Global shoreline forecasting with a clustering approach

May 20, 2021

B.J. Holland

Delft University of Technology

Deltares

TU Delft

Global shoreline forecasting with a clustering approach

May 20, 2021

by

B.J. Holland

Student number: 4382455
MSc student Hydraulic Engineering
Specialization in Coastal Engineering

to obtain the degree of

Master of Science
Civil Engineering

at the Delft University of Technology,
to be defended publicly on the 28 of May, 2021.

Student number:	4382455	
Project duration:	Sep 2020 – May 2021	
Thesis committee:	Prof. dr. ir. S.G.J. (Stefan) Aarninkhof	TU Delft, Civil Engineering
	Dr. ir. A.P. (Arjen) Luijendijk	Deltares & TU Delft, Civil Engineering
	Dr. ir. J.A.A. (José) Antolínez	TU Delft, Civil Engineering
	Dr. ir. J.P. (Juan Pablo) Aguilar-López	TU Delft, Civil Engineering
	Ir. E.C. (Etiënne) Kras	Deltares

This thesis is confidential and cannot be made public until the 1st of January, 2022.

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Cover: Photo by wendel moretti from Pexels

Deltares

TU Delft

Preface

This master thesis is the final result of my work at Deltares. It marks my last project and final step towards my (online) graduation at the Delft University of Technology, completing the master of Civil Engineering. This thesis brought two scientific fields together which both have my attention and interest: data science and hydraulic engineering. Both areas are finding their way through my research, and for me, the final outcome highlights the possible synergy between both.

First, I would like to thank my graduation committee for helping me to deliver this master thesis. Thank you, professor Stefan Aarninkhof for your supervision and input during the meetings, challenging me to bring the research to a higher scientific level. José Antolínez, thank you for your critical thinking along with the data science topic and your enthusiasm for this research field. Thank you, Juan Aguilar-López, for asking the right questions during the meetings and pushing me to think critically about my steps regarding clustering and forecasting. Secondly, I would like to thank my daily supervisors Etiënne Kras and Arjen Luijendijk. Thank you Etiënne, your approachable character made it easy for me to reflect on my work and share my doubts with you. In addition, your eye for detail helped to boost my report significantly. Arjen, thank you for your energetic and positive attitude during these last nine months. Your thoughts, ideas, and comments on my work during the weekly meetings helped to raise this thesis's quality undoubtedly.

Finally, I would like to express my gratitude to my family and friends. My parents, for providing me all the opportunities and the encouragement in everything I do. My two brothers, thank you for supporting me and the fun we can have together. My friends, thank you all for the memorable and pleasant times we had during our time in Delft and Rotterdam. Last but not least, Willemijn, thank you for the joy you bring and for your support while working on this thesis.

I hope this thesis will excite you about the possibilities we have in hydraulic engineering within our growing data-driven society. Also, in other scientific fields, data science is finding its way increasingly. May this thesis inspire you to explore opportunities in your specific scientific area.

B.J. Holland
Rotterdam, May 2021

Summary

Coastal areas around the world have always been densely populated areas. In half of the countries with coastal areas, 80% of the population lives within 100 km of the sea. However, sea-level-rise (SLR) and an increase in single extreme events (e.g., storms, hurricanes, floodings) due to climate change, threaten the coastal areas and their inhabitants. Governmental organizations, coastal managers and various private parties thrive for better insights into long-term shoreline behavior for sustainable decision-making. Currently, these insights are gathered by process-driven models that are often time expensive, require local input and calibration, and are often limited by equations.

Nowadays, studies into shoreline behavior can be accurate but are limited to local cases. In these local cases, only local information is used as an input, and the design of these models concentrates on the local applicability. However, satellite imagery proved to be a promising data source to derive historic shoreline behavior on a global scale. To enable the use of such a large amount of satellite data, image processing techniques are introduced to interpret such large datasets. Furthermore, Machine Learning (ML) algorithms are suggested to create an extra in-depth understanding of these satellite images. Though, these ML algorithms are computationally expensive and the ease of use is sometimes subject to discussion. Nevertheless, due to growing computational power and standardization of ML packages, ML is increasingly used in various fields. The increase in the availability of both satellite data and ML algorithms opens possibilities for shoreline forecasting on a global scale. This research aims to improve forecasting of shorelines by creating a global model in which cross-time series information can be used. To encourage a meaningful exchange of time series information, a clustering forecasting approach is proposed (Bandara et al., 2020). The shoreline data per cluster will be used to train forecast algorithms, leading to insights into the performance of algorithms for specific shoreline behavior.

This research builds upon the yearly Satellite Derived Shoreline (SDS) data behind the Shoreline Monitor (Luijendijk et al., 2018). The SDS dataset captures shoreline evolution and behavior on a global scale by quantifying it in time series. The SDS dataset consists of transects every 500 m along the global coast and contains 33 years (1984-2016) of shoreline evolution. The sandy transects in the Shoreline Monitor are the basis of this study, since these are relevant for long-term forecasting and have the highest accuracy of all transects in the dataset.
First, in order to cluster transects, the features for clustering were defined. Here, clustering features were divided into time series features and hydraulic and geomorphic features. Whether transect clustering could improve shoreline forecasting, based on these time series and hydraulic and geomorphic features, was explored as part of this thesis. It became apparent that grouping transects based on time series features resulted in meaningful insights into transects' behavior and the forecasting accuracy per cluster. Besides time series features, geomorphic and hydraulic features for clustering purposes were examined. However, the correlation between geomorphic and hydraulic features versus transect behavior was insufficient. So, for this thesis, it was decided to focus on time series features for further clustering usage.

Second, after exploring features for transect clustering, the final clustering input was defined. Eventually, a shape-based transect representation was chosen as input for the clustering algorithm. Nearly 350,000 globally distributed sandy transects (representing about 175,000 km of global shorelines) were assigned to nine different main clusters, with a semi-unsupervised clustering approach. These nine clusters captured global sandy shoreline behavior ranging from extreme erosion to extreme accretion and proved to be a practical tool to quantify the (distribution of) shoreline behavior on different spatial scales. For example, on a regional scale, the distribution of transects based on the clusters in the Baltic Sea corresponded with recently published research. Moreover, it became apparent that more than 30% of the sandy shorelines are in moderate and stable regimes.
Subsequently, a second supervised step was developed to create sub-clusters to gain more insights into the trends of the nine clusters. With this supervised sub-clustering step, three sub-clusters were generated for each of the five largest clusters, allowing for the distinction between accelerating and decelerating behavior in the last decade of time series per cluster. In order to show the potential of these sub-cluster refinements, several case studies on different spatial scales were examined. For example, on a global scale, the reversing trend in the Caspian Sea was confirmed by the sub-cluster approach due to the variety in sea level in the Caspian Sea. Next to that, the accelerating erosion along the coast in central Chile, due to SLR and increase in storm frequency, detected by C. Martínez et al. (2018), was also confirmed in the sub-clusters. The case studies in this thesis provided new insights for the nine main clusters and corresponding sub-clusters. Primarily, additional insights into the state of shoreline evolution were found. Secondly, previous research about shoreline behavior was coupled with (sub-)cluster information, strengthening these previous findings. Finally, new and reversing trends in shoreline behavior were detected.
Hereafter, the nine main clusters were separately used as input for four different forecasting algorithms. This resulted in predictions per cluster with a more reliable and more accurate seven-year forecast for >95% of the transects, sometimes improving accuracy up to factor 15, compared to recently published work. In addition, it became evident that different clusters performed better with varying forecasting algorithms, highlighting the relevance of this specific clustering approach. Furthermore, a multi-method approach was created to determine the most reliable forecast, with a future horizon of half the original time series' length, on a local scale by combining overall accuracy and information gained during clustering. Finally, a global method was generated for a high-resolution quantitative and qualitative vulnerability assessment of predictions on local coasts. In this vulnerability assessment, specific insights into the potential impact on the local coastal zone were discovered.

All of the described above contributes to a better understanding of global sandy shoreline behavior. Time series clustering enhances the forecasting of transects based on overall accuracy and local reliability. The multi-method approach for selecting the appropriate prediction on a local level, could be a practical tool for governments to apply. Besides, the vulnerability assessment explores the possible application of the predictions and underlines the need for global databases regarding the coastal zone.

However, further research should consider incorporating hydraulic and geomorphic features to strengthen the clustering of transects. Furthermore, it is expected that a higher resolution of publicly available data and SDS data will further improve global shoreline forecasting in the future. Hence, shoreline forecasting based on historical data should be considered as a valuable tool for sustainable decision-making regarding coastal zones.

Contents

Acronyms

ACF	Auto Correlation Function.
AE	Absolute Error.
AI	Artificial Intelligence.
ANN	Artificial Neural Network.
ARIMA	Autoregressive integrated moving average.
DBA	DTW Barycenter Averaging.
DTW	Dynamic Time Warping.
DWT	Discrete Wavelet Transformation.
ES	Exponential Smoothing.
GEE	Google Earth Engine.
GRIP	Global Road Inventory Project.
HWL	High Water Level.
LSTM	Long short-term memory.
MAE	Mean Absolute Error.
MAPE	Mean Absolute Percentage Error.
MASE	Mean Absolute Scaled Error.
ML	Machine Learning.
MLP	Multilayer Perceptron.
MSE	Mean Squared Error.
MWH	Mean Water Height.
NAN	Not A Number.
ND	Normalized Difference.
NDWI	Normalized Difference Water Index.
NN	Neural Network.
nRMSE	Normalized Root Mean Squared Error.
OLS	Ordinary Least Squares.
OSM	OpenStreetMap.
RMSE	Root Mean Squared Error.
RNN	Recurrent Neural Network.
SDS	Satellite Derived Shoreline.
SE	Squared Error.
SLR	sea-level-rise.
sMAPE	symmetric Mean Absolute Percentage Error.
TOA	Top of Atmosphere.

Chapter 1

Introduction

Shorelines are changing due to the constant exposure to the sea and environmental influences (see figure 1.1). Both natural and human factors affect long-term shoreline dynamics. These dynamics occur with a wide variability on temporal and spatial scales (Stive et al., 2002). Natural and human factors highly influence shorelines from an ecological and socio-economic perspective. In addition, nowadays sea-level-rise (SLR), as a result of climate change, accelerates the evolution of shorelines (Nicholls & Cazenave, 2010). This accelerated shoreline evolution increases the demand for long-term shoreline predictions.

Figure 1.1: Queenslands Gold Coast heavy erosive behavior threatening hinterland. Source: Bruce Miller/CSIRO

1.1 Natural and human pressure in coastal zones

Shorelines have always been heavily populated due to their diverse ecosystems and economic advantages, 50% of all coastal countries have 80 to 100% of its total population living within 100 km of their coastline, and 21 of the 33 world's megacities are found near the coast (M. L. Martínez et al., 2007). Despite the advantages and possibilities of living close to shorelines, it also has various disadvantages. Shorelines can be highly variable locally and some may experience extreme erosion that threatens the hinterland, like in South Vietnam, where erosion rates of -30 m/year occur. Next to local variability, coastal behavior may differ widely on a global scale. This difference is explained by the variety in tidal influences, wind conditions, and other natural processes. Moreover, the variety in daily forcing, the temporal and spatial shoreline response to these forces highly depends on the available and supplied sediment budget and the sediment composition itself. Besides the daily forcing, where seasonality due to the influence of macroclimate indices like El Niño and La Niña (Ranasinghe et al., 2004) may cause longer-term variations, (single-event) natural hazards also have a major impact on coastal zones. Cyclones, storms, floods and hurricanes accelerate shoreline behavior. For instance, the dune erosion rate for a (Dutch) design storm is 200/300 m^3/m (Bosboom & Stive, 2015) (see figure 1.2 for example of extreme storm erosion), where under normal conditions the water level hardly reaches the dunes.

Figure 1.2: Extreme dune erosion after storm in Egmond (NL) 1989. Source: City Archive Alkmaar

Considering the SLR (Nicholls & Cazenave, 2010) with the growing population in coastal areas, the number of people exposed to these threats increases (Neumann et al., 2015). Society is trying to cope with these threats by intervening in the coastal system. Since the coastal zone experiences a complicated interaction between water and land, human interventions can disturb natural processes by modifying the geological environment (Huang & Jin, 2018). The aforementioned growing population, combined with the increased occurrence of natural hazards (human interventions sometimes accelerate that), makes governments and other parties seek in-depth shoreline insights. They need insights into long-term behavior and evolution in shorelines to create improved strategies to mitigate these future shoreline hazards (Long & Plant, 2012).

1.2 Shoreline monitoring and forecasting

The current practice of monitoring shorelines has clear limitations, as field measurement campaigns are expensive as well as time-consuming (Davidson et al., 2010). New state-of-the-art methods could counter these limitations by using the widely available earth satellite data that has grown in the new data-driven society. By interpreting this satellite data, it can be used to understand historical shoreline behavior and possibly even be used to predict future changes. In this research, the Satellite Derived Shoreline (SDS) data in the Shoreline Monitor will be used to generate forecasts with state-of-the-art algorithms. The SDS data consists of yearly data points from 1984 till 2016. The SDS data was introduced by Luijendijk et al. (2018) and later confirmed (Hagenaars et al., 2018). The Shoreline Monitor is a large database with historical shoreline behavior and is openly available. Current deterministic and probabilistic shoreline forecasting is done using process-driven models, which are computationally expensive and only consider a particular set of variables and formulas that describe the (governing) physical processes. Data-driven models, nowadays enriched with Machine Learning (ML) techniques, have proved that these limitations can be overcome by describing statistical relationships instead of physical processes: they can maximize probability, emulate model components, smooth continuous nonlinear regression, and can include uncertainty (Goldstein et al., 2019). In this research, the possibilities to construct and improve shoreline forecasting using statistical relationships will be examined. By improving shoreline forecasts, coastal managers can rely on and use these improved forecasts for sustainable future shoreline management.

1.3 Research objectives and goal

Present forecasting of shoreline behavior is focused on local models. Locally means that the model is limited to forecasting shoreline behavior on a local scale. The (trained) model is designed and validated for purely local cases and not applicable to other cases. Next to that, the most frequently used forecasting models are process-driven and statistical forecast methods. Both process-driven and statistical methods are methods that do not actively include information on other locations worldwide. As this thesis will focus on sandy shorelines and since 31% of the ice-free shorelines are sandy shorelines (Luijendijk et al., 2018), there is a wide availability of shorelines data. ML forecasting algorithms could benefit from this large set of time series data, indicating a shift from a local to a global model. This study investigates possibilities to improve forecasting of shoreline positions by utilizing information from other coastal areas. A global model for shoreline forecasting will be created. A global model will enhance the exchange of time series information worldwide, learning from best practices.
The main goal of this research is to improve shoreline forecasting by creating clusters of similar time series. This clustering should result in insightful groups of time series. These clusters could provide a better understanding of the variety of shoreline behavior on different spatial scales. This provides the opportunity to conduct studies between similar coastal areas worldwide. The definition of *similar* in this context is two-folded. First, similar time series, are time series that show similar (statistical) features. Second, time series that are exposed to similar hydraulic conditions or share the same geomorphic features. Both, *time series based* as well as *hydraulic/geomorphic conditions based* similarities can be used for clustering. Next, the clusters are individually trained by different ML algorithms that enhance time series information exchange between similar transects. Each cluster will eventually be connected to the most suitable forecasting method. In short, this research will provide an in-depth understanding of global shoreline behavior. The added value of constructing a global model that is enriched with time series of shoreline dynamics will be further investigated in this thesis.

1.4 Research questions

In this section, the main research question is presented as well as the sub-questions used to help answering the main question. The main research question is:

How can shoreline forecasting be improved by making a global model using a clustering approach and public data?

The associated sub-questions are:

1. **To what extent can transect clustering improve shoreline forecasting?**
   In this first sub-question will be studied how the performance of current algorithms can be improved by simple time series features grouping (e.g., season, trend). It is expected that this will improve the forecast's accuracy and so the clustering approach can be continued. Furthermore, this sub-question answers which other time series features could be useful for better clustering. Besides time series clustering, coastal characteristics (e.g., hydraulic conditions and geomorphic conditions) will be considered as well. This sub-question elaborates on recent research that did not use a clustering approach, and hence one has built a single global model (Calkoen et al., 2021).

2. **What methods to use for clustered forecasting and how does this improve shoreline forecasting?**
   In this sub-question, several methods will be evaluated. First, an analysis is made on which clustering technique to use, considering the information gathered by answering the questions above. Next, the forecast model per cluster is examined, starting with models stated by Calkoen et al. (2021). After training and running those algorithms, the results will be combined and evaluated.

3. **How can open-source information be used to make a more reliable local forecast?**
   In this final sub-question, the predicted future shoreline positions are reviewed and the application is explored. An analysis is made to what extent the predicted transects are applicable, considering coastal characteristics and other physical boundaries.

For the objectives and research questions, the following hypothesis stands at the basis: *By clustering transects, a better global and local understanding of transect behavior is created. The resulting clusters as input for forecasting algorithms, combined with the increased understanding of the transect's behavior, will lead to more reliable predictions on the local scale.*

1.5 Report outline

This report consists of six main Chapters. In Chapter 1, the objective and problem statements are stated. In Chapter 2, the background information of the existing expertise of the relevant scientific fields is reviewed. In Chapter 3, the methodology and results per sub-question are defined. The results are presented after each step because outcomes of different steps influence subsequent steps and sub-questions. In Chapter 4, a discussion on the results and formed methodology is conducted. The assumptions made are reviewed, also the decisions and (future) applicability of the research are discussed. In Chapter 5, the results of Chapter 3, together with the discussion in Chapter 4, are combined to answer the main research question. Finally, in Chapter 6, recommendations are drawn to improve future research and possible directions for future research are explored.

Chapter 2

Background information

This chapter provides an overview of current research related to the research problem and objectives stated in Chapter 1. First, subsection 2.1 explains the assumptions and outcomes of the Shoreline Monitor. Next, Subsection 2.2 describes the different cluster and classifying techniques and approaches for time series clustering. Afterward, Subsection 2.3 defines the different forecasting models and methods for time series forecasting. Finally, the last Subsection 2.4 introduces the database for sub-question 3.

2.1 Shoreline Monitor

Despite the acknowledged importance of the coastal zones and shorelines, there is no extensive database that considers shoreline changes. To monitor shoreline changes, insight into the historical behavior is needed. These historical insights will help in the long-term estimation and management of these regions. Also, interactions between observed behavior can help in the understanding of interactions of human and natural phenomena. As mentioned before, human interventions can highly influence shoreline behavior. In addition, different natural phenomena such as storms, etc., also influence the behavior of shorelines. Current insights are focused on local shoreline behavior, providing specific insight into the local changes. However, the rise of satellite data opened ways for a more global examination. In order to interpret this satellite data and make it applicable for shoreline behavior examination, the Shoreline Monitor was introduced based on a transect system and Google Earth Engine (GEE) platform (Luijendijk et al., 2018). The technique used in the Shoreline Monitor is the Satellite Derived Shoreline (SDS). The Shoreline Monitor presents historical shorelines' behavior on a global scale every 500 m over a time of 33 years, from 1984 until 2016. These systems will be explained in the following sections.

2.1.1 Google Earth Engine (GEE)

The GEE contains over a petabyte of satellite data and geospatial datasets on a planetary scale. It enables rapid prototyping and visualization of the data in an online environment (Gorelick et al., 2017). In these datasets, one can find observations from a large variety of satellite imaging systems (Gorelick et al., 2017). Scientists, developers and researchers can analyze datasets to detect trends or quantify changes on the earth's surface. This analysing can be done in an open- and private environment.

2.1.2 500 m transect system

The transect system is created by creating transects every 500 m orthogonally to the global shoreline extracted from the Google Earth Engine (GEE) from 2016. These transects are 1 km in length, 500 m

land inwards, and 500 m seawards. 2.2 million transects were defined to identify the total of ice-free shorelines. These transects are used for calculating the length of the ice-free shorelines on earth. Summing the straight intercepts between the transects results in a total length of 1.1 million km of ice-free shorelines. These 2.2 million transects were split up by their position on the world to speed up computation time. Due to this splitting, the transects were labeled with IDs. A short description of this procedure is described in Appendix A.1. Finally, the entire set of 2.2 million transects is filtered with latitude filters (60° and −50° resulting in 1.8 million ice-free transects.

2.1.3 Sandy shoreline detection

For the detection of sandy shorelines, the world has been divided into boxes of 20 km x 20 km, and only the boxes which intersect with the 2016 shoreline from OSM are used. This results in 24,000 boxes. The detection of the sandy shorelines is schematically presented in figure 2.1.

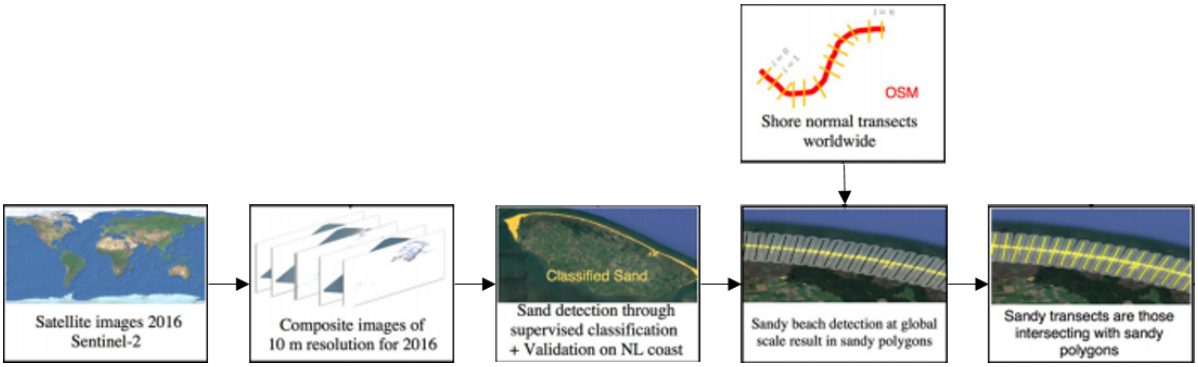

Figure 2.1: Sandy shoreline detection steps. Adjusted from (Luijendijk et al., 2018).

The detection of the shorelines is done by using a pixel-based supervised classification. The classification is to a global Top of Atmosphere (TOA) reflectance percentile composite image of 2016, using all Sentinel-2 images available. The supervised classifier is trained by a Classification And Regression Tree algorithm (CART) (Luijendijk et al., 2018). The algorithm was trained and tested on sandy beach locations on the Dutch island Texel resulting in a 97% accuracy. This trained supervised classification method is now used for all boxes created in the previous step. The result is a series of sandy polygons worldwide, including quartz- and carbonate sands and gravel. These series are further analyzed eventually, transects that intersect with a sandy polygon were classified as 'sand' and other transects as 'non-sand'. Transects where no classification was possible due to the absence of a cloud-free Sentinel-2 image, were classified as an undetermined sediment composition. The results of the sandy shoreline detection are shown in figure 2.2.

2.1.4 Shoreline detection

A coastal characteristic that is detectable for monitoring is the shoreline position (Hagenaars et al., 2018). Following literature, there are two types of shoreline position features: shoreline that is visibly discernible with coastal imagery like High Water Level (HWL) and the intersection of a tidal datum with the coastal profile like Mean Water Height (MWH) (Boak & Turner, 2005). However, recently a third type of shoreline feature detection was introduced, based on image processing techniques to extract shoreline position indicators from coastal images (Hagenaars et al., 2017). This shift towards data-driven shoreline detection resulted in the development of SDS algorithm (Hagenaars et al., 2017). Hagenaars et al. (2018) found out by using a Normalized Difference Water Index (NDWI) that the spatial accuracy of the SDS data was within a subpixel (less than 10-30 m). Even cloud and wave-free images resulted in an average accuracy of 1 m. Therefore, SDS proved to be suitable

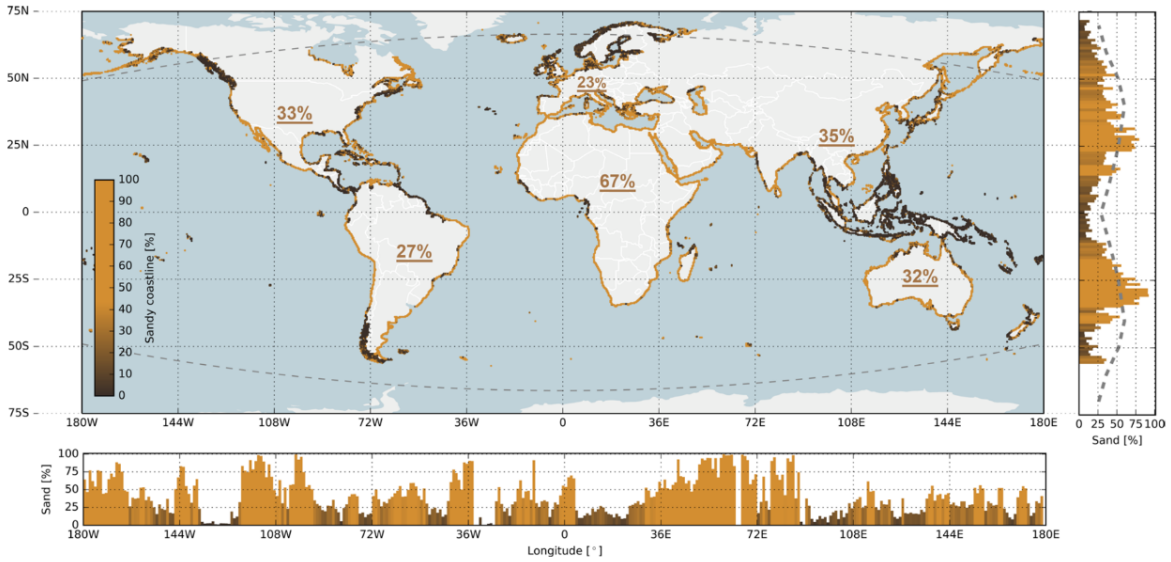

Figure 2.2: Global distribution of sandy beaches worldwide (Luijendijk et al., 2018).

for studying shoreline behavior larger than subpixel size ($+/-0.5m$) for the period 1984-2016. The process towards extracting an SDS is shown in figure 2.3.

Recently Luijendijk et al. (2018) proposed a method to derive annual global scale shoreline changes. The approach used in this research is shown in figure 2.3. This approach follows similar steps as the approach in figure 2.1. However, instead of only using the Sentinel-2 imagery in the sandy shoreline detection, this novel approach uses all the NASA satellite data from 1984-2016. Next to that, yearly composite images of a resolution of 30 m are created. These composite images are created using an averaging period of 192 days, given the 16-day revisiting time of the satellites. Due to the averaging, the water level in the average images corresponds to the yearly mean-sea-level, given cloud-free images as input for the averaging (Luijendijk et al., 2018). By averaging, all sub-yearly variability is filtered out, and long-term variability and trends are still available in the data, which is highly preferable for using the data for long-term forecasting. To finally obtain global annual shoreline position, the SDS detection is applied to the yearly composite images. This is done by combining the last step in figure 2.1 with the last step in figure 2.3. Computing the intersections with the global shore-normal transects system together with the annual shoreline results in transects every 500 with 33 data points (1984-2016) (Luijendijk et al., 2018). In figure 2.4 different transects are plotted for the 33 years. The plot shows the different behavior of shoreline behavior that is captured in the dataset.

On the global SDS transects, a linear shoreline trend is obtained by linear regression (Hagenaars et al., 2017). All SDS positions larger than three times the standard deviation of a transect are considered outliers and are not used. Next to linear regression, transects with a trend larger than $100m/year$ are neglected. Also, transects with less than 5 of 33 data points and with less than seven years of coverage are not used in this study.

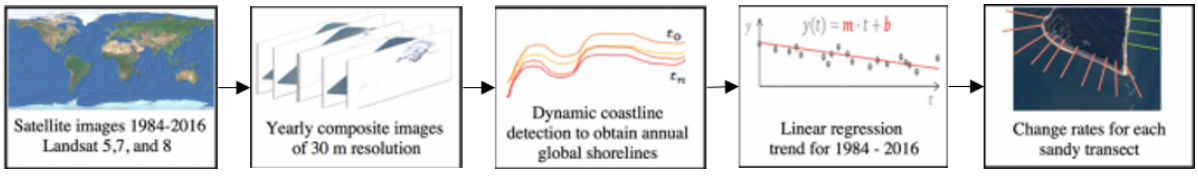

Figure 2.3: Dynamic shoreline detection steps. Adjusted from (Luijendijk et al., 2018).

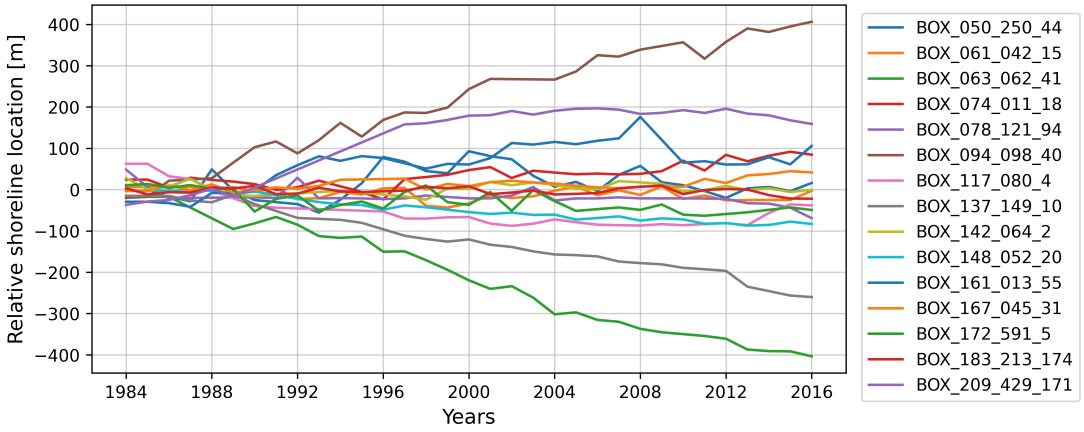

Figure 2.4: Set of 15 different transects, showing all different historical behavior. This highlights the diversity of shoreline behavior in general and captured in the dataset behind the Shoreline Monitor.

2.2 Classifying and clustering

The goal of this research is to create meaningful clusters of time series to improve forecasting. In order to create meaningful clusters, the following is essential:

- In each cluster, the similarity between data points should be maximized.
- Maximum dissimilarity should be created between clusters.

There are multiple ways to achieve these meaningful groups. In current research, there is one major distinction for making these groups:

- Clustering (unsupervised learning)
- Classifying (supervised learning)

Clustering is a technique to find interesting patterns and groups in features from the input data. Primarily multi-dimensional data benefits from clustering models since it is often too challenging to discover relations between variables with multiple dimensions. Clustering aims to create clusters where objects are similar (based on defined features) to each other. There are many different clustering techniques, of which each technique has specific benefits. Classifying has the same purpose as clustering, finding patterns in the data. However, the data is already labeled with the class they belong to, making it a supervised learning technique (Pedregosa et al., 2011). As the shoreline data in this research is not labeled with groups, meaning that they are not grouped yet, it is an unsupervised grouping problem. This thesis, therefore, focuses on the clustering of transects.

2.2.1 Time series clustering

Current research focuses on three types of time series clustering: whole time series clustering, subsequence clustering, and time point clustering (Zolhavarieh et al., 2014). Whole-time series clustering is focused on clustering individual time series based on their similarity. Subsequence clustering is clustering with a set of subsequences extracted from a single time series. Timepoint clustering focuses on the clustering of time points based on their temporal proximity and similarity. Both subsequence and time point clustering are performed on individual time series. Therefore, whole-time series clustering is necessary for these because this thesis focuses on multiple time series. Whole-time series clustering consists of four steps as depicted in figure 2.5: Time series representation, similarity or distance measures, clustering prototypes, and clustering technique (Zolhavarieh et al., 2014).

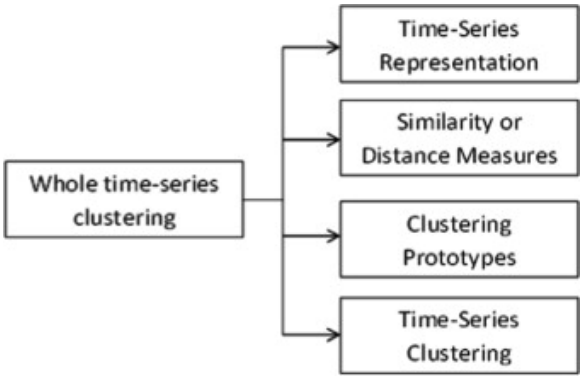

Figure 2.5: Whole time series clustering component overview (Aghabozorgi et al., 2015).

Time series representation

The first step of whole time series clustering is the time series representation. The step aims to optimally represent the time series and reduce its dimensions for optimal and faster clustering. So time series representation is two-folded, dimension reduction and optimal representation for the clustering. Two different representation techniques for time series clustering are often used according to Aghabozorgi et al. (2015): features-based and shape-based. With feature-based representation, one extracts relevant features (e.g., mean, variance, linearity) of the time series with length n and creates a vector of k features, where $k < n$. This will reduce the dimensions of the time series and help in representing the time series. Feature extraction is the process of creating a new dataset with features from the initial time series data. This new dataset covers the main properties of a time series dataset and represents it in a lower-dimensional space that stimulates and improves future modeling. Next to that, the reduction of dimensions can provide faster computation time. Whether this is necessary depends on the final use of the model.

Another representation approach is shape-based, here are the shapes of time series matched as well as possible. The shapes used for measuring similarity can be the raw-time series but can also be modified representations. A popular shape-based time series representation is Discrete Wavelet Transformation (DWT). DWT is a numerical analysis tool that reduces the dimensions and filters out different frequency bands of time series. It decomposes the time series into basis functions, which are called wavelets. A wavelet is a rapid wave-like oscillation with a finite duration. Wavelets are defined by two parameters: the scaling function (ϕ) and wavelet function (ψ). The scaling function characterizes the scale of the wavelet, and the wavelet function the basic shape of the wavelet. In figure 2.6 different wavelets are depicted. A DWT is gathered by passing a time series through a high-pass and low-pass filter, the high-pass filter gathering high frequent behavior and the low-pass low frequent behavior. The high- and low-pass filter are formed by changing the scaling factor ϕ for a chosen wavelet, the smaller the scale factor the higher the frequency will be of the wavelet. The high- and low-pass filter results in approximation and detailed coefficients per level of decomposition. Per level of decomposition, the approximation coefficients are down-sampled to half of the original time series length by a half-band filter see figure 2.7 for a schematic overview of a third level time series decomposition. Per decomposition, the approximation coefficients capture the first half of the previous frequency band and the detailed coefficients the upper half. For example, if a raw time series consists of signals in a frequency range of 0-1000, the first level approximation coefficients contain a signal with a frequency between 0-500 and the detailed coefficient between 500-1000. The approximation coefficients are then further used in a second-level decomposition. The basic idea behind DWT is to compute how much of a wavelet is in a time series or signal for a chosen scale and location. The main advantage of DWT is that it captures not only frequency infor-

mation but also information about location in time. Another popular technique, Discrete Fourier Transformation, only captures frequency information. What level of decomposition, detailed or approximation coefficient or what wavelet to use are depending on the data and aim of the transformation. The level of decomposition depends on the trade-off between dimension reduction and the practical application of the transformation.

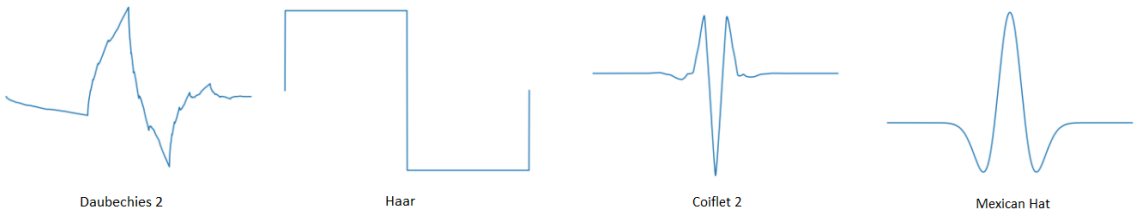

Figure 2.6: Daubechies 2, Haar, Coiflet 2 and Mexican Wavelet functions.

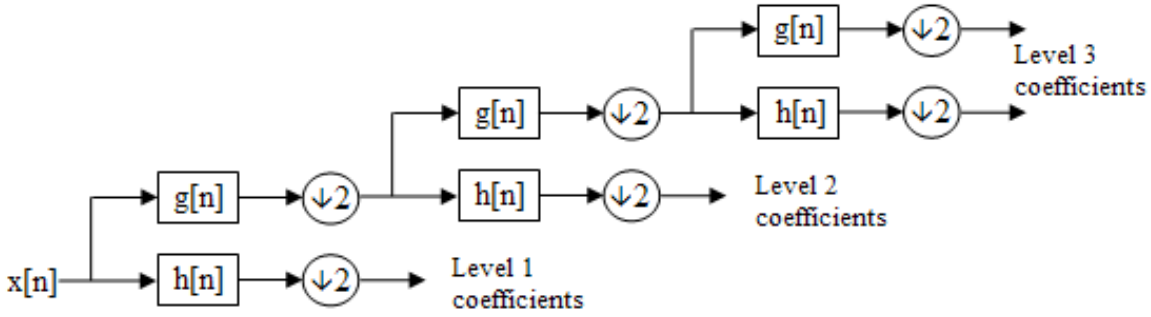

Figure 2.7: Time series are decomposed into detail coefficients (from the high-pass filter) and approximation coefficients (from the low-pass filter). The decomposition could than be repeated and the approximation coefficients are decomposed with high and low pass filters and down-sampled again.

Similarity or distance measures

This subsection will review the distance measurement approaches for clustering time series. Clustering depends on distance measurements to a great extent, it expresses the similarity and dissimilarity between time series features. The distance measurement method highly depends on the representation of the time series. The most frequently used distance measures for time series clustering are the Euclidean and Dynamic Time Warping (DTW) distance measures, see figure 2.8. The Euclidean distance measure is simple to use and has computational advantages in comparison with DTW. However, DTW can account for lagged behavior between time series and can compute the distance between different time series lengths.

- Dynamic Time Warping (DTW) is an algorithm for measuring the distance between time series, which may vary in length and speed. The idea is based on one-to-many and many-to-one so that the total distance between two time series can be minimized.
- Euclidean distance measure is the 'normal' distance that one would measure with a ruler. It is based on the Pythagoras rule and measures the distance between every point a time step t_n of two time series.

Choosing the right distance measure is highly important and there are many applicable in the the time series clustering domain (Aghabozorgi et al., 2015). It depends on the data and the final use of the clusters.

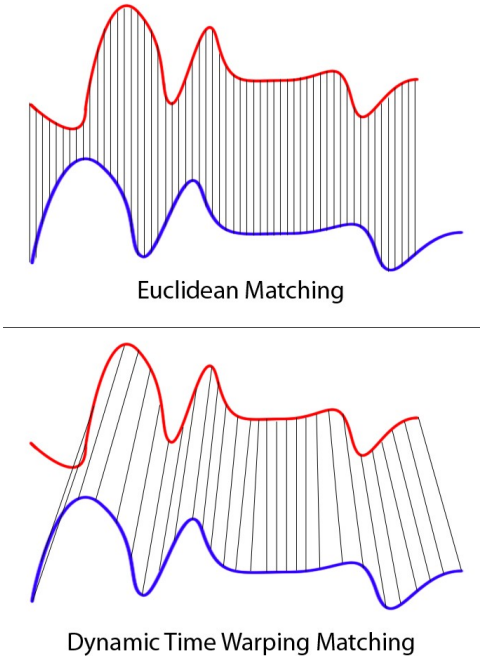

Figure 2.8: Euclidean vs. Dynamic Time Warping. Source: TowardsDataScience.com

Clustering prototypes

The third step in whole time series clustering is defining the cluster prototype. This is an essential part of the clustering of time series. Especially in the (most common) partitioning clusterings methods like K-means, K-medoids, or Hierarchical clustering, the quality of clusters is highly dependent on the definition and quality of the prototypes. For each cluster, the clustering algorithm aimed to reduce the distance between the features and the prototype within a cluster. For defining the prototypes, there a generally two methods (Aghabozorgi et al., 2015), illustrated in figure 2.9:

- Medoid cluster as a prototype is an approach where the center of a cluster is defined as a feature(set) that minimizes the squared distance to other objects in the same cluster.
- Mean cluster as the prototype is a simple averaging technique where the cluster's mean is used as the prototype. This prototype does not need to be an actual feature(set), unlike the medoid prototype. A disadvantage of the mean prototype approach that it needs non-elastic distance measures (e.g., Euclidean). When the time series features are different in shape or using DTW as distance measurement, the actual mean of the cluster is unable to capture.

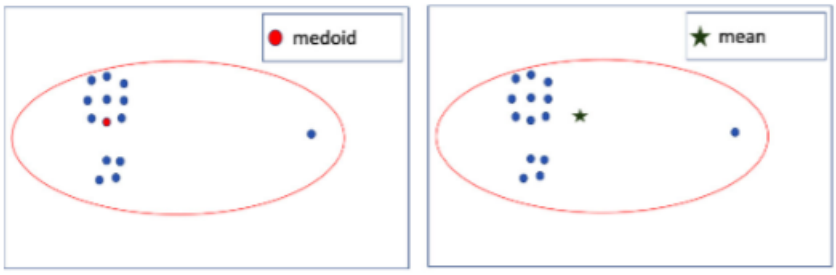

Figure 2.9: Medoid and mean prototypes for clustering purposes. Source: (Javed et al., 2020).

Time series clustering algorithms

In previous sections, the first three steps in time series clustering are discussed. Different representations of time series are possible, one can use raw time series or use a representative set of features. Similarity and dissimilarity can be measured with several methods and the cluster prototype can be defined in two ways. The last step in figure 2.5 is the clustering algorithm. In the current research, the following two clustering algorithm types for time series are mainly applied:

- **Partitioning algorithms**
  The main aim of partitioning algorithms is to minimize the total distance (mostly euclidean) between all objects with the cluster prototype. Defining the prototype is an important aspect of the algorithm as the quality depends upon the cluster prototype. The most common portioning algorithms are, Kmeans and Kmedoids. A major drawback of this algorithm is that it is required to assign the total number of clusters as input.

- **Hierarchical clustering**
  Hierarchical clustering is a clustering technique that makes a hierarchy of clusters using divisive or agglomerative algorithms, as shown in figure 2.10. Agglomerative algorithms consider each item as a cluster and then merge the clusters gradually (bottom-up). Opposite, divisive begins with all items and then splits the cluster to reach the clusters with one item (top-down). There are multiple options for measuring the distance between pairs of clusters for hierarchical clustering. A popular distance measure for agglomerative clustering is Ward's linkage, it minimizes the variance of items in the merged cluster (Javed et al., 2020). A limitation of hierarchical clustering is that it cannot adjust clusters after splitting or merging. The strong point is that it does not need the number of a cluster as an input parameter.

A partitioning clustering algorithm designed for time series clustering is TimeSeriesKmeans (Tavenard et al., 2020). It is a time series clustering algorithm following the common partitioning algorithm Kmeans. As an advantage, it can use (raw or modified) time series as input and it can use both DTW and Euclidean distance measures for clustering. Agglomerative clustering algorithm (Pedregosa et al., 2011) can handle (raw) time series in the form of a distance matrix. These distance matrices should be pre-computed with another package. Dtaidistance (Meert et al., 2020) is a python package that allows to fastly compute distance matrices with the DTW distance measures. These matrices could be used as input for agglomerative clustering algorithms.

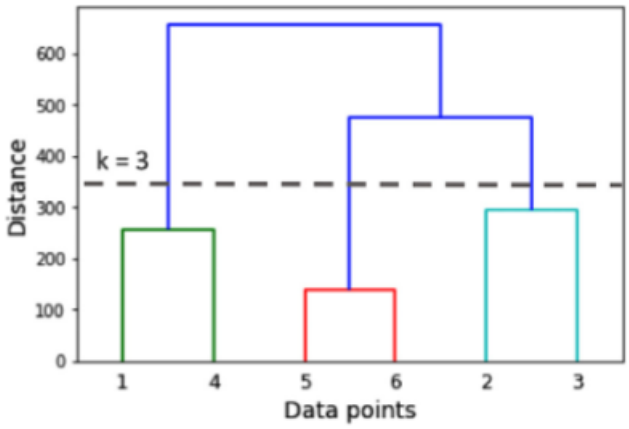

Figure 2.10: Agglomerative clustering. Source: (Javed et al., 2020).

2.3 Forecasting

The forecasting of time series has been an exciting topic for different applications. Over time, enormous progress has been made in many fields (De Gooijer & Hyndman, 2006). Numerous methods and algorithms are known, each good for specific time series and the desired output (Benidis et al., 2020). Forecasting could be traditionally divided into two forecasting methods; statistically- and ML forecasting. Both have a different fundamental approach but share the same aim and forecasting structure. This research will focus on the ML forecasting algorithms, as the aim is to create a global model where cross-time series information is used. In this section, the forecasting basics are explained together with ML forecasting principles.

2.3.1 Forecasting basics

For forecasting, time series are normally split in a training and test set (Hyndman & Athanasopoulos, 2018). The split is depicted in figure 2.11, in which the first part of the time series data is used as a training window and the last part as the testing window. Forecasting algorithms aim to use the training window as input and based on that input to forecast the data points in the test window.
The most commonly used size of the train- and test window is the 80-20 ratio. 80% is used as input to train the model and the other 20% to test the predicted values, this is presented in the graph in figure 2.11 (Hyndman & Athanasopoulos, 2018).

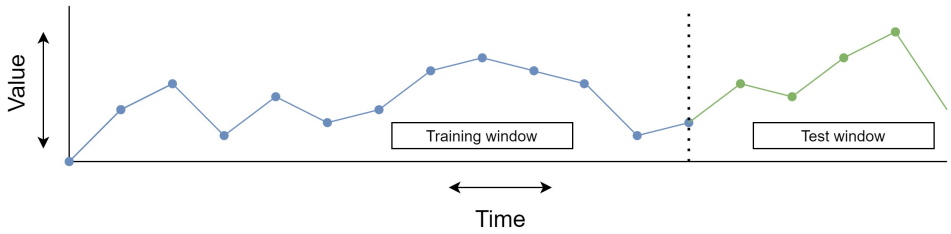

Figure 2.11: Train window and test window for forecasting. The training window is used to predict the data points in the test window. The test window is used as feedback for the algorithm during training and used for evaluating the final algorithm's performance.

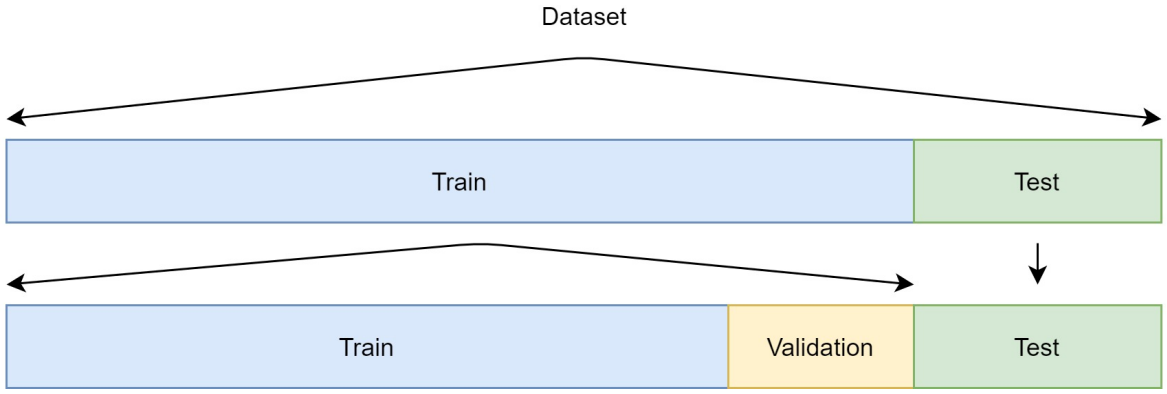

Figure 2.12: Dataset split in Machine Learning (ML). The train set is used for training the algorithm and the validation to tune hyper parameters during training. The test set is used for evaluating the performance of the trained algorithm.

The time series split in 2.11 is on a single time series level. If a dataset contains more than one time series, the dataset should be split as well. The data split is depicted in figure 2.12. ML algorithms sometimes even add a validation set. The training set is a dataset with examples from the original dataset to train the algorithm for the given time series. The test set is used to evaluate the accuracy of

the trained algorithm. The test data is independent of the training data, it is not used in determining the forecasts and should therefore provide a good indication of how well the model will forecast on new data. This makes the forecasting problem a supervised problem, where the training data trains the model to forecast the test data as accurately as possible. The validation set is sometimes generated to tune the hyperparameters of the model. It is split from the original training set and used to test the model and the trained data before the model is finally evaluated with the test data. The validation set is also used to prevent overfitting of the model on the data. Overfitted models do not generalize well from training data to test data and will underperform. This can be overcome by using a validation set while training the model. For all train, validation and test set, the train and test window split in figure 2.11 applies to the time series. However, the test window is used differently for every set. If the time series is in the training or validation set, the data points in the test window are used for training the algorithm. If the time series is in the test set, the test window is purely used to measure the performance of a trained forecasting model. The time series in the test set finally assess the performance of the algorithm. The predicted values for the testing window are compared with the observed values, where the mismatch is expressed in accuracy measures. An example of a measure is stated in equation 2.1, the Mean Absolute Error (MAE). The MAE is often used to compare different methods for the same set of data and is a scale-dependent error measurement.

$$\mathrm{MAE}_i = \frac{1}{N} \sum_{t=1}^{N} \left| z_{i,t} - \hat{z}_{i,t} \right| \tag{2.1}$$

The train, validation and test set split together with the training and test window in figures 2.11 and 2.12 is the standard approach for forecasting. The time series in a dataset are divided into a train, validation and test set. In addition to this approach, some forecasting algorithms use other ways for training the model and evaluating its outcomes. The two graphs in figure 2.13 show another approach for training the algorithm. The train and test split are on an individual time series level. In the train & validation set, the train and test window are generated following the rolling window principle. This can lead to better forecasts, however computing time may increase, and underlying structures in time series may be missed. The last red data points in the time series are not used during training, these data points will be unseen for the model and only used for evaluating the model's performance. With this approach, the algorithm's performance is not dependent on the dataset split, as all time series in the dataset will both be used for training and evaluating the forecast algorithm

2.3.2 Machine Learning

In this section, forecasting with ML methods is discussed. ML is considered as a subset of Artificial Intelligence (AI), which is used to let computers recognize patterns and connections between data and act on these interpretations. Many different fields in science are exploring the applications of AI in their fields. An often used Artificial Neural Network (ANN) in coastal research is a Multilayer Perceptron (MLP). A standard MLP consists of one input layer, one or more hidden layers, and one output layer (see figure 2.14).

