University of Applied Sciences and Arts Northwestern Switzerland
School of Business

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Bundesamt für Umwelt BAFU

UNIVERSITY OF APPLIED SCIENCES AND ARTS NORTHWESTERN SWITZERLAND (FHNW)

BACHELOR THESIS

Time Series Clustering with Water Temperature Data

Author:
Roman BÖGLI

Supervisor:
Dr. Vidushi Christina BIGLER

A thesis submitted in fulfilment of the requirements for the degree of Bachelor of Science

in

Business Information Technology (BIT)

August 7, 2020

Abstract

Time Series Clustering with Water Temperature Data

by Roman BÖGLI

This thesis studies three different approaches to cluster time series data using the unsupervised pattern recognition method called hierarchical clustering. The underlying data constitute long-term water temperature measurements of several Swiss water bodies and originates from metering stations which are managed by the Federal Office for the Environment in Switzerland. The goal is to group these stations according to the resemblance of their hydrologic temperature curve over a period of ten years with a ten-minute sampling rate of detail. Stations that exhibit very similar short-term as well as long-term temperature behaviour and evolution over time should be grouped into the same clusters. These clusterings should provide a better understanding of the data heterogeneity received from the various metering stations in Switzerland and support future decisions regarding the integration of new stations.

The first part of this work addresses the characteristics of time series data and surveys the field of pattern discovery techniques. The procedure of hierarchical clustering is explained in detail as it is the chosen technique applied for the cluster analysis of this thesis. Furthermore, four internal cluster validity indexes used to assess the quality of a cluster composition are elaborated.

The main part addresses the applied distance measuring strategies and assesses the quality of the received clustering results. Defining the level of similarity between two data objects is a fundamental concept in pattern recognition disciplines. This thesis elaborates the two shape-based strategies Pairwise Distance and Dynamic Time Warping and the feature-based strategy Discrete Wavelet Transformation. The cluster analyses are generated with different data aggregation levels and linkage methods. Finally, the various clustering approaches are challenged based on a forecast deviation analysis. This facilitates conclusions about the quality of the various cluster compositions in the form of quantifiable measures.

Keywords: Hydrology, Water Temperatures, Hierarchical Clustering, Time Series

Declaration of Authorship

I, Roman BÖGLI, the undersigned declare that all material presented in this paper is my own work or fully and specifically acknowledged wherever adapted from other sources. I understand that if at any time it is shown that I have significantly misrepresented material presented here, any degree or credits awarded to me on the basis of that material may be revoked. I declare that all statements and information contained herein are true, correct and accurate to the best of my knowledge and belief. This paper or part of it have not been published to date. It has thus not been made available to other interested parties or examination boards.

Signed:

Date:

Foreword

This thesis represents the final work to receive the Bachelor of Science in Business Information Technology from the School of Business at University of Applied Sciences and Arts Northwestern Switzerland (FHNW). During the course of this part-time study program over four years, I encountered a great variety of inter-connected subjects regarding information technology, economics, and business management. These learnings and the concurrently collected experiences on my working place as software developer provided a great balance and allowed me to deepen my knowledge in various disciplines.

The choice for this thesis was mainly driven by my search for programmable tasks that allow to produce something with a hands-on methodology. Preceding this paper was the creation of a Python library used to not only conduct the various cluster analyses at free level of parametrization but also to visualize them. The source code of this library was handed over to the Bern University of Applied Sciences which is the contractor of the Hydrology Division at the Federal Office for the Environment regarding this underlying project. The major problem to be solved at the beginning originated from the large amount of data to be processed. Since pattern analyses are rich of parametrization possibilities, finding superior models usually require lots of runs with the same data basis. Therefore, it is of great importance to keep the calculation times as low as possible. I could solve this issue successfully thanks to various code optimization measures. The library will be reused to conduct the same cluster analyses that are presented in this thesis on further data sets of equal size.

I would like to thank Dr. Vidushi Christina Bigler for her professional supervision and great teaching during the course of this thesis. It was a great pleasure to work alongside her and the team from the Institute for Optimisation and Data Analysis at Bern University of Applied Sciences.

Moreover, I would like to thank Dr. Thilo Herold and Dr. Adrian Jakob from the Hydrology Division at the Federal Office for the Environment for the data delivery and the support in domain specific questions.

In addition, I would like to thank Stefan Rey for introducing me to the world of programming a few years ago. The gained experiences working alongside him were decisive to enrol in this study program.

Finally, I would like to thank my family and my beloved partner for their support of all kinds during the course of writing the thesis.

Contents

1 Introduction

The Federal Office for the Environment (FOEN)[1] analyses several environmental aspects of Switzerland. The Hydrology Division focuses on water bodies such as lakes, rivers, and rills. The monitoring of water temperatures over long time periods belongs to one of the most important metrics as many environmental aspects rely on it. This accurate long-term data collection forms one of the key responsibilities of the hydrology division as it allows to conduct research on subjects such as rising water temperatures.

Since Switzerland counts numerous water bodies in various sizes, it requires a large amount of metering stations in order to capture insights from all different areas. At the moment, the FOEN maintains approximately 80 metering stations, of which 60 serve as a data basis for this thesis. The locations of these 60 metering stations are shown in the appendix in Figure A.1 and listed with additional information in Table A.1.

All stations record water temperature, discharge, and level at its individual water body location. The sampling frequency is consistently regulated at a ten minutes interval. Although the initiation date of monitoring varies between 1971 and 2015 as more stations were installed over time, concise water data is available over several decades. Data sets that index a value of a metric over time are commonly referred to as *time series*.

Besides the federal metering stations, the cantons of Switzerland maintain more than 700 additional stations. As is often the case in Switzerland, measurement policies vary in the different Cantons and any data mining technique will need to account for these deviations. Another difference concerns the locations of the stations. While federal stations mainly cover medium to large water bodies, the cantonal stations measure data alongside rather small to tiny waters.

Since the integration of a cantonal metering station into the federal monitoring network is connected to costs of several types (administration, labour, expenses), a wise prioritization is of interest. Cantonal stations, which enhance the diversity of the federal monitoring network and hence bear greater added value, should be integrated with higher priority. In order to create such a prioritization, the FOEN launched a project in cooperation with the Bern University of Applied Sciences[2] with the goal of grouping metering stations according to their similarity between each other. Besides new holistic insight regarding the existent data, it will facilitate the detection of cantonal stations which diversify the federal network the most. This thesis contributes to this project by performing and analysing such groupings.

[1] Bundesamt für Umwelt (BAFU)

[2] Berner Fachhochschule (BFH)

1.1 Project Description

This section outlines the major three parts of the named project regarding the forthcoming integration of cantonal metering stations into the federal network.

In the first part, the data from the cantonal metering stations is undertaken a quality assessment test. Since the metering strategies of the cantons may deviate from the federal one, it is important to analyse the existing data statistically and contextually. The received insights will be documented as they may serve as explanatory factors for consecutive analysis results.

The second and main part of the project focuses on the grouping of stations. A pattern recognition methodology is used called *clustering*. In order to conduct a cluster analysis of a given set of data objects (e.g. metering stations), each object must be described using a set of features. In statistical vocabulary, these features are also referred to as *predictors* and the process of finding such is also known as *feature engineering*. There exist numerous ways to derive predictors from data objects. Therefore, this project pursues three-stages feature engineering approach. In the first stage, the data objects will be described using simple statistical metrics such as mean, standard deviation, extreme values, and correlations. In a second stage, predictors are engineered by comparing and modelling the time variation curves (e.g. hydrographs). The third and last stage focuses on long-term trends using regression models. After each feature engineering stage, cluster analyses are conducted, and the resulting cluster compositions are assessed in quality.

The project's third part concerns the visualization and documentation of the elaborated work from the previous parts. As this project is publicly funded, the final results will be published alongside executive explanations in order to facilitate broad understanding.

1.2 Contribution

This section declares the aspects of the previously described project to which this thesis contributes to. As suggested by this document's title, the focus lies on the actual cluster analyses of water temperature data. Preliminary to this, three different methods to determine the similarity between two time series are elaborated. These methods are referred to as *distance measuring strategies*.

The first method includes the rather trivial approach of *pairwise distances* (PDIST) between two time series. The way this works is identical to the task of determining the distance between two points in a two-dimensional vector room. What makes time series more special, however, is the fact that each sampling moment represents one dimension. It results that this distance determination happens in a multi-dimensional vector room.

As a second method, the *Dynamic Time Warping* (DTW) algorithm is applied. This approach represents a prominent technique often encountered in time series analysis. The reason for this lies in the fact that DTW allows considering non-linear time lags in the comparison process of two signals that are indexed over time. As a consequence, two signals that share nearly identical shape, but express non-linear time lags will be declared as more similar than they would by the PDIST approach for instance.

The third method pursues the approach of extracting the most prominent components of a time series using *Discrete Wavelet Transformation* (DWT). Generally speaking, the water temperature data indexed over time can be handled as a signal consisting of multiple different sub-waves. DWT allows decomposing this signal into its constituent parts which eventually can be used to compare signals with each other. In particular, the 100 most effective coefficients will be used as features in order to describe a station, respectively the time series that it produces.

In order to perform the actual grouping of metering stations, *hierarchical clustering* with different *linkage methods* is used. The obtained cluster compositions are evaluated by means of different cluster quality assessment techniques. The essential goal of this thesis is to identify which distance measuring strategy and clustering parametrization lead to superior results. The ground truth, however, is unknown which is why it is hardly possible to claim a result as the best result.

1.3 Motivation

This section aims to demonstrate the virtue of cluster analyses and its resulting insights by means of an example drawn from the economic sphere. Roelofsen (2018) examined several different approaches to cluster public companies according to the behaviour of their share value traded on the stock market. Figure 1.1 shows the result of re-enacting this undertaking using the Standard & Poor's 500 (also known as S&P 500) large-capital companies traded on American stock exchanges.

From a technical point of view, share prices do not differ from water temperature data. Both data sets exhibit a metric value indexed over time and thus can be managed as time

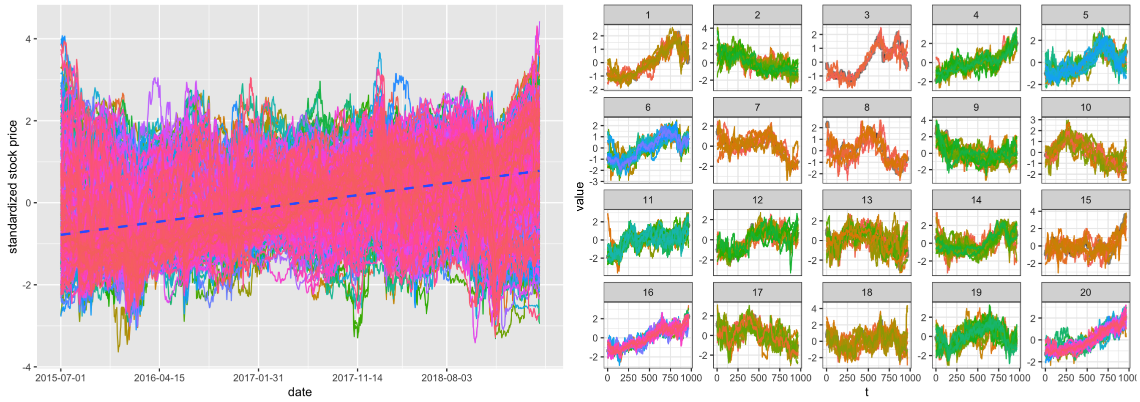


FIGURE 1.1: Clustering of S&P 500 Companies.
The cluster analysis disentangles the time variation curves of 500 companies' share prices into 20 clusters. (Image by V. Bigler)

series. However, it is recommended to normalize stock prices before any further treatment since the value domains may vary significantly.

Once the clustering is performed, the emerged groups embracing different disjoint sets of stocks can be investigated. A first insight is the revelation of the major curve shapes which are present in the underlying data set. Consequently, individual companies that exhibit a declining, increasing, or swaying trend in their share index can be indicated. This allows, for instance, to align an investment strategy on less prominent alternatives as a cluster's leading company. Since they all share resembling stock market developments, such a strategy is likely to result in a resembling return on investment as well. Also, a very diverse investment strategy can be established by embracing stocks from the most distinctive clusters.

The rest of this thesis is structured as follows. Chapter 2 describes the most important time series characteristics as well as common analysis techniques. In Chapter 3, clustering in general and hierarchical clustering specifically is explained in order to facilitate the understanding of the successive chapters. The concrete strategies applied in this thesis to define the disparities among the stations are documented in Chapter 4. Chapter 5 summarizes and interpreters the received results. Finally, the most important insights of this thesis are concluded and discussed in Chapter 6.

2 Time Series

In this Chapter, some important characteristics of time series are presented. Although there are disciplines in which the understanding of these properties is more relevant compared to cluster analyses (e.g. time series forecasting), knowing them provides a better understanding of the underlying data.

When comparing two different time series with each other, the question of *dependency* should be addressed as statistical models usually assume complete independence among the analysed data objects. This dependency relationship must be addressed when comparing two different time series. In terms of water temperature data, this is not truly the case. As an example, the water temperatures measured close to a spring located in an area with a high degree of glaciation influence the temperatures measured at the same water body some kilometres downstream in a less icy area. This dependency relation, however, does not apply in the opposite direction. To illustrate these dependencies, for instance, the set of metering stations could be represented as a directed graph where edges represent the dependency, weighted by the distance among the stations. All the remaining properties presented in this chapter concern the individual time series and can be assessed with no references to others.

Most time series represent highly *autocorrelated* data. Autocorrelation describes the degree to which observations over time correlate to itself and thus represents an individual measure per time series (Brownlee, 2017). Looking at water temperature data, for instance, the correlation originates from the fact that the temperature measured at time t_i is influenced by the previous measurement at time t_{i-1}. It is rather obvious that the temperature values measured in the present and in the close future (e.g. ten minutes later) will share great similarity and therefore are subject to autocorrelation. However, there are exceptions where this is not the case. Data representing the amount of withdrawn money from a cash dispenser over time may serve as an example for a type of time series which is subject to low correlation. Two different time series can also evidence correlation. For example, almost all water bodies in the northern hemisphere show some correlation.

Time series can be decomposed into the three components *trend*, *seasonality*, and *residuals*. Some definitions also include *level* as component which represents the mean value of the entire time series (Brownlee, 2017, p. 11). It must be said time series may not exhibit all these components. The changing level over time is considered as trend and recurring behaviour is represented as seasonality. The latter is not to be confused with cyclic behaviour which describes rises and falls at variable frequencies as they often occur

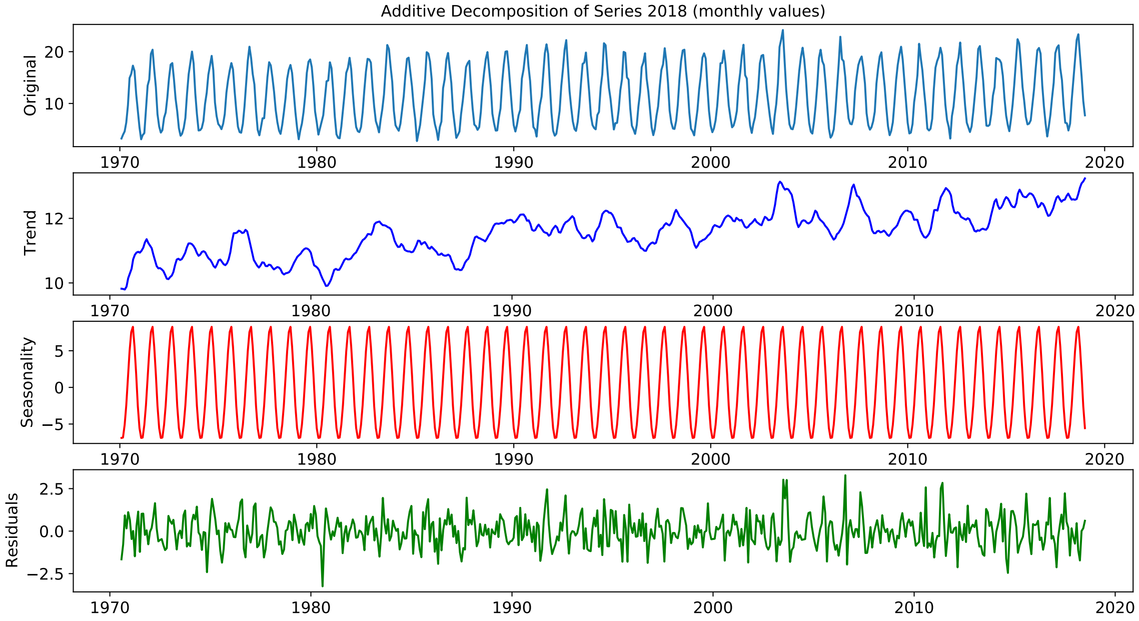


FIGURE 2.1: Time Series Decomposition
Decomposition of the water temperatures measured at station "Reuss - Mellingen" (ID: 2018) into trend, seasonality, and residuals (noise) over a period from 1971 to 2018.

in economic metrics (Hyndman & Athanasopoulos, 2018). The remaining information classifies as residuals which is commonly referred to as *noise*.

For decomposing a time series into these components, either an *additive* or *multiplicative* method can be used (Pal & Prakash, 2017). Figure 2.1 shows an additive decomposition which is defined as a function $y(t) = T_t + S_t + R_t$, whereas the temperature at time t is the sum of the components mentioned, namely trend (T), seasonality (S), and residuals (R). Level, on the other hand, is excluded in this equation as one can argue it is already encoded in the trend.

Another method to disaggregate time series is called *multiplicative* decomposition and formulates $y(t)$ as product of $[T_t, S_t, R_T]$ instead. While trend and seasonality behave linearly in the additive decomposition, the multiplicative method allows modelling non-linear behaviour i.e. increase or decrease of changes over time. The latter should be taken into consideration when the magnitude of seasonality depends on the magnitude of $y(t)$. Air passenger data from flight traffic provides a prominent example of such a relation. The more people fly, the more extreme the seasonal spikes become.

The rest of this chapter focuses on how to isolate the components trend, seasonality, and residuals.

2.1 Trend

Trend is probably the most intuitively interpreted component of a time series. Especially in finance, long-term development receives special attention as it forms an important investment decision factor. Despite this prominence, it is not always straightforward to detect a trend by human eye. When time series are subject to no trend at all, they are called *trend stationary* (Brownlee, 2017). Constant mean and variance over time are indicators for this. Forecasting models such as Autoregressive Integrated Moving Average (ARIMA) for instance, assume stationariness in time series since it is a sign of statistical consistency over the entire observation. There exist two common strategies to detect trend in time series.

The first is based on the bare data visualization using *moving* or *rolling averages*. This concept averages values of a series in a fixed window and moves this window alongside this series. The intensity of smoothness can be steered by choosing an appropriate windows size, which is the number of values to average. This window will trace the time series from beginning to end, resulting in a seemingly continuous trend visualization. Generally, rolling means plots start with a lag equal to the number of data points specified in the window size since these are the minimum required values in order to calculate the first mean value. In a *centred* moving average, however, this lag is reduced as the window extends over past and future values by half. An alternative to the moving average, a technique called *Locally Estimated Scatter-plot Smoothing* (LOESS) can be used which represents a trend by determining a fitting line based on a data point in the local area (James, Witten, Hastie, & Tibshirani, 2013).

