NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

JOÃO PEDRO BOTA VARGUES

BSc in Computer Science

CHARACTERIZING TIME RANGES OF COASTAL UPWELLING VIA UNSUPERVISED CLUSTERING

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon

July, 2024

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

CHARACTERIZING TIME RANGES OF COASTAL UPWELLING VIA UNSUPERVISED CLUSTERING

JOÃO PEDRO BOTA VARGUES

BSc in Computer Science

Adviser: Susana Maria Nascimento
Assistant Professor, NOVA University of Lisbon

Examination Committee

Chair: Sérgio Marco Duarte
Assistant Professor, NOVA University of Lisbon

Rapporteur: Marta Belchior Lopes
Assistant Researcher, NOVA University of Lisbon

Adviser: Susana Maria Nascimento
Assistant Professor, NOVA University of Lisbon

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
July, 2024

Characterizing Time Ranges of Coastal Upwelling via Unsupervised Clustering

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.10.5) [29].

Acknowledgements

Gostaria de começar por agradecer à professora Susana Nascimento, cuja constante disponibilidade, apoio e orientação foram fundamentais ao longo de toda a realização desta dissertação. O seu encorajamento contínuo, reuniões semanais e orientação não só influenciaram o progresso deste trabalho, mas também conduziram-me pelo caminho mais correto, promovendo discussões sobre vários tópicos e melhorias para o projeto.

Um agradecimento especial à minha família e amigos próximos, que proporcionaram momentos de alegria e um apoio inabalável durante os períodos mais desafiantes. A sua motivação e encorajamento inspiraram-me a procurar os melhores resultados possíveis.

Ao longo da minha jornada na FCT-NOVA, tive o privilégio de conhecer pessoas inspiradoras, incluindo colegas e professores, que consistentemente me incentivaram e apoiaram nos momentos mais difíceis. Com eles, este percurso árduo tornou-se menos intimidante e mais gratificante e produtivo. Estou confiante de que as amizades estabelecidas durante este tempo serão guardadas para toda a vida.

Abstract

A popular topic in the scientific literature is oceanic dynamics. When surface water is pushed offshore by winds, deep ocean water is brought to the surface through upwelling, leading to various phenomena, such as coastal upwelling events. The analysis, study, and comprehension of these structures and events are essential in a variety of fields and can be achieved through remote sensing data obtained from sensors. The increasing availability of Spatio-Temporal (ST) data and the need to understand the upwelling routines have led to efforts to produce an automatic method to analyze the upwelling regions and their fronts, as manual analysis is time-consuming and costly.

In this dissertation, we extend the Core-Shell clustering framework developed in [32], [42]. We developed an experimental pipeline to compare popular partitional clustering algorithms, K-means, K-means++, DBA and Mean-Shift, with the Iterative Anomalous Pattern (IAP) algorithm in segmenting time series to find periods of coastal upwelling stability, referred to as time ranges. The study includes the development of an experimental procedure to fine-tune the hyperparameters of the clustering algorithms and the implementation of a stability score measure to comparatively evaluate the effectiveness of those clustering algorithms in finding time ranges. The Core-Shell algorithm is initialized with the most stable time ranges. The segmentations produced by the Core-Shell algorithm are validated using various state-of-the-art clustering validation indices. The approach is applied to three annual collections of Sea Surface Temperature (SST) images covering 16 years each (2004-2019), of the coasts of Portugal, northern Morocco, and southern Morocco.

Our results show that the IAP algorithm is an adequate choice for the segmentation of time series, obtaining the most stable time ranges and leading to the best Core-Shell clustering upwelling segmentations when evaluated by those validity indices. Applying DBA algorithm with features extracted from Core-Shell cores, it produces meaningful clustering and averaging time series with respect to the upwelling periods of stability. Analyzing the long-term inter-annual upwelling time series derived from the core segmentation showed that the Core-Shell algorithm obtains meaningful and consistent periods of stability.

Keywords: partitional clustering, time series segmentation, clustering validation, coastal upwelling, sea surface temperature images

Resumo

Um tópico popular na literatura científica é a dinâmica oceânica. Quando a água de superfície é empurrada para longe da costa pelo vento, água profunda do oceano é trazida à superfície através de afloramento, levando a vários fenômenos, como eventos de afloramento costeiro. A análise, estudo e compreensão destas estruturas e eventos é essencial em uma variedade de áreas e pode ser alcançada através de dados de sensoriamento remoto obtidos através de sensores. O aumento da disponibilidade de dados de temperatura da superfície do oceano e a necessidade de compreender as rotinas de afloramento têm levado a esforços para produzir um método automático para analisar as regiões de afloramento e as suas frentes, já que a análise manual é demorada e dispendiosa.

Nesta dissertação, estendemos o framework do Core-Shell clustering desenvolvido em [32], [42]. Desenvolvemos um pipeline experimental para comparar algoritmos populares de clustering por partição, K-means, K-means++, DBA e Mean-Shift, com o algoritmo IAP na segmentação de séries temporais para encontrar períodos de estabilidade de afloramento costeiro, referidos como time ranges. O estudo inclui o desenvolvimento de um procedimento experimental para ajustar os hiperparâmetros dos algoritmos de clustering e a implementação de uma medida de cálculo de estabilidade para avaliar comparativamente a eficácia desses algoritmos de clustering na descoberta de time ranges. O algoritmo Core-Shell é inicializado com os time ranges mais estáveis. As segmentações produzidas pelo algoritmo Core-Shell são validadas usando vários índices de validação de clustering comuns na literatura. A abordagem é aplicada a três coleções anuais de imagens de SST cobrindo 16 anos cada (2004-2019), das costas de Portugal, norte do Marrocos e sul do Marrocos.

Os nossos resultados mostram que o algoritmo IAP é uma escolha adequada para a segmentação de séries temporais, obtendo os intervalos de tempo mais estáveis e levando às melhores segmentações de afloramento Core-Shell quando avaliadas por esses índices de validação. Aplicando o algoritmo DBA com dados extraídos dos Core-Shell cores, este produz clusters significativos e médias de séries temporais relevantes em relação aos períodos de estabilidade do afloramento. Analisando as séries temporais de afloramento inter-anuais de longa duração derivadas da segmentação de cores, mostrou-se que o

algoritmo Core-Shell obtém períodos de estabilidade significativos e consistentes.

Palavras-chave: clustering por partições, segmentação de séries temporais, validação de clustering, afloramento costeiro, imagens de temperatura de superfície oceânica

Contents

List of Figures

Acronyms

AMI Adjusted Mutual Information *(pp. 36, 44, 48, 51, 62)*
ARI Adjusted Rand index *(pp. 36, 44, 48, 49, 52, 62, 63, 69)*
AVHRR Advanced Very High Resolution Radiometer *(p. 1)*

CH Calinski-Harabasz index *(pp. 46, 47, 62–65)*

DTW Dynamic Time Warping *(pp. 19, 21–24, 28, 29, 58, 64)*

ECG Electrocardiogram *(pp. 18, 32)*
EEG Electroencephalography *(p. 32)*
EL1 Elbow Point 1 *(pp. 46, 62–65)*
EL2 Elbow Point 2 *(pp. 46, 62–65)*

GMT Generic Mapping Tools *(p. 38)*

IAP Iterative Anomalous Pattern *(pp. iv, vi, xi, 3, 4, 36, 37, 39, 40, 44, 53, 58, 59, 61, 67–71, 79, 81)*

KDE Kernel Density Estimation *(p. 31)*
KS Kulczynski Similarity *(pp. 36, 44, 50, 69)*

LCSS Longest Common Sub-Sequence *(pp. 19, 21)*

MI Mutual Information *(p. 51)*
MODIS Moderate-Resolution Imaging Spectroradiometer *(p. 1)*
MSCD Mean Shift Cloud Detection *(p. 32)*

NMI Normalised Mutual Information *(pp. 50, 51, 69)*

PCA Principal Component Analysis *(p. 21)*
PSO Particle Swarm Optimisation *(p. 6)*

RI Rand index *(p. 49)*

S-STSEC Sequential Self Tunning Seeded Expanding Cluster *(pp. xi, 2, 3, 7, 37–40, 42, 43, 48, 53–57, 69, 81)*
SEC Seeded Expanding Cluster *(pp. 2, 7)*
SRG Seeded Region Growing *(pp. 2, 7, 39)*
SSB between-cluster sum of squares *(pp. 46, 47)*
SSC Sea Surface Chlorophyll *(pp. 2, 6)*
SST Sea Surface Temperature *(pp. iv, vi, xi, 1–7, 15, 37–40, 42–44, 48, 54–59, 61, 62, 71, 80, 81)*
SSW within-group sum of squares *(pp. 28, 47)*
ST Spatio-Temporal *(pp. iv, 2–4, 7–15, 37)*
ST-SEC Self Tunning Seeded Expanding Cluster *(pp. 2, 7, 39)*
SW Silhouette width index *(pp. 47, 53, 62–65)*

TPI Topographic Position Index *(p. 7)*

WB Wu and Bailey index *(pp. 47, 63–65)*
WKS Weighted Kulczynski Similarity *(pp. 36, 44, 48, 50, 69)*

XU Xu index *(pp. 47, 63–65)*

1 Introduction

1.1 The Problem and its Importance

A popular topic in the scientific literature is oceanic dynamics. When surface water is pushed offshore by winds, deep ocean water is brought to the surface through upwelling, leading to various phenomena, such as coastal upwelling events.

The temperature, nutrient content, and chemical composition of deep water differ from those of surface water. This phenomenon is important because it fuels an ecosystem with increased primary production, resulting in the appearance of larger populations of sea life and making its detection very important to the commercial fishery sector [45]. A visual representation of a coastal upwelling event is presented in Figure 1.1.

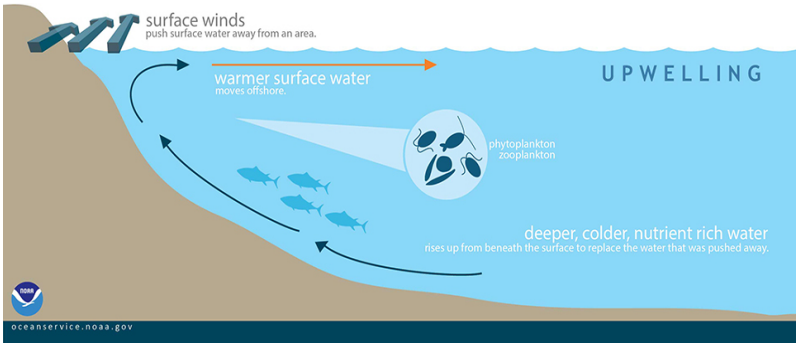


Figure 1.1: Representation of a coastal upwelling event. Image taken from [45].

Analysis, study and understanding of these structures and events are essential in a variety of fields, including fishing and coastal surveillance, research on climate change, identification of pollutants, and ocean movements. This can be accomplished by the growing accessibility of remote sensing data, mostly in the form of Sea Surface Temperature (SST) images obtained from sensors such as Moderate-Resolution Imaging Spectroradiometer (MODIS) or Advanced Very High Resolution Radiometer (AVHRR). In Figure 1.2, an example of a SST image of the Portuguese coast is presented.

Many research studies and methodologies have been created to tackle the challenge of automatically segmenting coastal upwelling from remote sensing images. The main

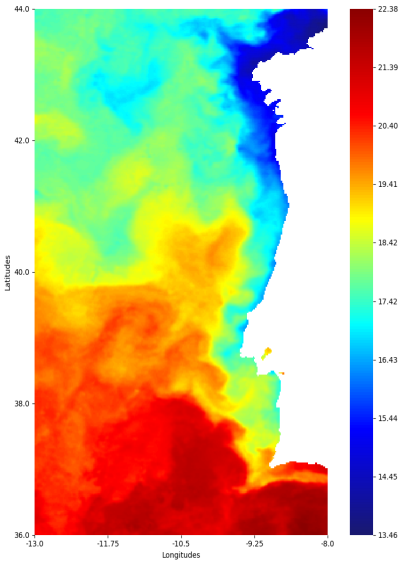


Figure 1.2: Example of a SST image of the Portuguese coast.

focus of this work is on unsupervised approaches, as supervised approaches require a significant manual analysis by the user, who must label and identify each data sample for training purposes, thereby making this approach both time-consuming and expensive. In El Aouni et al. [17], a method was proposed to segment the upwelling regions from SST and Sea Surface Chlorophyll (SSC) images obtaining accurate results throughout the Moroccan Atlantic coast. In the work by Nowicki et al. [46], a method was used to detect upwelling events in the Baltic Sea. Using SST images and wind data, a good correlation value was observed in the analysis of the upwelling area and the wind series.

Nascimento et al. [38] proposed the Seeded Expanding Cluster (SEC) algorithm, as an extension of the Seeded Region Growing (SRG) approach for automatic recognition of coastal upwelling from SST images and its self-tuning version Self Tunning Seeded Expanding Cluster (ST-SEC). This approach based its segmentation on the temperature of the pixels and also took into account their spatial context. It showed good and promising results in experiments conducted on SST images of the Portuguese coast. Additionally, the ST-SEC version makes use of an automatic computation of the homogeneity threshold, separating it from other adaptive thresholding approaches commonly employed in similar applications. The S-STSEC was introduced in [40], where the aforementioned work was improved and extended to iteratively extract clusters without specifying the final number of clusters to be extracted, thus allowing the segmentation of non-contiguous upwelling regions.

Numerous upwelling systems exhibit spatial and temporal variability and occur on a seasonal, monthly, and annual basis. The Spatio-Temporal (ST) data have the ability to understand the complexities of such occurrences and their dynamic patterns. Time series analysis, a technique for identifying changes in data over time, complements this effort. This approach, widely used in various fields, including medical monitoring,

climate change detection, speech recognition, image analysis, and human activity tracking, facilitates understanding of the intricate temporal patterns present in ST datasets.

Partition clustering has become a widely used technique to segment time series data into meaningful clusters. One significant benefit of these methods is their computational efficiency. In the work presented by Nascimento et al. [43], the IAP algorithm was used in the segmentation of time series data to find periods of temporal stability in an upwelling season. Each period found was designated as an upwelling time range. The Core-Shell clustering algorithm, implemented by Martins [32], receives as input sequential groups of SST images to delineate dynamic and static clusters, allowing for precise identification of coastal upwelling regions with remarkable accuracy.

One drawback of the Core-Shell clustering framework is that only a single clustering algorithm has been used for the unsupervised clustering of time series data, along with just two cluster validation indices to evaluate the effectiveness of the Core-Shell segmentations. In this dissertation, we developed an experimental pipeline to compare partitional clustering algorithms to cluster time series in an unsupervised way to find periods of coastal upwelling stability (referred to as time ranges), while implementing a stability score measure to analyze and validate the results obtained. This process is essential because using various algorithms in this phase will create different time ranges, which serve as the input for the Core-Shell algorithm. The validation of the Core-Shell clustering algorithm will be performed using the coastal upwelling segmentations of the SST images obtained by the S-STSEC algorithm as ground truth. Finally, a comprehensive experimental study will be conducted in various global regions, leading to the formulation of conclusive observations on the optimal approach.

1.2 Objectives of the Dissertation

This dissertation follows the work performed in [32], and the main goal is to extend the Core-Shell clustering framework. The main objectives of this dissertation are the following:

- To evaluate the effectiveness of the IAP algorithm in defining stable time ranges through time series segmentation, by comparing its results with those obtained from four partitioning clustering algorithms: K-means, K-means++, Dynamic Time Warping Barycenter Averaging (DBA), and Mean-Shift. An experimental protocol is developed where:
  - The clustering algorithms hyperparameters are fine-tuned;
  - A stability score measure is implemented to evaluate the stability of the time ranges obtained.

- To execute the Core-Shell algorithm with the optimal upwelling time ranges chosen based on the best stability scores and to validate the Core-Shell segmentations using various state-of-the-art validity indices.
- To examine the significance of the long-term inter-annual upwelling time series resulting from the segmentation of the upwelling cores when utilizing the DBA algorithm to generate average time series.

This will be performed using collections of SST images (covering a period of 16 years, from 2004 to 2019) in two distinct regions of the globe: Portuguese coast and the Atlantic coastal region of North Africa (North and South Morocco).

1.3 Organization of the Document

The remainder of the document is organized as follows. Chapter 2 describes various Automatic Recognition of Coastal Upwelling Approaches and their applications in the field. Chapter 3 begins with an introduction to Spatio-Temporal (ST) clustering, followed by a description of some of the most widely used approaches in the literature. Chapter 4 explains the topic of time series analysis along with a description of the algorithms used in our study: K-means, K-means++, DBA, and Mean-Shift. Chapter 5 presents the entire Core-Shell clustering framework followed by its proposed extension. In chapter 6, the experimental study is described, where every important and relevant analysis is performed and discussed, mainly the comparison study between IAP and the other algorithms used for clustering time series, and the validation and study of the Core-Shell clusters. Finally, the conclusions are available in Chapter 7.

2

Automatic Recognition of Coastal Upwelling: Unsupervised Approaches

Coastal upwelling is a phenomenon that has a significant impact on ocean productivity and fish populations. Comprehending and monitoring this procedure is essential for marine science and resource management. Advancement of automated recognition methods has transformed the study of coastal upwelling, allowing for a more complete understanding of its impacts on the marine ecosystem.

In the context of unsupervised methods, there is no requirement for training stages, which enables analysis of unlabeled image data. This can significantly help experiments and research activities, leading to a reduction in the amount of data needed for storage and processing.

Unsupervised methods for the study of coastal upwelling are based mainly on the analysis of SST images and use different approaches such as unsupervised classification, histogram-based separation, and fuzzy clustering algorithms to identify upwelling regions. Automation in the detection of this phenomenon has increased the accuracy and efficiency of monitoring this process [18]. There are multiple locations of significant coastal upwelling regions in the world, as presented in Figure 2.1, and throughout the rest of this chapter, we will introduce various approaches and studies that have been created.

In the work by Tamim et al. [58], a technique was developed to segment the upwelling region of the Moroccan Atlantic coast. This area is known for its upwelling activity, which is spatially variable and persists for most of the year. The method uses SST images and is carried out in two stages. Initially, the Otsu's method is used to delineate the upwelling region, and then a region-growing process is used to filter the results. Folowing the previous work, Tamim et al. [59] proposed an approach in which two algorithms, K-means and fuzzy c-means, were used to develop an automated tool for segmenting the upwelling region in SST images. A combination of cluster validity indices was used to identify the optimal number of clusters. Similarly to the previously mentioned approach, a region-growing process is applied to filter out noisy structures. The evaluation of the results was visually carried out by an oceanographer, demonstrating that the approach

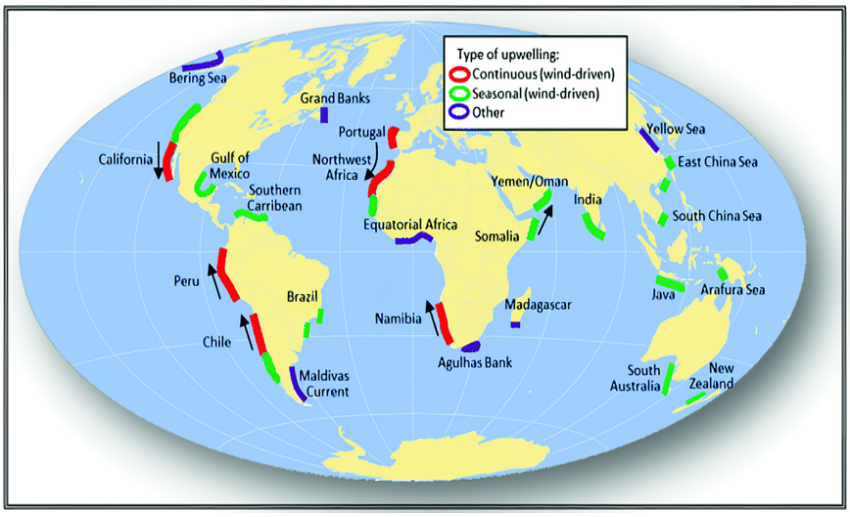


Figure 2.1: Locations of significant coastal upwelling regions in the world. Image taken from [23].

produces precise and reliable upwelling segmentations for the Moroccan Atlantic coast upwelling region.

In the work by El Aouni et al. [17], a method was proposed to delineate the upwelling regions on the entire Moroccan Atlantic coast using SST and SSC images while trying to address the problem of over segmentation that arises on the northern Moroccan coast. To achieve this, the images were divided into homogeneous regions and the Particle Swarm Optimisation (PSO) algorithm was applied to each. Subsequently, these smaller regions were merged to create the upwelling region. Following the work mentioned above, El Anouni et al. [18] proposed an improvement to the method by partitioning the images into a number of lines perpendicular to the coast, normalizing each line according to the maximum temperature offshore, and then applying the fuzzy c-means algorithm to the normalized images. This method was shown to be very accurate and robust in the results obtained along the entire coast of Morocco.

A study was conducted by Nowicki et al. [46] that aimed to detect upwelling events in the Baltic Sea. In the study, upwelling was identified by observing a temperature deviation of more than 2°C in specific pixels compared to the average temperature. This approach relies on spotting significant temperature decreases to indicate the presence of upwelling events. Moreover, wind data was employed to validate and gather additional details about the detected upwelling events.

Ramanantsoa et al. [51] carried out a study to analyze new discoveries on the structure, variability, and underlying factors of the coastal upwelling located south of Madagascar. An adaptive Canny edge detector was applied to multiple SST images to identify surface temperature fronts in the south of Madagascar. The study also investigated how winds and ocean currents affect upwelling. The results showed that coastal upwelling in this region can be defined in two clear different areas, each with their particular characteristics.

A study was developed by Shi et al. [56] to delineate the coastal upwelling in the

northern South China Sea. To map the upwelling region, SST images were used to calculate the Topographic Position Index (TPI), which is a local-based algorithm that calculates the difference between the center cell and its neighbors. Wind data was also used to analyze the mechanisms of coastal upwelling in this region. It was possible to observe in which areas of the northern South China Sea the coastal upwelling occurred more frequently.

A method called FuzzyUPWELL was developed by Nascimento et al. [41] for the automatic segmentation and recognition of coastal upwelling from SST images. The approach used a combination of a fuzzy clustering method and anomalous pattern initialization. Nascimento et al. [38] proposed the SEC algorithm, as an extension of the SRG approach for automatic recognition of coastal upwelling from SST images and its self-tuning version ST-SEC. This approach based its segmentation on the temperature of the pixels and also took into account their spatial context. It showed good and promising results in experiments conducted on SST images of the Portuguese coast. Following the previous work, in [40], a new method was proposed called S-STSEC. This approach is the sequential version of ST-SEC that extracts the clusters sequentially until a stop condition is reached, without specifying the total number of clusters. A novel clustering algorithm designated Core-Shell clustering was proposed by Nascimento et al. [42, 44] for the ST analysis of coastal upwelling. This algorithm is the focus of our work and is initially described in the context of ST data in Section 3.2.5, and its framework will be described in Section 5.1 in brief detail.

3

Spatio-Temporal Clustering

3.1 Introduction

Data with spatial and temporal information are referred to as Spatio-Temporal (ST) data. Numerous technologies such as remote sensing, mobile networks, and GPS devices generate large volumes of ST data. Dealing with the vast volume of data poses difficulties in terms of storing, organizing, analyzing, and knowledge discovery. ST data mining is the process of extracting implicit knowledge, spatial and temporal relationships, or similar patterns from ST data. There are multiple uses for ST data in fields such as ecology and climate, public safety, smart transportation, and human mobility [5].

Classical data mining techniques often struggle when applied to ST datasets for several reasons. First, ST data typically exist in a continuous space, unlike traditional datasets, which are often discrete. Second, ST data patterns involve both spatial and temporal characteristics, making them more intricate and challenging for traditional methods to capture the correlations effectively. Furthermore, traditional data mining methods assume that data samples are generated independently, which is not the case for ST data due to their high self-correlation [63].

Various techniques for ST data mining have been studied for more than a decade and can be divided into six methods: clustering, predictive learning, change point detection, frequent pattern mining, anomaly detection, and relationship mining [6].

ST clustering involves grouping objects based both on their spatial proximity and their temporal similarity. By combining geographic coordinates with temporal data, ST clustering provides a way to understand complex systems and facilitates pattern recognition in real-world scenarios [5].

3.1.1 Data Types

ST data types are classified according to two categories of information. Temporal information refers to the time component of the data, while spatial information refers to the placement and representation of objects in space. ST data can be categorised into five

types: event data, trajectory data, geo-referenced data items, geo-referenced time series, and moving objects [5].

- **Event data:** An ST event is typically defined by a specific point in space and time, indicating both where and when the event occurred. ST events can additionally have other non-spatial characteristics, commonly referred to as marked variables, that give each event additional information.
- **Trajectory data:** Trajectory data can be described by the paths followed by objects as they move through space during a specific time period. In addition to recording the sequence of positions traveled by each moving object over time, trajectory data may also include marked variables of the moving object.
- **Geo-referenced data item:** A geo-referenced data item is typically an observation of a specific phenomenon over time at a given location. Typically, only the most recent measurement is kept in this type of data.
- **Geo-referenced time series:** A geo-referenced time series is the measurement of a continuous ST field over a collection of stationary reference points throughout a specific time span. Data are recorded at specific time points that may either follow a regular interval between each recording or have varying gaps between them.
- **Moving objects:** A moving object changes its position in space over time and is assigned an identifier to track its movement over a period of time. Typically, only the latest positions are mantained, and there is no need to save previous locations. Normally, a moving object has an identification number to monitor the path of a particular individual.

3.1.2 Types of Spatio-Temporal Clustering

Analyzing ST data requires various steps, including pre-processing, transformation, data mining, and post-processing techniques to uncover meaningful patterns. Specifically, for ST clustering, it is crucial to conduct a careful data pre-processing, since irrelevant attributes negatively impact distance measures negatively and disturb the clustering patterns [5]. Following the explanation of each ST data type, we will now describe the ST clustering techniques.

- **Event clustering:** Event clustering refers to the discovery of clusters that are close in both space and time and may also have similar non-spatial characteristics. Clusters with a high number of ST event points are referred to as hot-spots.
- **Trajectory clustering:** Trajectory clustering involves selecting a suitable clustering algorithm and a distance measure. The shape of the cluster is determined by the algorithm chosen. The similarity of ST trajectories varies depending on the particular application.

- **Geo-referenced data item clustering:** Geo-referenced data item clustering involves identifying clusters that present similar spatial features at a specific time. It is also used to identify clusters of items with similar non-spatial characteristics at a specific time moment.
- **Geo-referenced time series clustering:** Geo-referenced time series clustering of objects requires analyzing how the time series of objects evolve with respect to the spatial positions of the objects.
- **Moving clusters:** Moving clusters refer to clusters of data points that change their position and composition over time while maintaining their identity. In this approach, the main idea is to discover groups of data points that are similar to each other at different time points.

3.1.3 Areas of Application

With the growth in ST data collection, several approaches and studies were developed in areas such as social media, health care, agriculture, transportation, and climate science. We will briefly describe the applications of ST data in different domains.