Each layer has one or more nodes/neurons that are connected with other nodes in other layers. The amount of nodes in the hidden layers depends on experience, empirical formulas and systematic analysis of the data and goals (Goldstein et al., 2019). The input is inserted in the network through the input layer, where normally, every node input layer represents an input variable. The output layer size depends on the amount of (dependent) variables to predict. A basic ANN is characterized by n input nodes (as input the independent variables), m hidden nodes, and k output node (amount of dependent variables). Nodes between layers are connected and transfer information to a node in the next layer with the following expression:

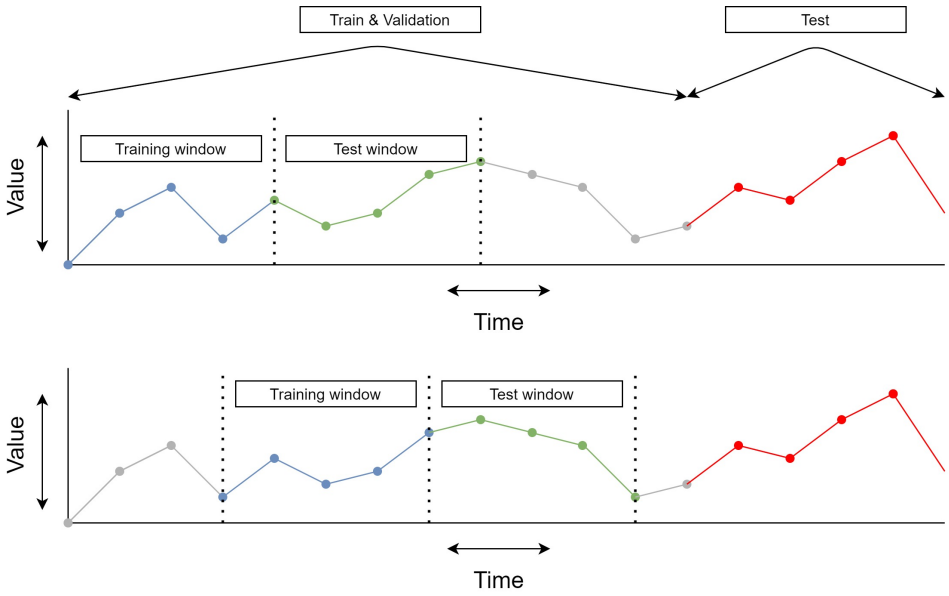

Figure 2.13: Train and test split on individual time series level. The data points in the train & validation set are used for training the model and the red data points in the test set are used for final evaluation.

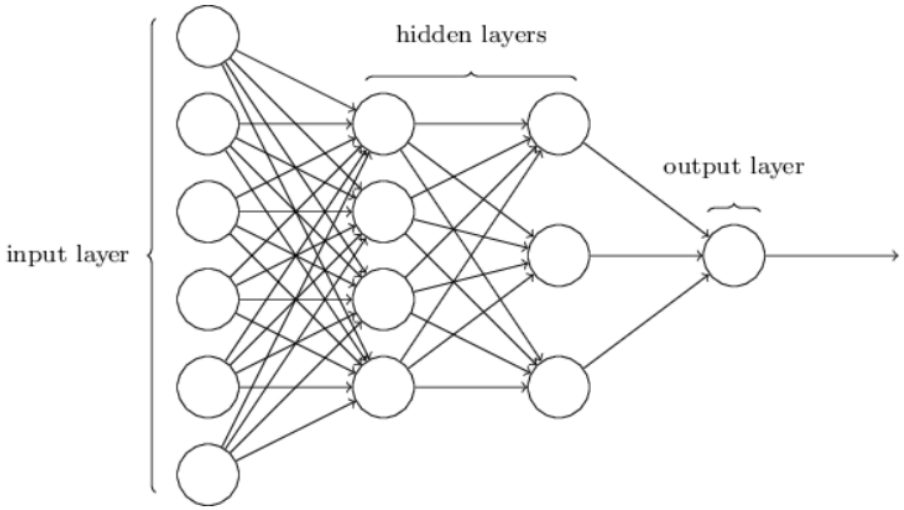

Figure 2.14: Schematically four layer neural network with one input layer, two hidden layers and one output layer (Michael A. Nielsen, 2015).

$$h_j = f \cdot \left(b_j + \sum_{i=1}^{n} w_i x_i \right) \tag{2.2}$$

Where b_j is the bias of the jth hidden neuron, x_i and w_i are the ith input and weight to the neuron, n is the number of inputs of the incoming layer, f is the transform function and h_j is the response of the jth neuron. When information is passed through the network, it is modified at every node by a transfer function, bringing non-linearity in the information. The parameters in the MLP are changed by an optimization algorithm that considers the observed input and desired output together with the output of the MLP. The error of the neural network output is propagated back through the network and at every node, the weights and biases are modified. This is done many times until the error of the neural network is minimized enough. By then, the neural network

can be considered trained with the existing data and for the desired purpose. The network can be modified by choosing different transform functions, how the network learns, and how information propagates through the network. There is a wide availability of ML forecasting algorithms, all with different aims and architectures. Different forecasting models are: SimpleFFN, DeepAR, MQCNN DeepSSM and SimpleLSTM, following (Calkoen et al., 2021) (Alexandrov et al., 2020). All these models are based on different neural networks and are elaborately explained by (Calkoen et al., 2021). All the hyperparameters and settings for the models are initially like (Calkoen et al., 2021) and only changed if necessary. A Neural Network (NN) class-specific for time series are Recurrent Neural Network (RNN). RNN is specially used for time series since they add extra information by order of observations when learning from inputs. However, classic RNNs experience a vanishing gradient problem when handling long time series. This problem was tackled by introducing Long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997). The use of a very basic LSTM showed promising forecasting results (Calkoen et al., 2021).
A short description of the forecasting algorithms in this paper:

- **The SimpleFFN** is an MLP Neural Network (NN) that predicts the next time steps by using the previous steps. The network uses as an input window of a pre-defined length and gives as an output a time series with a pre-defined length.

- **DeepAR** (Salinas et al., 2020) is an autoregressive RNN model that makes a global model by using historical data from time series. DeepAR uses the last time step of the input and some lagged values, depending on the frequency. This makes DeepAR autoregressive as it uses the values at the last step, but also RNN as the hidden nodes are returning their outputs to themselves, with LSTMcells.

- **Multi-quantile Convolutional Neural Network (MQCNN)** is built upon a sequence-to-sequence network. An advantage of this network is that it is capable of directly forecasting multi-horizon forecasts (Wen et al., 2017).

- **Deep state-space models (DeepSSM)** is a probabilistic forecast model that combines state-space models with Deep Learning (Rangapuram et al., 2018). The model generates linear state-space models for time series separately and with an RNN for learning the weights globally.

- **The SimpleLSTM** model designed by (Calkoen et al., 2021) contains a fully connected NN layer with one layer of LSTM cells. The model holds an encoder-decoder network, meaning it could transform input to an output with a different length. The input vector received is of length 26 (SDS from 1984 till 2009) and tries to forecast the forecast horizon with length 7 (from 2010 till 2016). The model is given as input pure SDS time series with no additional data. The LSTM cells output of length 64 are mapped with a linear sequential layer to length 7. Hyperparameters of the SimpleLSTM will be the same as used by (Calkoen et al., 2021).

2.4 Global Road Inventory Project (GRIP)

Global infrastructure or other physical boundary databases could provide more realistic local forecasts by projecting SDS on their geospatial location. One such database is the Global Road Inventory Project (GRIP) (Meijer et al., 2018). For GRIP, nearly 60 different geospatial datasets on road infrastructures were gathered and harmonized into a global dataset. The resulting dataset covers over 220 countries and consists of 21 million kilometers of global roads. This global dataset is available as a vector dataset but also as gridded layers for road length and road density on a $5^\circ \times 5^\circ$ arcminute resolution. For the GRIP database, national geospatial road data is used. The road data was searched

for by organizations that had an interest in this kind of data. Organizations like the United Nations (UN) organizations and the High Commission on Refugees, were consulted. Next to gathering information from organizations, OpenStreetMap was used to cover Europe and China. Furthermore, OpenStreetmap was used for more than 200 cities worldwide. All roads in the GRIP database are categorized into five categories: highways, primary roads, secondary roads, tertiary roads, and local roads, as can be seen in figure 2.15

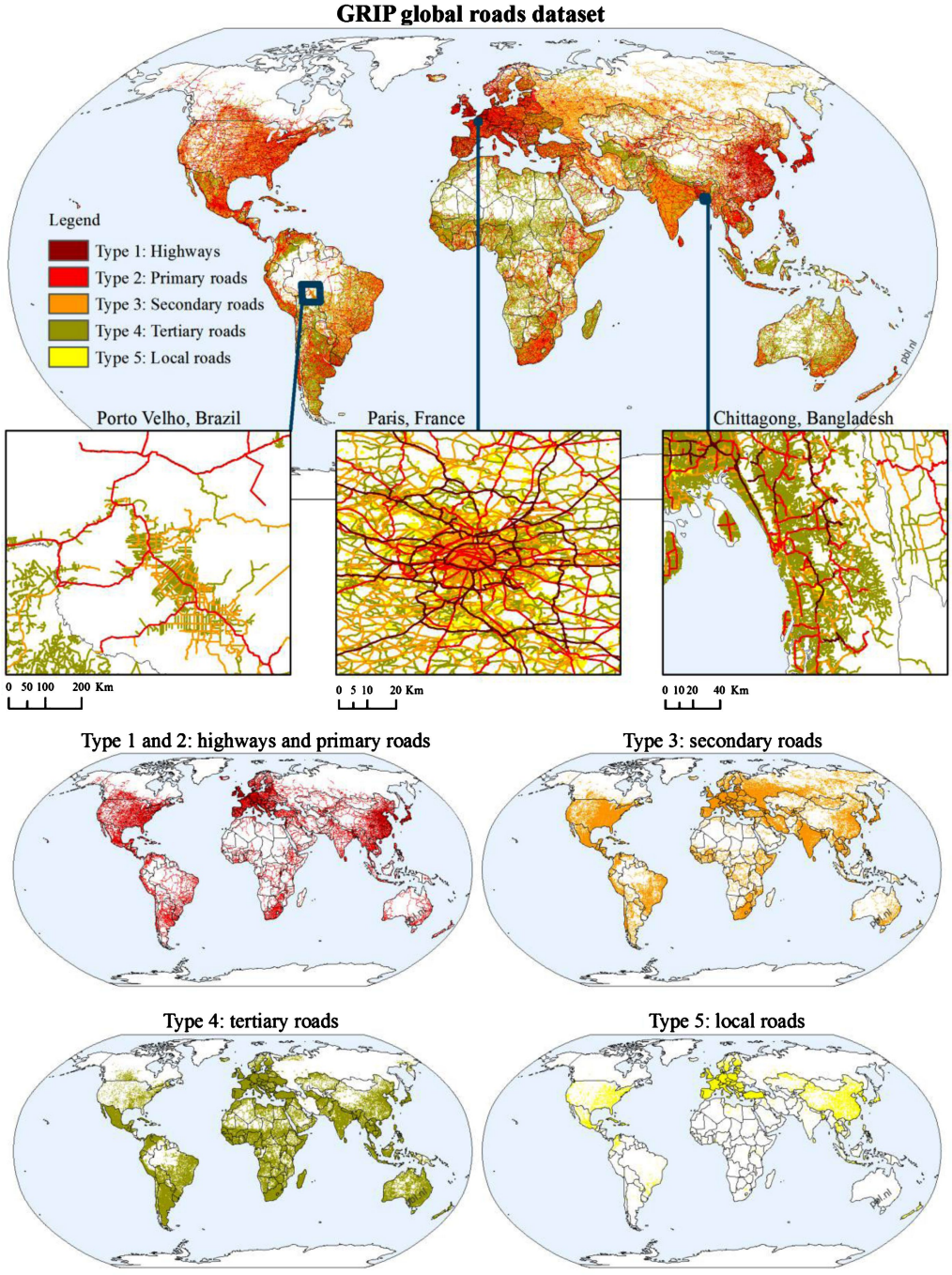

Figure 2.15: Grip Global Roads Dataset. The roads in the dataset are labelled by scale, ranging from highways to primary roads (Meijer et al., 2018).

Chapter 3

Methods & Results

In this chapter, the methods and corresponding results of this research will be presented. As discussed in Chapter 1 the aim is to improve shoreline forecasting by making a global model with a clustering approach. The approach of Bandara et al. (2020), a clustering approach will be used in this research. One key advantage of the clustering approach is that it adds more understanding to the data by clustering based on (given) features. Another key advantage is that it incorporates global historical data; it includes other local/neighboring information and it enables using different forecasting algorithms for different sets of time series. This is all particularly applicable to shorelines because the amount of shoreline data is immense and it is able to account for local biases. This research elaborates on previous research of Vousdoukas et al. (2020) and Calkoen et al. (2021) since these studies were also based on SDS data, and can therefore be used as a benchmark. The methods and results are combined in this chapter as the results per step and sub-question have a strong influence on the next steps. In Section 3.1 the research outline is presented with the relevant sections in this chapter.

3.1 Research outline

Figure 3.1 states the research outline with corresponding sub-questions. Before the sub-question and corresponding steps are considered, the data should be pre-processed. In the pre-processing step, time series with missing metadata are removed. In addition, missing values in time series are handled and a hydraulic threshold for data reliability is introduced. The detailed choices and procedures will be discussed in Appendix B.1 and B.2. In Section 3.2, the first sub-question is considered. This step will show the relevance of this research and show two clustering concepts and how they may improve forecasting. This sub-question has two steps, steps 1 and 2. In these two steps, the two directions for transect clustering are explored. In step 1, different time series features are presented. Forecast scores of different groups, generated by a classification based on a single time series feature, are presented. In step 2, some hydraulic and geomorphic features are reviewed, showing an understandable influence on shoreline behavior. In addition, it is examined whether these features show any correlation with shoreline behavior on a global scale. If there is any global correlation between transect behavior and hydraulic and/or geomorphic features, these features may be incorporated for clustering in step 3. Step 3 and step 4 will be discussed in Sections 3.4 and 3.5. In step 3 the information of the SDS historical behavior is extracted and transformed into features, such it is optimal for clustering. In addition, if the correlation was significant and useful, the features defined in step 2 are considered. Step 4 concerns the final clustering algorithm set-up. What similarity measure to use, what clustering prototype to use, and the clustering algorithm itself is considered. Also, the optimal number of clusters has to be defined and a representative dataset has to be designed to train the clustering algorithm. In the result section of step 4, the generated clusters are qualitatively

and quantitatively reviewed: whether the clusters improve understanding of the SDS data and corresponding shoreline behavior, and if the clusters could improve forecasting. In step 5 in Section 3.6 the forecasting models are chosen and trained with the clusters from step 4. The forecast models will be evaluated per cluster but also on the transect level. Steps 3, 4 and 5 are highlighted in a green box as these are highly iterative during the research. Finally, in Section 3.7 the last step, step 6, the application of the final predictions are considered. The predicted transects are projected on their geo-spatial location and are checked for any interference with local (physical) boundaries. By combining the predictions with local boundaries, the predictions become more realistic on a local scale. The last paragraph per step presents a short summary of the results of the step and the implications the results have on the following step(s). This paragraph will be highlighted in a dashed black box. All data and results are processed and built in a Python environment. Several packages are used during the research. Linear fits are determined with SciPy 1.5.2. The SimpleLSTM model is built with Torch 1.7.1 (Paszke et al., 2019) and all other forecasting models are made with GluonTS 0.6.4 (Alexandrov et al., 2020). Handling and plotting the transect's geographical location or direction was done using the GeoPandas package (Jordahl, 2014). Deriving the DWT was done the PyWavelets package (Lee et al., 2019).

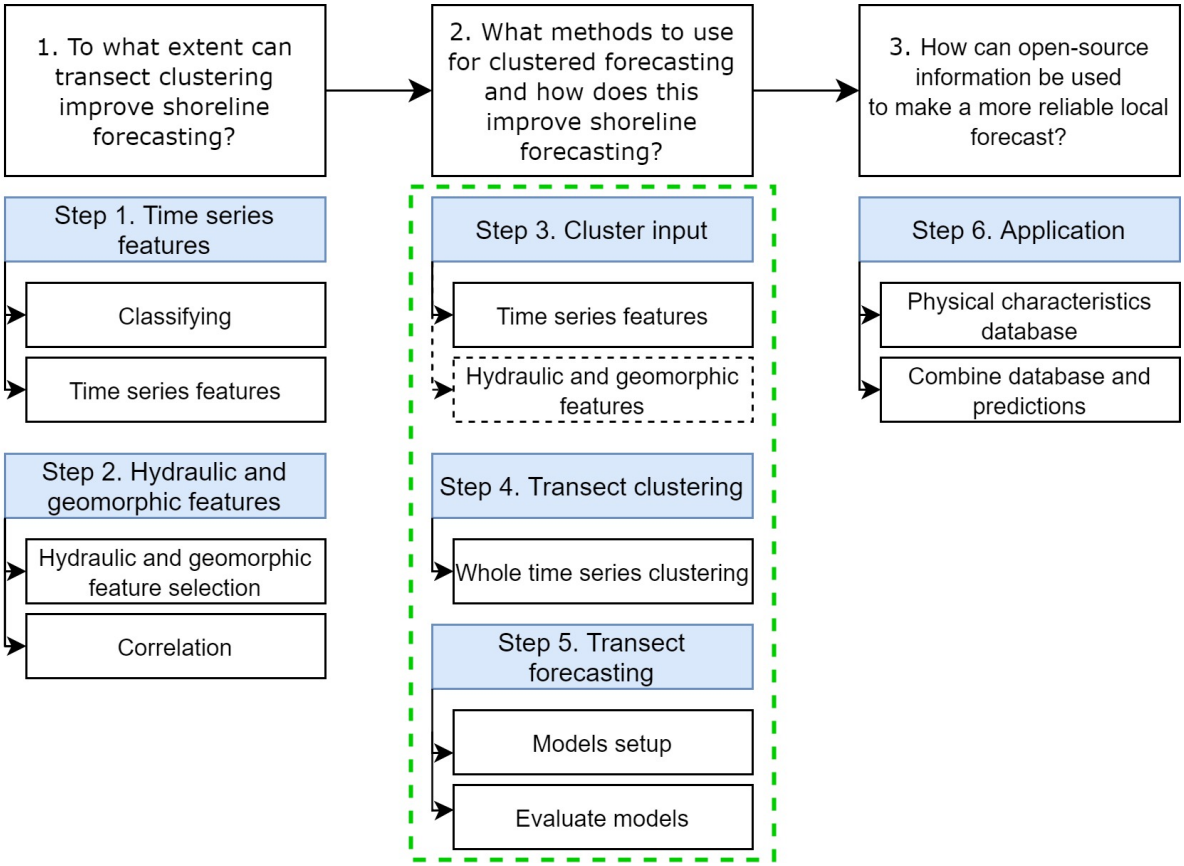

Figure 3.1: Research outline used in this report. In step 1, the concept of the clustering approach is presented by classification based on time series features. In step 2, publicly available information is examined for correlation with transect behavior. If a useful correlation is found, this could be incorporated in step 3. Steps 3, 4 and 5 are outlined in green as these steps are highly iterative. Step 3 and step 4 will investigate the clustering of the transects. Step 5 will research the forecasting of transects with the given clusters from step 4. Eventually, in step 6, the predictions per transect are combined with open-source information to obtain a more realistic forecast.

3.2 Step 1. Time Series features

In this section, the first step of the first sub-question will be answered. This first step aims to prove that by making groups of transects based on a single (time series) feature, useful insights in shoreline forecasting could be gained, as shown in previous work (Bandara et al., 2020). There are two ways how clustering of transects could benefit forecasting:

1. Clustering could give insights into transects' behavior and the performance of clusters per forecasting algorithm. This clustering approach could result in tailored fit forecasting algorithms per cluster.
2. Clustering could improve input uniformity and so improve forecasting results (Bandara et al., 2020). As large datasets have a wide spread of data, clusters could create more uniform input.

Step 1 is answered in two sub-steps. First, transects are grouped based on transect features. Second, the groups are used as input for the SimpleLSTM algorithm (Calkoen et al., 2021). The last sub-step will show differences in forecasting performances and so help to improve forecasting per group. Additional to the two sub-steps described above, a short investigation is done on the different features that can be extracted from time series, following literature and hydraulic knowledge. The difference and spread of the different time series features highlight the clustering potential the SDS dataset has.

3.2.1 Classifying

This subsection focuses on the first part of step 1, showing that grouping may improve shoreline forecasting. For this part, different groups of transects will be used as an input for the SimpleLSTM model. The concept of a clustered forecasting approach is that by using a group with similar transects, forecasting performance can be improved. For making these subsets in this subsection, a simple classifying approach is chosen (supervised learning). The classifying criteria will be based on the change rate of the transects and will follow the classification proposed by (Luijendijk et al., 2018). Most clustering algorithms like K-means are based on multidimensional clustering. Since the metadata only gives information about the time series' features in the form of change rate and changerate_unc a one-dimensional classifying approach is chosen for this sub-question. The change rate is the mean trend over 33 years in m/year and the changerate_unc quantifies the overall error of the linear regression per transect. A changerate_unc close to 0 indicates that the transect shows linear behavior and a high changerate_unc implies non-linear behavior. More features and information can be added to make clusters based on multiple features in a later stage. For this sub-question, a one-dimensional approach is sufficient to answer the sub-question.
For the classification of Luijendijk et al. (2018), extra distinctions are added for accretion transects: Accretion 0.5 to 1 m/year, Intense Accretion 1 to 3 m/year, Severe accretion 3 to 5 m/year and Extreme Accretion > 5 m/year. The representative subset used for this sub-question will have a size of 20k transects. Next to the classification based on change rate, a different group based on the non-dimensional changerate_unc feature is added. A different group based on changerate_unc < 0.1 will be used. Below are the groups with the number of transects in every group, see table 3.1. To verify, the number of clusters in group zero till eight sums up to 20,000 transects.
Finally, there are ten groups of transects: nine groups based on the distinction by Luijendijk et al. (2018) and one group based on the changerate_unc < 0.1 criteria. All groups will be individually be used for training a forecast model. As mentioned before, the SimpleLSTM model is used as the starting point of this forecast step (Calkoen et al., 2021). Per group, a total of 5 runs was done, for each run a different training, validation, and test set was used to cross validate the transects in every group over the last seven years. Per run the evaluation measures were calculated and in the end,

Table 3.1: Groups generated from sample of 20,000 transects for step 1. First column holds the trend per group, second column the grouping criteria and the other columns the number of transects and percentage of total.

0. Moderate / stable	-0.5 to 0.5 m/year	N = 11688	58.4%
1. Erosion	-1 to -0.5 m/year	N = 2103	10.5%
2. Intense erosion	-3 to -1 m/year	N = 1397	7.0%
3. Severe erosion	-3 to -5 m/year	N = 172	0.9%
4. Extreme erosion	<-5 m/year	N = 148	0.7%
5. Accretion	0.5 to 1 m/year	N = 2541	12.7%
6. Intense Accretion	1 to 3 m/year	N = 1618	8.1%
7. Severe Accretion	3 to 5 m/year	N = 220	1.1%
8. Extreme Accretion	>5 m/year	N = 113	0.6%
9. Changerate_unc	<0.1	N = 1905	9.5%

the mean per forecasting error over the total runs was is eventually calculated. The evaluation measures Mean Absolute Error (MAE) and Mean Squared Error (MSE), used in this step are described in equations 3.1 and 3.2. Here the $z_{i,t}$ is the observed value and $\hat{z}_{i,t}$ the predicted value and N the number of points, in this case, seven. The results are presented in Subsection 3.2.3.

$$\mathrm{MAE}_i = \frac{1}{N}\sum_{t=1}^{N}\left|z_{i,t} - \hat{z}_{i,t}\right| \tag{3.1}$$

$$\mathrm{MSE}_i = \frac{1}{N}\sum_{t=1}^{N}\left(z_{i,t} - \hat{z}_{i,t}\right)^2 \tag{3.2}$$

3.2.2 Different features

Next to the simple grouping described above, an investigation of the different features in the time series is done. This investigation shows the spread in features in the dataset and the possibilities for clustering input. It shows the potential of clustering by showing the number of different features that can be extracted from time series. To enforce the potential, some graphs are plotted to show the lack of correlation between features. Where no correlation indicates that the features hold unique information for clustering purposes. General features that can be used for time series clustering are presented in figure 3.2.
Next to standard time series features, also features within the hydraulic context could be used:

- **Absolute sum of all steps [m]**
  This parameter indicates whether the transect is acting around an equilibrium, when the value approaches 0 it means that the transects did not alter that much in the last 33 years (1984-2016).

- **Sum of all absolute steps [m]**
  This tells something about the magnitude of the transects' behavior. A big value of this parameter indicates large differences between the different years, indicating it as a high volatile transect. This could mean that the transect has a wide spread of different forcing and other influences as it behaves with big steps.

- **Change rate [m/year]**
  This is the same feature as the Trend feature in figure 3.2. This parameter is determined by linear regression on the raw time series. The slope of the linear regression is the change rate.

The change rate indicates the trend of the transects, whether it shows erosion or accretion behavior.

- **Autocorrelation function with lag 1** [-]
  The Auto Correlation Function (ACF) is an indication of how the data points in a stationary time series are connected. It indicates the similarity between observations $t-1$ and t. The autocorrelation function is only applicable on stationary time series, so the transects are differenced.

Feature	Description
Mean	Mean.
Var	Variance.
ACF1	First order of autocorrelation.
Trend	Strength of trend.
Linearity	Strength of linearity.
Curvature	Strength of curvature
Season	Strength of seasonality.
Peak	Strength of peaks.
Trough	Strength of trough.
Entropy	Spectral entropy.
Lumpiness	Changing variance in remainder.
Spikiness	Strength of spikiness
Lshift	Level shift using rolling window.
Vchange	Variance change.
Fspots	Flat spots using disretization.
Cpoints	The number of crossing points.
KLscore	Kullback-Leibler score.
Change.idx	Index of the maximum KL score.

Figure 3.2: Time series features (Hyndman et al., 2016)

To visually show that two features are uncorrelated, correlation plots can be made with normalized features; one feature is sorted and plotted, another feature for the same transect is scattered. Figure 3.3 shows an example of this type of plots. In the first plot, the normalized kurtosis values of a subset of 2000 transects are sorted and plotted. For every transect, the corresponding normalized sum of all absolute steps [m] are scattered. Figure 3.3 shows that the scattered values do not follow the distribution of the sorted feature. This indicates that both features are not (linearly) correlated. The same applies to the other plots shown in figure 3.3, as no visual correlation is present. The absence of direct correlations and dependencies highlights the potential to cluster based on these features. Clustering with every feature will lead to different clusters.

3.2.3 Results

In table 3.2 the aggregated mean MSE andMAE for the all the first nine groups are stated together with the performance of the total group of 20k transects.

Table 3.2: Forecast evaluation measures for sub-groups of table 3.1, group 0 till 8. Lower scores indicate higher accuracy. MSE is in m^2 and MAE in m.

Group:	**0.**	**1.**	**2.**	**3.**	**4.**	**5.**	**6.**	**7.**	**8.**	**All**
MSE	285.05	418.08	565.54	1197.82	2949.21	334.64	796.07	1343.34	3164.81	381.69
MAE	12.38	15.57	17.03	25.45	38.52	13.51	21.37	26.65	41.92	14.05

There is a big difference in forecasting performance for the eight groups based on the change rate criteria. For the groups with extreme behavior transects, the forecasting errors are all significantly

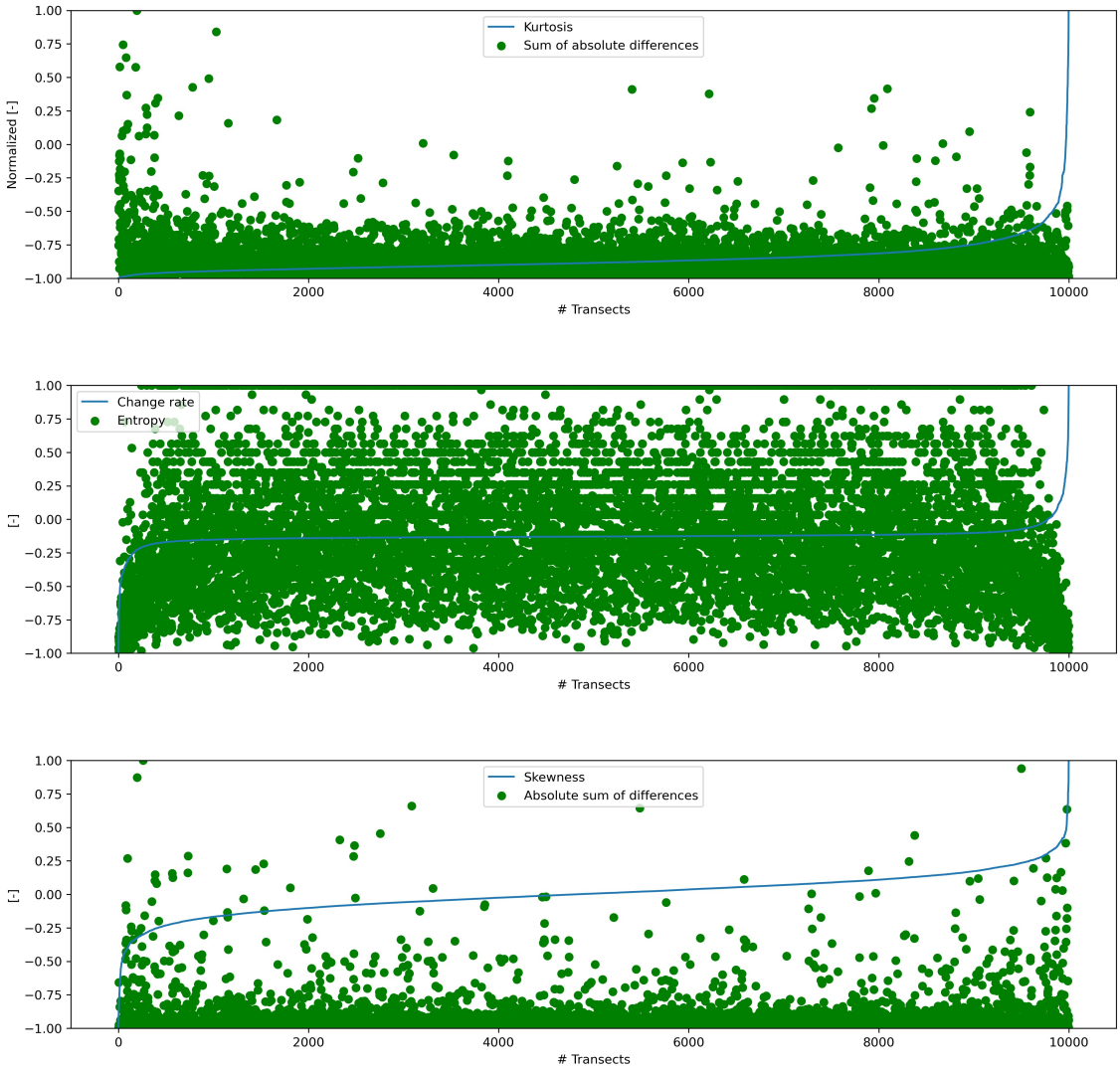

Figure 3.3: Example plot of normalized sorted features and normalized scattered features. Blue lines present sorted normalized values for a feature and the green scatter points the corresponding normalized value of the other feature. As there is no visible relation, no (linear) relations between features are present. No correlation between features highlights the potential of clustering based on features, as each feature captures unique information of the transect.

higher than those with more moderate transects. This difference in forecasting performance gives insights into the composition of the accuracy of the whole group. It can be stated that the more extreme groups (groups 4 and 8) are less well predicted by the SimpleLSTM model than the more moderate groups (groups 0, 1 and 5). These extreme groups may perform better with another forecasting algorithm than SimpleLSTM.

As expected, this simple grouping gave insights into transects' behavior and the performance of clusters per forecasting algorithm. These insights in different transects behavior per group and the subsequent forecasting accuracy with the SimpleLSTM per group can improve overall forecasting.

Next, when considering the changerate_unc group, group 9, the forecasting measures are significantly better than the whole group measures. This improvement indicates that transects with linear behavior are relatively well predicted with the SimpleLSTM model. Further, the changerate_unc transects were better predicted by only using similar transects as training input. This could be proved by considering all the transects in the whole group that have a changerate_unc value smaller than 0.1. In other words, considering all the forecast measures of the transects with a changer-

Table 3.3: Forecast evaluation measures for group 9 (change_rate <0.1). Lower scores indicate higher accuracy. MSE is in m^2 and MAE in m.

Group:	**9.**	**All**	**9. (All)**
MSE	62.86	381.69	97.92
MAE	5.78	14.05	7.63

ate_unc value smaller than 0.1 in the whole group, these errors are higher. Looking at table 3.3, in the first column, the forecasting measures for the changerate_unc group are defined and in the last column the forecasting measures for the same transects from the whole group are stated. The conclusion that can be drawn here is that by creating a more uniform input, the forecasting performance improves significantly.

Summary & implications next step(s)

Grouping transects based upon change rate and the linearity feature changerate_unc resulted in meaningful insights in transects behavior and the forecasting accuracy per cluster. This shows the relevance of the thesis and confirms the concept and hypothesis that forming insightful groups of similar transects could improve forecasting. Next to the quantitative confirmation, some features extracted from the transects show no or minimal correlation. This indicates that transects can be represented by different features that hold unique information about the transects. This unique information could be used in future clustering. In the following step, step 2, the focus will be on hydraulic and geomorphic features for clustering.

3.3 Step 2. Hydraulic and geomorphical features

After proving that by using time series features, insights are gained that could improve forecasting of transect's behavior, the next step is examined. This step considers publicly available information that could be used to create meaningful groups of time series. This information will be in the form of hydraulic and geomorphic features. Since an enormous amount of external forcing influences shorelines, this information could be helpful for clustering. First, different (primary) hydraulic and geomorphic features are derived from local case studies and literature. These features show a direct correlation with transect behavior on a local scale and are recognized in current literature. Besides coverage on a local scale and recognition in literature, the public availability of features is considered for further usage. After the hydraulic and geomorphic features are defined, the correlation on a global scale is examined. It is examined whether the defined features directly correlate with the transect's behavior on a global scale. If relevant and significant correlations are found, these could be further incorporated in step 3 for further clustering purposes.

3.3.1 Hydraulic and geomorphic feature selection

Following literature, different variables are influencing the shoreline evolution on large timescales (Stive et al., 2002). For some cases, the SDS data projected in the Shoreline Monitor can also be clarified by geomorphic and hydraulic conditions. In the following section, several case studies are presented in order to explain hydraulic and geomorphic influence on transect behavior. Every figure shows sandy shorelines and is therefore relevant for the study. The red transects show eroding behavior and the green accretion. All the figures are orientated such that the upper part is the north, the right the east, etc.

Australia east coast (Lat -34.8° Lon 150.8°)

The shoreline depicted in figure 3.4 is a concave shoreline named Berrys Bay in East-Australia. The green transects indicate accretion behavior, this is according to expectation as confirmed in (Lakhan & Pepper, 1997). Concave beaches show higher tendency toward accretion than convex or straight beaches.

Figure 3.4: Concave shoreline in the East coast of Australia. The the green transects indicate accretion behavior, the length of the transect is correlated with the magnitude of accretion. Source: Shoreline Monitor.

Angola coast (Lat -16.7° Lon 11.6°)

In figure 3.5 the small island, Baia dos Tigres, off the coast of Angola, shows accretion and erosive behavior. Following ERA5 open-source data, the mean wave angle of 2016 was 204°. This wave angle indicate that the waves were mainly directed from the south-south-west. Considering the mean wave direction and the relative position of this island, the southern part of this island is more exposed to wave forcing than the northern part of the island. The relative wave incident can clarify this. The wave incident on the southern part of this island is around 45°. Following literature, the maximum longshore transport is at a relative wave incident of 45°, which confirms the behavior observed in this case (Siegle et al., 2007).

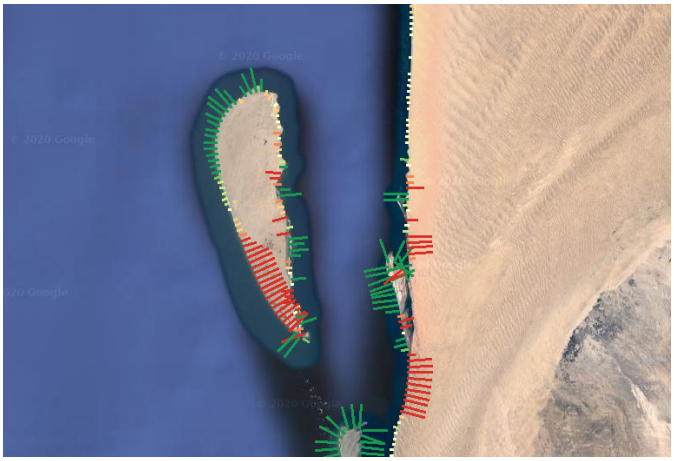

Figure 3.5: Exposed island off the coast of Angola, the southern part of the island is heavily exposed to incoming waves. The red transects indicate erosive behavior for the southern part, the length of the transect is correlated with magnitude of erosion. Source: Shoreline Monitor.

Mexico coast (Lat 18.7° Lon 92.0°)

The third case study shown in figure 3.6 is located in the Mexican Gulf. This shoreline has a curved form and is convex, indicating it is exposed to external forcing. Next to the convex form the shoreline follows, the mean wave direction following ERA5 of 2016 was 42°. The waves were coming from the northeast direction. The combination of relative wave direction and convex form of the shoreline is expected to result in erosive behavior. This tendency can be seen in figure 3.6, which shows heavy erosive behavior.

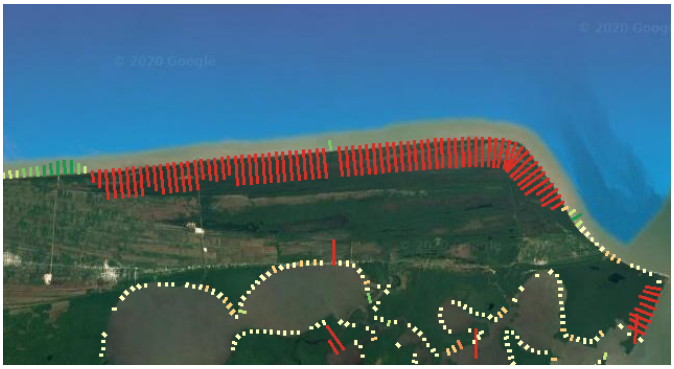

Figure 3.6: Exposed shoreline in the Mexican Gulf. The red transects indicate erosive behavior, likely due to the convex shape and angle of incoming waves. The length of the transect is correlated with magnitude of erosion. Source: Shoreline Monitor.

Global hydraulic and geomorphic features

As described by the examples in figures 3.4, 3.5 and 3.6, besides clustering driven by features extracted from the raw 33 years long SDS data, hydraulic and geomorphic features from SDS transects may provide additional information for meaningful clusters. Shoreline evolution is a multi-dimensional problem that cannot be completely captured in a small set of features. However, some hydraulic and geomorphic conditions are more dominant than others.

The three case studies described above show an understandable and direct correlation between hydraulic and geomorphic features with transect behavior. To use this information in further steps, this subsection is necessary. In this subsection, a few significant (and publicly available) features are described. These features are used to research possible dependencies within the yearly transect data. Previous research already showed that SDS could be used as an ideal tool for unraveling coastline dynamics on seasonal, inter-annual, and decadal scales governed by the seasonal forcing, climate variability, and land subsidence, respectively (Wang, 2018). However, the SDS used in that specific research had a higher temporal resolution than the yearly SDS data in this research.
To find valuable features for shoreline forecasting, dependencies between historical shoreline behavior and these features must be found. On a local scale, relations between hydraulic and geomorphic on one side and transect behavior on the other side are found in the cases above. However, this research step aims to find applicable relationships on a global scale. The list of features considered is presented below with a short explanation based on present hydraulic knowledge of how these features/variables may influence shoreline behavior. The features in this subsection are chosen based on their public availability and (dominant) influence on shoreline evolution following literature. The features set could be immense in theory, as the coastal zone experiences complex interactions between numerous phenomena.

- **Concave and convex beaches**
  The curvature of shorelines influences the amount of exposure. A hidden and concave shoreline is less frequently subject to forcing like waves and storms. Less forcing could lead to more moderate behavior of shorelines. Convex beaches on the opposite are shorelines exposed to the ocean and therefore experience the forcing in the coastal zone, thus having a higher tendency towards erosive behavior (Lakhan & Pepper, 1997). Besides concave and convex shorelines, there are extreme variants for both: headland beaches and pocket beaches. Headland beach will show extreme erosive behavior as no sediment can be stored due to the heavily forcing. On the other side are pocket beaches known for "absorbing" sediment from other coastal cells and should therefore show accretion.

- **Nearshore slopes**
  Nearshore slopes are important in the transition from ocean waves towards waves approaching shorelines. As more moderate slopes will absorb more energy from arriving (wind) waves, direct forcing on the shoreline will be less, resulting in less extreme shoreline behavior.

- **Wavedata** is expressed using the following two variables:

  - **Significant wave height**
    This wave height is defined as the mean of the 1/3 highest of all wave heights measured. The square of significant wave height $H_s{}^2$, represents the energy contained within a wave field. A higher amount of wave energy forcing on the coast could lead to more (extreme) variable shoreline behavior.

  - **Wave direction**
    This parameter is an indicator of the wave incident on shorelines. The angle of approach

is directly connected with the amount of longshore transport. This longshore transport could transport sediment from coastal cells towards other coastal cells, leading to erosion or accretion.

Concave and Convex

For testing the exposure of shorelines, the data self is used and no external information source is needed. The transects are sorted based on their geographical location and by their location on the shoreline. For this step, non-sandy transects are used as well. Every non-sandy transect is marked and all sandy transects between two non-sandy transects are labeled to the same shoreline stroke. For every shoreline stroke, the intersect coordinates of every transect belonging to that shoreline are scattered. Finally, per shoreline stroke, the first and last scattered transect locations are connected using a straight line. All the transects located on the left side of this line are then flagged as concave transects and all the transects on the right side are flagged as convex transects. In figure 3.7 the above-mentioned procedure is conducted for a shoreline in the South of Australia.

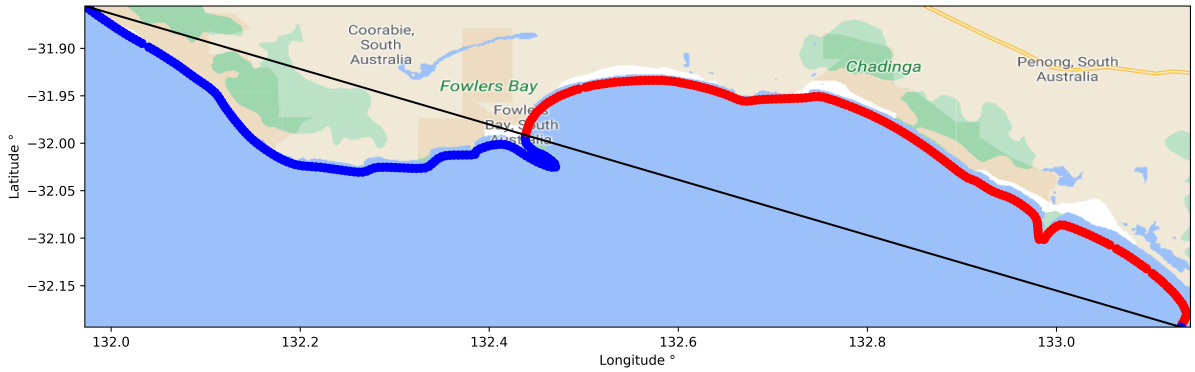

Figure 3.7: Convex and concave labelling for shoreline in South-Australia. Blue transects are labelled with a 1. indicating it is a transect on a convex shoreline. Red transects are labelled with a 0, indicating it is a transect on a concave shoreline. Figure created with Google Maps.

Nearshore Slopes

The nearshore slopes are deducted from an open-source from (Athanasiou et al., 2019). This dataset consists of a global distribution of nearshore slopes and depth of closures with a spatial resolution of 1 km along the global coastline. All the transect coordinates are matched with the spatial coordinates by finding the nearest defined slope and assigned to a transect.

Wave data

All the wave data used is from the ERA5 database (ECMWF) (Hersbach et al., 2020). ERA 5 is a successor of the ERA-interim database and provides hourly data on a global scale with a horizontal resolution 0.5°, 0.5° with different vertical levels. The data used here are the monthly means of 2016, the monthly means are then averaged again to the yearly mean of 2016 to obtain a single value. Only data of 2016 is considered since the mean of twelve different values is considered adequate. The data is downloaded with a spatial resolution of (0.5°, 0.5°). The transects within these spatial boxes add the corresponding wave data towards the metadata of the transects. The spatial resolution (31 km) is relatively large compared to the transect spatial scale (0.5 km). Some transects will not directly fit in the spatial boxes of the ERA5 data, the adjacent boxes are then used to find the wave data. By looking at the eight neighboring raster boxes of the ERA5 data and taking the mean of the boxes with wave data, most transect will be directed to the appropriate wave data (see figure 3.8 for an example).

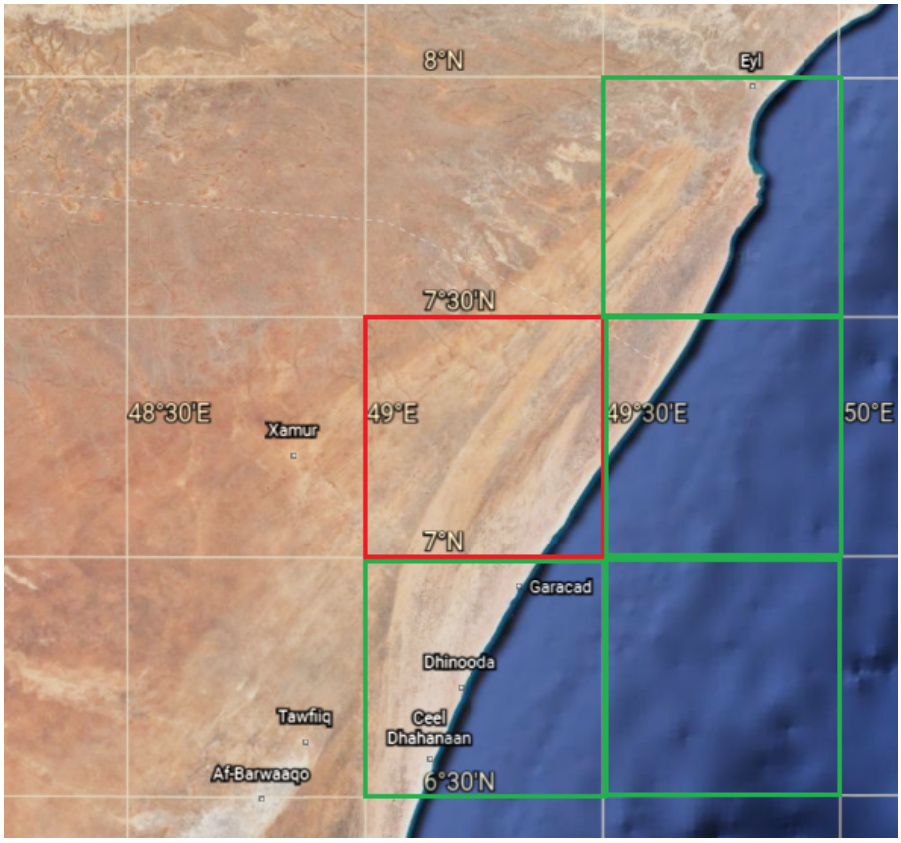

Figure 3.8: Screenshot of Google Earth east coast of Somalia. Example of procedure capturing wave data if transects fall in the red box (without wave data), taking mean of adjacent green boxes with wave data. Longitude and latitude lines in the figure follow the same resolution as ERA5 data (0.5°, 0.5°). Figure created with Google Earth.

After this procedure, still some transects are not assigned wave data. After visual inspection, these transects were only observed in the hidden sea where there is no wave data. However, since these transects lay in hidden seas, the significant wave height will be relatively low compared to open seas. These transects, therefore, will have a wave height assigned to half the mean wave height in the whole sample. For the mean wave direction, the value assumed for the transects in hidden seas is NAN, since the significant wave height will be low, the influence of the mean wave direction is assumed to be insignificant.
An additional step is needed after assigning the mean wave direction and significant wave height for both wave data values. First, for the significant wave height, the values are squared because the $H_s{}^2$ represents the energy contained in the wave field. Second, the wave direction is presented relative to the north pole, meaning that waves propagating from the north pole direction have angle 0° and waves propagating from the east have angle 90°. The mean wave direction should be expressed relative to the transect. To do so, the transect orientations are calculated with the metadata. As a result, the absolute difference between wave direction and the transect's orientation indicates the incoming wave angle. Relative wave angles larger than 90° or smaller than −90° were assigned a NAN value as these relative wave incidents are physically impossible.

Clustered coastal zones based on wave energy

Previous research tried to cluster different coastal zones by analyzing the spatial variability of global wave climate (Fairley et al., 2020). The clustering was meant to identify potential locations for the deployment of wave energy converters (WEC's). Using the ECMWF ERA5 open-source reanalyzed

data, six different clusters were formed with two different input datasets. The first set was a simple set, considering the significant wave height and peak wave period, groups based on W class. The second set included a wide range of relevant wave parameters, the WXSD class. The considered parameters are split into four groups:

- **Basic wave characteristics (W)**
  The square of significant wave height ($H_s{}^2$) represents the energy within a wave field, the peak wave period (T_p), and the coefficient of variation (CV). The CV is a non-dimensional ratio between the standard deviation and the mean and has been used in wave resource researches (Reguero et al., 2015).

- **Storm survivability and risk (X)**
  A risk factor of $\frac{H50}{Hs}$ was utilised.

- **Spectral characteristics (S)**
  Denoted by:
  $$Q_p = \frac{2}{m_0^2} \int_0^\infty f E(f)^2 df \tag{3.3}$$

- **Directional characteristics (D)**
  These characteristics consider the deviation of the mean wave direction. Also the wave directional width (WDW) is used, as it represents the directional spreading in wave energy about the mean wave direction.

The two sets of input were clustered with K-means and resulted in six clusters per input set, see figure B.7 and B.8. These six clusters per set (W class and WXSD class) were used as input in the following subsection to find correlations. The characteristics per cluster can be found in Appendix B.3.

3.3.2 Correlation

After adding the geomorphic and hydraulic data to the metadata of the transects, dependencies between these features with the transect's behavior are examined. These dependencies can be identified by looking for any correlation with the described features and transect behavior. To embody the transect's behavior, some features are extracted, as it is challenging to find a direct correlation between one-dimensional features and shapes of transects.
For the representation of the transects the following features of Section 3.2 were used:

- **Absolute sum of all steps [m]**
  This parameter indicates whether the transect is acting around an equilibrium, when the value approaches 0 it means that the transects did not alter that much in the last 33 years (1984-2016).

- **Sum of all absolute steps [m]**
  This tells something about the magnitude of the transects' behavior. A big value of this parameter indicates large differences between the different years, indicating it as a high volatile transect. This could mean that the transect has a wide spread of different forcing and other influences as it behaves with big steps.