A second method to determine whether or not a time series is trend stationary is called *Augmented Dickey-Fuller Test* (ADFT). This test introduced by Dickey and Fuller (1979) extends the class of statistical unit tests (Pal & Prakash, 2017).

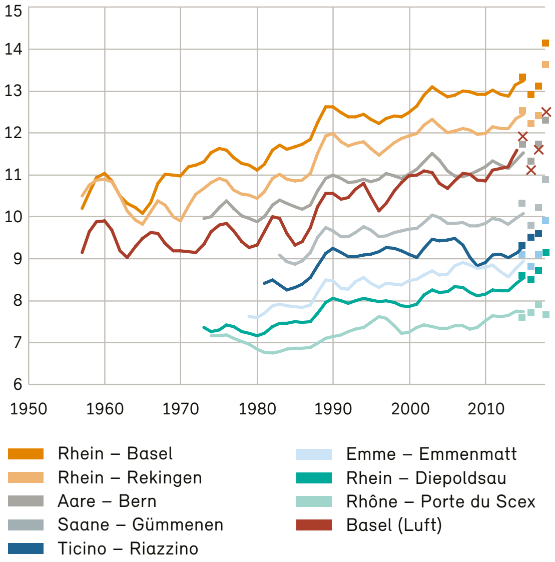


FIGURE 2.2: Rising Water Temperatures
The lines visualize a rolling mean over a period of seven years of the water temperatures measured in Celsius (vertical axis) over time (horizontal axis) for a selection of nine water bodies. The squares represent the annual means of the last four years, respectively the crosses but for air temperature (BAFU, 2019, p. 28).

2.2 Seasonality

A popular technique to search for seasonality is by means of an *autocorrelation* plot. It represents the similarity of observed values over an increasing time lag. In other words, it shows the correlation between the observed value on day i and the value on day $i - L$, whereas L denotes the size of the lag measured in number of time units present in the data. When performing an autocorrelation, the value which represents how the temperatures are correlated with a shifted version of itself is computed for an increase lag L. This technique can simultaneously be used for trend detection as the overall autocorrelation trend is associated with the actual trend in the inspected time series.

Looking at Figure 2.3, it seems reasonable that at the beginning with $L = 1$ the correlation between the two value sets is close to perfectly positive since one cannot expect significant change within 1 day of lag. The opposite is present with $L = 180$, meaning a lag of ½ year. This strong negative correlation originates from the fact that at this point warm summer temperatures correspond to cold winter temperatures. The length of a complete cycle can be read from the autocorrelation plot as well by looking at the lag between two peaks. The case shown in the figure implies that a whole thermic season for the station "Grossbach - Einsiedeln" (ID: 2635) lasts about one year.

To visualize the seasonality in the same purity as shown in Figure 2.1, a virtual cycle is created using the mean values of all observations captured during the same cyclic time. To provide an example, the average of all temperatures measured on 1st of January in the considered time period represents the cycle's temperature on that day. The entire cycle is concatenated a user-defined number of times resulting in a uniform chain of cycles.

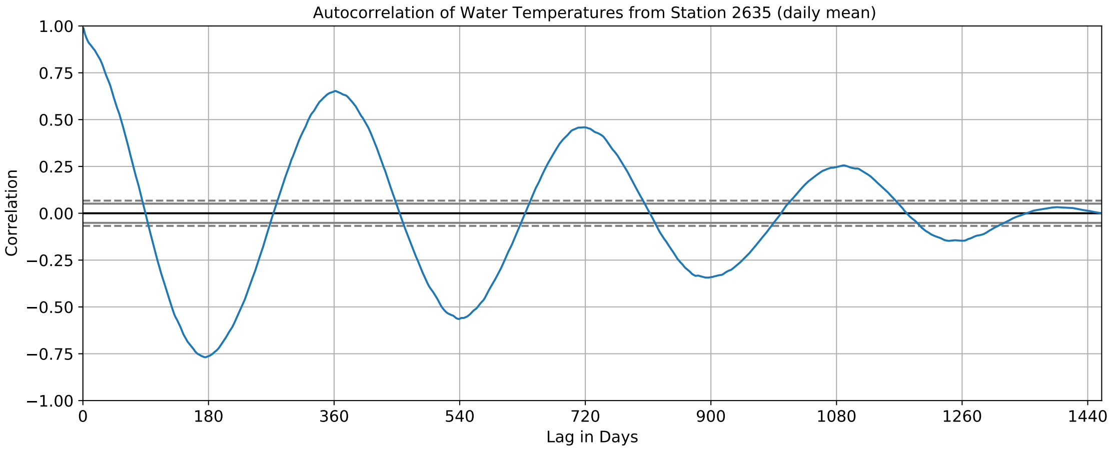


FIGURE 2.3: Autocorrelation
This autocorrelation was performed using daily mean water temperatures captured at station "Grossbach - Einsiedeln" (ID: 2635) over a period of four years from 2015 to 2018. The sinusoidal development of the correlation between the temperatures with an increasingly lagged version of itself attests that this data is subject to seasonality as it swings above and below the confidence interval of 95%. However, the correlation becomes less significant with increase lag as the amount of comparable data points decreases.

2.3 Residuals

Once trend and seasonality are identified, the remaining components are known as residuals. This is done by resolving the decomposition equation for the unknown R_t. Residuals must exhibit strong evidence of randomness i.e. they are uncorrelated or white noise as otherwise the two preliminary components trend and seasonality may not be extracted completely. When applying an autocorrelation analysis on the pure residuals, for instance, no pattern should be discoverable anymore.

3 Clustering

A common goal in data analysis is the automatised discovery of patterns which in some cases are undetectable by human eye. Research in the field of *pattern recognition* (PR) is comprehensively conducted as the practical applications of it deliver valuable benefits. That is, for instance, the automatic detection of spam emails, determination of a cell's cancer potential, or the classification of creditworthy customers.

This chapter will therefore provide a brief introduction to the different PR families. The focus, however, lies on *hierarchical clustering* since this is the technique chosen for the grouping of metering stations in this thesis.

3.1 Pattern Recognition Families

Before addressing the different approaches in PR, it is useful to declare the common aspects of it. Generally, the goal is to separate or partition a landscape of data objects into classes or clusters based on resembling features. The objects embraced in the same group should share high similarity (intra-class) and ideally show high dissimilarity to data objects embraced in other groups (inter-class) (Mann & Kaur, 2013). In pattern recognition, the techniques used to determine how similar two data objects are referred to as *distance functions*. Generically spoken, a distance function f_d takes two data objects $\{x, y\}$ as input and returns a scalar $f_d(x, y) = s$ which is referred to as distance. A small distance value indicates that the two data objects share high similarity or low dissimilarity while a high distance value expresses low similarity or high dissimilarity. Fully qualified distance metrics satisfy four properties, as described by (Cullinane, 2011). These includes non-negativity, symmetry, and triangle inequality.

Statistical PR represents the first and probably the most scientifically elaborated family (Jain, Duin, & Jianchang Mao, 2000; Webb & Copsey, 2011). It divides into *supervised* and *unsupervised* PR, which is derived from the fact of whether or not a *learning* or *training set* is available. In both categories, the data objects are described by the same number of features and a *target feature* that represents the predicted class or cluster number.

Supervised PR is commonly referred to as classification. A pre-labelled learning set is used in order to train a model that is capable of labelling or classifying new unseen data object accordingly. Pre-labelled in this context means that the target feature is known for the data objects in this set (i.e. the ground truth). Money lending data provides a popular example of this kind of use case. Here, the data objects are individuals that apply for a loan. The data objects are described by features like age, yearly income, and assets. The target feature indicates whether or not someone is able to payback a granted loan. New individuals that share high similarity with data objects from the training set are likely to be classified with the same target feature. Prominent supervised PR algorithms are among others the *k-Nearest-Neighbour*, *decision trees*, or *support vector classifiers*.

Unsupervised PR techniques are used when no training data exists which could supervise the process of grouping similar objects together. Instead, one solely relies on the given data objects and its relative resemblance to each other. Customer categorization in online stores serves as an example. Features such as login frequency, average search time, and monthly revenue could describe the data objects in this domain. A cluster analysis may reveal a certain grouping pattern. The emerged clusters ideally justify with real-world explanations and therewith enhance the understanding of the business (e.g. one group contains mainly young students with rather low solvency). *Iterative Relocation Algorithms* (IRAs) such as *k-means*[1] or *k-medoids*, hierarchical clustering (further discussed in next section), or clustering performed using *minimum spanning trees* serve as examples for popular unsupervised PR techniques.

In structural PR, the underlying data is represented using graphs (Fu, 1974; Riesen, Jiang, & Bunke, 2010). This allows to not only analyse data objects atomically based on the describing features but also to include relationships among different data objects. This results in a more comprehensive and thus realistic representation of the data scenario. Picturing relationships is not possible in statistical PR which is seen as a disadvantage as there is barely real-world data with perfect inter-independence. On the other hand, structural pattern discovery algorithms are usually subject to higher computational complexity as they rely on graph theory (Rosen, 2019). Images for example can be structurally described by representing maximized pixel areas with similar colouring as *nodes* in a graph. The node's connections, so-called *edges*, refer to the boundaries between these color areas. One approach to determine the (dis)similarity of two graphs is to count the number of minimum manipulations required to transform one graph into the other, which is the so-called *graph edit distance* (Riesen, 2015).

Two more families can be declared. One focuses on the unification of statistical and structural PR and can thus be named as *Hybrids*. They aim to combine the benefits from these two methodologies while excluding the disadvantages. The other entitles the field of *End-To-End Learning*. Here, the significant difference to the other families is the fact that features are learned autonomously rather than being engineered by the operator. This is especially useful when working with images or tones as they exist to large extends in the domain of autonomous driving for example.

[1] applied in Figure 1.1

3.2 Hierarchical Clustering

As previously mentioned, hierarchical clustering belongs to the family of statistical unsupervised PR. Together with IRAs, they belong to a group of techniques that are very popular in cluster analyses. The concept of distance functions for (dis)similarity determination between data objects is crucial in both as well. They differ, however, in the way the resulting cluster allocation of data objects is evolved and presented at the end. The simple algorithmic processes make both approaches trivial to understand and are applicable with low computational effort. Numerous open-source programming libraries exist that implement these algorithms. One of the most known libraries is called *scikit-learn* (Buitinck et al., 2013; Pedregosa et al., 2011). This specific library was also used for the analyses in this thesis.

Hierarchical clustering algorithms generally receive three elements as an input, namely the set of data objects to be clustered, a concrete distance function f_d, and the *linkage* definition. The latter will be explained in the next section.

In a first step, a symmetric distance matrix is generated with a size equal to the number of data objects in the set denoted as n. The values in this matrix are calculated using f_d. The number calculation required is $\frac{n(n-1)}{2}$ as only every unique data object pair needs to be processed, excluding the distance to itself.

The resulting distance matrix could already be used to visualize similarities by applying conditional formatting on the individual distance values. *Heat maps* implement this strategy by colouring high values brighter than low ones. To provide an analogy, consider the data objects projected in the vector room as cities located on a map and the distance matrix as a lookup table for travelling times between them. The distance matrix allows establishing a road map where the data objects are located in relative distance to each other.

What follows next is the actual grouping of data objects. Here, two different types of algorithms exist that are either *agglomerative* or *divisive*. In agglomerative algorithms[2], each data object forms its own cluster at the beginning. The clusters are then gradually combined resulting in fewer clusters that contain more data objects. The combining order orients itself at the inter-cluster distances going from close to far. Technically, this process ends when just one cluster embraces all the data objects. Practically, however, this is not a desirable state as such a clustering does not deliver valuable insights. Divisive algorithms[3] reverse this process starting with one cluster that contains all the data objects and gradually dividing it into smaller clusters embracing fewer data objects. The process ends when each data object represents one cluster, which would be again not insightful.

Both agglomerative and divisive algorithms output clusters that represent disjoint sets of data objects which is why they are called *hard* clusters. On the contrary, *fuzzy* clustering

[2] also known as *agglomerative nesting* (AGNES)

[3] also known as *divisive analysis* (DIANA)

Data objects	d_1	d_2	d_3	d_4	d_5	d_6
1 cluster	0	0	0	0	0	0
2 clusters	0	1	1	0	0	0
3 clusters	2	1	1	0	0	0
4 clusters	2	0	0	1	3	1
5 clusters	2	4	3	0	1	0
6 clusters	5	4	3	2	1	0

TABLE 3.1: Hierarchical Cluster Composition
This table shows an example of how the hierarchy received from a hierarchical clustering algorithm could look like, given a distance matrix for six data objects. The digits$[0 - 5]$ refer to the cluster number and serve as identification. The total number of clusters is stated in the first column.

algorithms allow expressing the cluster memberships of a data object in the form of proportions. Another overlapping feature and predominantly the most important one is the fact that the entire cluster composition hierarchy is obtained. This reveals two advantages. First, it is recognizable in which order and at what stage groups of data objects merge, respectively divide. Second, since all the cluster compositions are recorded from minimum (one cluster for all data objects) to maximum (each data objects forms its own cluster), all of them can be used for quality assessments. This makes a brute-force approach i.e. testing all possibilities to find the ideal number of clusters comparatively cheap and thus more attractive.

An example of such a hierarchy can be seen in Table 3.1. It shows the hierarchical clustering of six data objects. The cluster membership of each data object is represented by numbers. In the first row, all objects belong to the same single cluster 0. With an increasing number of clusters, the label variety grows and reaches its maximum in the last row where every data object represents its own cluster enumerated as $[0 - 5]$. Since the labels only serve the purpose of cluster membership identification, changing labels as it is the case for d_2 and d_3 from 3 to 4 clusters, for instance, has no meaning.

At this point, it must be stated that such a cluster membership trace as shown in this table could also be derived using techniques from other PR families such as IRAs. However, there are two differences compared to hierarchical cluster algorithms. One is the higher computational effort as IRAs are usually not applied on distance matrix but work directly with the individual (multi-dimensional) data objects. The other difference is the lack of determinism[4]. IRAs use random starting points for the required number of clusters which is given as an input parameter.

[4] given input leads repetitively to the same output

3.2.1 Visualization

Once a hierarchical system of information has been established, the desire for visualisation arises. This is especially true in hierarchical clustering as it provides a more intuitive understanding of the similarities among the data objects compared to a heat map. Probably the most common approach to visualize hierarchical clusters is by means of a so-called *dendrogram*. An example is seen in Figure 3.1. It represents the hierarchical cluster composition in the form of a tree. The 55 data objects at the bottom represent the leaves connected by converging branches to the root of the tree at the top. Although the tree's orientation has no relevant meaning, it is usually drawn as shown or rotated through 90°.

The lengths of the lines connecting the data objects directly or grouped subsets of them play an important role. The longer such a connection line is, the more dissimilar the connected elements are. This positive correlation between line length and disparity is a second important insight provided by a dendrogram. Generally, high-level groupings are desirable as they indicate great differences between the sub-clusters underneath. This implies that the given data landscape can be apportioned into rather compact clusters that are located at decent distances to each other. Such clearly distinguishable clusters, however, are rarely produced when dealing with real-world problems.

Once a dendrogram is established, one can set a user-defined cutting point through the tree to receive a specific number of clusters. In Figure 3.1, this is demonstrated with the grey dashed line. The number of clusters equals the number of connected data object bundles or branches that would "fall down" when the tree is cut at this point. The illustrated example would therefore produce six clusters. These are namely the green,

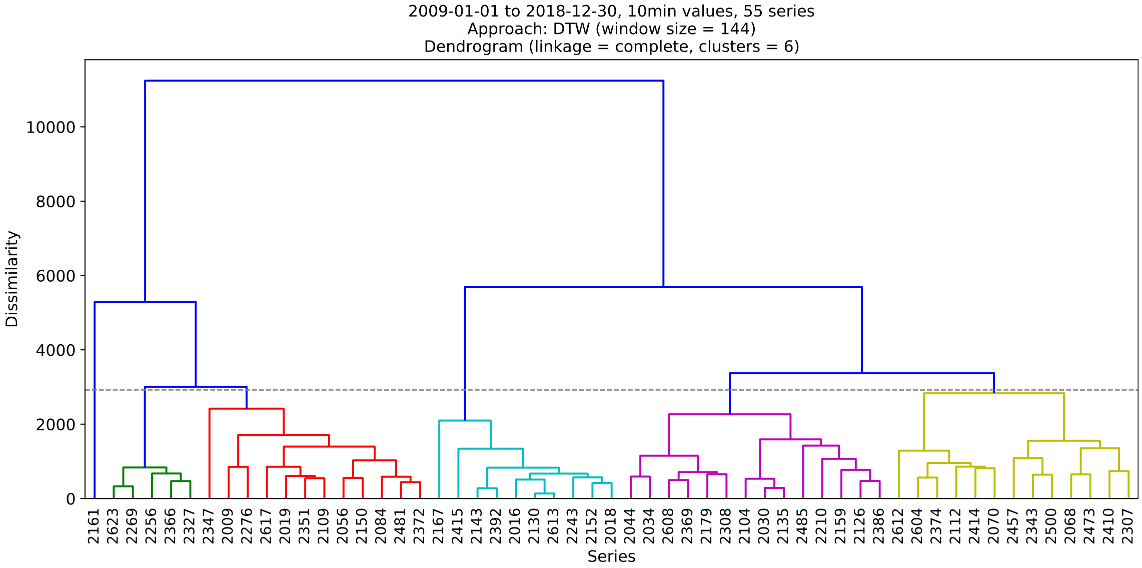


FIGURE 3.1: Dendrogram
Shows the hierarchical cluster composition of 55 metering stations using ten-minute values over 10 years. A cut line is set to highlight a composition consisting of six clusters. The vertical fork-like lines indicate the degree of dissimilarity among the sub-clusters being grouped. Higher lines indicate that the assembled or dissembled groups share greater dissimilarity.

red, cyan, magenta, and yellow bundle plus the blue one which points only to a single data object (station 2161, "Massa - Blatten bei Naters") representing its own cluster. This concrete grouping is visualized in Figure 3.2 by plotting the individual time series into their clusters of membership. For the sake of render efficiency, the plotted hydrographs in this thesis rely on weekly mean values.

Such a concrete representation can be constructed for every possible cut point in the dendrogram. This allows a comfortable perusal of the individual results including the course of mutation. With an increasing number of clusters, the overlaying hydrographic shapes per cluster are subject to a better alignment as the group size decreases and the remaining time series share a higher resemblance. Cluster 6 in the figure, for example, only holds one data object as the temperature measure at this station (i.e. its hydrographic shape) apparently is too unique to be aligned in any other group.

However, it must be stated that the shape alignment does not necessarily serve as a cluster quality indicator. The hierarchical grouping decisions solely rely on the given distance matrix which on the other hand was solely constructed using a specific distance measuring strategy. This strategy may declare two time series as highly similar although their hydrographic shapes do not verify this claim. It results that images as shown in Figure 3.2 should be interpreted with caution.