Climate and Weather Science: In weather data, different weather sensors are installed at fixed or floating sites to measure atmospheric and oceanic values such as wind speed, temperature, air quality, and precipitation. Since climate data from multiple locations can present high ST correlations, various ST clustering algorithms can be used in weather forecasting [63]. A new methodology called DcSTCA was proposed by Liu et al. [27] to mine and cluster fluctuations patterns in maritime anomalies. A ST grid cube is built, where the proximities are calculated in the attributes of space, time, and theme, and then ST clustering patterns are derived.

Neuroscience: Multiple brain imaging technologies are widely used in the domain of neuroscience, and the spatial and temporal values of brain activity measured by these technologies vary significantly [63]. Doborjeh and Kasabov presented [16] a new clustering method that used ST brain data based on the NeoCube spiking neural network architecture. This clustering method, in order to understand the brain's functional dynamics, uses the spiking neural network architecture and functional analysis to create and analyze clusters.

Crime Data: In the days we live in, it is common for cities to make their crime data publicly available for research purposes, and these data usually contain attributes such as the type of crime, the time and place of the crime, and some other information [63].

Nakaya and Yano [37] presented a new approach to analyze ST crime patterns in Kyoto by creating a three-dimensional mapping of crime occurrences in a space-time cube. In the work by Hu et al. [21] a ST framework to predict hotspots and evaluate crimes in Louisiana was presented. With a kernel density estimator to calculate the density of an event and a predictive accuracy index to evaluate predictive accuracy, it was possible to identify crime hotspots.

Epidemiology: Traditional epidemiology surveillance comprises a set of epidemiological procedures that monitor the spread of a disease and determine how it spreads [19]. Gomide et al. [19] presented a method using regression models and ST-DBSCAN that was successful in active surveillance of the dengue epidemic by looking at Twitter information. In the work by Alkhamis et al. [3], it was possible to explore the ST dynamics at the beginning of the COVID-19 pandemic in Kuwait using a time-dependent reproductive numbers model and a multi-variable permutation scan statistic implemented in SatScan.

3.2 Spatio-temporal Clustering Methods

This section provides an analysis and description of multiple ST algorithms. At the end of the section, we present Table 3.2.6 with the advantages and disadvantages of each algorithm explored.

3.2.1 ST-DBSCAN

A new clustering algorithm based on DBSCAN was proposed by Birant and Kut [9] with the new ability to find clusters according to non-spatial, spatial, and temporal attributes of objects.

In DBSCAN, the density associated with a point is determined by counting its neighbor's points regarding a region predetermined by a radius given as a parameter. Points that have a density higher than a threshold are then grouped as clusters.

The ST-DBSCAN algorithm brings three major improvements over its predecessor in the processing of ST data: First, it can cluster data with non-spatial, spatial, and temporal attributes. Second, it solves the problem that DBSCAN has of not being able to catch noise points when clustering with various densities. This is resolved by giving a density factor (a measure of how densely packed the points in a cluster are) to each cluster. Third, in cases where neighboring objects exhibit slight variations in their values and there exists a significant difference in the values of boundary items within a cluster in comparison to those of boundary items on the opposite side, the ST-DBSCAN algorithm addresses this problem by examining the average value of a cluster in comparison to a new value.

The algorithm has four user defined parameters:

- **Eps1** is the maximum distance between two data points in order for them to be classified as part of the same cluster in terms of spatial attributes (latitude and longitude).
- **Eps2** is the maximum distance between two data points in order for them to be classified as part of the same cluster according to their non-spatial attributes.
- **MinPts** is the minimum number of data points that must be in a region for that area to be considered a dense area.
- $\boldsymbol{\Delta\epsilon}$ is used to control the discovery of combined clusters due to slight differences in non-spatial attributes for neighboring locations.

The algorithm starts by choosing the initial point in the dataset and retrieves all points that can be reached by density according to Eps1 and Eps2 from that point. If it is a core object, which means that it has at least a MinPts number of neighbors within Eps1 and Eps2, then a new cluster is formed, and all neighboring objects that can be directly reached by density are marked as members of the new cluster. If it is a border object, that is, it is not a core object but is density reachable from one, the next point in the dataset is used.

Following the completion of the initial step, the algorithm proceeds to choose the subsequent point in the dataset and continues with the process of discovering density reachable objects from core objects, until all points in the dataset have been processed. As a result, in the end, the algorithm finds a set of clusters and noise points (points that do not belong to any cluster).

3.2.2 ST-OPTICS

Agrawal et al. [2] proposed a new clustering algorithm based on OPTICS with the new improvement to discover clusters according to non-spatial, spatial, and temporal attributes of objects.

The OPTICS algorithm does not directly form clusters. Instead, it arranges data points in a way that is favorable for creating a clustering structure based on density. Typically, OPTICS is employed to establish this ordering, which is subsequently utilized as input for an Agglomerative algorithm to achieve a more insightful visualization. It is important to note that the original OPTICS algorithm does not inherently accommodate ST data.

The ST-OPTICS algorithm has three user-defined parameters, two radiuses, ϵ_1 (spatial distance) and ϵ_2 (non-spatial distance), and MinPts, similar to ST-DBSCAN. In order to account for the temporal dimensions, the data are initially combined by preserving the temporal neighbors along with their associated spatial dimension [2].

The framework for ST-OPTICS, as illustrated in Figure 3.1, is divided into two phases:

1. The clustering phase, responsible for generating clusters.
2. The agglomerative phase that combines the clusters created at the micro-level during the clustering phase.

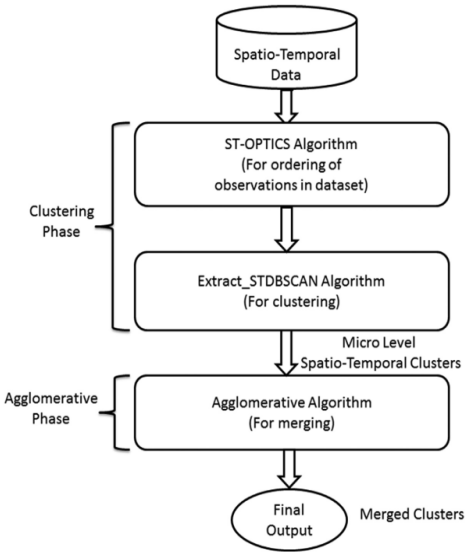


Figure 3.1: ST-OPTICS framework. Image taken from [2].

Initially, the ST data are provided as input to the ST-OPTICS algorithm. The main goal of this algorithm is to reorganize all the items in the dataset, ensuring that the items with similar features are grouped together.

Subsequently, these rearranged items are fed into the Extract_STDBSCAN (modified version of the DBSCAN) algorithm for clustering purposes. Since the resulting clusters are at a micro-level (meaning there is a large number of clusters, making the interpretation process difficult), an agglomerative phase is employed to facilitate visualization and interpretability.

The agglomerative algorithm is responsible for combining clusters that can be merged. In [2], two different types of clustering algorithms, namely density-based and hierarchical-based algorithms, were used to merge clusters in order to improve the analysis, visualization and interpretation of the resulting clusters efficiently.

3.2.3 CorClustST

Hüsch et al. [22] proposed an ST clustering algorithm, Correlation-based clustering of big spatio-temporal datasets (CorClustST), a novel correlation-based clustering algorithm to be applied to large ST datasets.

This algorithm addresses the drawbacks of some ST clustering algorithms such as ST-DBSCAN and ST-OPTICS. These algorithms do not directly identify meaningful cluster centers, which can affect the further understanding of cluster relationships, especially for the purpose of big data reduction. Furthermore, these clustering algorithms have multiple parameters that need to be fine-tuned to obtain an optimal clustering result based on specific optimization criteria.

CorClustST can detect clusters that are easier to understand, regardless of the number of time points, by calculating the correlations between spatial points over time. In ST data,

positive correlation is typically observed only up to a specific spatial range, making it unnecessary to calculate correlations for all pairs of points. By focusing solely on objects within a defined spatial proximity, the computational time and memory demands of the clustering method are reduced.

This clustering algorithm starts by computing the Pearson sample correlation coefficient between all spatial points within a distance (ϵ) from each other. After that, the algorithm determines the number of spatial neighbors for each point that has a correlation value greater than a predefined value. These points are called ST neighbors and are sorted in descending order according to the number of ST neighbors.

The algorithm groups the points according to their ST neighbors. At first, the point with the most neighbors is selected as the center of the initial cluster, and then all its neighbors are assigned to this cluster. For the remaining points, the algorithm checks if each point is not already part of a cluster and if at least 50% (a user defined threshold) of its ST neighbors are also not assigned to a cluster. If these conditions are met, the point becomes the center of a new cluster. Subsequently, its neighbors are assigned to this new cluster and further checks are made to ensure that these points do not belong to other clusters. If a neighbor strongly correlates with a new center point, it is reassigned to this cluster. This iterative process continues until all points are successfully assigned to a cluster.

3.2.4 Stable Clusters algorithm

Chen et al. [11] proposed a novel ST clustering approach to cluster in a continuous ST field, where clusters are dynamic and can vary in their size, shape, position and statistical properties between consecutive time intervals.

The clustering approach is divided into the following steps:

1. Specify the clustering objectives, such as separating a specific activity from the background or labeling the clusters with target labels.

2. Identify the "core points," which are points that maintain consistent cluster memberships within a given time window.

3. Finalize the memberships of the cluster along the boundary of the cluster.

4. Post-process the cluster result if all points have not been labeled.

The algorithm implemented to test the approach is an extension of DBSCAN, which detects "core points" having at least m neighbors within a distance ϵ in the feature space. It considers neighboring points with comparable feature values within a defined time window. By incorporating spatial and temporal information, the technique identifies Eps neighbors similar to the ST-DBSCAN method presented in [9]. This strategy identifies clusters at every time interval and associates these clusters over time, regardless of noise or missing data.

3.2.5 Core-Shell Clustering

The Core-Shell clustering algorithm, introduced by Nascimento et al. [42, 44], is a ST clustering approach inspired by the methodology proposed in [11]. It takes advantage of the observation that, in many fields, certain "core points" consistently maintain their cluster memberships over a specific time window, even as the clusters themselves change. In contrast to the DBSCAN approach that identifies dense regions in the data distribution, Core-Shell employs a distinct strategy to explicitly differentiate between core and boundary points, which is reflected in the notions of "core" and "shell" clusters.

Using the concept of the Core-Shell structure, a simplified model is proposed to identify the upwelling patterns in SST images. Consider a constant "core" that initially forms rapidly and then gradually expands into the offshore waters up to certain physical boundaries before eventually returning back to the core region. This allowed Core-Shell clustering to establish a clustering criterion and to utilize the least-squares method to estimate SST data. Consequently, the key parameters of the Core-Shell cluster model are automatically determined as minimizers of the criterion. This aspect sets this approach apart from other ST clustering algorithms, where key parameters such as thresholds and the number of clusters to extract from the data need to be arbitrarily selected [43]. A summary description of the Core-Shell clustering framework is presented in Chapter 5 (Section 5.1).

3.2.6 Summary

A summary comparison of the advantages and disadvantages of the previously described ST clustering algorithms is now presented.

Comparing Cluster Algorithms	
ST-DBSCAN	Advantages: • The number of clusters does not have to be pre-determined • Able to discover clusters with arbitrary shape • Unusual observations are declared as noise points Disadvantages: • Very large databases need extreme computing power • User-defined determination of four parameters significantly impacts clustering quality. The amount of parameter selection can potentially lead to suboptimal results • Cluster centers are not directly provided

ST-OPTICS	Advantages: • Able of recognizing nested and neighboring clusters and managing multi-dimensional data • The number of clusters does not have to be pre-determined • Unusual observations are declared as noise points • Improved performance compared to ST-DBSCAN Disadvantages: • Very large databases need extreme computing power • Does not support spatial indexing structures • Cluster centers are not directly provided
CorClustST	Advantages: • The number of clusters does not have to be pre-determined • Unusual observations are declared as noise points • Meaningful cluster centers are provided and facilitate data reduction and the analysis of interconnections between clusters Disadvantages: • A particular quality criterion is not taken into account when optimizing the clustering solution • Higher complexity than ST-DBSCAN and ST-OPTICS for large values of ϵ

Stable Clusters	Advantages: • The number of clusters does not have to be pre-determined • Associates clusters across time, regardless of noise and missing data Disadvantages: • No structured method to determine the appropriate size of the time window
Core-Shell	Advantages: • Defined by a clustering criterion based on least squares minimization that allows deriving key parameters of the algorithm in an automatic way • Associates clusters across time, regardless of noise and missing data Disadvantages: • The algorithm was specifically designed for the application in coastal upwelling analysis • Approach lacks a robust validation methodology

4

Time Series Analysis

4.1 Introduction

A time series is a sequence of data points or observations collected, recorded, or measured at successive points in time. These data points are typically ordered chronologically and are often taken at evenly spaced intervals. Time series data can be classified into two types: univariate, which involves measuring only one variable over time, and multivariate, which involves observing multiple variables simultaneously at each time point. These data points can be used to track changes in a particular variable over time, such as changes in stock prices, weather patterns, or other data types [4].

Figure 4.1 presents an example of Electrocardiogram (ECG) measurements that have been partitioned into segments, each representing a single heartbeat cycle. The ECG time series is divided into sequences that correspond to individual heartbeats, and dashed red lines indicate the boundaries of these cycles. This segmentation process is applied to both regular and irregular ECG readings. By analyzing other ECG measurements, it is possible to detect patterns that indicate whether the heartbeat is regular or irregular [57].

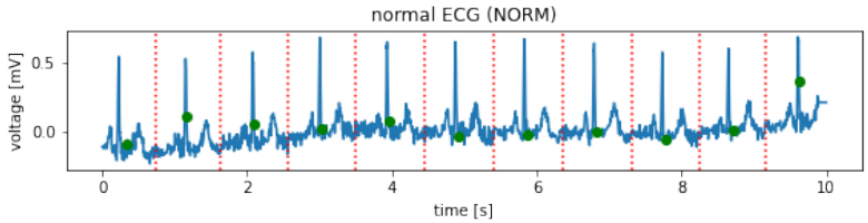


Figure 4.1: ECG time series segmented into heartbeat cycles. Image taken from [57].

The storage of time series datasets has increased significantly due to advancements in data storage and processors. This has opened up new opportunities for the analysis of time series data, leading to the development of numerous projects and research efforts in different fields. These advances have also facilitated the creation and improvement of techniques for effectively analyzing time series data.

Clustering has become a widely used approach in time series analysis because of its ability to uncover patterns and structures in temporal datasets. Unlike traditional static

data clustering, time series clustering considers the sequential nature of data, making it particularly advantageous in various scientific domains and real-world applications. By grouping similar temporal behaviors together, this approach enables the discovery of frequent and rare patterns, facilitates anomaly detection, and improves the understanding of complex processes [1].

The definition of clustering time series can be defined as follows:

Given a dataset of n time series data $D = F_1, F_2, ..., F_n$, the task of unsupervised partitioning of D into clusters $C = C_1, C_2, ..., C_k$, to group homogeneous time series according to a certain measure of similarity i is named time series clustering [57].

The similarity measure in this definition refers to the process of calculating and matching time series throughout the dataset. The computation of distances involves the calculation of all the lengths of time series. However, this procedure is complicated due to the inherent noise, outliers, and shifts present in time series data. Some distance metrics were designed to specify the similarity between time series, such as Dynamic Time Warping (Dynamic Time Warping (DTW)) and Longest Common Sub-Sequence (LCSS).

In order to emphasize the importance and necessity of clustering time series datasets, the objectives for clustering such data are presented below [1]:

- **Uncovering Valuable Information:** Time series databases store valuable information that can be extracted by discovering patterns. Clustering is a widely used approach to analyze these patterns in time series datasets.
- **Dealing with Large Datasets:** Due to the size of time series databases, manual inspection becomes impractical. To effectively manage these voluminous datasets, users often prefer to work with structured data. Clustering allows the representation of time series data as groups of similar sequences, achieved through the aggregation of data into non-overlapping clusters.
- **Exploratory Technique:** Time series clustering is a commonly used technique in data analysis, often used for exploration purposes. It is also used as a sub-component in more intricate data mining algorithms such as rule discovery, indexing, classification, and anomaly detection.
- **Visual Representation for Better Understanding:** Transforming time series cluster structures into visual images, also known as visualization of time series data, offers a simple comprehension of data structure, clusters, anomalies, and regularities present within datasets. This visualization helps users gain a deeper understanding of the underlying patterns of the data.

However, time series clustering poses significant challenges. First, the large size of time series data often exceeds memory capacity, leading to a considerable drop in clustering speed. Second, the high dimensionality of time series data complicates their handling for many clustering algorithms, resulting in slow clustering processes. Third, the

selection of appropriate similarity measures to form clusters is complicated. Time series similarity matching involves calculating the similarity among entire time series using a distance measure. However, this process is difficult due to the presence of natural noise, outliers, and shifts in the time series data. Additionally, variations in time series lengths make distance calculation difficult, making the choice of a suitable similarity measure a substantial challenge.

Time series clustering algorithms share the common goal of adapting existing static data clustering algorithms to accommodate time series data or transforming time series data into a static data format suitable for direct use with existing algorithms. The former approach involves working directly with raw time series data, often referred to as the raw data based approach. In this approach, the main change consists in replacing the distance or similarity measure utilized for static data with a suitable one designed for time series data. However, the alternative method starts by transforming raw time series data into a lower-dimensional feature vector or a set of model parameters. Then, a traditional clustering technique is used on these extracted feature vectors or model parameters [26].

Time series clustering approaches can be categorized into three main methods [1]: shape-based, feature-based, and model-based.

- **Shape-Based Approach:** In the shaped-based approach, the main goal is to match the shapes of two time series as closely as possible through non-linear stretching and contracting of the time axes. This technique, also known as the raw data based approach, works directly with the raw data of the time series. Shape-based techniques usually apply traditional clustering techniques, originally intended for static data. However, these methods are adjusted by integrating appropriate distance or similarity metrics designed to time series data.

- **Feature-Based Approach:** The feature-based method includes the conversion of the original time series data into lower-dimensional feature vectors. These vectors are subsequently utilized as inputs for traditional clustering algorithms. In this method, a feature vector of equal length is computed from each time series, typically followed by the utilization of the Euclidean distance metric. By doing so, the data's dimensionality is decreased, making it more manageable for clustering algorithms to handle.

- **Model-Based Approach:** In the model-based methodology, raw time series data are converted into model parameters by employing a parametric model for each time series. Following this, an appropriate measure of model distance and a clustering technique (typically conventional clustering techniques) are chosen and utilized on the derived model parameters. However, model-based approaches are known to encounter challenges related to scalability and their effectiveness tends to diminish when clusters are close to each other.

In general, time series clustering comprises four essential components: dimensionality reduction or representation method, distance measurement, clustering algorithm, and prototype definition. Each component plays a crucial role in the clustering process, and the selection and combination of these components depend on the specific characteristics and requirements of the time series data being used.

In the next section, the topic of distance measures will be explored in more detail, by also explaining two commonly used distance measures in time series clustering.

4.2 Distance Measures

Time series clustering heavily depends on the use of distance measures. Various distance measures are available to calculate the distance between time series data. Some measures are versatile and can be applied regardless of the representation method used, or even with raw time series data.

In traditional clustering, the distance between static objects is precisely determined, whereas, in time series clustering, distances are approximated. This approximation is particularly important when comparing time series with irregular sampling intervals and varying lengths. There exists a range of distance measures designed to measure the similarity between time series. Among them, popular options include the Hausdorff distance, the modified Hausdorff, the HMM-based distance, DTW, the Euclidean distance, the Euclidean distance in a Principal Component Analysis (PCA) subspace, and LCSS [1].

One of the simplest approaches to compute the distance between two time series is to consider them as univariate time series and then determine the distance between each time point. In this section, we will describe two distance measures in time series clustering, the Euclidean distance and DTW.

4.2.1 Euclidean distance

Euclidean distance measures the distance between two time series by computing the straight-line distance between their data points in a multi-dimensional space. Given two univariate time series of equal lengths, $S = s_1, s_2, ..., s_i, ..., s_n$ and $T = t_1, t_2, ..., t_i, ..., t_n$, the definition of the Euclidean distance measure is as follows [8]:

$$d(S,T) = \sqrt{\sum_{i=1}^{n} (s_i - t_i)^2}. \tag{4.1}$$

In time series analysis, it is commonly a good practice to standardize the time series either on a global or local scale to accommodate significantly varied ranges [8]. When using the Euclidean distance for clustering time series, this measure calculates the dissimilarity between these vectors by computing the square root of the sum of the squared differences between their corresponding data points. Smaller Euclidean distances indicate greater similarity between time series, whereas larger distances indicate greater dissimilarity.

4.2.2 Dynamic Time Warping distance

Dynamic Time Warping (DTW) is a distance measure specifically designed for time series data. Figure 4.2 shows how the DTW and Euclidean distances match two time series, given by the red and blue lines. In this measure, the initial and final points of the time series must be in alignment, but all other points can align differently in time to find better matches. Although DTW is computationally expensive, it is generally better than the common Euclidean distance when comparing time series. Therefore, the use of DTW could be a more effective method to assess the similarity between two time series that are not aligned in time, speed, or length [12].

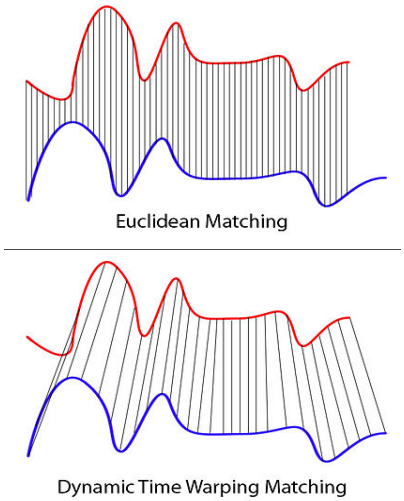


Figure 4.2: Dynamic Time Warping and Euclidean matching two time series. Image taken from [12].

A formal definition, as given by Mizutani [35], outlines a distance measure between two time series S and T. Let $S = s_1, s_2, \ldots, s_i, \ldots, s_n$ and $T = t_1, t_2, \ldots, t_j, \ldots, t_m$ be the time series. The distance measure $\text{Dist}(S, T)$ is defined as the sum of the absolute distances:

$$\text{Dist}(S,T) = \sum_{i=1}^{\min\{N,M\}} ||S_i - T_i|| + \begin{cases} \sum_{i=\min\{N,M\}+1}^{N} ||S_i||, & \text{if } N > M \\ 0, & \text{if } N = M \\ \sum_{i=\min\{N,M\}+1}^{M} ||T_i||, & \text{otherwise.} \end{cases} \tag{4.2}$$

Here, $\min\{N, M\}$ ensures that the sum includes elements up to the length of the shorter sequence between S and T. The distance $d(i, j)$ between elements S_i and T_j is defined as the absolute distance:

$$d(i,j) = ||S_i - T_j||. \tag{4.3}$$

Absolute distance measures the difference between the corresponding elements of the sequences. Provides a measure that directly quantifies the differences between elements.

DTW offers a more flexible approach to measuring similarity or dissimilarity between patterns. DTW aligns the two sequences so that their differences are minimized. The minimum difference implies the lowest cost between S and T under the following assumptions:

Assumption 1: The first and last elements of the sequences must be in a match, that is,

$$\begin{cases} (1) & S_1 \sim T_1 \text{ with a distance } d(1,1) \\ (2) & S_N \sim T_M \text{ with a distance } d(N,M). \end{cases} \tag{4.4}$$

Assumption 2: Three movements are considered to allow variations in the sum of $d(i,j)$ even when $N = M$. These movements follow a transition rule from position (i,j), consisting of the following three actions:

$$\begin{cases} (1) & \text{"expansion" action } (\rightarrow) \text{ from } (i-1,j) \text{ to } (i,j) \\ (2) & \text{"match" action } (\nearrow) \text{ from } (i-1,j-1) \text{ to } (i,j) \\ (3) & \text{"contraction" action } (\uparrow) \text{ from } (i,j-1) \text{ to } (i,j). \end{cases} \tag{4.5}$$

A matrix $n \times m$ where the element (i,j) of the matrix contains the distance $d(s_i, t_j)$ (the Euclidean distance is usually used) between two points s_i and t_j. The minimum distance between the two time series is composed of a warping path, $W = w_1, w_2, ..., w_k$, where $max(m,n) \leq K \leq m+n-1$, is a group of matrix elements. This group of elements satisfies three constraints: boundary condition, continuity, and monotonicity. The boundary condition constraint mandates that the warping path must begin and end in diagonally opposite corner cells within the matrix. In other words, it must start at $w_1 = (1,1)$ and end at $w_k = (m,n)$. The continuity constraint further narrows down the permissible steps to neighboring cells. Additionally, the monotonicity constraint enforces that points along the warping path must be spaced in a monotonic fashion with respect to time. The primary focus lies in identifying the warping path that minimizes the distance between the two series. It can be defined as:

$$d = min \frac{\sum_{k=1}^{K} w_k}{K}. \tag{4.6}$$

Dynamic programming offers an effective method to determine this path by calculating the cumulative distance using the following recurrence. This formula defines the total distance as the sum of the current element's distance and the smallest total distance among its neighboring elements:

$$d(i,j) = d(s_i, t_j) + min\{d(i-1, j-1), d(i-1, j), d(i, j-1)\}. \tag{4.7}$$

This minimum cumulative distance between two time series is their DTW distance value.

4.3 Clustering Methods for Time Series Segmentation

4.3.1 K-Means

A partitioning clustering method creates K clusters from a set of n unlabeled objects, ensuring that each cluster contains at least one object. One of the most popular cluster partitioning algorithms is K-means clustering [28, 31]. This algorithm is an unsupervised learning method that partitions a dataset into K distinct, non-overlapping clusters, where each data point belongs to the cluster with the nearest mean (centroid).

The hyperparameter K represents the number of clusters to identify in a dataset and is a user-defined parameter. It is a crucial decision when using K-means because it directly impacts the granularity and quality of the clustering. The pseudo-code of K-means is presented below.

Algorithm 1 K-means Algorithm

procedure KMEANS(X, K) ▷ X: Dataset, K: Number of clusters
  Initialize clusters $C_1, C_2, ..., C_K$ as empty sets
  Randomly initialize K cluster centroids $\mu_1, \mu_2, ..., \mu_K$
  while not converged **do**
    for $x \in X$ **do**
      Assign x to the nearest centroid: $c \leftarrow \arg\min_k \|x - \mu_k\|^2$
      Add x to cluster C_c
    end for
    for $k \leftarrow 1$ **to** K **do**
      Update centroid: $\mu_k \leftarrow \frac{1}{|C_k|} \sum_{x \in C_k} x$
    end for
  end while
  return $\{C_1, C_2, ..., C_K\}, \{\mu_1, \mu_2, ..., \mu_K\}$ ▷ Clusters and centroids
end procedure

The K-means algorithm approach follows a procedure called Expectation-Maximization. The Expectation-step (or Assignment step) is the assignment of the data points to the nearest cluster, while the Maximization-step (or Update step) is the computation of the centroid of each cluster.