- **Change rate [m/year]**
  This is the same feature as the Trend feature in figure 3.2. This parameter is determined by linear regression on the raw time series. The slope of the linear regression is the change rate. The change rate indicates the trend of the transects, whether it shows erosion or accretion behavior.

- **Autocorrelation function with lag 1 [-]**
  The Auto Correlation Function (ACF) is an indication of how the data points in a stationary time series are connected. It indicates the similarity between observations $t-1$ and t. The autocorrelation function is only applicable on stationary time series, so the transects are differenced.

The geomorphic features (curvature and nearshore slopes), will only be examined for the transects in Australia. The transects need to be ordered to extract the curvature features, the ordered transect data for Australia is provided by (Kras, 2019). The hydraulic features will also be examined on a global (sample) dataset. The correlation that will be used is the Spearman ranked correlation. The correlations are all checked for significance with the p-value. A p-value less than 0.05 ($\alpha \leq .05$) is considered significant. A p-value lower than 0.05 indicates that the NULL-hypothesis can be rejected, where the NULL-hypothesis states that the features are uncorrelated with transect behavior.

3.3.3 Results

As described in Subsections 3.3.1 and 3.3.2, correlation between geomorphic and hydraulic features on one side and transect features was examined. If there was any significant correlation, the geomorphic and/or hydraulic feature could be incorporated for the clustering of transects. In table 3.4 and 3.6 the correlations for two datasets are stated. In table 3.4 a random and representative global subset of 22,388 transects is used as input for this step. This subset is a representation of the whole dataset, considering all transects globally distributed and with all sorts of behavior. In the columns, the hydraulic features are presented and in the rows, the features extracted from the time series. As can be seen, the maximum absolute correlation found is between the squared significant wave height and the sum of absolute differences [m], of 0.178. This indicates that whenever the significant wave height increases, the transect's behavior will have a higher magnitude, meaning that the stepsize between every year will increase. However, as the correlation only holds 0.178, it is considered too small and not usable for further clustering purposes. In table 3.5 the corresponding p-values of the correlations in table 3.4 are stated. The cells that show p-values higher than 0.05 indicate that for these features, the correlations are insignificant. Meaning that the NULL-hypothesis cannot be rejected. The insignificant correlations do all occur for the change rate feature. This can be explained by the fact that the change rates can differ very locally and correlation is hard to find between the SDS data with the low(er) resolution classes of Fairley et al. (2020) and ERA5 data.

Table 3.4: Spearman correlation between hydraulic features in columns and transect features in rows. Table contains mean spearman correlations of sample 22,388 transects on global scale. The maximum absolute correlation found here is 0.178 between ${H_s}^2$ and Sum of absolute differences.

n = 22,388 transects (World)	**W class**	**WXSD class**	**Relative wave angle**	**Squared significant wave height**
Absolute sum of differences [m]	0.051891	0.053713	-0.032430	0.022032
Sum of absolute differences [m]	0.090041	0.116177	-0.056351	0.177771
change rate [m/year]	-0.001835	0.003243	0.008874	0.020339
Autocorrelation function (lag=1)	0.070356	0.042355	-0.033488	-0.076357

In table 3.6 all the sandy transects of Australia are used, a total of 24,769 transects. As all the transects in this subset were ordered along the coastline, the shape of shorelines could be obtained (Kras, 2019). The maximum absolute correlations found here are the classes of Fairley et al. (2020) together with the Auto Correlation Function (ACF) with lag 1, a correlation around -0.22. This value indicates that a higher energy sea state will lead to a lower ACF value. An ACF around 0 indicates that the position of the shoreline at $t-1$ barely influences the position of the shoreline a t. The correlations in table 3.6 have a maximum absolute (significant) correlation of 0.22. Furthermore, the correlation between the squared significant wave height and the sum of absolute differences [m] is similar to

Table 3.5: P-values spearman correlation of table 3.4 between hydraulic features in columns and transect features in rows. Correlations with p-values ≤ .05 indicate significance and thus the NULL-hypothesis can be rejected. The NULL-hypothesis is that the features are uncorrelated with transect behavior.

n = 22,388 transects (World)	W class	WXSD class	Relative wave angle	Squared significant wave height
Absolute sum of differences [m]	0.00000	0.00000	0.000267	0.002437
Sum of absolute differences [m]	0.00000	0.00000	0.00000	0.00000
change rate [m/year]	0.800747	0.655487	0.318689	0.005140
Autocorrelation function (lag=1)	0.00000	0.00000	0.00000	0.00000

the one found in 3.4. Similarly to the results in table 3.4, the correlations in table 3.4 are relatively low and are therefore not useful in further clustering purposes. In table 3.7 the p-values for the correlations in table 3.6 are stated. Again the p-values for the change rate value are insignificant, for all other correlations, the p-value is lower than 0.05. However, all correlations with a significant p-value show correlations in table 3.6 that are too low to add extra valuable information to clustering.

Table 3.6: Spearman correlation between hydraulic and geomorphical features in columns and transect features in rows. Table contains mean spearman correlations of sample 24,769 transects of Australia. The maximum absolute correlation found here is 0.22 between the WXSD cluster and the Autocorrelation function

n = 24,769 transects (Australia)	W class	WXSD class	Relative wave angle	Squared significant wave height	Concave Convex	Nearshore Slope
Absolute sum of differences [m]	-0.072913	-0.073437	0.019280	-0.070169	-0.048133	-0.070289
Sum of absolute differences [m]	0.093237	0.109997	-0.129257	0.132335	0.051033	0.136779
change rate [m/year]	0.057180	0.056014	0.003709	0.073788	-0.009641	0.061076
Autocorrelation function (lag=1)	-0.211113	-0.220220	0.142426	-0.212177	-0.073012	-0.213440

Table 3.7: P-values spearman correlation of table 3.6 between hydraulic and geomorphical features in columns and transect features in rows. Correlations with p-values ≤ .05 indicate significance and thus the NULL-hypothesis can be rejected. The NULL-hypothesis is that the features are uncorrelated with transect behavior.

n = 24,769 transects (Australia)	W class	WXSD class	Relative wave angle	Squared significant wave height	Concave Convex	Nearshore Slope
Absolute sum of differences [m]	0.00000	0.00000	0.009975	0.00000	0.00000	0.00000
Sum of absolute differences [m]	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
change rate [m/year]	0.00000	0.00000	0.620184	0.129212	0.00000	0.00000
Autocorrelation function (lag=1)	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Summary & implications next step(s)

In this section, the first and second steps of the diagram in figure 3.1 are investigated. In this subsection two different ways of transect clustering were explored. In step 1, current literature findings were confirmed: by making groups of similar time series, valuable insights in forecasting performance could be gathered. In step 2, transect clustering based on hydraulic and geomorphic features was examined. Before clustering could be done with these kinds of features, correlation with transect behavior had to be determined. On a local scale, a direct correlation could be found. However, on a global scale the correlations found were not useful. This is most likely caused by the mismatch in the resolution of the SDS data and public information. The Shoreline Monitor holds transects per 500 m and the ERA5 data contained data with a spatial resolution of approximately 50 km x 50 km. For the latter of the research, only time series features derived from transects will be used as clustering input.

3.4 Step 3. Cluster input

The hydraulic and geomorphic features defined in Subsection 3.3.1 show no useful correlation with transect's behavior as presented in Subsection 3.3.2. This indicates that these features would not add any extra useful information for clustering of SDS data. The hashed box in the research outline in Subsection 3.1 is not relevant in the remainder of this research. The clustering input will focus only on the representation of the time series features extracted from the SDS data. In this step, the clustering input for step 4 is defined.

There are two types of time series representation relevant for this clustering problem: feature-based clustering and shape-based clustering. Feature-based clustering can diminish black-box-feeling as the time series are expressed in a vector with meaningful features, describing its behavior. However, feature-based clustering demands high knowledge of the data and the final purpose of the model because appropriate features should be selected for the time series and aim. The clustering outcomes highly depend on these characteristic selections. Shape-based clustering considers the shape of the time series and uses their shape as cluster input. Using time series shapes as input is most of the time computational more demanding. For the time series representation three requirements were created:

1. the representation should preserve the physical understanding of the transects. It should still be possible to derive meaningful hydraulic behavior from the representation.

2. the representation should focus on the long-term behavior of the transects. As this research aims to create long-term predictions, the low frequency changes of the transects are important.

3. the representation should reduce the dimensions of the transects. As the transects now hold 33 data points, a reduction in data points should lead to computation advantages.

This thesis aims to make clusters of similar time series to improve forecasting shoreline in the long-term, shape-based clustering seems most convenient. With shape-based clustering, magnitudes of transects are better preserved and the chances of biased clustering with feature-based clustering is prevented. The raw time series or a time series' representation could be used to cluster based on shapes. Discrete Wavelet Transformation (DWT), transforms the time series to a lower dimension and can filter behavior based on frequency and amplitude. PyWavelets (Lee et al., 2019) is an open-source wavelet transform package in python. The PyWavelets DWT module provides a range of wavelets for the discrete wavelet transformation. By manually iterating different wavelets and decomposition levels, Daubechies 2 wavelet (see most left wavelet in figure 2.6) with a 2nd level decomposition results in the most optimal representation, using the approximation coefficients. The representation is still physically interpretable and the low frequency trends are captured. After a 2nd level decomposition, the approximation coefficients represent the frequencies in the lower quadrant of the raw time series' frequency band. So, the approximation coefficients of the second level capture frequencies in the range of $0-0.25\%$ of the original frequency band. This maximum of the original frequency band will be 0.5 ($\frac{1}{2Year}$), indicating that the lowest quadrant will capture frequencies in a range of 0-125. Where a frequency of 0.125 indicates a span of $\frac{1}{0.125} = 8 years$.
In addition, the DWT creates a significant dimension reduction from 33 data points (1984-2016) towards 10 with the half-band filter 2.7. The mean variance reduction for the time series is about 30%. The high-frequent trends are captured in detailed coefficients in the first and second level decomposition. These time series indicate the high frequent behavior per level decomposition. The detailed coefficients are not used for this research since the main interest is the low-frequent changes for long-term forecasting purposes. Since the DWT uses a half-band filter, the last step is to half all the

data points per final 2nd level decomposition to maintain the magnitude over the transects. The results of this decomposition for the transects of figure 2.4 are depicted in figure 3.9. As one can see in figure 3.9 the DWT representations follow the main trends of the raw data. Finally, another advantage of shape-based clustering is that local behavior is captured as well. The chance that adjacent transects show similar behavior is large and these adjacent transects will be clustered in the same clusters as well. This still makes it a global model, as it can be clustered with other global transects, but it also captures local information that could be useful.

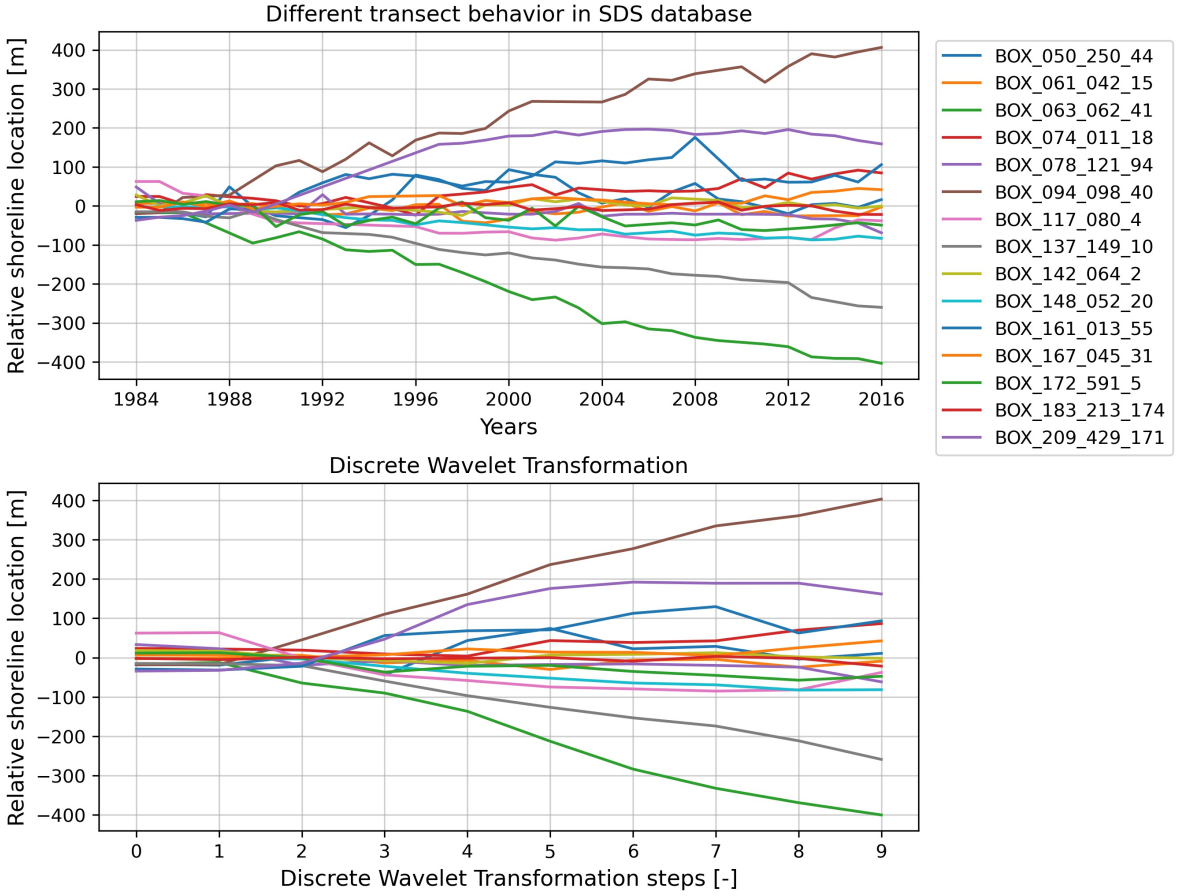

Figure 3.9: Discrete Wavelet Transformation on SDS transects with Daubechies 2 wavelet and second level decomposition leaves approximation coefficients with 10 data points (lower graph). The representation follows the low-frequent changes in the SDS transects, this is desirable as these trends are useful for long-term forecasting. The BOX_xxx_xxx_xxx labels in the legend are explained in Appendix A.1

Summary & implications next step(s)

The cluster input used in the next clustering step, step 4, will be the Discrete Wavelet Transformation (DWT) representation. This representation maintains the physical presentation of the transects and, therefore, is easily interpretable in hydraulic context. Furthermore, the DWT approximate coefficients purely capture low-frequent behavior, which is desirable for long-term forecasting. Last, the dimension reduction from 33 data points to 10 data points will lead to computational advantages.

3.5 Step 4. Transect clustering

The results of steps 1 and 2 directed step 3 to the most optimal input for this section. For the clustering, only time series features will be used, this will be in the form of a Discrete Wavelet Transformation (DWT) as described in Section 3.4. This step will present the final choices and setup regarding the clustering of the transects. In the results sections, the formed clusters with corresponding transects will be examined.

3.5.1 Whole time series clustering

Following the whole time series clustering approach sketched in 2.5 (Aghabozorgi et al., 2015), the next three parts of whole time series clustering that should be defined are: the distance measure, cluster prototype, and clustering algorithm. For the distance measure Dynamic Time Warping (DTW) is chosen as it is superior to Euclidean distance measures (Keogh & Pazzani, 2000) (Ann Ratanamahatana Eamonn Keogh, n.d.). The Euclidean distance measure has difficulties with distortions in the time axis, where DWT counters this.
The clustering algorithm considered in this thesis will be TimeseriesKmeans from the package *Tslearn* (Tavenard et al., 2020). Tslearn is a Python package that provides ML algorithms for analyzing time series. The main advantage of this packages is that it provides as distance measure DTW. Other cluster algorithms only provide DTW as a distance measure when a distance matrix is provided as input. Since the whole available dataset consists of about n = 350k transects, a distance matrix of n x n should be calculated, which would be computation-wise too complex. TimeSeriesKmeans accepts raw and modified time series as input for clustering. As the package name already exposes, the clustering prototype is based on the Kmeans principle where the mean of clusters is used as the prototype. Since the Kmeans is based on minimizing the variance in a cluster by a least-square estimation and not for arbitrary distances between time series in a cluster like with DTW, the DTW Barycenter Averaging (DBA) algorithm is designed (Petitjean et al., 2011). The barycenter refers to the mean of the cluster. The DBA algorithm minimizes by iterating the sum of squared DTW distances between the barycenter and the time series in a cluster, see figure 3.10. In the end, the barycenter of the clusters follows the shape of the time series in a cluster.

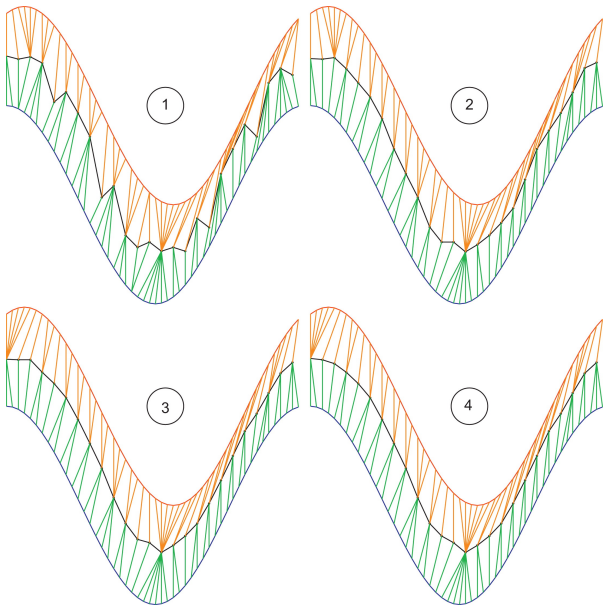

Figure 3.10: From step 1 to 4, the DBA algorithm iteratively finds the barycenter of two time series (Petitjean et al., 2011)

To sum up, following the whole time series clustering approach sketched in figure 2.5 in Chapter 2:

- **Time series representation:** Discrete Wavelet Transformation (DWT) with Daubechies 2 wavelet second level decomposition. Dimension reduced from 33 yearly data points to 10 data points, where the 10 data points present the lower quadrant of the original frequency band.
- **Distance measures:** Dynamic Time Warping (DTW)
- **Clustering prototype:** DTW Barycenter Averaging (DBA)
- **Cluster algorithm:** TimeSeriesKmeans

Next to the four steps described in figure 2.5, two additional steps are needed before clustering is possible. The optimal number of clusters has to be defined and a representative dataset has to be designed to train the clustering algorithm. The subset for training the clustering algorithm should be large enough to be a representative subset but not too large as it will be computationally expensive. This will be an iterative process by incrementally increasing the number of transects in the subset and considering the time it takes to cluster. After the subset is defined, the total number of clusters should be determined before the final clustering could be executed. There are numerous amount of algorithms to determine the optimal number of clusters. However, these algorithms focus primarily on point data instead of time series data and are computationally expensive. Popular algorithms for determining the optimal number of clusters like the Silhouette Score and Elbow method were not applicable to the SDS dataset. The number of clusters is eventually researched by reasoning and considering the purpose of clustering. For this research, the number of clusters is incrementally increased, starting at N=2 clusters. For every step, the barycenters of the clusters are plotted, by carefully looking at these barycenters, the optimal number of clusters N will be chosen. The main aims of clustering are to generate maximum dissimilarity between clusters and to create maximum similarity within clusters, by plotting the barycenters, this could be examined visually. Another aspect important for choosing the optimal number of clusters is to keep in mind the final forecasting purpose. This means that the barycenters should be reviewed with a hydraulic perspective, trying to understand the barycenters' shape in context of shoreline behavior.

3.5.2 Cluster results

In this subsection, the results of step 4 transect clustering are presented. The setup for the clustering is presented above and follows the four steps in figure 2.5. The results section can be divided into two parts. First part, the final hyperparameters had to be defined: the subset for training the clustering algorithm and the optimal number of clusters. The second part, the final clusters are presented and reviewed quantitatively and qualitatively. The results that will be presented in this subsection are results from a final iteration. TimeSeriesKmeans is highly dependent on the initialization of the cluster centers and therefore varies per iteration. As a consequence, the results are not binding. Nonetheless, the results presented here will help answer the research question and show the added value of the described methodology.
The final clusters and sub-clusters are already presented in figure 3.11. How these clusters are derived and what they indicate will become apparent in the remainder of this subsection. The figure can be helpful in understanding the different clusters can be used as a tool to see how different clusters relate to each other. In the remainder of this research, different clusters and sub-cluster will be named and to keep the overview, this figure can be helpful.

Training set and number of clusters

The subset for training the TimeSeriesKmeans clustering algorithm should represent the whole dataset and be small enough for computational purposes. Using a subset of 10k random transects,

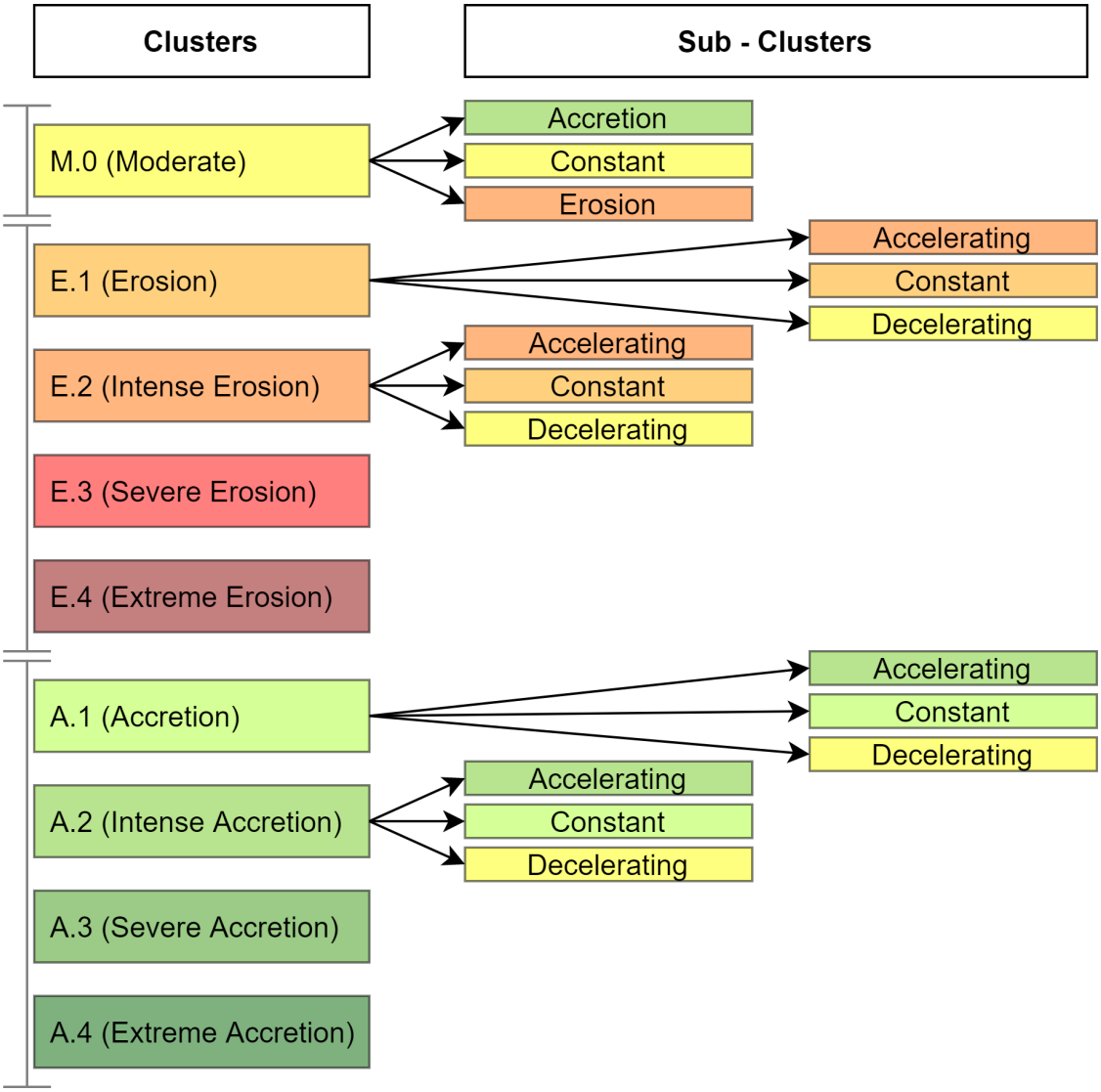

Figure 3.11: Final cluster overview of clusters formed in Step 4. Transect clustering. There are nine main clusters, with one moderate cluster, four erosive and four accretion clusters. The erosive and accretion clusters contain transects ranging from moderate behavior to more extreme behavior. The numbers indicate the type of behavior, where 1 indicates more moderate behavior and 4 extreme behavior. For clusters M.0, E.1, E.2, A.1 and A.2 three sub-clusters per cluster were derived. The workflow that led to these clusters and sub-clusters will be shown in this subsection. In addition, the analysis of the different clusters and sub-cluster will be discussed in this subsection. Furthermore, the added value of these clusters and sub-clusters concerning the research questions will be explained.

training the algorithms takes between 2-20 minutes, depending on the number of clusters used as input. The computational time of 20 minutes is considered allowable for quick iterations with different hyperparameters for the clustering algorithm. The transects in the subset are plotted in figure C.1 in Appendix C.1. The figure is showing that the transects represent all kinds of shoreline behavior.

The described subset can be used to determine the optimal number of clusters. The number of clusters is chosen by incrementally increasing the number of clusters. The start number of clusters was set at 2 clusters. For every iteration, the centers of clusters are plotted based on the DBA concept. The plots with barycenters for iterations with up to seven clusters and ten to twelve clusters are plotted in Appendix C.1. The plots with an input of eight, nine and ten clusters are plotted in figures 3.12, 3.13 and 3.14.

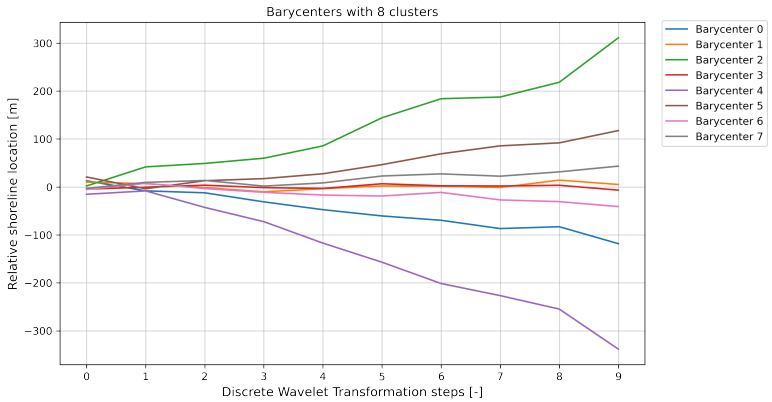

Figure 3.12: Barycenters for eight clusters, generated by clustering with TimeSeriesKmeans algorithm. Eight clusters result in a clear distinction of different transect behavior.

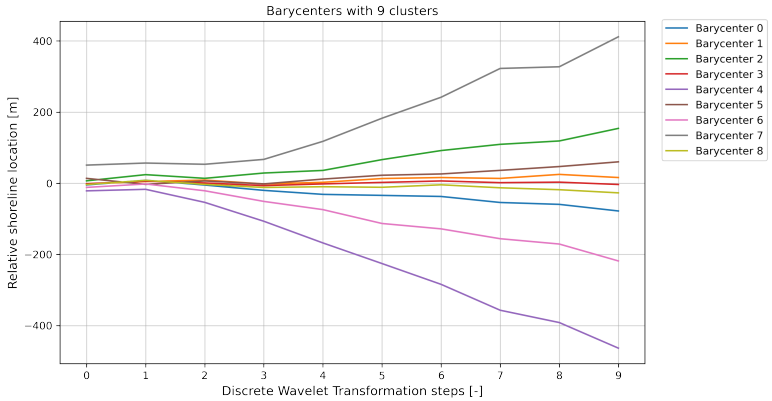

Figure 3.13: Barycenters for nine clusters, generated by clustering with TimeSeriesKmeans algorithm. Nine clusters result in a clear distinction of different transect behavior.

What strikes is that for the iterations with eight and nine clusters, the cluster barycenters are following clear trends and are not intersecting with others after DWT step 3. Considering the iterations with ten, eleven, and twelve clusters, some cluster barycenters are not following one single trend. The transects in these clusters are showing oscillating behavior, first erosive and later accretion or vice versa. Next to that, some clusters in figure 3.14 are intersecting with others. Considering that and recalling one of the purposes of clustering: creating maximum dissimilarity between clusters, this crossing of cluster centers is undesirable. Furthermore, the barycenters for the figures 3.12 and 3.13 are showing clear and dissimilar constant trends. Another purpose of clustering is to maximize similarity within clusters. Therefore nine clusters is most preferable over eight clusters as it adds an extra cluster in the more moderate region. This is a desirable feature as it creates an understanding of the type of behavior the clusters represent (e.g., erosive or accretion). Also, the change rate is the most important long-term shoreline feature. The forecasting algorithms could benefit from it by creating clusters with different change rates since the data is more uniform.

To conclude, using more than 10 clusters is not optimal as an input because barycenters are intersecting, and understanding the shoreline behavior it presents, is more challenging. Nine clusters are the best number of clusters since it is the maximum number of clusters before cluster barycenters start to intersect. With nine clusters, there is maximum dissimilarity between clusters, a widespread of understandable shoreline behavior is captured and the barycenters represent the most important feature for shoreline forecasting, namely trend.

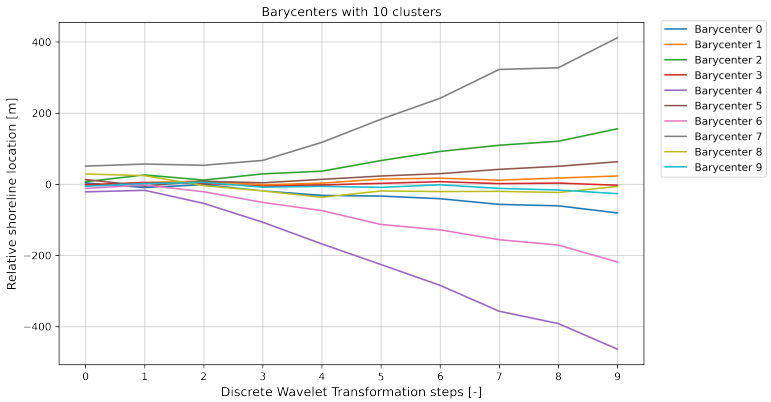

Figure 3.14: Barycenters for ten clusters, generated by clustering with TimeSeriesKmeans algorithm. Ten clusters results in a intersection of different clusters centers, this is not preferable since an aim is to create maximum dissimilarity between cluster centers.

Clusters

After the subset is defined together with the optimal amount of nine clusters, clustering starts. First, the subset is used to train the clustering algorithm, this was done by setting the max number of iterations to 100 and using k-means++ as initialization technique (Arthur & Vassilvitskii, 2007). After the algorithm is trained, the rest of the transects are assigned to clusters, so all transects were assigned to a cluster. A disadvantage of the TimeSeriesKmeans, is that the final results are dependent on the initialization and the final clusters are different per run. These disadvantages will be further elaborated on in Chapter 4. For this section and in the remainder of the report, a representative run is used to describe the results of the methods.
The final barycenters are plotted in figure 3.16 with the corresponding number of transects per cluster. The transects per cluster are all plotted in the figures C.10 till C.27 in Appendix C.1. The barycenters of the final clusters follow clear trends and can be named after their shapes, see table 3.8. For example, in figure 3.15 all the transects in cluster E.2 are presented. Here it becomes clear that the main behavior of all transects in cluster E.2 is intense erosive behavior.

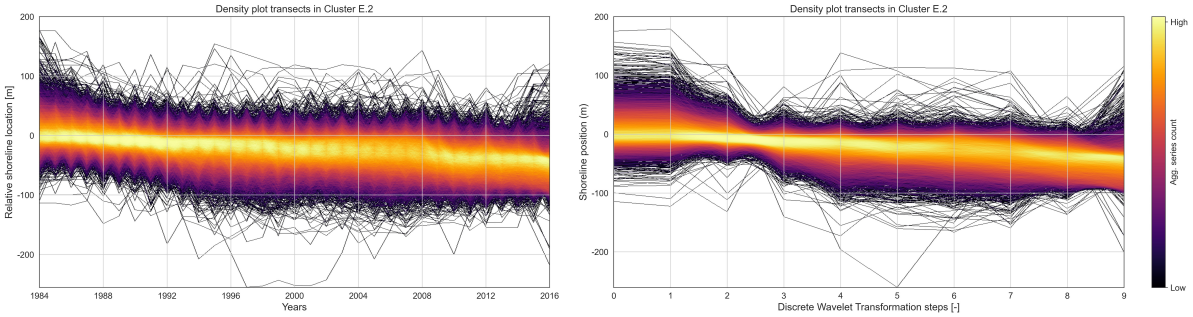

Figure 3.15: 24,067 transects in cluster E.2 are showing intense erosive behavior. Left, all the transects with DWT representation and right all the raw transects in cluster E.2.

The clusters are named following the same principle in sub-question 1 (Luijendijk et al., 2018). The cluster names are following the behavior of the transects in each cluster. The M, E and A indicate whether the cluster shows moderate, erosion or accretion behavior. The number following expresses the magnitude of the behavior. Ranging from 1 till 4, where clusters E.1 and A.1 show moderate behavior and clusters E.4 and A.4 show more extreme behavior. Cluster M.0 is indicated with a zero since there is only one moderate cluster.

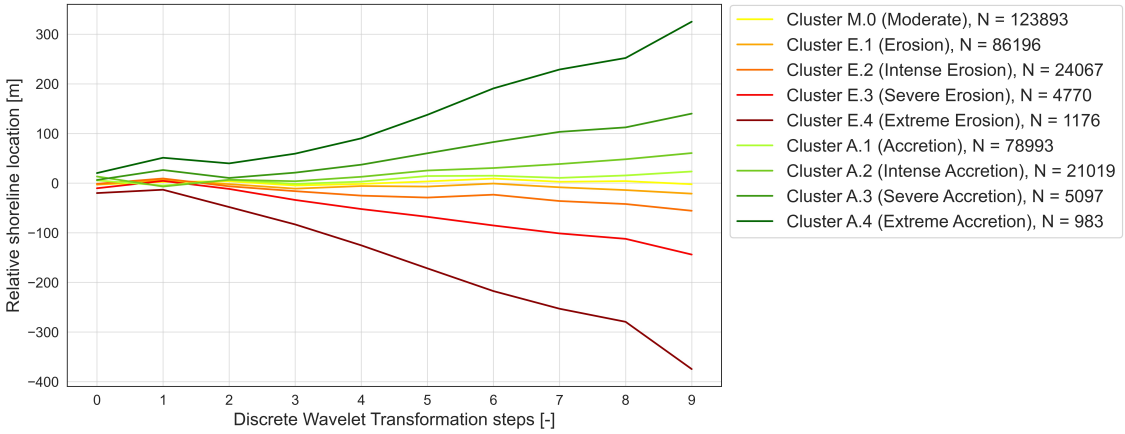

Figure 3.16: Final barycenters of the nine clusters. In the legend the total transects per cluster are stated together with the overall behavior of transects in the cluster.

Table 3.8: Final clusters with behavior and amount of transects per cluster. Last column indicates the fraction of transects of the total number of clusters, 346,194. In the remainder of the report, there will be referred to the clusters by the cluster names.

Cluster names	**Behavior**	**N**	**Percentage**
M.0	Moderate	123,893	35.8%
E.1	Erosion	86,196	24.9%
E.2	Intense Erosion	24,087	7.0%
E.3	Severe Erosion	4,770	1.4%
E.4	Extreme Erosion	1,176	0.3%
A.1	Accretion	78,993	22.8%
A.2	Intense Accretion	21,019	6.1%
A.3	Severe Accretion	5,097	1.5%
A.4	Extreme Accretion	983	0.3%

Next to the statistical analysis, the cluster approach makes a spatial distribution possible. In figure 3.17 the distribution of clusters on a planetary scale are described. This distribution figure is helpful for detecting hotspots of clusters on a global scale and adds extra understanding to global shoreline behavior. For example, considering the horizontal density plot, extreme erosion hotspots are detected for the south coast of the USA and near the coast of Bangladesh. The west coast of South Africa and the Amazon River delta area are standing out with erosive behavior on the vertical density plot.

In the coming paragraphs, clusters M.0, E.2, and A.1 (moderate, intense erosion and accretion, respectively) are discussed in more detail. These clusters contain a large amount of the total dataset. Cluster E.2 is the exact opposite of cluster A.2 and cluster A.1, the opposite of cluster E.1. So, the analysis approach of these clusters is relevant for a large set of other transects. The other clusters are presented as well in table 3.8 and furthermore in Appendix C.1. Following the results of clusters M.0, E.2, and A.1, a sub-cluster analysis helps create a better in-depth understanding of different global transect behavior.

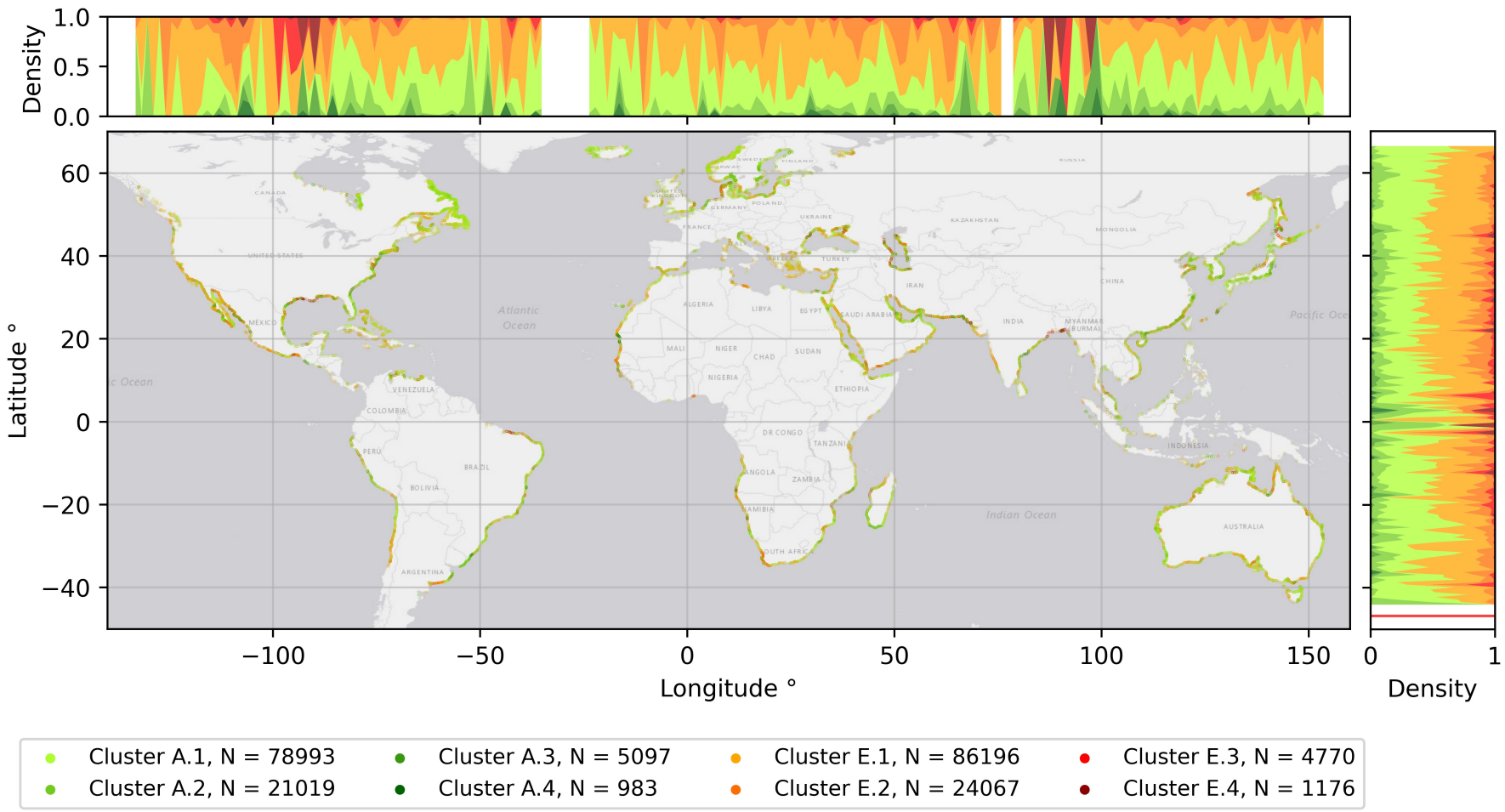

Figure 3.17: Distribution of sandy transects in clusters on planetary scale. At the axes, a kernel density plot is added to show distribution of the eight clusters per longitude and latitude. Cluster M.0 is not scattered in this plot to better highlight the other clusters.

Cluster M.0 Moderate

The first cluster is cluster M.0 (moderate behavior), the cluster holding the most transects, with 123,893 transects. Looking at the barycenter in figure 3.16, the barycenter presents moderate behavior. That cluster M.0 is holding the most transects agrees with the general knowledge that most beaches and shorelines on earth show moderate behavior (Luijendijk et al., 2018). In figure C.10 and C.11 all the transects in cluster M.0 are plotted, for both raw data as DWT representation. As one can see, all the transect positions lie between -100 m and 100 m. The transects in this cluster are moderate and are therefore not a priority for forecasting, this moderate regime is not threatening coastal zones. The key result here is to get confirmation that most shorelines are in the moderate regime. To quantify the visual findings, the four features of step 2 Section 3.3 are described in table 3.9. Where the change rate is the mean trend over 33 years, gathered by linear regression over the time series. The mean change rate is close to 0, indicating stable and moderate behavior. Next to the change rate, the absolute sum of steps over 33 years is almost 260 m. Dividing 260 with 33, results in a mean step size of 8 m between years. The mean step size lies within the sub-pixel (15 m) accuracy of the SDS data and highlights again the moderate behavior of all transects in this cluster. The mean step of 8 m also has the same magnitude as the absolute sum of differences, 9 m. In addition, the ACF value is close to zero, this implies that the location at t is hardly correlated by the location at $t-1$. This low value could indicate that these transects are in equilibrium and as a consequence, the spatial analysis for the cluster with moderate behavior is not elaborated on in the remainder of this study.

Table 3.9: Final clusters with behavior described by four characteristics, defined in step 2, Section 3.3. Clusters M.0, E.2, and A.1 (moderate, intense erosion and accretion respectively) are shaded since these will be discussed in Section 3.5.2.

	Behavior	Absolute sum of differences [m]	Sum of absolute differences[m]	change rate [m/year]	ACF 1
Cluster M.0	Moderate	9.22	257.1	0.071	0.22
Cluster E.1	Erosion	21.40	311.7	-0.476	0.38
Cluster E.2	Intense Erosion	55.15	375.2	-1.545	0.64
Cluster E.3	Severe Erosion	130.85	410.3	-4.263	0.84
Cluster E.4	Extreme Erosion	332.43	516.28	-11.033	0.90
Cluster A.1	Accretion	20.13	330.59	0.614	0.42
Cluster A.2	Intense Accretion	47.98	404.42	1.642	0.66
Cluster A.3	Severe Accretion	117.01	444.79	4.057	0.83
Cluster A.4	Extreme Accretion	273.01	542.46	9.386	0.90

Cluster E.2 Intense Erosion

The barycenter of cluster E.2 in figure 3.16, shows erosive behavior, more precisely, intense erosion. The cluster is holding over 24k transects and resembles cluster A.2, which is holding the same magnitude of transects but shows the exact opposite behavior, intense accretion. The density plots of transects in cluster E.2 are presented in figures 3.15. The mean features are stated in table 3.9. The change rate confirms the form of the barycenter and the transects in figure 3.15. Next to that, it stands out that the ACF almost tripled compared to the ACF value of cluster M.0. This tripling indicates that the data point at t is correlated with the data point at time $t-1$. Furthermore, the sum of all absolute differences is around 375 m, indicating a mean step size of 11 m per year. The magnitude of changes in this cluster is larger than compared to cluster M.0. Next to temporal behavior, spatial analysis on the transects in this cluster is made. In figure 3.18 all the transects intersects with the OSM shoreline are scattered on a global map. The markers are transparent, so areas with a high density of transects of a cluster stand out. Also, the histograms on the axes quantify high dense areas. Here it strikes that the west coast of Africa and the west coast of Cape Town have a relatively high density of transects in cluster E.2, indicating intense erosion (Dube et al., 2021). For all other clusters, a map with the distribution of its transects is displayed in Appendix C.1.

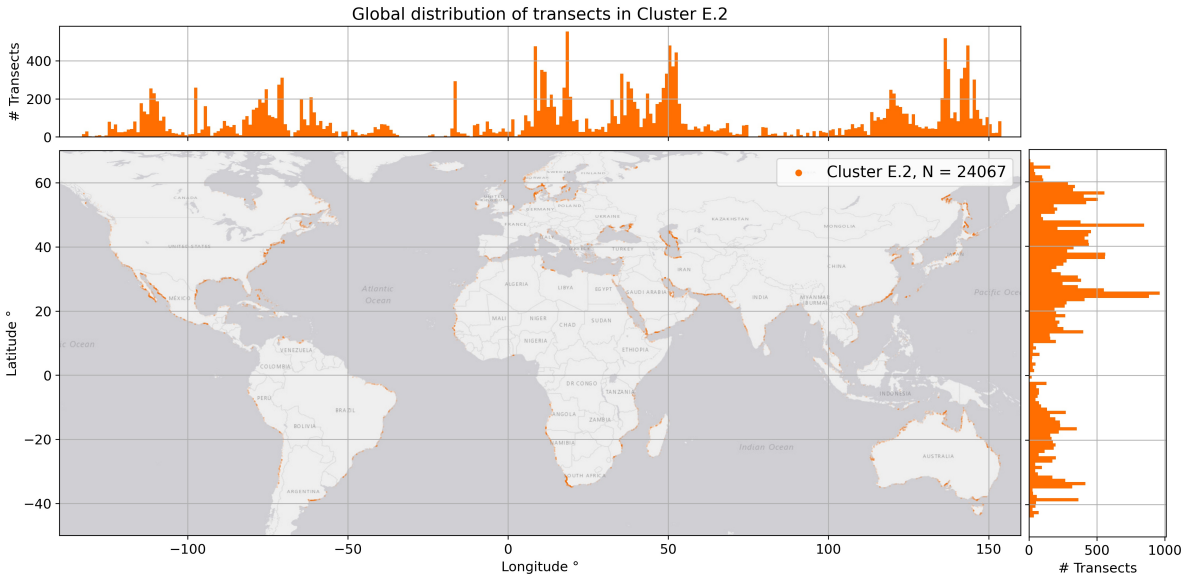

Figure 3.18: Global distribution of sandy transects in cluster E.2 Intense Erosion

Finally, the transects in cluster E.2 are compared to the transects of other clusters. By comparing clusters to each other, the clustering approach and final clusters could be understood as optimal. The most similar clusters to cluster E.2 are cluster E.1, cluster E.3, and cluster M.0. Cluster E.1 showing erosive behavior and cluster E.3 showing severe erosive behavior. Looking at the barycenters in figure 3.16, it seems that the clustering algorithm's critical distinction feature is the change rate of the clusters. To investigate this hypothesis, a histogram is made for all the transect per cluster based on their change rate, see figure 3.19. Two major findings presented in figure 3.19 are important to

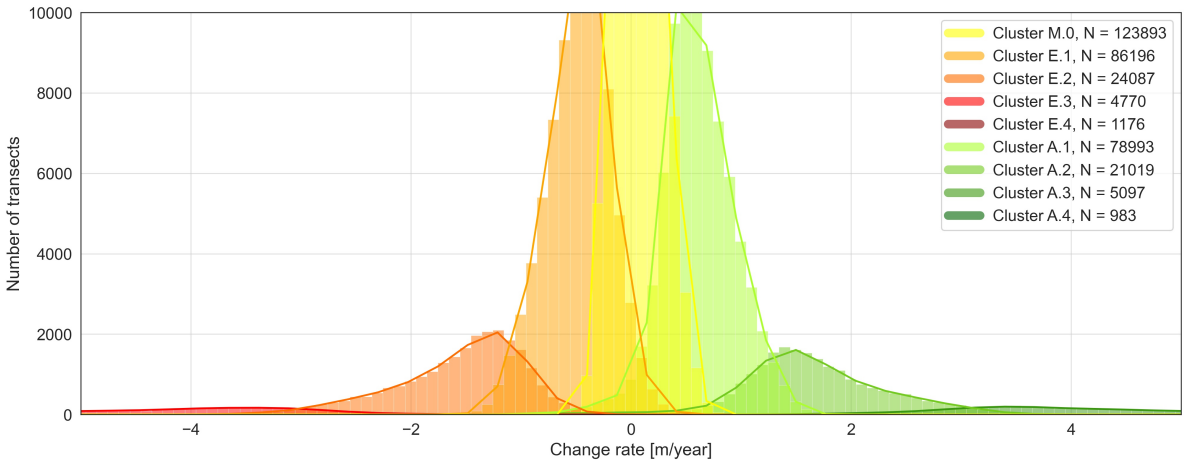

Figure 3.19: Histogram based on change rate per cluster. The y and x limits are adjusted so that overlapping of different histograms is visible. Cluster E.4 Extreme Erosion and A.4 Extreme Accretion are therefore not visible.

state. First, most of the transects (middle of the histogram), have a change rate that corresponds with the barycenters in figure 3.16. Another thing that immediately strikes in figure 3.16 is the overlapping of histograms at the tails. This overlapping indicates that the clustering approach makes a distinction between transects, even though they have the same change rate. This distinction is already a key-finding, as the clustering already adds some extra nuance to simple transect classification as described in step 1. For better understanding the overlapping of the histogram tails, transects of different clusters in the overlapping tails are plotted.

In figure 3.20 a sample of 100 transects per cluster is plotted. All the transects in the figure are transects that have a change rate between -1.1 and -1.0 m/year. These transects correspond to the overlapping of the histograms of clusters E.1 and E.2 in figure 3.19.