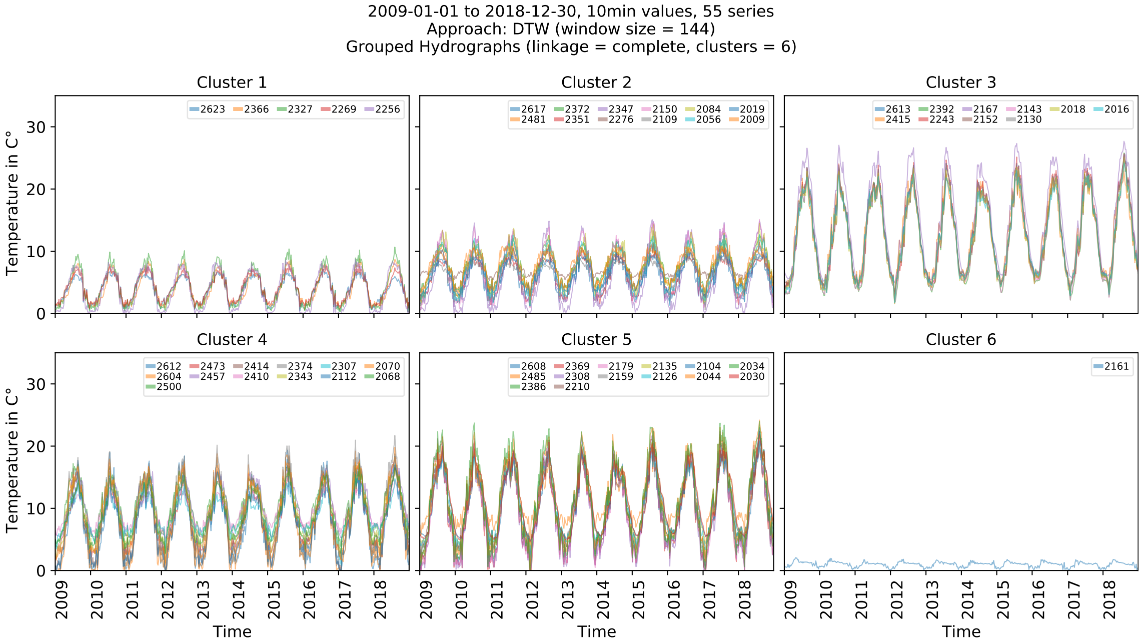


FIGURE 3.2: Clustered Hydrographs

This cluster composition was created based on the dendrogram and the given cut-point shown in Figure 3.1. Each cluster embodies a specific type of hydrographic shape which is an indicator for a successful clustering. The individual stations that generated these time series are declared in the legends of the sub-plots.

3.2.2 Linkages

The beginning of Section 3.2 outlined the input elements for a hierarchical clustering algorithm. Intentionally skipped in the course of explanation was the *linkage method*[5]. The linkage defines the methodology of the agglomerative composition approach. In other words, it specifies when which (bundles of) data objects are to be combined. *Single*, *complete* and *average* linkage are the most common methods (Webb & Copsey, 2011).

An agglomerative hierarchy clustering algorithm gradually combines the two sub-clusters which are closest to each other. In order to determine the next cluster pair to be combined, one must calculate a distance d_C between all possible cluster pairs. This distance, however, is not to be confused with the distance function f_d used to establish the initial distance matrix. While d_C is used for decision making during the cluster agglomeration process, f_d allows calculating the (dis)similarity of two data objects in the form of a value. The latter has already been applied in an antecedent stage in order to receive the distance matrix.

Always the two sub-clusters that minimize the distance d_C between them will be combined in the next iteration. The linkage criterion, however, defines the implementation of this inter-cluster distance method as shown in Figure 3.3. Single linkage defines, that the distance between two sub-clusters is measured using the closest data objects from each set[6]. Complete linkage focuses on the two most distanced data objects. In average linkage, the mean distance between all inter-cluster pairs is used.

All the cluster analyses conducted in this thesis are consistently performed in three versions using all the presented linkage methods. The reasoning behind this approach is the goal of covering a broader parametrization variety. Iterating through these three linkages during the clustering process is an especially cheap operation given the pre-calculated distance matrix.

The dendrogram shown in Figure 3.1 was build using complete linkage. For comparison purposes, see the dendrograms of the same scenario with average and single linkage in the annexed Figures A.3 and A.5.

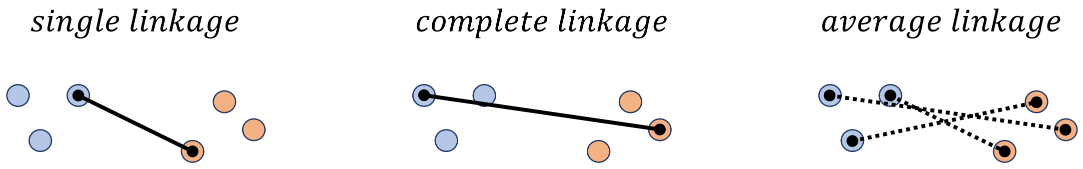


FIGURE 3.3: Three Different Linkage Methods
The linkage criterion defines the combining methodology in hierarchical clustering algorithms. In each hierarchical stage the two clusters with the shortest distance between them are combined. To receive this value, single linkage considers the least-distanced data objects of two different clusters, complete linkage the most-distanced, and average linkage the mean distance of all inter-cluster pairs.

[5] also called *linkage criterion*

[6] neighbours

3.3 Cluster Validity Indexes

The quality assessment of a conducted clustering is in contrast to classification problems more difficult since no ground truth exists which could serve as a reference. Another difficulty is to find the correct or ideal number clusters. There are various techniques to express the quality characteristics of clusterings in terms of a number which eventually allows differentiating a better clustering from a worst. These techniques are referred to as *Cluster Validity Indexes* (CVIs). They either persuade an *internal* or *external* validation methodology in order to provide insights about how good a specific cluster composition is relative to another. Internal indexes calculate a quality measure by exclusively relying on the partitioned data landscape while external indexes work with additional or external information such as prior knowledge about the data or even a reference partition (Wang, Wang, & Peng, 2009).

The two main properties assessed by internal metrics are the *compactness* and the *separation* of the individual clusters (Liu, Li, Xiong, Gao, & Wu, 2010). To understand both properties best, it helps to think of a two-dimensional vector space in which several data objects are grouped into a few clusters. The more similar the data objects in a distinct cluster are, the less dispersion is present which results in greater compactness. For the concept of separation, the distance between two clusters is observed. Very distanced clusters result in better separation. Superior clusterings minimize inter-cluster distances and maximize intra-cluster distances.

Although there are numerous CVIs available nowadays (Desgraupes, 2017), this thesis focuses on four internal ones. Since the underlying data for the cluster analysis is not associated with any prior cluster knowledge which could be used for external CVIs. The selection of indexes in this thesis was inspired by the most popular ones, namely the Calinski-Harabasz Index, Davies-Bouldin Index, Dunn Index, and Silhoutte Index. The rest of this section surveys all of them. The provided explanations rely on Desgraupes.

3.3.1 Calinski-Harabasz Index

The Calinski-Harabasz Index (CH) was proposed by Calinski and Harabasz in 1974. It measures the ratio between the average *within-group-scatter* (WG) and the average *between-group-scatter* (BG), where group refers to cluster. The scatter is defined as the sum of squares error SS of each data object in reference to a cluster's centroid (WG), respectively the barycentre of the whole data set (BG). The Euclidean distance serves as metric to determine the difference between to vectors $\|v_i - v_j\|$.

To determine the between-group-scatter sum of squares error $BGSS$, the deviation between each cluster's centroid m to the barycentre of the whole data set M is calculated, whereas K denotes the total number of clusters and n the number of data objects in a cluster acting as a weighting factor. This is formulated in Equation 3.1.

$$BGSS = \sum_{k=1}^{K} n_k \|m_k - M\|^2 \tag{3.1}$$

To determine the within-group-scatter sum of squares error $WGSS$, the deviation between each data object x in a cluster C to its centroid m_k is calculated. It is formally stated in Equation 3.1.

$$WGSS = \sum_{k=1}^{K} \sum_{x \in C_k} \|x - m_k\|^2 \tag{3.2}$$

Eventually, the index expresses the ratio between the two group-scatter metrics with respect of the degrees of freedom $N - K$ for $BGSS$ and $K - 1$ for $WGSS$. In Equation 3.3, N denotes the total number of data objects in the set while K expresses the cluster count.

$$CH(K) = \frac{BGSS}{WGSS} \cdot \frac{N - K}{K - 1} \tag{3.3}$$

As the number of clusters K increases, the number of data object x per cluster declines and with it the $WGSS$. Meanwhile, $BGSS$ increases since the number of clusters rises. Superior clusterings try to minimize $WGSS$ while maximizing $BGSS$. Hence, the CH is intended to be maximized. In other words, the higher the index value, the better the clustering.

3.3.2 Davies-Bouldin Index

The Davies-Bouldin Index (DB) was suggested by Davies and Bouldin in 1979. Here, each cluster k is compared to all other clusters in order to find Q_k, which is a maximum of the definition as

$$Q_k = \underset{k \neq k'}{max} \left\{ \frac{\delta_k + \delta_{k'}}{\|m_k - m_{k'}\|} \right\}, \tag{3.4}$$

where δ_k represents the mean distance of a cluster's data objects to its centroid m_k. The same metrics for k' refer to any other cluster but itself. Once Q_k has been found for each cluster, DB expresses the average of it given the total number of clusters K.

$$DB(K) = \frac{1}{K} \sum_{k=1}^{K} Q_k \tag{3.5}$$

Clusters that embrace their associated data objects tightly around its centre will obtain a relatively small δ_k. Simultaneously, clusters that are located very remote will exhibit large $\|m_k - m_{k'}\|$ distances which eventually leads to a small Q_k as a maximum. In a superior clustering, the average Q_k manages to remain small. Thus, the optimal cluster number can be found by minimizing DB.

3.3.3 Dunn Index

The Dunn Index (DI) represents the ratio between the minimal distance of two data objects from different clusters d_{min} and the maximal distance of two data objects within the same cluster d_{max} (Dunn, 1973). The latter metric is also referred to as diameter. The utilization of these extreme values reasons why this index is highly sensitive to outliers. Equation 3.6 states the DI formally.

$$DI(K) = \frac{d_{min}}{d_{max}} \tag{3.6}$$

The smaller the diameter d_{max} is, the less scatter is present within the clusters which in turn is a sign for high compactness. A large minimal inter-cluster distance d_{min} testifies great dispersion among the individual clusters indicating high separation. It results that superior cluster compositions maximize this index.

3.3.4 Silhouette Index

The Silhouette Index (SI) represents a fourth internal CVI. It was originally introduced by Rousseeuw (1987). In order to receive an intuition for this metric, it is worth decomposing it as follows.

For each data object, the mean distance to all other data objects in the same cluster $a(i)$ is calculated. The total number of data objects in the cluster is denoted by n_k. The value of $a(i)$ represents how well a data object x_i fits into the allocated cluster. It is formally defined in Equation 3.7.

$$a(i) = \frac{1}{n_k - 1} \sum_{\substack{i' \in C_k \\ i' \neq i}} f_d(x_i, x_{i'}) \tag{3.7}$$

Secondly, the dissimilarity of each data object to all other clusters is calculated, with the exception of the own member-cluster. To derive this value, the mean distance of a data object x of cluster C_k to all data objects of a cluster $C_{k'}$ is determined. The minimum mean distance is denoted by $b(i)$ which simultaneously uncovers the data object's closest neighbour cluster (see Equation 3.8). In other words, the bigger $b(i)$ becomes, the more remote this particular data object x_i is located from all other clusters. One can conclude that x_i demonstrates high dissimilarity compared to the rest. The variable $n_{k'}$ represents the number of data objects which do not belong to the same cluster as the data object x_i.

$$b(i) = \min_{k \neq k'} \frac{1}{n_{k'}} \sum_{i' \in C_{k'}} f_d(x_i, x_{i'}) \tag{3.8}$$

After $a(i)$ and $b(i)$ are determined, the silhouette width $s(i)$ can be calculated for each data object. This is formally expressed in Equation 3.9. This width will result in a value between -1 and 1. A value very close to 1 testifies that the underlying data object fits perfectly in its allocated cluster. On the other hand, a negative $s(i)$ implies that there is another cluster in which the data object would fit better.

$$s(i) = \frac{b(i) - a(i)}{max\left\{a(i), b(i)\right\}} \text{ if } n_k > 1 \tag{3.9}$$

After $s(i)$ is determined for all data objects in the set, one can calculate the mean silhouette width per cluster $S(k)$. Eventually, the sum of all $S(k)$ divided by the total number of clusters K represents the final value of the SI as shown in Equation 3.10. As a value $s(i)$ very close to 1 testifies that a data object is placed in the ideal cluster, the mean of the mean should also be close to 1. Therefore, superior clusterings maximize this index.

$$SI(K) = \frac{1}{K} \sum_{i=1}^{K} S(k_i) \tag{3.10}$$

4 Distances for Time Series

The characteristics of time series data as well as the algorithmic process of clustering algorithms, in particular hierarchical, have been addressed so far. This includes the representation of water temperature data over time as multi-dimensional data objects and the importance of distance measurement between such data objects in order to determine how similar these objects are. This chapter elaborates three different distance measurement strategies applicable to time series. An extensive review on this is provided by Aghabozorgi, Shirkhorshidi, and Wah (2015).

Beforehand, the general procedure is declared in order to provide an understanding of where the distance measurement is located in the overall process. After the data is loaded into the memory, there exist two intermediate steps before the actual distance calculations. That is the definition of the period in focus (e.g. ten years) and the aggregation level of the underlying data (e.g. daily mean values). All the subsequent tasks are predicated on the received distance matrix and hence have unified character. In other words, the only differencing task in the entire clustering process chosen strategy to create the distance matrix. What follows is the application of the hierarchical clustering algorithms with different linkages, building the individual cluster compositions, and finally assessing their quality.

The amount of data to be processed plays also an important role. In order to receive a distance matrix, all unique pairs of data objects in a set have to be compared. A set of n data objects requires $\frac{n(n-1)}{2}$ number of comparisons for this task. Thus, additional data objects increase the computational effort at a growing rate. Depending on the chosen distance measurement strategy, this could have serious negative repercussions on the performance.

The use of *code parallelization* qualifies to counteract this issue (Wolohan, 2020). This concept demands to organize all data objects in pairs alongside with the distance function to be applied without performing any kind of calculation yet [1]. Afterwards, this job list is passed to a pool of central processing units (CPU) which executes the predefined orders in parallel. One also speaks of *workers* in this context.

This asynchronous operating technique is essential when high computational performance is desired. Multiple processes start at the same time, each performing one comparison after another. The process is further accelerated the more random access memory (RAM) is available. Duplications are excluded as the pool instance only allocates unattended jobs to workers. The independently received results are eventually consolidated and ordered.

[1] comparable with a to-do-list

Thanks to this technique the computation time for one distance matrix with 55 series over ten years of ten-minute value could be reduced to a few seconds. It is to be mentioned, however, that all calculations were executed on a calculation server of BFH providing 88 CPUs with approximately 1 terabyte of RAM.

4.1 Pairwise Distance

The first concrete distance strategy describes the rather trivial approach of comparing two time series according to their sequentially ordered values. This strategy is simply called *Pairwise Distance* (PDIST) since it does not consider any aspects other than a *bijective*[2] mapping of the values with identical indexes from both series. PDIST belongs to the group of *shape-based* distances and is called a *lock-step* measure. As already mentioned, the values indexed over time received from one metering station is represented as a multi-dimensional data object, whereas each dimension captures one index. Given two time series t_1 and t_2 of daily mean water temperatures, for instance, PDIST compares the temperature captured at any date in t_1 with the value on the same day in t_2.

PDIST is actually a generic strategy that can be implemented with a range of concrete distance functions. The selected function, in this case, is the so-called *Euclidean* distance, which is defined in Equation 4.1. The distance between the two data objects a and b is defined by the rooted sum of all squared differences of all their dimensions n.

$$f_d(a,b) = \|a-b\|_2 = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{4.1}$$

The rationale behind this selection is based on the function's simplicity and popularity in the clustering domain. There exist many other distance functions that are applicable in the context of PDIST (Cha, 2007).

A prerequisite in statistical PR is that the two compared data objects are described with the same amount of attributes or dimensions. Applied on times series, this means the measuring period must exhibit the same number of values in order to allow such a comparison. However, the actual time periods may differ. Concretely this means that one could compare two time series with each other that capture data from two different decades as long as these series share the same value count. The results might be dubious and misleading but the technique can be applied.

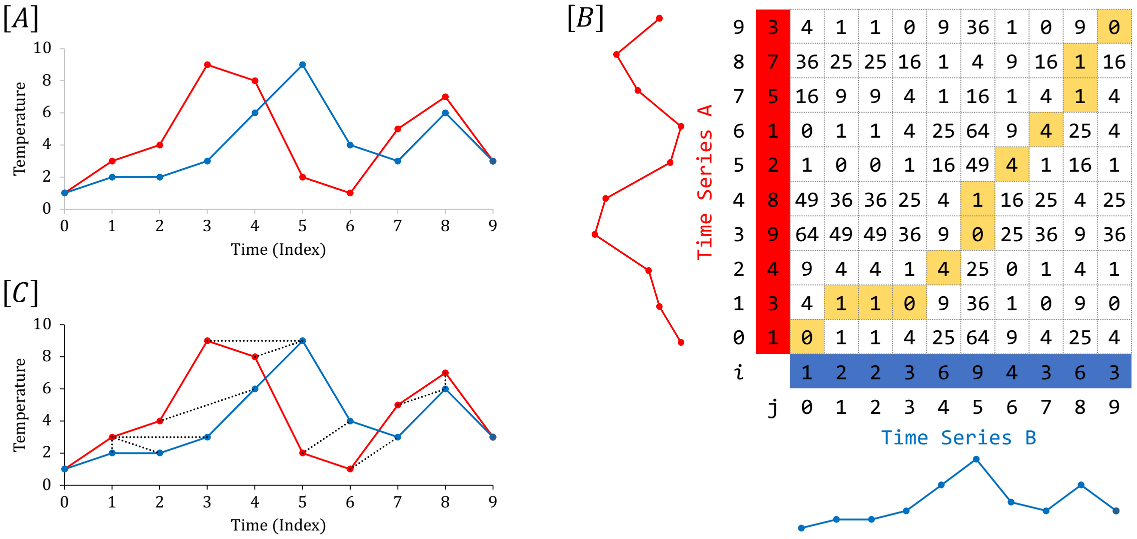
[2] one-to-one

4.2 Dynamic Time Warping

The second applied distance measuring is based on the *Dynamic Time Warping* (DTW) algorithm, which was introduced by Sakoe and Chiba (1978). DTW was originally constructed for speech processing but in the meantime has found large application in time series analysis (Nielsen, 2019). This strategy also belongs to the group of shape-based distances but is called an *elastic* measure.

Consider the scenario where two metering stations are located on the same water body. One of them, however, resides a couple of kilometres further downstream. Detected temperature changes at the first location (e.g. due to ice melting in spring) are likely to be observed at the second location as well, provided no other influential forces such as underwater springs or river mouths apply along the way. As a result, these two metering stations exhibit a very similar shape of their hydrographs[3] in terms of temperature. The only salient difference could be a non-linear shift of the shape due to irregularly lagged discovery at the second station. One could also say that the measurements are *warped* in time or vary in speed.

Despite these non-linear warps, the two described stations in this example show high similarity in their temperature behaviour which should be accounted for. The PDIST strategy would disregard this due to the sequential processing of the data object's dimensions. DTW on the other hand addresses this issue by establishing a *surjective*[4] mapping of the temperature indexes, as shown in Figure 4.1. Consequently, the distance between these two exemplary time series will correctly be declared as low, indicating high similarity.