The clustering criterion for K-means is defined as follows [13]:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|x^i - \mu_k\|^2, \tag{4.8}$$

where $w_{ik} = 1$ for data point x^i if it belongs to cluster k. If it does not belong to cluster k, then $w_{ik} = 0$. The μ_k is the centroid of x^i's cluster.

The algorithm starts by selecting the initial centroids K randomly chosen from the dataset or using some other heuristic method. These centroids serve as initial cluster centers. Then, for each data point, the distance is calculated (typically using Euclidean distance) between that data point (x^i) and each of the centroids (μ_k). The data point is assigned to the cluster associated with the nearest centroid. This step creates K clusters on the basis of the current centroids. Mathematically, this step can be defined as:

$$\frac{\partial J}{\partial w_{ik}} = \sum_{i=1}^{m}\sum_{k=1}^{K} \|x^i - \mu_k\|^2 \Rightarrow w_{ik} = \begin{cases} 1 & \text{if } k = argmin_j \|x^i - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases} \tag{4.9}$$

After this step, the centroids of each cluster are recalculated by computing the mean of all data points assigned to that cluster. The new centroids become the central points for the next iteration. The update step is defined as:

$$\frac{\partial J}{\partial \mu_k} = 2\sum_{i=1}^{m} w_{ik}(x^i - \mu_k) = 0 \Rightarrow \mu_k = \frac{\sum_{i=1}^{m} w_{ik} x^i}{\sum_{i=1}^{m} w_{ik}}. \tag{4.10}$$

The algorithm then iterates between the assignment and update steps until the convergence criteria are met. Convergence can be defined by different conditions, such as when there is no significant change in the centroids or when a predetermined number of iterations have been completed. Upon reaching convergence, the algorithm produces K clusters, with each data point assigned to a specific cluster.

Given a time series dataset, the K-means clustering algorithm computes the distances between all pairs of time series using a distance metric such as the Euclidean distance. Afterwards, it designates one of the time series within the cluster as the prototype, specifically the one with the lowest sum of squared errors [12]. Additionally, when the distance metric is non-elastic, such as the Euclidean distance, or when the cluster centroid can be determined, it can be affirmed that the prototype corresponds to the time series nearest to the centroid. In Figure 4.3, the result of a clustering of K-means is shown using the Euclidean distance as the base metric, where in red the centroid is observable, and in each subfigure a cluster of ranges is shown.

Before analyzing the best way to select the best K, it is important to explore the various methods to validate the quality of partitions in the clustering process. Cluster validation measures can be divided into two approaches: external clustering indices and internal clustering indices. The main distinction between the two lies in the utilization of external information for cluster validation. External validation indices make use of information not present in the data to evaluate if the clustering partitions/structure obtained by a clustering algorithm match an external partition/structure (e.g., a specified class label). Alternatively, internal validation measures evaluate the goodness of clustering partitions/structure obtained without making use of any external information [65].

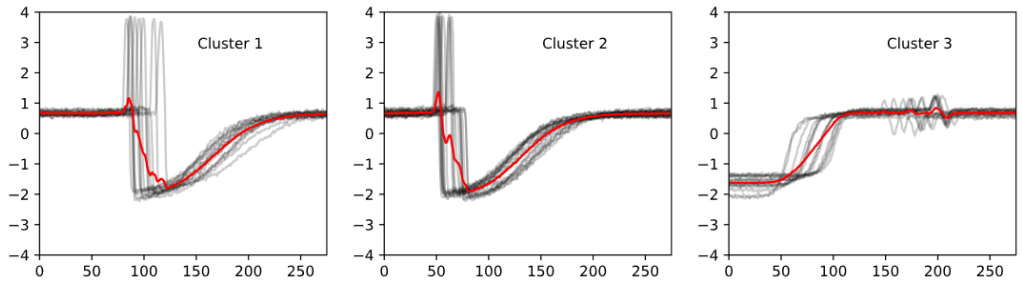


Figure 4.3: Example of K-means clustering with Euclidean distance applied to a dataset of time series data. Image taken from [60].

Both external and internal validation indices play a crucial role in various application scenarios. External validation indices, which possess knowledge of the "true" cluster count in advance, serve the purpose of selecting the most suitable clustering algorithm for a specific dataset. For example, when external validation metrics demonstrate that a clustering algorithm can produce results that closely align with the categorization performance achieved by human experts, it provides strong evidence of the algorithm's practical efficacy in clustering. On the other hand, internal validation metrics come into play when determining the optimal clustering algorithm and cluster count without relying on additional external information. In practice, external information, such as class labels, may not be available for some real-world applications. In such instances, internal validation metrics become the only viable option for cluster validation, as they do not depend on external information [65].

To determine the optimal number of clusters for a given dataset, internal validation indices such as the Calinski-Harabasz index, the Silhouette index, and the Davies-Bouldin index can be used. The usual procedure to determine the optimal cluster number of a dataset using internal validation indices involves initializing a set of clustering algorithms with different parameters, computing the corresponding internal validation index for each obtained partition, and selecting the best partition and optimal cluster number based on predefined criteria [54].

In our work and analysis, we will focus on internal validation indices to better estimate the best number of clusters for the K-means clustering algorithm. The usual procedure to determine the optimal cluster number of a dataset using internal validation indices is as follows [54]:

1. Specify a range of K values ($K = 2, \ldots, 7$ for example);

2. For every K within the specified range, run K-means with multiple initializations where random K entities are selected as initial centroids each time. The resulting local minimum inertia values can be considered as an approximation to the global minimum inertia value $D(K)$ for the dataset under consideration;

3. Internal validation indices are used to calculate the optimal K values based on the minimum (or maximum) value of the respective index.

Choosing the right value of K is a crucial step in applying K-means effectively, as an inappropriate choice of K can lead to either over-segmentation or under-segmentation of the data.

4.3.2 K-means++

The K-means algorithm described above picks the first centers at random. One disadvantage of this algorithm is its sensitivity to the initialization of centroids or mean points. If a centroid is initialized as a "far away" point, it might end up with no points associated with it, while more than one cluster might end up linked with a single centroid. Similarly, poor initialization can result in multiple centroids in the same cluster, leading to suboptimal clustering [14].

K-means++ chooses the initial centroids that are more likely to be close to the true cluster centers, leading to faster convergence and better overall clustering results. Essentially, K-means++ is the standard K-means algorithm coupled with a more intelligent initialization process for the centroids. The K-means++ pseudocode is presented below [14].

Algorithm 2 K-means++ Algorithm

procedure K-MEANS++(X, K) ▷ X: Dataset, K: Number of clusters
  Initialize clusters $C_1, C_2, ..., C_K$ as empty sets
  Select centroids $\mu_1, \mu_2, ..., \mu_k$:
    1.1. Choose one center μ_i uniformly at random from X.
    1.2. Take a new centroid μ_i, for each data point x, calculate $D(x)$, the distance between x, and the closest centroid that has just been selected. The data point $x \in X$ is then selected with a probability of $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$.
    1.3. Continue with Step 1.2 iteratively until a total of K centers have been selected.
  while not converged **do**
    for $x \in X$ **do**
      Assign x to the nearest centroid: $c \leftarrow \arg\min_k \|x - \mu_k\|^2$
      Add x to cluster C_c
    end for
    for $k \leftarrow 1$ **to** K **do**
      Update centroid: $\mu_k \leftarrow \frac{1}{|C_k|} \sum_{x \in C_k} x$
    end for
  end while
  return $\{C_1, C_2, ..., C_K\}, \{\mu_1, \mu_2, ..., \mu_K\}$ ▷ Clusters and centroids
end procedure

After initializing the centroids, K-means++ proceeds similarly to the standard K-means algorithm. It iteratively assigns each data point to the nearest centroid and updates the

centroids according to the mean of the data points assigned to each cluster.

By following this initialization procedure, the centroids are selected from the data points in such a way that they are far from each other. This increases the likelihood of selecting centroids from distinct clusters at the beginning. Moreover, as centroids are selected from the data points of the dataset, each centroid will have some data points associated to it by the end of the initialization process.

4.3.3 Dynamic Time Warping Barycenter Averaging

Petitjean et al. [49] presented in a global averaging method known as Dynamic Time Warping Barycenter Averaging (DBA). The objective of DBA is to reduce the total sum of squared DTW distances between the average sequence and the entire set of sequences. This sum is formed by the individual distances between each coordinate of the average sequence and the coordinates of the associated sequences. Therefore, the contribution of a coordinate of the average sequence to this total sum is actually the sum of Euclidean distances between the coordinate and the associated sequences during the DTW computation. The fundamental principle of DBA is to determine each coordinate of the average sequence as the barycenter of its associated coordinates from the set of sequences. Consequently, each coordinate aims to minimize its share of the within-group sum of squares (SSW) to ultimately minimize the total SSW. The updated average sequence is defined once all barycenters are calculated.

Consider the following barycenter function [47]:

$$\text{barycenter}(\{X_1, X_2, \ldots, X_l\}) = \frac{X_1 + X_2 + \ldots + X_l}{l}. \tag{4.11}$$

If we consider each point in the set as an n-dimensional vector, the addition sign corresponds to vector addition. Given a set S, the sequence i of length m in the set is a succession of points:

$$S_i = s_{i1} s_{i2} \ldots s_{im}, \tag{4.12}$$

where each point is a vector in a n-dimensional space:

$$s_{ij} = (s_{ij1}, s_{ij2}, \ldots, s_{ijn}). \tag{4.13}$$

The average sequence A of length m' is written as follows:

$$A = a_1 a_2 \ldots a_{m'}, \tag{4.14}$$

where:

$$a_j = (a_{j1}, a_{j2}, \ldots, a_{jn}). \tag{4.15}$$

For each point of A, a set of all points in the sequences of S is created so that the points are associated when applying DTW between A and the sequences of S. The created space can be defined as:

$$\text{assoc}(a_j). \tag{4.16}$$

The DBA algorithm averages each point of A individually by replacing it by the barycenter of all its associated points. In other words, at each iteration of the DBA algorithm, we perform the following transformation for each point a of A:

$$a_j = \text{barycenter}(\text{assoc}(a_j)). \tag{4.17}$$

The DBA method then consists of the following steps:

1. Create an initial average sequence A. The sequence can be random, preprocessed, or even an extracted sequence from the data.

2. For each sequence S from the data, compute the DTW algorithm between S and A. For each point a from A, create a set $s(a)$ containing all points associated with a from all sequences of the DTW algorithm.

3. For each point a from the average sequence A, compute the barycenter of all points from the $s(a)$ set, coordinate by coordinate. The resulting barycenter becomes the new point a, thus modifying all points in the sequence A.

4. Repeat steps 2 and 3 until the average sequence A is stable (does not change during step 3) or until a determined number of iterations.

5. A is the average sequence returned.

This computational method is expensive due to the nature of the DTW calculations that must be performed. However, it proves to be highly competitive when using the DTW distance metric. An advantage lies in its ability to directly accommodate time series of varying lengths [55].

In concomitant DBA clustering, the K-means algorithm and Dynamic Time Warping Barycenter Averaging (DBA) are integrated so that the centroid update step in K-means is replaced by the DBA algorithm. This integration allows for simultaneous refinement of both cluster assignments and centroid values [47].

The concomitant DBA clustering method follows the following steps:

1. Select k sequences $S_1, S_2, \ldots, S_k$ from the dataset as centroids, denoted $C_1, C_2, \ldots, C_k$.

2. For each sequence S in the dataset, compute the DTW algorithm between S and each centroid. Associate each sequence with the closest centroid on the basis of the minimum DTW computed value.

3. For each centroid C, compute the barycenter of each point from C with its associated set of points from sequences associated with C in step 2. The sequence consisting of successive barycenters becomes the new centroid C.

4. Repeat steps 2 and 3 until all centroids are stable (that is, do not change during step 3) or until a predetermined number of iterations.

5. The k centroids represent centroids of k sets of sequences, forming the clusters.

In future chapters, we will refer to the concomitant DBA clustering method only as DBA for simplicity in the explanations. In Figure 4.4, the result of the concomitant DBA clustering is presented, similar to Figure 4.3 previously presented for K-means with Euclidean distance. By comparing the two figures, it is possible to observe that the prototypes of the concomitant DBA clustering better capture the variances of the time series.

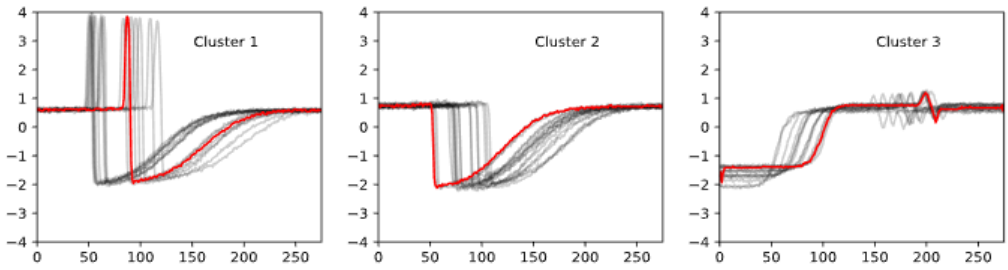


Figure 4.4: Example of concomitant DBA clustering applied to a dataset of time series data. Image taken from [60].

4.3.4 Mean-Shift

Mean-shift is a non-parametric density estimation and clustering algorithm. It is used to find modes, or local maxima, in a dataset by iteratively shifting points towards areas of higher density. This process is repeated until convergence, resulting in a collection of modes that can be used to determine clusters in the data. Mean-shift is especially helpful when the clusters in a dataset have complex, non-convex shapes, as it does not require the user to specify the number of clusters or initialize the algorithm with arbitrary starting points [10].

In Mean-Shift clustering, the input to the algorithm is the data points and the bandwidth. The bandwidth (window size) determines the size of the local region used for density estimation. Taking the $\{X_n\}_{n=1}^{N} \subset \mathbb{R}^D$ data points that will be clustered, a kernel density estimate can be defined as [10]:

$$p(x) = \frac{1}{N} \sum_{n=1}^{N} K\left(\left\| \frac{x - x_n}{\sigma} \right\|^2 \right) \qquad x \in \mathbb{R}^D, \tag{4.18}$$

with bandwidth $\sigma > 0$ and kernel $K(t)$, e.g. the Gaussian kernel, $K(t) = e^{-t/2}$. There are some practical situations where each data point has its own weight and bandwidth, but in the explanation bellow, we will take on the case where all the points have the same bandwidth (σ) and weight ($\frac{1}{N}$) value, since this is the most common case used in practice. The Gaussian kernels will also be used since this are the most simple to analyse and used in multiple scenarios.

An iterative scheme can be derived $x^{\tau+1} = f(x^{\tau})$ for $\tau = 0, 1, 2...$ to find the modes of p by calculating its gradient to zero and rearranging the terms, the following is obtained:

$$f(x) = \sum_{n=1}^{N} \frac{K'\left(\left\|\frac{x-x_n}{\sigma}\right\|^2\right)}{\sum_{n'=1}^{N} K'\left(\left\|\frac{x-x_{n'}}{\sigma}\right\|^2\right)} x_n, \tag{4.19}$$

where $K' = dK/dt$ and the vectors $f(x) - x$ are the mean-shift, since it averages the individual shifts $x_n - x$ with the weights described above. Using the Gaussian kernel, $K' \propto K$, and the Bayes' theorem (where $p(n|x) = p(x|n)p(n)/p(x)$ is the posterior probability of the component centered at x_n given a point x), the equation is simplified to the following:

$$p(x|n) = \frac{\exp(-\frac{1}{2}(\|\frac{x-x_n}{\sigma}\|^2)}{\sum_{n'=1}^{N} \exp(-\frac{1}{2}(\|\frac{x-x_n}{\sigma}\|^2)} \qquad f(x) = \sum_{n=1}^{N} p(n|x)x_n. \tag{4.20}$$

In the applications of Mean-shift clustering algorithm, each mode of p represents one cluster, and each data point x_n converges to a mode under the mean-shift iteration.

The main parameter in this algorithm is the bandwidth, which determines the number of clusters. The field of statistics has developed various methods to estimate the bandwidth in Kernel Density Estimation (KDE), often focusing on the one-dimensional context. These approaches may involve minimizing specific loss functions such as the mean integrated squared error or following heuristic rules such as setting the bandwidth proportionally to the average distance of each point to its nearest neighbor k^{th}.

Different types of kernels lead to distinct variations of the Mean-Shift. The Epaechnikov kernel is popular due to its computational efficiency. This efficiency comes from the fact that kernel evaluations involve only pairs of neighboring points, as opposed to assessing all possible pairs of data points. Additionally, convergence becomes known within a finite number of iterations when this kernel is used. Nevertheless, in practical applications, it has been observed that the Gaussian kernel outperforms the Epanechnikov kernel. The Gaussian kernel produces KDEs that are smoother and more continuous than those generated by the Epanechnikov kernel. KDEs created with the Epanechnikov kernel may have differentiability and potentially contain false modes [10].

There are some challenges in using the Mean-Shift clustering algorithm. For example, some data points may not converge to a mode at all, or the algorithm may not converge to the true mode. Additionally, the algorithm is typically run for a fixed number of iterations, which means that data points that would theoretically converge to the same mode may end up at slightly different positions. To address these issues, post-processing steps can

be used, such as merging data points that are close together into a single cluster. The user should also set the tolerance level for iteration and the distance threshold for proper merging of points [10].

In the work by Qian et al. [50], a data-driven technique called Mean Shift Cloud Detection (MSCD) was proposed to analyze multi-temporal image data. This method capitalized on the insight that the reflectance of the underlying landscape typically remains consistent over extended periods, whereas the sporadic appearance of clouds causes significant fluctuations in pixel values, marking them as outliers. Consequently, the task at hand involved identifying modes within the data and detecting outliers. The Mean-Shift algorithm was employed to locate the modes within the pixel value time series, subsequently identifying and excluding outliers situated far from these modes, which correspond to cloud cover.

4.4 Change Point Detection

One common approach in time series analysis is change point detection, which is the process of identifying changes in the data where some pieces of the time series data have altered, usually indicating that a significant event or change in the system or the data has occurred [30].

Looking at Figure 4.5, an example of gait analysis, the motions of a patient are observed with accelerometers and gyroscopes while performing simple activities such as walking or standing still. The result as can be observed in the flowchart is a series of non-overlapping parts, each one corresponding to a specific activity, and each one has its gait characteristics. But to extract features from the segments identified, preliminary processing of the signals before analysis is needed, therefore, the need for change point detection.

Change point detection covers a wide spectrum of real-world applications, such as [4]:

- **Medical condition monitoring:** Real-time automated monitoring of a patient's health necessitates the identification of trends in physiological data including heart rate, Electroencephalography (EEG), and ECG. Change point detection has been the subject of research studies addressing various medical difficulties such as sleep disorders, epilepsy, MRI interpretation, and brain activity analysis.

- **Climate change detection::** The increasing concern over climate change and the elevated levels of greenhouse gases in the atmosphere have made change point detection indispensable in climate research, monitoring, and forecasting in recent years.

- **Speech recognition:** Speech recognition involves converting spoken speech into text. Change point detection techniques are employed for audio segmentation to differentiate between silence, sentences, words, and noise.

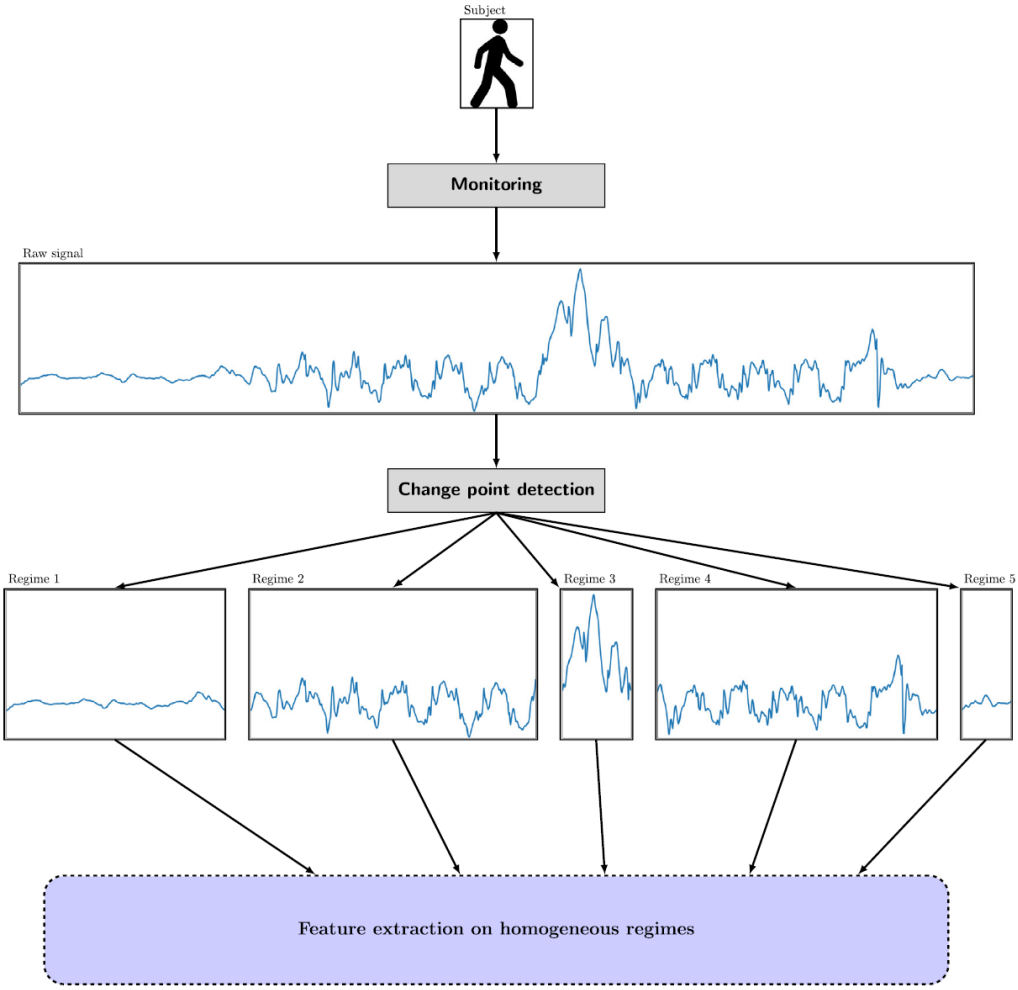


Figure 4.5: Flowchart of a study scheme, for gait analysis. Image taken from [61].

- **Image analysis:** Researchers and professionals collect image or video information continuously for video surveillance purposes. An example of a change point problem is identifying sudden occurrences such as security breaches, with each moment representing a digital image encoding.

- **Human activity analysis:** Based on sensor data gathered from mobile devices or smart homes, change point detection can be used to identify pauses or changes in activity. These change points are helpful for segmenting tasks, reducing interruptions when interacting with people, providing activity-based services, and identifying behavioral changes that may indicate a person's health status.

Change point detection techniques can be grouped into two categories: online methods and offline methods. Online methods are designed to detect changes as they occur in real time. However, offline methods are used to retrospectively detect changes after all data have been collected. There are various unsupervised approaches to time series segmentation, such as statistical models, kernel-based methods, graph-based methods,

and clustering techniques. These methods can be divided into three main categories: top-down, bottom-up, and sliding windows [30].

- **Top-down approach:** Top-Down approach, also referred to as divide and conquer method, is used for segmenting non-segmented time series data. It starts by observing the data as one main segment and identifying the most suitable location to divide the data into two segments in a way that maximizes the difference between them. Then, these segments are tested for the level of approximation error. If the error is below a user-defined threshold, the segmentation process stops and the segment is accepted. If the error is above the threshold, the segment is divided into two new segments, and the process is repeated. The algorithm continues to divide the segments until certain stopping criteria are met. The main disadvantage of this algorithm is its inflexibility, as the breaking points determined in previous iterations remain unchanged until the end of the process.

- **Bottom Up Approach:** The Bottom-Up approach, also known as the iterative merge algorithm, is a complement to the Top-Down algorithm. It starts by separating a time series into many small segments of equal length, then it merges pairs of consecutive segments that cause the smallest increase in error. The algorithm repeats these steps until a stopping criterion, such as a specified number of segments or an approximation error above a threshold, is met.

- **Sliding Window Approach:** Sliding window approach is a segmentation process that begins by defining the left border of the first potential segment and uses a sliding window to identify and select segments that meet a predefined criterion. The size of the window gradually increases as it moves along the time series, and segments are formed when the error of the potential segment becomes less than the user-specified threshold. The algorithm repeats this process until the entire time series is segmented.

The identification of change points can be seen as a clustering issue, in which data points within the same cluster exhibit similarities, while those across neighboring clusters show differences. When the data at time stamp t belong to a different cluster than at time stamp $t + 1$, a change point is identified. The computational complexity of algorithms for detecting change points is a crucial aspect to take into account. Characterizing the cost of supervised methods is particularly challenging due to the intricate nature of the involved complexities [4].

Change point detection algorithms have been extensively studied, but less focus has been given to evaluating them using actual time series data from real-life scenarios. Novel algorithms are often evaluated using simulated time series with known change points, which may not accurately represent real-world dynamics. Additionally, evaluations based on a small sample of real-world time series can be biased due to preprocessing or lack

of ground truth. Post hoc analysis is commonly used, but it is considered neither fair nor accurate for assessing new methods. Evaluation metrics in change point detection include clustering metrics, such as information variation and segmentation metrics, and classification metrics, focusing on precision, recall, and the F1 score [62].

Keogh et al. [24] introduced the SWAB algorithm by combining the bottom-up and sliding window approaches. This novel method combines online features of the sliding window algorithm with the precision of the bottom-up method, enabling piecewise linear approximation. In particular, SWAB operates efficiently with constant space requirements, by producing high-quality linear approximations of the provided data.

5

The Core-Shell Clustering Framework and its Extension

The Core-Shell clustering framework implemented by Martins [32] [1] will be extended as follows:

1. Model time ranges with various clustering algorithms and compare with IAP algorithm:
    - Evaluate the performance of the IAP algorithm in grouping time ranges and compare with K-means, K-means++, DBA, and Mean-Shift clustering algorithms.
    - Use inertia-based and internal validity indices for hyperparameter tuning and selecting optimal partitions.
    - Assess the stability of obtained time ranges across upwelling years for each algorithm.
2. Evaluation of Core-Shell clusters:
    - Assess the quality of Core-Shell upwelling segmentations from the results of IAP, K-means, K-means++, Mean-Shift, and DBA.
    - Use five cluster validity indices Adjusted Rand index (ARI), Adjusted Mutual Information (AMI), Kulczynski Similarity (KS), Weighted Kulczynski Similarity (WKS) and Mirkin distance.
3. Analysis of long-term inter-annual upwelling time series:
    - Construct datasets with distinct core segmentation characteristics: core intensity and core total area.
    - Apply DBA algorithm to produce meaningful clustering and averaging time series with respect to the stability periods derived from Core-Shell's core.