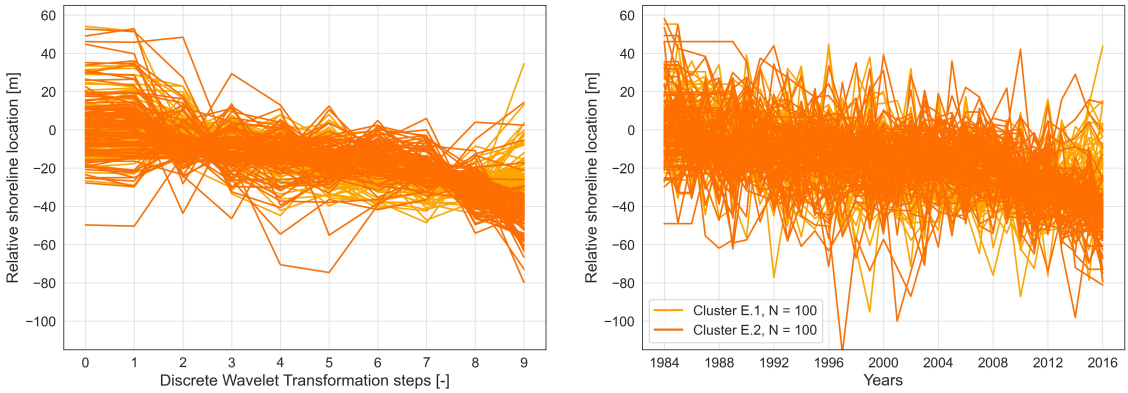

Figure 3.20: 100 Random transects of clusters E.1 and E.2 with same change rate between -1.1 and -1.0 m/year. Discrete wavelet transformation representation left and right raw time series.

The transects in both graphs of figure 3.20 follow the same overall trend as indicated by the mean change rate they share. However, looking at the DWT representation of the transects, subtle differences can be detected, the transects of cluster E.1 are following a different shape than the transects of cluster E.2. The transects of cluster E.1 are showing more erosive behavior at the beginning and more moderate, or even accretion behavior, in the last years. The transects in cluster E.2 are showing the opposite, with more moderate behavior in the beginning and more erosive behavior in the last DWT steps. In even more detail, the main difference between the two sets of transects lies in the last steps. The clustering approach adds the most distinction for the last DWT steps of the transects, which is highly beneficial for forecasting purposes. The clustering approach of this thesis makes a distinction in shapes of time series, and thus changes on sub-time series level can be detected. For forecasting, the latest trend in time series is most important as it describes the current behavior of transects.

Next to the distinction made by the clustering algorithm, the right graph in figure 3.20 of raw time series underlines the reason for using the DWT representation. Comparing the two graphs in 3.20, via the left graph long-term trend differences can be detected and within the right graph, it is much harder to see differences in long-term behavior. The DWT representation perfectly filters out long-term trends and reduces dimensions, so long-term trends can be used for clustering.

Cluster A.1 Accretion

Next to cluster M.0 and cluster E.2, cluster A.1 Accretion is described in this subsection. The barycenter of cluster A.1, shows accretion behavior. The cluster is holding over 78k transects and resembles cluster E.1, which is holding the same magnitude of transects and shows the exact opposite behavior, erosion. The transects in cluster A.1 are plotted in a density plot in figures C.20 and C.21. The density plot shows the same behavior as the barycenter in figure 3.16. Besides, the mean features are stated in table 3.9. Here the change rate is as expected, showing moderate accretion. The ACF value here is lower than for the transects in cluster E.2 but higher for the transects in cluster M.0. This lower ACF can indicate that for more extreme behavior, the absolute ACF value increases.

Considering the global distribution of the transects in cluster A.1 in figure C.32, the distribution of the transects becomes clear. Here most information can be extracted by comparing all distribution plots for a given location. For example, the south of Madagascar shows a high density in cluster A.1 transects especially compared to other clusters. Another example on a global scale is presented in figure 3.21, here all the transects are scattered in the Baltic Sea region. The first thing that strikes when looking at the legend is that the largest cluster represented is cluster A.1 (accretion). Second, the distribution of the clusters over the latitude adds extra understanding. By considering the density plot on the y-axis, it appears that the transects in cluster A.1 are together with cluster M.0 dominant in the north. The largest erosive clusters, cluster E.1 and E.2 are less represented in the north. However, in the southern part of the Baltic Sea the erosive transects take over. This cluster distribution over the latitude could be clarified because the southern part of the Baltic Sea coastal areas are experiencing subsidence and the northern areas are rising (Weisse et al., 2021). Furthermore, the large variability in the longitude density plot corresponds with the reported spatial and temporal variation in processes. Finally, like cluster E.2, the transects in cluster A.1 are compared to other transects in different clusters. Considering the graph in figure 3.19, overlapping of the lower tail of cluster A.1 is detected with the middle of cluster M.0 and the upper tail of cluster E.1 Erosion. In figure 3.22 a sample of 100 random transects per cluster is plotted. All the transects in the figure are transects that have a change rate between -0.1 and 0.1 m/year. These transects correspond to the overlapping of the histograms of clusters M.0, E.1, and A.1 in figure 3.19.
The transects in the left graph of figure 3.22 follow the same overall trend as indicated by the mean change rate they share. Looking again at the DWT representation of the transects, the same subtle differences can be detected. The transects per cluster are following different shapes. The transects of cluster A.1 show more erosive behavior in the beginning years and more accretion behavior in the last years. The transects in cluster E.1 show the opposite, with more moderate behavior in the beginning and more erosive behavior in the last DWT steps. The transects in cluster M.0 are showing moderate behavior with no significant trend changes in time. Also, in this graph, the significant difference per cluster lies in the last steps. Cluster A.1 shows clear accretion behavior, cluster E.1 erosive, and cluster M.0 constant. Using the clustering approach as described, a distinction can be made in trend changes in a transect. Again, looking at the left and right graph in figure 3.22, the added value of DWT representation gets underlined. In the right graph, the long-term trends visible in the left graph are hard to detect.

3.5.3 Sub-cluster results

In this subsection, the derivation of the sub-clusters for clusters M.0, E.1, E.2, A.1 and A.2 in figure 3.11 are explained. Figure 3.19 can be leading for understanding how transects behave. Considering the explanation above and the graphs in figures 3.20 and 3.22, a noteworthy conclusion can be drawn. The location of a transect in the histograms of figure 3.19 says something about the behavior it showed in the last years; the transects in the tails of the histograms all show the same behavior. All transects in the upper tail of a cluster histogram in figure 3.19, show behavior tending towards erosion. In the erosive clusters, the transects in the upper tail show accelerating erosive behavior. For the accretion clusters, the transects in the upper tail show a decelerating behavior, tending more to moderate/erosive behavior. Reviewing the lower tail of the cluster, the latest trend is vice-versa. For the erosive clusters, the transects in the lower tail show a decelerating erosive behavior towards a more accretion/moderate trend in the last years. For the lower tails of the histograms in the accretion clusters, an accelerating behavior is found, more extreme accretion. With this result, a better understanding of transect behavior can be created.

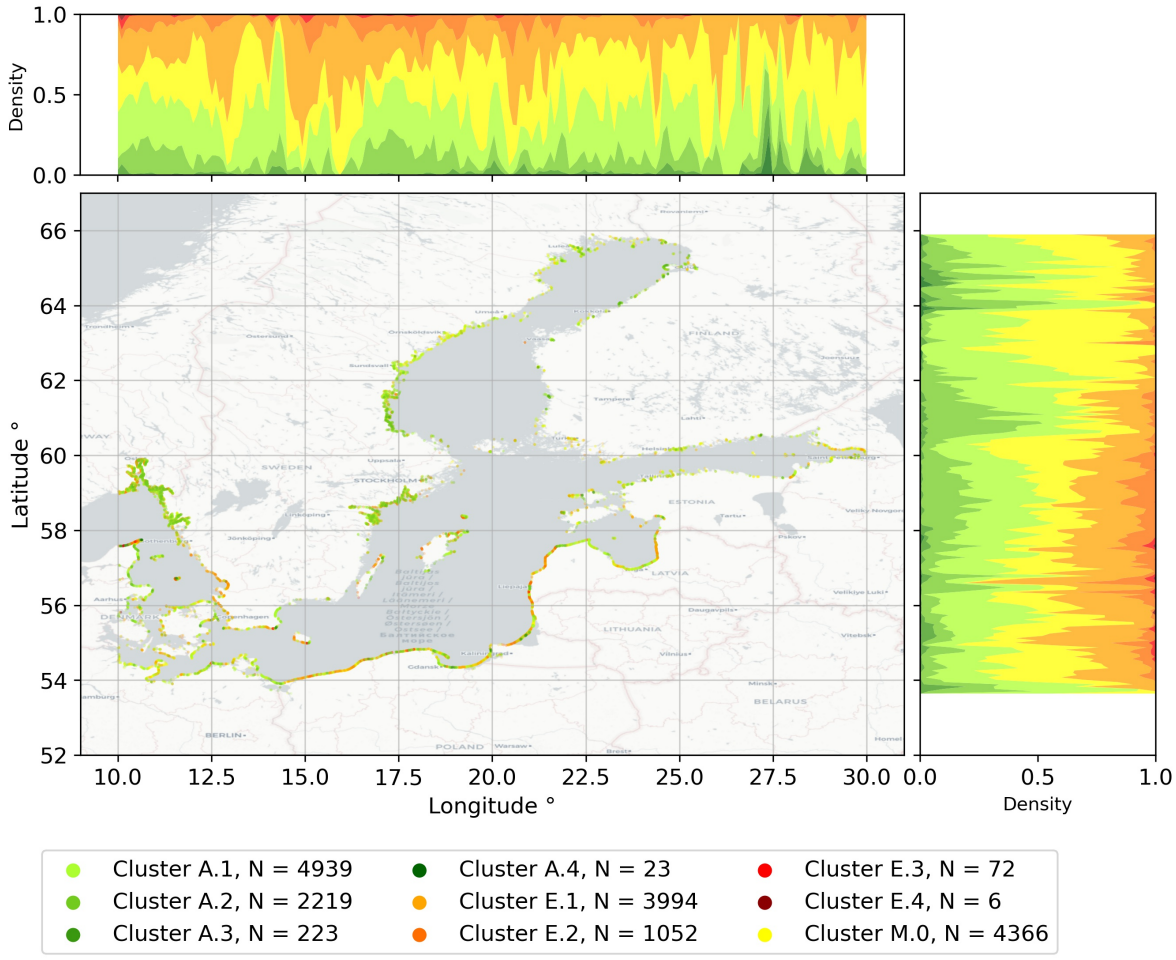

Figure 3.21: Regional distribution of sandy transects in Baltic sea region. In the northern part of the Baltic sea, the accretion clusters are dominant, but in the southern part the erosive transects are dominating. This distribution can be clarified by subsiding behavior in the southern part and rising land in the north (Weisse et al., 2021).

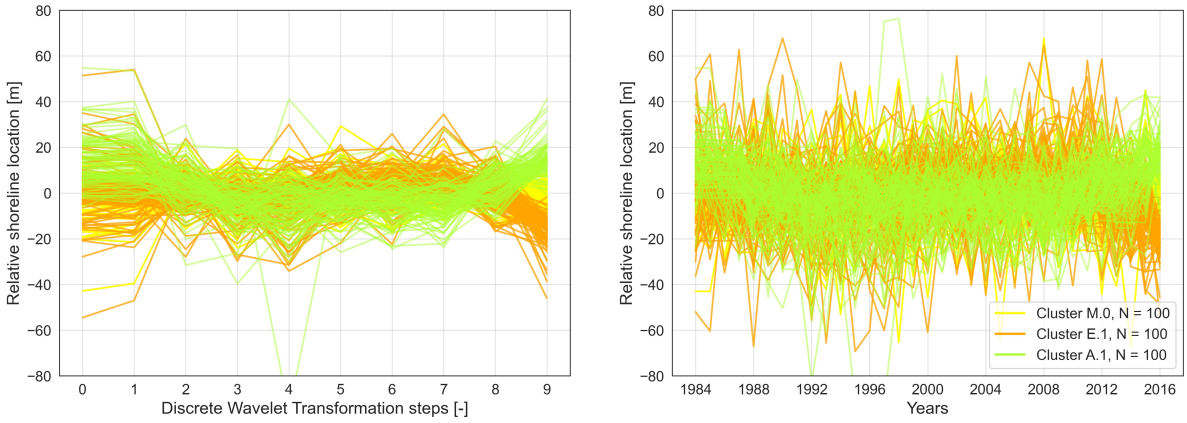

Figure 3.22: 100 Random transects of clusters M.0, E.1 and A.1 with same change rate between -0.1 and 0.1 m/year. Discrete wavelet transformation representation left and right raw time series.

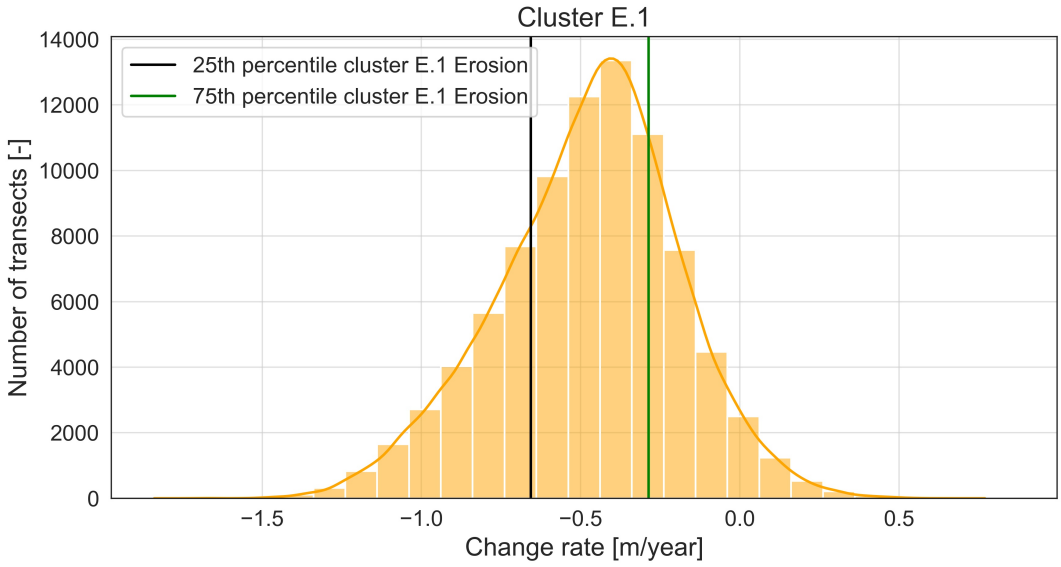

Figure 3.23: Histogram of cluster E.1, based on change rate. The 25th and 75th percentile are presented with vertical lines. These lines indicate the start and end of the lower- and upper tail. All transects left of the 25th percentile line are in the lower tail, these are transects that showed more moderate and accretive behavior in the last years. In the upper tail, all the transects right of the 75th percentile, the transects showed accelerating erosive behavior.

Following the result described above, sub-clusters can be formed. A challenge here is to find the boundary between tails of the histograms in figure 3.19. Different percentiles were used and overall transect behavior in a sub-cluster was reviewed. It finally resulted in a definition of the lower tail ranging from minimum value to the 25th percentile. For the upper tail, the 75th percentile till the max value was used. The transects between the 25th and 75th percentile are considered constant, and showing mostly constant behavior, according to the barycenter in figure 3.16. The 25th and 75th percentile of the histogram for cluster E.1 are presented in figure 3.23. In figures C.36 till C.40 the sub-clusters of clusters M.0, E.1, E.2, A.1 and A.2 are presented respectively.
Next to the sub-clusters of cluster E.1, a more systematic approach was used for the rest of the clusters. For all clusters, except the severe and extreme clusters (cluster E.3, E.4, A.3, and A.4), a meaningful sub-cluster was formed with the same approach as described for cluster E.1. The severe and extreme clusters are not separated into sub-clusters, as these clusters contain too few transects and there is not a clear difference in behavior in the clusters. The sub-clustering method leads to three sub-clusters for all the other clusters, where all the sub-clusters follow the same procedure and name. For example, cluster E.2 is separated into three sub-clusters:

- Cluster E.2 - Accelerating
- Cluster E.2 - Constant
- Cluster E.2 - Decelerating

The last term per sub-cluster indicates the trend of the last years of the transects. For example, the transects in sub-cluster E.2 - Decelerating are transects that showed intense erosion but tend towards a more moderate and/or accretion behavior in the last years. In figure 3.24 the final sub-clusters of cluster E.2 are stated, here again, the shape of the transects and the mean correspond with the name of the sub-cluster it is in.
Only for cluster M.0, the sub-clusters are named differently. Since there is not a clear overall trend, it cannot decelerate or accelerate. Here the sub-clusters are named: Erosion, Constant and Accretion. For example, transects in sub-cluster Erosion show moderate behavior but tend more towards erosive behavior in the last years.

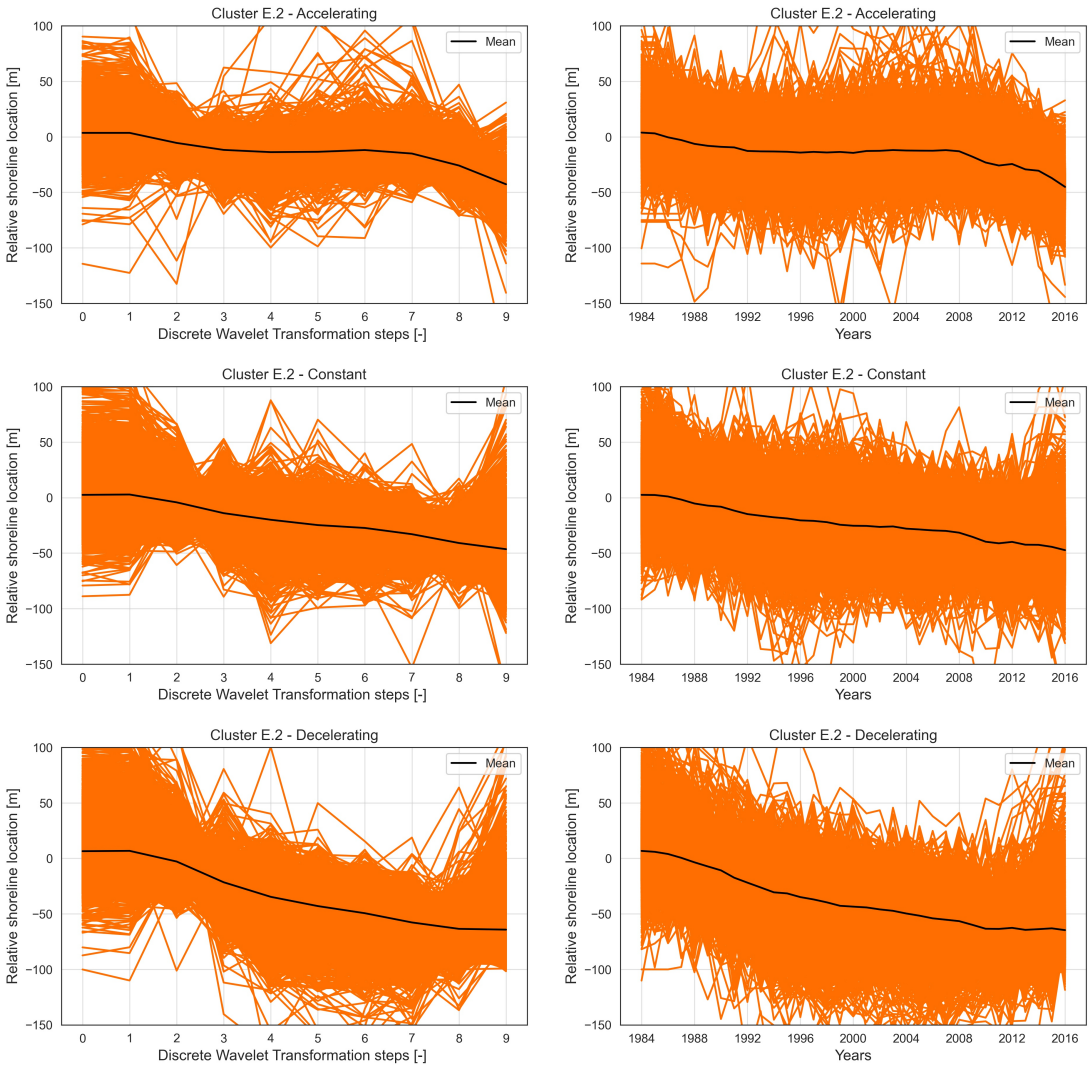

Figure 3.24: Plots of all transects in the three different sub-clusters of cluster E.2. Left, the plots of the DWT representation and right the raw transect time series. The first row plots are the transects in the upper tail of the histogram, the middle row the transects in the center and the lowest row all the transects in the lower tail.

Sub-clusters geographical

The sub-clusters defined above can be used to understand the global distribution of different transects' behavior. With four case studies, the added value of this clustering refinement is presented. In Appendix C.1, global scatter plots with distributions of sub-clusters per main cluster are presented in figures C.41 till C.45. The first case study is on a global level, considering the Caspian sea. In figure 3.25 the sub-clusters of cluster E.1 Erosion and cluster E.2 Intense Erosion are scattered for the Caspian Sea. The Caspian Sea is known for its substantial sea-level fluctuations over the past hundred years. With an observed sea-level increase of +12,74 cm/year and decrease -6,72 cm/year for the periods 1979-1995 and 1996-2015 respectively (Chen et al., 2017). These observed sea-level changes could be found in the sub-clusters of cluster E.1 and cluster E.2. Considering the legend in figure 3.25, most transects are in the decelerating sub-clusters. This distribution corresponds with observed sea level changes, where an increase in sea level caused erosion rates and vice versa, where accretion rates follow a decrease in sea level. By plotting the transects in the decelerating sub-clusters, the described behavior above is visualized, see figure 3.26. Furthermore, the map in figure 3.25 helps to detect "hotspots" on sub-cluster level.

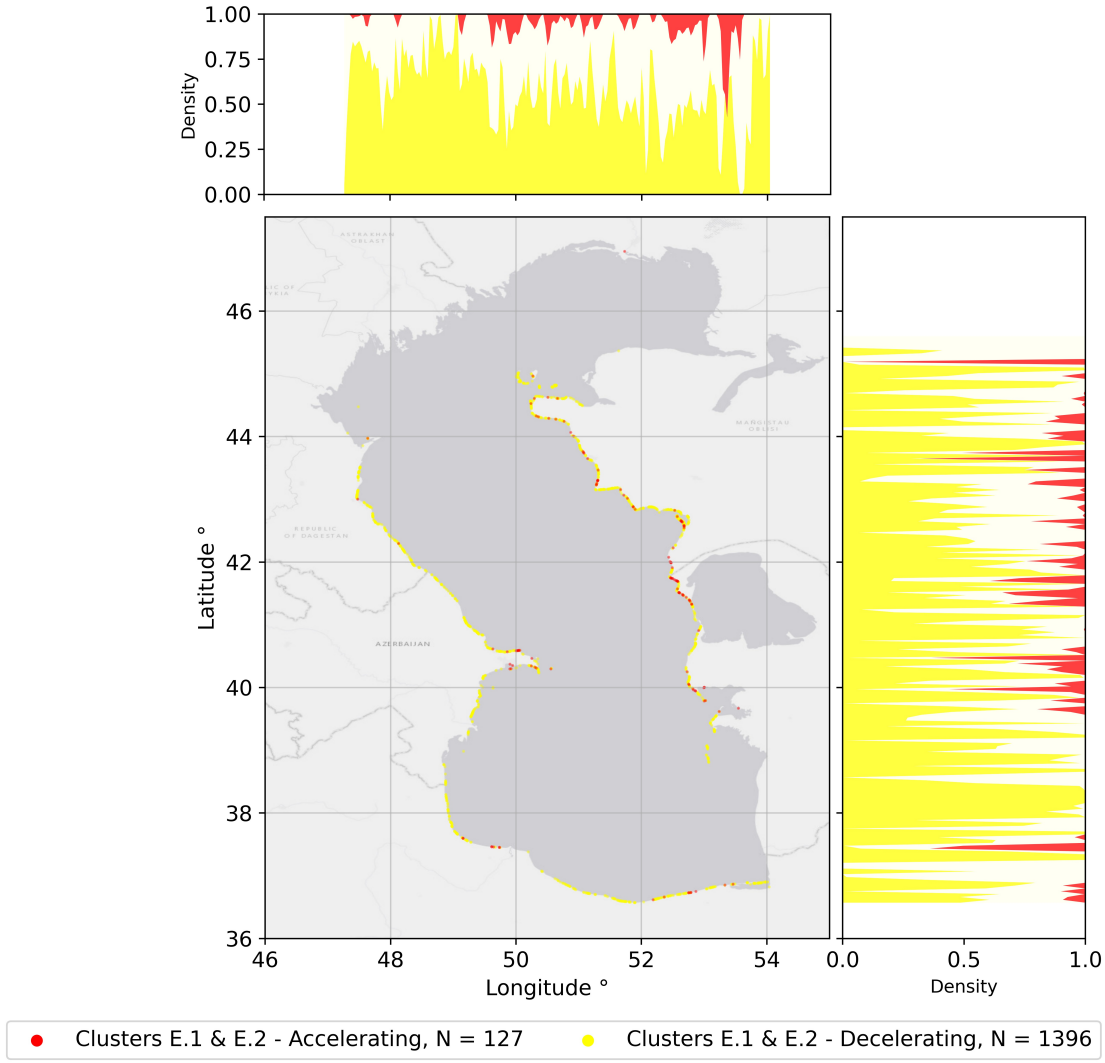

Figure 3.25: Global distribution of sub-clusters in Caspian Sea. The transects in sub-clusters Accelerating and Decelerating of cluster E.1 and cluster E.2 are scattered. The sub-clusters are combined for visualization purposes. The N = 985 transects in the Constant sub-clusters are not scattered for visualization reasons but can be found back in the density plots by the white color. The dominant decelerating trend, indicated by the yellow color in the density plots and the number of transects in the legend, corresponds with current literature (Chen et al., 2017)

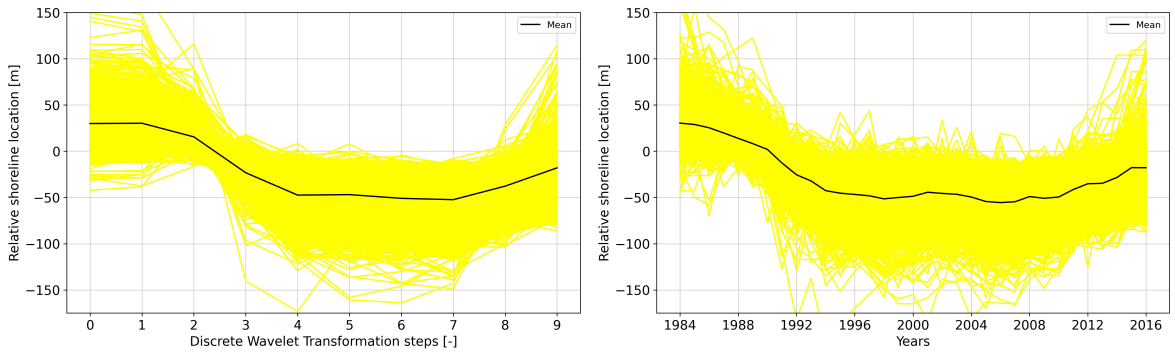

Figure 3.26: Transects in cluster E.1 and E.2 and sub-clusters Decelerating in the Caspian Sea, presented in figure 3.25.

As a second case study, the coast of Chile is reviewed. Some beaches in central Chile are experiencing an increase in erosion rates due to SLR and more frequent storms (C. Martínez et al., 2018). This finding could be enforced by the sub-clusters found in previous sections. In figure 3.27 the sub-clusters of clusters E.1 and cluster E.2 are scattered along the coast of Chile. Looking at the scattered points, it is hard to conclude something. However, looking at the legend gives insights into the behavior of transects on the coast of Chile. For clusters E.1 and E.2, most of the transects fall in the accelerating sub-cluster and this corresponds to the findings in the literature that erosion rates in central Chile are increasing (C. Martínez et al., 2018). To illustrate it more visually, all the transects in sub-cluster Accelerating of cluster E.2 are plotted in figure 3.28. This example shows that the clustering refinement could lead to a better understanding on a regional level.

Next to the regional level of the Chile coast case study, the refinement could also lead to more understanding on a local level. Considering the beaches in front of the Los Angeles International Airport, (intense) erosion and accretion can be detected following the Shoreline Monitor. Using the sub-clusters, an extra refinement in understanding could be created by determining whether the trends are accelerating or decelerating. In figure 3.29 the transects are scattered on the coast of Los Angeles. With this figure, trends on a local scale can be detected. Following the long stroke of transects with the color purple on the Playa Del Rey Beach, they are in cluster E.1 and in the sub-cluster Constant. This could be a warning that over the last 33 years, the beach is eroding with relatively constant rates. These erosion rates are due to the construction of Marina Del Rey, where jetties have interrupted longshore transport since the '60s. The constant erosion rates of the transects on the Playa Del Rey Beach are threatening the hinterland, especially the Los Angeles Airport. The considered transects on the Playa Del Rey Beach are also plotted in time in figures 3.30. Here can be seen that for both DWT as raw time series, the transects are indeed eroding at a relatively constant level.

The last example is a shoreline in the northern part of the Japanese main island Honshu. In the Akita Perfecture, lies the Kamayahama Beach in front of Mitane. Zooming in results in the map presented in figure 3.31. Here it strikes that the largest groups here are the sub-cluster Erosion of cluster M.0 and sub-cluster Decelerating of cluster A.1. This distribution indicates that this beach is possibly affected by a reversing trend tending more towards erosive behavior in the last years. In figures 3.32 and 3.33, the transects show indeed behavior tending more towards erosive behavior. In figure 3.32, the erosive trend in the last years is very clear and indicates a change in trend. For figure 3.33 the accretion trend is indeed slowly decelerating. The transects in figures 3.32 and 3.33 could trigger further research on what causes this reverse in trend and how this might work out in the future.

Summary & implications next step(s)

The clustering workflow resulted in nine main clusters and for each of the five largest clusters, three sub-clusters, see figure 3.11 for an overview. The sub-clusters and clusters presented are used for improving understanding of transect behavior on a global and local scale, as shown in the examples in this subsection. In short, with the nine main clusters and corresponding sub-clusters, new quantitative insights into the state of the shoreline evolution can be found, literature can be coupled with the cluster information and new trends in shoreline behavior can be detected that may threaten coastal zones. Furthermore, the sub-cluster information could help select the most reliable and optimal predictions, this will be further highlighted in step 5. In step 5, the transects of the original nine clusters (M.0, E.1, E.2, etc.) are used as input for the forecasting algorithms. The sub-clusters are not used as input, this is because the sub-clusters are derived on a subjective basis and using the sub-clusters as input could probably lead to overfitting of the forecasting algorithm.

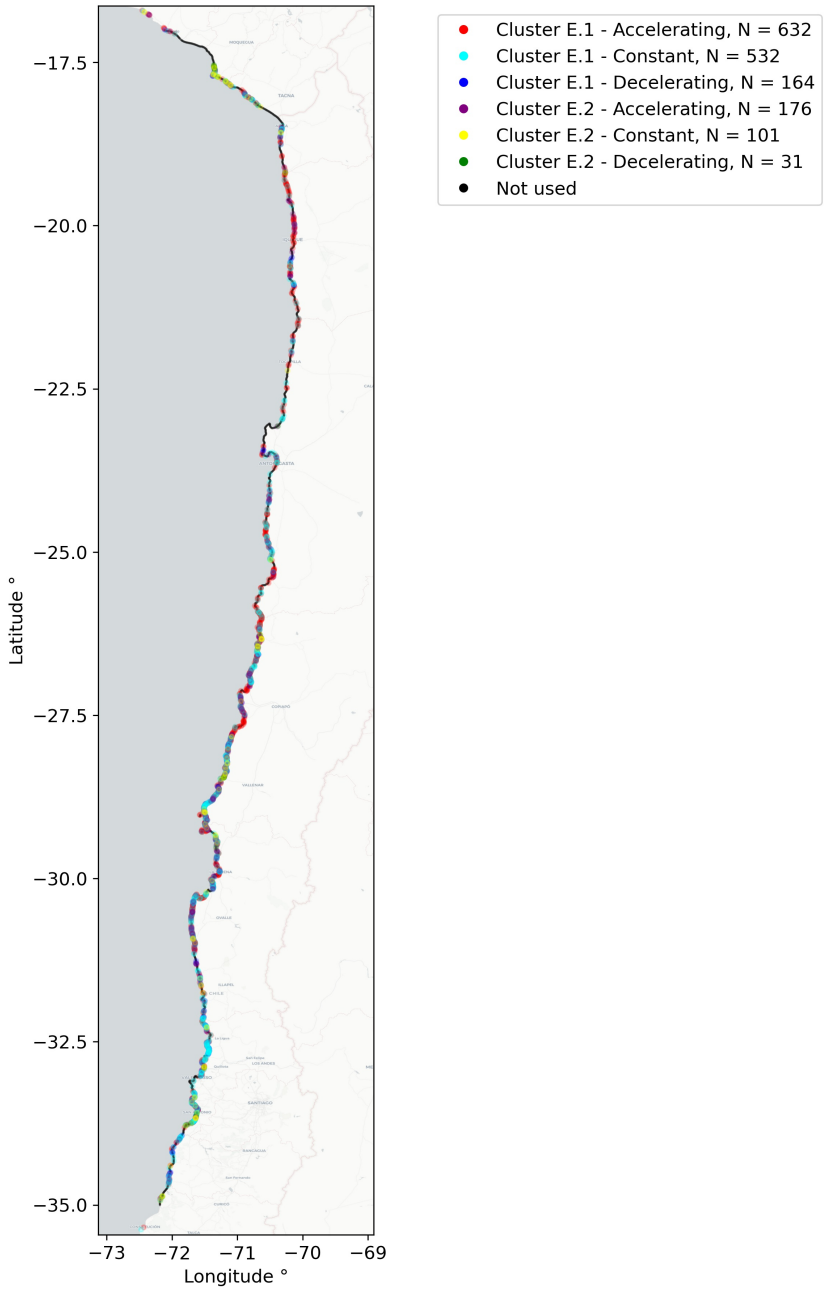

Figure 3.27: Country-level distribution of sandy transects in cluster E.1 and cluster E.2 along the coast of Chile. The legend confirms that erosion rates are increasing, since most transects are in the Accelerating sub-clusters.

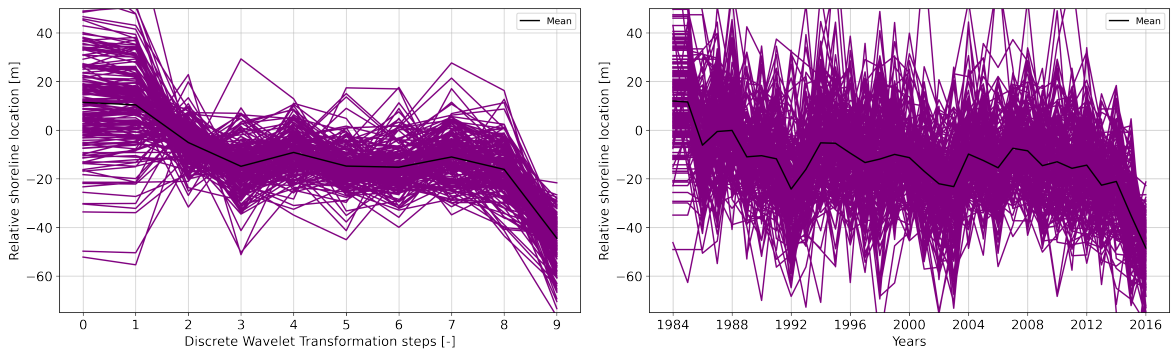

Figure 3.28: Transects along the coast of Chile in cluster E.2 Intense Erosion sub-cluster Accelerating.

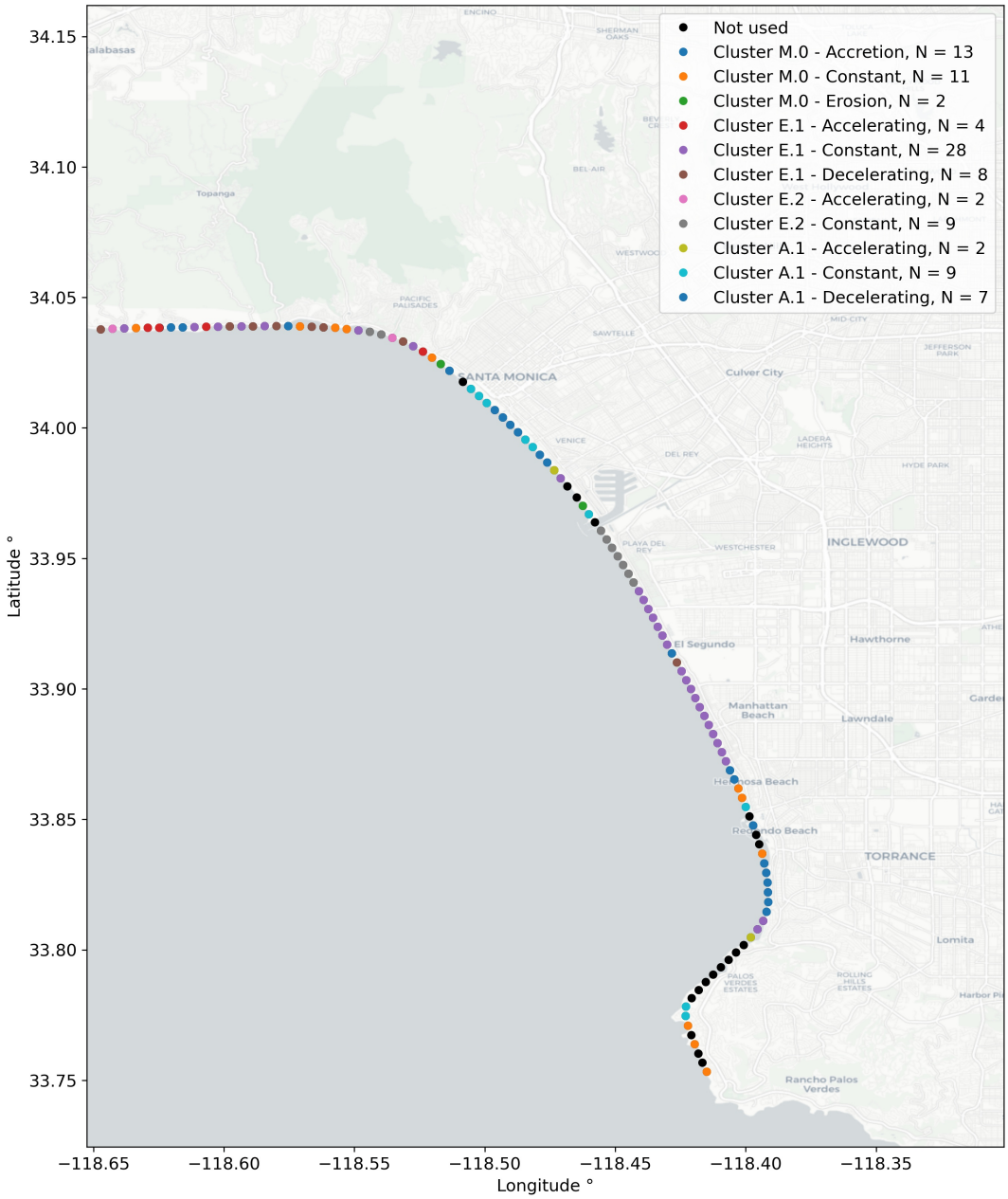

Figure 3.29: Local distribution of sandy transects along the beaches in front of the Los Angeles International Airport. The sub-clusters indicate whether behavior in clusters is accelerating or decelerating.

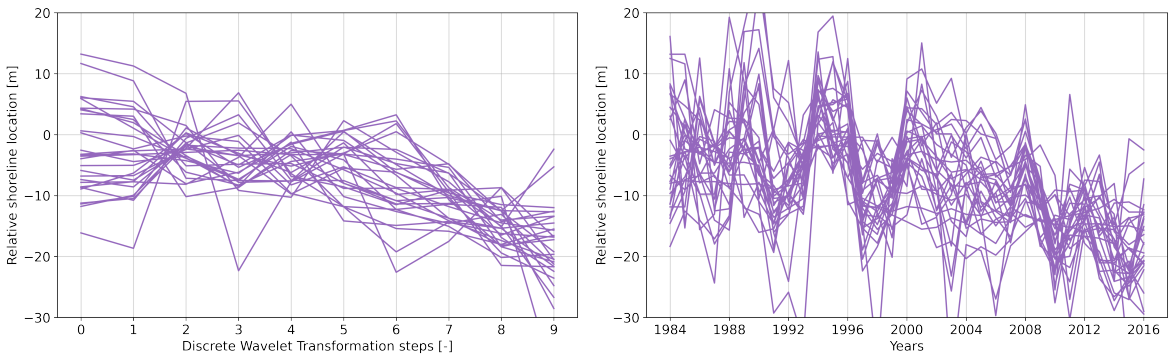

Figure 3.30: Transects in cluster E.1 Erosion in front of the Los Angeles International Airport showing constant erosive behavior

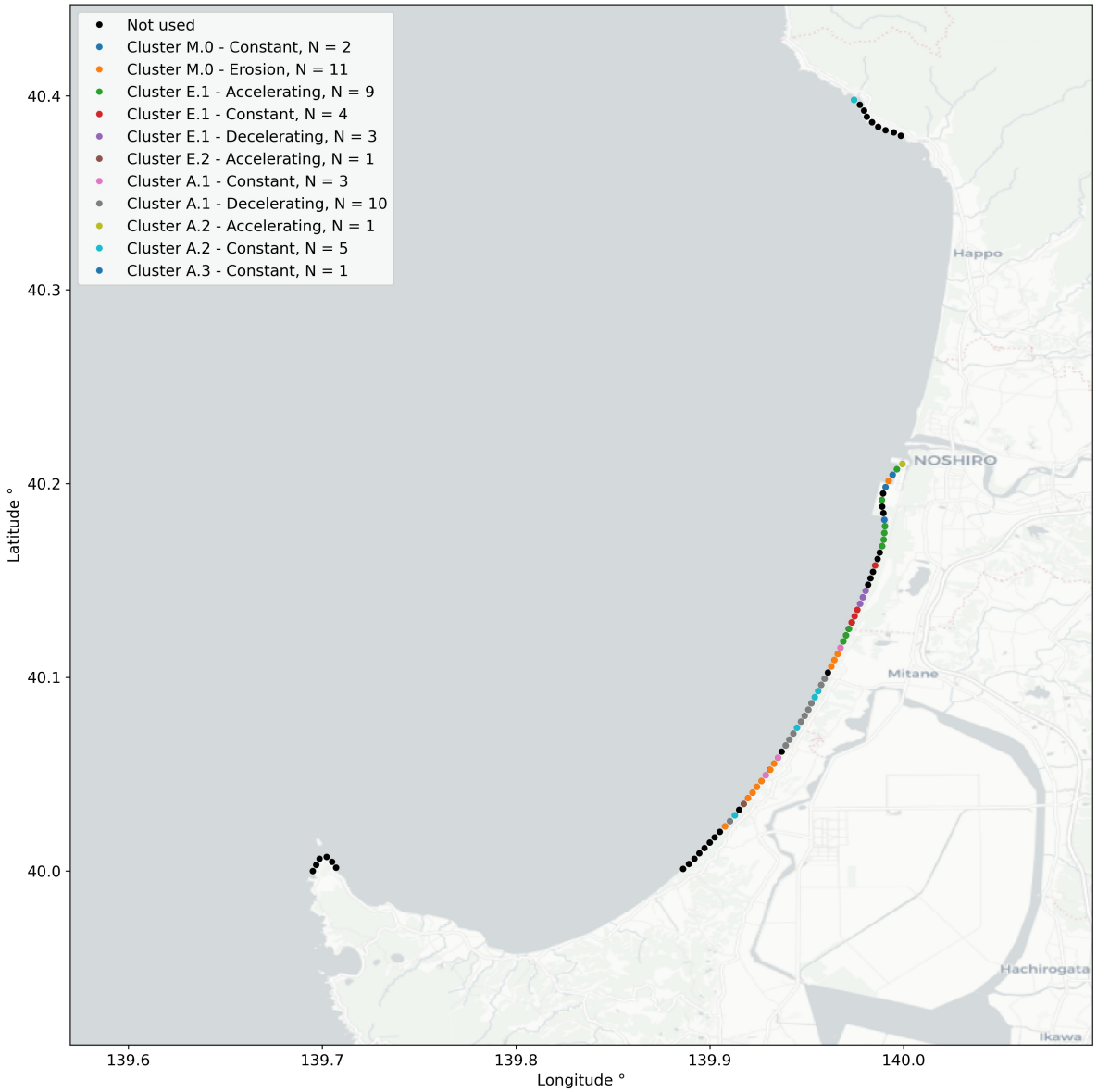

Figure 3.31: Distribution of sandy transects along the beaches in front of Mitane, Northern part of the Honshu main island of Japan. The sub-clusters indicate whether behavior in clusters is accelerating or decelerating on a local scale

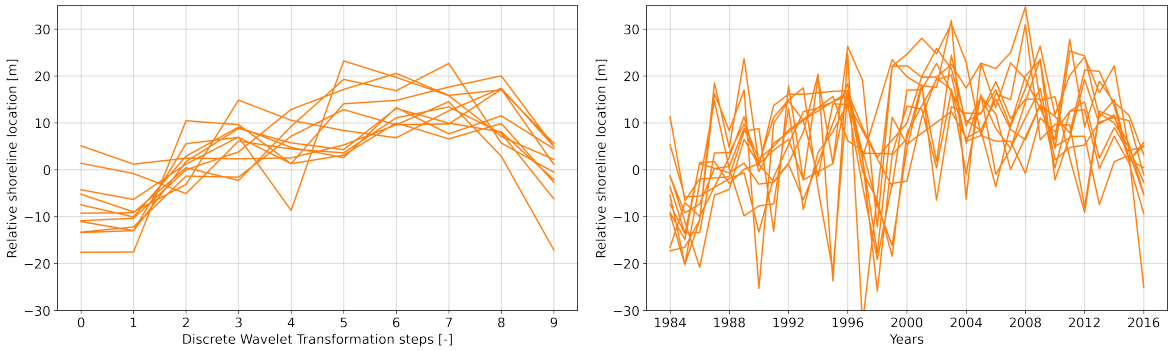

Figure 3.32: Transects in cluster M.0 Moderate and sub-cluster Erosion in front of Mitane along the Kamayahama beach in North Japan. Showing moderate behavior, tending more towards erosive behavior in last years.

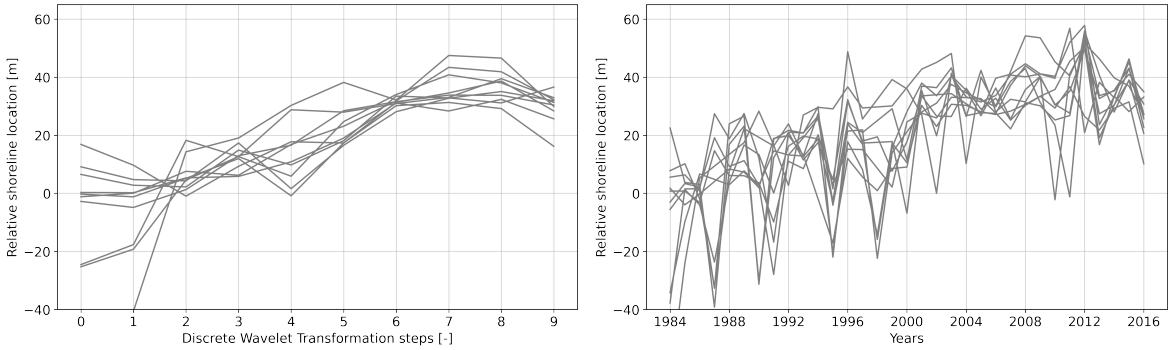

Figure 3.33: Transects in cluster A.1 Accretion and sub-cluster Decelerating in front of Mitane in along the Kamayahama beach North Japan. Showing accretion behavior, however in the last years the accretion trend are decelerated.

3.6 Step 5. Transect forecasting

After all the transects are assigned to a cluster, different models will be trained per cluster. The hypothesis states that different clusters holding transects with different behavior may perform better with different forecasting algorithms. The forecasting algorithms considered will be the algorithms: SimpleFFN, DeepAR, MQCNN, DeepSSM (Calkoen et al., 2021) (Alexandrov et al., 2020). These algorithms enable global training of the models and are therefore useful for this research. SimpleLSTM is not used in this section as it trains and evaluates itself differently than the four algorithms mentioned above. First, the evaluation measures to calculate model performance are introduced. Second, the general model setups are defined and the aimed results are described. Third, the results of step 5 are presented in general and exciting transects are featured. Finally, the optimal form of the predictions is presented for further research in step 6.

3.6.1 Evaluation measures and forecast setup

Current literature used the same algorithms for forecasting of SDS data and can be used as a benchmark (Calkoen et al., 2021). For using this particular study as a benchmark, the same forecast measurements should be used as well. MSE and MAE are already introduced in Section 3.2 in equations 3.1 and 3.2. The rest of the measurements are presented in equations 3.4, 3.5, 3.6 and 3.7. All measurements have their advantages and disadvantages and are used for different purposes. Mean Absolute Error (MAE) and Mean Squared Error (MSE) are both scale-dependent (the scale depends on the scale of the data) and often used to compare different forecast algorithms on the same dataset. MSE is often more sensitive to outliers as it gives more weight to them by squaring them. MAE on the other hand can be biased if the data is skewed as it tries to split the data in half. Furthermore, there are three other accuracy measures using also Absolute Error (AE) and Squared Error (SE): Root Mean Squared Error (RMSE) = $\sqrt{MSE}$, Normalized Root Mean Squared Error (nRMSE) = $\frac{\sqrt{MSE}}{|\text{ target }|}$ and Normalized Difference (ND) = $\frac{AE}{|\text{ target }|}$.

Mean Absolute Percentage Error (MAPE) and symmetric Mean Absolute Percentage Error (sMAPE) are error measures based on percentage errors. Since these are percentage measures, these measurements can be used to compare forecast algorithms along with different datasets, as these measurements are scale independent. As sMAPE counters the shortcomings of MAPE, both are sensitive to values close to 0. If z_t approach zero for a value, both measurements tend to infinity. To overcome this problem, the number 1 is added to both denominator and nominator for observations with values $zi, t < 1e8$.

$$\text{MAPE}_i = \frac{100}{N} \sum_{t=1}^{N} \frac{\left|z_{i,t} - \hat{z}_{i,t}\right|}{z_{i,t}} \tag{3.4}$$

$$\text{sMAPE}_i = \frac{100}{N} \sum_{t=1}^{N} \frac{\left|z_{i,t} - \hat{z}_{i,t}\right|}{\left(\left|z_{i,t} + \hat{z}_{i,t}\right|\right) * 2} \tag{3.5}$$

Next to percentage errors, scaled errors can be used to compare forecast accuracy across series with different units. Mean Absolute Scaled Error (MASE) was introduced by Hyndman and Koehler (2006) and is based upon scaled errors by using naïve forecasts to calculate this error.

$$\text{MASE}_i = \frac{1}{N} \frac{\sum_{t=1}^{N} \left|z_{i,t} - \hat{z}_{i,t}\right|}{\frac{1}{M-1} \sum_{t=2}^{M} \left|z_{i,t} - z_{i,t-1}\right|} \tag{3.6}$$

Probabilistic forecasting algorithms give a probability distribution as output instead of point estimates. As these forecasting algorithms generate prediction intervals, evaluating a single forecast is

not possible. For these forecasts, a sample of forecasts is generated with the trained model. These samples follow the empirical distribution of the forecasts. Evaluation is finally done by looking at whether a sample occurs below a prediction interval. Equation 3.7 is used to measure the difference between the observed values and Cumulative Distribution Function (CDF) (Matheson & Winkler, 1976). Here is $\hat{Q}$ the estimated CDF, p the probability level, and z are the observed values. For the probability forecast algorithms, the CDF is estimated by generating 300 samples. The quantile levels used in this research are $p \in [0.1, 0.2, \ldots, 0.9]$, where the mean per quantile approximates the final CRPS.

$$\text{CRPS}(\hat{Q}, z) = 2 \int_0^1 [\hat{Q}(p) - z] \left[1_{\{z \leq \hat{Q}(p)\}} - p \right] dp \tag{3.7}$$

The last evaluation measure used is the Mean Scaled Interval Score (MSIS) (Gneiting & Raftery, 2007). Here the lower and upper intervals are used to evaluate the score.