9	3	4	1	1	0	9	36	1	0	9	0
8	7	36	25	25	16	1	4	9	16	1	16
7	5	16	9	9	4	1	16	1	4	1	4
6	1	0	1	1	4	25	64	9	4	25	4
5	2	1	0	0	1	16	49	4	1	16	1
4	8	49	36	36	25	4	1	16	25	4	25
3	9	64	49	49	36	9	0	25	36	9	36
2	4	9	4	4	1	4	25	0	1	4	1
1	3	4	1	1	0	9	36	1	0	9	0
0	1	0	1	1	4	25	64	9	4	25	4
i		1	2	2	3	6	9	4	3	6	3
	j	0	1	2	3	4	5	6	7	8	9

FIGURE 4.1: DTW Cost Matrix
The two time series shown in [A] are non-linearly lagged in time. The cost matrix in [B] allows to find a mapping path that minimizes the total costs and thus accounts time warps appropriately. The resulting surjective mapping is shown in [C]. (Inspired by Keogh and Ratanamahatana (2005))

[3] a graph showing the change of a hydrologic variable over time

[4] one-to-many

The algorithmic process of the DTW is divided into two steps. The following explanations rely on Keogh and Ratanamahatana (2005). The first step includes the creation of a cost matrix with $n \cdot m$ elements, whereas n and m correspond to the number of observations in the compared time series A and B. This can formally be expressed as follows.

$$\begin{aligned} A &= \{a_0,\ a_1,\ \ldots,\ a_i,\ \ldots,\ a_n\} \\ B &= \{b_0,\ b_1,\ \ldots,\ b_j,\ \ldots,\ b_m\} \end{aligned} \tag{4.2}$$

The individual cost is calculated using the distance function $d(a_i, b_j) = (a_i - b_j)^2$. The second step includes the finding of a mapping path P which consists of matrix elements seen as weights $\{w_0,\ w_1,\ \ldots,\ w_k,\ \ldots,\ w_K\}$. These elements are highlighted in yellow in Figure 4.1. There are three constraints that affect P to ensure a continuous monotone path evolution between the two corners of the matrix:

- The *boundary* constraint forces the path to start at the matrix element (i_0, j_0) and end at (i_n, j_m).
- The *monotonicity* constraint ensures a monotone path evolution between the two end points with $i_{s-1} \leq i_s$ and $j_{s-1} \leq j_s$, with s as index enumerator.
- The *continuity* constraint prevents time jumps with $i_s - i_{s-1} \leq 1$. The same applies to j.

It is common to introduce a fourth constraint that acts as boundaries for the expansion of the warping path. These boundaries imply that an index i only can be mapped with an index j in the range of $[j - u, j + v]$, given that $i = j$. The absolute sum of (u, v) is known as *window size*. A modest window allows reducing the computational effort as fewer mapping candidates exist. A window size of 1, however, transforms the DTW algorithm into the PDIST strategy. The rationale behind this windowing is inspired by the fact that also a possible time-shift has a certain limit. In terms of water temperatures, such a limit lies between one to three days. In other words, the window size defines the freedom of non-linear movement during the index pairing process.

Although there exist many different eligible versions of P, the DTW algorithm focuses on the path that minimizes the total cost. This subsequently declares the distance between the two candidate time series as formally defined below.

$$f_d(A, B) = min \left\{ \sqrt{\sum_{k=0}^{K} w_k} \right\} \tag{4.3}$$

To summarize, the DTW algorithm skews the dimension pairing in a way that potential non-linear time warps are addressed in a favoured manner. This eventually results in a

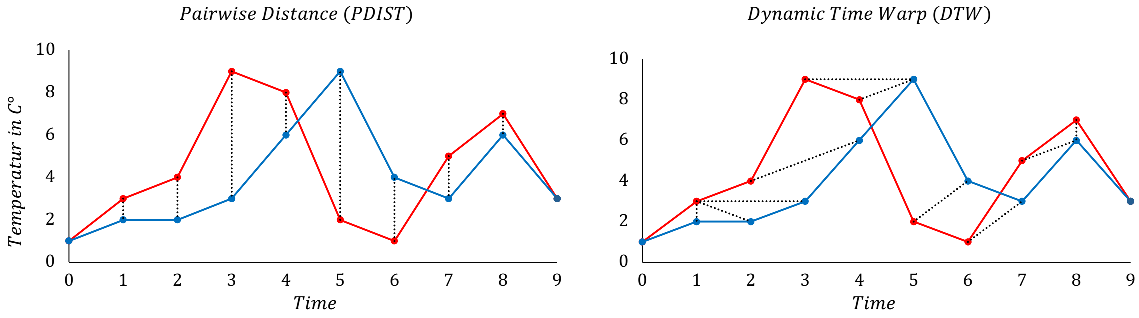


FIGURE 4.2: Time Warping Effect
Shows the subsequent bijective index mapping conducted during PDIST in comparison to the DTW approach that establishes a non-linear surjective mapping resulting in the warping effect.

higher and thus more realistic distance declaration of two time series that indicate similar values over time but at varying speed. The DTW implementation employed in this thesis uses the Euclidean distance. However, there are implementations that allow other distance metrics to apply, such as the *City Block* distance[5]. One also speaks of differently *flavoured* DTWs in these cases.

Figure 4.2 illustrates the DTW behaviour by setting it in contrast to the more trivial distance measuring strategy PDIST. The resemblance of the two time series shown in red and blue is captured more properly by addressing these time warps.

An important requirement for a DTW application is the fact that the compared time series must start and end at identical points in time. When this requirement is disrespected, for instance by comparing two time series (t_1, t_2) whereas t_2 represents a subset of t_1, the shorter series will be stretched in time to meet this requirement again. This event may influence the integrity of the comparison in a negative way. The treatment of time series with different lengths, on the other hand, does not represent an issue. This ability is even claimed to be another advantage of the DTW algorithm since it allows to further reduce the computational costs. However, there is no significant impact on the result's accuracy as shown by Ratanamahatana and Keogh (2004).

4.3 Discrete Wavelet Transformation

The third distance measurement strategy pursued in this thesis is based on the signal decomposition procedure called *Discrete Wavelet Transformation* (DWT) as conducted by Hong-fa (2012). In contrast to the previous two strategies, DWT belongs to the family of *feature-based* distance metrics since the actual form of the temperature curve is not taken into consideration. Instead, a time series is described by independent attributes or features with no notion of time. This implies that the individual feature values are all somehow derived from the original series and thus no longer share any order-defining index. An example would be the trivial description of a time series using its mean, extreme values, amplitude, phase, and average wavelength.

[5] also known as *Manhattan* distance

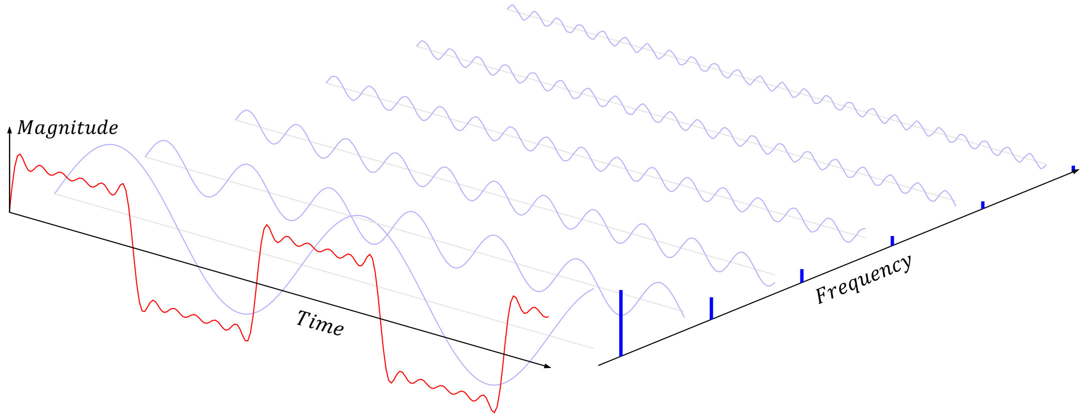


FIGURE 4.3: Signal Frequency Decomposition using DFT
The original signal in red is decomposed into composite sine waves illustrated in blue that, when added together, represent the original again. These building blocks differ in their frequencies as wells as magnitudes and thus possess different level of importance. (Image adapted from Roelofsen (2018))

It is helpful to explain the idea of DWT by means of the *Discrete Fourier Transformation* (DFT) which follows a similar concept. DFT aims to decompose a signal into its building blocks. Figure 4.3 illustrated this decomposition of the original signal in red into the various composite signals in blue ordered by their impact force. All of them represent infinite sine waves which is why DFT is best applicable on stationary data.

DWT on the other hand decomposes a signal by means of different discrete wavelets that serve filters. The signal is successively passed through these filters in order to receive coefficients that approximate the original time series in versatile levels of detail. Thanks to this passage, the point in time when this specific wavelet occurred is captured. This additional information describes the original signal more extensively as DFT and thus leads to better clustering results (Aghabozorgi et al., 2015). This seems to contradict the principle of a feature-based distance metric where chronological information is absent. Since it is only the most influential composite signals over the entire period, descending ordered according their coefficients, this time information is neutralized again. Furthermore, it is to mention that DWT should only be applied on stationary data as the individual wavelets only represent excerpts and thus possess no knowledge regarding an overall trend.

The size of these extracted sub-signal coefficients correlates with their importance, whereas large coefficients indicate highly influential sub-signals. The concrete implementation of this distance measurement strategy in this thesis extracts the top 100 most influential coefficients as describing features for a station. The filter used in the DWT implementation for this thesis is called *Haar* wavelet and represents the simplest of its kind (Haar, 1910).

As only the most influential signal components are selected, it becomes feasible to compare time series of different lengths. This dimensionality reduction has the consecutive advantage of de-noising the underlying data. The intensity of this effect is user-defined. For subsequent clustering tasks, an ideal level may be found by incrementalism.

5 Results

This chapter examines the cluster results received by applying the different distance measuring strategies explained in the previous chapter. It is comparatively easy to assess the performance of a classification model used for supervised PR problems since a ground truth is available. This does not apply to unsupervised PR problems such as cluster analyses. Nevertheless, there are still techniques to identify superior clusterings. It must be said, however, that the insights from these kinds of methods are not equally meaningful as the ones received in a supervised setting due to the fact that they express the qualities of the different cluster compositions relative to each other.

Besides the question of which clustering approach works best, it is also questionable what number of clusters is ideal. Compositions with a high number of clusters are generally not appreciated as they become meaningless. An extreme example would be the clustering of n data objection into K clusters with $n = K$. One rather demands a clustering method to yield superior CVIs values as soon as possible, meaning with a low number of clusters. This event aligns with the issue of determining the ideal number of clusters[1] as reviewed by Kodinariya and Makwana (2013). A common method is to pay attention to so-called *elbows* in the CVI plots. They occur when the index value exhibits a steep fall or rise at the beginning of increasing cluster numbers followed by a rather linear course.

The detection of elbows in CVI plots is rather simple by human eye but at a cost of subjectivity. Its automated determination using algorithms is possible, however, requires parametrization which again is subjective. The cluster quality assessments in this thesis focus on distinct indicators that are programmatically detectable such as global extreme values, rate of changes, or inflection points[2].

The rest of this chapter structures into two parts. Foremost it is explained, how to measure the quality of a single cluster with a more independent metric. Afterwards, it is shown how these metrics are further processed to finally compare the different clustering approaches. Although the visuals shown in this thesis mainly focus on one specific approach that is DTW on ten-minute data and with a window size 144, they were produced for all three distance measurement strategies and aggregation levels. An extensive visual comparison of all the different CVIs and how they performed per presented linkage method can be found in the attached Figures B.8, B.9, and B.10.

[1] also known as *true* number of clusters

[2] point where a curve transitions from convex to concave

5.1 Cluster Quality

As previously stated, a superior clustering maximizes both intra-cluster similarity and inter-cluster dissimilarity. Popular metrics to assess these characteristics are internal CVIs which were addressed in Chapter 3 and applied for all cluster compositions in this thesis. However, it is valuable to also test the cluster qualities with more independent metrics. These metrics are intentionally distinguished from CVIs as they embody domain-specific calculation processes.

One way to introduce such an independent metric is by means of *forecasting deviation*. Figure 5.1 visualizes this undertaking. The illustrated cluster composition was established using DTW as a distance strategy and complete linkages as a cluster aggregation method. The same scenario with average and single linkage is shown in the annexed Figures A.4 and A.6.

To measure this forecasting deviation, one first has to create a virtual series s_V by averaging all the time series in a cluster. This mean series can also be seen as a cluster representative. Subsequently, the first ⅔ of s_V (blue line) trains a forecasting model in order to predict the remaining ⅓ (red line).

The forecasting model can only be as good as the derived virtual time series representing this cluster. In this thesis, the rather trivial forecasting method of a repetitive yearly cycle

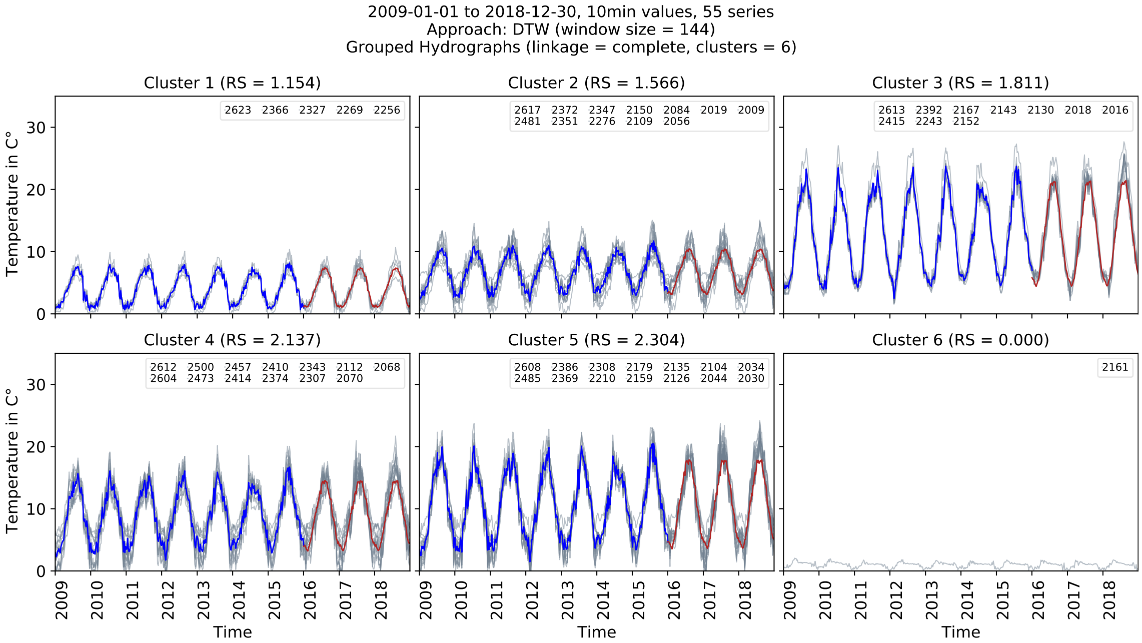


FIGURE 5.1: Clustered Hydrographs with Residual Scores
Shows the water temperatures series from 55 stations over the period of ten years grouped into six clusters. The first seven years of the mean series (blue) is used to predict the last three years (red). The cluster quality is eventually determined by assessing the misfit of this forecast compared to the actual values using the RMSE. The average misfit is expressed as Residual Score (RS) in the individual cluster titles.

created by averaging the cycles in the training period was applied. The more sophisticated model ARIMA could also be used here as demonstrated by Roelofsen (2018).

Lastly, the prediction accuracy is assessed by comparing the prognosticated temperatures with the actual ones from each time series in this specific cluster. This is accomplished using the popular concept of the *Root-Mean-Square Error* (RMSE)[3] for each comparison as formally defined in Equation 5.1.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{5.1}$$

The overall misfit is determined by the deviation between the measured temperature y and its prediction $\hat{y}$ under consideration of the total number of data points n. The RMSE is identified between all the time series in a cluster and its representative series s_V. The *Residual Score* (RS) describes the mean value of all RMSEs in this cluster. Formally, this can be expressed as $RS = \frac{\sum RSME}{n}$, whereas n denotes the number of time series in the specific cluster. This averaging is crucial as otherwise clusters with high n would automatically result in high RS.

Cluster 6 in Figure 5.1 indicates a special case as it incorporates only one station. The calculation of the RS becomes meaningless as no cluster representative s_V is required and thus equals zero. The hypothesis of having a superior cluster vindicates with a low RS as then the prediction misfit is minimized. Since the prediction is solely based on s_V, the grouped time series in this cluster must share great similarity.

The RS per cluster builds the basis for three consecutive quality metrics. This allows to assess the different cluster composition from a second and more independent perspective than the one received through the internal CVIs discussed in Section 3.3. Thanks to the hierarchical clustering method applied in this thesis it is especially inexpensive to create all the different cluster compositions to cover the entire range of cluster count. With an increasing number of clusters, the average RS declines as the groups become more specific and thus lose their generalization that causes misfit. An example of this is shown by means of a dendrogram with cut point at 15 clusters in the annexed Figure A.7 and as cluster composition in Figure A.8.

The first independent quality metric is called *Mean RS I* and averages the RS of all clusters in a composition. Its formal definition is stated in Equation 5.2, with K representing the number of clusters in this composition.

$$Mean\ RS\ I = \frac{1}{K}\sum_{i=1}^{K} RS_i \tag{5.2}$$

[3] also known as *Root-Mean-Square Deviation* (RMSD)

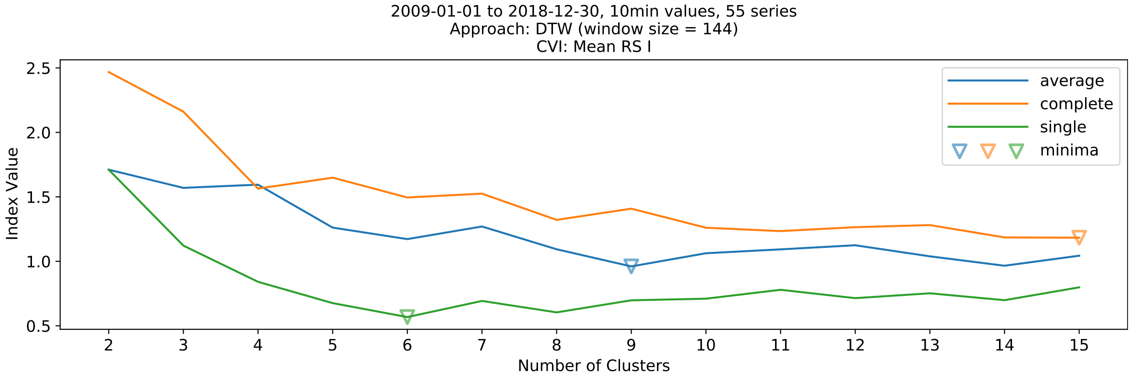


FIGURE 5.2: Cluster Quality Assessment using Mean RS I
Quality assessment using the mean of all cluster's RS. Since lower RS indicates a smaller forecasting deviation, superior cluster compositions exhibit rapid reach of the Mean RS I minimum.