[1] https://github.com/thisDash/CoreShellClusteringAlgorithm

The rest of this chapter is organized as follows: We start with a brief description of the Core-Shell Clustering framework followed by the motivation for the extension.

5.1 Core-Shell Clustering Framework

In the work developed by Nascimento et al. [43], a novel ST clustering approach was introduced for the automatic recognition and analysis of coastal upwelling. The proposed method was based on the concept of a Core-Shell structure for upwelling recognition and tracking from SST data.

In Figure 5.1 it is possible to observe the workflow of the Core-Shell clustering framework implemented by Martins [32]. This framework is defined by six main steps briefly described next:

1. Preprocessing N SST grids (from an upwelling season) using a preprocessing pipeline to obtain preprocessed SST grids;

2. These preprocessed SST grids are given as input to the S-STSEC algorithm that in an unsupervised way automatically segments the upwelling regions;

3. Four features are extracted from the automatically segmented coastal upwelling regions to create a time series. The features are: total coastal upwelling area, average temperature, latitude of the northernmost region and latitude of the southernmost region;

4. Time series are in an unsupervised way segmented by the IAP to group consecutive SST instants with similar behaviour into upwelling time ranges;

5. Each collection of consecutive T SST instants belonging to a time range is then given as input to the Core-Shell clustering algorithm. This algorithm will produce a Core-Shell cluster with a constant part, the core that defines the regular part of the upwelling region, and T shells, each characterizing the dynamic parts of the upwelling regions;

6. Finally, there are Core-Shell cluster time series with features extracted (average temperature, total area) from the core and the shell parts. Inter-annual analysis of those time series allows for studying trends of coastal upwelling in different regions of the world.

5.1.1 Image Preprocessing

To improve the quality of SST images, a pre-processing procedure is applied to every selected SST grid. The preprocessing methodology comprises three steps:

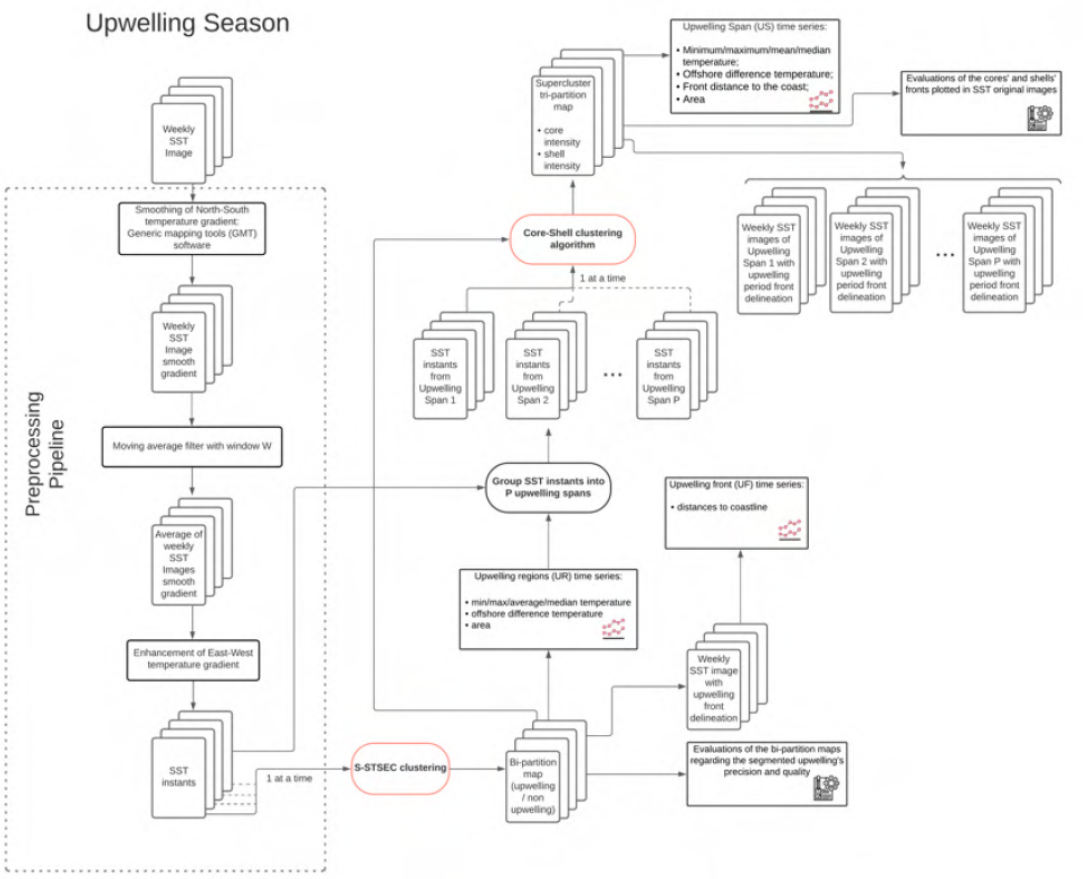


Figure 5.1: Full experiment pipeline. Image taken from [32].

1. **Removing the North-South temperature gradient:** Higher latitudes receive less solar energy than lower latitudes, resulting in a gradient. The Grdtrend module of the Generic Mapping Tools (GMT) software is used to eliminate it [64].

2. **The moving average filter:** This uses the sliding window algorithm as an average filter. By doing this, any white noise that could have been in the SST data, the result of measurement errors and atmospheric events, is eliminated.

3. **Enhancement of East–West temperature gradient:** The method normalizes the temperatures of the SST grid along the shorelines by adapting a technique from [18]. Using grid rows as perpendicular shorelines, the temperature of each point is adjusted relative to the row's average temperature. This prevents over-segmentation in S-STSEC clustering, producing preprocessed SST-averaged grids for analysis.

From this point on, every SST averaged grid produced during the pre-processing phase is referred to as an SST instant.

5.1.2 S-STSEC algorithm

Determining the "upwelling front", which divides the oceanic waters offshore from the colder upwelled waters along the coast, is essential to determining the coastal upwelling pattern. ST-SEC [38], an unsupervised spatial clustering technique, extracts coastal upwelling zones.

The widely used SRG technique is expanded upon by the ST-SEC method, which has been shown to overcome the acknowledged limitations of SRG algorithms, within the context of "anomalous clustering" [34]. First, an adaptively optimized threshold is derived from the clustering criterion, which uses a product's format instead of the traditional difference between a pixel and the region of interest's mean. Second, the cluster growth process is regularized using a moving window in the method.

The ST-SEC algorithm starts with the coldest pixel in a given SST grid as its initial seed and gradually expands the region as long as the evaluated pixels meet an automatically determined similarity condition. When there are no more pixels that meet this requirement, a bipartition grid is produced that depicts the upwelling and non-upwelling areas of the grid.

The S-STSEC, introduced in [40], is an iterative version of ST-SEC that extracts clusters one by one until a stop condition that derives the number of clusters in an unsupervised way.

5.1.3 Finding Time Ranges: Periods of Upwelling Stability

The goal in this phase is to determine the temporal stability segments of an upwelling season, which are groups of successive SST instants that share similar upwelling attributes. Each group is designated as an upwelling time range. To achieve this purpose, the Iterative Anomalous Pattern (IAP) [34] was used.

The IAP algorithm was used to identify groups of SST instants in an unsupervised manner in [43] and found 3 to 4 sets of successive SST instants for each upwelling season.

In Figure 5.2, the results of the application of the IAP algorithm to the SST instants obtained from the segmentation of S-STSEC for the year 2019 are presented, as discussed in [43]. This step is important because it obtains groups of SST instants that will be the input of the Core-Shell clustering algorithm. In Figure 5.3, one time range with T instants leads to T Core-Shell clusters, with a common core (highlighted in orange) and T shells (highlighted in green). In Figure 5.2, it is observed that each SST instant provided to the Core-Shell clustering algorithm is extracted from the upwelling range 3 of the year 2019, serving as one of the four inputs for the analysis of that particular year.

The IAP algorithm automatically determines the number of clusters to be obtained from the data. Clusters are extracted from the dataset sequentially, one at a time, as explained in Table 5.1 [39].

The total data scatter $T(Y)$ of all data points in the standardized dataset Y is defined

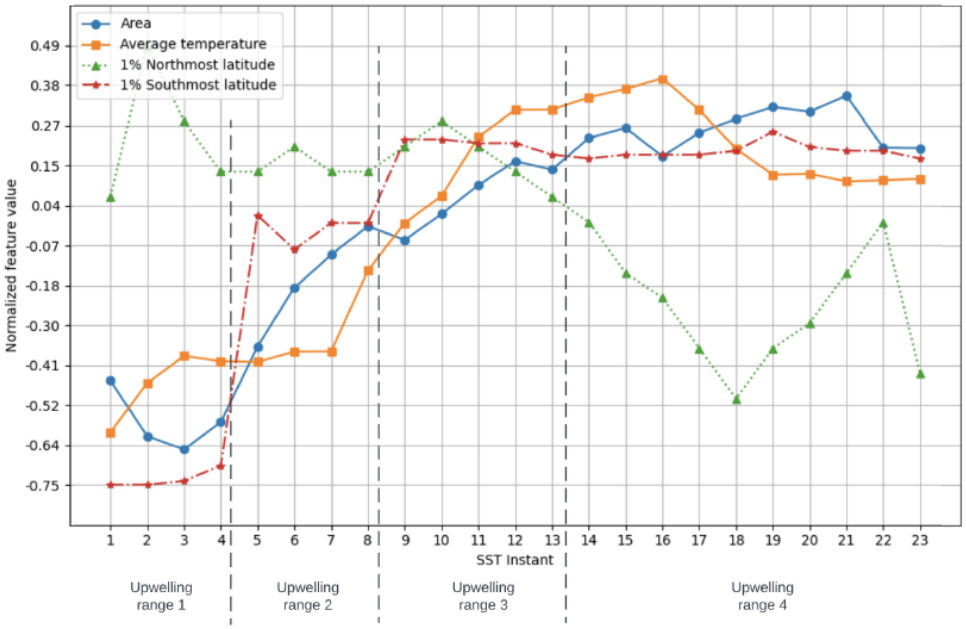


Figure 5.2: SST instant ranges obtained by IAP for time series extracted from S-STSEC segmentations [43].

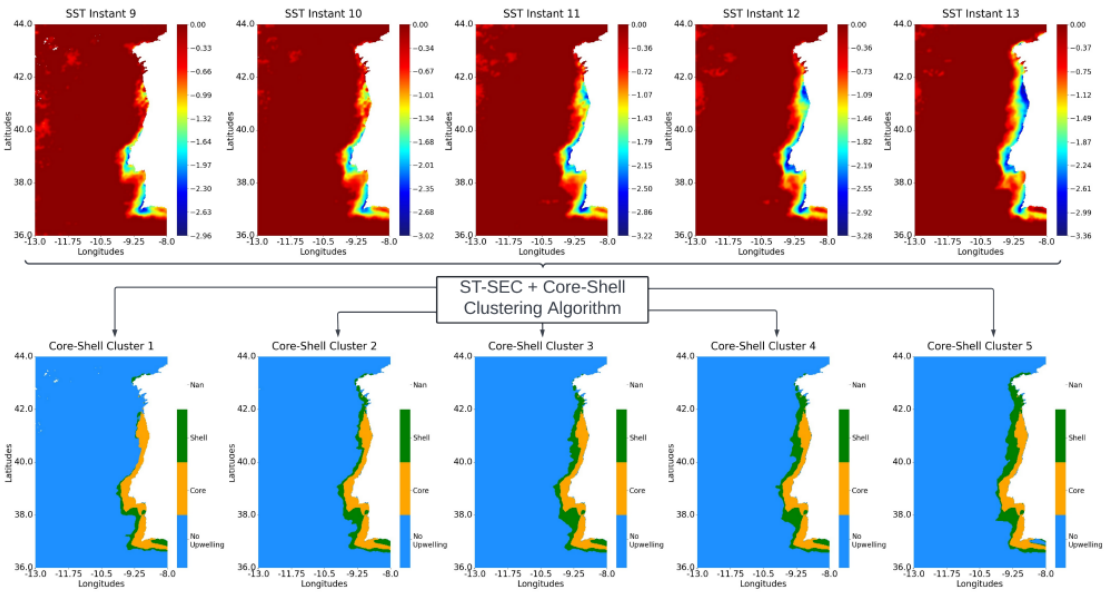


Figure 5.3: Core-Shell clustering example [43].

Iterative Anomalous Pattern:

1. Preprocessing: Let Y denote the standardized dataset obtained by shifting the origin of the original data X to the mean x.
2. Initial setting: The feature vector x is taken as the unvaried reference point throughout the sequential process and takes as the seed point the data point that is farthest from the reference point x.
3. Cluster update: Construct a cluster C_t iteratively, defined as the set of points closer to the seed point than to the reference point.
4. Centroid update: Substitute the cluster seed of C_t by the gravity center of the cluster and repeat the procedure until convergence.
5. Stop condition check: The procedure continues on the residual data set $Y_{t+1} = Y_t - C_t$ until any of the following conditions are met:

1. **All entities are clustered:** This condition is satisfied when there are no remaining data points to cluster.
2. **Cumulative contribution:** The algorithm ends when the cumulative contribution of the clusters exceeds a predefined threshold. This means that the clusters have collectively captured a sufficient amount of the data's variance or information.
3. **Relative contribution:** If the individual contribution of a cluster is deemed too small, the algorithm stops. This condition ensures that clusters with minimal significance are not further subdivided.
4. **Cardinality of clusters:** The algorithm may conclude when the number of extracted clusters reaches a predefined threshold. This condition prevents excessive segmentation, ensuring that the algorithm does not create an excessive number of clusters.

6. Output: Outputs the centroids c, the saved sets S, and additional useful data.

Table 5.1: Explanation of Iterative Anomalous Pattern

as:

$$T(Y) = \sum_{i=1}^{N} \sum_{h=1}^{D} y_{ih}^2, \tag{5.1}$$

where $N \times D$ matrix Y represents the entity-to-feature data matrix.

In the work by Mirkin [34], it is shown how $T(Y)$ can be decomposed into an explained part due to the retrieved cluster structure and an unexplained part corresponding to the clustering criterion of K-means. The relative contribution of each cluster (C, v) to the data scatter is defined as:

$$W((C, v)) = \frac{\sum_{h=1}^{n} v_h^2}{T(Y)} = \frac{\sum_{i=1}^{N} \sum_{h=1}^{D} y_{ih}^2}{\sum_{h=1}^{n} v_h^2}, \tag{5.2}$$

where n is the cardinality of cluster C.

5.1.4 Core-Shell Clustering

The entire model is established on a dynamic Core-Shell clustering algorithm that integrates the segmentation results of the S-STSEC algorithm. This model is unlike other approaches, since it does not use the DBSCAN methodology to collect dense portions of the data allocation. Instead, a different procedure is followed to differentiate the core points from the boundary points. Using the concept of Core-Shell structure, Nascimento et al. [43] proposed a model for the recognition and tracking of upwelling of SST maps. A description of the Core-Shell clustering based on [43] is as follows:

The model:

Given a preprocessed SST grid, $A^t(I, J) = (a^t_{ij})$, with a^t_{ij} being the temperature value at point (i, j), where i is the longitude and j the latitude and t is a timely moment in a period.

Two non-overlapping sets, $R \cup S^t$, of binary values, compose a Core-Shell cluster, with $r_{ij} \in R$ being the core and $s^t_{ij} \in S^t$, being the shell at a time t, such that the multiplication of a core point and a shell point at the same longitude and latitude at any time period t is equal to 0.

Assuming that the shells are represented by their intensity values, λ^t, the intensity value of the core should always be greater than the intensity value of the shell, and then the core intensity is represented by $\lambda^t + \mu^t$ with $\mu^t > 0$. The model that defines an upwelling sea surface at a moment t can be defined as

$$a^t_{ij} = (\lambda^t + \mu^t)r_{ij} + \lambda^t s^t_{ij} + e^t_{ij}, \tag{5.3}$$

where e^t_{ij} is the residual value that must be minimised according to the least squares criterion.

$$\Delta = \sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{j=1}^{J}(a^t_{ij} - (\lambda^t + \mu^t)r_{ij} - \lambda^t s^t_{ij})^2. \tag{5.4}$$

After derivation in order of λ^t and $\lambda^t + \mu^t$, we get the two formulas that represent the value for a moment t of the intensities for the shell and the core.

$$\lambda^t = \frac{\sum_{i,j} a^t_{ij} s^t_{ij}}{\sum_{i,j} s^t_{ij}}. \tag{5.5}$$

$$\lambda^t + \mu^t = \frac{\sum_{i,j} a^t_{ij} rij}{\sum_{i,j} rij}. \tag{5.6}$$

Changing the intensity values in equation (5.4) with the new formulas represented above, we get

$$\Delta = \sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{j=1}^{J}(a^t_{ij} - \sum_{t}((\lambda^t + \mu^t)^2 \times |R| + (\lambda^t)^2 \times |S^t|), \tag{5.7}$$

where $|R| = \sum_{i,j} r_{ij}$ is the total number of points in the core and $|S^t| = \sum_{i,j} s^t_{ij}$ is the number of points in the shell.

The criterion (5.7) can also be written as

$$\Delta = D - G, \tag{5.8}$$

where D defines the entire data scatter and G represents the contribution of the Core-Shell cluster to that.

The algorithm:

The algorithm builds a sequence of T Core-Shell clusters and corresponding intensity values. An iterative algorithm was proposed in [43], which allows to find a suboptimal solution for the equation (5.7). The algorithm acquires a sequence of T SST instant grids (already pre-processed) as input, corresponding to an upwelling **"time range"**, each already segmented by the S-STSEC algorithm. The results are T clusters that correspond to each cluster to an upwelling region for a time period T. The result of the algorithm is the T Core-Shell clusters and their corresponding intensity values.

The initial Core-Shell clusters are built from the T clusters. The core cluster is achieved by taking an intercept of the T S-STSEC clusters. Each shell is defined as the set difference between the S-STSEC cluster of a given moment and the core is represented by all common areas shared between all the T S-STSEC clusters. The intensity values are then calculated using Equations (5.5) and (5.6).

To join each initial Core-Shell cluster with the remaining upwelling region at a time moment t, the set of grid points that form a 4 neighbourhood, F^t, is also joined with them in a set. The initial Core-Shell clusters are then defined by $B^t = R \cup S^t \cup F^t$. This set B contains all the pixels of interest to be visited by the Core-Shell clustering algorithm, whose goal is to find suboptimal Core-Shell clusters.

The algorithm then iterates as follows: For each point (i, j) in B, three possible scenarios can occur:

- Make point (i, j) part of the core;
- Make the point belong to any of the T shells;
- Remove point from any of them such that criterion G is maximum (from equation (5.8)).

The algorithm stops when there are no more improvements in criterion G.

Having the Core-Shell clusters, essential characteristics and attributes are extracted from it. These features can provide useful information about the properties of the upwelling regions and can be used for further analysis and understanding of the upwelling dynamics. The exact features extracted from the Core-Shell clusters will depend on the specific research questions and goals, but examples of common features include temperature, area, shape, and location.

5.2 Comparing Clustering to Model Time Ranges

Since the Core-Shell clustering receives as input the sequence of SST instants, different initializations with different time ranges lead to different Core-Shell segmentation results.

In this work, we extend the Core-Shell clustering framework with an evaluation module to explore the following aspects:

1. To compare the results of IAP algorithm on finding time ranges, result from the segmentation of time series, with four partitional clustering algorithms, i.e., K-means, K-means++, DBA algorithm and Mean-shift. We developed an experimental protocol that takes advantage of inertia-based and other internal validity indices inspired by [54], to fine-tune the hyperparameters of the clustering algorithms and select the best partitions (Section 5.3). To compare the best partitions obtained by each algorithm, which define the time ranges of an upwelling season, we employ a stability score measure (Section 5.4). Our assumption is that the more stable the derived time ranges are along the upwelling years, the better.

2. To comparatively evaluate the quality of Core-Shell clusters (i.e. core-shell upwelling segmentations) derived from the best time ranges provided by the IAP, K-means, K-means++, DBA and Mean-Shift segmentations. For this, we explore a collection of validity indices (ARI, AMI, KS, WKS and the Mirkin distance).

3. To analyze how interesting are the long-term inter-annual upwelling time series derived from the core segmentation when given to the DBA algorithm to produce average time series. We built datasets with two distinct core segmentation characteristics: core intensity (i.e. average temperature) and core total area. Then, each was treated by DBA algorithm whose clusters and corresponding averaging time series are meaningful respecting the Core-Shell derived periods of stability. This will be presented and discussed in Section 6.6.

The selection of the four partitional clustering algorithms for the comparative analysis of time series segmentation results were based on an exhaustive exploration and experimentation phase involving diverse algorithms within the domain of time series segmentation. Following meticulous testing and evaluation, these four algorithms emerged as the most promising contenders for accurately segmenting time series data.

While discussing other notable algorithms, such as the SWAB algorithm [24] mentioned in Section 4.4, it is crucial to note the challenges encountered during the fine-tuning of user-defined parameters, particularly in the context of the use of multivariate time series data. Furthermore, we explore the k-shaped algorithm [48], which uses cross-correlation to cluster time series. However, its performance proved to be unsatisfactory, which led us to exclude it from our study.

5.3 Protocol to Determine the Number of Clusters based on Internal Validity Indices

Determining the optimal number of clusters in a dataset is a crucial step in cluster analysis. Internal validity indices help assess the quality of clustering solutions without relying on external information. Several popular internal validity indices include the Silhouette Score, the Davies-Bouldin index, the Calinski-Harabasz index, and the Elbow Method [54]. In this section, we will present some internal validity indices that were used in the selection of the optimal number of clusters for the results of K-means, K-means++ and DBA in the clustering of time ranges.

5.3.1 Inertia-based indices

Inertia, also known as the sum of squares within a cluster, is a metric used to evaluate the quality of a clustering algorithm, particularly in the context of K-partitioning algorithms. Inertia measures the compactness of the produced clusters. For each point in the dataset, the squared Euclidean distance is calculated between that point and the centroid of the cluster to which it belongs. All these values are summed, and we obtain the inertia value.

Overall, a low inertia value is better, since it indicates that the points in a cluster are close to the centroid, meaning that they are well defined. However, inertia is also sensitive to the number of clusters, since inertia tends to decrease with increasing number of clusters. The main goal in inertia validity indices is to find a balance between a low inertia value and the number of clusters.

A common inertia-based index is the elbow method. The usual way this method is applied with K partitioning algorithms will be described as follows:

- **Compute K-partitioning clustering** - The K-partitioning algorithm is computed for a dataset, usually for a range of multiple K values.
- **Calculate inertia** - For each value of K, the inertia value is calculated.
- **Plot the inertia of different K** - A plot is created where the x-axis represents the number of clusters (K), and the y-axis represents the corresponding inertia values. An example of a plot can be observed in Figure 5.4.
- **Identify the "Elbow" point** - In the plot, the inertial values typically decrease with increasing K. This reduction is not usually linear, so often an "elbow" shape is observed, where the inertial values start to decrease less. In Figure 5.4, we can notice that this "elbow" shape is formed in K equal to three.
- **Choose the optimal K** - This "elbow" in the plot often indicates the optimal K to select. Here, the clusters cease to provide a significant reduction, and further partition does not improve cluster quality.

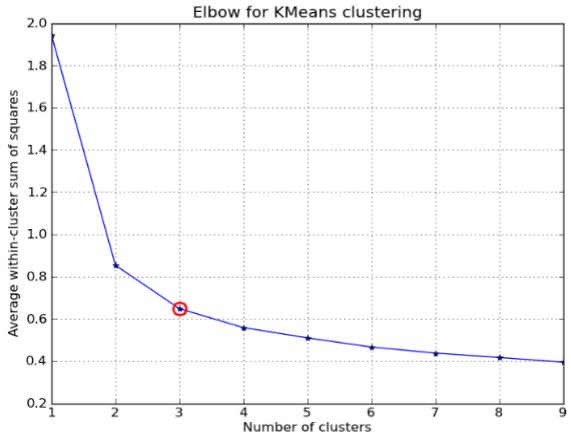


Figure 5.4: Example of a plot of the inertia for different K value. Image taken from [36]

It is important to note that the optimal K depends on a specific dataset and problem. This method provides a visual heuristic, but in some cases it may not be clear, so other validation techniques may be necessary.

In the work carried out by Rykov et al. [54], a new concept of the elbow method was analyzed. This concept admits various interpretations depending on the number of clusters that are used before and after a number of clusters K. Using different ratios of the difference between the value of inertia for K, it is possible to compute not only the elbow method with the difference of inertia for one step but also for more. The typical elbow method uses the difference in inertia in one step. We will call this approach Elbow Point 1 (EL1), and it can be computed as follows:

$$ElbowPoint1(K) = \frac{D(K-1) - D(K)}{D(K) - D(K+1)}, \tag{5.9}$$

where K is the number of clusters and D is the value of inertia, so D(K-1) is the value of inertia for K-1 and D(K+1) is the value of inertia for K+1. It is also possible to compute another function for a two-step difference. We will call it Elbow Point 2 (EL2) and can be defined as:

$$ElbowPoint2(K) = \frac{D(K-2) - D(K)}{D(K) - D(K+2)}. \tag{5.10}$$

The study that explored this concept observed that for most cases, both with synthetic and real-world datasets, EL1 performed better than classic EL2. We will employ these two metrics in clustering time ranges using them to determine the optimal K value in the K-partitioning algorithms.

Our procedure for fine-tuning the hyperparameter K is inpired in the work developed in [54], where we will use the previous concept of the elbow method and the following indices:

- **Calinski-Harabasz index (CH)** is a measure of the effectiveness of a clustering algorithm. It is defined as the ratio between the between-cluster sum of squares

(SSB) and SSW. The index increases as the ratio of between-cluster sum of squares to within-cluster sum of squares increases.

$$CH(K) = \frac{SSB(K)/(K-1)}{SSW(K)/(N-K)}, \tag{5.11}$$

where N is the number of entities and K is the number of clusters.

- **Xu index (XU)** is obtained from a logarithmic equation for divergence within a variation of a Gaussian mixture model. The formula for XU is as follows:

$$XU(K) = V \log\left(\sqrt{\frac{D(K)}{VN^2}}\right) + \log K, \tag{5.12}$$

where V is the dimensionality of the data, N is the number of entities and K is the number of clusters. It should reach its minimum value at the optimal number of clusters K.

- **Wu and Bailey index (WB)** is a validation index for cluster analysis that balances between SSB and SSW. It can be regarded as a scaled inverse of the CH index and is defined as follows:

$$WB(K) = K\left(\frac{SSW(K)}{SSB(K)}\right). \tag{5.13}$$

5.3.2 Silhouette width index

Silhouette width index (SW) [53] is a widely used cluster validation index that does not rely on the inertia formula. Measures the similarity between an object and its respective cluster in contrast to other clusters. SW ranges from -1 to 1, where a value close to 1 indicates that the object is appropriately matched to its own cluster but is poorly matched with adjacent clusters (meaning that it is a good fit to the cluster), a value of 0 indicates that the object is on or very close to the decision boundary between two neighboring clusters, and a value close to -1 indicates that the object is better matched to a neighboring cluster than to its own cluster.