The forecasting algorithms named in the introduction of these steps and Chapter 2 are the same as used in (Calkoen et al., 2021). Since previous research could be used as a benchmark, the hyperparameters per forecasting algorithm will be the same (Calkoen et al., 2021). Per cluster, the different forecasting algorithms are fitted and the accuracy is measured, with equations 3.1 till 3.7. The final seven data points per transect will be used as the test set for evaluation and the twenty-six other data points will be used for training and validating the algorithm, see figure 2.13. This split results in a large set of different inputs per cluster, where different train and validation windows were used per transect. Furthermore, all four algorithms support early stopping, which counteracts the overfitting of the algorithms on the data.
The hypothesis is that some clusters may better cope with different forecasting algorithms. To investigate this hypothesis, the mean evaluation measures per cluster are reviewed. However, expected is that, since the variance in behavior of transects per cluster is high, not one forecasting algorithm can be chosen as the most optimal for a cluster. After the evaluation, the algorithms will be trained again, now for pure prediction purposes. Here all thirty-three years will be used as input for the algorithms and the forecast horizon will be set to fourteen years, till 2030. For all clusters, all the forecasting algorithms will be trained. This will result in N x K trained algorithms, where N = number of clusters and K = number of forecasting algorithms. For a transect, the four different forecasting algorithms are trained. Every algorithm is only trained with transects in the same cluster. Finally, per transect, an expert judgment has to be made. This expert judgment considers the overall performance of an algorithm for a cluster and the performance of algorithms on a transect scale.

3.6.2 Results

In this subsection, the final results of step 5 are presented. First, the mean evaluation measures of all the forecast models per cluster are presented. Second, some predictions results will be shown by picking interesting transects in different clusters and review how the algorithms perform per transect. The comparison between overall algorithm performance and individual transect performance is researched. For local cases, the sub-clusters will improve understanding and help to choose the best forecast algorithm for a single transect.

Forecast per cluster

In table 3.10 the mean scores per evaluation measure for the cluster E.1 Erosion are stated. The table holds the different evaluation measures for every forecasting algorithm per cluster. For comparison, the forecast errors of Ordinary Least Squares (OLS) are added as well. In this section, only MSE and

MAE highlighted as these are the most important for the use case. Since other scores are scaled measures of percentage errors, these errors are dependent on the relative location of the end of the train window. These are of larger importance for a cluster with more severe and extreme behavior. However, for forecasting shorelines, an indication of the absolute difference between the predicted location and observed value is most important, MSE and MAE are used. The remainder scores per forecasting algorithm per cluster are stated in Appendix C.2. By looking at the MSE and MAE in table 3.10, the scores are significantly lower compared to the one of previous studies (Calkoen et al., 2021). In previous research, an overall MSE around 1800-2000 was found for all four algorithms and a MAE between 17-19 m. Especially the lower MAE score is an improvement, the mean difference between an observed and predicted value has been improved by 40%. Furthermore, the MSE is sometimes 10% of previous scores. This decrease in errors indicate that the maximum error between an observed and predicted value is reduced drastically, as the maximum value influences the MSE most. In previous research, the mean evaluation measures were heavily influenced by large and absolute steps in the last seven years. In this study, these large and absolute steps were tackled by a threshold, as described in Appendix B.1. Another result that stands out is that the four ML algorithms all outperform the OLS based on MSE and MAE. The improved mean scores provide more confidence in using all the predictions for the transects in cluster E.1 Erosion.

Table 3.10: Mean aggregated evaluation measures forecasting algorithms for cluster E.1. MAE and RMSE are in meters. Some scores for MQCNN are left out as the calculation of these scores was different compared to the other algorithms. The MSE and MAE are here significant lower than for previous research, indicating a higher accuracy in m^2 and m. Furthermore, all forecasting algorithms outperform OLS. The best algorithms based on MSE and MAE are DeepAR and MQCNN.

N = 86,196	MSE	MAE	MASE	MAPE	sMAPE	RMSE	NRMSE	ND	MSIS	CRPS
SimpleFFN	224	10.8	2.25	20.15	0.81	14.97	0.80	0.58	51.26	0.47
DeepAR	190	10.1	2.18	15.94	0.77	13.78	0.73	0.54	50.45	0.44
MQCNN	-	10.1	2.52	14.39	0.81	-	-	0.54	-	0.42
DeepSSM	254	11.6	2.36	18.08	0.88	15.92	0.85	0.62	53.79	0.51
OLS	297	12.4	2.45	20.98	0.88	17.22	0.92	0.67	-	-

Forecast per transect

Next to the overall accuracy of the algorithms, evaluated on the last seven years, interesting transects will be highlighted. For this subsection, all thirty-three data points are used as input for the algorithms, and the forecast horizon was set for fourteen years, till 2030. Here the cluster refinements added value for forecasting becomes evident. Considering the four different predictions and the sub-cluster the transect is in, a more weighted choice can be drawn whether to follow which forecast. This weighted choice will be demonstrated with examples that show the approach and analysis. For SimpleFFN, DeepAR and DeepSSM, the probability spreads of 90% and 50% interval is plotted. The MQCNN algorithm generates quantiles as output and thus, for this algorithm, the 80% and 40% interval is sketched. The colors of the example predictions differ, the erosive transects are predicted in red, the moderate transects in blue and the accretion transects in green.
First, considering a beach in Chile, an overall accelerating erosive trend was detected in Section 3.5.2. For this example, a transect around Viña Del Mar is considered; BOX_073_004_46. This transect falls in cluster E.1 and sub-cluster Accelerating. Indicating that this transect shows erosive behavior and in the last years, the erosion rates are increasing. In figure 3.34 the predictions per forecasting algorithm are presented. Following the overall accuracy in table 3.10, the DeepAR and MQCNN show the best performance based on MSE and MAE, and should be followed. This is the most convenient approach for choosing the best forecast. However, when considering the sub-cluster the transect is in, the erosion rates are increasing. This accelerating behavior should be detected by the forecasting algorithm and likewise be predicted that way. By looking at the fore-

casts, the first thing that stands out is that the SimpleFFN forecast probability spread is relatively large compared to other forecasts. This large spread could be because the SimpleFFN is a relatively simple MLP network and does not consider the previous behavior of a time series. Furthermore, the DeepSSM forecast has a relatively small probability spread and seems heavily biased by the latest trend of the transect. The predictions of the MQCNN and DeepAR algorithms are in trend the same, but MQCNN has a smaller probability spread.
As indicated, the predictions should follow the sub-cluster behavior of this transect. MQCNN and DeepAR are showing slightly accelerating behavior. This can be visually checked by comparing the median prediction with the OLS, the lines are slightly diverging. The DeepSSM prediction is heavily accelerating the erosion trend. Considering the predictions and sub-cluster information, the predictions of MQCNN, DeepAR, and DeepSSM are the most reliable. By following the worst-case scenario of these predictions, the transect will further erode with $\pm 5 - 15m$ compared to the shoreline location of 2016. Besides choosing what prediction is most reliable, the sub-cluster could also be used to account for behavior in the 90% or 80% interval.

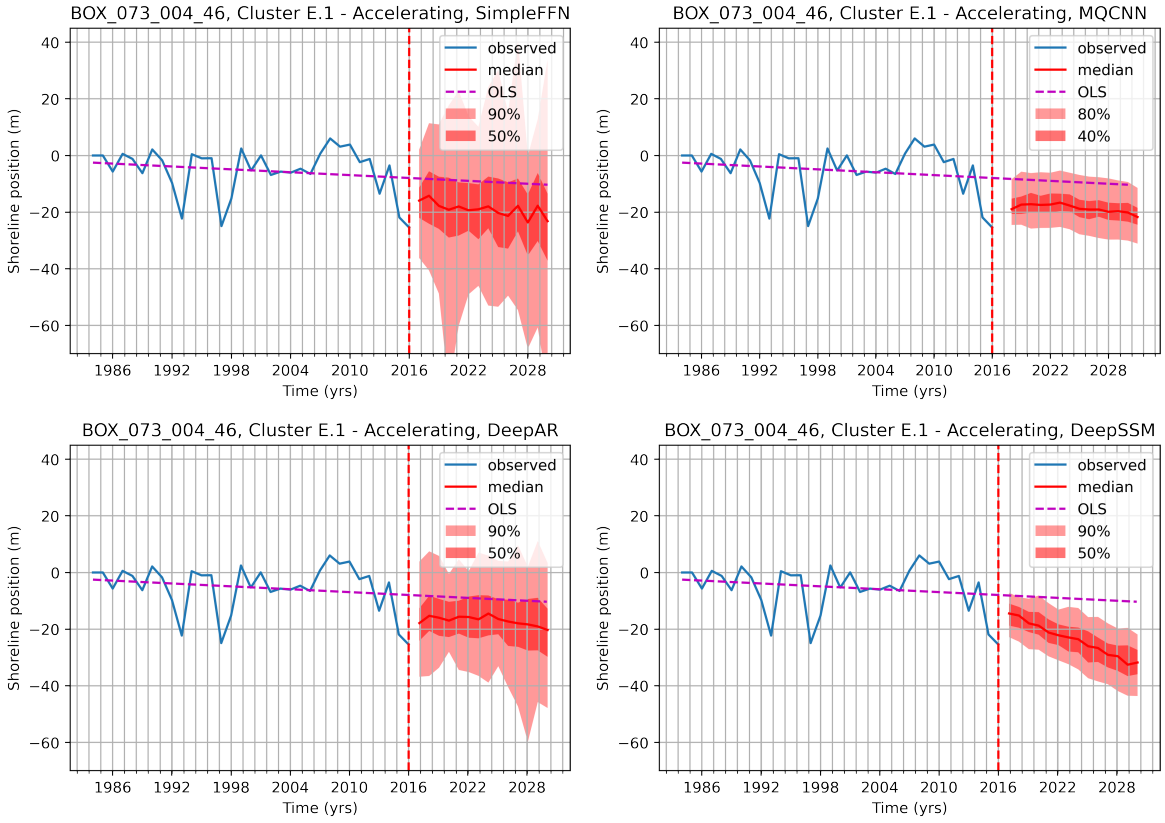

Figure 3.34: Forecast of transect BOX_073_004_46 in Viña Del Mar, Chile. The transect is in cluster E.1, showing erosion. The sub-cluster is Accelerating. The predictions are generated by using all 33 years of all transects in cluster E.1 as input. The prediction horizon is 14 years, from 2016 till 2030. Per algorithm the median is presented with probability intervals. These are derived by generating 300 different forecast paths.

Another transect that is considered is one in Melbourne in Australia at the Aspendale Beach. The transect BOX_063_119_52 is clustered in cluster E.1 and sub-cluster Decelerating. This (sub-)cluster indicates that the transect shows erosive behavior but that this erosive trend is decelerating in the last years. In figure 3.35 the predictions for all the algorithms are presented. Again, the overall accuracy in table 3.10 proposes DeepAR and MQCNN as the best forecasting algorithms. Next to the overall quantitative analysis, the individual predictions in figure 3.35 should be (qualitatively) examined. The first thing that strikes is that the latest erosive trend again highly biases the DeepSSM

algorithm. Another thing is that for the SimpleFFN and DeepAR algorithms, the predictions show decelerating predictions following the sub-cluster of the transect. This is an interesting finding, as these forecasting algorithms are detecting the same long-term behavior as the clustering approach did. So for these predictions, the reliability is increased as the predictions follow the behavior of the sub-cluster. This means that the SimpleFFN and DeepAR algorithm are the most reliable as a forecast for this transect, considering sub-cluster behavior. This analysis will eventually lead to the DeepAR prediction to follow since it covers sub-cluster behavior and has high overall accuracy.

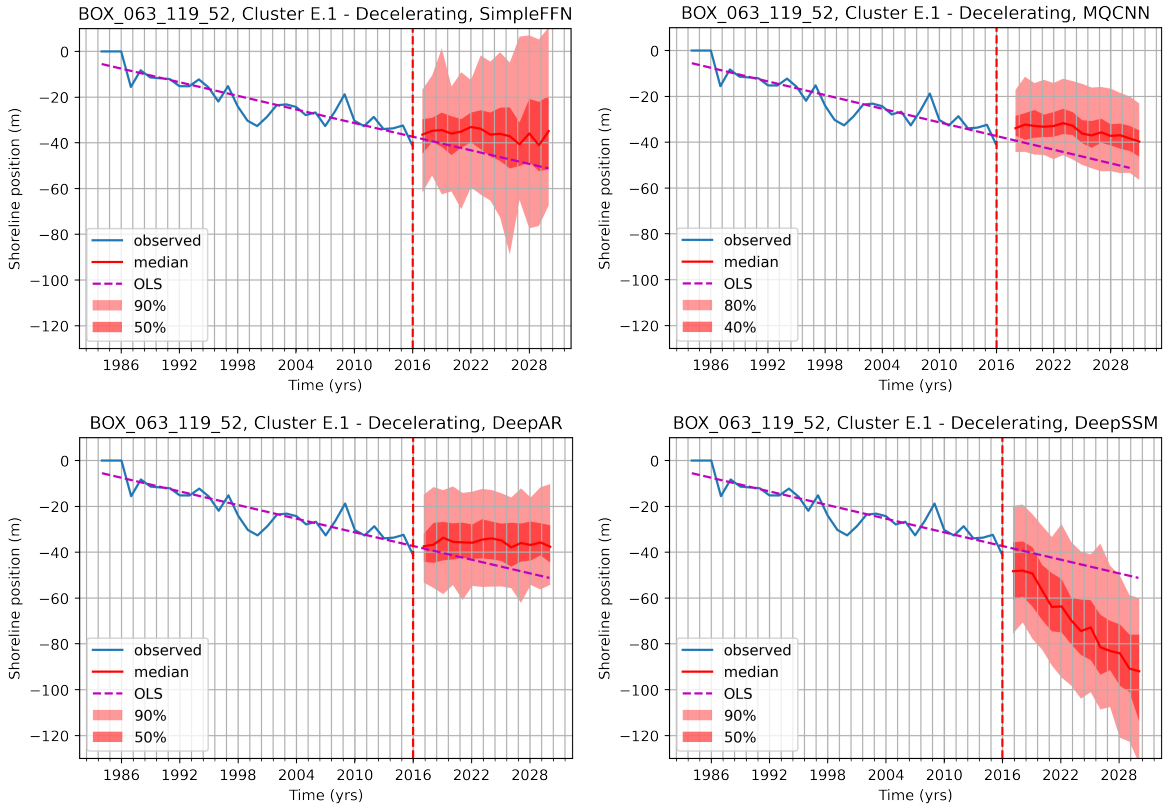

Figure 3.35: Four different predictions of transect BOX_063_119_52 in Melbourne, Australia. The transect is in cluster E.1, showing erosion. The sub-cluster is Decelerating. The predictions are generated by using all 33 years of all transects in cluster E.1 as input. The prediction horizon is 14 years, from 2016 till 2030. Per algorithm the median is presented with probability intervals. These are derived by generating 300 different forecast paths.

The final transects considered are at the Kamayahama beach in Japan and are already examined in figure 3.31. First, BOX_172_380_5 is a transect in cluster M.0 and sub-cluster Erosion, meaning that the last years tend more towards erosive behavior. The predictions for this transect are depicted in figure 3.36. Second, transect BOX_172_381_68 lies on the same shoreline as transect BOX_172_380_5, in the Northern Japan of the main island. However, this transect is clustered in cluster A.1 - Decelerating. This sub-cluster indicates that the transect was showing accretion but that this accretion trend decelerated in the last years.
The predictions of transect BOX_172_380_5 are stated in figure 3.36. All predictions, except DeepAR, are following the sub-cluster behavior of this transect. This transect in sub-cluster M.0 - Erosion, meaning that the overall trend is moderate but that the transect tends towards erosive behavior. In addition, considering the overall accuracy in table C.1, SimpleFFN and DeepAR are performing best on MSE and MAE. MQCNN and DeepSSM have the highest MAE score of all four algorithms. Combining both qualitative and quantitative analysis, the SimpleFFN predictions is most reliable for this transect with the best overall accuracy and best representing sub-cluster behavior.

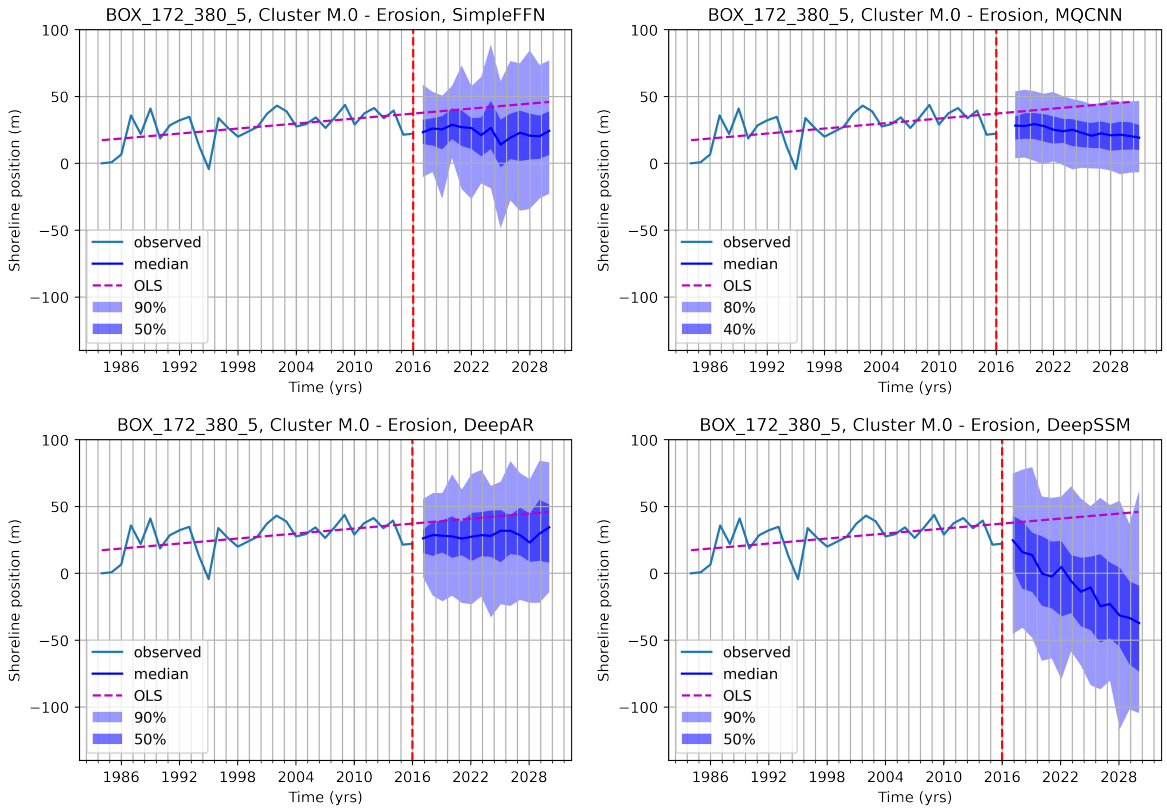

Figure 3.36: Forecast of transect BOX_172_380_5 in the north of Honshu, the main island of Japan. The transect is in cluster M.0 and sub-cluster Erosion. The predictions are generated by using all 33 years of all transects in cluster M.0 as input. The prediction horizon is 14 years, from 2016 till 2030. Per algorithm the median is presented with probability intervals. These are derived by generating 300 different forecast paths.

For transect BOX_172_380_68 in figure 3.37, all forecasts predict the same decelerating trend found in the clustering step. One thing that stands out is that the DeepSSM shows the most extreme decelerating prediction. This again, is caused by the high influence the last years in the training set have on the predictions of the DeepSSM algorithm. For final predictions usage, all four are reliable since they recognize the same transect behavior as in the clustering section. However, the scores in C.5 clearly state that DeepAR and MQCNN have the highest accuracy, with MQCNN even outperforming DeepAR on MAE.

Summary & implications next step(s)

In the examples above, the multi-method approach for selecting the most reliable and accurate prediction is presented. This multi-method approach could be a helpful tool for third parties to choose the most reliable local predictions. In addition, it became apparent that per cluster, no absolute best forecasting algorithm can be chosen for all transects. Next to that, the performance of the forecasting algorithm differed per cluster, highlighting the value of the clustering forecasting approach. Furthermore, the examples in this section help comprehend the overall accuracy per cluster on a local scale. Overall, the SimpleFFN and DeepSSM have the lowest overall accuracy, considering MSE and MAE. Next to that, SimpleFFN and DeepSSM seem to have the largest probability spread and DeepSSM predictions are highly biased by the latest trend. The predictions till 2030 will be further used in step 6.

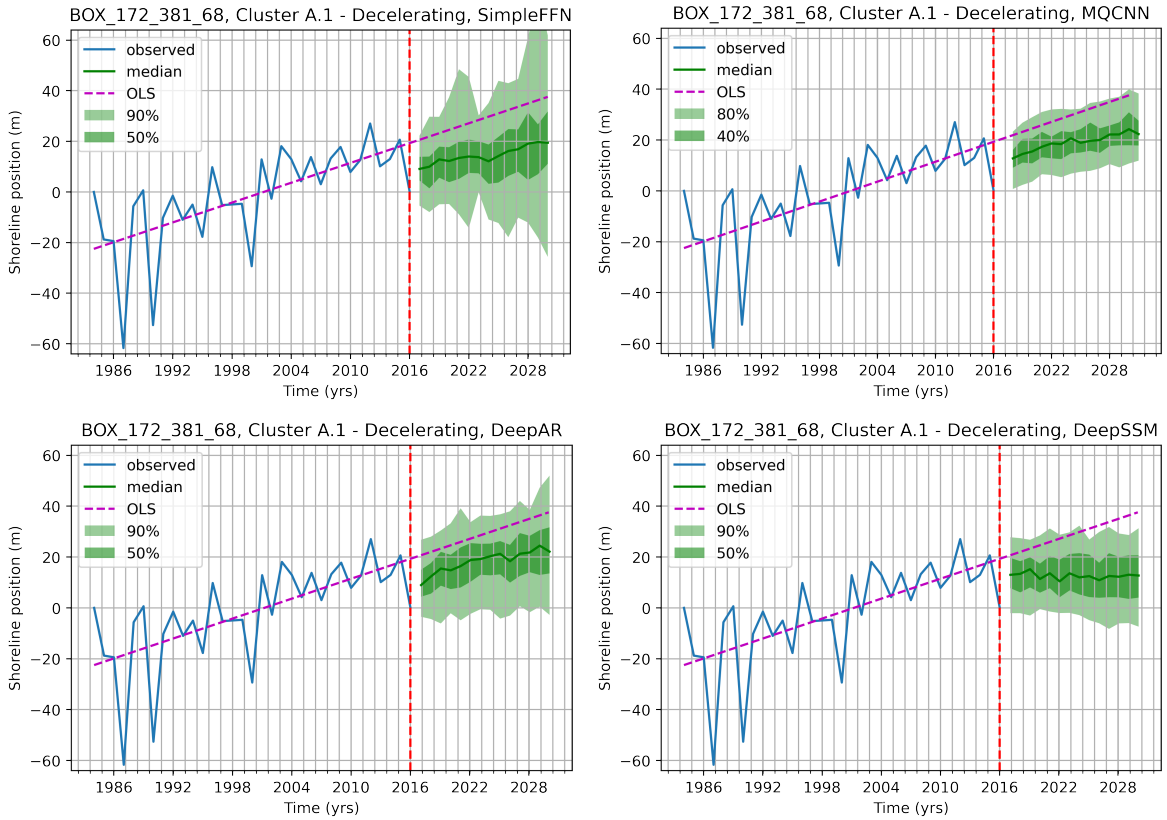

Figure 3.37: Forecast of transect BOX_172_381_68 in the north of Honshu, the main island of Japan. The transect is in cluster A.1 and sub-cluster Decelerating. The predictions are generated by using all 33 years of all transects in cluster A.1 as input. The prediction horizon is 14 years, from 2016 till 2030. Per algorithm the median is presented with probability intervals. These are derived by generating 300 different forecast paths.

3.7 Step 6. Application

In this final sub-question, the predicted transects of step 5 will get reviewed. The review is two-folded: First, to what extent are the forecasts possible regarding coastal features and other physical boundaries. Second, the predicted transects are projected at a local scale. If there will be any interference with roads, housings, or other human infrastructures, a flag should be raised. First, the global database Global Road Inventory Project (GRIP) is shortly introduced. Second, the method to combine the predictions with the database is presented. Finally, the results are presented with some highlighted transects that show the final use of the GRIP database.

3.7.1 Global database setup

The physical boundaries database used in this step will be roads from the GRIP database. Coastal areas give homes to large populations, and this comes with infrastructure. The GRIP database provides vector maps with roads per continent on a local scale. First, the predictions of transects are plotted orthogonally to the global shoreline, extracted from the OpenStreetMap (OSM) from 2016. Second, combining the predictions orthogonal to the global shoreline and the vector maps of the GRIP database, one could identify when the predicted behavior of a transect may interfere with a road.

3.7.2 Results

In this results section, the last sub-question will be answered: *To what level can open-source information be used to obtain a more realistic local forecast?*. In this section, the forecasts will be projected back locally on the scale to examine whether there is any interference with road infrastructure from the GRIP database. In figure 3.38, transect BOX_085_048_12 is used to show the result of this step. The transect is an accelerating transect in cluster E.1 and situated at Margate Beach in Brisbane, Australia. By projecting the prediction of this transect on the scale back on a map, any interference with physical boundaries can be checked.

In figure 3.38 the same graphs are plotted like the ones in Section 3.6.2. However, here a blue horizontal line is added at -25 m to indicate a possible shoreline located in the future if the worst-case predictions of all four predictions are followed. By projecting these predictions back on the geographical location, together with the GRIP database and satellite images as a background, the impact of predictions on the coastal zone can be reviewed. In figures 3.39 and 3.40 the result of this step is presented. Here the red line is the prediction following the -25 m erosion prediction. Necessary to state here is that the starting point of the transect is the transect intersection with the 2016 OSM shoreline. Thus, the shoreline at -25 m should be projected relative to the 2016 location of the transect, which is in this case -14 m, which means that the erosion projected amounts to 25 - 14 = 11 m. The yellow lines are the roads in the GRIP database, and as can be seen, the erosion will not interfere with the local roads. Following the projected prediction in figure 3.39 and the DeepSSM prediction in figure 3.38, there is a 50% chance that the beach will be almost fully eroded in 2028. This approach could be used for coastal managers to review whether a prediction could be threatening the hinterland, as the case is in this example.

Also, in figure 3.41 the predictions of transect located near Cape Town in South Africa are plotted. The beaches of Cape Town are experiencing SLR and an increased storm intensity that leads to threatening erosion rates (Dube et al., 2021). The same holds for the transect that is clustered in cluster E.2 and the sub-cluster Constant. Here, two of the four predictions show a future relative shoreline location of -140 m, SimpleFFN already 2021, and DeepSSM in 2020. By projecting

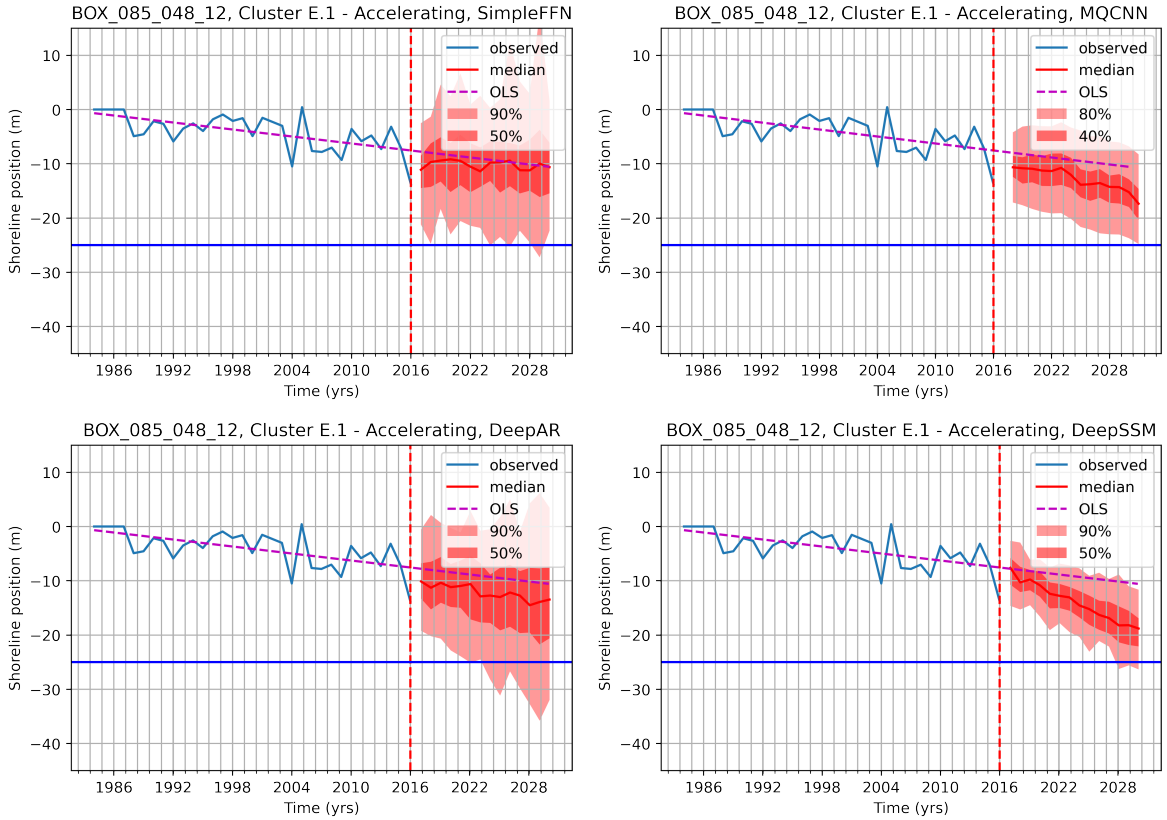

Figure 3.38: Forecast of transect BOX_085_048_12 situated in Brisbane, Australia on the Margate Beach. The transect is in cluster E.1 and sub-cluster Accelerating. The blue horizontal line indicates a possible shoreline position of -25 m in the future.

this erosion path back on its geographical location, any interference with roads can be detected. Again, the -140 m should be projected relative to the 2016 shoreline position, which is around -80 m. Meaning that the red line in figures 3.42 and 3.43 has a length of 60 m. In this case, the erosion path is intersecting with roads from the GRIP database. This could be used to quantify the intersection point. By scaling the intersection back to meters, the following output was generated: *Transect BOX_056_007_9 intersects with a road at 51.39 m from the coast at Longitude and Latitude degrees: 18.47223421067029 -33.81514925236754.* Next to this vulnerability assessment, reliability could also be considered. By carefully looking at the predicted erosion in figures 3.39 and 3.43 it can be observed that the prediction is interfering with vegetation before reaching the road. The erosive prediction was generated by erosion rates of sandy shorelines, that these erosion rates will keep going with overgrown beaches in unlikely. Thus it is likely that erosion rates will decrease when it starts interfering with the vegetation.

So, when the GRIP database is complete, the impact can be quantified instead of only a qualitative assessment, like in the previous example. The fact that the prediction intersects at 50 meters from the transect's intersection coordinates could be used to review further the graphs in figure 3.41. Visualizing a horizontal line at -80 + -50 = -130 m in the graphs, the year when the interference occurs can be extracted. For example, visualizing the horizontal line at -130 m in the SimpleFFN forecast of figure 3.41, results in an intersection with the median prediction around 2026. This means that by 2026, if the transect follows the median prediction of SimpleFFN, the beach will be entirely eroded and waves will start to interact with roads.

Figure 3.39: Predictions of transect BOX_085_048_12 projected back on its location in Brisbane, Australia on the Margate Beach. The small red line is the prediction for -25 m following the worst-case predictions. The yellow lines are the roads incorporated in the GRIP database. The map is used to compensate for missing data in the GRIP database and to help visualize local coastal vulnerability.

Summary & implications next step(s)

The results of this subsection show that by using a global dataset, in this case the GRIP database, the predictions become more reliable and valuable on a local scale. The result is a high-resolution vulnerability assessment, both qualitative and quantitative. Still, it is not complimentary, as the transects are only every 500 m, and the aim is to have a continuous prediction along the shoreline. Next, it is debatable whether the GRIP database is the most optimal database to use in this sub-question. These restrictions will all be discussed furthermore in Chapter 4.

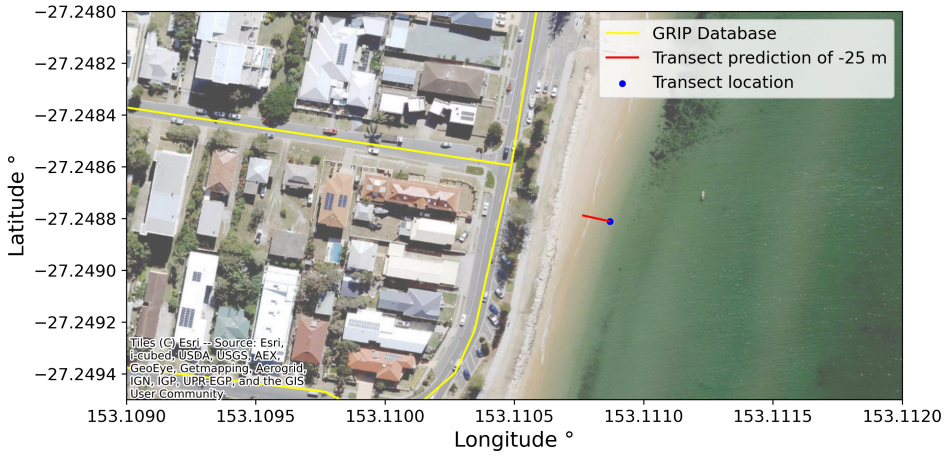

Figure 3.40: Zoomed in version of figure 3.39 Predictions of transect BOX_085_048_12 projected back on its location in Brisbane, Australia on the Margate Beach. The small red line is the prediction for -25 m following the worst-case predictions. The yellow lines are the roads incorporated in the GRIP database. The map is used to compensate for missing data in the GRIP database and to help visualize local coastal vulnerability.

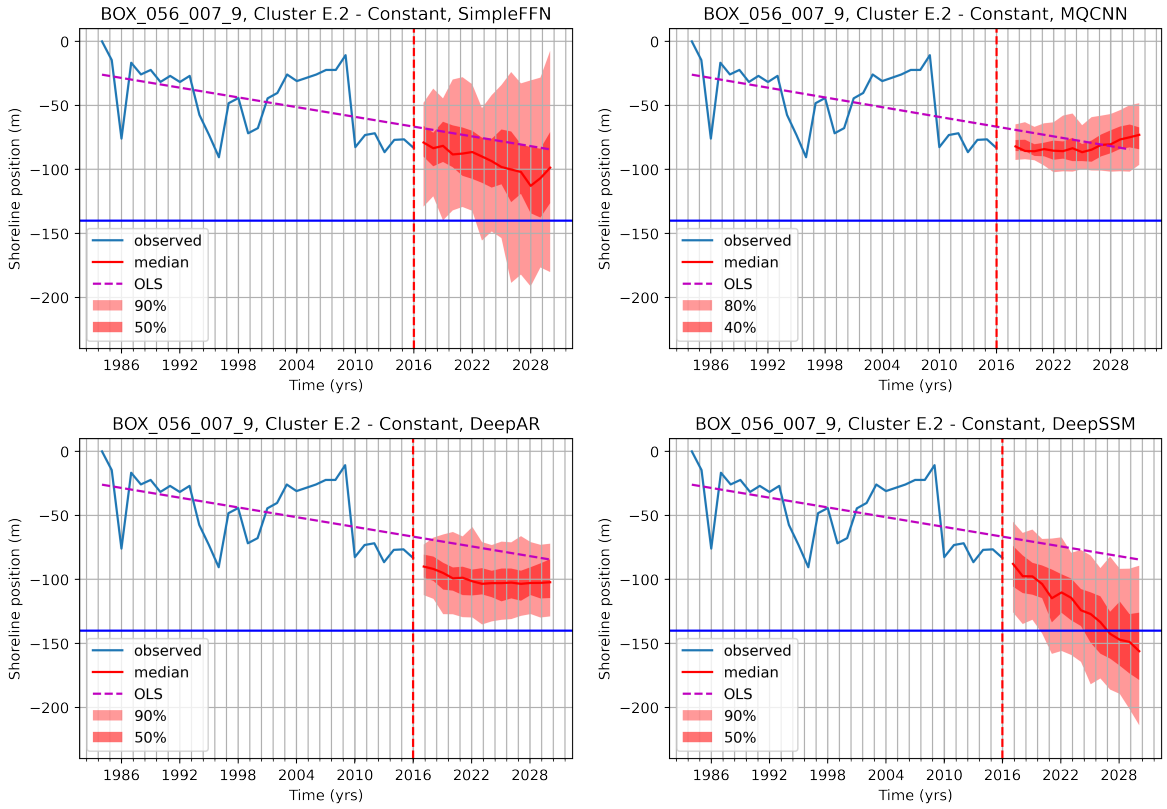

Figure 3.41: Forecast of transect BOX_056_007_9 situated in Cape Town, South Africa on the Blouberg Beachfront. The transect is in cluster E.2 and sub-cluster Constant. The blue horizontal line indicates a possible shoreline position of -140 m in the future.

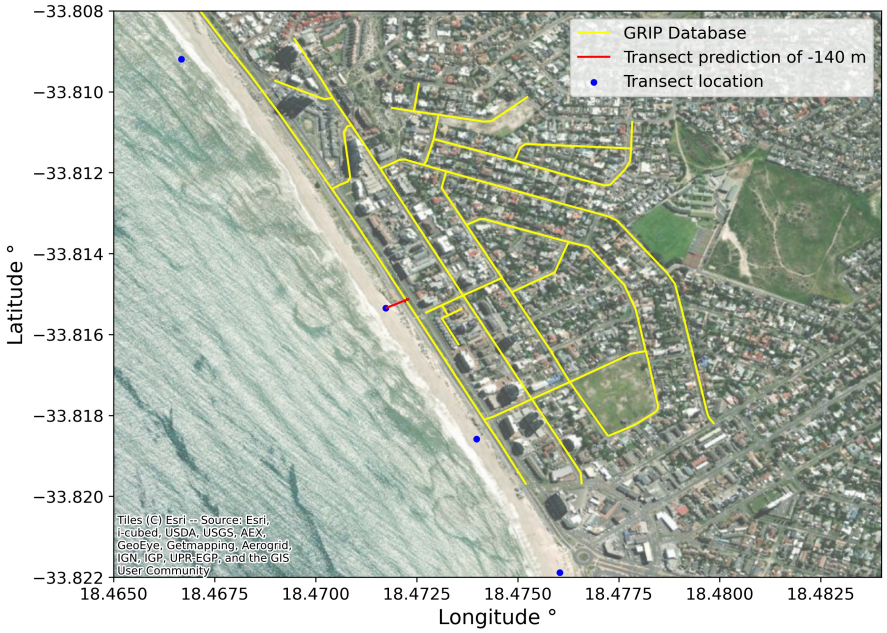

Figure 3.42: Predictions of transect BOX_056_007_9 projected back on its location in Cape Town, South Africa on the Blouberg Beachfront. The small red line is the prediction for -140 m following the worst-case predictions. The yellow lines are the roads incorporated in the GRIP database. The map is used to compensate for missing data in the GRIP database and to help visualize local coastal vulnerability.

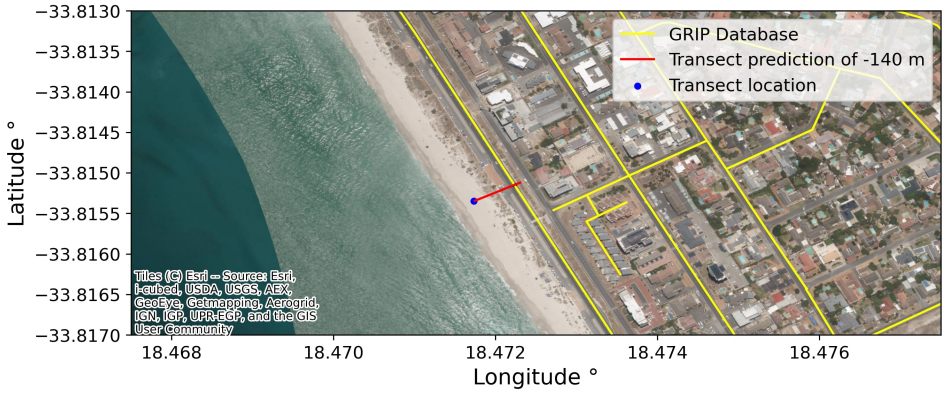

Figure 3.43: Zoomed in version of figure 3.42. Predictions of transect BOX_056_007_9 projected back on its location in Cape Town, South Africa, on the Blouberg Beachfront. The small red line is the prediction for -140 m following the worst-case predictions. The yellow lines are the roads incorporated in the GRIP database. The map is used to compensate for missing data in the GRIP database and to help visualize local coastal vulnerability.

Chapter 4

Discussion

In this chapter, the results will be discussed. The aim is to go in-depth about the meaning of the results and to put them in context. Also, the assumptions made in this thesis are reviewed. The chapter will be divided into three sections. In the first Section 4.1, the results' interpretation and implications are considered. In Section 4.2 limitations due to assumptions and methodology are described. Final, in the last Section 4.3 some future directions are explored.

4.1 Implications

In this section, the implications of the results are described. The main results are listed below and for each result the applicability is described:

- Clustering of transects based on long-term behavior creates unique and new insights into shoreline behavior on a global and local scale.
- By combining (sub-)cluster information and tailor-made predictions, a weighed interpretation of a forecast can be made. This multi-method approach results in a more reliable and accurate prediction per transect on a local scale.
- Combining transect predictions and local information in the form of physical boundaries, predictions can be translated towards a qualitative and quantitative impact assessment on the specific coastal zone.

Human interventions highly influence shoreline behavior and thus have a major role in shaping the coastal zone. For Example, as highlighted in Section 3.5.2 in figure 3.29, the shoreline is formed by building jetties for the Marina Del Rey. In order to understand the long-term response of adjacent beaches on this kind of human intervention, the clustering approach can be used as it considers the sub-clusters of these transects. Furthermore, the long-term response of transects to other phenomena on a global and local level can be examined as well. Such as the clustering information facilitates the research on the influence of SLR on local shorelines.

In addition, the combination of clustering information with the predictions leads to a quantitative and qualitative-based decision for a forecasting algorithm. A specific forecasting algorithm could have a significantly better overall accuracy for a cluster. Based on this accuracy, all the transects in the cluster could be forecasted using this algorithm. However, by reviewing single transect predictions based on transect behavior, more reliable predictions are made than purely relying on overall prediction accuracy.

Translating the predictions towards local erosion paths will qualitatively support the understanding of the impact of the forecast. The predictions can be projected back on their location, which results in a visual assessment of whether there is any interference with physical boundaries. In the case of a global and local complete database, the interference between predictions and boundaries can be quantified.

4.2 Limitations

This section elaborates on the limitations of this research. These limitations are mainly due to assumptions and choices made during the process and methodology, as described in Chapter 3. The section is divided into four subsections following the four central pillars of this research. First, the SDS data together with the pre-processing steps is reviewed. Second, the choices and assumptions regarding the clustering of transects are considered. Furthermore, the limitations of transect forecasting are examined. Last, the applicability of the generated predictions in step 6 is researched.

Data & pre-processing

The SDS data used in this thesis describes the location of transects relative to the 2016 derived shoreline from OSM. The relative distance is presented for every year from 1984 till 2016, resulting in 33 data points per transect. Every data point in the Shoreline Monitor is derived from bi-weekly measurements during a time period of one year. A data point represents the mean transect position over a year. Bi-weekly data helps to filter out tides, storm surges, and other extreme water level events. However, it could also be regarded as skipping valuable data.
The bi-weekly data that is input for the yearly SDS data must be used to obtain a higher temporal resolution. This lower temporal scale adds extra information for understanding global and local shoreline behavior and improving transect forecasting. The bi-weekly data will capture seasonal patterns like beach rotations and storm seasons, whether this is preferable for long-term predictions is questionable and removing seasonality with DWT should be done. On the other hand, the bi-weekly will probably generate a less wide probability distribution for the predictions leading to more reliable forecasts since more data is available for the forecasting algorithms. However, this will increase the dataset size and therefore decrease workability. It is a delicate balance between the information resolution needed and dataset workability.
Besides the temporal scale, the spatial scale also influences the results. Now transects every 500 m are considered, so spatial variability on a lower spatial scale will be left out. If transects with a smaller distance in between are considered, more local shoreline variability could be gathered. Next to that variability, more information will be created to use in a final forecasting algorithm, which will lead to more realistic forecasting for smaller stretches of sandy shorelines.
The SDS data is pre-processed before it could be used for this thesis. Two significant steps will be discussed here; the linear interpolation to fill missing data and the threshold of 80 m between two consecutive years. First, linear interpolation for transects holding less than nine missing years was used in this study. This thesis's algorithms and analysis were not all compatible with time series that have missing values. To overcome the missing values problem, the missing values are filled by interpolating between existing data points. For missing values at the beginning or end of the time series, the closest data point was used for these positions. One could argue whether filling the missing values with non-existing data is legitimate for forecasting. However, if only using the transects with no missing values, the final dataset becomes relatively tiny ($\approx 37k$) and not evenly distributed over the world (Calkoen et al., 2021). By using a threshold of nine missing data points, the dataset has a size of $\approx 410k$ transects.

The second assumption is the 80 m threshold used in this study. The approach is purely data-driven and only based on the information given by the SDS data. Since some data points in the SDS data are influenced by human interventions or are incorrect data, the threshold mentioned above is used to filter out these transects. The choice of 80 m is based on SDS accuracy, natural shoreline behavior occurrence, and the data loss for a threshold. This threshold derivation was generated to derive an objective threshold. However, some assumptions are at the basis of this derivation and therefore could be researched more thoroughly.

Clustering

In this subsection, the clustering part of this thesis will be discussed. The main topic in this section will be; the representation used for clustering, the hydraulic and geomorphic features, the algorithm itself, and the number of clusters.
Considering the representation, the DWT main goal was to capture the long-term behavior and maintain a physical understanding of the transect representation. Using the DWT representation, the clustering was done based on the time series's shapes. The goal was to create clusters that would be used for long-term forecasting, the approximation coefficients of the second level DWT decomposition were used as clustering input. Here the first and second level detailed coefficients were not used for clustering. These were left out because they captured the transects' high-frequent behavior like beach rotations. This type of seasonal information is not relevant for clustering transects based on long-term behavior. However, dismissing a large amount of data can influence the results, therefore in future clustering, these series could also be used as input. Whether to use the raw time series as input or the detailed and approximation coefficients separately as input should be examined carefully.
Following the clusters' barycenters in figure 3.16, all the transects in each cluster follow a clear trend. What strikes when considering density plots of single clusters in appendix C.1, is the constant behavior in the first DWT step and the bottleneck between steps 2 and 3. These components are most likely clarified by the DWT approach because the DWT approach extrapolates the beginning and end of the time series before applying the DWT procedure. So, the first step of the DWT representation is (partially) derived from self-generated data. However, this first step was evenly handled by the clustering algorithm compared to other DWT steps and could have led to faulty clustering of transects. The (impact of the) DWT procedure could be examined more thoroughly in future research. The choice of the wavelet and decomposition level are of the most prominent importance. In this research, the results show clear patterns, and thus the approach is considered suitable.
Besides DWT, a more statistical approach could have used as well for clustering. Clustering already showed to improve forecasting as it extracts a broad set of features (Bandara et al., 2020). A disadvantage of using features for clustering is that it will diminish understanding of clustering. The clusters are based on a large set of features, and understanding what kind of transects are in each cluster demands an excellent understanding of the feature set. Next to that, the availability of features is enormous. Selecting the most optimal features for the final purpose of this thesis is complex, subjective, and time-consuming.
Next to the time series representation, an attempt and investigation were conducted to incorporate hydraulic and geomorphic features in the clustering algorithm. The correlation between the features and the transect's behavior was examined to determine whether the hydraulic and geomorphic features could add any value to the clustering of transects. The spearman correlations found here were too low for further use in later cluster steps. By using the spearman correlation, an assumption was that the relationship must be monotonic. However, relationships between geomorphic and hydraulic features and transects' behavior could be more complex and multi-dimensional. Also, the influence of natural factors can be variable over a wide temporal and spatial scales (Stive et al., 2002). Besides, the feature considered were tested for correlation one-on-one with a single

transect's feature. This is debatable, as the shoreline evolution is caused by numerous phenomena that involve an extensive set of features. Although, in tables 3.4 and 3.6 a significant correlation was found twice between the sum of absolute differences and the squared significant wave height. Unfortunately, the correlation was not useful globally, but the significant correlation tells that a higher resolution SDS could be compatible with open-source data in the future. This leads to the last point that the features considered are not complete. Other features could be used as well for dependency search, like storms or SLR (Wang, 2018).
The clustering algorithm TimeSeriesKmeans used in this thesis is a partitioning clustering algorithm based on the K-means algorithm. A major disadcantage of the clustering algorithm used is that it is highly dependent on the initial conditions. This also means that in every run, the clusters are different. This makes it challenging to use a final cluster and label a transect with a binding cluster. Another disadvantage is that the clustering algorithm requires the number of clusters as an input. Where other algorithms can determine the optimal number of clusters themselves, K-means requires additional analysis on the forehand to determine the optimal number of clusters. Since the variety in transect behavior is endless and the optimal number is subjective, the workflow's total performance depends on this number.
Moreover, the distance measure between the DWT representation was Dynamic Time Warping (DTW). The DTW distance measure was chosen based on performance in previous literature (Keogh & Pazzani, 2000) (Ann Ratanamahatana Eamonn Keogh, n.d.). However, the computation time for the DTW is sometimes factor 50 larger. In this research, this was still acceptable, but for live computing in an application, this could be problematic. The use of the euclidean distance measure should in that case be reconsidered.
The sub-clusters found during the clustering procedure were found by examining the main clusters. The final boundaries for the sub-cluster were based on iterating and visual inspection of the resulting sub-clusters. This thesis's sub-clusters show the promising of the cluster methodology that different trends can be detected on a sub-transect level. However, the 25th and 75th percentile choices per main cluster as a boundary for the sub-clusters are subjective and could differ per cluster.