How the Mean RS I evolves per linkage method over increasing number of clusters is visualized in Figure 5.2. As previously stated, the RS value can be seen as average misfit between the forecast virtual mean series and the concrete series in the clusters. The lower this indicator is, the higher the resemblance of the grouped series which is attests high cluster quality. Mean RS I is thus to be minimized.

Single linkage outperforms the other two linkage methods. Single linkage typically produces stairs-alike dendrograms whereas very particular data objects are sequentially aggregated at the upper end closer to the root (see Figure A.5). Lowering the cut-point in these kinds of hierarchical clusterings separates the most unique data objects at the beginning as clearly visible in Figure A.6. As a consequence, many clusters bearing only one series and thus having an RS of zero are created at the beginning which causes the superior Mean RS I.

The second metric based on the RS is derived from previously described one and thus called *Mean RS II*. The motivation for this derivative originates from the situation with clusters where $RS = 0$. Mean RS II counteracts this behaviour as it converts the existence of single-member clusters into a penalty q as formulated in Equation 5.3. This punishing factor q denotes the number of clusters with only one member. Therewith, the divisor that downsizes the sum of all RS in a specific cluster composition is weakened by subtracting it from the total number of clusters K.

$$Mean\ RS\ II = \frac{1}{K - q} \sum_{i=1}^{K} RS_i \tag{5.3}$$

The effect of this change is shown in Figure 5.3. Single linkage is now clearly outperformed by the other two linkage methods. However, all of them converge at around 10 clusters. At this stage, the cluster composition starts to resemble each other irrespective of the linkage.

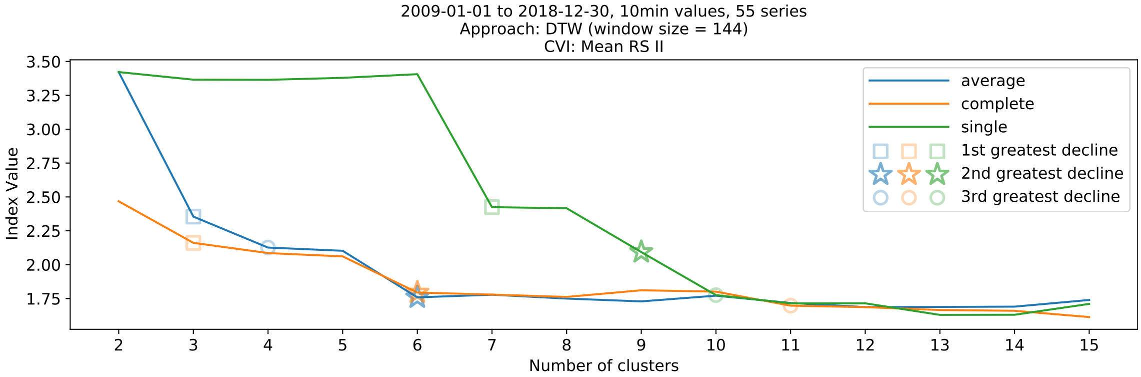


FIGURE 5.3: Cluster Quality Assessment using Mean RS II
Quality assessment using the mean of all cluster's RS, whereas single-member clusters receives punishment. As usually the greatest decline emerges at the beginning, the focus is set to the second greatest decline to highlight the elbow.

The second greatest decline in the evolution of the Mean RS II offered to be a useful point of interest to determine an ideal cluster number as it approximately highlights the elbow. The greatest and third greatest decline are also marked.

The third and last cluster quality metric is called L_p *Norm*. The *norm* of a vector describes its extension in space which is why it also denoted as vector *length* (Savov, 2017). The formal definition is stated in Equation 5.4. Here, a cluster composition x is described as a vector using the RS of all clusters K. The chosen implementation set $p = K$.

$$L_p\ Norm = \|x\|_p = \left(\sum_{i=1}^{K}(RS_i)^p\right)^{\frac{1}{p}}, \text{with } p = K \tag{5.4}$$

The way this index evolves over increasing cluster numbers is shown in Figure 5.4. The first inflection point of the index evolution is chosen to be an optimal cluster number indicator.

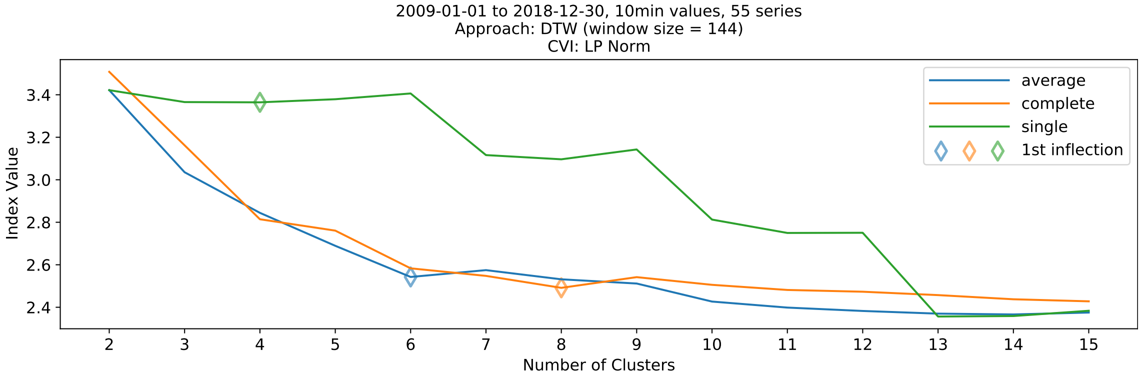


FIGURE 5.4: Cluster Quality Assessment using L_p Norm
Quality assessment using the individual RS for a vectorial representation. The length or norm of this vector serves as a quality factor, whereas lower values mean better clusterings. The first inflection point highlights the optimal cluster number.

5.2 Approach Comparison

So far the clustering quality with potential ideal cluster numbers was evaluated between the different linkage methods. Next, the same assessment is conducted among the different clustering approaches. The goal is to receive insights about what distance measuring technique demonstrates superiority in the clustering of water temperature data.

The candidate approaches in this thesis represent three different distance metrics on three different data aggregation levels which totals in $3^2 = 9$ scenarios. It should be mentioned that this selection only represents a subset of all the possible clustering approaches especially under the consideration of all plausible parameter settings. Decisions on parameters such as the distance function in PDIST, the window size in DTW, or the wavelet in DWT represent entire subjects of scientific research itself.

To further reduce the combination complexity in the subsequent approach comparison, the focus is narrowed on one linkage method instead of all three. The process of elimination relies on the forecast deviation evaluation discussed in the previous section. Namely these are Mean RS I, Mean RS II, and L_p Norm. Table 5.1 shows the average index value per clustering approach (first column) and linkage method. Since all three metrics are to be minimized, the lowest index values are coloured in green while the highest are coloured red.

Looking at Mean RS I, it is single linkage that performs significantly better compared to the other two linkages. The reason for this behaviour, as mentioned, lies in the early sorting of peculiar stations. Complete linkage performs consistently worst, while average linkage is just slightly better.

Roughly the contrary appears in the Mean RS II column, where the existence of single-member clusters is penalized. The advantageous conditions single linkage enjoys in Mean RS I appear as severe disadvantages now. Complete linkage excels here as the hierarchical aggregation process produces high-level groupings. This barely leads to

	Average Mean RS I			Average Mean RS II			Average L_p Norm		
Approach	average	complete	single	average	complete	single	average	complete	single
DTW_daily_w7	1.164	1.476	0.723	1.864	1.80	2.292	2.501	2.573	2.848
DTW_hourly_w24	1.222	1.499	0.823	1.941	1.859	2.437	2.605	2.656	2.981
DTW_10min_w144	1.212	1.500	0.812	1.951	1.86	2.432	2.605	2.657	2.979
DWT_daily_haar_euc	1.344	1.505	1.052	1.711	1.766	1.862	2.497	2.558	2.582
DWT_hourly_haar_euc	1.439	1.538	1.043	1.834	1.832	1.957	2.611	2.615	2.654
DWT_10min_haar_euc	1.394	1.666	0.953	1.958	1.906	1.954	2.712	2.696	2.774
PDIST_daily_euc	1.200	1.440	0.775	1.871	1.786	2.396	2.486	2.530	2.849
PDIST_hourly_euc	1.235	1.520	0.800	1.969	1.887	2.399	2.579	2.638	2.902
PDIST_10min_euc	1.236	1.524	0.800	1.970	1.892	2.399	2.579	2.639	2.903
Total Average	1.272	1.519	0.865	1.897	1.843	2.236	2.575	2.618	2.830

TABLE 5.1: Mean Quality Index per Clustering Approach
Compares the different clustering approaches per linkage method by means of the quality indexes Mean RS I, Mean RS II, and L_p Norm. The values represent the average index over the evolution from 2 to 15 clusters. As all three metrics are to be minimized, the lowest value per row and quality metric is highlight green while the highest is coloured in red.

single-member clusters at the beginning of the clustering process where the total cluster number is low. It is also noticeable that average linkage performs nearly identical.

According the L_p Norm it is the average linkage that holistically performs best but directly followed by complete linkage. This confirms the harmonic influence of a cluster aggregation strategy based on averages as the consequences of outliers are dulled. As a conclusion, solely average linkage is used for the final approach comparison.

The scope for the approach comparisons is further narrowed by choosing one out of three value aggregation levels. The process of elimination is identical to the one applied to the linkage methods. Table 5.2 shows the average performance per aggregation level and quality metric. The total average (last row) indicates the identical values as in Table 5.1 because the data set for these calculations remained the same but pivoted.

The aggregation level based on daily values (first row) consistently shows better performance over all metrics. This most possibly due to the fact of less noisy data. The level of detail provided with hourly or even ten-minute values seems to be too high. It is to remember that the processed timeline extends to 10 years. With a granularity of ten-minute values, a time series hold $10 * 365 * 24 * 6 = 512'640$ data points. Insight creation from such detailed information can become distracted or washy by noise. Daily mean values seem to be a better choice in this context. Besides all this, working with a higher aggregation level reduces the data load significantly and thus increases the performance on calculations.

The evolution of these quality indexes per applied distance measuring strategy is shown in Figure 5.5. This cross-strategy comparison, however, has some disadvantages that are discussed in Chapter 6. The remaining clustering approaches, of which all received daily temperature values as an input, are DTW with a window size of seven days, DWT using the Haar wavelet to extract the 100 most influential signal components for consecutive Euclidean distance determination, and lastly, PDIST applying the Euclidean distance directly on the time series. The points of interest are also highlighted using the minimum value for Mean RS I, the second greatest decline for Mean RS II, and the first inflection point for L_p Norm. The complete comparison on average linkage with all data aggregation levels can be found in the annexe from Figure B.1 to B.7.

	Average Mean RS I			Average Mean RS II			Average L_p Norm		
Aggregation Level	average	complete	single	average	complete	single	average	complete	single
daily	1.236	1.474	0.85	1.815	1.784	2.183	2.494	2.554	2.76
hourly	1.299	1.519	0.889	1.915	1.859	2.264	2.598	2.636	2.846
10min	1.281	1.563	0.855	1.96	1.886	2.262	2.632	2.664	2.885
Total Average	1.272	1.519	0.865	1.897	1.843	2.236	2.575	2.618	2.83

TABLE 5.2: Mean Quality Index per Value Aggregation Level
Compares the three value aggregation levels daily, hourly, and ten-minute per linkage method by means of the quality indexes Mean RS I, Mean RS II, and L_p Norm. The values represent the average index over the evolution from 2 to 15 clusters. As all three metrics are to be minimized, the values per row and quality metric are highlighted in green for lowest and red for highest.

Holistically, one can say that DTW and PDIST exhibit a very similar course of index evolution in all three quality metrics. Most noticeably is the outburst from this alignment in Mean RS I between 7 and 11 clusters. Afterwards, PDIST even performs better than DTW.

DWT on the other hand presents a salient deviation from the others. This can be explained by the fact that this feature-based distance measurement strategy is conceptually different compared to the two shape-based strategies DTW and PDIST. In both Mean RS I and II, DWT reaches the point of interest very early at a cluster number of five.

All three approaches behave more or less equally in the L_p Norm, which decreases the ability to extract valuable insights. It is to remember, however, that this norm helped to decide on the prior linkage limitation.

Overall one can conclude three things. First, DTW does not outperform the more trivial strategy of PDIST to a remarkable extend. The application of DTW should therefore be carefully reconsidered, at it requires a higher computational effort to conduct. The second conclusion to be drawn refers to the cluster numbers. The point of interest in all three metrics lies somewhere between five and eleven clusters. This evidence can be used to contain the range for the true cluster number which might be helpful for later subject-specific analyses in the field of hydrology. Lastly, DWT demonstrated that a competitive clustering also can be established with a reduced amount of information. Good results do not automatically come with more data, in fact, noise starts to spread which may worsen the outcome.

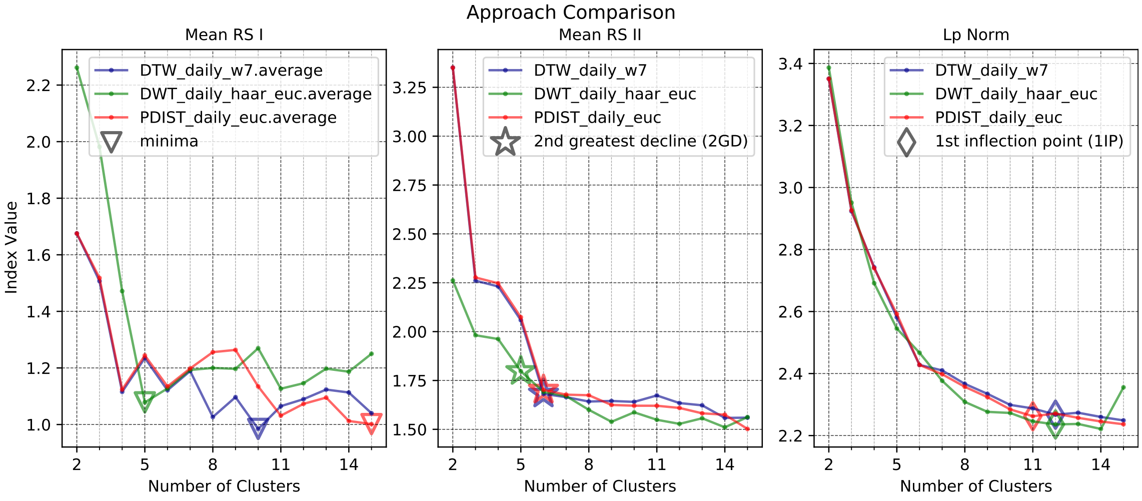


FIGURE 5.5: Approach Comparison
Shows the three distance measuring strategies DTW, DWT, and PDIST in comparison. For sake of readability, only the clustering approaches based on daily values and average linkage are shown.

6 Discussion

The work presented in this thesis persuaded the goal of grouping water temperature metering stations by means of resulting time series data. The characteristics and analysis methods of time series were elaborated. The actual grouping was performed using the statistical PR approach of hierarchical clustering with three different distance measurement strategies and linkage methods. The resulting cluster compositions were assessed using internal CVIs. These compositions resulted from the various clustering approaches using the different distance measurement strategies with input data on different aggregation levels. Finally, a selected subset of the applied approaches was compared by analysing the forecast deviation with the help of three different metrics that are based on these residuals.

A first point to be reconsidered is that the presented work did not cluster water bodies but water temperature data mined at FOEN metering stations on strategically interesting locations on water bodies. Differentiation is crucial here as the thermic insights received at a metering station does by no means necessarily represent the entire or partial water body. The annexed Figure A.2 supports this claim as it shows the temperature heterogeneity that can exist in a rather close area of a water body. It results that cluster allocation of a metering station may drastically be influenced by the choice of its location. However, the FOEN has great interest to position these stations in order to maximise the water body representativeness.

A second discussion point regards the conducted approach comparison. The method of using the residual scores from the forecast deviation is from a statistical perspective slightly misleading. The temperature prognosis that acted as a reference to determine the RMSE per time series in a cluster was solely constructed by averaging the first ⅔ of these cluster members. The average cycle of this reference is then perpetually continued for the last ⅓ where total misfit (RS) compared to all cluster members is calculated. The cluster compositions created using DTW is by design disadvantaged in this quality assessment. DTW is more likely to group metering stations that exhibit a time-shift in their temperature series than PDIST. An extreme example would be two stations that exhibit a shift by a half phase in their recorded time series. The PDIST approach, on the other hand, does not address time lags at all. Therefore, PDIST would rather allocate these non-linearly shifted series into different clusters. This is an unfair advantage compared to DTW as PDIST will generally yield a better RMSE value. Therefore, the shown comparison should strictly speaking only be conducted between approaches with identical distance measuring strategies.

In order to build a statistically more representative approach comparison, the use of an external CVI is indispensable. It would also allow comparing approaches among different distance measuring strategies. An external CVI requires a manual cluster composition performed by humans with high-level of domain knowledge that could serve as a ground truth. Consequently, one could compare all the generated clustering from all kinds of approaches with this single ground truth in order to declare a superior technique. The crux in this methodology, however, is that this human-generated ground truth will be the leading force in the nomination process of a superior clustering. This intersects with the conception that computational pattern discovery algorithms should provide humans with new insights.

Several points could be addressed in future work. The results have shown that the underlying data is likely to be noisy. Therefore, it would be interesting to conduct the same clustering approach but with high data aggregation levels (e.g. weekly or monthly). This might be especially helpful when clustering periods that exceed a period of ten years. One could even try to work with semi-annual or annual data when focusing on periods that extend over multiple decades. Furthermore, one could subdivide the periods into shorter sequences and address them separately with the same data set. This would allow to observe how the time series may become grouped with different companions over the course of these two subsequent periods. Finally, one could extend the repertoire of distance measuring strategies to the area of structural PR recognition.

The project initiated by the FOEN is still ongoing and further distance metrics for time series will be evaluated. Hence, it is too early to deliver a distinct recommendation for the best clustering technique to be applied. However, this thesis allowed to gain first experiences regarding advantages and disadvantages of the elaborated strategies and produced a Python library which can be applied to conduct and parametrise the same cluster analyses with the data received from cantonal metering stations. Together with the explanations provided in this work, it supports the process of finding an ideal prioritization regarding the incorporation of cantonal metering stations into the federal network.

Bibliography

Aghabozorgi, S., Shirkhorshidi, A. S., & Wah, T. Y. (2015). Time-series clustering – A decade review. *Information Systems*, *53*, 16–38. doi:10.1016/j.is.2015.04.007

BAFU. (2019). *Hydrologisches Jahrbuch der Schweiz 2018 - Abfluss, Wasserstand und Wasserqualität der Schweizer Gewässer* (tech. rep. No. UZ-1907-D). Bundesamt für Umwelt (BAFU), Abteilung Hydrologie. Bern.

Brownlee, J. (2017). *Introduction to Time Series Forecasting With Python: How to Prepare Data and Develop Models to Predict the Future*. Machine Learning Mastery.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., . . . Varoquaux, G. (2013). API design for machine learning software: Experiences from the scikit-learn project. In *Ecml pkdd workshop: Languages for data mining and machine learning* (pp. 108–122).

Calinski, T., & Harabasz, J. (1974). A DENDRITE METHOD FOR CLUSTER ANALYSIS. *Communications in Statistics - Theory and Methods*, *3*(1), 1–27. doi:10.1080/03610927408827101

Cha, S.-H. (2007). Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, (4), 300–307.

