Figure A.6: Difference of $\hat{D}$ from the assumed true value of 400 km, across 1000 runs of the SR and USR algorithms with the relevant stopping criteria applied. The count represents the number of runs where this difference occurs when the algorithm run stops. Note that this is not the absolute value of the difference.

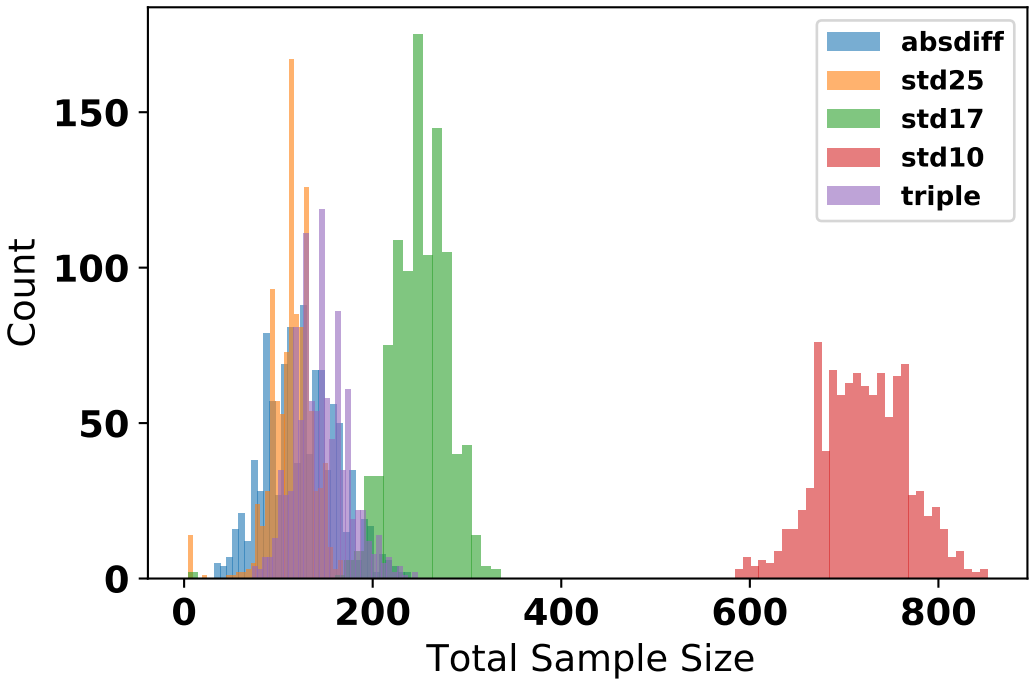


(a) U5

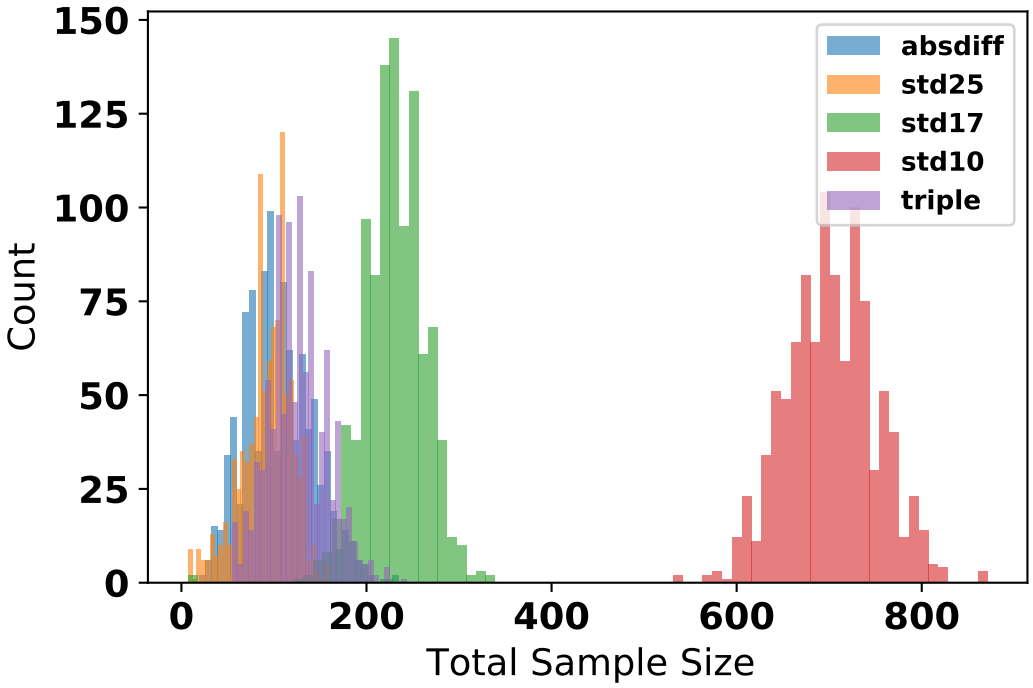


(b) UA

Figure A.7: Difference of $\hat{D}$ from the assumed true value of 400 km, across 1000 runs of U5 and UA with the relevant stopping criteria applied. The count represents the number of runs where this difference occurs when the algorithm run stops. Note that this is not the absolute value of the difference.