Forecasting

The major assumption of this research is that the forecasting of transects would benefit from a global model. All the forecasting algorithms in this research are based on the same principle: considering multiple time series to forecast a single one. Frequently used univariate forecasting algorithms like ES, ARIMA and OLS are only considering a single time series while forecasting (Vousdoukas et al., 2020). These algorithms are popular for their robustness, ease of use, and good performance. In addition, considering the MASE score of all clusters in tables of Appendix C.2, the score was always higher than 1. This indicated that the naïve (seasonal) benchmark resulted in predictions with a lower error score than the considered algorithm. However, these univariate models are not used in this research since the assumption was that the forecasting would benefit by creating a global model.
Next to that, a more comprehensive set of forecasting models could be used, where different principles are combined. For instance, hybrid models are models that use the concept of equilibrium and showed promising results (Montaño et al., 2020). However, as indicated, the choice for the four algorithms was most convenient as previous research than could be used as a benchmark.
For forecasting, the raw SDS dataset that remains after the pre-processing steps, so using 33 data points from 1984 till 2016, was used as input. However, in the clustering steps, capturing the long-term behavior was the main target and resulted in a DWT representation. This approach and thought could have been used as well for the forecasting of the transects. For forecasting purposes, a transect representation that only represents the long-term behavior could have been used as input instead of the raw transects.

The algorithms used in this research are the same as used in previous literature (Calkoen et al., 2021). The availability of forecasting algorithms is enormous, and development is emerging rapidly. Improved algorithms could have been developed during this thesis, making the used ones already deprecated. Also, the mathematical structure and approach of the used algorithms were not examined thoroughly.
Furthermore, the hyper-parameters used for the algorithms are following current research (Calkoen et al., 2021). Whether these were optimal for shoreline forecasting with SDS data is not examined thoroughly. Varying with different lengths of training sets and windows could lead to other predictions (accuracy). However, research showed that the test set's length did not influence the models (Calkoen et al., 2021). This was also examined by determining the mean standard deviation of the differenced train and test set of a time series per cluster. For all clusters, the mean standard deviation for the train and test set was the same.
Besides, for the extreme and severe clusters the number of transects in the clusters is relatively small. A risk for using small datasets to train ML algorithms, is that there is not enough data to solve all equations and unknowns in the system. This could lead to trained algorithms with difficulties generalizing from trained data and will lead to poor forecasts. However, due to the rolling window principle in figure 2.13, multiple train and validation windows are generated per transect.
Last, the forecasting approach was purely focused on univariate time series forecasting and not considering other useful exogenous inputs. For example, the DeepSSM algorithm considers dynamic and static variables as input per time series to create better forecasts for time series that share the same static and dynamic variables.

Application

As already mentioned in the results, the application step shows the potential of using an open-source database to improve forecasts on a local scale. The GRIP database is used as it is globally available and relatively easy to work with. However, one could discuss whether a roads database is the best "physical-boundary" database to use. For example, considering the beach figure 4.1 presenting a beach in Miami, USA, multiple boundaries are closer to the shoreline than roads. This indicates that the erosive behavior will encounter other boundaries first than roads. Next to that, the height difference in the cross-shore profile is not considered in the post-processing. Often, erosion rates will decrease due to an increase in sediment more inland. This increase in sediment supply and/or height is not considered in the post-processing step. The boundaries described above and presented in figure 4.1 are available on a local scale and could be used for local predictions but are not available on a global scale. However, to make more realistic global predictions, global datasets with this information are necessary. For example, a global bathymetry database or a database with widths of sandy shorelines is better. Those datasets will improve forecasts on a local scale.

Figure 4.1: Miami beach showing the cross profile of the coastal zone. Before erosive behavior reaches roads it first encounters vegetation/small housing and pavements.

4.3 Future directions

In this section, three future directions and possibilities for shoreline forecasting will be sketched, considering that all of the aforementioned limitations are resolved.

Spatial and temporal resolution

An improvement in spatial and temporal resolution of the SDS data and publicly available information will enhance shoreline forecasting drastically. The current database of the Shoreline Monitor is derived from bi-weekly satellite data. When processing power is up to date and faster computational methods are developed, these bi-weekly data could be beneficial. A higher temporal resolution of the SDS data will increase understanding of high-frequent behavior but mainly improve understanding of long-term behavior. Since more data can be used to derive long-term behavior, this becomes more reliable and useful. Second, an increase of spatial resolution of the SDS data will help to create better local models. Herefore, more transects can be used for local beaches and local shoreline features are better understood. Last, when the spatial resolution of the open-source information used in Step 2. and Step 6. is improved, SDS data can be coupled to hydraulic / geomorphic / physical data. This will add to the knowledge of the influence of natural phenomena on shoreline behavior and examine the impact of human interventions on the coastal zone.

Gravel shorelines

This research focused on the forecasting of sandy shorelines globally. However, there is an extensive range of beach types found on earth. Mangrove and gravel shorelines are shorelines that also experience changes that can be measured via satellite data. Considering that 31% of the ice-free shorelines are sandy beaches, it leaves 69% of ice-free shorelines unconsidered in this thesis (Luijendijk et al., 2018). In the future, the sediment composition of these shorelines can be determined, which will result in a better understanding and possible forecasting of these not-sandy shorelines.

Quantitative risk assessment

The third direction is regarding the quantification of the impact of predictions. The impact of a possible prediction is currently presented qualitatively, but a risk assessment could lead to better decisions. With more reliable and more frequent predictions along the coast, the risk of these predictions on the hinterland could be expressed in a quantitative form. Quantifying could be in the form of financial risk and people exposed to the future shoreline location. By translating the predictions into measurable impact or risk, the decision-making with these predictions is eased.

Chapter 5

Conclusion

The combination of an increase in SLR and intense populated coastal zones demands long-term predictions of shoreline behavior. These predictions can be relevant for coastal managers and other stakeholders. Current shoreline forecasting is computationally expensive and only considers a particular set of variables to describe coastal phenomena. Growing and improving satellite data together with state-of-the-art ML algorithms could boost shoreline forecasting. Recently derived Satellite Derived Shoreline (SDS) data captures historical shoreline data on a large temporal and spatial scale. This dataset consists of shorelines with orthogonal transects every 500 m, on 33 years time scale. Current research already explored the opportunities to predict long-term sandy shoreline behavior with statistical and ML forecasting methods. However, a deepening in transect behavior to improve sandy shoreline forecasting is still undiscovered.

In an increasingly data-driven society, the amount and availability of open-source data are growing. These available data provide the opportunity to create a global approach for this thesis. With a global approach, cross-time series information is used so that forecasting individual transects could benefit from the information of the whole dataset. Clustering the SDS data could enhance knowledge of global and local transect behavior and result in tailor-made forecasting models. The main goal of this research is to improve shoreline forecasting by cluster with similar time series behavior, which is investigated following the main research question:
How can shoreline forecasting be improved by making a global model using a clustering approach and public data?

In order to answer the main research question, multiple sub-questions and a workflow were created, which will be answered below:

RQ 1: To what extent can transect clustering improve shoreline forecasting?
First, the concept of transect clustering was demonstrated and possible transect features for clustering were explored. For this, classes were generated based on mean change rates (m/year). Each class was used as an input for a forecasting algorithm. This time series approach resulted in meaningful insights into transect behavior and how different transects were predicted. Next to that, the correlation between geomorphic and hydraulic features and transect behavior was too low and therefore not useful for a global model and other clustering purposes. Hence, only features from the transect data were used for further clustering purposes, and geomorphic or hydraulic features were left out.

RQ 2: What methods to use for clustered forecasting and how does this improve shoreline forecasting?
Clustering based on long-term transect behavior resulted in nine clusters. These clusters provided information about the distribution of transect behavior on a global and regional level, ranging from extreme erosion to extreme accretion. For example, on a regional scale, the distribution of transects based on the clusters in the Baltic Sea corresponded with recently published research. Furthermore, clusters M.0, E.1, E.2, A.1 and A.2 were further examined at the transect level. By a newly developed sub-clustering workflow, beyond state-of-the-art transect clustering was enabled. This sub-cluster workflow resulted in three sub-clusters for each of the five clusters, where a distinction was made in accelerating, decelerating, and reversing trends of transects per cluster. These sub-clusters provided new insights into shoreline behavior globally, regional and local, and corresponded with recent literature findings. For example, the decelerating erosive trend in the Caspian Sea, due to change in sea level, was confirmed by the sub-cluster approach. Next to that, accelerating erosion along the coast in central Chile was found, corresponding to increasing erosion rates reported in recently published research. The main added value showed by the examples is that the (sub-)clustering approach enables to quantify behavior on different spatial scales based on data of the Shoreline Monitor. Primarily, additional insights into the state of shoreline evolution were found. Secondly, previous research about shoreline behavior was coupled with (sub-)cluster information, strengthening these previous findings. Finally, new and reversing trends in shoreline behavior were detected.
After the clustering step, the nine main clusters were used as an input for forecasting the transects till 2030. Here, four different algorithms were used for forecasting, with Ordinary Least Squares (OLS) as a benchmark. By using all the SDS data per cluster to train and evaluate the four forecasting algorithms, the overall forecasting accuracy (in MSE and MAE) greatly improved for >95% of the total SDS dataset, sometimes up to factor 15. Besides, the best overall forecast algorithm differed significantly per cluster, indicating transects forecasting flourish with tailor-made algorithms. Finally, the predictions on a single transect level gained improved reliability due to a developed multi-method approach. The most reliable local prediction was chosen by weighing overall accuracy improvement and transect knowledge gathered while clustering.

RQ 3: How can open-source information be used to make a more reliable local forecast?
In the end, the predictions' application and reliability are checked using local maps and the GRIP database. This approach led to a qualitative and quantitative high-resolution vulnerability assessment where the reliability of predictions was be reviewed. By this approach, the possible year of interference of predictions with local roads was found. Furthermore, third parties could use the generated prediction and the GRIP database to act on shoreline behavior and make sustainable long-term decisions regarding the coastal zone.

All of described above contributed to improving shoreline forecasting on a global and local scale. Coastal managers and other parties can consult predictions of 346,194 sandy transects globally. Per transect, the predictions are used to review how it affects the respective stretch of coast. Furthermore, the (sub-)clustering approach provides a more in-depth understanding of changing shoreline behavior, leading to more reliable predictions. However, all this was eventually achieved using a single feature as input for clustering and specific forecasting algorithms. As deliberated in Chapter 4, more and different (public available) features could be used for clustering together with other forecasting algorithms. Data-driven prediction of shorelines on a global scale is still in its infancy and must be further researched to improve its accuracy and reliability. By using its potential, data-driven predictions can become leading for choices and insights for sustainable coastal management.

Chapter 6

Recommendations

Finally, after the results are discussed, and the conclusions are drawn, some concrete recommendations are provided for future research. A set of recommendations is formulated below to improve successive research on shoreline forecasting with SDS data.

1. Increase the temporal and spatial resolution of the SDS data behind the ShorelineMonitor. Improving the temporal scale by using the bi-weekly data, extra information is added to the system. Furthermore, lowering the spatial scale will increase understanding on a local scale. Most of the limitations in Chapter 4 will be resolved by improving the SDS dataset behind the ShorelineMonitor.

2. The pre-processing steps could be reviewed, as the input for a model has an immense impact on the final output. First, the period used in this research is from 1984 till 2016, containing 33 data points per time series. The maximum number of NAN values is only considered on an individual time series level with a maximum number of nine values per time series. However, when considering the whole dataset, the first years contain a relatively large amount of NAN values. The number of NAN values becomes constant and relatively low in 1999. One could investigate the response of the workflow by only using data from 1999 and later. Next, the threshold value and approach described in appendix B.1 could be further explored.

3. In step 2 of the diagram presented in figure 3.1 in Chapter 3, an attempt was made to add hydraulic and geomorphic information to the system. This could be done again with data with a lower temporal and spatial scale, as described above. The characteristics could follow the one described by Wang (2018). Besides more and different features, the investigation should be done with a more advanced clustering approach instead of monotonic correlation research. Next to the clustering, extra information could be added as well while forecasting the transects. For instance, coupling the date of human interventions with the relevant transects, the system could better predict shoreline response to human interventions. Furthermore, incorporating geomorphic and hydraulic (static or dynamic) characteristics in the forecasting algorithms would create a multivariate forecasting system. The forecasting algorithms could identify correlations between shoreline behavior and those exogenous variables.

4. The clustering representation is a choice that has a significant influence on the final forecasting performance. The reasoning and argumentation of the representation in this research are discussed in Chapter 3. However, as already mentioned in Chapter 4, the representation possibilities are endless. The main recommendation here would be to investigate more into feature-based clustering (Bandara et al., 2020), investigate other wavelets for the DWT and go more in-depth in the signal extension modes for finite time series.

5. Related to the recommendation above, the forecasting representation is also an area for further research. For example, using a moving average filter or using a lower level DWT decomposition, only low-frequent behavior will be considered for predicting long-term shoreline behavior.

6. For this research, a specific set of ML algorithms was selected. However, other models can be considered for both clustering and forecasting purposes. In the case of unlimited processing power, other clustering algorithms like hierarchical clustering could be used as well. The hierarchical clustering could be used with a DTW distance matrix with a size of N x N, where N is the total size of the SDS dataset. Next to clustering algorithms, the AI field is a fast-developing field of science, so new and better forecasting models will be available in the future. Testing and using these new models on the transect forecasting use case is necessary and will probably improve predictions as well.

7. Next to ML algorithms for forecasting, statistical forecasting models proved their ability to create predictions with high accuracy. Some (type of) transects probably do not benefit from a global model approach and will be best predicted with statistical models. In the case of unlimited processing power, statistical models should be incorporated as well to review predictions of those.

8. Couple predictions with (financial) impact models. The predictions made in this research are already projected back on their geographic location to create a more realistic forecast. However, these projections are only descriptive and not coupling any quantitative risk to possible behavior. Therefore it is recommended that future research combines the forecasting models with models that can assess the quantitative risk of the predictions.

9. Lastly, as already pointed out in the Discussion, this research focused on the sandy shorelines. The remaining 69% of ice-free unsandy shoreline is, therefore, an exciting area for future research. New sediment composition filters could be created in order to determine other sediment types. These beaches and shorelines can then be used to forecast and create a more comprehensive knowledge of global shoreline behavior.

In case of questions and/or remarks about the methods, results, discussion, conclusions, or recommendations, feel free to reach out: bramholland23@gmail.com.

References

Aghabozorgi, S., Seyed Shirkhorshidi, A., & Ying Wah, T. (2015, 5). Time-series clustering - A decade review. *Information Systems*, *53*, 16–38. doi: 10.1016/j.is.2015.04.007

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., ... Wang, Y. (2020). *GluonTS: Probabilistic and Neural Time Series Modeling in Python* (Vol. 21; Tech. Rep.). Retrieved from `https://github.com/awslabs/gluon-ts`

Ann Ratanamahatana Eamonn Keogh, C. (n.d.). *Making Time-series Classification More Accurate Using Learned Constraints* (Tech. Rep.).

Arthur, D., & Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. In *Soda '07: Proceedings of the eighteenth annual acm-siam symposium on discrete algorithms* (pp. 1027–1035). Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Athanasiou, P., Van Dongeren, A., Giardino, A., Vousdoukas, M., Gaytan-Aguilar, S., & Ranasinghe, R. (2019, 10). Global distribution of nearshore slopes with implications for coastal retreat. *Earth System Science Data*, *11*(4), 1515–1529. doi: 10.5194/essd-11-1515-2019

Bandara, K., Bergmeir, C., & Smyl, S. (2020, 2). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, *140*. doi: 10.1016/j.eswa.2019.112896

Benidis, K., Rangapuram, S. S., Flunkert, V., Wang, B., Maddix, D., Turkmen, C., ... others (2020). Neural forecasting: Introduction and literature overview. *arXiv preprint arXiv:2004.10240*.

Boak, E. H., & Turner, I. L. (2005, 7). Shoreline Definition and Detection: A Review. *Journal of Coastal Research*, *214*, 688–703. Retrieved from `https://doi.org/10.2112/03-0071.1` doi: 10.2112/03-0071.1

Bosboom, J., & Stive, M. J. F. (2015). Coastal Dynamic I: Lecture Notes CIE4305. *Delft, The Netherlands: Delft University of Technology, 573p*.

Calkoen, F., Luijendijk, A., Rivero, C. R., Kras, E., & Baart, F. (2021, 3). Traditional vs. Machine-Learning Methods for Forecasting Sandy Shoreline Evolution Using Historic Satellite-Derived Shorelines. *Remote Sensing*, *13*(5), 934. Retrieved from `https://www.mdpi.com/2072-4292/13/5/934` doi: 10.3390/rs13050934

Chen, J. L., Pekker, T., Wilson, C. R., Tapley, B. D., Kostianoy, A. G., Cretaux, J. F., & Safarov, E. S. (2017, 7). Long-term Caspian Sea level change. *Geophysical Research Letters*, *44*(13), 6993–7001. doi: 10.1002/2017GL073958

Davidson, M. A., Lewis, R. P., & Turner, I. L. (2010, 6). Forecasting seasonal to multi-year shoreline change. *Coastal Engineering*, *57*(6), 620–629. Retrieved from `https://doi.org/10.1016/j.coastaleng.2010.02.001` doi: 10.1016/j.coastaleng.2010.02.001

De Gooijer, J. G., & Hyndman, R. J. (2006). 25 years of time series forecasting. *International Journal of Forecasting*, *22*(3), 443–473. doi: 10.1016/j.ijforecast.2006.01.001

Dube, K., Nhamo, G., & Chikodzi, D. (2021, 3). Rising sea level and its implications on coastal tourism development in Cape Town, South Africa. *Journal of Outdoor Recreation and Tourism*, *33*. doi: 10.1016/j.jort.2020.100346

Fairley, I., Lewis, M., Robertson, B., Hemer, M., Masters, I., Horrillo-Caraballo, J., ... Reeve, D. E. (2020, 3). A classification system for global wave energy resources based on multivariate clustering. *Applied Energy*, *262*. doi: 10.1016/j.apenergy.2020.114515

Gneiting, T., & Raftery, A. E. (2007, 3). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, *102*(477), 359–378. doi: 10.1198/016214506000001437

Goldstein, E. B., Coco, G., & Plant, N. G. (2019, 7). *A review of machine learning applications to coastal sediment transport and morphodynamics* (Vol. 194). Elsevier B.V. doi: 10.1016/j.earscirev.2019.04.022

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017, 12). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, *202*, 18–27. Retrieved from `https://doi.org/10.1016/j.rse.2017.06.031` doi: 10.1016/j.rse.2017.06.031

Hagenaars, G., de Vries, S., Luijendijk, A. P., de Boer, W. P., & Reniers, A. J. (2018, 3). On the accuracy of automated shoreline detection derived from satellite imagery: A case study of the sand motor mega-scale nourishment. *Coastal Engineering*, *133*, 113–125. doi: 10.1016/j.coastaleng.2017.12.011

Hagenaars, G., Luijendijk, A., de Vries, S., & de Boer, W. (2017). Long term coastline monitoring derived from satellite imagery. In T. Aagaard, R. Deigaard, & D. Fuhrman (Eds.), *Proceedings of coastal dynamics 2017* (pp. 1551–1562). Retrieved from `http://coastaldynamics2017.dk`

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., ... Thépaut, J. N. (2020, 7). The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, *146*(730), 1999–2049. doi: 10.1002/qj.3803

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Huang, Y., & Jin, P. (2018, 8). Impact of human interventions on coastal and marine geological hazards: a review. *Bulletin of Engineering Geology and the Environment*, *77*(3), 1081–1090. doi: 10.1007/s10064-017-1089-1

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Hyndman, R. J., & Koehler, A. B. (2006, 10). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688. Retrieved from `https://doi.org/10.1016/j.ijforecast.2006.03.001` doi: 10.1016/j.ijforecast.2006.03.001

Hyndman, R. J., Wang, E., & Laptev, N. (2016, 1). Large-Scale Unusual Time Series Detection. In *Proceedings - 15th ieee international conference on data mining workshop, icdmw 2015* (pp. 1616–1619). Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ICDMW.2015.104

Javed, A., Lee, B. S., & Rizzo, D. M. (2020, 4). *A benchmark study on time series clustering.* arXiv. doi: 10.1016/j.mlwa.2020.100001

Jordahl, K. (2014). GeoPandas: Python tools for geographic data. *URL: https://github. com/geopandas/geopandas*.

Keogh, E. J., & Pazzani, M. J. (2000). Scaling up Dynamic Time Warping for Datamining Applications. In *Proceedings of the sixth acm sigkdd international conference on knowledge discovery and data mining* (pp. 285–289). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/347090.347153 doi: 10.1145/347090.347153

Kras, E. C. (2019). *Planetary-scale classiication of natural and human-induced sandy shoreline evolution A semi-automated method that employs Machine Learning and Satellite Derived Shorelines over the past decades* (Tech. Rep.). Retrieved from http://repository.tudelft.nl/.

Lakhan, V. C., & Pepper, D. A. (1997). Relationship between Concavity and Convexity of a Coast and Erosion and Accretion Patterns. *Journal of Coastal Research*, *13*(1), 226–232. Retrieved from http://www.jstor.org/stable/4298609

Lee, G., Gommers, R., Waselewski, F., Wohlfahrt, K., & O'Leary, A. (2019, 4). PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software*, *4*(36), 1237. doi: 10.21105/joss.01237

Long, J. W., & Plant, N. G. (2012, 7). Extended Kalman Filter framework for forecasting shoreline evolution. *Geophysical Research Letters*, *39*(13). doi: 10.1029/2012GL052180

Luijendijk, A., Hagenaars, G., Ranasinghe, R., Baart, F., Donchyts, G., & Aarninkhof, S. (2018, 12). The State of the World's Beaches. *Scientific Reports*, *8*(1). doi: 10.1038/s41598-018-24630-6

Martínez, C., Contreras-López, M., Winckler, P., Hidalgo, H., Godoy, E., & Agredano, R. (2018, 4). Coastal erosion in central Chile: A new hazard? *Ocean and Coastal Management*, *156*, 141–155. doi: 10.1016/j.ocecoaman.2017.07.011

Martínez, M. L., Intralawan, A., Vázquez, G., Pérez-Maqueo, O., Sutton, P., & Landgrave, R. (2007, 8). The coasts of our world: Ecological, economic and social importance. *Ecological Economics*, *63*(2-3), 254–272. doi: 10.1016/j.ecolecon.2006.10.022

Matheson, J. E., & Winkler, R. L. (1976). SCORING RULES FOR CONTINUOUS PROBABILITY DISTRIBUTIONS. *Management Science*, *22*(10), 1087–1096. doi: 10.1287/mnsc.22.10.1087

Meert, W., Hendrickx, K., & Craenendonck, T. V. (2020, 8). *wannesm/dtaidistance v2.0.0.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.3981067 doi: 10.5281/zenodo.3981067

Meijer, J. R., Huijbregts, M. A., Schotten, K. C., & Schipper, A. M. (2018, 6). Global patterns of current and future road infrastructure. *Environmental Research Letters*, *13*(6). doi: 10.1088/1748-9326/aabd42

Michael A. Nielsen. (2015). Neural Networks and Deep Learning. *Determination Press*.

Montaño, J., Coco, G., Antolínez, J. A., Beuzen, T., Bryan, K. R., Cagigal, L., ... Vos, K. (2020, 12). Blind testing of shoreline evolution models. *Scientific Reports*, *10*(1). doi: 10.1038/s41598-020-59018-y

Neumann, B., Vafeidis, A. T., Zimmermann, J., & Nicholls, R. J. (2015, 3). Future coastal population growth and exposure to sea-level rise and coastal flooding - A global assessment. *PLoS ONE*, *10*(3). doi: 10.1371/journal.pone.0118571

Nicholls, R. J., & Cazenave, A. (2010, 6). Sea-Level Rise and Its Impact on Coastal Zones. *Science*, *328*(5985), 1517–1520. Retrieved from https://doi.org/10.1126/science.1185782 doi: 10.1126/science.1185782

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

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

Petitjean, F., Ketterlin, A., & Gançarski, P. (2011, 3). A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, *44*(3), 678–693. doi: 10.1016/j.patcog.2010.09.013

Ranasinghe, R., McLoughlin, R., Short, A., & Symonds, G. (2004, 3). The Southern Oscillation Index, wave climate, and beach rotation. *Marine Geology*, *204*(3-4), 273–287. doi: 10.1016/S0025-3227(04)00002-7

Rangapuram, S. S., Seeger, M. W., Gasthaus, J., Stella, L., Wang, Y., & Januschowski, T. (2018). Deep State Space Models for Time Series Forecasting. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 31). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2018/file/5cf68969fb67aa6082363a6d4e6468e2-Paper.pdf

Reguero, B. G., Losada, I. J., & Méndez, F. J. (2015, 6). A global wave power resource and its seasonal, interannual and long-term variability. *Applied Energy*, *148*, 366–380. doi: 10.1016/j.apenergy.2015.03.114

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020, 7). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, *36*(3), 1181–1191. doi: 10.1016/j.ijforecast.2019.07.001

Siegle, E., Edvin Asp, N., & Leandro Ribeiro, A. (2007, 6). Wave refraction and longshore transport patterns along the southern Santa Catarina coast. *Brazilian Journal of Oceanography*, *55*(2), 109–120.

Stive, M. J. F., Aarninkhof, S. G. J., Hamm, L., Hanson, H., Larson, M., Wijnberg, K. M., ... Capobianco, M. (2002). Variability of shore and shoreline evolution. *Coastal Engineering*, *47*(2), 211–235. Retrieved from http://www.sciencedirect.com/science/article/pii/S0378383902001266 doi: https://doi.org/10.1016/S0378-3839(02)00126-6

Tavenard, R., Faouzi, J., Vandewiele, G., Divo, F., Androz, G., Holtz, C., ... Woods, E. (2020). Tslearn, A Machine Learning Toolkit for Time Series Data. *Journal of Machine Learning Research*, *21*(118), 1–6. Retrieved from http://jmlr.org/papers/v21/20-091.html

Vousdoukas, M. I., Ranasinghe, R., Mentaschi, L., Plomaritis, T. A., Athanasiou, P., Luijendijk, A., & Feyen, L. (2020, 3). *Sandy coastlines under threat of erosion* (Vol. 10) (No. 3). Nature Research. doi: 10.1038/s41558-020-0697-0

Wang, N. (2018, 7). *Unravelling the Sandy Shorelines Dynamics Derived from Satellite Images* (Tech. Rep.). Delft: TU Delft.

Weisse, R., Dailidiene, I., Hünicke, B., Kahma, K., Madsen, K., Omstedt, A., ... Zorita, E. (2021). Sea Level Dynamics and Coastal Erosion in the Baltic Sea Region. *Earth System Dynamics Discussions*, 1–40. doi: 10.5194/esd-2021-6

Wen, R., Torkkola, K., Narayanaswamy, B., & Madeka, D. (2017, 11). A Multi-Horizon Quantile Recurrent Forecaster. Retrieved from `http://arxiv.org/abs/1711.11053`

Zolhavarieh, S., Aghabozorgi, S., & Teh, Y. W. (2014). *A review of subsequence time series clustering* (Vol. 2014). Hindawi Publishing Corporation. doi: 10.1155/2014/312521

Appendices

Appendix A

Background information

A.1 Global Box system

As described in chapter 2 the transect system contains 2.2 million transects on a global scale. However, when working with the entire set of transects in Python, this will lead to high computational times. That is why all the transects are split and labeled by their location on earth, as shown in figure A.1, resulting in 330 boxes. 130 of the 330 boxes consist only of land or sea and are therefore not relevant for this research. For the other 210 boxes, 74 boxes do not contain valuable transects, as data is unavailable or shorelines are covered in ice. In the end, there are 126 boxes (green boxes in figure A.1 left with the mentioned 2.2 million transects. Every transects finally ends with a label of the form BOX_xxx_xxx_xxx. The first xxx is the box where the transects are in, the second xxx is the subbox where the transects is in, and the third xxx is the alongshore position of the transect within a subbox.

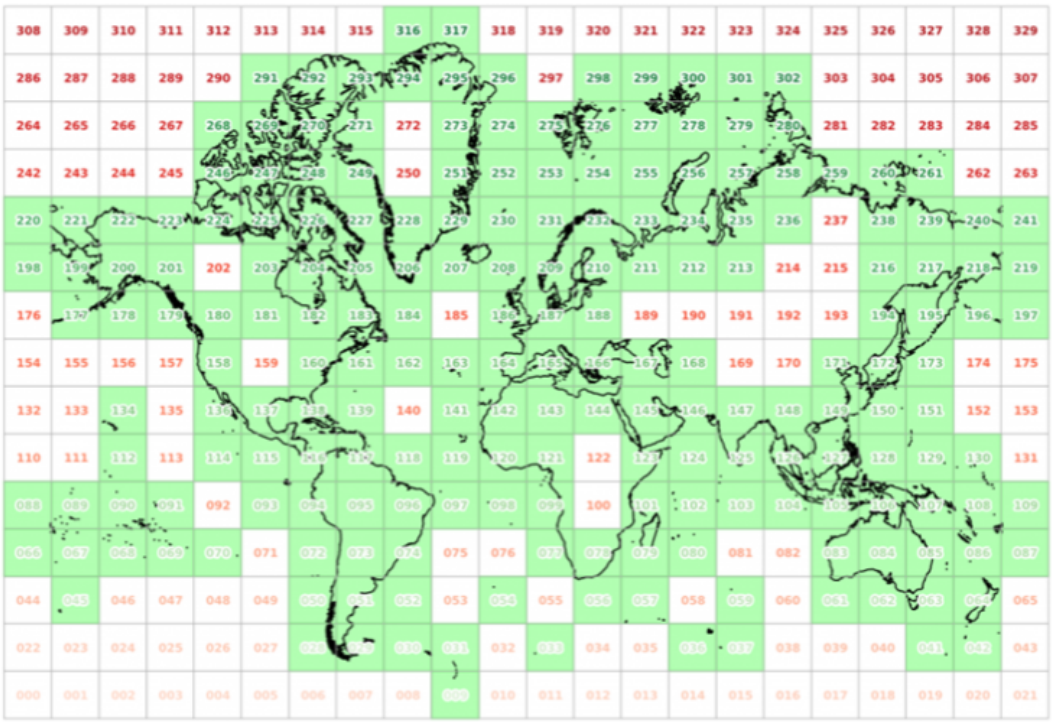

Figure A.1: Gridded world with boxes related to the IDs of transects in the Shoreline Monitor

Appendix B

Methodology

B.1 Pre-processing

As mentioned, the final dataset provided by (Luijendijk et al., 2018) contains around 1.8 million transects. However, these transects also include non-sandy transects. The first step is to drop all non-sandy transects and transects with the label $no\_sedcomp$, meaning that determining the shoreline sediment composition was not possible. Also, transects with errors in the change rate are deleted and without any longitude and latitude coordinates.
Next to that, all outliers defined by Luijendijk et al. (2018) and Kras (2019) were removed.
Finally, all transects missing more than nine observations (> 25%), nine missing years, will be neglected. This because otherwise, the data contains too many missing values to be valuable for any forecasting. Only considering the transects without missing values would lead to a dataset of about 37,000 transects, only at specific locations in the world. Transects with missing data (but less than nine missing data points) will be modified by linear interpolation. Missing data points are filled by linear interpolation. In figure B.1 an example is sketched of the procedure of filling these missing data points. The total dataset after the steps above contains about 410,000 transects globally distributed, as shown in figure B.2. When the missing values are at the beginning (1984) or the end (2016), the closest value will be used for these points.

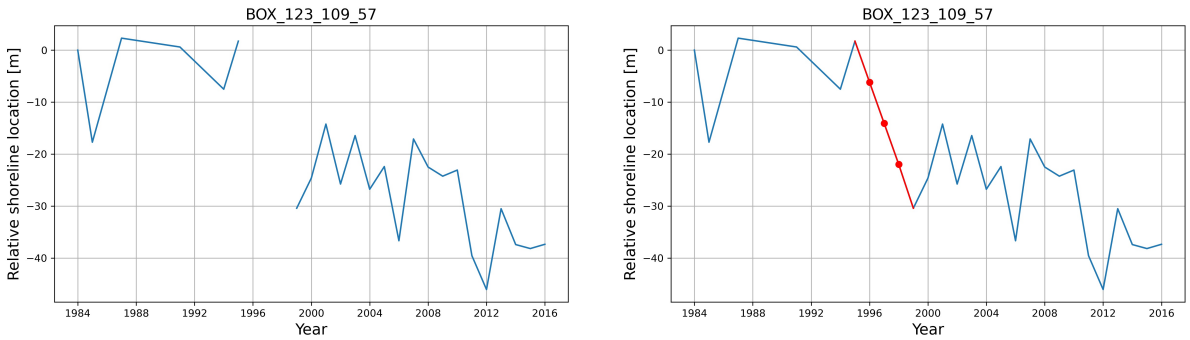

Figure B.1: Linear interpolating missing data points in SDS data. Left graph shows missing data from 1995 till 1999 for transect BOX_123_109_57. The right graph shows the red line filling the missing data points for year 1996, 1997 and 1998 by linear interpolating between two observed data points.

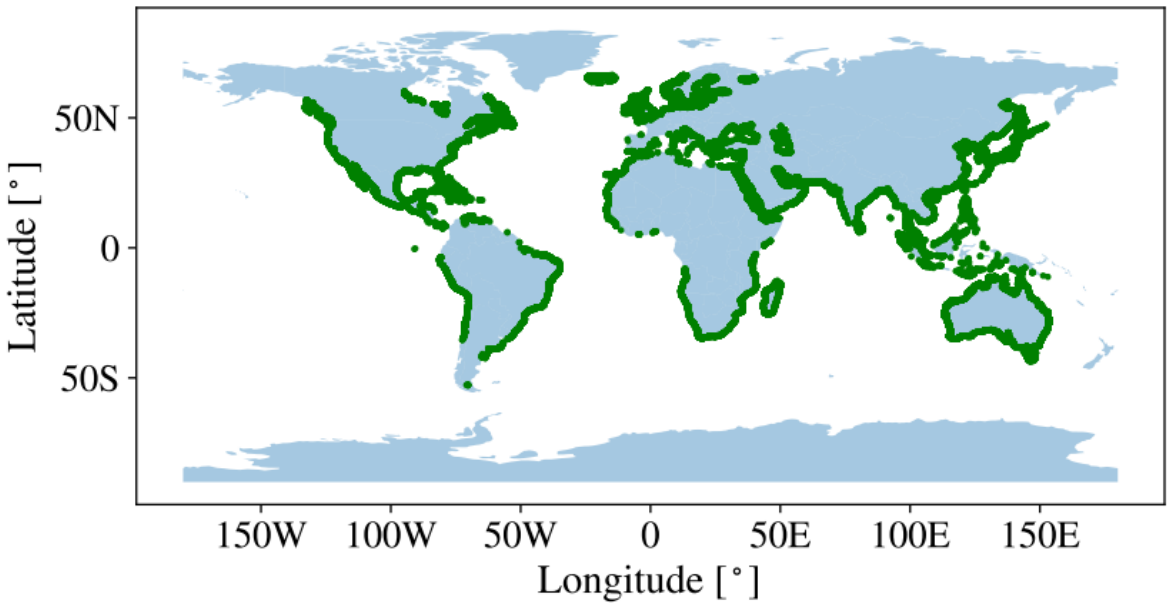

Figure B.2: Global distribution of transects considered in this research.

B.2 Anomaly detection in time series

Before the SDS transects are used for analysis, the SDS transects data were examined critically on anomalies. Anomalies are shoreline behavior that is rare to happen without human interventions and could therefore be considered as an anomaly. Sometimes large steps between consecutive years are detected. These steps can be faulty (e.g., wrong SDS derivation) or can be caused by (human) interventions (e.g., port expansion or land reclamation). In figures B.3 and B.4 a transect in Rio de Janeiro (BOX_074_110_11) and a transect in China (BOX_149_292_21) are plotted. For both cases, a large step between two consecutive years is detected. There is no clear explanation in the Rio de Janeiro case, and it is possibly caused by measure- or derivation errors. However, for the China case, a big port expansion is the cause of the large jump of the transects coordinates.
Since the study aims to use only univariate time series and no exogenous variables (like the date of the port expansion in China), these kinds of transects should be removed. A threshold should be defined as the maximum allowable step size between consecutive years to manage these irregularities.

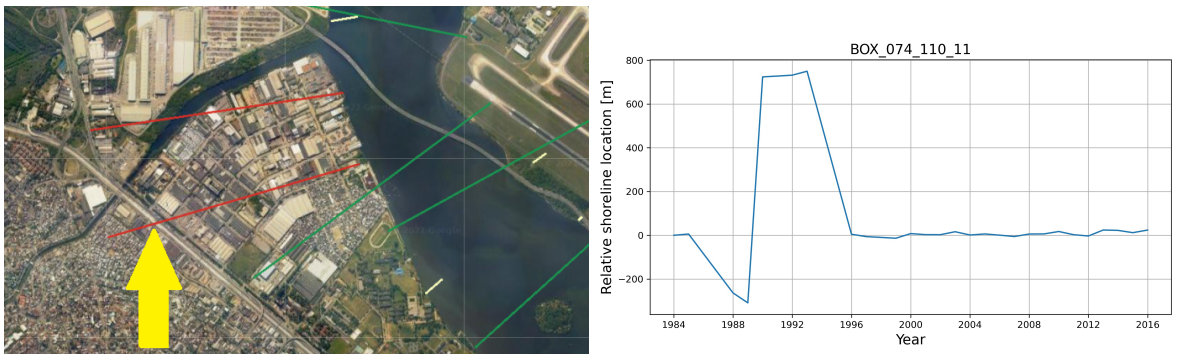

Figure B.3: Faulty data in transect BOX_074_110_11 in Rio de Janeiro. The graph on the right shows an abrupt change and should be formed by measuring faults

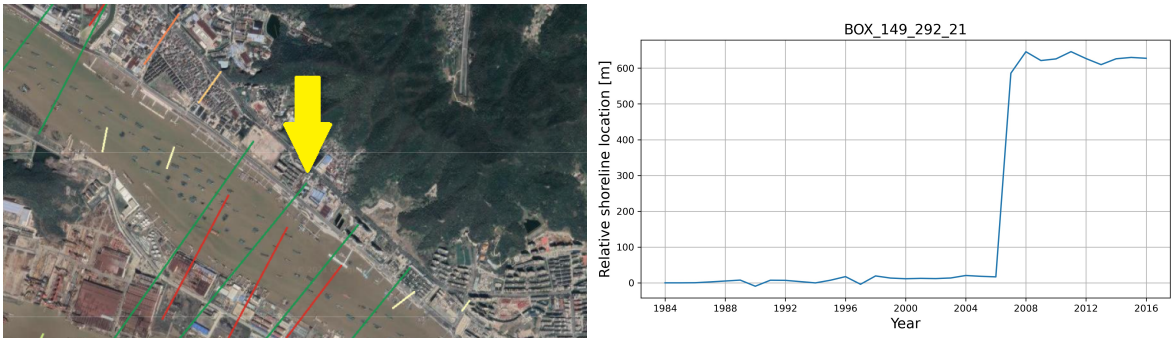

Figure B.4: Transect BOX_149_292_21 in China showing extreme accretion. Caused by the expansion of the port of Ningbo-Zhoushan.

For defining the threshold, three different things should be considered: the physical and natural plausibility that a step could occur (without external interventions), the SDS subpixel precision of 15-30 meter, and last the percentage of transects removing of the raw set for a given threshold value.

The hydraulic and geomorphic probability of a large step between consecutive years should be examined. By carefully examining the SDS data, constant change rates approaching ±40 m/year occur. These transects show constant erosive and accretion behavior and seem the cause of natural processes and phenomena. For example, in the south of Vietnam, where erosion rates of almost -40 m/year occur and in the Philipinnes, and Angola constant change rate of almost +40 m/year are detected. Next to these long-term change rates, the change rates between two consecutive years can be relatively high due to single events like storms and sand bars arriving at shorelines. A 50 m threshold is chosen to account for the maximum step between consecutive years as a physical boundary. This 50 m is a maximum threshold for naturally occurring shoreline changes. Next to this physical criteria, the accuracy of the yearly SDS transects should be considered. This accuracy lies within subpixel size, 15-30 m (Hagenaars et al., 2018). Inaccuracy is caused by wave-induced foam and cloud coverage, in the order of 40 m and 200 m, respectively. This kind of shoreline measurement inaccuracies is decreased by a moving average (Hagenaars et al., 2018). Thus, considering the physical change rate occurrence and the SDS accuracy, the maximum allowable step size will be 80 m (50 m + 30 m) between two consecutive years. Lastly, next to the physical feasibility and SDS

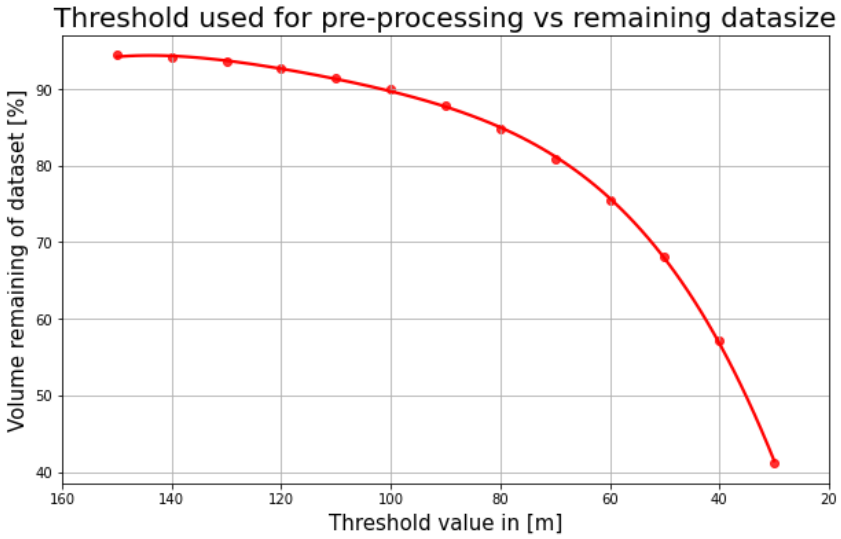

Figure B.5: Polynomial fit on dataset size after threshold value. Generated on a representative dataset of 50k transects.

accuracy, the impact on the dataset should be examined. With the threshold, the dataset will reduce in size. In the graph in figure B.5 the threshold value in meters is plotted against the remaining size of the dataset in percentage. The "bending-point" of this graph lies between the 80-60 m threshold value. After this zone, the reduction in dataset size starts to be exponential. The 80 m threshold is reasonable, as it only reduces the total dataset by 15.5%, leaving a total SDS dataset of about 346,000 transects.
To conclude, by eventually using a threshold value of 80 m, the physical probability of such a step occurring is considered together with the SDS accuracy and dataset reduction.
To enforce the conclusion of using the 80 m threshold, a more quantitative approach is also possible. In the graph in figure B.6 the threshold is plotted against the data size (left y-axis) but also versus the MSE (right y-axis). The MSE is derived by starting with a raw dataset of 5k transects and for every threshold, the remaining transects were used as input for SimpleLSTM algorithm. The blue line is the polynomial fit for all the MSE scores per threshold. By using a more quantitative approach, the 80 m threshold seems most optimal. With the 80 m threshold, the maximum MSE reduction was gained with a minimum dataset reduction. More visually explained, the space between the blue en red line is largest in the area for the 80 m threshold.
After the "large steps" are detected and the time series with these steps are deleted with the described threshold, a linear regression on the whole dataset is done. The linear regression will lead to the slope and intercept of the linear fit. The linear slope indicates the change rate/trend of the time series, and the intercept enables to scale of the time series around 0. All time series are subtracted with linear intercepts, and this eliminates the location in 1984 relative to the 2016 shoreline from OSM.

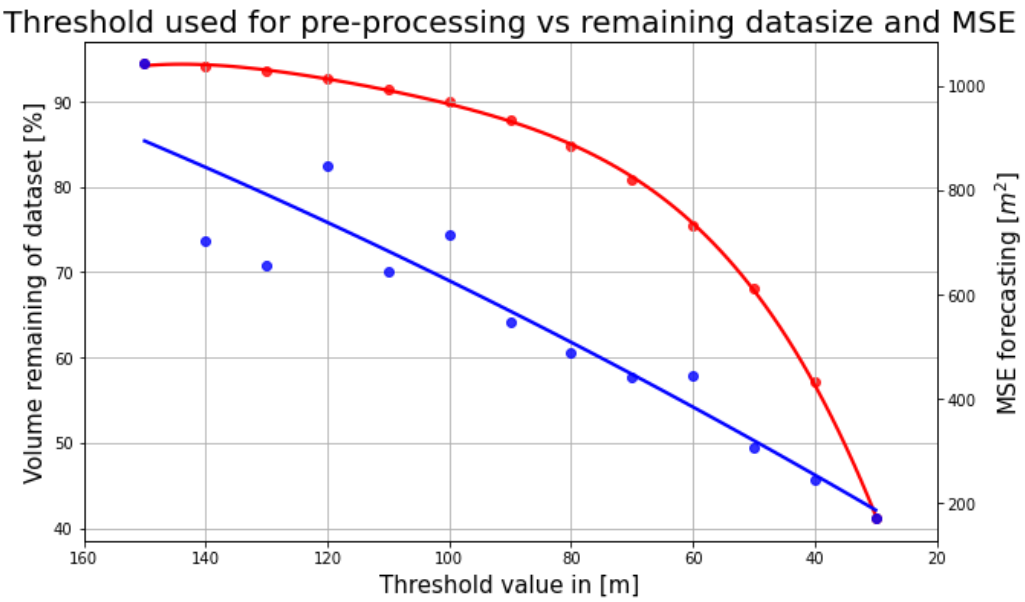

Figure B.6: Threshold value plotted against the volume size and Mean Squared Error (MSE) of SimpleLSTM.

B.3 Coastal zones by Fairley et al., (2020)

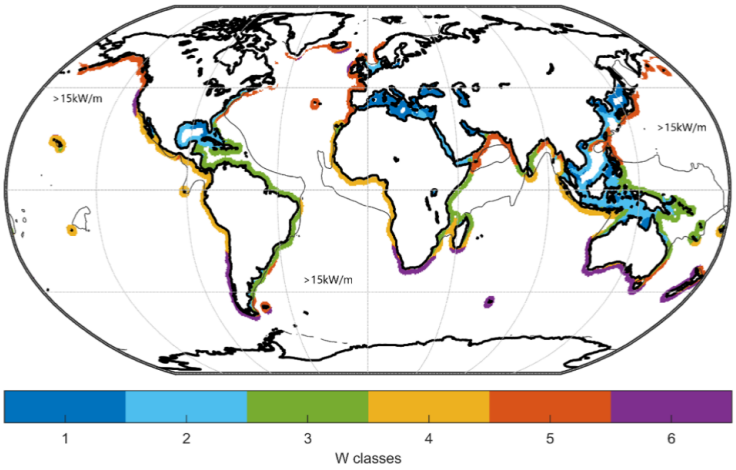

Figure B.7: The geographic spread of the 6 classes for the W classification. The grey contour indicates the commonly considered 15 kW/m limit for viability of WEC deployment. (Fairley et al., 2020)

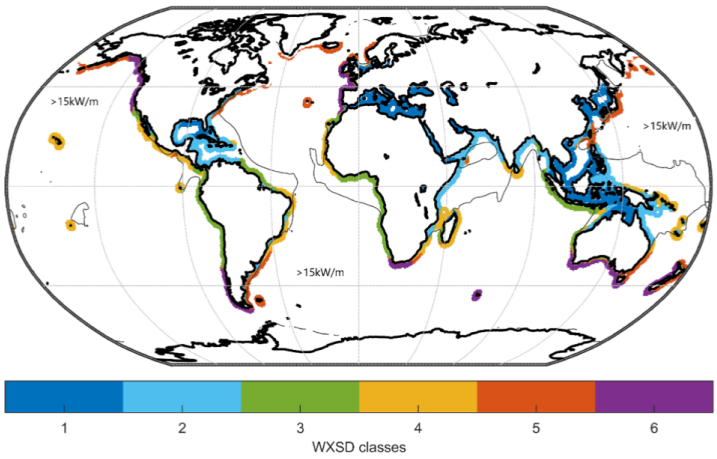

Figure B.8: The geographic spread of the 6 classes for the WXSD classification. The grey contour indicates the commonly considered 15 kW/m limit for viability of WEC deployment.(Fairley et al., 2020)

	Class	Wave parameters (W)				Risk (X)		Spectra (S)		Directionality (D)		
		$\tilde{H_s^2}$	$\sigma_{H_s^2}/\tilde{H_s^2}$	$\tilde{T_p}$	$\sigma_{T_p}/\tilde{T_p}$	H_{s0}	$H_{s0}/\tilde{H_s}$	$\tilde{Q_p}$	σ_{Q_p}	$\sigma_{dir.}$	$\tilde{wdw}$	σ_{wdw}
W	1	0.84	0.69	5.19	0.30	7.02	8.29	4.34	1.13	62.76	34.75	7.34
	2	0.89	0.52	6.19	0.31	5.51	6.10	3.95	1.13	54.51	32.96	6.21
	3	1.46	0.33	8.79	0.24	5.90	4.07	3.61	1.06	35.02	33.93	5.53
	4	1.72	0.26	12.22	0.22	5.87	3.41	3.90	1.44	26.27	32.98	6.10
	5	2.14	0.51	9.55	0.25	13.17	6.13	3.90	1.16	54.29	35.63	6.80
	6	3.03	0.34	11.91	0.19	13.09	4.31	4.02	1.17	36.40	34.09	6.60
WXSD	1	0.86	0.62	5.52	0.31	6.36	7.39	4.20	1.13	61.91	34.22	7.02
	2	1.26	0.42	8.26	0.25	6.19	4.96	3.72	1.10	36.61	31.11	4.98
	3	1.73	0.26	12.06	0.20	5.94	3.41	4.04	1.42	17.07	28.26	4.82
	4	1.73	0.27	10.75	0.24	6.13	3.55	3.55	1.23	38.55	39.23	7.09
	5	2.30	0.49	9.23	0.25	13.89	6.05	3.90	1.13	63.27	39.01	7.94
	6	2.93	0.39	11.67	0.21	14.01	4.87	4.06	1.18	35.12	32.70	6.25

Figure B.9: Characteristics of classes W and WXSD (Fairley et al., 2020)

Appendix C

Results

In this appendix, the remainder of the results, described in Chapter 3, are presented. The chapter can be divided into two sections where the cluster and forecasting results are presented. The results in this appendix are not in-depth analysed as in Chapter 3, but could be interpreted and analysed following the same approach as in Chapter 3.

C.1 Cluster results

Here the results regarding clustering transects are presented. The figures mainly follow the analysis in the main report. Since not all clusters are described in the main report, figures are presented to analyse the other clusters.

C.1.1 Subset for clustering

The subset for training the clustering algorithm is presented in this subsection.