Cullinane, M. J. (2011). Metric axioms and distance. *The Mathematical Gazette*, *95*(534), 414–419. doi:10.1017/S0025557200003508

Davies, D. L., & Bouldin, D. W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*(2), 224–227. doi:10.1109/TPAMI.1979.4766909

Desgraupes, B. (2017). *Clustering Indices*. University Paris Ouest, Lab Modal'X.

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *Journal of the American Statistical Association*, *74*(366a), 427–431. doi:10.1080/01621459.1979.10482531

Dunn, J. C. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, *3*(3), 32–57. doi:10.1080/01969727308546046

Fu, K. S. (1974). *Syntactic Methods in Pattern Recognition*. New York: Academic Press.

Haar, A. (1910). Zur Theorie der orthogonalen Funktionensysteme: Erste Mitteilung. *Mathematische Annalen*, *69*(3), 331–371. doi:10.1007/BF01456326

Hong-fa, W. (2012). Clustering of Hydrological Time Series Based on Discrete Wavelet Transform. *Physics Procedia*, *25*, 1966–1972. doi:10.1016/j.phpro.2012.03.336

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice* (2nd ed.). OTexts.

Jain, A., Duin, P., & Jianchang Mao. (2000). Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 4–37. doi:10.1109/34.824819

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. doi:10.1007/978-1-4614-7138-7

Keogh, E., & Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. *Knowledge and Information Systems*, *7*(3), 358–386. doi:10.1007/s10115-004-0154-9

Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of Cluster in K-Means Clustering. *International Journal of Advanced Research in Computer Science and Management Studies*, *1*(6). Retrieved from http://www.ijarcsms.com/

Liu, Y., Li, Z., Xiong, H., Gao, X., & Wu, J. (2010). Understanding of Internal Clustering Validation Measures. In *2010 IEEE International Conference on Data Mining* (pp. 911–916). doi:10.1109/ICDM.2010.35

Mann, A. K., & Kaur, N. (2013). Review Paper on Clustering Techniques. *Global Journal of Computer Science and Technology*, *13*(5), 42–48.

Nielsen, A. (2019). *Practical Time Series Analysis: Prediction with Statistics & Machine Learning* (1st ed.). ISBN: 978-1-4920-4165-8. Sebastopol, California: O'Reilly.

Pal, A., & Prakash, P. (2017). *Practical Time Series Analysis: Master Time Series Data Processing, Visualization, and Modeling using Python*. ISBN: 978-1-78829-419-5. Packt Publishing.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Ratanamahatana, C. A., & Keogh, E. (2004). Everything you know about Dynamic Time Warping is Wrong. In *Third Workshop on Mining Temporal and Sequential Data*, Seattle, WA: University of California.

Riesen, K. (2015). *Structural Pattern Recognition with Graph Edit Distance: Approximation Algorithms and Applications* (Springer International Publishing AG, Ed.). doi:10.1007/978-3-319-27252-8

Riesen, K., Jiang, X., & Bunke, H. (2010). Exact and Inexact Graph Matching: Methodology and Applications. In C. C. Aggarwal & H. Wang (Eds.), *Managing and Mining Graph Data* (Vol. 40, pp. 217–247). doi:10.1007/978-1-4419-6045-0_7

Roelofsen, P. (2018). *Time series clustering* (Doctoral dissertation, Vrije Universiteit Amsterdam, Amsterdam, Netherlands).

Rosen, K. H. (2019). *Discrete Mathematics and Its Applications* (Eighth edition). New York, NY: McGraw-Hill Education.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. doi:10.1016/0377-0427(87)90125-7

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *26*(1), 43–49. doi:10.1109/TASSP.1978.1163055

Savov, I. (2017). *No Bullshit Guide to Linear Algebra* (2nd ed.). ISBN: 978-0-9920010-2-5. Minireference Co.

Wang, K., Wang, B., & Peng, L. (2009). CVAP: Validation for Cluster Analyses. *Data Science Journal*, *8*, 88–93. doi:10.2481/dsj.007-020

Webb, A. R., & Copsey, K. D. (2011). *Statistical Pattern Recognition* (3rd ed.). Hoboken, NJ, USA: Wiley.

Wolohan, J. T. (2020). *Mastering Large Datasets with Python: Parallelize and Distribute Your Python Code.* ISBN: 978-1-61729-623-9. Manning Publications Company.

List of Abbreviations

ADFT	**A**ugmented **D**ickey **F**uller **T**est
AGNES	**Ag**glomerative **Nes**ting
ARIMA	**A**utoregressive **I**ntegrated **M**oving **A**verage
BAFU	**B**undes**a**mt **f**ür **U**mwelt
BFH	**B**erner **F**ach**h**ochschule
BG	**B**etween-**G**roup-**S**catter
BGSS	**B**etween-**G**roup-**S**catter **S**um of **S**quares
CH	**C**alinski-**H**arabasz Index
CPU	**C**entral **P**rocessing **U**nit
CVI	**C**luster **V**alidity **I**ndex
DB	**D**avies-**B**ouldin Index
DFT	**D**iscrete **F**ourier **T**ransformation
DI	**D**unn **I**ndex
DIANA	**D**ivisive **Ana**lysis
DTW	**D**ynamic **T**ime **W**arping
DWT	**D**iscrete **W**avelet **T**ransformation
FHNW	**F**ach**h**ochschule **N**ord**w**estschweiz
FOEN	**F**ederal **O**ffice for the **En**vironment
IRA	**I**terative **R**elocation **A**lgorithm
LOESS	**L**ocally **E**stimated **S**catterplot **S**moothing
PDIST	**P**airwise **Dist**ance
PR	**P**attern **R**ecognition
RAM	**R**andom **A**ccess **M**emory
RMSD	**R**oot-**M**ean-**S**quare **D**eviation
RMSE	**R**oot-**M**ean-**S**quare **E**rror
RS	**R**esidual **S**core
S&P	**S**tandard **&** **P**oor
SI	**S**ilhouette **I**ndex
WG	**W**ithin-**G**roup-**S**catter
WGSS	**W**ithin-**G**roup-**S**catter **S**um of **S**quares

List of Figures

List of Tables

A Metering Stations

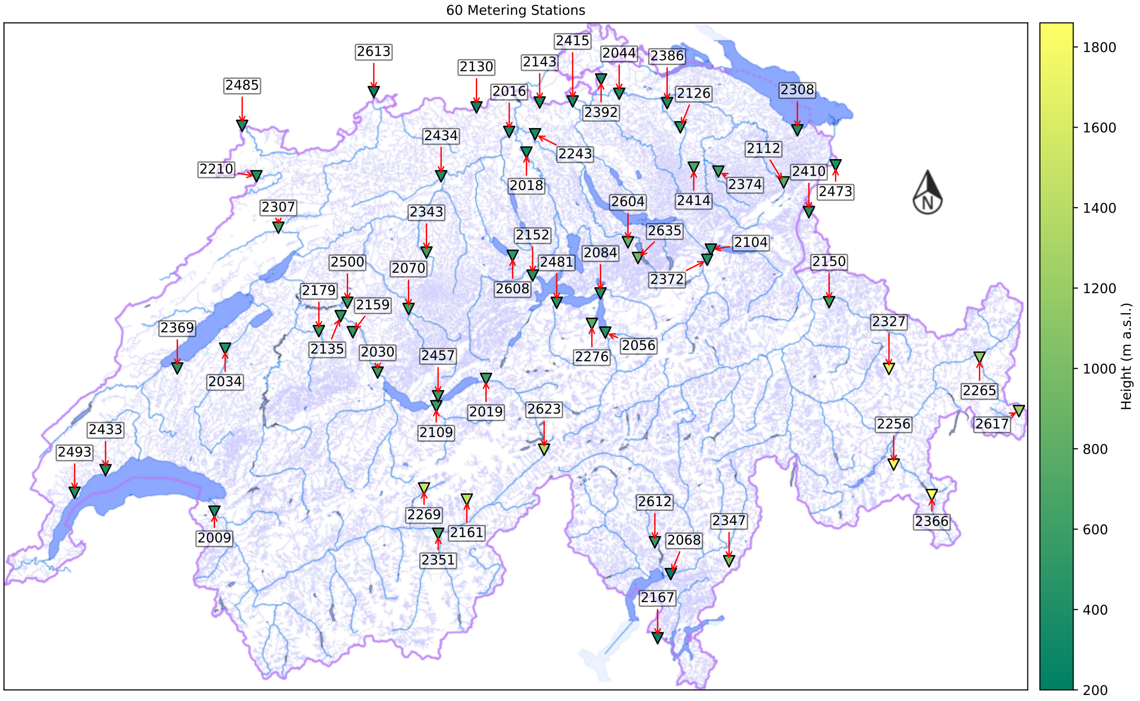


FIGURE A.1: Geo-Referenced FOEN Metering Stations
This map shows the location of 60 FOEN metering stations in Switzerland. The elevation from sea level is illustrated using a green-to-yellow fade, whereas green implicates a low-lying location.

ID	Name	Starting Year	Elevation (m)	Glaciation (%)
2009	Rhône - Porte du Scex	1974	377	11.1
2016	Aare - Brugg	1974	332	1.5
2018	Reuss - Mellingen	1974	345	1.8
2019	Aare - Brienzwiler	1974	570	15.5
2030	Aare - Thun	1971	548	6.9
2034	Broye - Payerne, Caserne d ’aviation	1976	441	0.0
2044	Thur - Andelfingen	1974	356	0.0
2056	Reuss - Seedorf	1974	438	6.4
2068	Ticino - Riazzino	1977	200	0.1
2070	Emme - Emmenmatt, nur Hauptstation	1976	638	0.0
2084	Muota - Ingenbohl	1974	438	0.0
2104	Linth - Weesen, Biäsche	1974	419	1.6
2109	Lütschine - Gsteig	1986	585	13.5
2112	Sitter - Appenzell	2005	769	0.1
2126	Murg - Wängi	2002	466	0.0
2130	Rhein (Oberwasser) - Laufenburg	1971	299	1.1
2135	Aare - Bern, Schönau	1974	502	5.8
2143	Rhein - Rekingen	2003	323	0.4
2150	Landquart - Felsenbach	2003	571	0.7
2152	Reuss - Luzern, Geissmattbrücke	1973	432	2.8
2159	Gürbe - Belp, Mülimatt	2006	522	0.0
2161	Massa - Blatten bei Naters	2003	1’446	56.5
2167	Tresa - Ponte Tresa, Rocchetta	2002	268	0.0
2179	Sense - Thörishaus, Sensematt	2003	553	0.0
2210	Doubs - Ocourt	2002	417	0.0
2243	Limmat - Baden, Limmatpromenade	2002	351	0.7
2256	Rosegbach - Pontresina	2004	1’766	21.7
2265	Inn - Tarasp	2016	1’183	3.5
2269	Lonza - Blatten	1986	1’520	24.7
2276	Grosstalbach - Isenthal	2004	767	6.7
2307	Suze - Sonceboz	2004	642	0.0
2308	Goldach - Goldach, Bleiche, nur Hauptstation	2004	399	0.0
2327	Dischmabach - Davos, Kriegsmatte	2004	1’668	0.7
2343	Langeten - Huttwil, Häberenbad	2002	597	0.0
2347	Riale di Roggiasca - Roveredo, Bacino di compenso	2003	980	0.0
2351	Vispa - Visp	2002	659	23.1
2366	Poschiavino - La Rösa	2004	1’860	0.0
2369	Mentue - Yvonand, La Mauguettaz	2002	449	0.0
2372	Linth - Mollis, Linthbrücke	1974	436	2.9
2374	Necker - Mogelsberg, Aachsäge	2007	606	0.0
2386	Murg - Frauenfeld	2006	390	0.0
2392	Rhein (Oberwasser) - Rheinau	1974	353	0.6
2410	Liechtensteiner Binnenkanal - Ruggell	1999	435	0.0
2414	Rietholzbach - Mosnang, Rietholz	2002	682	0.0
2415	Glatt - Rheinsfelden	1976	336	0.0
2433	Aubonne - Allaman, Le Coulet	2010	390	0.0
2434	Dünnern - Olten, Hammermühle	2013	400	0.0
2457	Aare - Ringgenberg, Goldswil	1980	564	12.1
2473	Rhein - Diepoldsau, Rietbrücke	1984	410	0.7
2481	Engelberger Aa - Buochs, Flugplatz	1983	443	2.5
2485	Allaine - Boncourt, Frontière	2002	366	0.0
2493	Promenthouse - Gland, Route Suisse	2011	394	0.0
2500	Worble - Ittigen	1988	522	0.0
2604	Biber - Biberbrugg	2002	825	0.0
2608	Sellenbodenbach - Neuenkirch	2003	515	0.0
2612	Riale di Pincascia - Lavertezzo	2004	536	0.0
2613	Rhein - Weil, Palmrainbrücke	1995	244	1.0
2617	Rom - Müstair	2002	1’236	0.0
2623	Rhone - Oberwald	2003	1’368	19.3
2635	Grossbach - Einsiedeln, Gross	2012	942	0.0

TABLE A.1: Metering Stations

Listing of the 60 FOEN stations which provided temperature data as data basis for this thesis. However, the stations 2265, 2433, 2434, 2493, and 2635 were excluded during the analyses as the recorded data does not span over the defined time period starting in 2009.

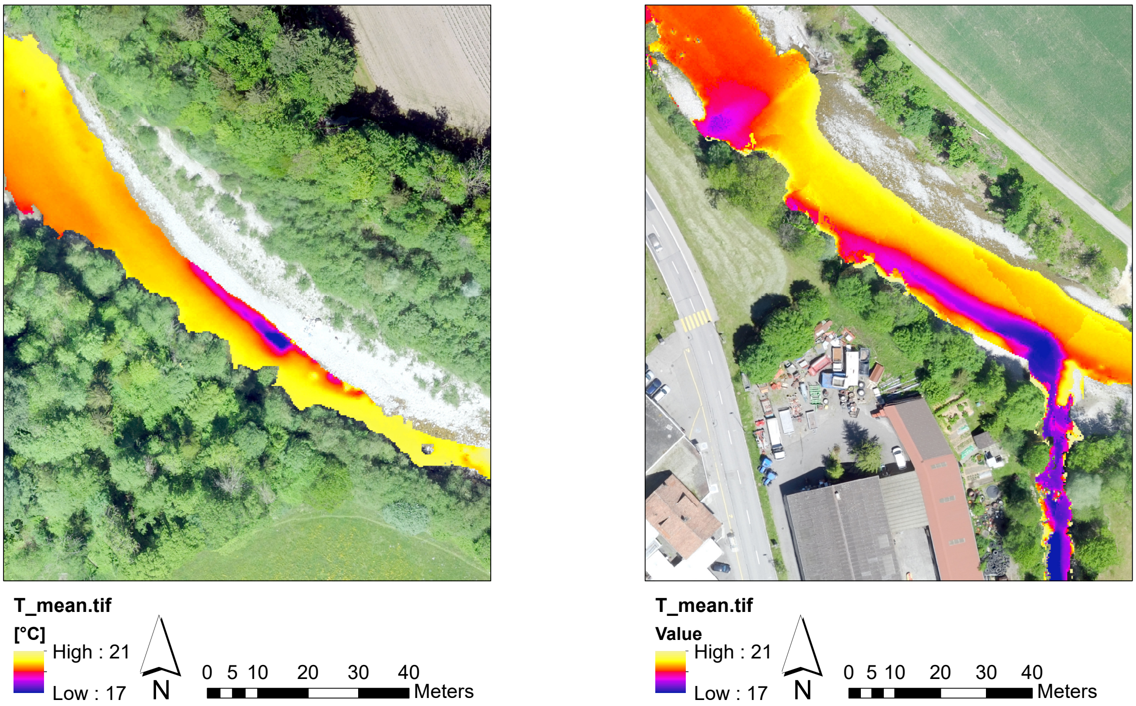


FIGURE A.2: Thermal-Infrared Orthofotomosaic
Thermal image revealing the temperature heterogeneity existing in rather close area of two different water bodies. The images were recorded on March 22, 2020 in Aeschau (left) and Eggiwil (right), Switzerland. Groundwater exfiltration or the inflow of lower tempered water may causes this. (Image by © Ecohydrology Research Group ZHAW)

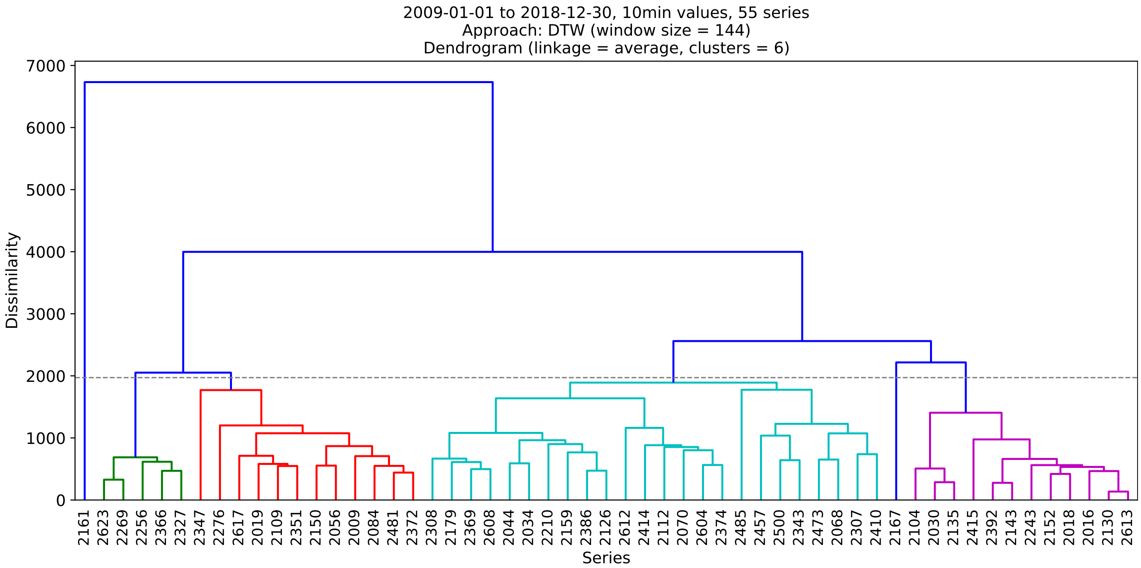


FIGURE A.3: Dendrogram using Average Linkage (6 Clusters)