The SW for a data point can be defined as [54]:

$$S(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}, \tag{5.14}$$

where $a(i)$ is the average distance from the data point i to the other data points in the same cluster (intra-cluster distance), and $b(i)$ is the minimum average distance from the data point i to the data points in a different cluster (inter-cluster distance).

To compute the overall SW for the entire dataset, typically the average Silhouette width is calculated over all data points. A higher average Silhouette width indicates better cluster separation, and it can be used to help determine the optimal number of clusters for a given dataset.

This value is calculated for multiple K values, and the highest Silhouette value obtained indicates the right number of clusters to be used. We will use this index in our study as a comparison to inertia-based indices.

5.4 Stability Score Measure

Cluster stability allows for explicit quantification of the quality of a clustering solution, without being dependent on external information. In Roth et al. [52], a measure of the stability of a clustering algorithm was presented, based on the assumption that when various partitions produced by a clustering algorithm are similar to each other, it indicates the effectiveness of the setting.

The unsupervised cluster stability value, $S(c)$, that is used in [52] is calculated as the average pairwise similarity between the M partitions and is defined as follows:

$$S(c) = \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \frac{d(U_i, U_j)}{\frac{M(M-1)}{2}}, \tag{5.15}$$

where U_i and U_j, $1 \leq i < j \leq M$, are two partitioning produced for c clusters and $d(U_i, U_j)$ is an arbitrary similarity index of partitioning [25].

In our work, this measure will be used to measure the stability of the time ranges obtained by the different clustering partitions to access the metrics and set better model time ranges. Essentially, the stability measure calculates the average similarity between time ranges across different years, providing insight into how consistently a particular clustering algorithm can identify similar time ranges over the years.

The main reason for this approach is the fact that we have no ground truth for the validation of the unsupervised discovery of these time ranges, so this stability score function will be employed to evaluate their quality.

We will evaluate all cluster partitioning settings, where M will be the total number of years of upwelling and U_i, U_j are two distinct partitions of SST instants, each partition corresponding to the collection of time ranges of two upwelling seasons, i, j, respectively. As our similarity index for partitioning, we will use ARI and AMI. Both similarity indexes will be described in the next Section.

5.5 Evaluating the performance of Core-Shell clustering

As stated previously, different initializations of the Core-Shell clustering lead to distinct Core-Shell clusters. In the implementation of the Core-Shell framework [32], the validation process involved comparing the Core-Shell segmentations with the SST instants within their respective time range. Two similarity measures, WKS and ARI, were used for this comparative analysis. It is important to note that both similarity measures require S-STSEC segmentations as input, which are binary maps used as ground truth.

However, relying solely on these indices is insufficient to validate the precision of the clustering results. This limitation arises from its limited perspective, vulnerability to specific data characteristics, subjectivity, and the possible failure to cover all aspects of cluster quality. Therefore, the use of a variety of indices is essential for a comprehensive and robust evaluation of cluster quality [20].

We propose to extend the evaluation of the Core-Shell clustering, by creating a more robust validity procedure by using multiple state-of-the-art cluster similarity indices. The validity indices proposed to be used in this evaluation process are described in the following sections.

5.5.1 Adjusted Rand index

The Adjusted Rand index (ARI) is a measure of similarity between two clustering results, with values ranging from 0 to 1, with 1 corresponding to perfect agreement between clusterings and 0 corresponding to total disagreement between clusterings. The Rand index (RI), which counts the number of pairs of data points that are assigned to the same or distinct clusters in both the true and predicted clusterings, is converted into the ARI using corrected data [33].

When comparing the observed similarity to the predicted similarity in a random clustering model, the ARI corrects the RI by taking into account the potential of a similar result occurring by chance. In many clustering settings, the number of clusters or the size distribution of those clusters could vary significantly, giving rise to various random clustering models. This requires a correction for chance. As a result, the ARI provides a more accurate measure of the similarity between two clusters in different scenarios, compared to the RI [20].

Considering two clusters, C and C', there exist four potential categories for pairs of objects: N_{11} - where both objects are part of the same cluster in both sets of clusters, N_{10} - where they are in the same cluster in C but in different clusters in C', N_{01} - where they are in different clusters in C but in the same cluster in C', and N_{00} - where they are in different clusters in both clusterings. With this notation, the Rand index is defined as [33]:

$$RI(C, C') = \frac{N_{11} + N_{00}}{(n/n-1)/2}, \tag{5.16}$$

where $(n/n-1)/2$ is the total number of pairs and can also be represented as $N_{11} + N_{10} + N_{01} + N_{00}$. For the description of ARI, the total number of pairs will be represented as N. ARI is defined as:

$$ARI(C, C') = \frac{(RI(C, C') - E[RI])}{1 - E[RI]}, \tag{5.17}$$

where $E[RI]$ (expected RI) is a value obtained when partitions are made at random, but cluster distributions are kept. A more concise formula can be read as follows:

$$ARI(C, C') = \frac{N(N_{11} + N_{00}) - ((N_{11} + N_{10})(N_{11} + N_{01}) + (N_{01} + N_{00})(N_{10} + N_{00}))}{N^2 - ((N_{11} + N_{10})(N_{11} + N_{01}) + (N_{01} + N_{00})(N_{10} + N_{00}))}. \quad (5.18)$$

5.5.2 Kulczynski Similarity

The Kulczynski Similarity (KS) is a measure used to assess the similarity between two clustering partitions or assignments of the same dataset. It quantifies how similar the two clustering results are by considering the extent to which the data points are grouped into the same clusters in both partitions. KS is often used when two clustering solutions are expected to be compared without assuming any ground truth or reference clustering [66].

In order to provide a suitable assessment and a relevant quantitative analysis of segmentation algorithms, a new metric based on the KS index called the Weighted Kulczynski Similarity (WKS) index was presented by Zakani et al. [66]. This index has an advantage compared to the KS index, which allows the comparison of a single cluster with a set of associated ground-truth clusters.

Let $\{S_1, \ldots, S_n\}$ be the set of ground truths where each $S_i = \{R^1_{S_i}, \ldots, R^k_{S_i}\}$ is a ground truth segmentation and $A = \{R^1_A, \ldots, R^m_A\}$ the result partition from a given segmentation algorithm. First, for each R^i_A its optimal correspondent segments $R^{i_t}_{S_j}$ are searched for every available ground truth as follows [66]:

$$i_t = max_j \left(max_k ||R^i_A \cap R^k_{S_j}|| \right). \quad (5.19)$$

Subsequently, the KS index is computed for each pair R^i_A and $R^{i_t}_{S_j}$ by applying the following formula:

$$KS(R^i_A, R^{i_t}_{S_j}) = \frac{1}{2} \left(\frac{||R^i_A \cap R^{i_t}_{S_j}||}{||R^i_A \cap R^{i_t}_{S_j}|| + ||R^i_A \setminus R^{i_t}_{S_j}||} + \frac{||R^i_A \cap R^{i_t}_{S_j}||}{||R^i_A \cap R^{i_t}_{S_j}|| + ||R^{i_t}_{S_j} \setminus R^i_A||} \right), \quad (5.20)$$

with $||x||$ representing the cardinality of set x and $\setminus$ denoting the operation of set difference. The WKS is subsequently determined as the mean of all the KS indices calculated:

$$WKS(A, \{S_n\}) = \frac{\sum_{i=1}^{m} KS(R_A, R_{S_i})}{m}. \quad (5.21)$$

5.5.3 Normalised and Adjusted Mutual Information

Normalised Mutual Information (NMI) is a measure used to assess the quality of the clustering results by comparing the similarity between two different cluster assignments or partitions of the same dataset. NMI is particularly useful when you want to evaluate how well a clustering algorithm has grouped data points into clusters compared to a ground truth or reference clustering. The NMI scales from 0 to 1, with 1 denoting that the two partitions are identical and 0 denoting that they are independent of one another.

The NMI is normalized, which means that the result is rescaled to take into account the variation in cluster variations between the two partitions. As a result, even when two methods produce partitions with a differing number of clusters, they may be compared using the NMI [20].

To better understand NMI, Mutual Information (MI) will be described. MI measures the amount of information shared between two random variables, which, in the context of clustering, are the ground truth labels and the cluster assignments. MI is defined as [33]:

$$MI(C, C') = \sum_{k=1}^{K} \sum_{k'=1}^{K'} P(k, k') log \frac{P(k, k')}{P(k)P'(k')}, \tag{5.22}$$

where $P(k, k')$ is the joint probability distribution of the clusters C and C', $P(k)$ is the marginal probability distribution of the cluster C and $P(k')$ is the marginal probability distribution of cluster C'. The higher the MI value, the more information the two labels share, indicating better clustering results.

Although MI provides a useful measure of similarity between two labels, it does not account for the scale of the MI values, making it difficult to interpret in isolation. NMI is introduced to address this problem by normalizing the MI value to ensure that it falls within a specific range, making it easier to compare between different datasets and clustering results. NMI can be defined as:

$$NMI(C, C') = \frac{2(MI(C, C'))}{H(C) + H(Y)}, \tag{5.23}$$

where $MI(C, C')$ is the Mutual Information between clusters C and C', $H(C)$ is the entropy of cluster C and $H(C')$ is the entropy of cluster C'. Entropy measures the uncertainty or disorder of a random variable. For discrete random variables, it is defined as:

$$H(X) = -\sum P(x) * log(P(x)), \tag{5.24}$$

where P(x) is the probability of a specific label or cluster assignment. The AMI is adjusted for chance and works well even if the clusters have different sizes. The AMI function is defined as:

$$AMI(C, C') = \frac{MI(C, C') - E[MI(C, C')]}{\max(H(C), H(C')) - E[MI(C, C')]}, \tag{5.25}$$

where $MI(C, C')$ is the mutual information, $E[MI(C, C')]$ is the expected mutual information under the null distribution, and $H(C)$ and $H(C')$ are the entropies.

5.5.4 Mirkin distance

The Mirkin distance, also known as the Mirkin index or the Mirkin metric, is a measure used to assess the dissimilarity between two clustering partitions or assignments of the same dataset. Quantifies how different the two clustering results are by considering the differences in the way data points are grouped into clusters. The Mirkin distance is

particularly useful when you want to compare two clustering solutions without assuming any ground truth or reference clustering [33].

To calculate the Mirkin distance between two clustering partitions, the four possible types of object pairs (N_{11},N_{10},N_{01} and N_{00}) that were explained in ARI will also be used in the current explanation. The Mirkin distance formula can be described as follows:

$$M(C, C') = \frac{N_{01} + N_{00}}{N_{11} + N_{10} + N_{01} + N_{00}}. \tag{5.26}$$

The Mirkin distance ranges from 0 to 1, with higher values indicating greater dissimilarity between the two clustering partitions. A Mirkin score of 0 indicates that the two partitions are identical, while a score of 1 suggests that the two partitions are entirely dissimilar.

It is important to note that the Mirkin distance does not take into account the quality or correctness of the clustering solutions, only their dissimilarity. Therefore, it is often used in conjunction with other clustering evaluation metrics to provide a comprehensive assessment of clustering performance.

6

Experimental Study

In this chapter, we present and discuss the experimental results. The chapter is organized as follows: Section 6.1 presents the objectives of the current study and Section 6.2 explains the imagery data used. The discussion of the experimental results begins in Section 6.3, where a feature extraction analysis is carried out for the S-STSEC segmentations. In Section 6.4, we compare the results of IAP, K-means, K-means++, DBA, and Mean-Shift algorithms in the findings of the upwelling time ranges in the three geographic regions of this study (Portugal, northern Morocco, and southern Morocco). Fine-tuning of the algorithms hyperparameters with the validity experimental protocol was described in Section 5.3. For each algorithm, we select the partitions corresponding to time ranges with higher stability scores and use them as input for the Core-Shell clustering algorithm, evaluating the results obtained in Section 6.5. The time series of features extracted from the upwelling cores are analyzed using the DBA algorithm in Section 6.6. Finally, Section 6.7 provides a summary of the results.

6.1 Goals of the Study

The main goal of this study was the development of an experiment pipeline to compare partitional clustering algorithms (K-mean, K-means++, DBA, and Mean-Shift) with IAP in segmenting time series to find periods of coastal upwelling stability, the so-called upwelling time ranges [32], evaluating the effectiveness of the results using a stability measure, as well as their influence on the quality of the Core-Shell clustering segmentations.

The following stages were developed:

1. Upwelling regions features selection through correlation analysis;
2. Implementation of the experimental procedure to fine-tune the hyperparameter of partitional clustering algorithms:
    a) Protocol to determine the number of clusters, K, in K-means, K-means++ and DBA algorithms, via inertia-based indices and SW;
    b) Bandwidth fine-tuning and analysis of the Mean-Shift kernels.

3. Implementation of the stability score measure and evaluation of the stability values of each algorithm;

4. Run the Core-Shell algorithm initialized with the best upwelling time ranges, selected according to the best stability scores and validation of Core-Shell segmentations through several state-of-the-art validity indices;

5. Analyze how interesting are the long-term inter-annual upwelling time series derived from the cores' segmentation when given to the DBA algorithm and the produced average time series.

6.2 Imagery Data

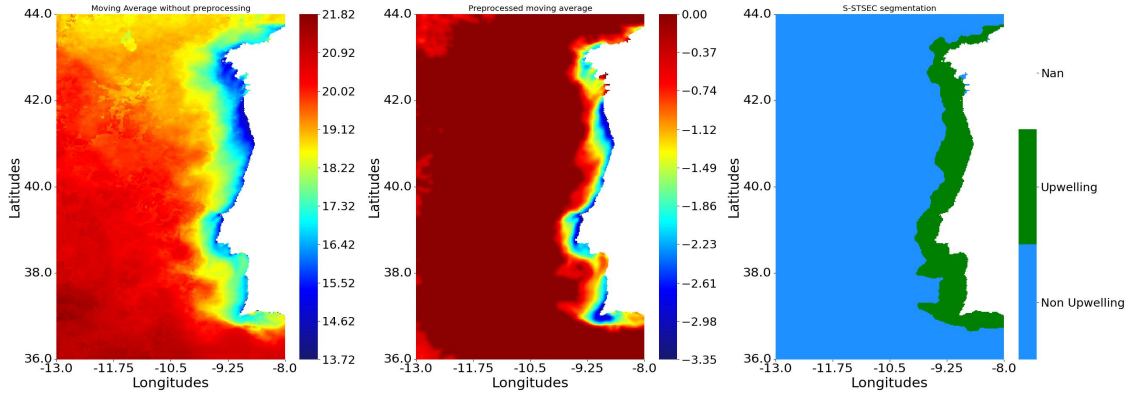
The data collections used in this dissertation were taken from the work in [32, 42, 44]. They consist of three annual collections of SST images, each for 16 years, from the coasts of Portugal, northern Morocco, and southern Morocco, which were used in this study, covering the years 2004 to 2019. SST images from the Portuguese coast range from a latitude of 36°N to a latitude of 44°N and a longitude from 8°W to 13°W. Images of the northern Moroccan coast range from a latitude of 30°N to a latitude of 35°N, and a longitude of 16°W to 5.5°W. Images from the southern Moroccan coast range from a latitude of 20°N to a latitude of 27°N and a longitude of 21.5°W to 13°W. The 27°N and 30°N band was left out due to the presence of the Canary Islands, which are known to present irregularities in the upwelling routine [7].

For each region, SST grids were built. The Portuguese coast's SST grid is 401 pixels by 251 pixels, the northern Moroccan coast's SST grid is 251 pixels by 501 pixels, and the southern Moroccan coast's SST grid is 351 pixels by 426 pixels. Each pixel belonging to the grids is represented by a floating value of temperature in degrees Celsius and has a spatial resolution of 2km by 2km.

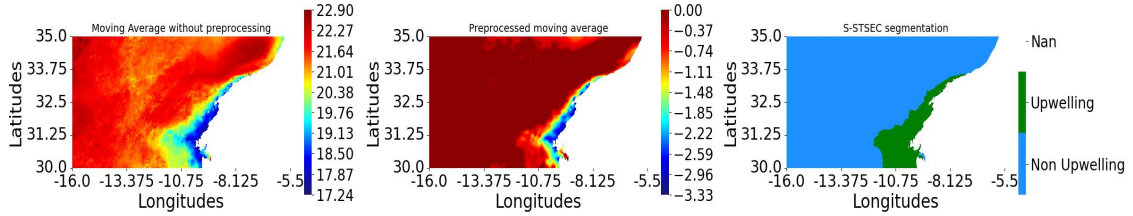
In each collection, certain pixels may contain errors, such as meteorological events-related errors or non-sea surface regions, and these are replaced with a NaN value. Figure 6.1 shows, for each geographic region, an SST image before preprocessing, an SST image after the preprocessing stage, and the corresponding S-STSEC segmentation for the week of June 18 to June 24 of 2012. In the S-STSEC segmentations, we can observe in white the NaN values that represent the land region.

We considered only SST data from approximately March 30 to October 30 for each of the three regions, having 27 SST grids per season.

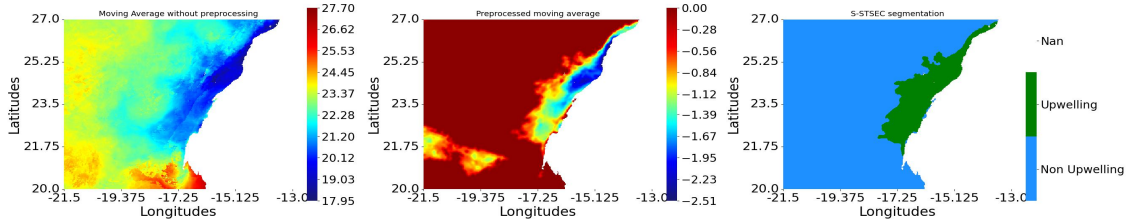
We deal with 23 SST instant grids per upwelling season after the preprocessing stage that was explained in Section 5.1.1.



(a) Portugal



(b) North Morocco



(c) South Morocco

Figure 6.1: SST instant before preprocessing, preprocessed with moving average and the correspondent S-STSEC segmentation of week of 18 June to 24 June for each geographic region being analyzed

6.3 Feature Extraction from S-STSEC segmentations

In [32, 42, 44] the authors extracted four features from the S-STSEC segmentation time series:

- The total area of the upwelling region;
- The average temperature over the upwelling pattern;
- The maximum latitude of the upwelling region (northernmost latitude);
- The minimum latitude of the upwelling region (southernmost latitude).

Based on the domain knowledge, in [32, 42, 44], the authors selected the area of the upwelling regions and the mean temperature of those regions to define the upwelling regions. The maximum and minimum latitudes of the upwelling region were chosen based

on empirical evidence to describe the morphological differences between the upwelling segmented SST instants.

In our current work, we are working with three geographic regions, each with specific upwelling characteristics, so we decided to perform a feature extraction analysis for each geographic region to extract the set of features that better correlate with the total area and average temperature of the upwelled region.

For each year, we computed the correlation between total area and average temperature with the northern and southern latitudes and the western longitudes of the upwelling region. From the three aforementioned features, only the features having a positive correlation (equal to or greater than 0.5) or a negative correlation (equal to or less than -0.5) with both the total area and the average temperature were chosen for the clustering of time series to obtain time ranges.

	Portugal		North Morocco		South Morocco	
Features	**Total Area**	**Average Temperature**	**Total Area**	**Average Temperature**	**Total Area**	**Average Temperature**
Northernmost Latitude	0.74	0.56	0.75	0.67	0.26	0.003
Southernmost Latitude	-0.39	-0.46	N/A	N/A	-0.18	-0.61
WesternMost Longitude	-0.04	0.12	-0.67	-0.45	-0.17	0.60

Table 6.1: Feature average correlation

The results presented in Table 6.1 are summarized as follows:

- **Portugal -** The northernmost latitude has a strong positive correlation (correlation greater than 0.7) with the total area with a value of 0.74 and a positive correlation with the average temperature with a value of 0.56. The southernmost latitude and westernmost longitude do not have a positive or negative correlation with any of the domain knowledge features. In Figure 6.2, two SST instants from the year 2007 are shown. The area highlighted in green represents the upwelling region. These two Figures represent what was observed for most of the years, where the northern latitude varied through the upwelling season, but the other two features remained mostly constant for the most part of the years.

(a) SST instant 10 (b) SST instant 17

Figure 6.2: S-STSEC segmentations of the region of Portugal for the year 2007

- **North Morocco -** The value of the southern latitude feature was constant throughout all the upwelling seasons that were analyzed. Since the value did not change, no correlation with this feature was observed. The northernmost latitude has a strong positive correlation with the feature of the total area of the upwelling region with a value of 0.75 and a positive correlation with the average temperature with a value of 0.67. The westernmost longitude has a negative correlation with the total area with a value of -0.67 and a correlation value of -0.45 with the average temperature. Similarly to the upwelling in the region of Portugal, the northernmost latitude value is the only features that correlates with the total area and the average temperature in North Morocco, as shown in Figure 6.3.

(a) SST instant 10 (b) SST instant 17

Figure 6.3: S-STSEC segmentations of the region of North Morocco for the year 2007

- **South Morocco -** The northernmost latitude does not correlate with any of the domain knowledge features. The southernmost latitude does not correlate with the total area and has a negative correlation with the average temperature with a value of -0.61. The westernmost longitude does not correlate with the total area and has a positive correlation with the average temperature with a value of 0.60. In Figure 6.4, a contrast is observed compared to previous geographic regions, as there is no visible correlation between the three geographic features and the total area and the average temperature.

(a) SST instant 10 (b) SST instant 17

Figure 6.4: S-STSEC segmentations of the region of South Morocco for the year 2007

The result of this analysis of the features extracted from the upwelling segmentations per geographic region is presented in Table 6.2. These will be the extracted features for the next stage, the modeling of upwelling time ranges.

	Total Area	Average Temperature	Northernmost Latitude
Portugal	✓	✓	✓
North Morocco	✓	✓	✓
South Morocco	✓	✓	

Table 6.2: Selected features for each geographic region

6.4 Finding Upwelling Time Ranges

Following the details provided in Section 5.2, the algorithms IAP, K-means, K-means++, DBA, and Mean-Shift were used to find upwelling time ranges.

For each algorithm, we constructed for each geographic region an entity-to-feature data matrix whose rows are the SST instants (23) that make up an upwelling season and the columns are the set of features shown in Table 6.2. We construct the matrix described for each year, where the entry i, j is the value of the feature j for instant i. Each hyperparameter fine-tuned in this study is also discussed in each subsection.

In the application of the algorithms, we obtained different clusters of different sizes. In the work developed by Nascimento et al. [43], it was determined that a time range must consist of a minimum of three consecutive SST instants, and we will include this aspect in our study, since it is important to obtain meaningful time ranges that represent periods of stability in the upwelling events.

For time ranges composed of less than three consecutive SST instants, an automatic correction is implemented, reassigning these segments to one of the neighboring clusters. The specific process depends on the algorithm. Specifically:

- **IAP:** Reassignment is directed to the consecutive cluster with the lowest contribution to the data scatter.

- **K-means, K-means++, DBA and Mean-Shift:** Reassignment uses the corresponding distance measure (Euclidean for K-means, K-means++ and Mean-Shift, and DTW for DBA) to consecutive cluster centroids, selecting the cluster with the nearest centroid.

A time range that starts at the SST instant i and ends at the SST instant j will be represented as [i-j]. In the following sections, we will designate time ranges according to this nomenclature.

The subsequent subsections present an analysis and discussion of the time ranges obtained by each algorithm. The final subsection presents an analysis and discussion of all the best-performing algorithm results to identify the most suitable time ranges for the subsequent stage, the evaluation of the Core-Shell clustering results. The evaluation of the time ranges will be performed using the stability score measure presented in Section 5.4.

6.4.1 IAP Time Ranges

IAP was configured to segment time series until all available data was clustered, following the predefined stop condition, where the algorithm stops when every SST instant was clustered.

With this configuration, IAP obtained three to five groups of consecutive SST instants for each upwelling season, as shown in Table 6.3 for the Portugal region.

We generated graphs showing the cluster contribution and cluster cardinality for each year analyzed in the study to examine the patterns identified in the time ranges produced by the IAP algorithm. Specifically, the cluster contribution of the IAP cluster to the data scatter (defined in Equation 5.2), which determines the effectiveness of the IAP algorithm by grouping data points into distinct clusters.

In the following subsections, we will examine the results by geographic region, covering Portugal, North Morocco, and South Morocco.

6.4.1.1 Portugal

In Figure 6.5, the cluster contributions for the Portugal region are displayed. The peak cluster contribution was obtained in the initial time ranges of 2005, reaching 60.9%. On the contrary, the minimum contribution in the first time range was observed in 2014, with a value of 26.9%. The intermediary clusters, representing the middle time ranges of the upwelling season, registered the smallest contributions of the clusters, ranging from 17.7% to 0.1%. A pattern emerges in the final time ranges, marked by an upward increase in the cluster contribution values in contrast to the intermediate cluster values of each year.

In Figure 6.6, the cardinality of the clusters is shown for the region of Portugal. There is a clear pattern showing that the final time ranges always contain the highest number of SST instants in a cluster, ranging from 9 to 13. In contrast, intermediate clusters consistently have the fewest SST instants in a cluster, varying between 3 and 5. This suggests that the beginning and end of an upwelling season consist of several weeks exhibiting similar upwelling characteristics.

The results highlight a consistent pattern in the IAP algorithm, revealing two clusters with high contribution values: the initial and final upwelling time ranges. These time ranges distinctly represent the start and end of the upwelling season, respectively. However, the intermediate ranges depict the changes in the intensity of the upwelling activity, characterized by clusters with smaller contributions and cardinalities.