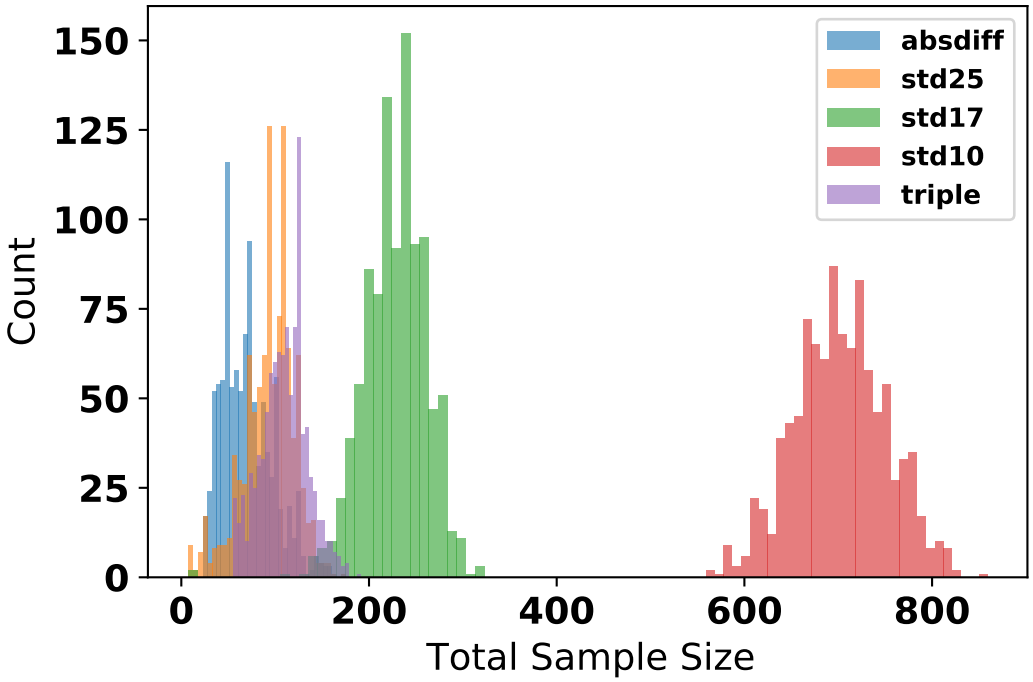


(a) SR

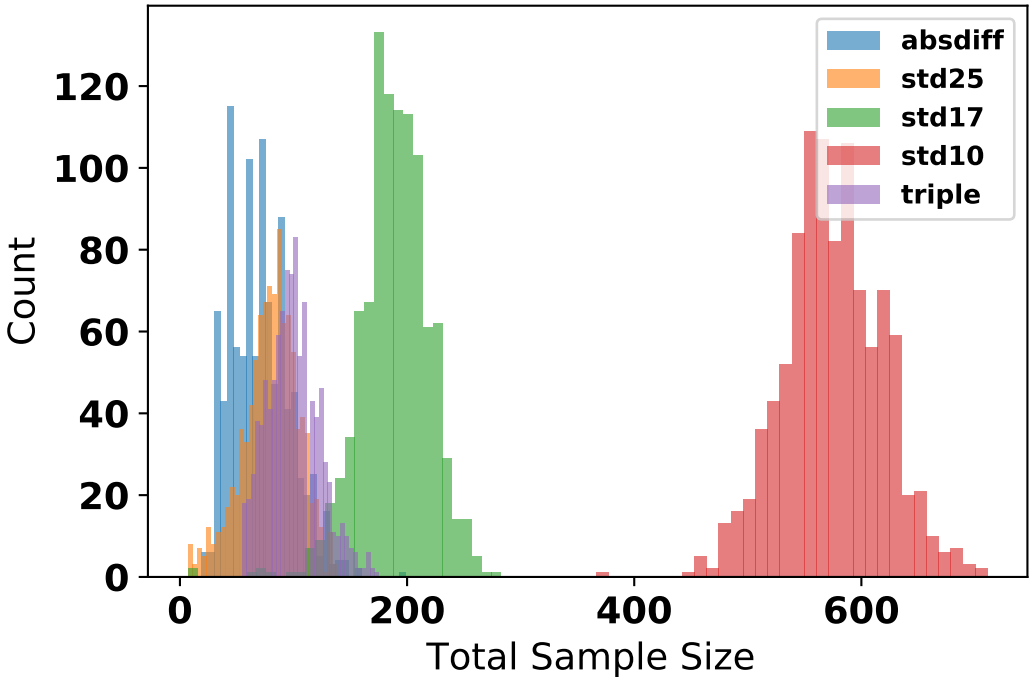


(b) USR

Figure A.8: Total sample size at the stopping iteration when the stopping criteria is reached, across 1000 runs of SR and USR algorithms. The count represents the number of runs that ended with a given sample size.



(a) U5



(b) UA

Figure A.9: Total sample size at the stopping iteration when the stopping criteria is reached, across 1000 runs of U5 and UA algorithms. The count represents the number of runs that ended with a given sample size.