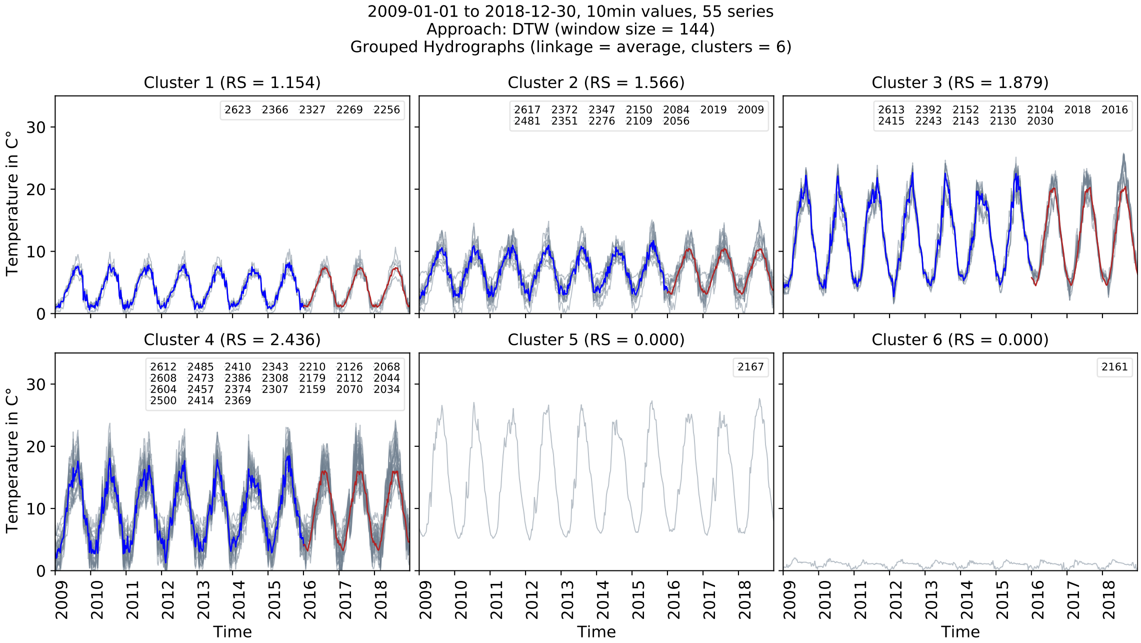


FIGURE A.4: Clustered Hydrographs using Average Linkage (6 Clusters)

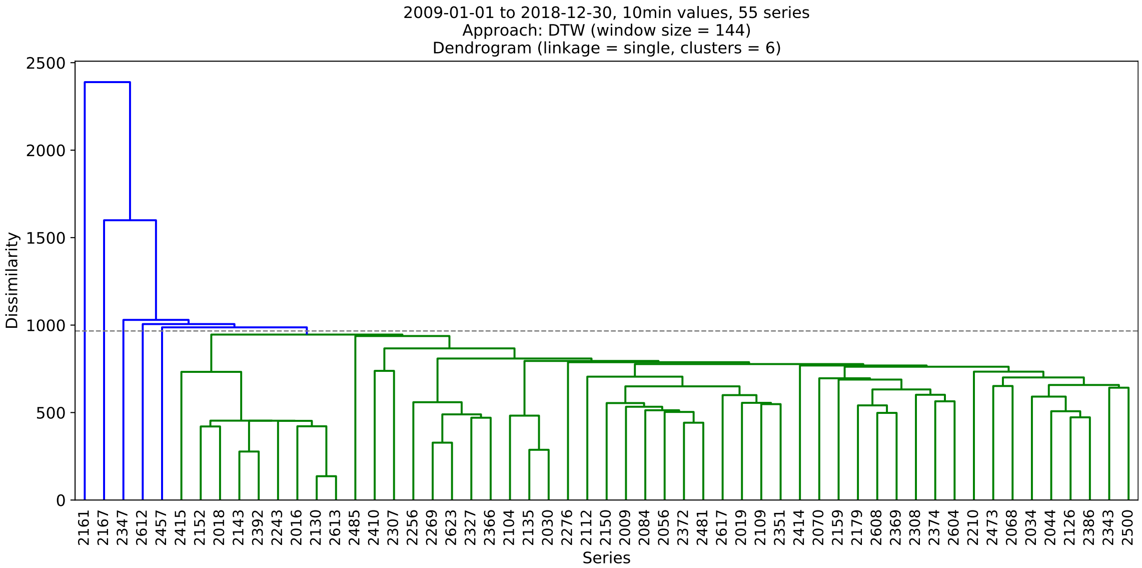


FIGURE A.5: Dendrogram using Single Linkage (6 Clusters)

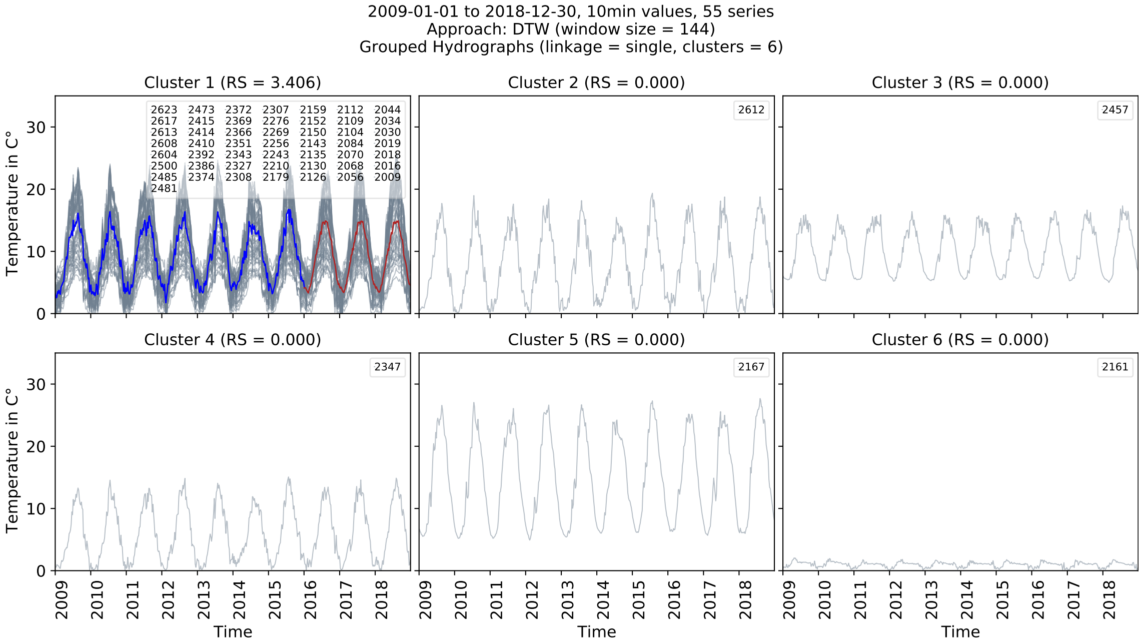


FIGURE A.6: Clustered Hydrographs using Single Linkage (6 Clusters)

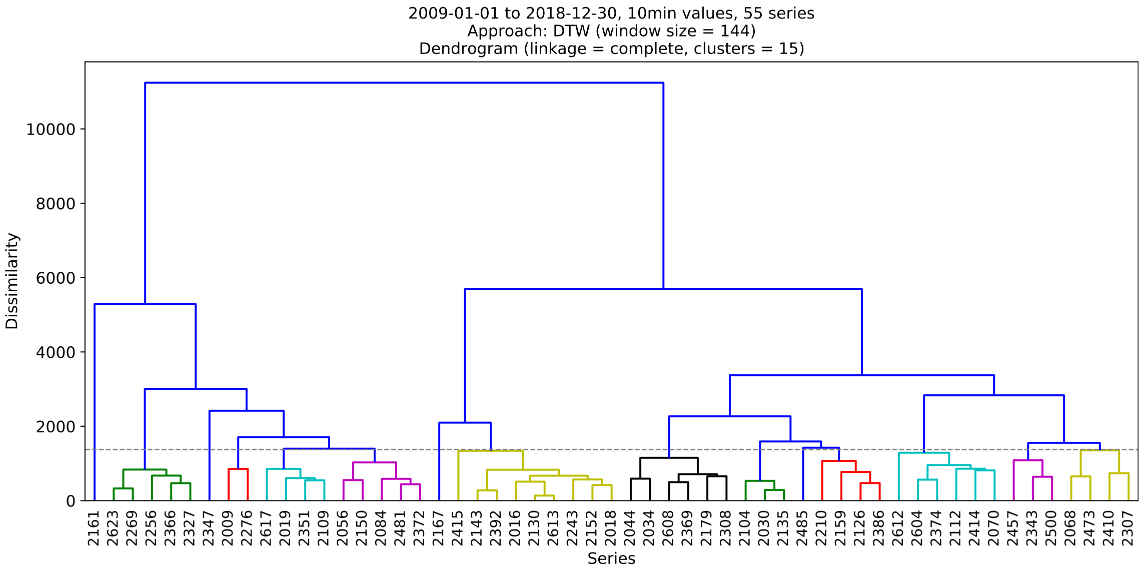


FIGURE A.7: Dendrogram using Complete Linkage (15 Clusters)

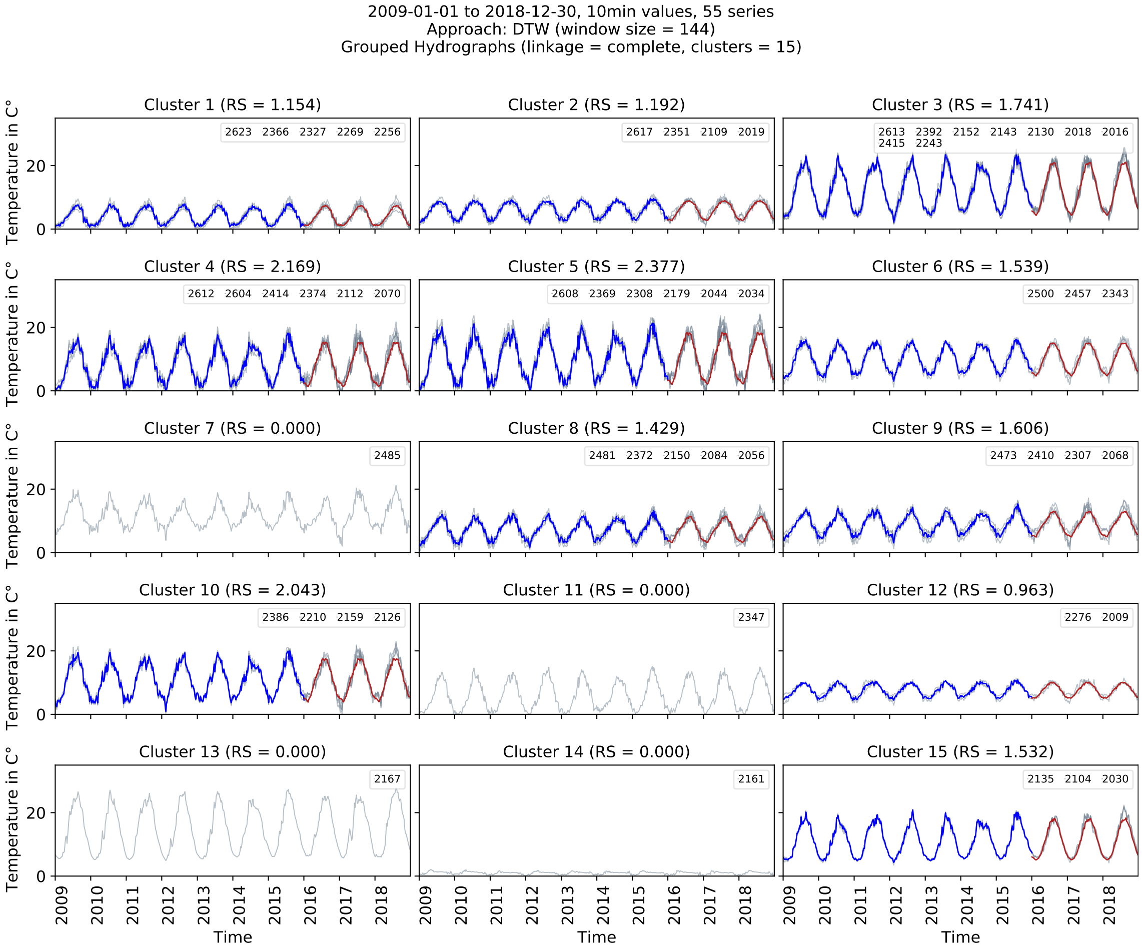


FIGURE A.8: Clustered Hydrographs using Complete Linkage (15 Clusters)