Year	IAP
2004	[1-6], [7-10], [11-23]
2005	[1-5], [6-11], [12-23]
2006	[1-6], [7-9], [10-12], [13-23]
2007	[1-9], [10-13], [14-23]
2008	[1-5], [6-10], [11-23]
2009	[1-7], [8-10], [11-13], [14-23]
2010	[1-4], [5-7], [8-12], [13-23]
2011	[1-5], [6-8], [9-12], [13-23]
2012	[1-7], [8-11], [12-23]
2013	[1-3], [4-8], [9-11], [12-13], [15-23]
2014	[1-6], [7-12], [13-23]
2015	[1-3], [4-6], [7-12], [13-23]
2016	[1-7], [8-12], [13-23]
2017	[1-5], [6-10], [11-23]
2018	[1-6], [7-9], [10-13], [14-23]
2019	[1-5], [6-8], [9-13], [14-23]

Table 6.3: IAP Time Ranges Results for the region of Portugal

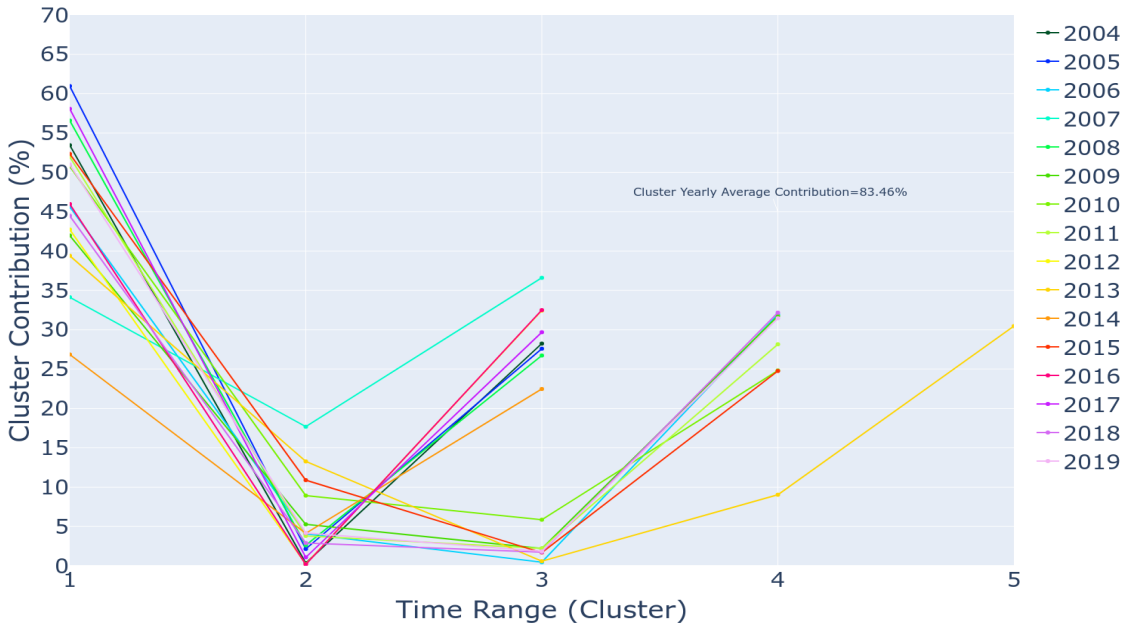


Figure 6.5: IAP Cluster Contribution of the clusters in the region of Portugal

Figure 6.6: IAP Cluster Cardinality of the clusters in the region of Portugal

6.4.1.2 North Morocco

In Figure B.1, the cluster contributions for the northern region of Morocco are shown. In particular, the final time range for 2010 stands out with the highest contribution value of 51.6%. The initial and final time ranges show significant contributions to the cluster, sometimes exceeding 50%. However, there are two exceptions that do not follow this pattern. In 2005, the second time range had a contribution of 26.9%, higher than the first respective time range, resulting in an overall contribution of 21.7%. Similarly, in 2013, a distinctive pattern was observed, with cluster contribution values decreasing throughout the upwelling season (32.7%, 28.5%, and 20.4%, respectively). Nonetheless, a clear trend emerges where the first period typically obtains contributions between 50.3% and 35.8%, while the final clusters consistently obtain contributions ranging from 29.4% to 51.6%.

In Figure B.2, the cardinality of the time ranges is shown for the northern Moroccan region. Although not consistent across all years, a trend is observed. Similarly to the Portugal region, the final ranges consistently display the highest cluster cardinality, ranging from 7 to 13 SST instants. In contrast, the initial ranges obtain cluster cardinalities ranging from 4 to 10, indicating a stability of at least 4 weeks of the upwelling season in all the years covered. There are also outliers, such as the year of 2009, where the second time range obtains the highest cardinality, deviating from the typical pattern.

Although the results for northern Morocco exhibit less uniformity compared to those of the Portugal region, patterns are observed in the time ranges of the IAP algorithm in both geographic regions.

6.4.1.3 South Morocco

Looking at the cluster contributions in Figure B.3, a distinct pattern emerges in the yearly results for this region. Both the initial and final clusters share a cluster contribution interval, with the highest value at 43.8% (final range for the year 2006) and the lowest at 31.8% (final range for the year 2009). Despite generally lower average contribution values compared to the other two regions, southern Morocco shows great consistency in cluster contributions, consistently featuring two high-contribution clusters each year.

In Figure B.4, the cardinality of the clusters is presented for the southern Moroccan region. The highest cardinality cluster, observed in 2007, reached a total of 11 SST instants. The initial ranges obtained cardinality values ranging from 5 to 9, while the final ranges recorded values between 7 and 11 SST instants. A trend in cluster cardinality is observed in the final ranges, consistently having the highest cardinality values within their respective upwelling seasons.

When examining the three regions, it is evident that the IAP consistently captures the stability patterns of different upwelling seasons.

6.4.2 K-means Time Ranges

We used the *TimeSeriesKMeans* model from the tslearn package[1] for the time series segmentation with K-means, K-means++ and DBA.

We initialize K-means, from the initial centers determined using the traditional approach (random centroid initialization), as described in Section 4.3.1, 10 times each for values of K ranging from 2 to 7, and we select the optimal K partition applying the validation procedure described in Section 5.3.

To evaluate the time ranges obtained from the application of K-means with the various validity indices used in the process mentioned above, we created tables with the stability scores obtained. The stability score measure was presented in Section 5.4.

The selection of the best partitions to be selected for the next stage will take into account the ARI values, with the AMI values being employed to confirm the results observed with the aforementioned index and for comparison purposes.

The stability score results of the K-means are presented in Table 6.4 and the analysis of the different regions is presented below:

- **Portugal -** The highest ARI stability score (0.6) is achieved by K means with SW and EL2. The lowest ARI stability score (0.5) is obtained by the K-means with EL1.

- **North Morocco -** The highest ARI stability score (0.58) is achieved by the time ranges of K-means with EL2, while the lowest score (0.51) was obtained by the results of K-means with CH.

[1] https://github.com/tslearn-team/tslearn/blob/9937946/tslearn/clustering/kmeans.py

- **South Morocco -** Similarly to Portugal, K-means with SW and EL2 obtained the highest ARI stability scores with a value of 0.58. The lowest ARI stability scores were observed for K-means with WB and XU, both registering a value of 0.51.

	Portugal (PT)		**North Morocco (NM)**		**South Morocco (SM)**	
K-means	SS ARI	SS AMI	SS ARI	SS AMI	SS ARI	SS AMI
Silhouette index (SW)	**0.60**	0.63	0.52	0.60	**0.58**	0.65
Elbow Point 1 (EL1)	0.50	0.60	0.54	0.63	0.53	0.62
Elbow Point 2 (EL2)	**0.60**	0.65	**0.58**	0.66	**0.58**	0.66
Calinski–Harabasz index (CH)	0.51	0.62	0.51	0.61	0.53	0.63
Within-Between (WB)	0.51	0.62	0.55	0.64	0.51	0.62
Xu (XU)	0.51	0.62	0.55	0.64	0.51	0.62

Table 6.4: Stability scores by ARI / AMI of the Time Ranges obtained by the K-means for the geographic regions of Portugal, North Morocco and South Morocco

K-means SW and the K-means EL2 provide the most stable time ranges in Portugal and South Morocco. In North Morocco, K-means EL2 are the best K-means configuration to obtain the most stable time ranges.

In the section dedicated to analyzing the best performing time ranges of each algorithm, we will prioritize the time ranges obtained from K-means (or any other K-means algorithm used in the study, such as K-means++ and DBA) with the Silhouette Width index (SW) over those with Elbow Point 2 (EL2), in cases where they obtain identical results. In this case, for example, for the Portugal and South Morocco region, we will use the time ranges obtained from SW, while for North Morocco, the time ranges from EL2 will be used. This preference for the values of SW over EL2 is driven by the preference for the Silhouette Width score, a well-established internal validity index in the literature to assess the quality of the clustering results, in contrast to the novel Elbow Point 2 approach.

6.4.3 K-means++ Time Ranges

We run K-means from the initial centers using the k-means++ approach, as described in Section 4.3.2, using the same methodology described for traditional K-means to obtain the optimal K.

Table 6.5 presents the stability scores for the K-means++ results which incorporate the same indices as used previously. The subsequent analysis for the regions in our study is presented below:

- **Portugal -** The highest ARI stability score (0.59) is achieved by K-means++ with SW and EL2. The lowest ARI stability scores (0.51) are observed for K-means++ with CH, WB, and XU.

- **North Morocco -** The highest ARI stability score (0.58) is achieved by K-means++ with EL2, while the lowest score (0.5) is obtained by K-means++ with CH.

- **South Morocco -** Similarly to Portugal, K-means++ with SW and EL2 achieved the highest ARI stability scores of 0.58. The lowest ARI stability scores were observed for K-means with CH, WB, and XU, all obtaining a value of 0.53.

	Portugal (PT)		**North Morocco (NM)**		**South Morocco (SM)**	
K-means++	SS ARI	SS AMI	SS ARI	SS AMI	SS ARI	SS AMI
Silhouette index (SW)	**0.59**	0.63	0.52	0.60	**0.58**	0.65
Elbow Point 1 (EL1)	0.56	0.62	0.53	0.61	0.55	0.63
Elbow Point 2 (EL2)	**0.59**	0.65	**0.58**	0.65	**0.58**	0.65
Calinski–Harabasz index (CH)	0.51	0.62	0.50	0.62	0.53	0.63
Within-Between (WB)	0.51	0.62	0.53	0.63	0.53	0.63
Xu (XU)	0.51	0.62	0.53	0.63	0.53	0.63

Table 6.5: Stability scores by ARI / AMI of the Time Ranges obtained by the K-means++ for the geographic regions of Portugal, North Morocco and South Morocco

The stability score value analysis indicates that K-means++ SW and K-means++ EP2 correspond to the best K-means++ configurations to achieve the highest stability score clusters in Portugal and South Morocco. In North Morocco, K-means++ EP2 emerges as the preferred K-means++ configuration to obtain stable time ranges.

6.4.4 DBA Time Ranges

Initially, the algorithm selects K time series from the dataset as barycentres. This selection was done on the basis of the k-means++ heuristics. Then DBA is used as described in Section 4.3.3, using the same methodology as described in previous approaches to determine the best value of the parameter K, but instead of using the Euclidean distance as its distance measure, the DBA algorithm uses DTW.

In the context of the DBA algorithm, inertia is not calculated directly as in K-means clustering. Instead, DBA aims to find a representative or "barycenter" time series that minimizes the sum of distances between this barycenter and a set of input time series. Inertia is defined as the minimization of the sum of distances between the barycenter and the aligned input time series.

Table 6.6 presents the stability scores for the DBA results that incorporate the same indices as used previously. The subsequent analysis for the regions in our study is presented below:

- **Portugal -** The highest ARI stability score (0.68) is achieved by DBA with SW. The lowest ARI stability score (0.52) is observed for DBA with CH.
- **North Morocco -** DBA achieves the highest ARI stability score (0.57) with EL2, while the lowest score (0.53) is obtained for DBA with EL1.
- **South Morocco -** DBA with SW and EL2 achieved the highest ARI stability scores of 0.56. The lowest ARI stability score was observed for the DBA with CH, registering a value of 0.53.

	Portugal (PT)		North Morocco (NM)		South Morocco (SM)	
DBA	SS ARI	SS AMI	SS ARI	SS AMI	SS ARI	SS AMI
Silhouette index (SW)	**0.68**	0.67	0.54	0.61	**0.56**	0.63
Elbow Point 1 (EL1)	0.54	0.62	0.53	0.62	0.54	0.62
Elbow Point 2 (EL2)	0.55	0.62	**0.57**	0.65	**0.56**	0.64
Calinski–Harabasz index (CH)	0.52	0.63	0.54	0.63	0.53	0.63
Within-Between (WB)	0.53	0.63	0.54	0.63	0.54	0.63
Xu (XU)	0.53	0.63	0.54	0.63	0.54	0.63

Table 6.6: Stability Scores by ARI / AMI of the Time Ranges obtained by the DBA for the geographic regions of Portugal, North Morocco and South Morocco

The analysis of the results indicates that DBA with SW is the optimal configuration to achieve the highest stability score clusters in Portugal, and DBA with EL2 is the best configuration for the region of North Morocco. In South Morocco, DBA with SW and EL2 emerge as the preferred DBA indices to obtain stable time ranges, but the time ranges of the DBA with SW will be used as input for the Core-Shell clustering, similar to the previous algorithms.

6.4.5 Mean-Shift Time Ranges

In Section 4.3.4 we describe the Mean-Shift algorithm. The two hyperparameters for this algorithm are the kernel type and the bandwidth. To obtain the best Mean-Shift time ranges, we analyzed different configurations of its hyperparameters. This analysis involved the utilization of both Flat and Gaussian kernels, providing a thorough study of their influence on clustering results.

In our analysis of the bandwidth of the Mean-Shift algorithm, we used the *estimate_bandwidth()* function from the scikit-learn library [15]. This function takes a dataset and a quantile value as parameters, where the quantile ranges from 0 to 1. A quantile value of 0.5 indicates the use of the median of all pairwise distances to estimate the bandwidth.

Figure 6.7 shows the relationship between different quantile values and the estimated bandwidth resulting for the region of Portugal. As discussed in Section 4.3.4, a higher bandwidth value used in the Mean-Shift produces fewer clusters compared to a smaller bandwidth value. In our preliminary analysis, we observed that time ranges obtained with higher bandwidth values show higher stability scores compared to those with lower bandwidth values.

During this phase, we determined the quantile values that estimate the maximum bandwidth while avoiding results where only two time ranges are obtained. As explained in Section 6.4, an upwelling season is divided into at least three different periods. To achieve this, we will identify, by looking at the higher bandwidth values, the first year or years that reach two clusters. Following this, we will select the first quantile value lower than the one used to estimate the bandwidth that resulted in only two clusters. This way we ensure that we select a quantile value suited to estimate a bandwidth for every upwelling year. The analysis is as follows:

- **Portugal -** In Figure 6.8a, Mean-shift with Gaussian Kernel in 2005 results in two clusters with an estimated bandwidth of around 0.4. Similarly, Figure 6.7 shows a quantile value between 0.27 and 0.3. Similarly, Figure 6.8b for the Mean-Shift with Flat kernel in 2017 results in two clusters with an estimated bandwidth of approximately 0.36, corresponding to a quantile range of 0.27 to 0.3. To ensure clusters obtained by Mean-Shift have a minimum of 3 SST instants, we employ the estimated bandwidth with a quantile value of 0.25.

- **North Morocco -** Analyzing Figure B.6a for the Mean-Shift Gaussian kernel in 2019, two clusters with an estimated bandwidth of approximately 0.4 were obtained. This bandwidth corresponds to a quantile value between 0.27 and 0.30, as shown in Figure B.5. Similarly, in Figure B.6b for the Mean-Shift with Flat kernel in the same year, a bandwidth estimate of 0.35 to 0.39 resulted in two clusters. Therefore, we will employ an estimated bandwidth value using a quantile value of 0.25.

- **South Morocco -** In Figure B.8a, the Mean-Shift with Gaussian kernel in 2011 results in two clusters with an estimated bandwidth of around 0.37. As shown in Figure B.7, this bandwidth corresponds to a quantile value between 0.31 and 0.34. Similarly, in Figure B.8b for the Mean-Shift with Flat kernel in 2015, two clusters are obtained with an estimated bandwidth of approximately 0.37. Referencing Figure B.7, this bandwidth corresponds to a quantile range of 0.35 to 0.39. For the South Morocco region, we will estimate the bandwidth for the Gaussian kernel using a quantile value with the value of 0.28 and for the Flat kernel with a value of 0.32.

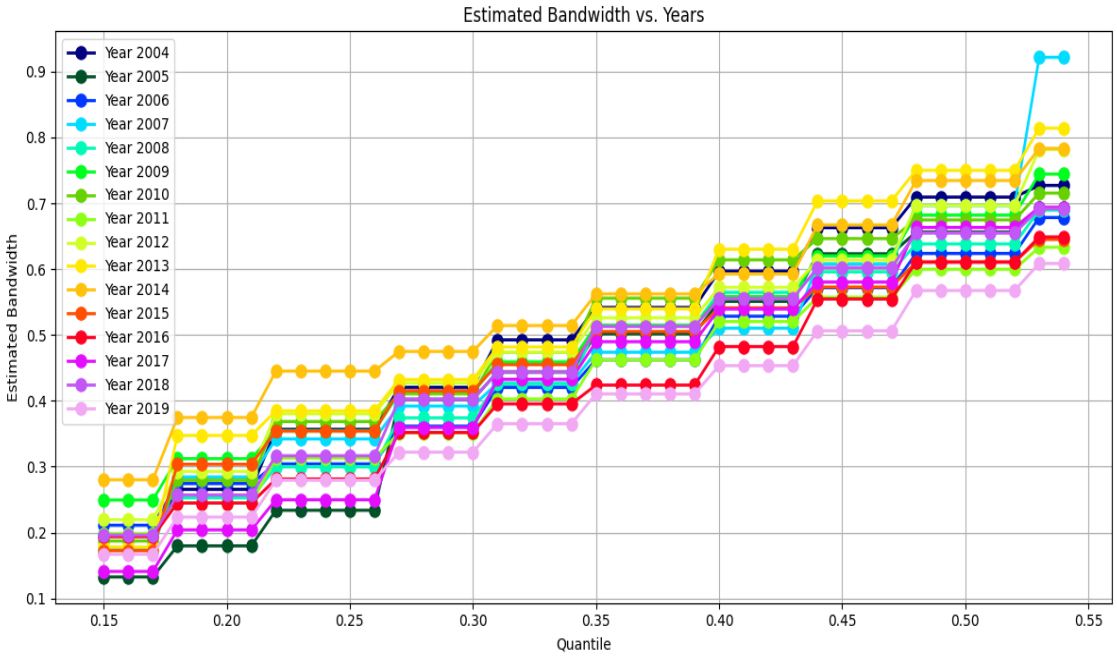
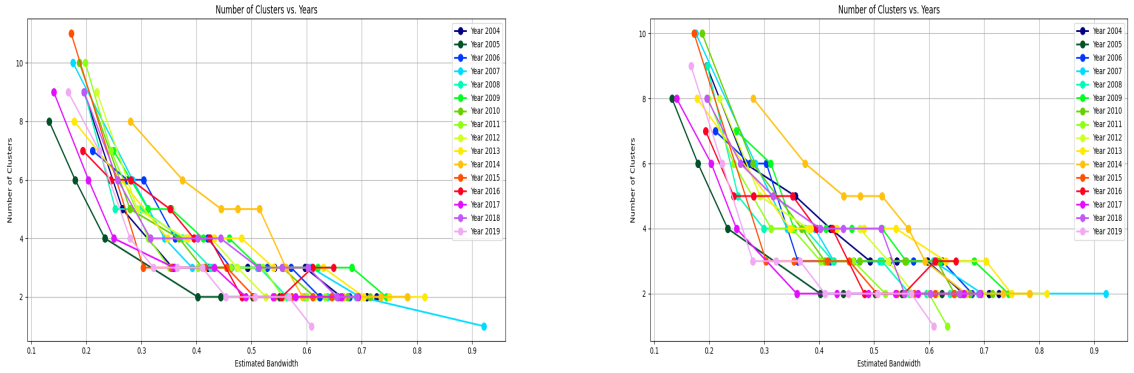


Figure 6.7: Estimated bandwidth for different quantile values across the years



(a) Gaussian kernel

(b) Flat kernel

Figure 6.8: Bandwidth analysis for the region of Portugal

From these quantile values, we created Table 6.7 with stability scores as in the previous algorithms and observed the following:

- **Portugal -** The Mean-Shift using the Gaussian kernel obtained a stability score of 0.56, slightly higher than the value 0.53 obtained by Mean-Shift using the Flat kernel.
- **North Morocco -** The best kernel for the Mean-Shift results continues to be the Gaussian kernel with a value of 0.53, with the Mean-Shift using the Flat kernel obtaining a stability score of 0.51.
- **South Morocco -** In the region of South Morocco, both the results of the Mean-Shift obtained a value of 0.55.

	Portugal (PT)		**North Morocco (NM)**		**South Morocco (SM)**	
Mean-Shift	SS ARI	SS AMI	SS ARI	SS AMI	SS ARI	SS AMI
Flat	0.53	0.60	0.51	0.61	**0.55**	0.62
Gaussian	**0.56**	0.62	**0.53**	0.62	**0.55**	0.62

Table 6.7: Stability Scores by ARI / AMI of the Time Ranges obtained by the Mean-Shift for the geographic regions of Portugal, North Morocco and South Morocco

From the analysis of the tables, it is clear that Mean-Shift with the Gaussian kernel consistently obtains more stable time ranges than Mean-Shift with the Flat kernel. Although in the region of southern Morocco, the result was similar, we will only provide the time ranges obtained from the Mean-Shift with Gaussian kernel to the Core-Shell clustering algorithm.

6.4.6 Comparing the Time Ranges Stability

For each geographic region, we will compare the time ranges obtained by IAP against the best stability scores obtained with K-means, K-means++, DBA and Mean-Shift, measuring

their stability with the implemented stability score measure. The time ranges analyzed in this subsection were used as input to the Core-Shell algorithm.

	Portugal (PT)		**North Morocco (NM)**		**South Morocco (SM)**	
Algorithms	SS ARI	SS AMI	SS ARI	SS AMI	SS ARI	SS AMI
IAP	**0.72**	0.70	0.57	0.64	**0.70**	0.64
K-means	0.60	0.63	**0.58**	0.66	0.58	0.65
K-means++	0.59	0.63	**0.58**	0.65	0.58	0.65
DBA	0.68	0.67	0.57	0.65	0.56	0.63
Mean-Shift	0.56	0.62	0.53	0.62	0.55	0.62

Table 6.8: Stability Scores of the Time Ranges obtained by all algorithms for the geographic regions of Portugal, North Morocco and South Morocco

- **Portugal -** The IAP algorithm achieved the highest stability score in this region, obtaining a value of 0.72. The second highest value was obtained by the DBA algorithm with a score of 0.68. Subsequently, K-means obtained a stability score of 0.6, while K-means++ obtained a score of 0.59. The Mean-Shift algorithm got the lowest stability score, with a value of 0.56.

- **North Morocco -** The K-means and K-means++ obtained the highest stability scores in this region, achieving a value of 0.58. Following with the next highest score, both the IAP and DBA algorithms obtained a score of 0.57. The Mean-Shift algorithm obtained the lowest stability score, with a score of 0.53.

- **South Morocco -** In this region, the IAP algorithm obtained the highest stability score of 0.70. K-means and K-means++ achieved a stability score of 0.58. Subsequently, the DBA achieved a stability score of 0.56. At the lower end, the Mean-Shift algorithm obtained the lowest stability scores, with a score of 0.55.

The IAP algorithm clearly obtained the most stable time ranges for the geographic regions of Portugal and South Morocco. Although in North Morocco, both K-means and K-means++ obtained slightly better results, we can note that for the two regions with a higher intensity in costal upwelling, that is, Portugal and South Morocco, IAP is clearly able to determine the periods of time where the upwelling has a similar behavior.

Computation time [s]	**Portugal**	**North Morocco**	**South Morocco**
IAP	0.002	0.002	0.002
K-means	0.315	0.288	0.297
K-means++	0.253	0.252	0.256
DBA	2.726	2.670	2.705
Mean-Shift	0.029	0.022	0.027

Table 6.9: Average computation time in seconds

The efficiency of each algorithm was also taken into account by measuring the average computation time for each geographic region. The computation time in this analysis was measured by the time of both the computation of each algorithm and the post-processing of the results.

Both the IAP algorithm and Mean-Shift are deterministic clustering algorithms, meaning that their execution produces consistent and predictable results without any randomness involved. This inherent determinism contributes to their efficiency. In contrast, algorithms such as K-means, K-means++, and DBA require multiple initializations for each value of K, in our case ranging from 2 to 7. This additional computational step significantly increases the processing time, making them less efficient. In particular, DBA obtained the poorest computation time because of its expensive procedure and the necessity for multiple initializations.

6.5 Evaluation Core-Shell clustering results

To assess the accuracy of the Core-Shell clustering segmentations, we used the segmentations obtained through the S-STSEC algorithm as ground truth. These segments received favorable evaluation scores after being reviewed by a group of experienced oceanographers in the works developed in: [32, 42, 43].

We evaluated the results of the Core-Shell clusters using five validity measures described in Section 5.5: Adjusted Rand index (ARI), Kulczynski similarity index (KS), Weighted Kulczynski similarity index (WKS), Normalized Mutual Information (NMI) and Mirkin distance. These measures determine the level of similarity between the results, except for the Mirkin distance, which calculates the degree of dissimilarity.

The results are discussed in the next sections organized by geographic region.

6.5.1 Portugal

Table 6.10 shows the evaluation of the Core-Shell clustering segmentation, initialized with time ranges obtained by IAP, K-means++, K-means, DBA, and Mean-Shift, with the five validity measures mentioned previously.

The segmentations produced by various algorithms exhibited favorable results on the five validity measures. The IAP algorithm obtained the best scores, with an ARI of 0.77 ± 0.031, a WKS of 0.874 ± 0.012, a KS of 0.978 ± 0.005, NMI of 0.655 ± 0.033, and Mirkin distance of 0.134 ± 0.023. The Mean-Shift algorithm achieved scores similar to IAP scores in most indices, with a slightly lower value only in the WKS index with a value of 0.873 ± 0.016. The K-means++ algorithm obtained similar results, with its KS score matching that of IAP at 0.978 ± 0.005.

These results indicate that the partitions obtained by all algorithms were of good quality and did not deviate significantly from the average results. Of all algorithms,

the IAP and Mean-Shift algorithms obtained the time ranges that led to the overall best Core-Shell clusters based on the values of the five indices.

Core-shell initialization by:	IAP	K-means++	K-means	DBA	Mean-Shift
ARI (↑)	0.77 ± 0.031	0.76 ± 0.040	0.757 ± 0.040	0.749 ± 0.039	0.77 ± 0.040
WKS (↑)	0.874 ± 0.012	0.87 ± 0.017	0.869 ± 0.017	0.868 ± 0.014	0.873 ± 0.016
KS (↑)	0.978 ± 0.005	0.978 ± 0.006	0.977 ± 0.006	0.976 ± 0.006	0.978 ± 0.005
NMI (↑)	0.655 ± 0.033	0.644 ± 0.043	0.642 ± 0.043	0.634 ± 0.040	0.655 ± 0.043
Mirkin distance (↓)	0.134 ± 0.023	0.142 ± 0.028	0.145 ± 0.029	0.150 ± 0.029	0.134 ± 0.029

Table 6.10: Mean values of the indexes results for the region of Portugal

6.5.2 North Morocco

Taking into account the results of the northern Morocco region, Table 6.11 shows the evaluation of Core-Shell clustering segmentation for the region in analysis.

Similarly to the results obtained in the region of Portugal, the segmentations produced by all algorithms exhibited favorable results across the five validity measures. In this region, the K-means++ algorithm obtained the best scores, with an ARI of 0.808 ± 0.036, a WKS of 0.856 ± 0.012, a KS of 0.971 ± 0.006, NMI of 0.696 ± 0.042, and Mirkin distance of 0.084 ± 0.019.