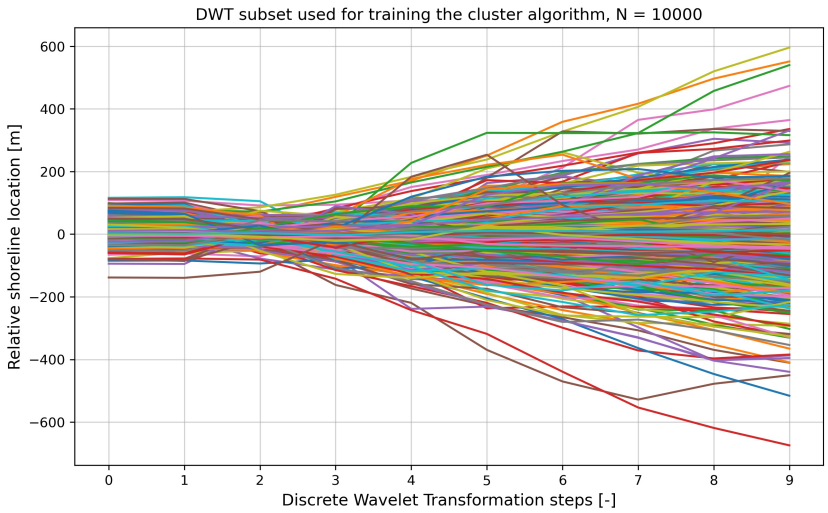

Figure C.1: Subset used for training the clustering algorithm.

C.1.2 Optimal number of clusters

The different barycenters for various number of clusters are presented here. The graphs were used to determine the optimal number of clusters as described in Section 3.5.1.

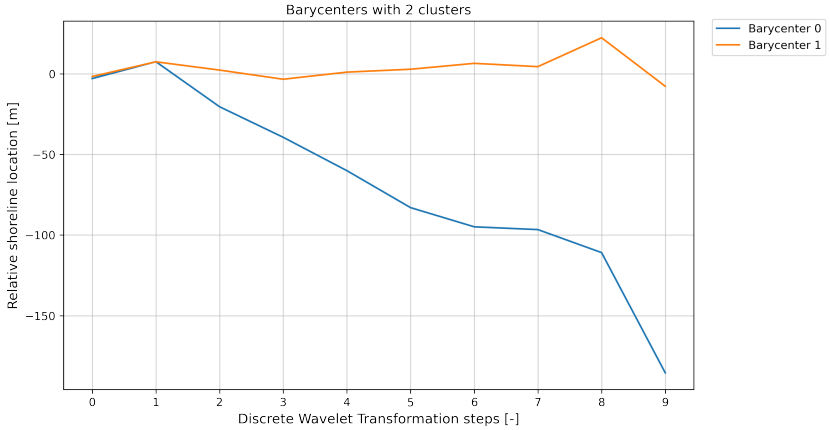

Figure C.2: Barycenters for two clusters, generated by TimeSeriesKmeans.

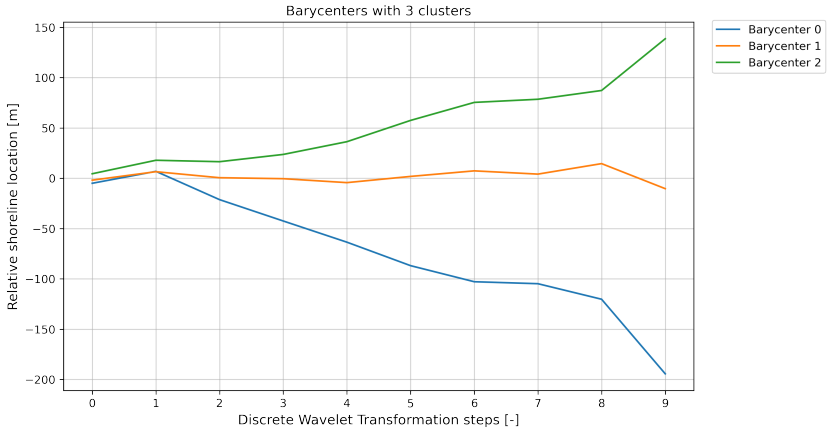

Figure C.3: Barycenters for three clusters, generated by TimeSeriesKmeans.

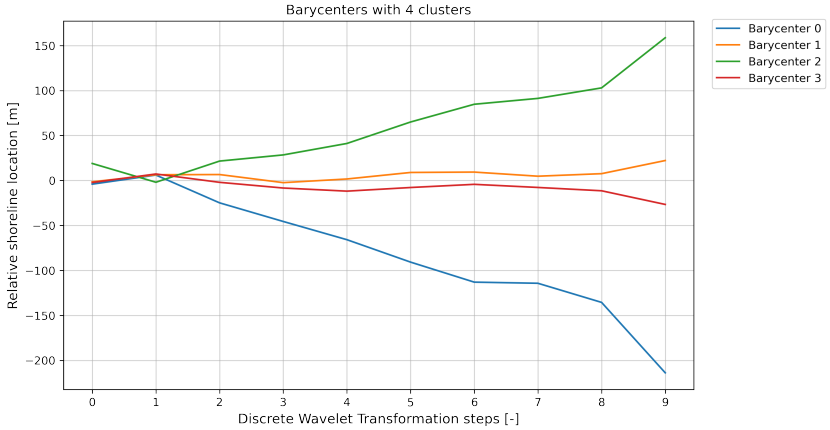

Figure C.4: Barycenters for four clusters, generated by TimeSeriesKmeans.

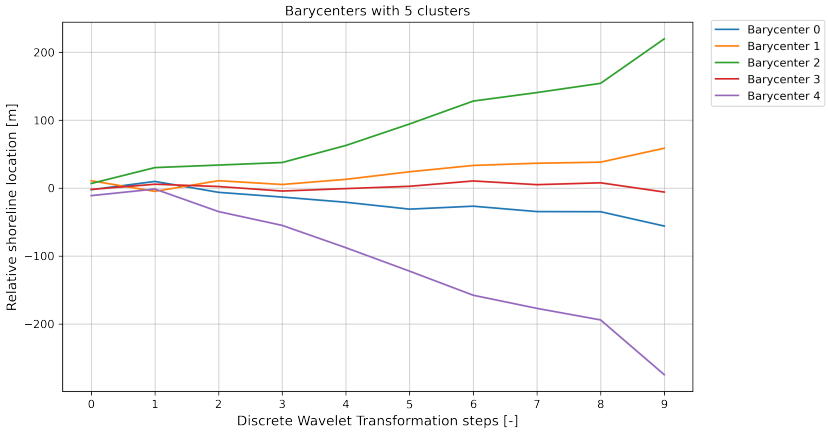

Figure C.5: Barycenters for five clusters, generated by TimeSeriesKmeans.

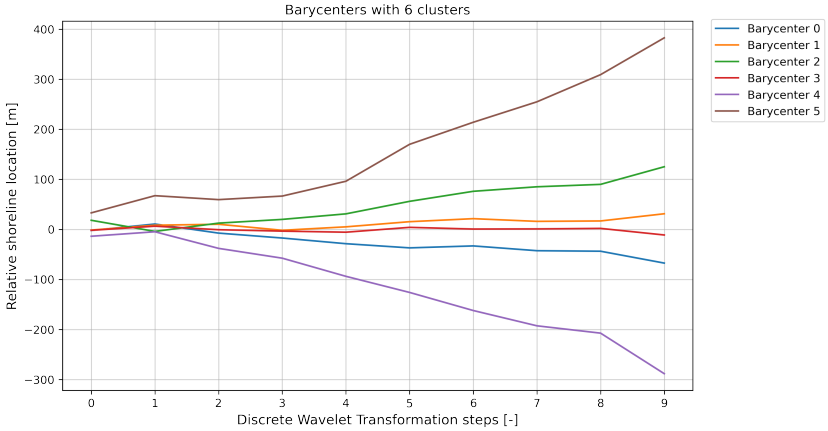

Figure C.6: Barycenters for six clusters, generated by TimeSeriesKmeans.

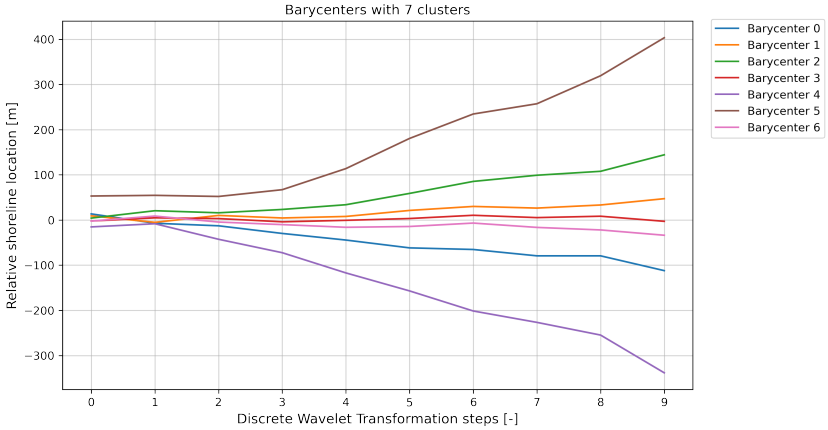

Figure C.7: Barycenters for seven clusters, generated by TimeSeriesKmeans.

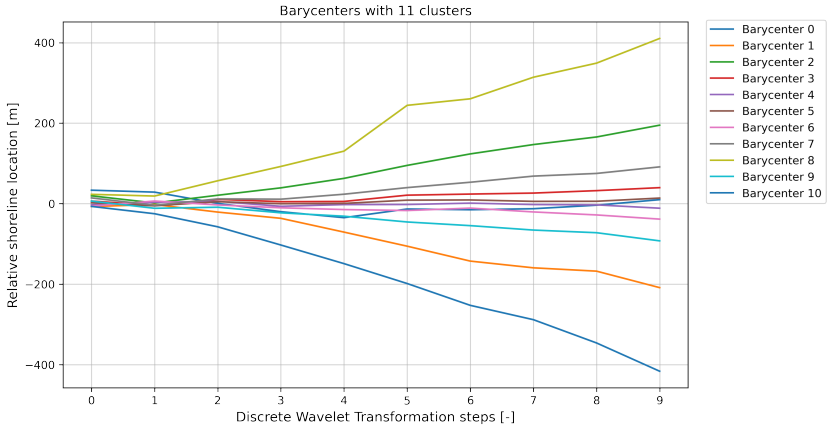

Figure C.8: Barycenters for eleven clusters, generated by TimeSeriesKmeans.

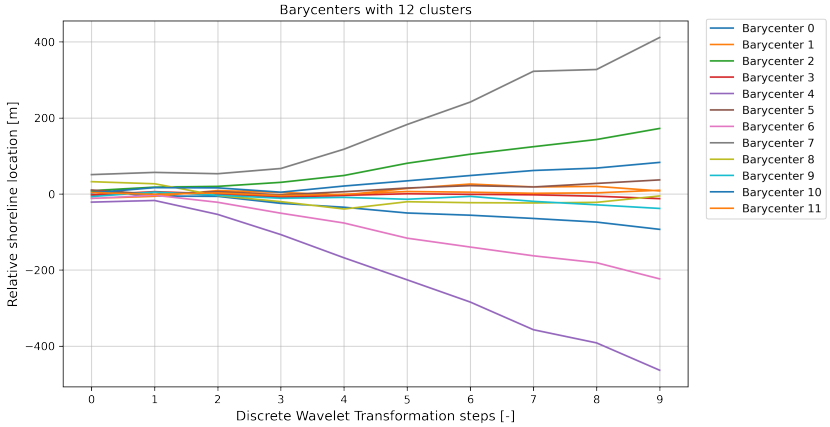

Figure C.9: Barycenters for twelve clusters, generated by TimeSeriesKmeans.

C.1.3 Final clusters

Here, the density plots of all transects in the nine main clusters are presented. The plots show the different behavior of transects in a cluster. Furthermore, the distribution of transects' behavior become visibly by the density plot.

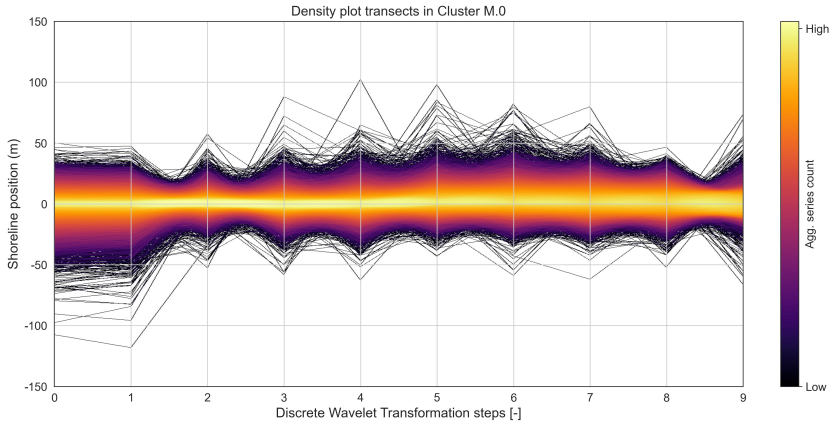

Figure C.10: Density plot of DWT representation of transects in cluster M.0 Moderate.

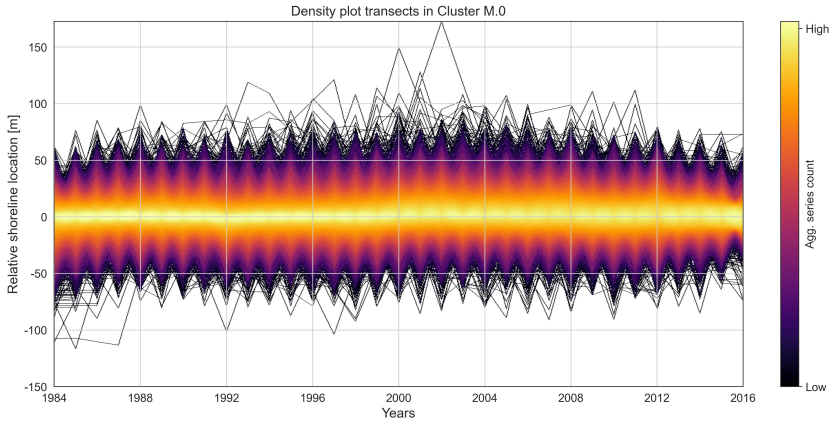

Figure C.11: Density plot of raw transects in cluster M.0 Moderate.

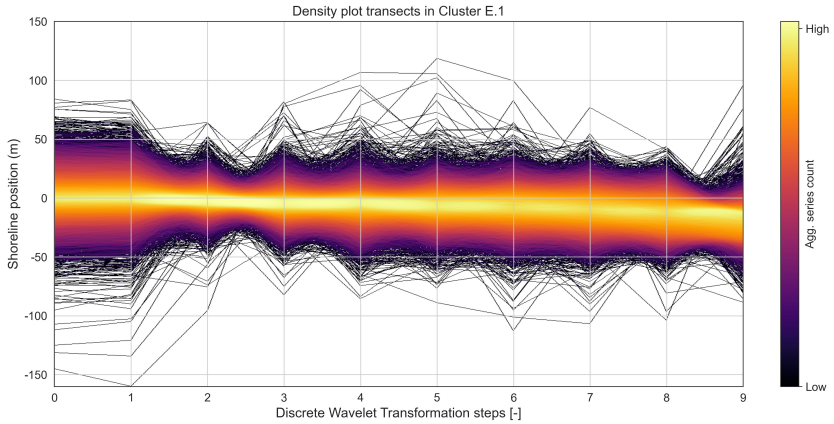

Figure C.12: Density plot of DWT representation of transects in cluster E.1 Erosion.

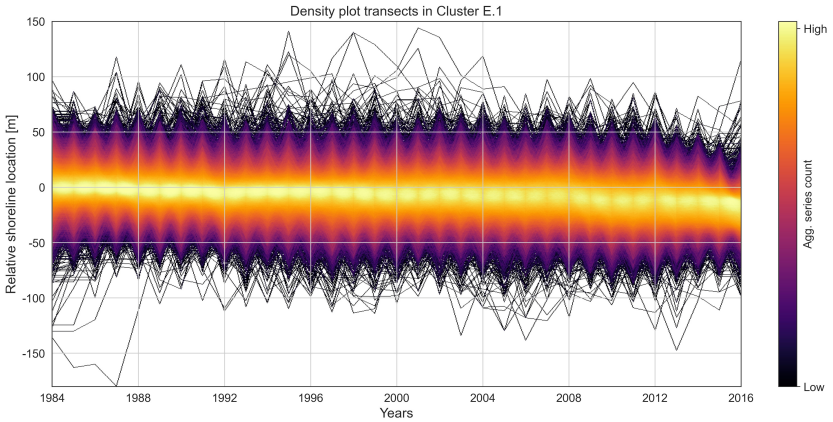

Figure C.13: Density plot of raw transects in cluster E.1 Erosion.

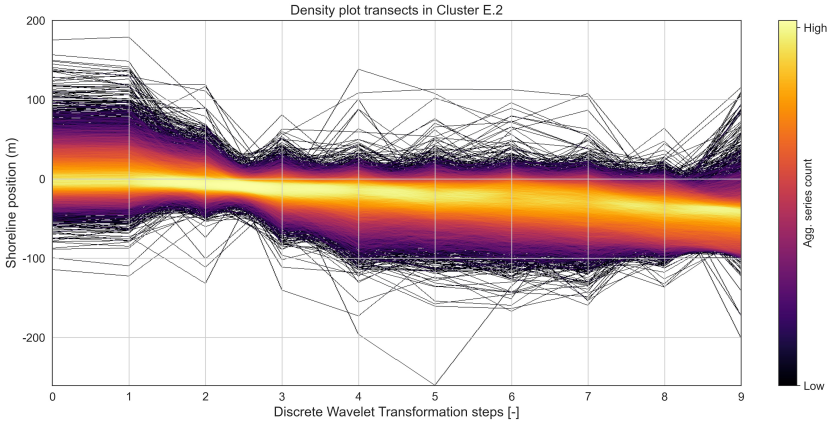

Figure C.14: Density plot of DWT representation of transects in cluster E.2 Intense Erosion

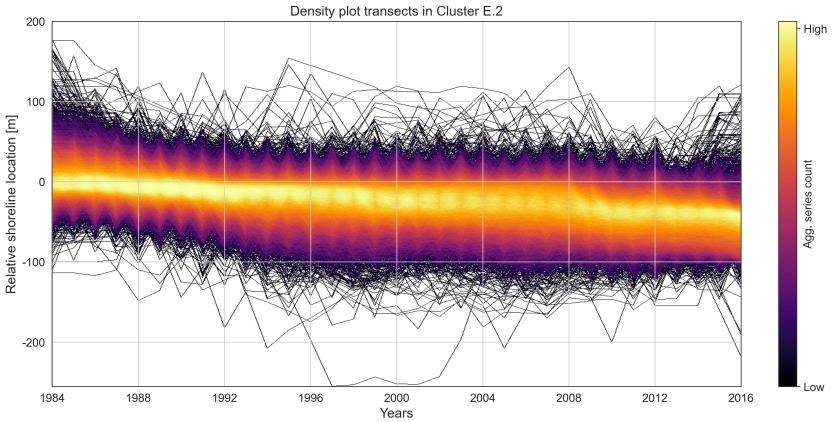

Figure C.15: Density plot of raw transects in cluster E.2 Intense Erosion.

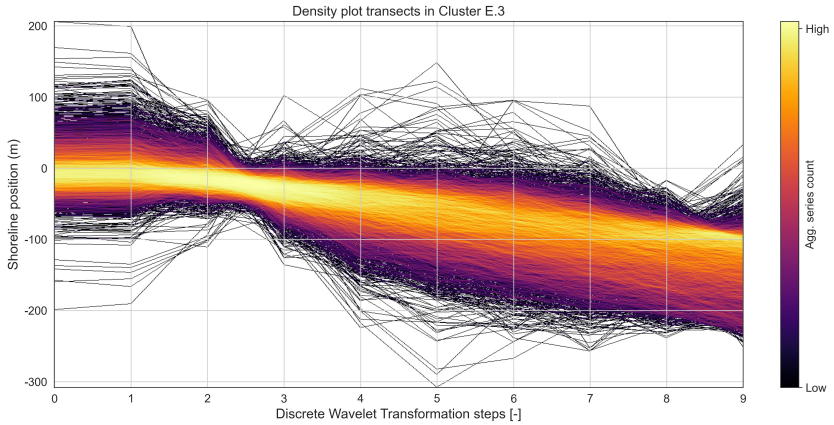

Figure C.16: Density plot of DWT representation of transects in cluster E.3 Severe Erosion.

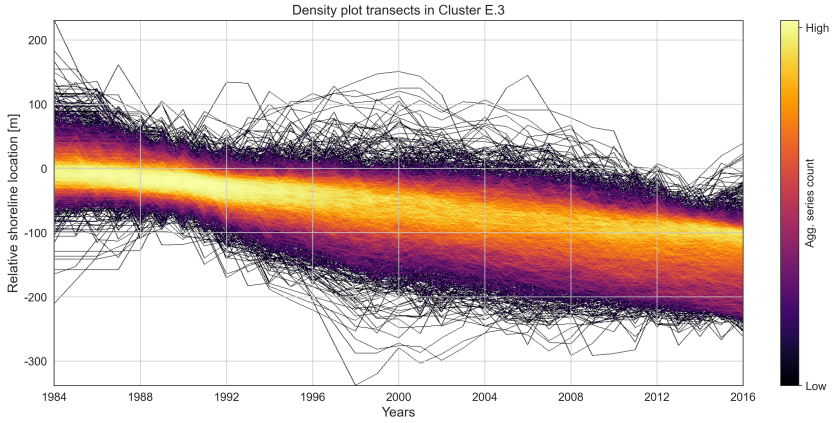

Figure C.17: Density plot of raw transects in cluster E.3 Severe Erosion.

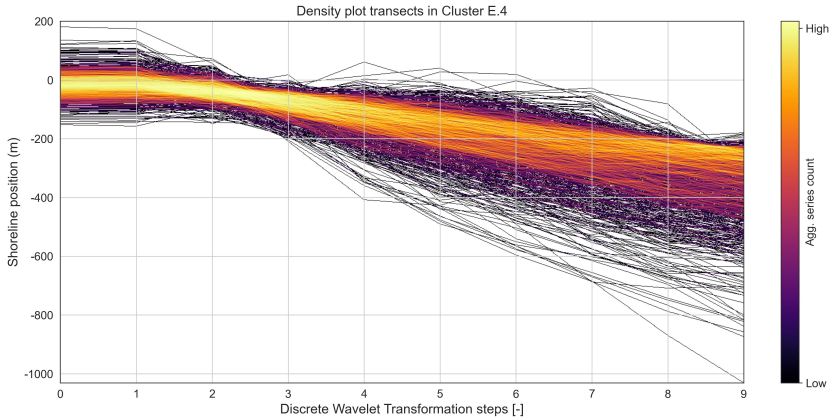

Figure C.18: Density plot of DWT representation of transects in cluster E.4 Extreme Erosion.

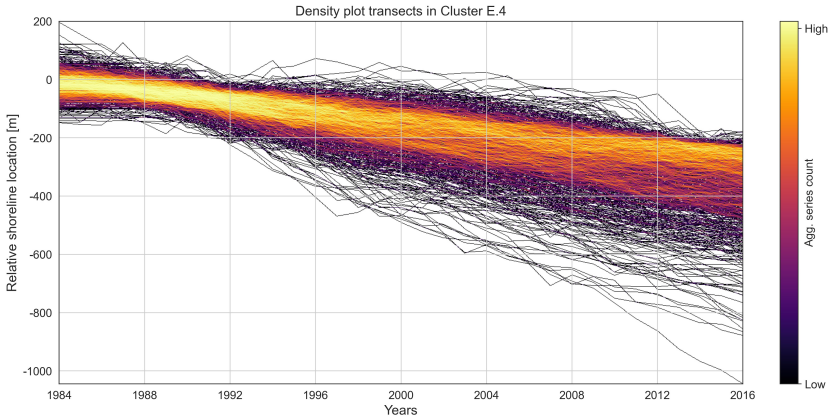

Figure C.19: Density plot of raw transects in cluster E.4 Extreme Erosion.

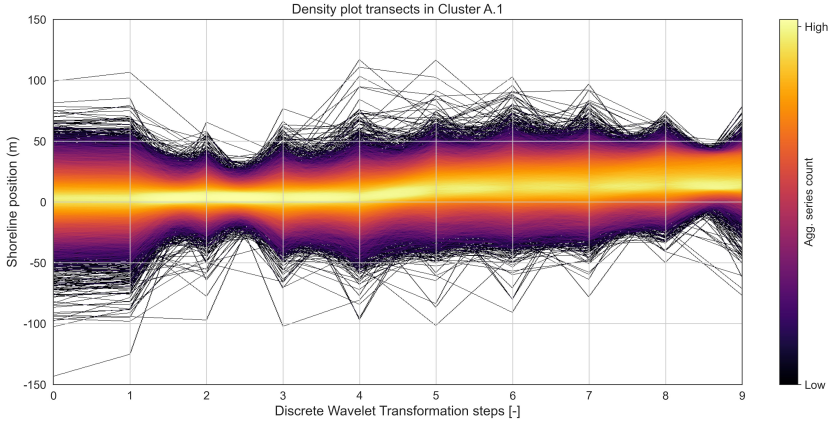

Figure C.20: Density plot of DWT representation of transects in cluster A.1 Accretion.

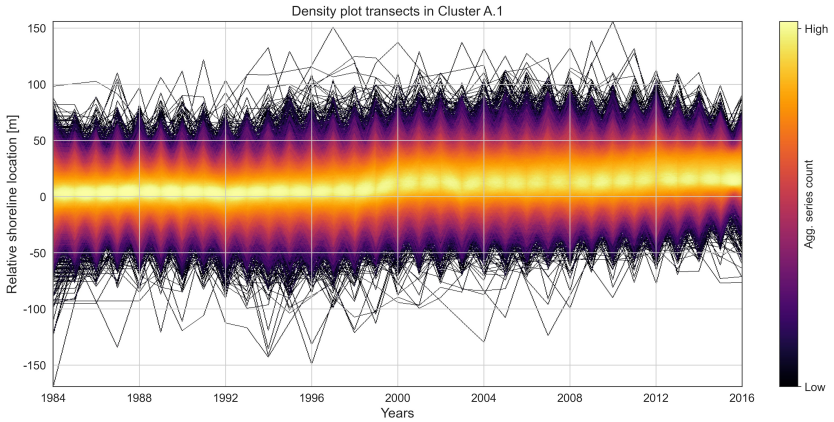

Figure C.21: Density plot of raw transects in cluster A.1 Accretion.

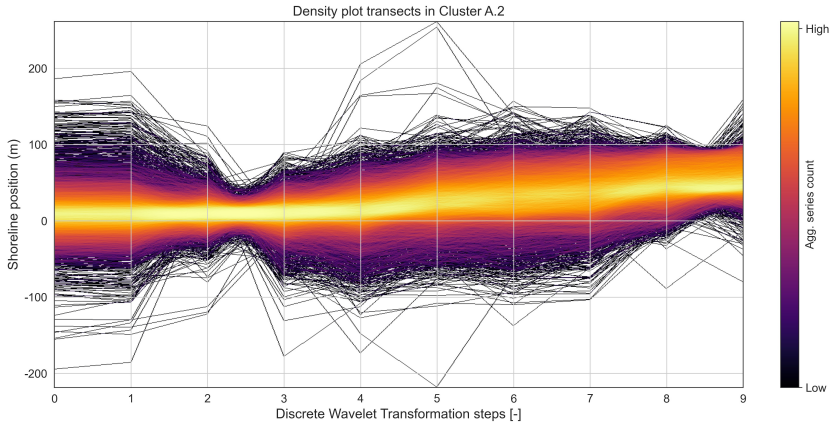

Figure C.22: Density plot of DWT representation of transects in cluster A.2 Intense Accretion.

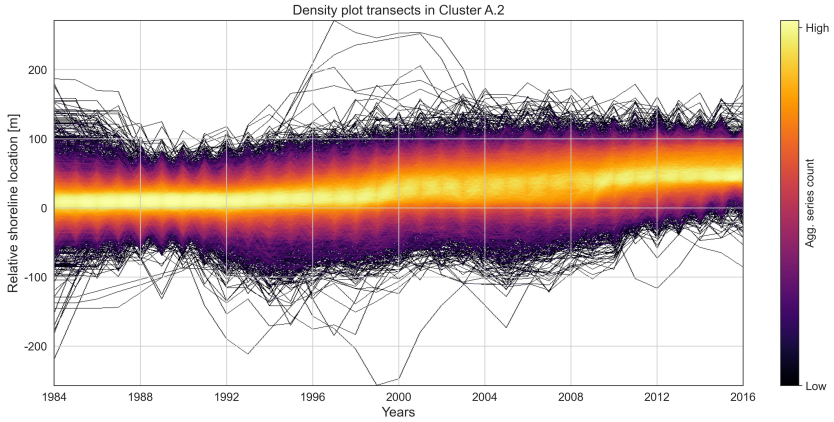

Figure C.23: Density plot of raw transects in cluster A.2 Intense Accretion.

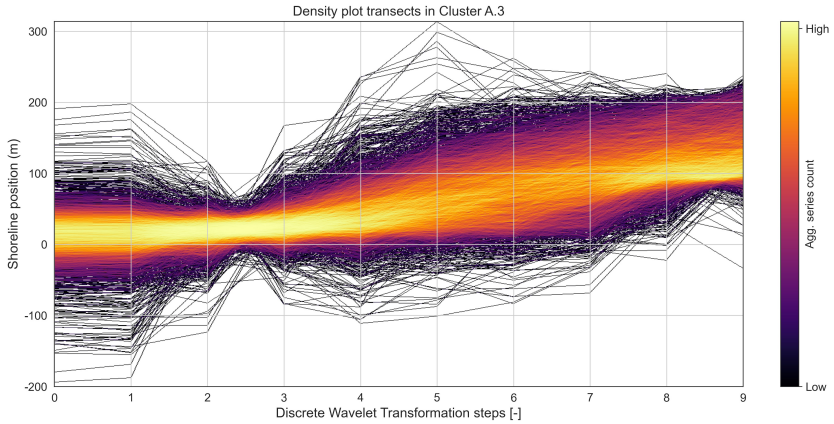

Figure C.24: Density plot of DWT representation of transects in cluster A.3 Severe Accretion.

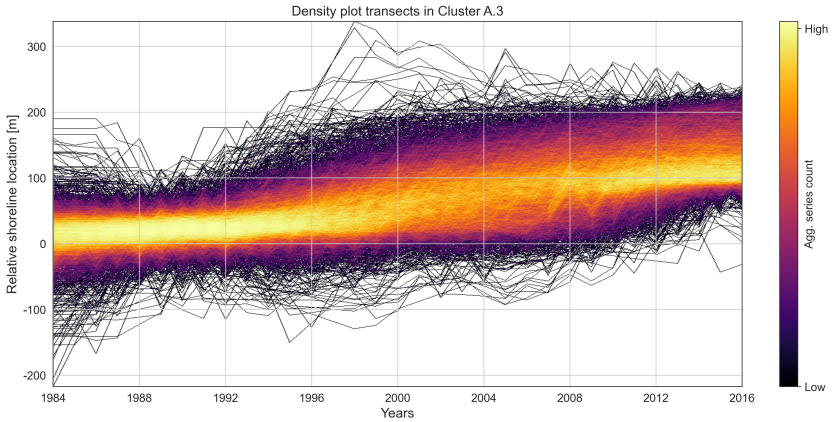

Figure C.25: Density plot of raw transects in cluster A.3 Severe Accretion.

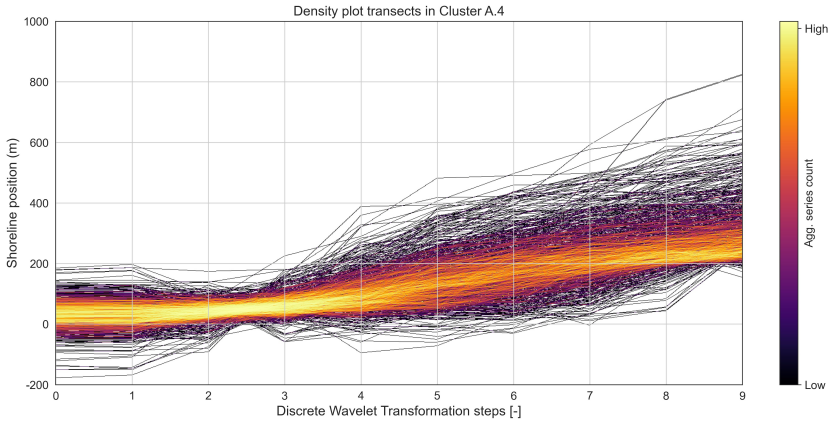

Figure C.26: Density plot of DWT representation of transects in cluster A.4 Extreme Accretion.

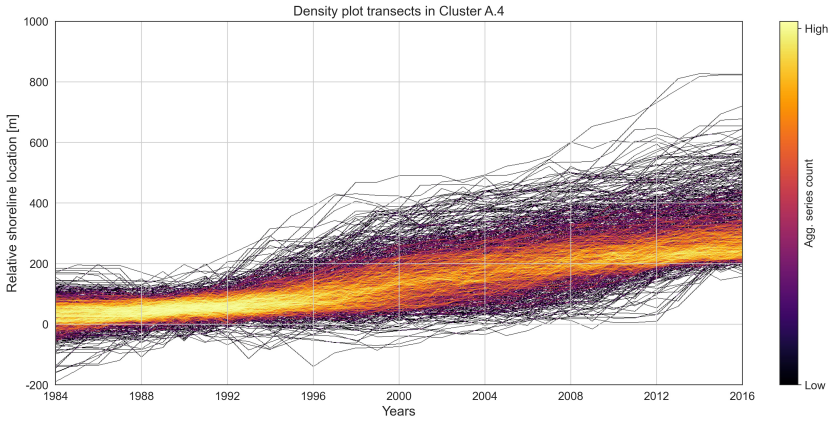

Figure C.27: Density plot of raw transects in cluster A.4 Extreme Accretion.

C.1.4 Distribution clusters

In this subsection the distributions of transects in the nine clusters are scattered on a global scale. The histograms on the axes count the number of transects per cluster per latitude and longitude. These maps can be used to quantify the distribution of shoreline behavior.

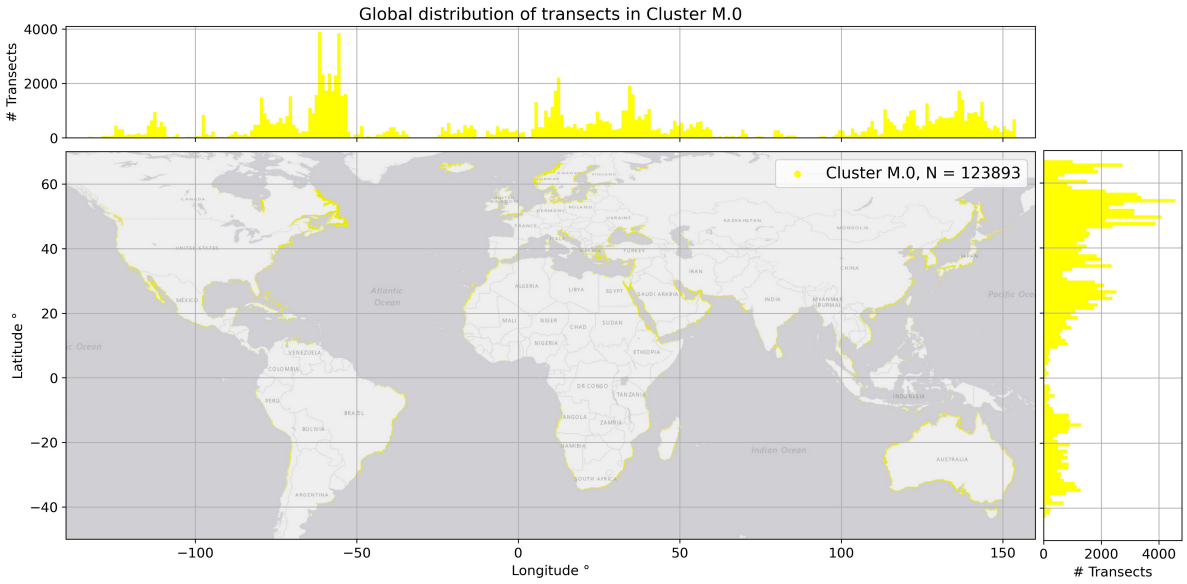

Figure C.28: Global distribution of sandy transects in absolute values in cluster M.0 Moderate.

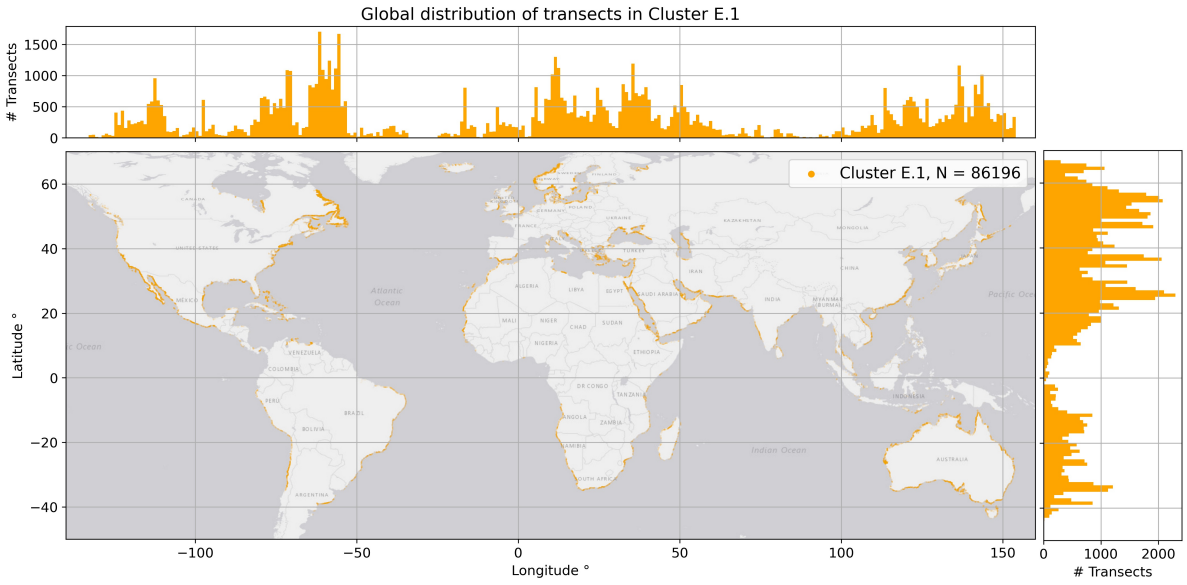

Figure C.29: Global distribution of sandy transects in absolute values in cluster E.1 Erosion.

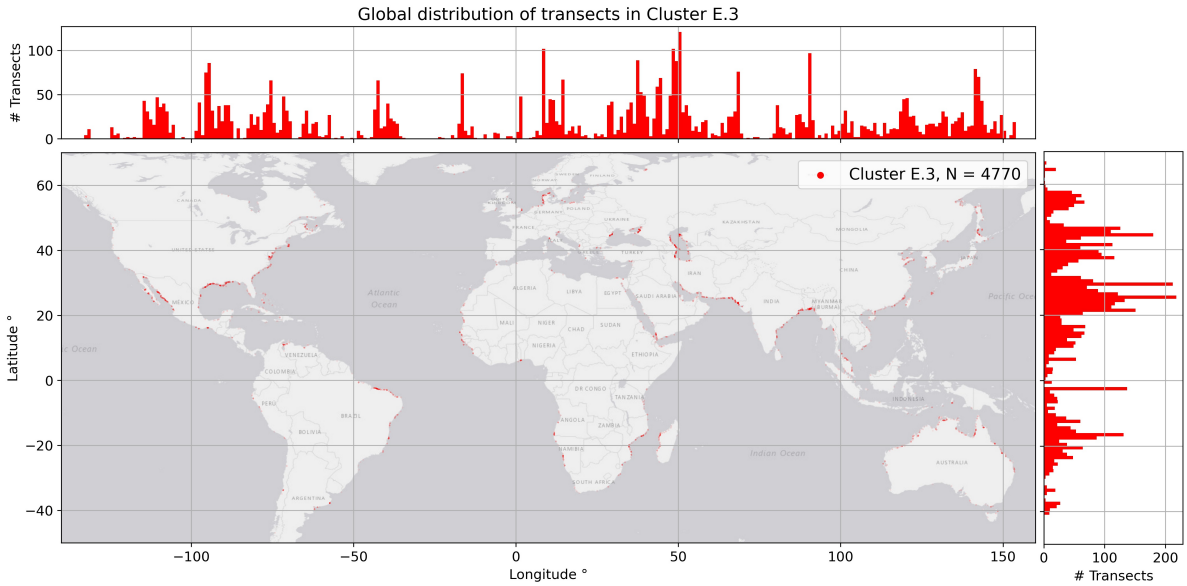

Figure C.30: Global distribution of sandy transects in absolute values in cluster E.3 Severe Erosion.

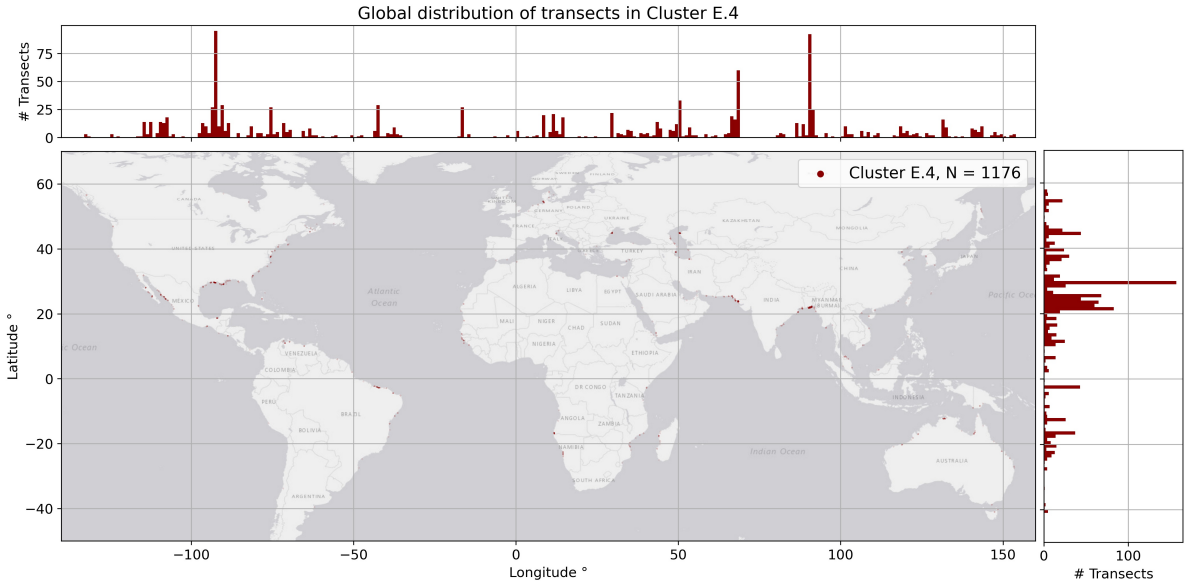

Figure C.31: Global distribution of sandy transects in absolute values in cluster E.4 Extreme Erosion.

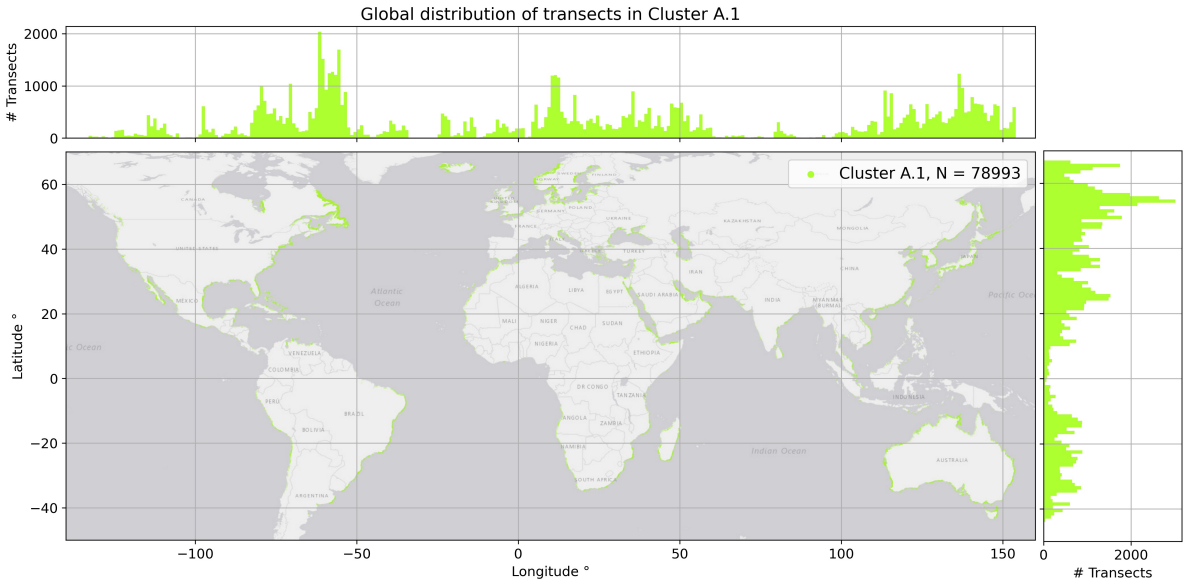

Figure C.32: Global distribution of sandy transects in absolute values in cluster A.1 Accretion.

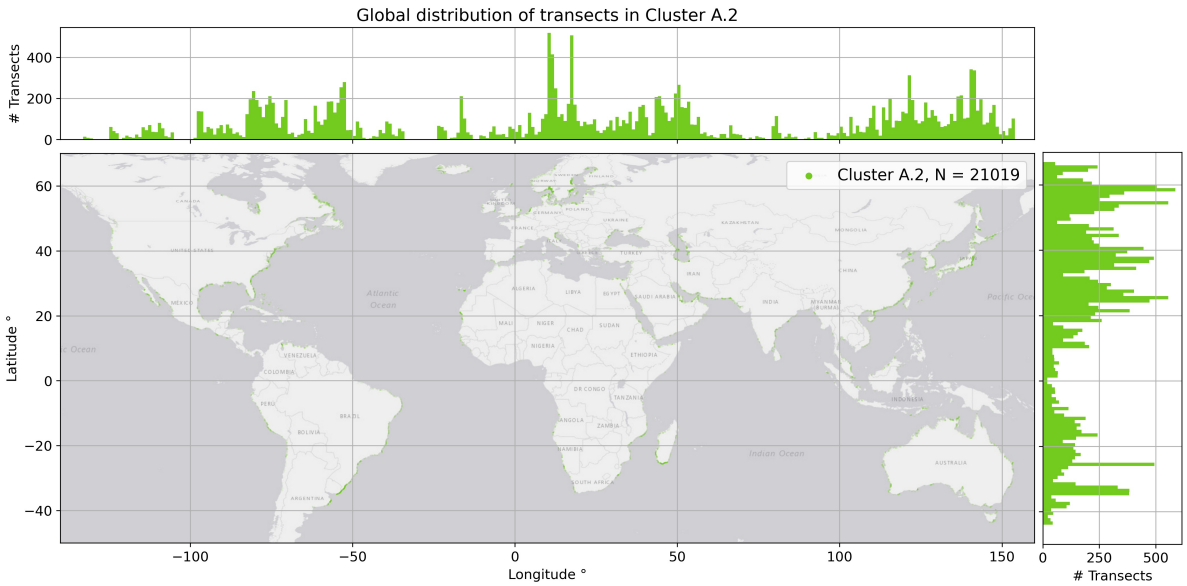

Figure C.33: Global distribution of sandy transects in absolute values in cluster A.2 Intense Accretion.

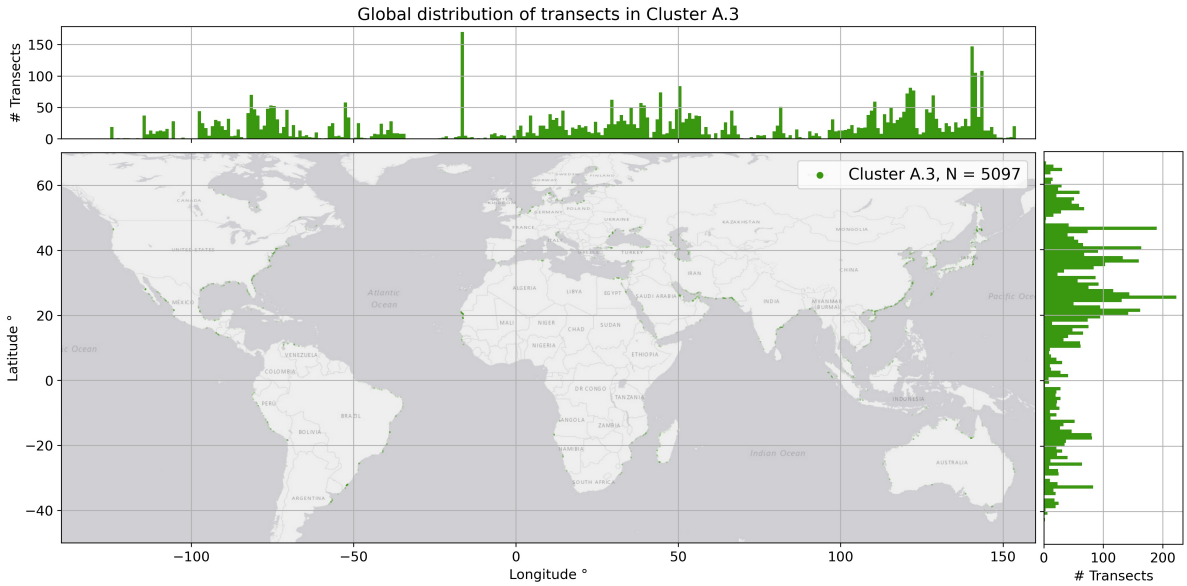

Figure C.34: Global distribution of sandy transects in absolute values in cluster A.3 Severe Accretion.