B Cluster Quality Assessment

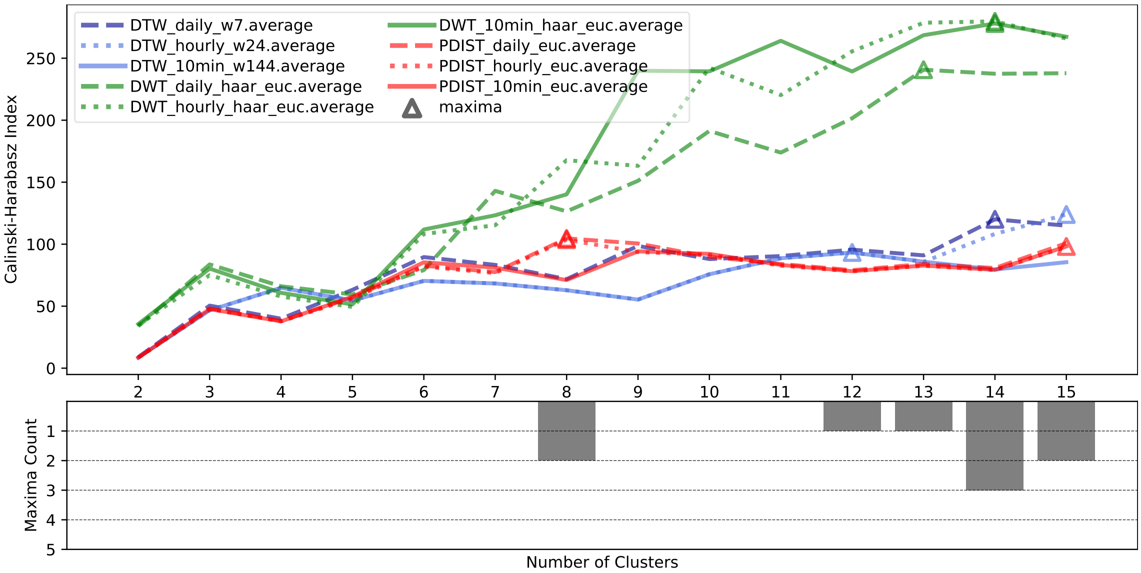


FIGURE B.1: Approach Comparison using Calinski-Harabasz Index

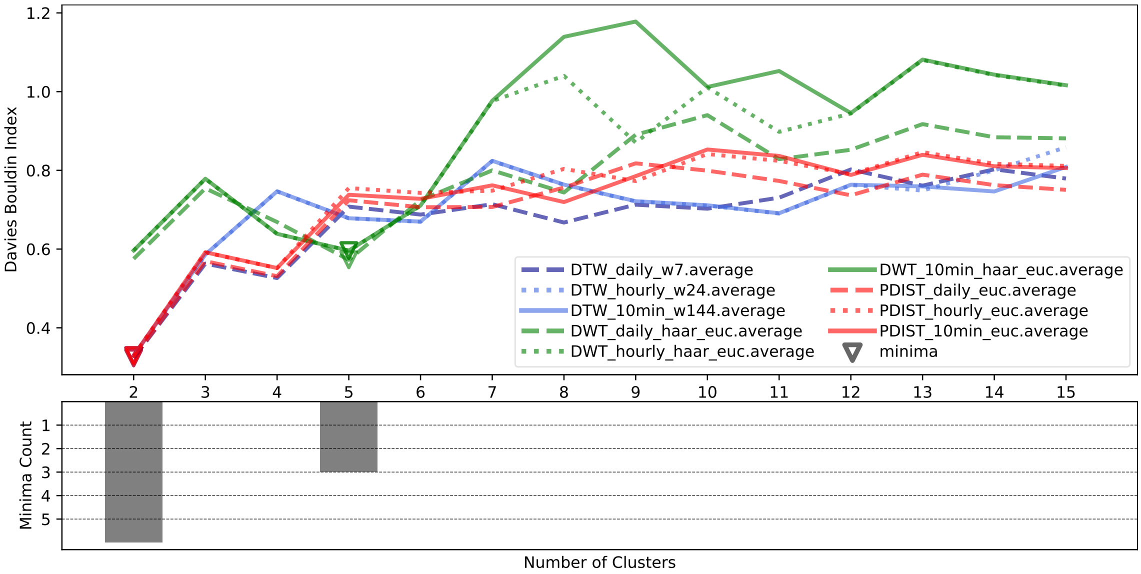


FIGURE B.2: Approach Comparison using Davies-Bouldin Index

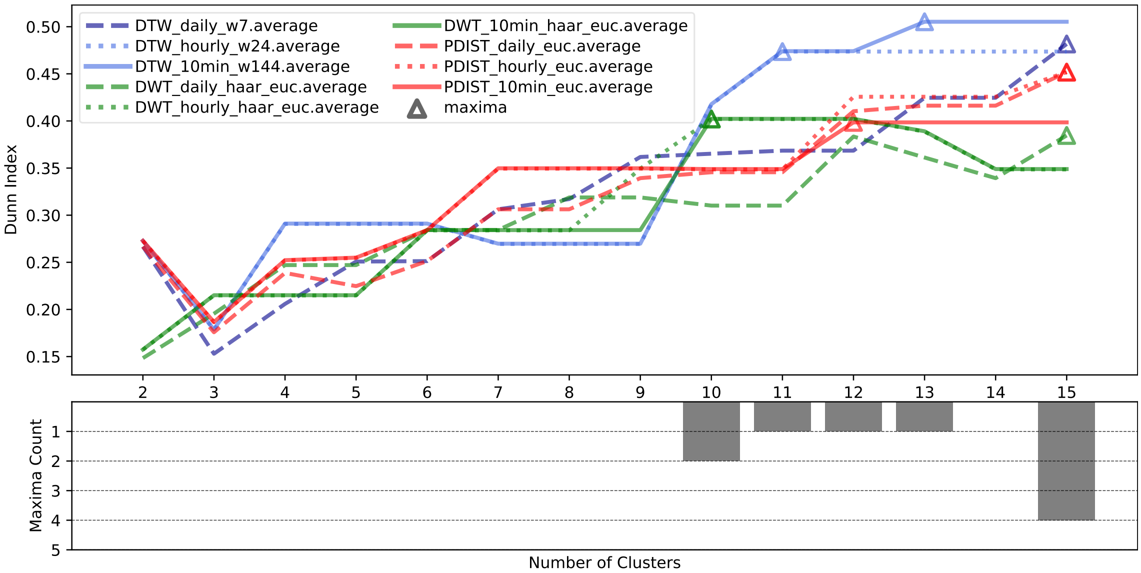


FIGURE B.3: Approach Comparison using Dunn Index

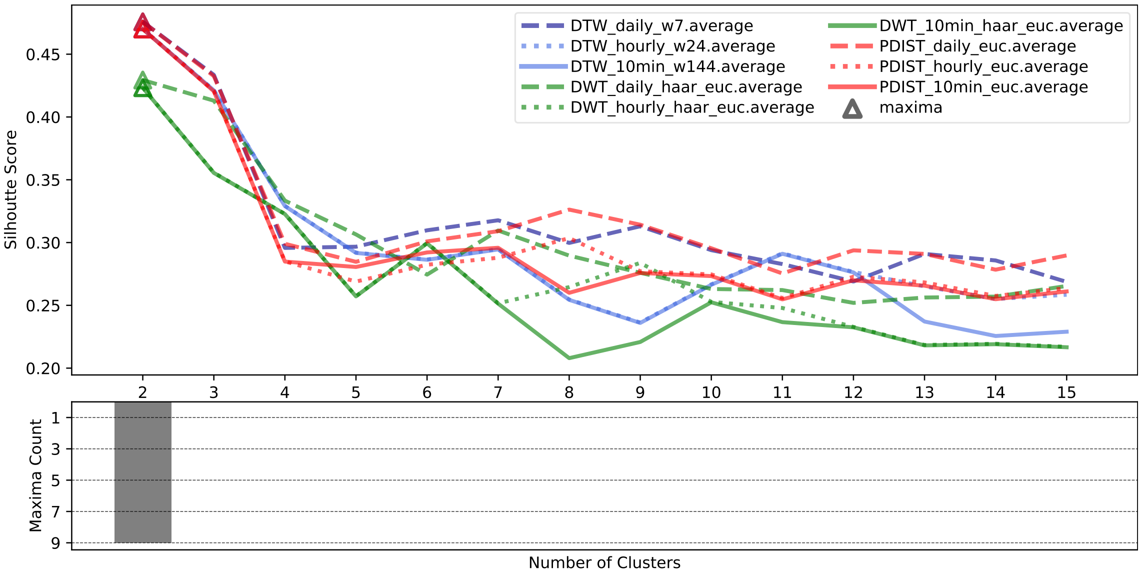


FIGURE B.4: Approach Comparison using Silhoutte Index

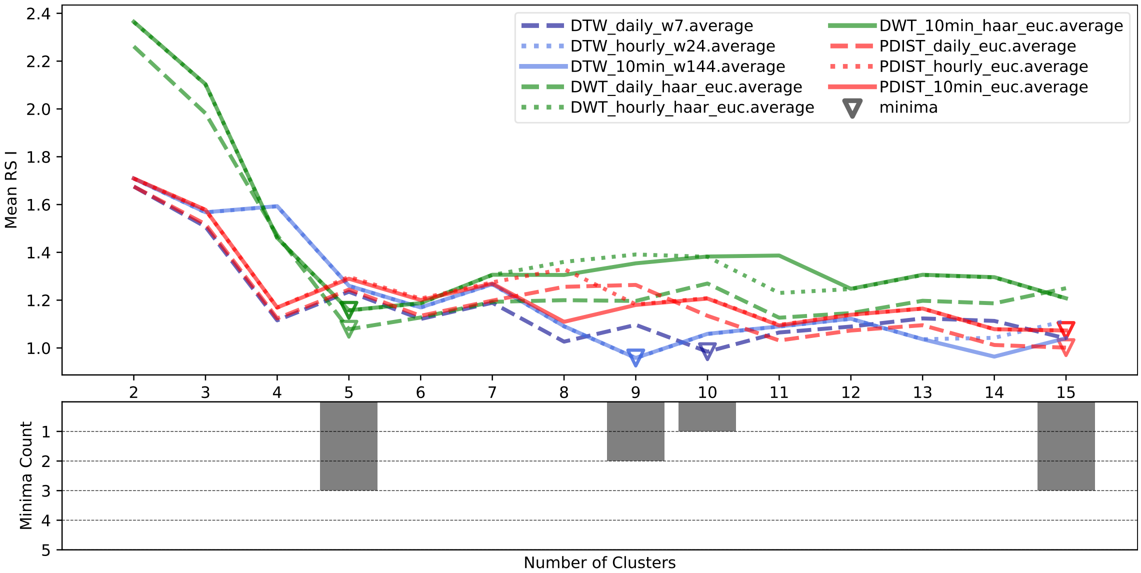


FIGURE B.5: Approach Comparison using Mean RS I

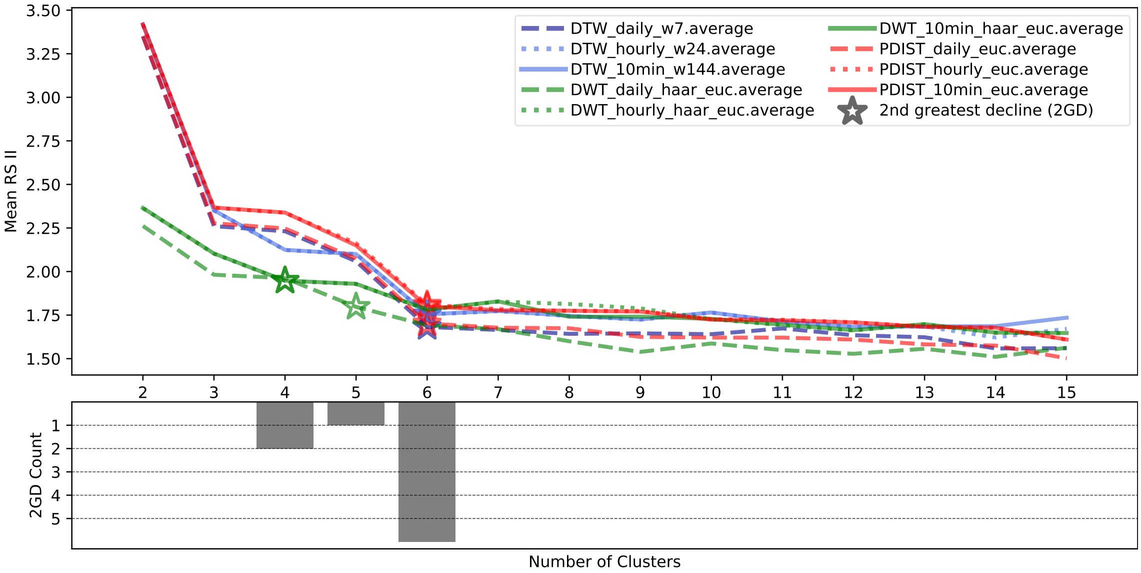


FIGURE B.6: Approach Comparison using Mean RS II

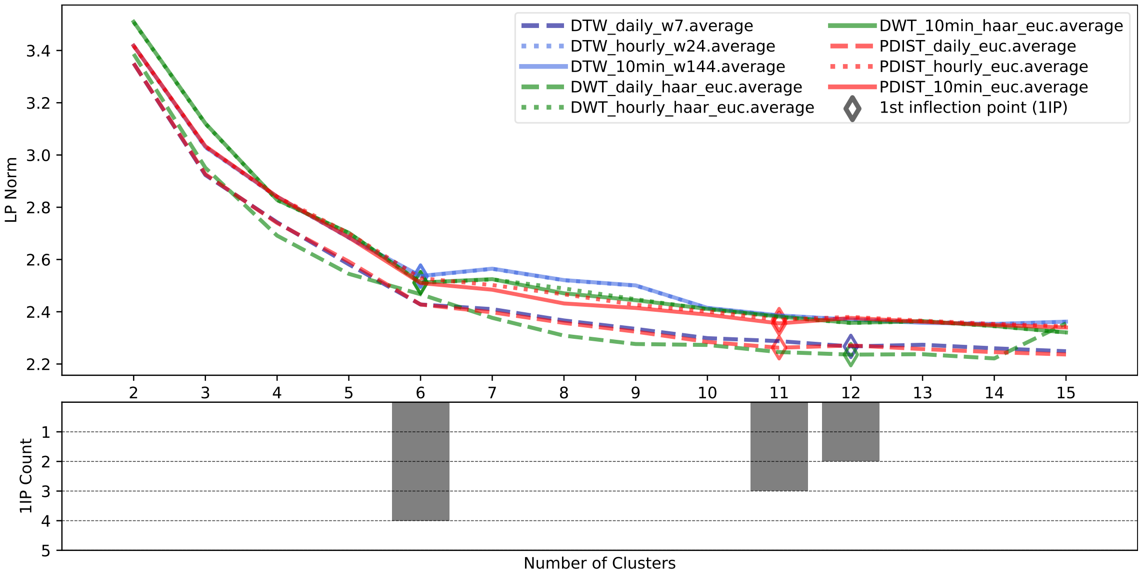


FIGURE B.7: Approach Comparison using L_p Norm

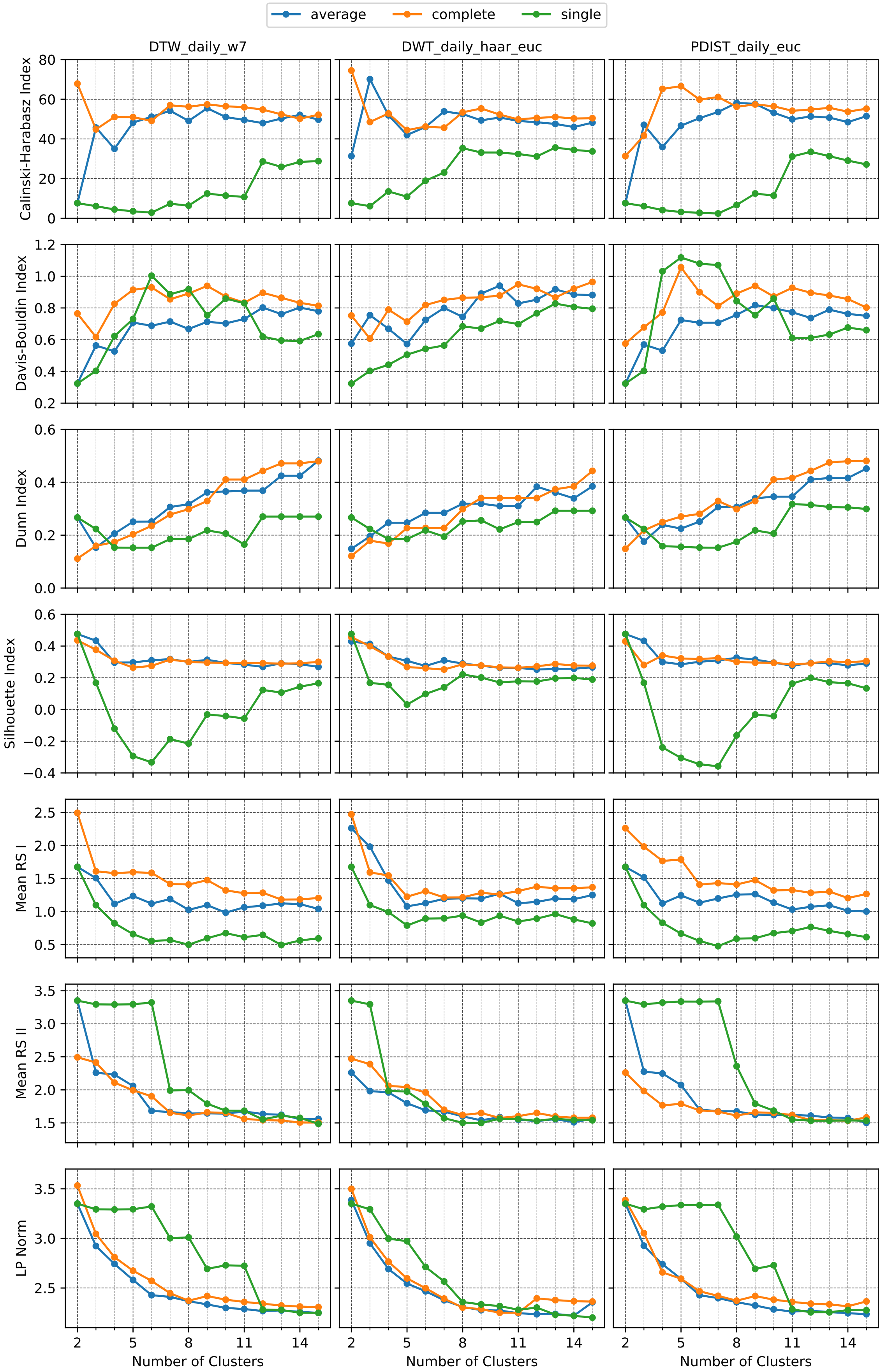


FIGURE B.8: Cluster Validity Indices Comparison (daily values)

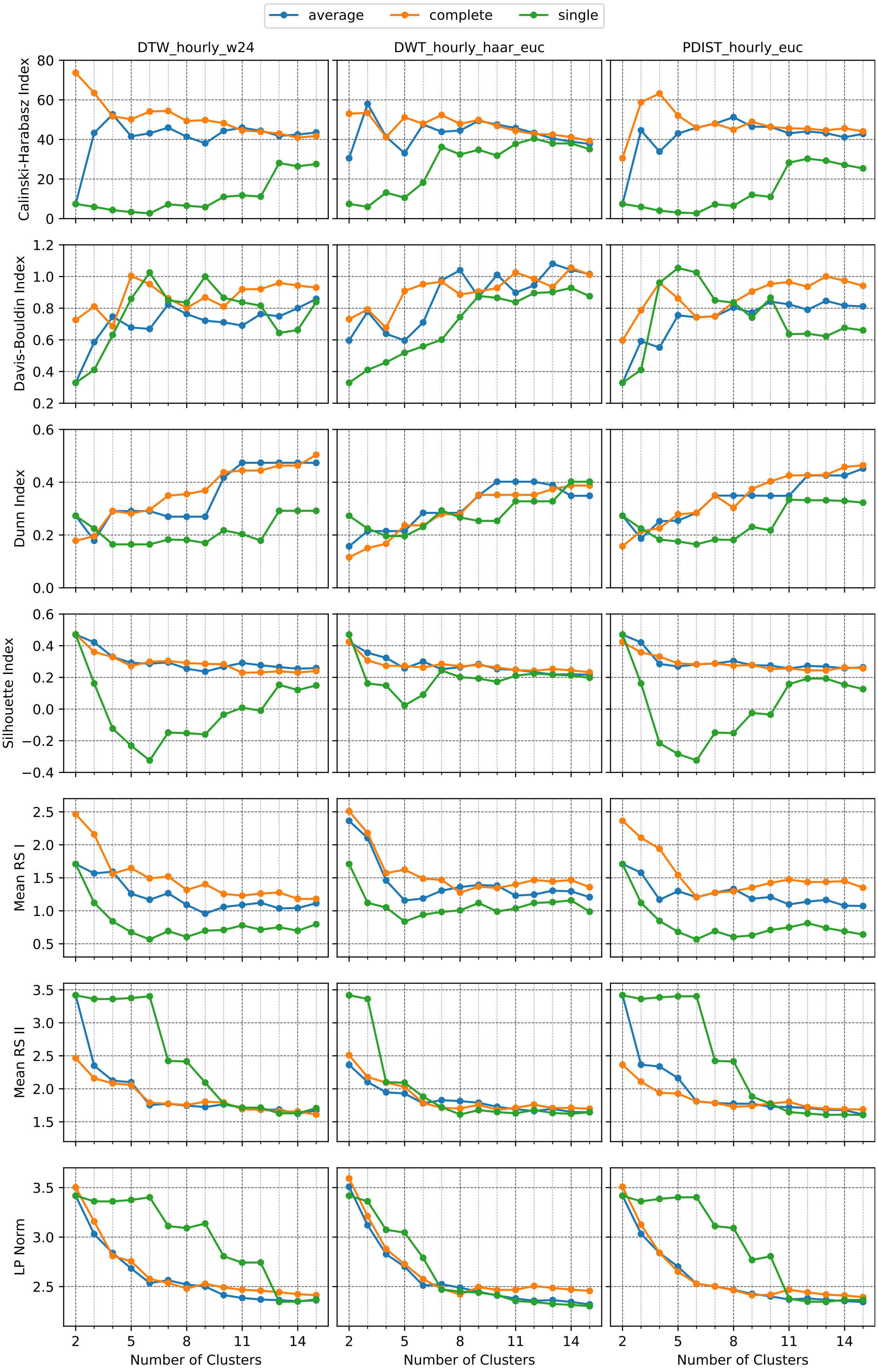


FIGURE B.9: Cluster Validity Indices Comparison (hourly values)

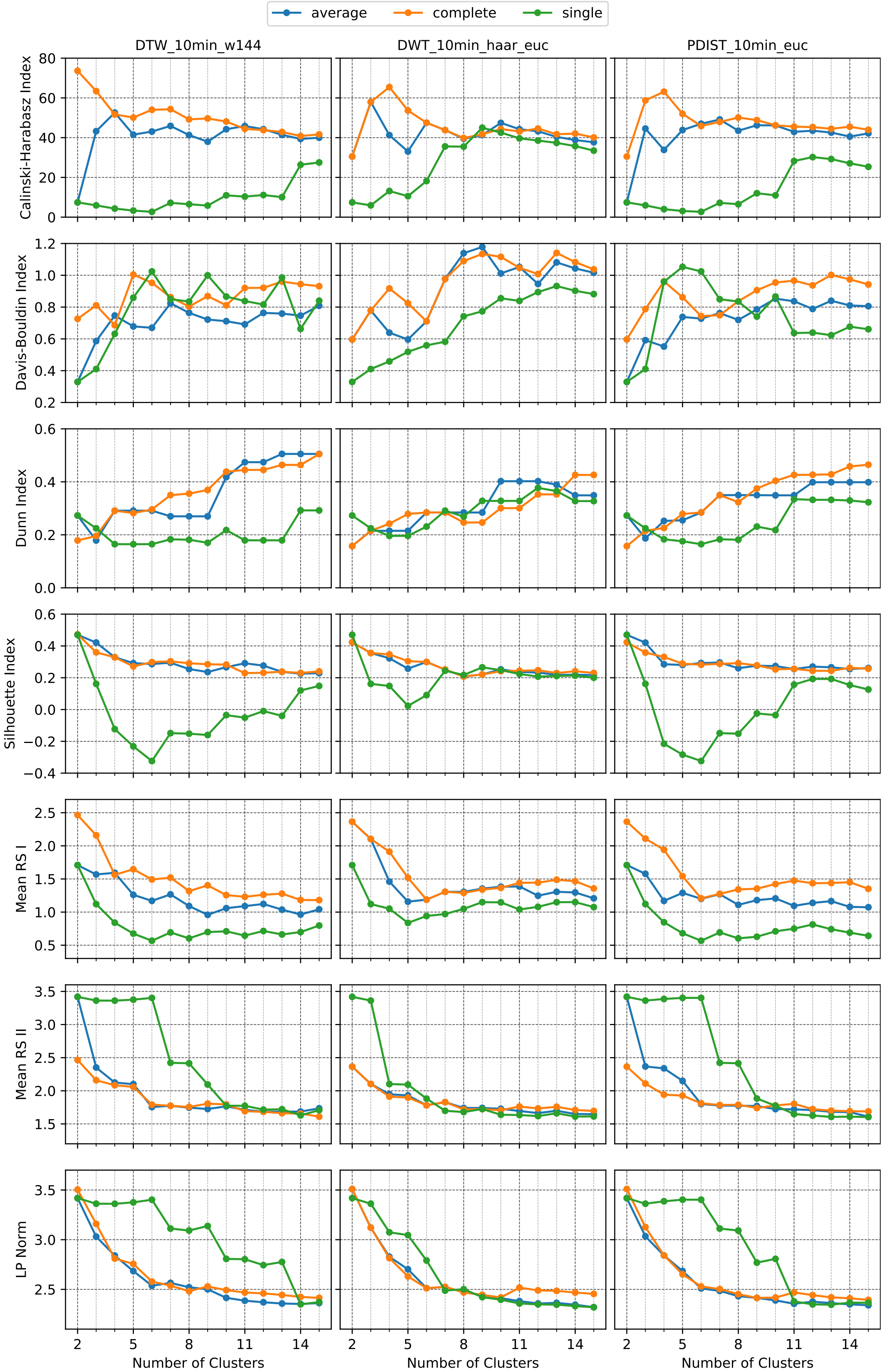


FIGURE B.10: Cluster Validity Indices Comparison (10min values)