These results indicate that the partitions obtained by all algorithms were of good quality, similar to the region of Portugal, and did not deviate significantly from the average results. Of all the clustering algorithms, K-means++ time ranges resulted in the best Core-Shell clusters according to the values of the five indices in this region.

Core-shell initialization by:	IAP	K-means++	K-means	DBA	Mean-Shift
ARI (↑)	0.78 ± 0.038	0.808 ± 0.036	0.796 ± 0.048	0.794 ± 0.047	0.792 ± 0.045
WKS (↑)	0.848 ± 0.012	0.856 ± 0.012	0.851 ± 0.016	0.85 ± 0.017	0.852 ± 0.016
KS (↑)	0.97 ± 0.006	0.971 ± 0.006	0.97 ± 0.008	0.97 ± 0.007	0.971 ± 0.007
NMI (↑)	0.664 ± 0.042	0.696 ± 0.042	0.683 ± 0.055	0.681 ± 0.055	0.679 ± 0.052
Mirkin distance (↓)	0.099 ± 0.024	0.084 ± 0.019	0.091 ± 0.026	0.092 ± 0.027	0.094 ± 0.025

Table 6.11: Mean values of the indexes results for the region of Northern Morocco

6.5.3 Southern Morocco

Finally, we will analyze the results of the South Morocco region. Table 6.12 shows the evaluation of Core-Shell clustering segmentation for the region in the analysis.

In South Morocco, the segmentations produced by the various algorithms presented good results in the five stability measures. In this region and similar to Portugal, the IAP algorithm obtained the best scores, with an ARI of 0.817 ± 0.025, a WKS of 0.84 ± 0.011, a KS of 0.984 ± 0.004, NMI of 0.71 ± 0.030, and Mirkin distance of 0.109 ± 0.016.

After analyzing all the regions, we conclude that all the results obtained indicate that the time ranges obtained by all algorithms were of good quality and did not deviate significantly from the average results.

Core-shell initialization by:	IAP	K-means++	K-means	DBA	Mean-Shift
ARI (↑)	0.817 ± 0.025	0.811 ± 0.033	0.807 ± 0.031	0.809 ± 0.030	0.816 ± 0.032
WKS (↑)	0.84 ± 0.011	0.838 ± 0.011	0.837 ± 0.011	0.836 ± 0.012	0.839 ± 0.012
KS (↑)	0.984 ± 0.004	0.983 ± 0.004	0.983 ± 0.004	0.983 ± 0.004	0.983 ± 0.004
NMI (↑)	0.71 ± 0.030	0.703 ± 0.037	0.698 ± 0.034	0.7 ± 0.035	0.709 ± 0.037
Mirkin distance (↓)	0.109 ± 0.016	0.114 ± 0.022	0.116 ± 0.021	0.115 ± 0.020	0.11 ± 0.021

Table 6.12: Mean values of the indexes results for the region of Southern Morocco

6.6 Analysis of Upwelling Core Features

The goal of this experiment is to analyze whether the time series of each of the two main features extracted from the upwelling cores: the average temperature and the total area, provide meaningful results when segmented by the popular DBA algorithm widely used for estimating the mean of a given set of point sequences. The Core-Shell segmentation used received the IAP time ranges as input.

We constructed an entity-to-feature data matrix whose rows are the SST instants (23) that constitute an upwelling season and the columns are the feature (average temperature or total area) along the years under analysis, 16 years in our case. The entry of i, j in the matrix is the core feature value for SST instant i in year j. Each line in the graph of Figure 6.9 displays the average temperatures of the cores throughout a one-year upwelling season (stored in one column of the entity-to-feature data matrix).

We easily observe that at SST instant 7 (which corresponds to the week of 13th-19th of May) the cores' average temperatures are very close. Instant 7 marks the transition to a higher intensity upwelling stage.

DBA algorithm was run taking the former core average temperature (and total area) matrix as input, ranging the number of clusters K from 2 to 7.

In the following sections, a regional analysis of the results is performed for each core feature: average temperature and total area.

6.6.1 Upwelling Cores' Temperature

6.6.1.1 Portugal

Figure 6.10 displays a plot of the Silhouette validity index for the DBA partitions. Based on the results, the DBA K partition with $K = 2$ clusters is selected.

The two DBA clusters obtained for the region of Portugal are shown in Figure 6.11. Analyzing the two groups, we can note that a trend is visible in the first cluster, with most

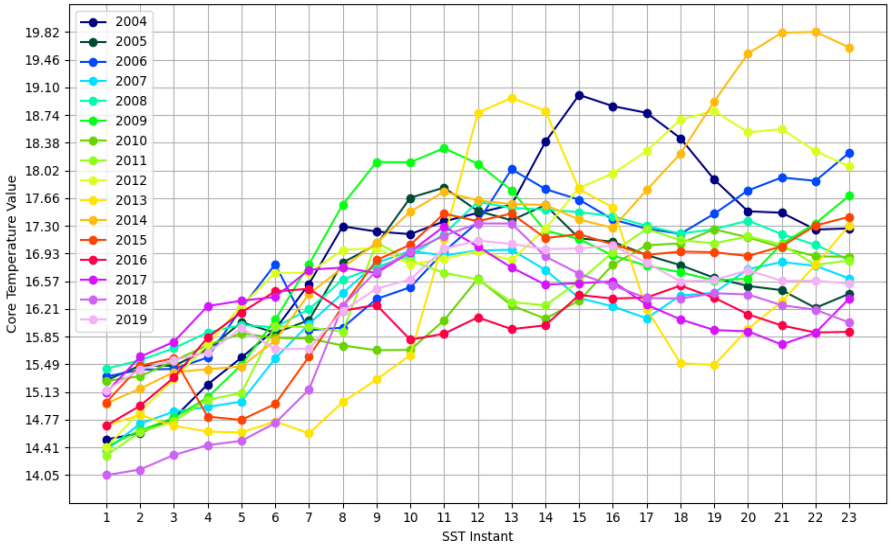


Figure 6.9: Core temperature values for the sixteen years across the 23 SST instants

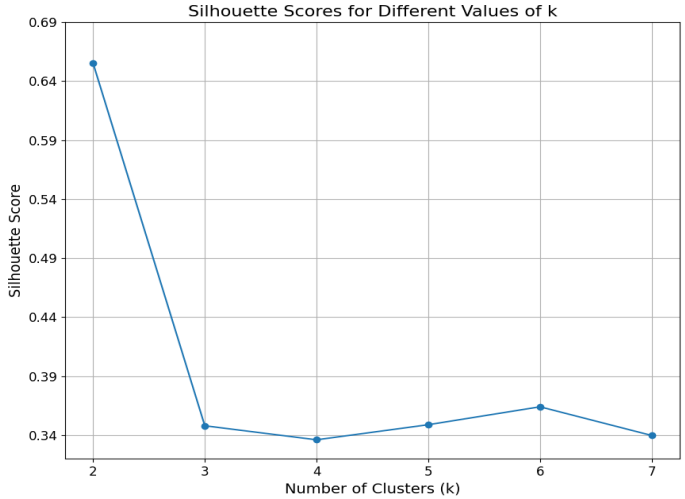


Figure 6.10: Silhouette scores for different values of K using DBA with the core temperature for the region of Portugal

of the years having a similar increase in temperature along the first 7 SST instants. This first cluster also represents the initial upwelling stage, where the upwelling intensity is still low. Near the "border" between clusters, we can observe that the temperature values are close to each other. The second cluster covering SST instants 8 to 23, has years with variate trends, with some years like 2013 in light orange having both increasing and decreasing temperature slopes, and others like 2016, with a more stable temperature value across the SST instants.

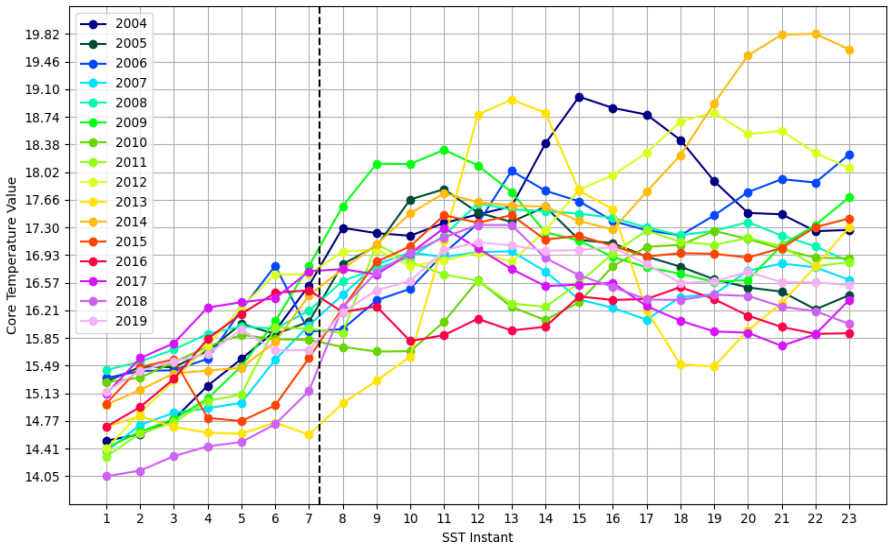


Figure 6.11: Core temperature values across the 23 SST instants with each of the 16 years represented and "border" separating the different clusters for the region of Portugal

Taking the same DBA 2-partition, Figure 6.12, show a graphic for each cluster where each gray line is the temperature value of each SST instant, the green line is the average temperature value of each SST instant, and the red line is the barycenter of DBA.

Analyzing the first cluster, the DBA barycenter captures the low upwelling intensity present in all the initial SST instants. In the second cluster, although it is more irregular due to the increase in the upwelling intensity, we can observe that the DBA barycenter represents well the averaging of the different instants for every year.

Before proceeding to the analysis of the other regions, we will show a careful (and similar) analysis for the case of DBA k = 3. Figure 6.13 shows the three groups of SST instants that can be divided as follows: the beginning of the upwelling season (from the SST instant 1 to 6), the middle of the upwelling season (from the SST instant 7 to 10) and the end of the upwelling season (from SST instant 11 to 23).

From the analysis of these groups of SST instants, it can be observed that the beginning of the upwelling season is characterized by a low upwelling intensity, where the core temperatures of all years are at the lowest values. The intermediate cluster is the period where the upwelling intensity increases, also characterized by an increase in the temperature values. The third cluster is represented by various changes in temperature

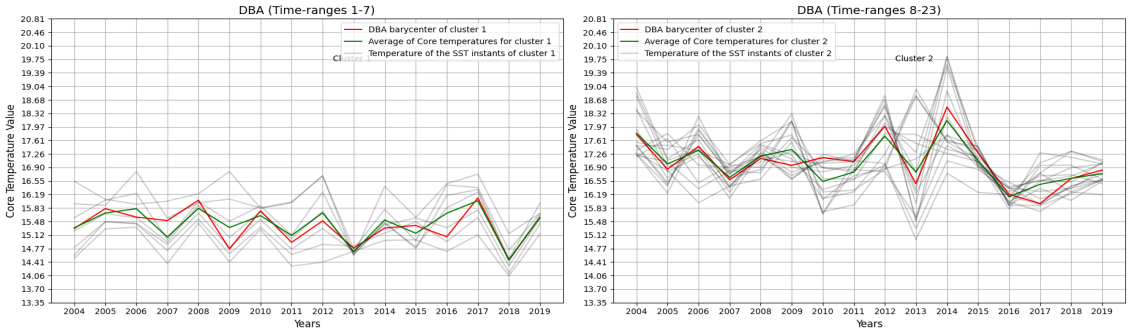


Figure 6.12: Core temperature values across 16 years divided by each cluster obtained by the DBA algorithm, with the SST instant temperatures of each year, the value of the DBA barycenter and the average of the core temperature for the region of Portugal

for some years, but for most of the years analyzed, this cluster is when the increase in the temperature value stops.

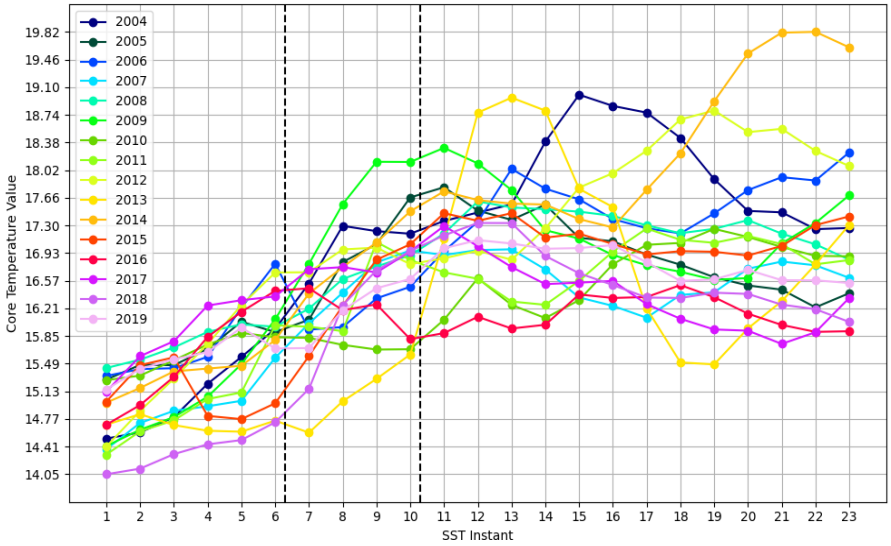


Figure 6.13: Core temperature values across the 23 SST instants with each of the 16 years represented and "border" separating the different clusters for the region of Portugal for K = 3

6.6.1.2 North Morocco

Analyzing the region of North Morocco, Figure B.9 shows the different Silhouette values and K equal to 2 will be used since it obtained the highest Silhouette value.

The DBA 2-partition clusters for the region of North Morocco are shown in Figure B.10, where the second cluster obtained starts at SST instant 10.

The temperature values in this region do not vary as much as those in the Portugal region. But similar to the observed in the previous region analysis, the first cluster, which represents the start of the upwelling season, has a trend with a constant increase in

the temperature while in the second cluster, different behaviors for multiple years are observed. For example, the year 2016 from SST instant 12 to SST instant 13 increased the temperature value by more than three degrees Celsius.

Analyzing Figure B.11, we observe a similar behavior to the Portugal region, concluding that the barycenter of DBA captures the two main different periods during an upwelling season, one where the upwelling intensity is lower and the other where the upwelling intensity increases and eventually stabilizes.

Before analyzing the core temperature for the region of South Morocco, we will also analyze the case of DBA k = 3. Figure B.12 shows the three groups of SST instants that can be divided as follows: the beginning of the upwelling season (from the SST instant 1 to 7), the middle of the upwelling season (from the SST instant 8 to 12) and the end of the upwelling season (from SST instant 12 to 23).

The beginning of the upwelling season is characterized by a low upwelling intensity. The middle cluster is when the upwelling intensity increases, distinguished by an increase in temperature. In this region, we can observe that some years have a high increase in temperature value, like the year 2013 for example. The final cluster is characterized by multiple changes in temperature, although all temperature values in this cluster are contained in a small temperature range (between 18.32 and 21.20 degrees Celcius).

6.6.1.3 South Morocco

In Figure B.13 we show the different Silhouette scores for the region of South Morocco, where K equal to 2 was once again the highest Silhouette value.

The two DBA clusters obtained for the region of South Morocco are presented in Figure B.14.

When comparing the years, we observe that of all the previous regions analyzed, the South Morocco region has the years with the most similar temperature variation. A trend of a consistent increase in temperature in the first cluster is once again observed, with the second cluster being characterized by the stabilization of the temperature increases for most years. In the "border" represented by the vertical line, we can see that most of the years have similar temperature values.

Analyzing Figure B.15 we again observe a similar trend between the three different geographic regions, confirming the meaningful clusters obtained by the DBA algorithm.

Observing the case of DBA k = 3 in Figure B.16, the three groups of SST instants are divided as follows: the beginning of the upwelling season (from the SST instant 1 to 6), the middle of the upwelling season (from the SST instant 7 to 11) and the end of the upwelling season (from SST instant 12 to 23).

South Morocco has more consistent changes in temperature values, as mentioned previously. This is observed in the beginning cluster, where the upwelling intensity is at its lowest point, with little change in the temperature value. Then, in the middle cluster, the increase in the upwelling intensity is visible, with the temperature values increasing

across all years. Finally, the ending cluster is where the value temperatures are at the highest temperature values for all years.

In summary of the analysis of the core temperatures for the three different regions, we can conclude that DBA obtains meaningful clustering and averaging time series with respect to the upwelling periods of stability.

6.6.2 Upwelling Cores' Areas

6.6.2.1 Portugal

We will now analyze the core areas. Figure 6.14 displays a plot of the Silhouette validity index for DBA partitions that use this feature. According to the results, the DBA K partition with K = 2 clusters is selected.

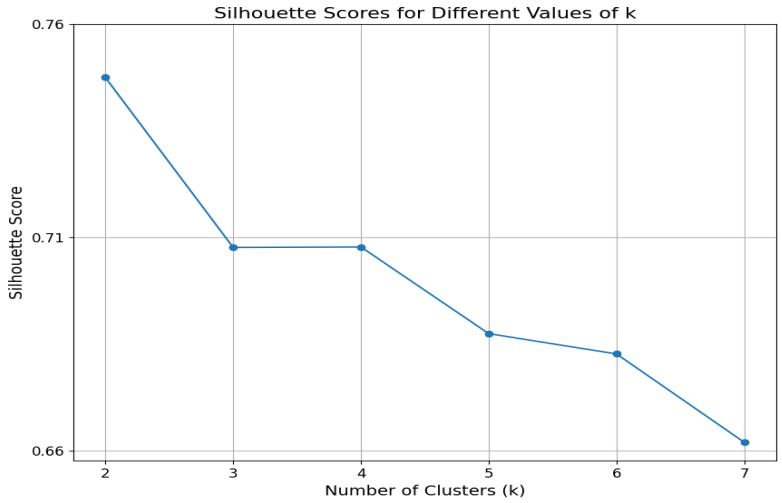


Figure 6.14: Silhouette scores for different values of K using DBA with the Core areas for the region of Portugal

The DBA 2-partition clusters obtained for the region of Portugal is presented in Figure 6.15.

When observing the information on the areas, we note that there is some information common between years. Note that since a time range only has a single Core-Shell cluster, composed of one Core and multiple Shells, the Core areas are constant for each time range obtained. Analyzing the graph, we note that there is a constant interval in areas at the start of the first cluster and the end of the second cluster. From SST instants 6 to 12, we have an increase in the intensity of the upwelling seasons and, because of that, a variation in the core area values.

Taking the same DBA 2-partition Figure 6.16 shows a graphic for each cluster where each gray line is the area value of each SST instant, the green line is the average of the area value of each SST instant, and the red line is the DBA barycenter. Analyzing the first

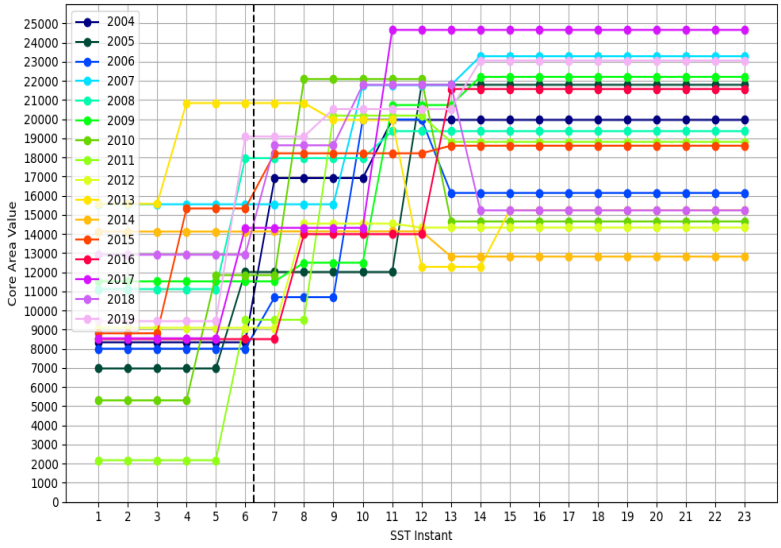


Figure 6.15: Core area values across the 23 SST instants with each of the 16 years represented and "border" separating the different clusters for the region of Portugal

cluster, the DBA barycenter captures the low upwelling intensity present in all the initial SST instants, which is represented by a constant area value across SST instants. In the second cluster, although it is more irregular due to the increase in the upwelling intensity, we can observe that the DBA barycenter captures the averaging of the different instants for every year.

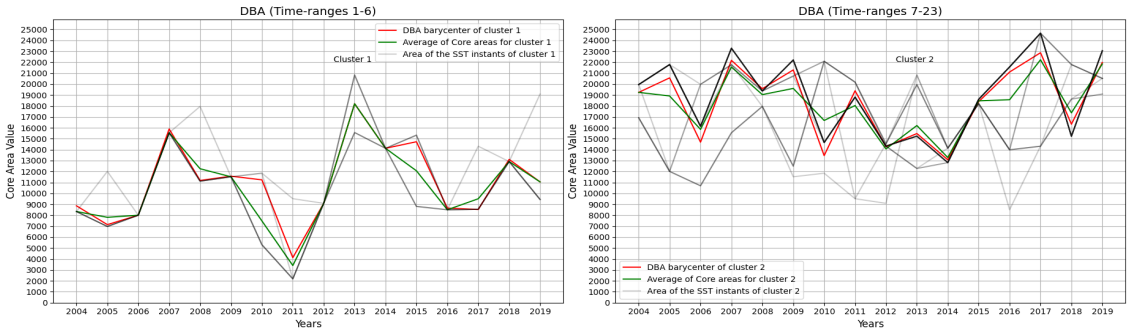


Figure 6.16: Core area values across 16 years divided by each cluster obtained by the DBA algorithm, with the SST instant areas of each year, the value of the DBA barycenter and the average of the core areas for the region of Portugal

Before proceeding to the analysis of the other regions, we will show an analysis of the case of DBA k = 3 for this feature. Figure 6.17 shows the three groups of SST instants that can be divided as follows: the beginning of the upwelling season (from the SST instants 1 to 5), the middle of the upwelling season (from the SST instants 6 to 10) and the end of the upwelling season (from SST instant 11 to 23).

The difference across each obtained cluster is more evident when the area features are analyzed. The beginning cluster once again represents the period in each year where the

upwelling intensity is at its lowest. Then, in the middle group, we can observe a period of variation in the values of the total area for each year. Also in this period, the upwelling intensity is at its highest. Then, the ending cluster is characteristic of a consistent area value across all years until the end of each upwelling season.

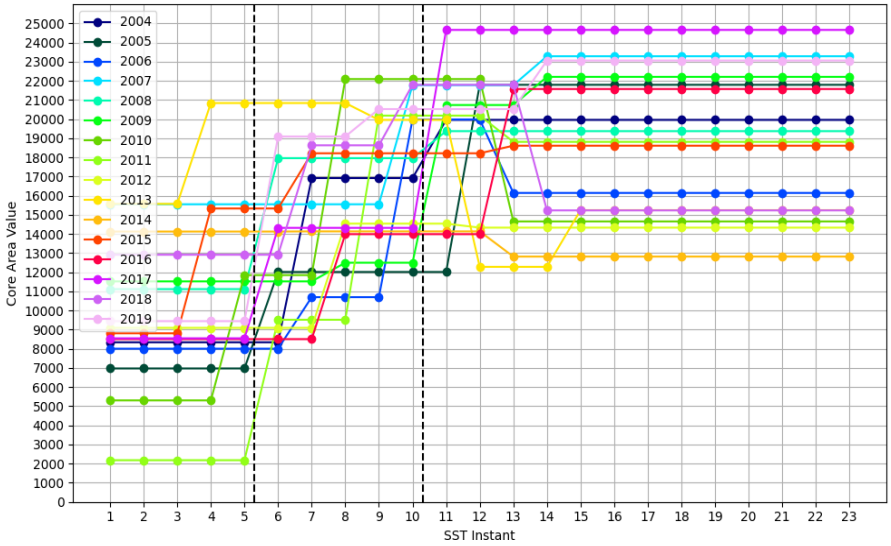


Figure 6.17: Core area values across the 23 SST instants with each of the 16 years represented and "border" separating the different clusters for the region of Portugal for K = 3

6.6.2.2 North Morocco

Analyzing the region of North Morocco, Figure B.17 shows the different Silhouette values and K equal to 2 will be used since it obtained the highest Silhouette value.

The DBA 2-partition clusters for the region of North Morocco are shown in Figure B.18.

The area values in this region do not vary as much as the areas in the Portugal region. North Morocco shows two clear groups of areas, and the "border" between the clusters represented by the vertical line represents the increase in the intensity of upwelling.

Analyzing Figure B.19, we observe a similar behavior to the Portugal region, concluding that the barycenter DBA captures the two main stages during an upwelling season, one stage where the upwelling intensity is low and the other where the upwelling intensity increases and then stabilizes.

Before analyzing the core area for the region of South Morocco, we will also analyze the case of DBA k = 3. Figure B.20 shows the three groups of SST instants that can be divided as follows: the beginning of the upwelling season (from the SST instants 1 to 8), the middle of the upwelling season (from the SST instants 9 to 10) and the end of the upwelling season (from SST instant 11 to 23).

The middle cluster obtained for this region contains only 2 SST instants. One thing we can note is that, compared to the Portugal region, there is a small set of SST instants

where the upwelling intensity increases (from SST instants 6 to 11). In SST instants 7 and 8, most of the years have this change in intensity, and the DBA captured this behavior. The beginning and ending clusters are once again characterized by constant area values over the years.

6.6.2.3 South Morocco

In FigureB.21 we present the different Silhouette scores for the region of South Morocco, where K equal to 6 was the highest Silhouette value.

The six DBA clusters obtained for the region of South Morocco are presented in Figure B.22. We have the first and last clusters containing most of the SST instants, and the intermediate clusters containing the variations in the Core area value.

When comparing the years, we observe that of all the regions, the South Morocco region has the years with the most similar area variation.

Analyzing now Figure B.23 we cannot conclude meaningful observations since each plot has a low number of SST instants, making the plots difficult to draw conclusions.

Finally, we will analyze the core area for the South Morocco region for the case of DBA k = 3. Figure B.20 shows the three groups of SST instants that can be divided as follows: the beginning of the upwelling season (from the SST instants 1 to 7), the middle of the upwelling season (from the SST instants 8 to 15), and the end of the upwelling season (from the SST instants 16 to 23).

The DBA was able to separate the two stable periods in an upwelling season into clusters, in the beginning and ending clusters, and the period where the upwelling intensity is at its highest, in the middle cluster.

In summary of the analysis of the core features for different regions, we can conclude that DBA obtains meaningful clusters that correctly represent the coastal upwelling stability in different regions of the world.