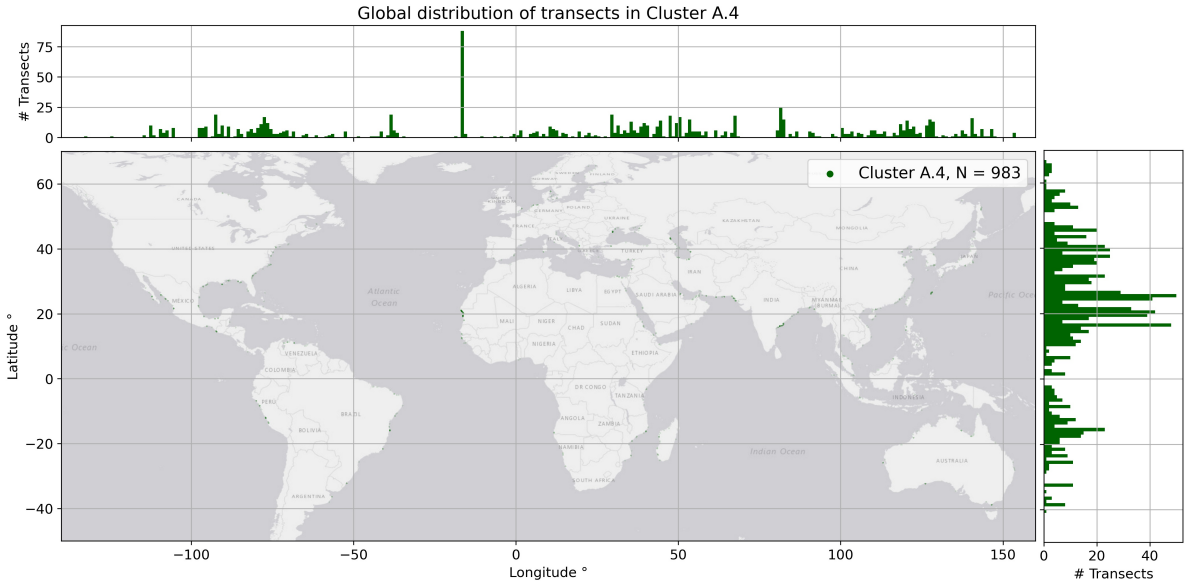

Figure C.35: Global distribution of sandy transects in absolute values in cluster A.4 Extreme Accretion.

C.1.5 Sub-clusters

In this subsection the sub-clusters for clusters M.0, E.1, E.2, A.1 and A.2 are plotted. Every page consists of the three sub-clusters per cluster. With the sub-clusters a distinction between accelerating, constant and decelerating shoreline is possible. Besides, the distributions of the sub-clusters per cluster are included.

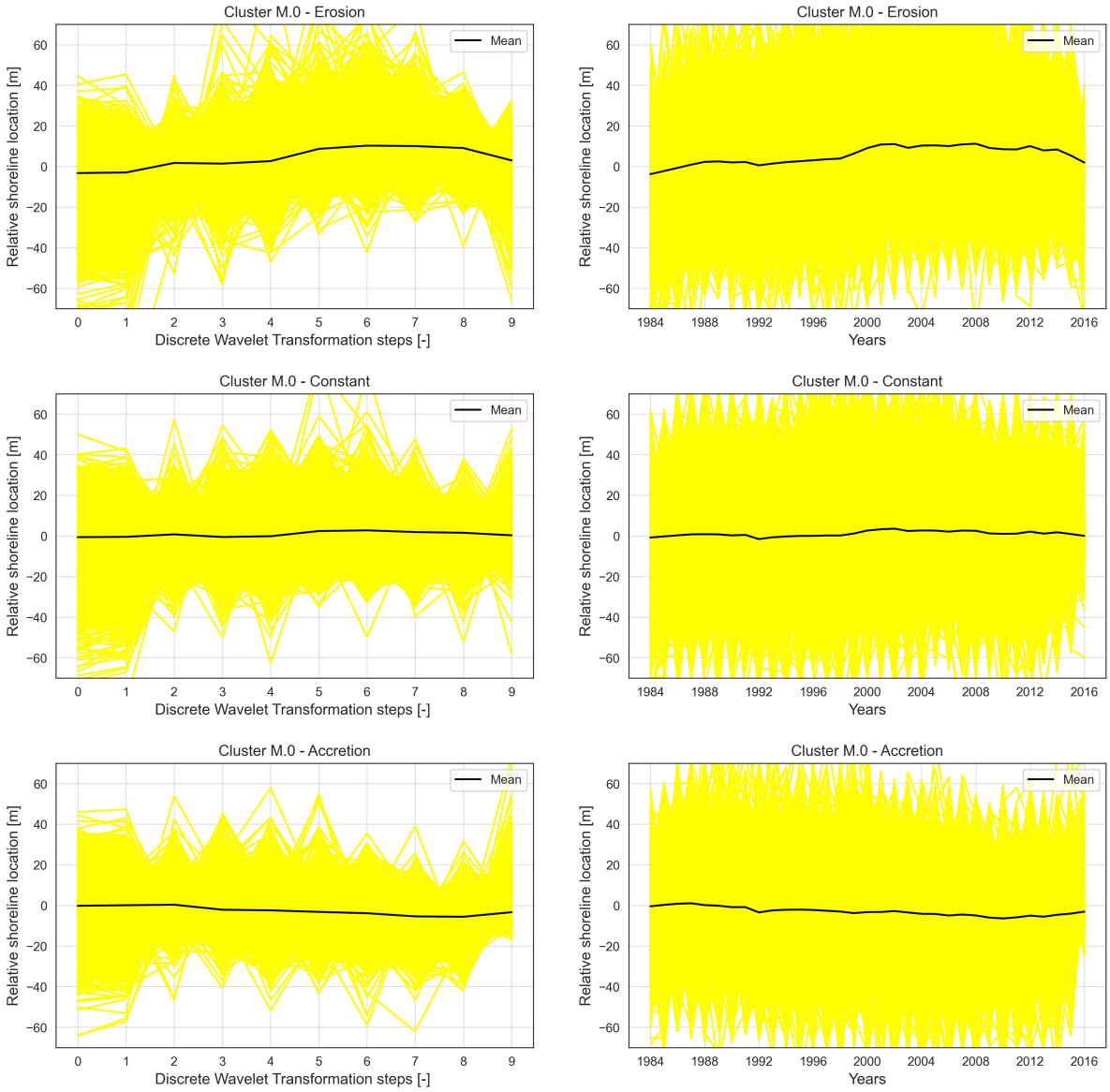

Figure C.36: Plots of all transects in the three different sub-clusters of cluster M.0. Left, the plots of the DWT representation and right the raw transect time series. The first row plots are the transects in the upper tail of the histogram, the middle row the transects in the center and the lowest row all the transects in the lower tail.

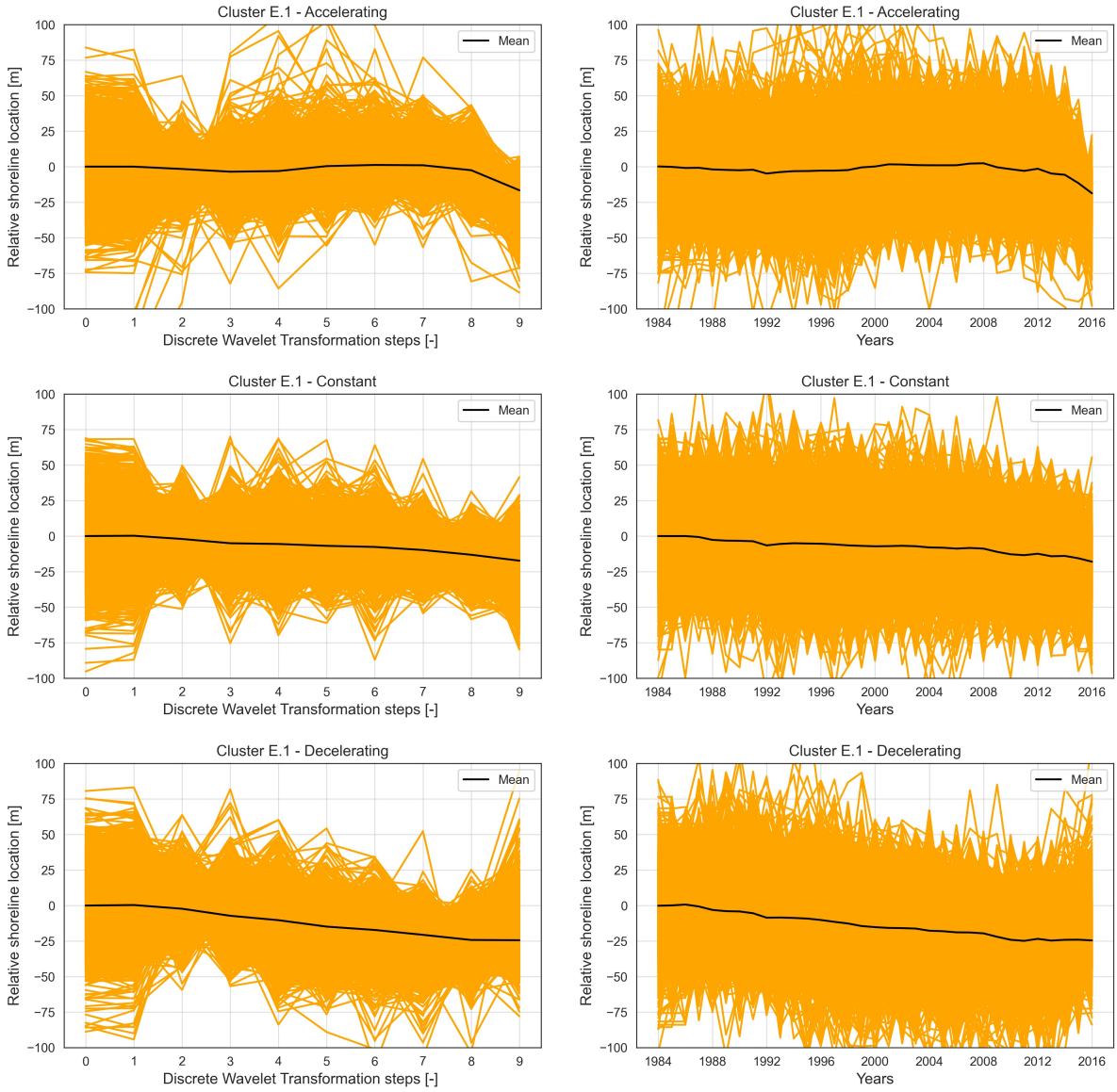

Figure C.37: Plots of all transects in the three different sub-clusters of cluster E.1. Left, the plots of the DWT representation and right the raw transect time series. The first row plots are the transects in the upper tail of the histogram, the middle row the transects in the center and the lowest row all the transects in the lower tail.

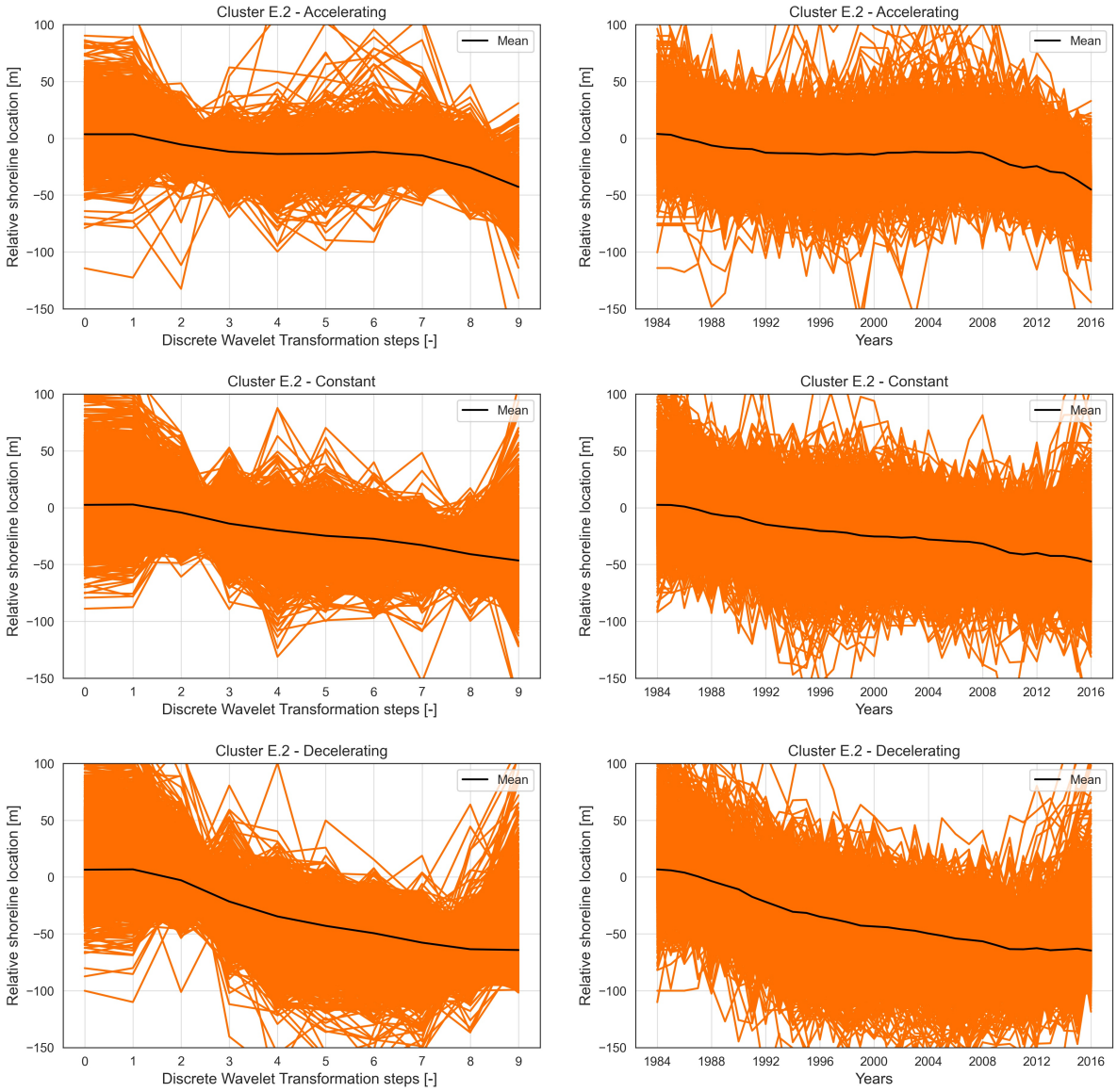

Figure C.38: Plots of all transects in the three different sub-clusters of cluster E.2. Left, the plots of the DWT representation and right the raw transect time series. The first row plots are the transects in the upper tail of the histogram, the middle row the transects in the center and the lowest row all the transects in the lower tail.

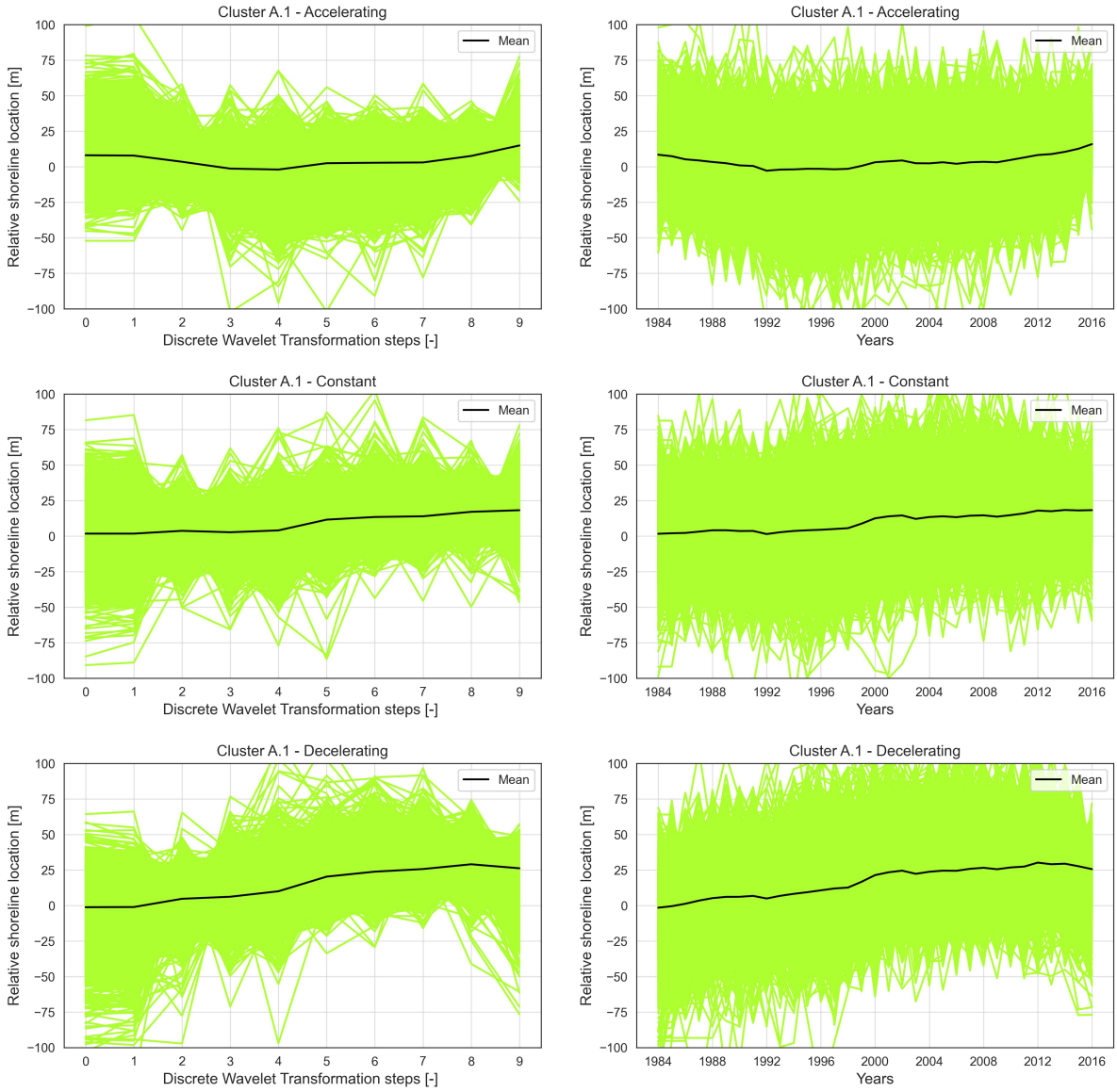

Figure C.39: Plots of all transects in the three different sub-clusters of cluster A.1. Left, the plots of the DWT representation and right the raw transect time series. The first row plots are the transects in the lower tail of the histogram, the middle row the transects in the center and the lowest row all the transects in the upper tail.

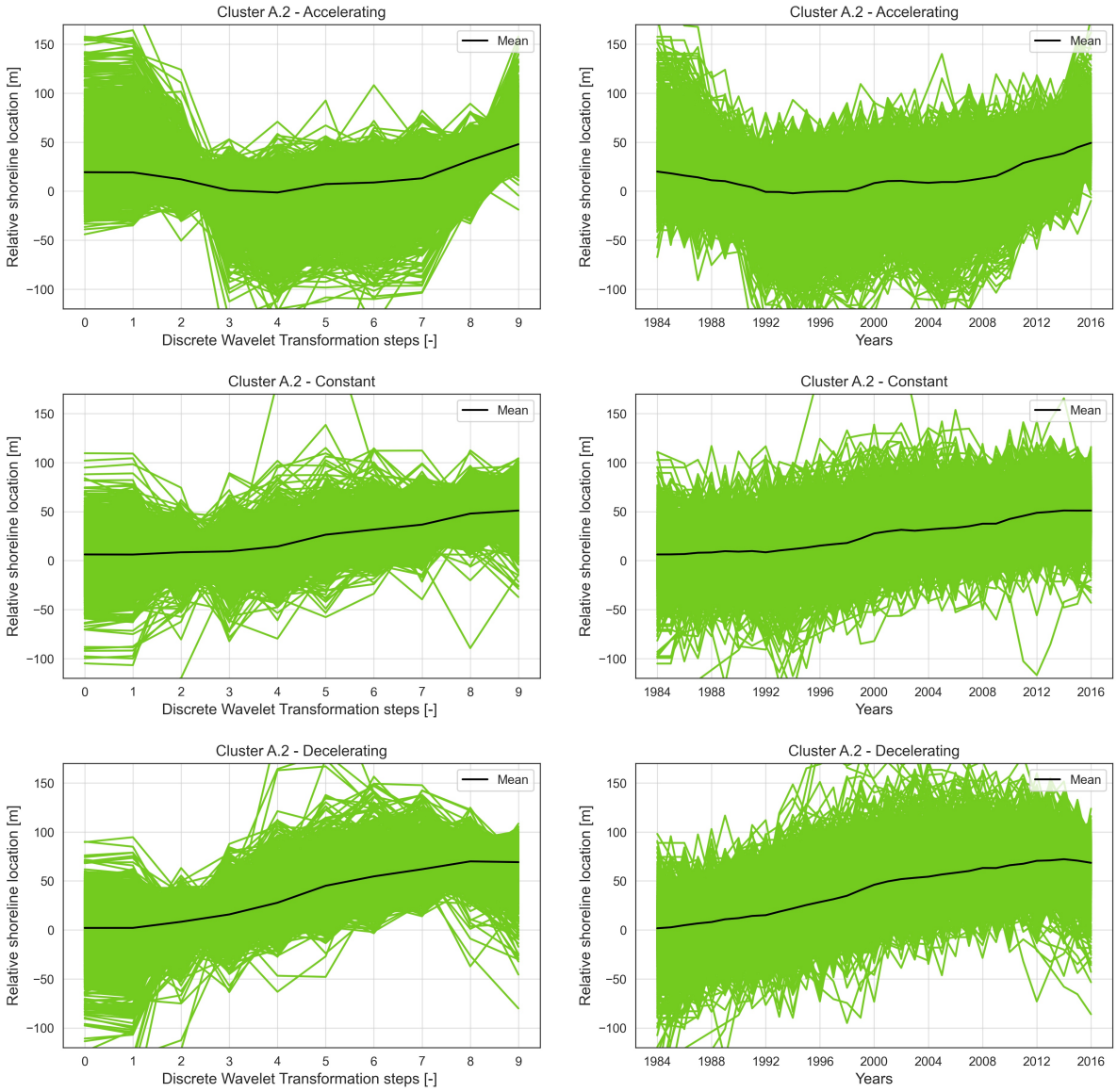

Figure C.40: Plots of all transects in the three different sub-clusters of cluster A.2. Left, the plots of the DWT representation and right the raw transect time series. The first row plots are the transects in the lower tail of the histogram, the middle row the transects in the center and the lowest row all the transects in the upper tail.

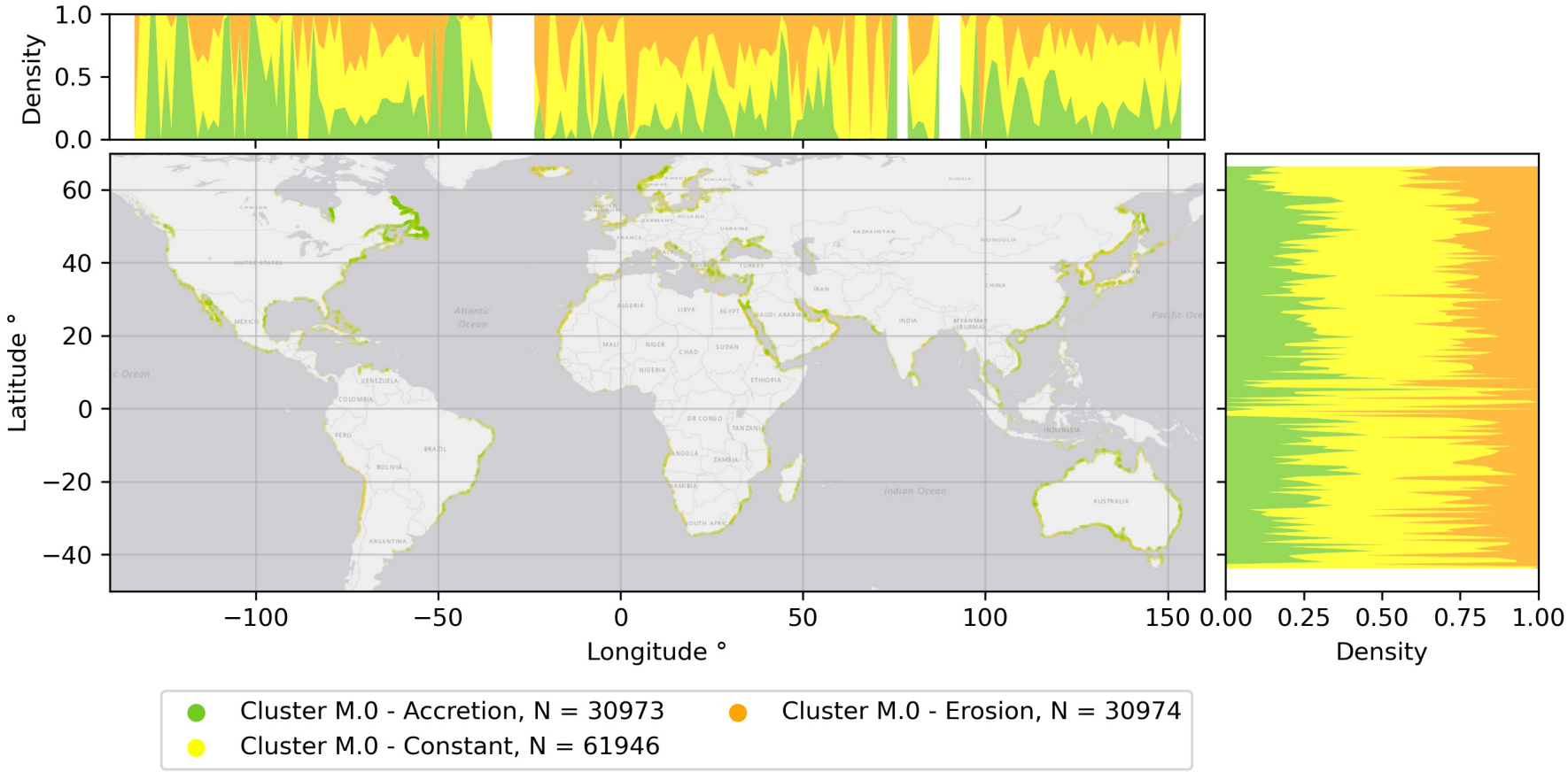

Figure C.41: Density plot of sandy transects in sub-clusters of cluster M.0 Moderate. At the axes, a kernel density plot is added to show distribution of the sub-clusters per longitude and latitude.

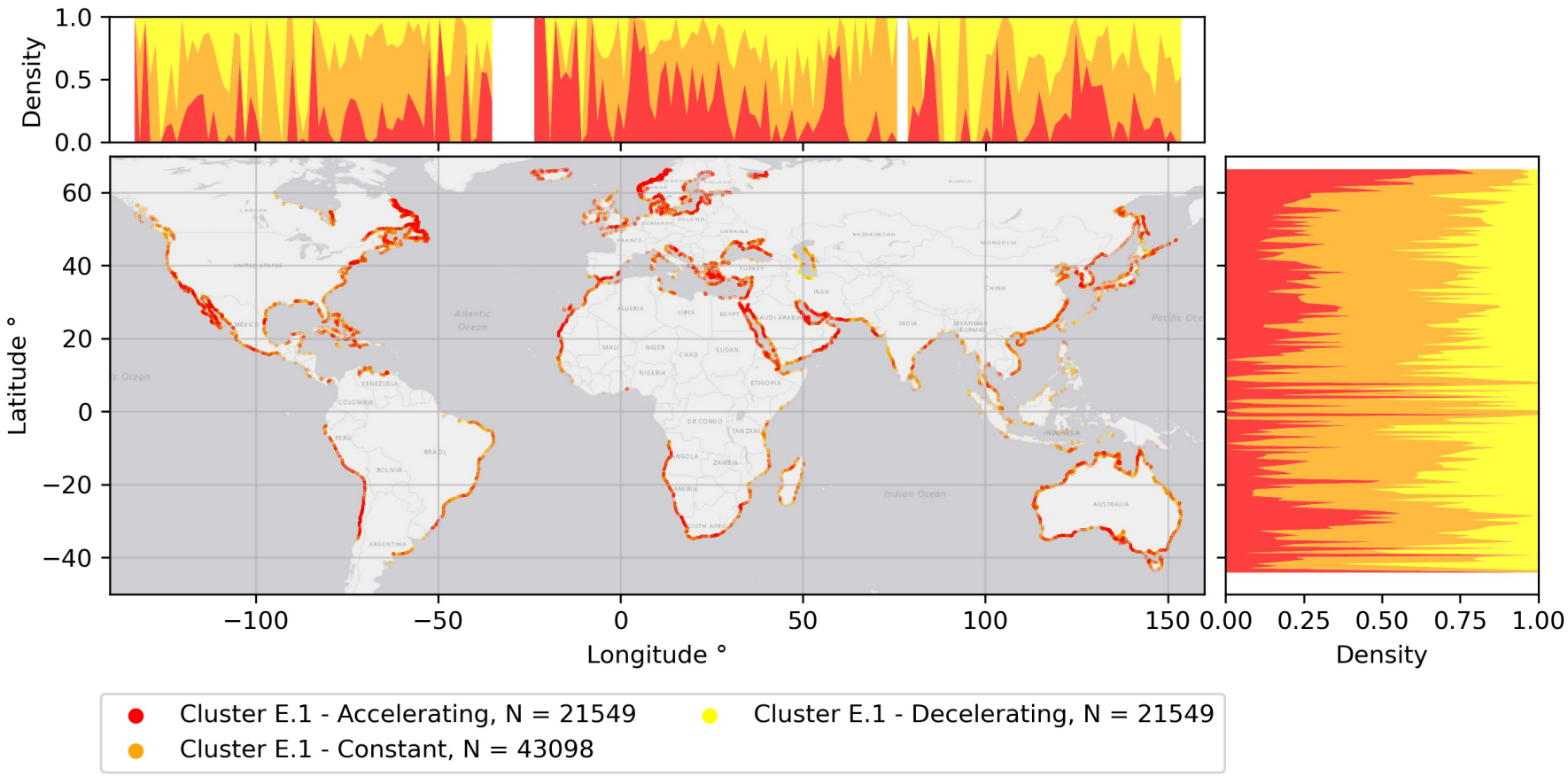

Figure C.42: Density plot of sub-clusters of cluster E.1 Erosion. At the axes, a kernel density plot is added to show distribution of the sub-clusters per longitude and latitude.

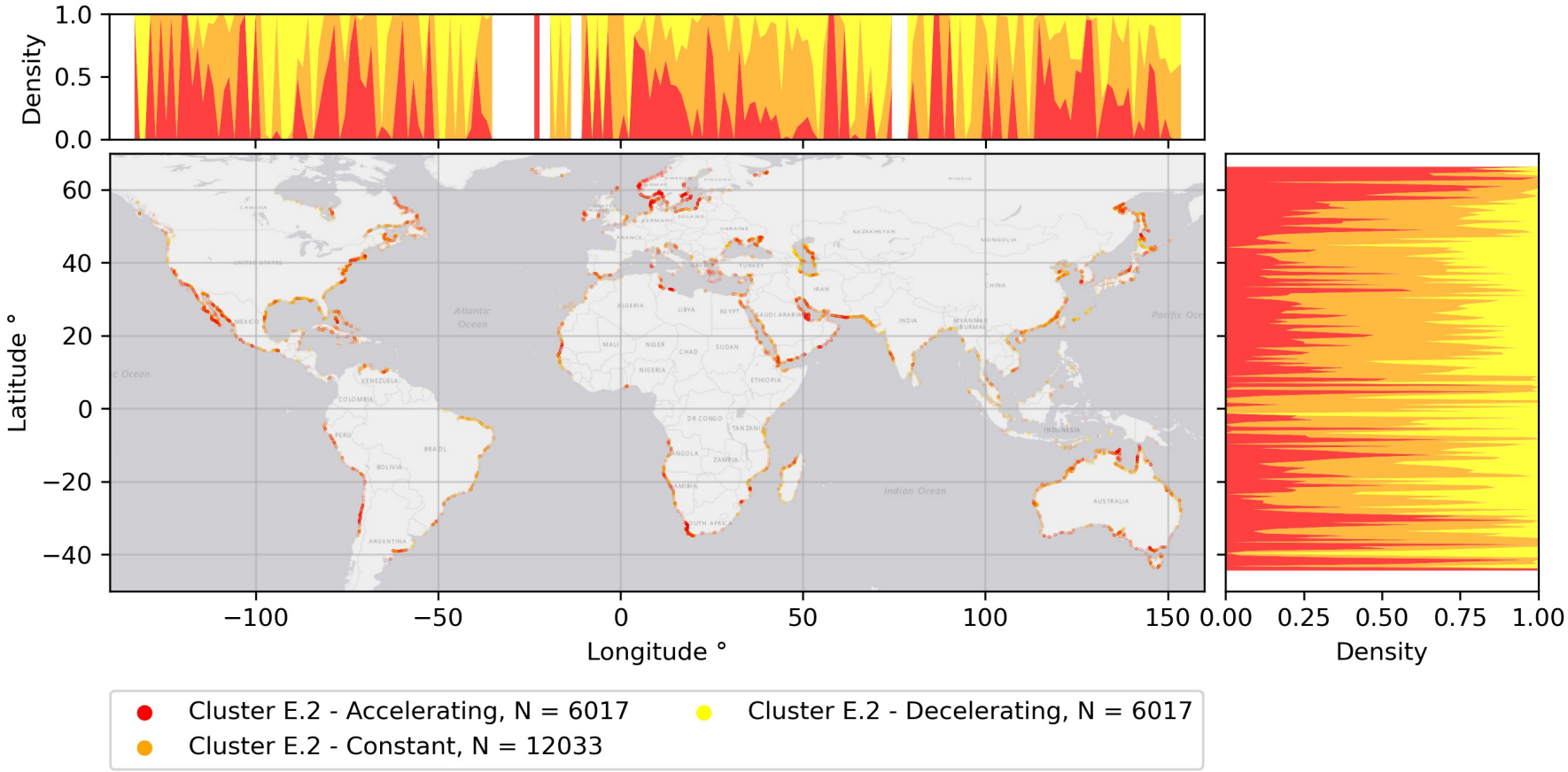

Figure C.43: Density plot of sandy transects in sub-clusters of cluster E.2 Intense Erosion. At the axes, a kernel density plot is added to show distribution of the sub-clusters per longitude and latitude.

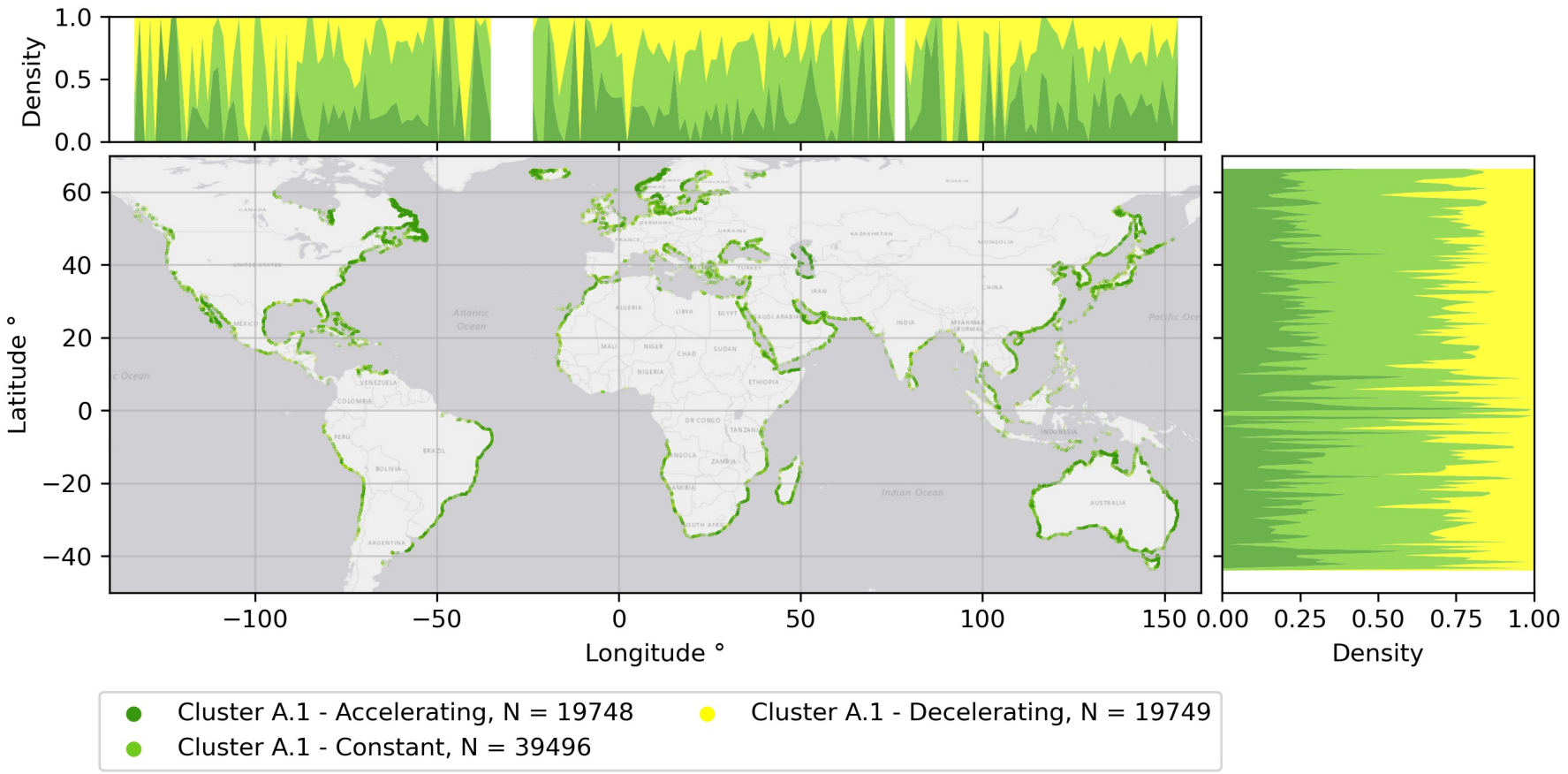

Figure C.44: Density plot of sandy transects in sub-clusters of cluster A.1 Accretion. At the axes, a kernel density plot is added to show distribution of the sub-clusters per longitude and latitude.

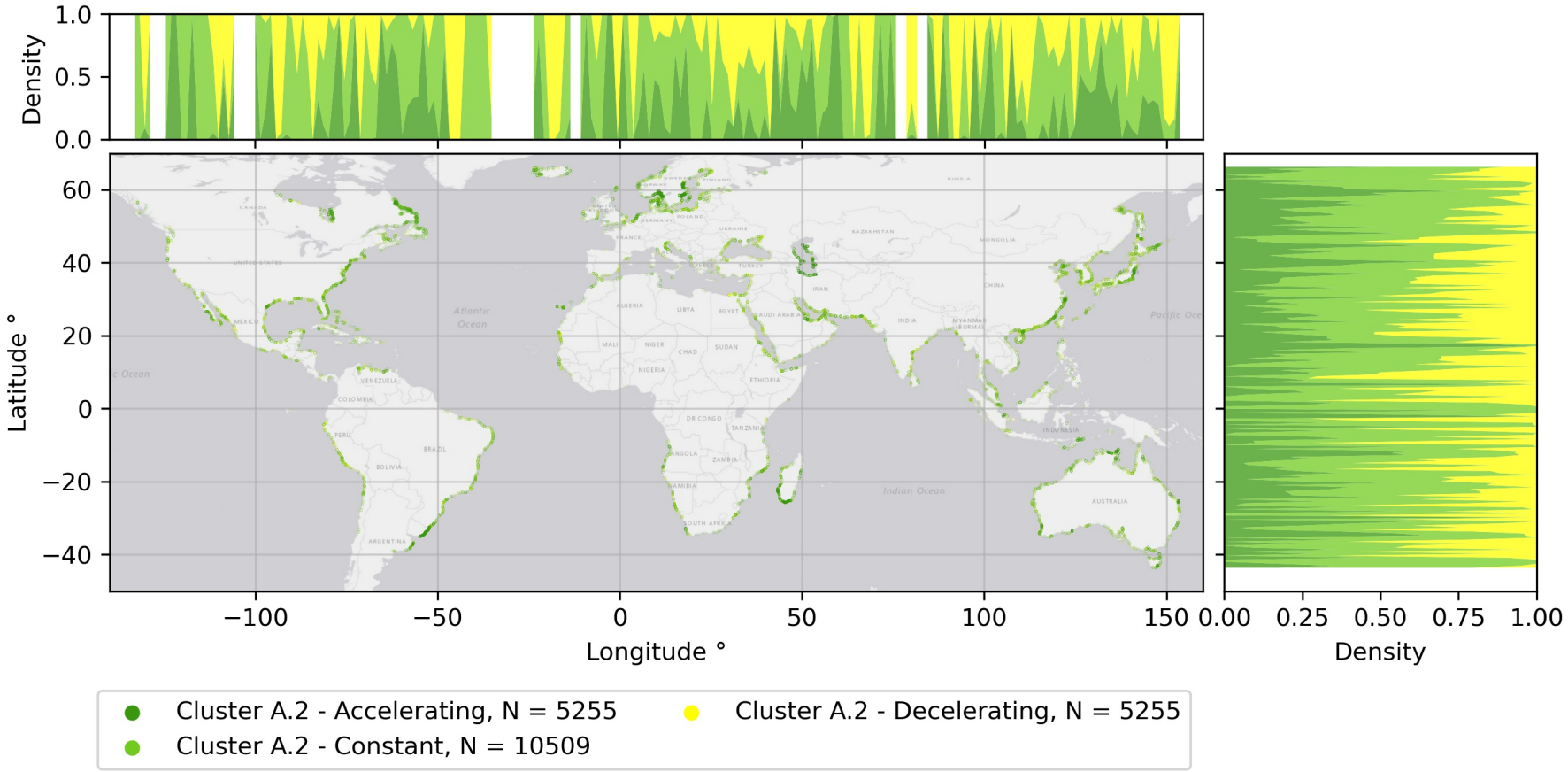

Figure C.45: Density plot of sandy transects in sub-clusters of cluster A.2 Intense Accretion. At the axes, a kernel density plot is added to show distribution of the sub-clusters per longitude and latitude.

C.2 Forecasting results

Here the results regarding forecasting transects are presented. The tables contain the forecasting accuracy scores per cluster for all four considered forecasting algorithms, where lower scores indicate higher accuracy. Again the MSE and MAE are highlighted in green as these indicate the accuracy in m^2 and m respectively, and thus are of primary importance for understanding model performance.

Table C.1: Evaluation measures forecasting algorithms for cluster M.0 Moderate. MAE and RMSE are in meter. Some scores for MQCNN are left out as the calculation of these scores was different compared to the other algorithms. The MSE and MAE are here significant lower than for previous research, indicating a higher accuracy in m^2 and m. MAPE is significant higher due to bias of predictions close to 0. Furthermore, all forecasting algorithms outperform OLS. The best overall algorithms here are DeepAR and simpleFFN. The fact that SimpleFFN is performing well, corresponds with the general knowledge of cluster M.0. The ACF is low and the behavior is moderate, which can indicate less sequence information in these transects.

N = 123,893	MSE	MAE	MASE	MAPE	sMAPE	RMSE	NRMSE	ND	MSIS	CRPS
SimpleFFN	121	7.6	1.58	506.48	1.00	11.02	1.07	0.74	11.18	0.61
DeepAR	120	7.7	1.61	477.24	0.99	10.95	1.06	0.75	10.98	0.62
MQCNN	-	8.6	1.57	684.92	1.02	-	-	0.83	-	0.66
DeepSSM	134	8.1	1.95	661.91	1.01	11.58	1.12	0.79	9.60	0.64
OLS	161	8.9	1.23	782.15	1.02	12.70	1.23	0.86	-	-

Table C.2: Evaluation measures forecasting algorithms for cluster E.2 Intense Erosion. MAE and RMSE are in meter. Some scores for MQCNN are left out as the calculation of these scores was different compared to the other algorithms. The MSE and MAE are here significant lower than for previous research, indicating a higher accuracy in m^2 and m. Furthermore, all forecasting algorithms outperform OLS. The best algorithms here are DeepAR and MQCNN.

N = 24,067	MSE	MAE	MASE	MAPE	sMAPE	RMSE	NRMSE	ND	MSIS	CRPS
SimpleFFN	502	15.6	1.58	2.99	0.39	22.40	0.45	0.32	15.68	0.26
DeepAR	457	15.5	1.61	2.89	0.42	21.38	0.43	0.32	15.16	0.25
MQCNN	-	15.6	1.57	3.09	0.45	-	-	0.32	-	0.25
DeepSSM	731	19.5	1.95	4.13	0.54	27.04	0.55	0.40	25.04	0.33
OLS	919	21.0	2.13	4.54	0.62	30.32	0.62	0.43	-	-

Table C.3: Evaluation measures forecasting algorithms for cluster E.3 Severe Erosion. MAE and RMSE are in meter. Some scores for MQCNN are left out as the calculation of these scores was different compared to the other algorithms. The MSE is here lower than for previous research. The MAE although, is relative larger. A lower MSE and higher MAE, indicate a lower maximum forecast error. Furthermore, all forecasting algorithms outperform OLS. The best algorithms here are DeepAR and MQCNN.

N = 4,770	MSE	MAE	MASE	MAPE	sMAPE	RMSE	NRMSE	ND	MSIS	CRPS
SimpleFFN	1514	26.3	2.18	0.29	0.24	38.90	0.32	0.22	18.15	0.17
DeepAR	1044	22.6	1.94	0.28	0.22	32.31	0.27	0.19	15.85	0.15
MQCNN	-	22.0	1.85	0.25	0.21	-	-	0.18	-	0.15
DeepSSM	1805	29.5	2.47	0.33	0.28	42.48	0.35	0.25	26.49	0.20
OLS	2667	35.6	2.99	0.43	0.35	51.64	0.43	0.30	-	-

Table C.4: Evaluation measures forecasting algorithms for cluster E.4 Extreme Erosion. MAE and RMSE are in meter. Some scores for MQCNN are left out as the calculation of these scores was different compared to the other algorithms. The MSE and MAE are here significant higher and thus forecasting transects with these algorithms does not lead to a better overall accuracy. The MAPE is low, indicating errors are low in percentage of magnitude. Furthermore, DeepAR does not outperform OLS, indicating that this global models does not contribute to forecasting improvement. The MQCNN and DeepSSM are here the best algorithms.

N = 1,776	MSE	MAE	MASE	MAPE	sMAPE	RMSE	NRMSE	ND	MSIS	CRPS
SimpleFFN	4812	44.8	2.79	0.16	0.14	69.37	0.23	0.15	19.60	0.12
DeepAR	5512	53.2	3.37	0.19	0.17	74.25	0.24	0.18	29.49	0.14
MQCNN	-	37.3	2.37	0.13	0.13	-	-	0.12	-	0.10
DeepSSM	3392	40.0	2.53	0.14	0.14	58.24	0.19	0.13	27.49	0.11
OLS	5352	52.2	3.32	0.19	0.20	73.16	0.24	0.17	-	-

Table C.5: Evaluation measures forecasting algorithms for cluster A.1 Accretion. MAE and RMSE are in meter. Some scores for MQCNN are left out as the calculation of these scores was different compared to the other algorithms. The MSE and MAE are here significant lower than for previous research, indicating a higher accuracy in m^2 and m. Furthermore, all forecasting algorithms outperform OLS, except the MAPE score for the MQCNN, that is probably caused by values close to zero. The best algorithms here are DeepAR and MQCNN.

N = 78,993	MSE	MAE	MASE	MAPE	sMAPE	RMSE	NRMSE	ND	MSIS	CRPS
SimpleFFN	264	11.7	1.35	204.99	0.77	16.24	0.80	0.58	14.33	0.47
DeepAR	223	11.1	1.29	171.97	0.78	14.92	0.73	0.54	13.77	0.44
MQCNN	-	10.7	1.25	408.51	0.76	-	-	0.53	-	0.42
DeepSSM	272	12.1	1.41	111.93	0.83	16.48	0.81	0.59	16.63	0.49
OLS	323	13.1	1.52	346.71	0.83	17.96	0.88	0.64	-	-

Table C.6: Evaluation measures forecasting algorithms for cluster A.2 Intense Accretion. MAE and RMSE are in meter. Some scores for MQCNN are left out as the calculation of these scores was different compared to the other algorithms. The MSE and MAE are here significant lower than for previous research, indicating a higher accuracy in m^2 and m. Furthermore, all forecasting algorithms outperform OLS. The best algorithms here are DeepAR and MQCNN.

N = 21,019	MSE	MAE	MASE	MAPE	sMAPE	RMSE	NRMSE	ND	MSIS	CRPS
SimpleFFN	694	19.7	1.82	2.12	0.59	26.35	0.57	0.43	17.07	0.34
DeepAR	540	17.7	1.66	1.95	0.56	23.24	0.51	0.39	14.95	0.31
MQCNN	-	17.3	1.60	2.12	0.54	-	-	0.38	-	0.30
DeepSSM	871	22.4	2.10	2.20	0.67	29.51	0.64	0.49	27.33	0.40
OLS	1160	24.5	2.30	2.61	0.72	34.06	0.74	0.54	-	-

Table C.7: Evaluation measures forecasting algorithms for cluster A.3 Severe Accretion. MAE and RMSE are in meter. Some scores for MQCNN are left out as the calculation of these scores was different compared to the other algorithms. The MSE is here lower than for previous research. The MAE although, is relative larger. A lower MSE and higher MAE, indicate a lower maximum forecast error. Furthermore, all forecasting algorithms outperform OLS. The best algorithms here are DeepAR and MQCNN.

N = 5,097	MSE	MAE	MASE	MAPE	sMAPE	RMSE	NRMSE	ND	MSIS	CRPS
SimpleFFN	1594	29.0	2.31	0.53	0.32	39.92	0.37	0.27	21.61	0.21
DeepAR	1260	26.9	2.19	0.51	0.31	35.50	0.33	0.25	15.94	0.19
MQCNN	-	23.4	1.90	0.58	0.29	-	-	0.22	-	0.17
DeepSSM	2193	35.8	2.91	0.64	0.41	46.83	0.43	0.33	35.75	0.27
OLS	2854	39.3	3.17	0.95	0.49	53.42	0.49	0.36	-	-

Table C.8: Evaluation measures forecasting algorithms for cluster A.4 Extreme Accretion. MAE and RMSE are in meter. Some scores for MQCNN are left out as the calculation of these scores was different compared to the other algorithms. The MSE and MAE are here significant higher and thus forecasting transects with these algorithms does not lead to a better overall accuracy. The MAPE is low, indicating errors are low in percentage of magnitude. In addition, the high MSE and MAE score for DeepSSM, show that this algorithms is not compatible with this dataset. Furthermore, all forecasting algorithms, except DeepSSM, outperform OLS. The MQCNN is here the best algorithms.

N = 983	MSE	MAE	MASE	MAPE	sMAPE	RMSE	NRMSE	ND	MSIS	CRPS
SimpleFFN	3936	45.5	2.86	0.21	0.20	62.73	0.25	0.18	26.06	0.14
DeepAR	3096	41.9	2.73	0.19	0.18	55.64	0.22	0.17	17.95	0.13
MQCNN	2206	33.5	2.20	0.16	0.16	-	-	0.13	-	0.11
DeepSSM	45551	163.4	10.03	0.66	0.50	213.43	0.86	0.66	62.89	0.50
OLS	7587	64.5	4.21	0.33	0.36	87.11	0.35	0.26	-	-