6.7 Summary

Reviewing the results obtained, it was demonstrated that the feature selection via correlation analysis and the experimental procedure for optimizing the hyperparameters of the clustering algorithms were effective in determining the optimal configuration of each algorithm for the unsupervised definition of upwelling time ranges.

With the implementation of the stability score measure, we were able to analyze which algorithms obtained the most stable time ranges. Overall, the IAP algorithm proved once again to be an appropriate choice for such a task, especially for the geographical regions of Portugal and South Morocco.

After selecting the best time ranges and executing the Core-Shell clustering algorithm, the segmentations obtained showed that the cores capture very well the constant regions of an upwelling event, being also able to find meaningful clusters that represent the different

stages of an upwelling season when clustered by the DBA algorithm. The segmentations produced by the Core-Shell clustering algorithm demonstrated favorable overall results compared to the corresponding SST instants when evaluated using the state-of-the-art clustering validation indices.

The work was developed using a Lenovo Legion 5 Pro, with a 3.2GHz 8-Core AMD Ryzen™ 7 5800H processor, 16 GB 3200 MHz DDR4 of memory and an NVIDIA GeForce RTX 3070 8 GB graphics card, on the Ubuntu 22.04.2 LTS operating system.

7

Conclusion and Future Work

The work developed in this dissertation addresses the limitations of the Core-Shell clustering framework by exploring the effectiveness of multiple clustering algorithm in segmenting time series data to define time ranges. In addition, it incorporates a more robust validation procedure to assess the quality of Core-Shell segmentations.

The adaptability of the framework was further evaluated through the analysis of three collections of SST images for the period 2004 to 2019, covering the geographic regions of Portugal, North Morocco and South Morocco. This allowed for the unsupervised definition and analysis of coastal upwelling patterns in these regions, making it possible to evaluate the efficiency of the Core-Shell framework in various geographic settings.

In order to ensure the quality of the results between the different regions of the study, an initial step involved conducting feature selection by analyzing correlations for each region using the features derived from the S-STSEC segmentation time series.

Such features were then used in the comparative study of the IAP algorithm with K-means, K-means++, DBA and Mean-Shift. The objective was to identify various periods of upwelling stability during each upwelling season, characterized by similar and consistent upwelling characteristics.

To evaluate the results obtained, a stability score measure was implemented to measure the stability of the time ranges obtained over the years. The IAP algorithm obtained the most stable time ranges for the geographic regions of Portugal and South Morocco, while K-means++ identified the most stable time ranges in the northern region of Morocco.

The best upwelling time ranges of each algorithm were used by the Core-Shell clustering algorithm to obtain Core-Shell segmentations. The results were evaluated by comparing the obtained Core-Shell segmentations with the SST images of the respective time ranges. Various state-of-the-art validity measures were employed to evaluate the segmentations, obtaining overall good mean values.

Examining the long-term inter-annual upwelling time series derived from the core segmentation revealed that the Core-Shell algorithm consistently captures meaningful periods of stability.

Regarding future work following the progress made in this dissertation, there are two

aspects that can be further explored:

- Extend furthermore the algorithms used for time series segmentation in defining upwelling time ranges. Until now, only partitional clustering algorithms have been employed. In the initial phase of the study, SWAB, a change point detection algorithm, was initially utilized. However, fine-tuning its hyperparameters proved challenging, leading to its eventual abandonment. There is the possibility that different algorithms could obtain better results. Therefore, a comprehensive study may be conducted to explore the efficacy of employing new algorithms in this context.
- Develop a study in which the data of all years is analyzed collectively, regardless of the upwelling seasons. Instead of creating separate datasets for each year, consolidate the data from all years into a single dataset.

Bibliography

[1] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah. "Time-series clustering–a decade review". In: *Information systems* 53 (2015), pp. 16–38 (cit. on pp. 19–21).

[2] K. Agrawal et al. "Development and validation of OPTICS based spatio-temporal clustering technique". In: *Information Sciences* 369 (2016), pp. 388–401 (cit. on pp. 12, 13).

[3] M. A. Alkhamis et al. "Spatiotemporal dynamics of the COVID-19 pandemic in the State of Kuwait". In: *International Journal of Infectious Diseases* 98 (2020), pp. 153–160 (cit. on p. 11).

[4] S. Aminikhanghahi and D. J. Cook. "A survey of methods for time series change point detection". In: *Knowledge and information systems* 51.2 (2017), pp. 339–367 (cit. on pp. 18, 32, 34).

[5] M. Y. Ansari et al. "Spatiotemporal clustering: a review". In: *Artificial Intelligence Review* 53.4 (2020), pp. 2381–2423 (cit. on pp. 8, 9).

[6] G. Atluri, A. Karpatne, and V. Kumar. "Spatio-temporal data mining: A survey of problems and methods". In: *ACM Computing Surveys (CSUR)* 51.4 (2018), pp. 1–41 (cit. on p. 8).

[7] E. D. Barton and J. Arístegui. "The Canary Islands coastal transition zone–upwelling, eddies and filaments". In: (2004) (cit. on p. 54).

[8] M. R. Berthold and F. Höppner. "On clustering time series using euclidean distance and pearson correlation". In: *arXiv preprint arXiv:1601.02213* (2016) (cit. on p. 21).

[9] D. Birant and A. Kut. "ST-DBSCAN: An algorithm for clustering spatial–temporal data". In: *Data & knowledge engineering* 60.1 (2007), pp. 208–221 (cit. on pp. 11, 14).

[10] M. A. Carreira-Perpinán. "A review of mean-shift algorithms for clustering". In: *arXiv preprint arXiv:1503.00687* (2015) (cit. on pp. 30–32).

[11] X. Chen et al. "Clustering dynamic spatio-temporal patterns in the presence of noise and missing data". In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015 (cit. on pp. 14, 15).

[12] B. Costa et al. "Fault Classification on Transmission Lines Using KNN-DTW". In: 2017-07, pp. 174–187. ISBN: 978-3-319-62391-7. DOI: 10.1007/978-3-319-62392-4_13 (cit. on pp. 22, 25).

[13] I. Dabbura. *K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks*. https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a. [Online; accessed 25-September-2023]. 2018 (cit. on p. 24).

[14] S. Daoudi et al. "Parallelization of the K-Means++ Clustering Algorithm." In: *Ingénierie des Systèmes d'Information* 26.1 (2021) (cit. on p. 27).

[15] scikit-learn developers. *sklearn.cluster.estimate$_b$andwidth*. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.estimate_bandwidth.html. [Online; accessed 6-September-2023]. 2023 (cit. on p. 65).

[16] M. G. Doborjeh and N. Kasabov. "Dynamic 3D clustering of spatio-temporal brain data in the NeuCube spiking neural network architecture on a case study of fMRI data". In: *International Conference on Neural Information Processing*. Springer. 2015, pp. 191–198 (cit. on p. 10).

[17] A. El Aouni et al. "Physical and biological satellite observations of the northwest african upwelling: Spatial extent and dynamics". In: *IEEE Transactions on Geoscience and Remote Sensing* 58.2 (2019), pp. 1409–1421 (cit. on pp. 2, 6).

[18] A. El Aouni et al. "Robust detection of the North-West African upwelling from SST images". In: *IEEE Geoscience and Remote Sensing Letters* 18.4 (2020), pp. 573–576 (cit. on pp. 5, 6, 38).

[19] J. Gomide et al. "Dengue surveillance based on a computational model of spatio-temporal locality of Twitter". In: *Proceedings of the 3rd international web science conference*. 2011, pp. 1–8 (cit. on p. 11).

[20] M. M. Gösgens, A. Tikhonov, and L. Prokhorenkova. "Systematic analysis of cluster similarity indices: How to validate validation measures". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 3799–3808 (cit. on pp. 49, 51).

[21] Y. Hu et al. "A spatio-temporal kernel density estimation framework for predictive crime hotspot mapping and evaluation". In: *Applied geography* 99 (2018), pp. 89–97 (cit. on p. 11).

[22] M. Hüsch, B. U. Schyska, and L. von Bremen. "CorClustST—Correlation-based clustering of big spatio-temporal datasets". In: *Future Generation Computer Systems* 110 (2020), pp. 610–619 (cit. on p. 13).

[23] J. Kämpf et al. "The functioning of coastal upwelling systems". In: *Upwelling Systems of the World: A Scientific Journey to the Most Productive Marine Ecosystems* (2016), pp. 31–65 (cit. on p. 6).

[24] E. Keogh et al. "An online algorithm for segmenting time series". In: *Proceedings 2001 IEEE international conference on data mining*. IEEE. 2001, pp. 289–296 (cit. on pp. 35, 44).

[25] G. Klassen, M. Tatusch, and S. Conrad. "Cluster-based stability evaluation in time series data sets". In: *Applied Intelligence* 53.13 (2023), pp. 16606–16629 (cit. on p. 48).

[26] T. W. Liao. "Clustering of time series data—a survey". In: *Pattern recognition* 38.11 (2005), pp. 1857–1874 (cit. on p. 20).

[27] J. Liu et al. "Dual-constraint spatiotemporal clustering approach for exploring marine anomaly patterns using remote sensing products". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11.11 (2018), pp. 3963–3976 (cit. on p. 10).

[28] S. Lloyd. "Least squares quantization in PCM". In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137 (cit. on p. 24).

[29] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. ii).

[30] M. Lovrić, M. Milanović, and M. Stamenković. "Algoritmic methods for segmentation of time series: An overview". In: *Journal of Contemporary Economic and Business Issues* 1.1 (2014), pp. 31–53 (cit. on pp. 32, 34).

[31] J. MacQueen et al. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297 (cit. on p. 24).

[32] A. G. Martins. "Unsupervised Spatio-Temporal Analysis of Coastal Upwelling from Sea Surface Temperature Images". Master's thesis. NOVA University of Lisbon, 2022 (cit. on pp. iv, vi, 3, 36–38, 48, 53–55, 69).

[33] M. Meilă. "Comparing clusterings—an information based distance". In: *Journal of multivariate analysis* 98.5 (2007), pp. 873–895 (cit. on pp. 49, 51, 52).

[34] B. Mirkin. *Clustering: A Data Recovery Approach, Vol. 19*. 2012 (cit. on pp. 39, 41).

[35] E. Mizutani. "The dynamic time warping algorithms". In: *Mechanical Engineering Seminar, Tokyo Metropolitan University*. 2006, p. III (cit. on p. 22).

[36] MLNerds. *How to find the Optimal Number of Clusters in K-means? Elbow and Silhouette Methods*. https://machinelearninginterview.com/topics/machine-learning/how-to-find-the-optimal-number-of-clusters-in-k-means-elbow-and-silhouette-methods/. [Online; accessed 13-September-2023]. 2020 (cit. on p. 46).

[37] T. Nakaya and K. Yano. "Visualising crime clusters in a space-time cube: An exploratory data-analysis approach using space-time kernel density estimation and scan statistics". In: *Transactions in GIS* 14.3 (2010), pp. 223–239 (cit. on p. 11).

[38] S. Nascimento, S. Casca, and B. Mirkin. "A seed expanding cluster algorithm for deriving upwelling areas on sea surface temperature images". In: *Computers & Geosciences* 85 (2015), pp. 74–85 (cit. on pp. 2, 7, 39).

[39] S. Nascimento and N. Madaleno. "Unsupervised initialization of archetypal analysis and proportional membership fuzzy clustering". In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer. 2019, pp. 12–20 (cit. on p. 39).

[40] S. Nascimento, S. Mateen, and P. Relvas. "Sequential Self-tuning Clustering for Automatic Delimitation of Coastal Upwelling on SST Images". In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer. 2020, pp. 434–443 (cit. on pp. 2, 7, 39).

[41] S. Nascimento et al. "Automated computational delimitation of SST upwelling areas using fuzzy clustering". In: *Computers & Geosciences* 43 (2012), pp. 207–216 (cit. on p. 7).

[42] S. Nascimento et al. "Core-shell clustering approach for detection and analysis of coastal upwelling". In: *Computers & Geosciences* (2023), p. 105421 (cit. on pp. iv, vi, 7, 15, 54, 55, 69).

[43] S. Nascimento et al. "Novel Cluster Modeling for the Spatiotemporal Analysis of Coastal Upwelling". In: *EPIA Conference on Artificial Intelligence*. Springer. 2022, pp. 563–574 (cit. on pp. 3, 15, 37, 39, 40, 42, 43, 58, 69).

[44] S. Nascimento et al. "Piece-wise constant cluster modelling of dynamics of upwelling patterns". In: *Expert Systems* 40.10 (2023), e13446 (cit. on pp. 7, 15, 54, 55).

[45] NOAA. *What is upwelling?* https://oceanservice.noaa.gov/facts/upwelling.html. [Online; accessed 22-September-2023]. 2023 (cit. on p. 1).

[46] A. Nowicki, M. Janecki, and L. Dzierzbicka-Głowacka. "Operational system for automatic coastal upwelling detection in the Baltic Sea based on the 3D CEMBS model". In: *Journal of Operational Oceanography* 12.2 (2019), pp. 104–115 (cit. on pp. 2, 6).

[47] A. Palaude and T. Viéville. "DBA and K-means clustering: Explainability problems for research on behaviors and strategies". PhD thesis. Inria & Labri, Université Bordeaux, 2023 (cit. on pp. 28, 29).

[48] J. Paparrizos and L. Gravano. "k-shape: Efficient and accurate clustering of time series". In: *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 2015, pp. 1855–1870 (cit. on p. 44).

[49] F. Petitjean, A. Ketterlin, and P. Gançarski. "A global averaging method for dynamic time warping, with applications to clustering". In: *Pattern recognition* 44.3 (2011), pp. 678–693 (cit. on p. 28).

[50] J. Qian et al. "Cloud detection of optical remote sensing image time series using mean shift algorithm". In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE. 2016, pp. 560–562 (cit. on p. 32).

[51] J. D. Ramanantsoa et al. "Coastal upwelling south of Madagascar: Temporal and spatial variability". In: *Journal of Marine Systems* 178 (2018), pp. 29–37 (cit. on p. 6).

[52] V. Roth et al. "A resampling approach to cluster validation". In: *Compstat: Proceedings in Computational Statistics*. Springer. 2002, pp. 123–128 (cit. on p. 48).

[53] P. J. Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65 (cit. on p. 47).

[54] A. Rykov et al. "Inertia-based indices to determine the number of clusters in K-means: an experimental evaluation". In: *IEEE Access* (2024) (cit. on pp. 26, 44–47).

[55] A. Sardá-Espinosa. "Comparing time-series clustering algorithms in r using the dtwclust package". In: *R package vignette* 12 (2017), p. 41 (cit. on p. 29).

[56] W. Shi, Z. Huang, and J. Hu. "Using TPI to map spatial and temporal variations of significant coastal upwelling in the Northern South China Sea". In: *Remote Sensing* 13.6 (2021), p. 1065 (cit. on p. 6).

[57] S. Śmigiel, K. Pałczyński, and D. Ledziński. "Deep learning techniques in the classification of ECG signals using r-peak detection based on the PTB-XL dataset". In: *Sensors* 21.24 (2021), p. 8174 (cit. on pp. 18, 19).

[58] A. Tamim et al. "An efficient tool for automatic delimitation of Moroccan coastal upwelling using SST images". In: *IEEE Geoscience and Remote Sensing Letters* 12.4 (2014), pp. 875–879 (cit. on p. 5).

[59] A. Tamim et al. "Automatic detection of Moroccan coastal upwelling zones using sea surface temperature images". In: *International Journal of Remote Sensing* 40.7 (2019), pp. 2648–2666 (cit. on p. 5).

[60] R. Tavenard et al. "Tslearn, a machine learning toolkit for time series data". In: *Journal of machine learning research* 21.118 (2020), pp. 1–6 (cit. on pp. 26, 30).

[61] C. Truong, L. Oudre, and N. Vayatis. "Selective review of offline change point detection methods". In: *Signal Processing* 167 (2020), p. 107299 (cit. on p. 33).

[62] G. J. Van den Burg and C. K. Williams. "An evaluation of change point detection algorithms". In: *arXiv preprint arXiv:2003.06222* (2020) (cit. on p. 35).

[63] S. Wang, J. Cao, and P. Yu. "Deep learning for spatio-temporal data mining: A survey". In: *IEEE transactions on knowledge and data engineering* (2020) (cit. on pp. 8, 10).

[64] P. Wessel et al. “The generic mapping tools version 6”. In: *Geochemistry, Geophysics, Geosystems* 20.11 (2019), pp. 5556–5564 (cit. on p. 38).

[65] H. Xiong and Z. Li. “Clustering validation measures”. In: *Data Clustering*. Chapman and Hall/CRC, 2018, pp. 571–606 (cit. on pp. 25, 26).

[66] F. R. Zakani et al. “Kulczynski similarity index for objective evaluation of mesh segmentation algorithms”. In: *2016 5th International Conference on Multimedia Computing and Systems (ICMCS)*. IEEE. 2016, pp. 12–17 (cit. on p. 50).

A

Appendix 1

A.1 Time Ranges results

Year	IAP
2004	[1-8], [9-12], [13-16], [17-23]
2005	[1-4], [5-11], [12-17], [18-23]
2006	[1-4], [5-10], [11-23]
2007	[1-4], [5-11], [12-23]
2008	[1-8], [9-11], [12-15], [16-23]
2009	[1-7], [8-16], [17-19], [20-23]
2010	[1-10], [11-15], [16-23]
2011	[1-6], [7-13], [14-16], [17-23]
2012	[1-6], [7-11], [12-14], [15-23]
2013	[1-10], [11-16], [17-23]
2014	[1-7], [8-13], [14-23]
2015	[1-8], [9-14], [15-23]
2016	[1-8], [9-11], [12-14], [15-23]
2017	[1-8], [9-11], [12-23]
2018	[1-9], [10-12], [13-15], [16-23]
2019	[1-7], [8-12], [13-23]

Table A.1: North Morocco IAP time ranges results

Year	IAP
2004	[1-6], [7-14], [15-23]
2005	[1-5], [6-11], [12-16], [17-23]
2006	[1-8], [9-12], [13-16], [17-23]
2007	[1-9], [10-12], [13-23]
2008	[1-8], [9-13], [14-16], [17-23]
2009	[1-7], [8-15], [16-23]
2010	[1-9], [10-12], [13-15], [16-23]
2011	[1-7], [8-11], [12-14], [15-23]
2012	[1-8], [9-14], [15-23]
2013	[1-9], [10-15], [16-23]
2014	[1-9], [10-13], [14-16], [17-23]
2015	[1-9], [10-14], [15-23]
2016	[1-7], [8-10], [11-15], [16-23]
2017	[1-8], [9-15], [16-23]
2018	[1-9], [10-13], [14-23]
2019	[1-9], [10-16], [17-23]

Table A.2: South Morocco IAP time ranges results

Year	K-means SW	K-means EP1	K-means EP2	K-means CH	K-means WB	K-means XU
2004	[1-3], [4-8], [9-23]	[1-3], [4-9], [10-14], [15-18], [19-23]	[1-3], [4-8], [9-14], [15-23]	[1-3], [4-9], [10-14], [15-18], [19-23]	[1-3], [4-9], [10-14], [15-18], [19-23]	[1-3], [4-9], [10-14], [15-18], [19-23]
2005	[1-5], [6-10], [11-23]	[1-5], [6-14], [15-19], [20-23]	[1-5], [6-9], [10-14], [15-23]	[1-5], [6-14], [15-19], [20-23]	[1-5], [6-14], [15-19], [20-23]	[1-5], [6-14], [15-19], [20-23]
2006	[1-3], [4-10], [11-23]	[1-3], [4-10], [11-23]	[1-8], [9-13], [14-23]	[1-6], [7-10], [11-13], [14-20], [21-23]	[1-6], [7-10], [11-13], [14-20], [21-23]	[1-6], [7-10], [11-13], [14-20], [21-23]
2007	[1-9], [10-12], [13-23]	[1-9], [10-12], [13-23]	[1-5], [6-9], [10-12], [13-23]	[1-5], [6-9], [10-12], [13-15], [16-20], [21-23]	[1-5], [6-9], [10-12], [13-15], [16-20], [21-23]	[1-5], [6-9], [10-12], [13-15], [16-20], [21-23]

2008	[1-4], [5-7], [8-23]	[1-4], [5-7], [8-12], [13-23]	[1-4], [5-7], [8-12], [13-23]	[1-5], [6-12], [13-18], [19-23]	[1-5], [6-12], [13-18], [19-23]	[1-5], [6-12], [13-18], [19-23]
2009	[1-5], [6-12], [13-23]	[1-7], [8-12], [13-23]	[1-7], [8-12], [13-23]	[1-5], [6-9], [10-12], [13-16], [17-23]	[1-5], [6-9], [10-12], [13-16], [17-23]	[1-5], [6-9], [10-12], [13-16], [17-23]
2010	[1-6], [7-14], [15-23]	[1-4], [5-7], [8-12], [13-17], [18-23]	[1-4], [5-7], [8-15], [16-23]	[1-4], [5-7], [8-10], [11-13], [14-17], [18-23]	[1-4], [5-7], [8-10], [11-13], [14-17], [18-23]	[1-4], [5-7], [8-10], [11-13], [14-17], [18-23]
2011	[1-3], [4-8], [9-23]	[1-5], [6-8], [9-14], [15-19], [20-23]	[1-3], [4-7], [8-14], [15-23]	[1-5], [6-8], [9-14], [15-19], [20-23]	[1-5], [6-8], [9-14], [15-19], [20-23]	[1-5], [6-8], [9-14], [15-19], [20-23]
2012	[1-6], [7-10], [11-23]	[1-6], [7-10], [11-16], [17-23]	[1-6], [7-10], [11-16], [17-23]	[1-6], [7-10], [11-16], [17-23]	[1-6], [7-10], [11-16], [17-23]	[1-6], [7-10], [11-16], [17-23]
2013	[1-3], [4-9], [10-14], [15-23]	[1-3], [4-9], [10-14], [15-23]	[1-3], [4-9], [10-14], [15-23]	[1-3], [4-7], [8-11], [12-15], [16-20], [21-23]	[1-3], [4-7], [8-11], [12-15], [16-20], [21-23]	[1-3], [4-7], [8-11], [12-15], [16-20], [21-23]
2014	[1-3], [4-10], [11-14], [15-19], [20-23]	[1-3], [4-10], [11-23]	[1-3], [4-10], [11-23]	[1-3], [4-10], [11-13], [14-23]	[1-3], [4-10], [11-13], [14-23]	[1-3], [4-10], [11-13], [14-23]
2015	[1-3], [4-10], [11-23]	[1-3], [4-10], [11-23]	[1-3], [4-6], [7-12], [13-23]	[1-3], [4-6], [7-12], [13-19], [20-23]	[1-3], [4-6], [7-12], [13-19], [20-23]	[1-3], [4-6], [7-12], [13-19], [20-23]
2016	[1-5], [6-9], [10-23]	[1-5], [6-10], [11-16], [17-23]	[1-5], [6-10], [11-16], [17-23]	[1-5], [6-8], [9-16], [17-23]	[1-5], [6-8], [9-16], [17-23]	[1-5], [6-8], [9-16], [17-23]

2017	[1-5], [6-10], [11-23]	[1-5], [6-10], [11-23]	[1-5], [6-9], [10-15], [16-23]	[1-7], [8-10], [11-16], [17-23]	[1-7], [8-10], [11-16], [17-23]	[1-7], [8-10], [11-16], [17-23]
2018	[1-7], [8-13], [14-23]	[1-7], [8-13], [14-19], [20-23]	[1-7], [8-13], [14-19], [20-23]	[1-7], [8-11], [12-14], [15-19], [20-23]	[1-7], [8-11], [12-14], [15-19], [20-23]	[1-7], [8-11], [12-14], [15-19], [20-23]
2019	[1-4], [5-10], [11-23]	[1-6], [7-10], [11-15], [16-23]	[1-4], [5-9], [10-14], [15-23]	[1-6], [7-10], [11-15], [16-23]	[1-6], [7-10], [11-15], [16-23]	[1-6], [7-10], [11-15], [16-23]

Table A.3: Portugal K-means time ranges results

Year	K-means SW	K-means EP1	K-means EP2	K-means CH	K-means WB	K-means XU
2004	[1-10], [11-16], [17-23]	[1-8], [9-12], [13-16], [17-23]	[1-7], [8-12], [13-16], [17-23]	[1-7], [8-12], [13-16], [17-23]	[1-7], [8-12], [13-16], [17-23]	[1-7], [8-12], [13-16], [17-23]
2005	[1-4], [5-10], [11-13], [14-17], [18-23]	[1-4], [5-10], [11-13], [14-17], [18-23]	[1-4], [5-10], [11-13], [14-17], [18-23]	[1-4], [5-10], [11-13], [14-17], [18-23]	[1-4], [5-10], [11-13], [14-17], [18-23]	[1-4], [5-10], [11-13], [14-17], [18-23]
2006	[1-4], [5-8], [9-11], [12-23]	[1-4], [5-10], [11-14], [15-18], [19-23]	[1-4], [5-10], [11-15], [16-18], [19-23]	[1-4], [5-10], [11-14], [15-18], [19-23]	[1-4], [5-10], [11-14], [15-18], [19-23]	[1-4], [5-10], [11-14], [15-18], [19-23]
2007	[1-4], [5-11], [12-23]	[1-4], [5-11], [12-23]	[1-6], [7-11], [12-14], [15-23]	[1-4], [5-11], [12-14], [15-17], [18-23]	[1-4], [5-11], [12-14], [15-17], [18-23]	[1-4], [5-11], [12-14], [15-17], [18-23]
2008	[1-9], [10-15], [16-23]	[1-9], [10-14], [15-20], [21-23]	[1-8], [9-15], [16-23]	[1-9], [10-15], [16-23]	[1-7], [8-14], [15-20], [21-23]	[1-7], [8-14], [15-20], [21-23]
2009	[1-7], [8-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]
2010	[1-10], [11-15], [16-23]	[1-10], [11-15], [16-23]	[1-9], [10-15], [16-23]	[1-7], [8-10], [11-15], [16-20], [21-23]	[1-7], [8-10], [11-15], [16-20], [21-23]	[1-7], [8-10], [11-15], [16-20], [21-23]
2011	[1-7], [8-13], [14-16], [17-23]	[1-7], [8-13], [14-16], [17-23]	[1-7], [8-13], [14-16], [17-23]	[1-7], [8-13], [14-17], [18-23]	[1-7], [8-13], [14-17], [18-23]	[1-7], [8-13], [14-17], [18-23]

2012	[1-6], [7-14], [15-23]	[1-6], [7-11], [12-14], [15-19], [20-23]	[1-6], [7-14], [15-18], [19-23]	[1-6], [7-11], [12-14], [15-19], [20-23]	[1-6], [7-11], [12-14], [15-19], [20-23]	[1-6], [7-11], [12-14], [15-19], [20-23]
2013	[1-10], [11-16], [17-23]	[1-10], [11-16], [17-23]	[1-10], [11-17], [18-23]	[1-10], [11-16], [17-23]	[1-11], [12-16], [17-23]	[1-11], [12-16], [17-23]
2014	[1-7], [8-14], [15-23]	[1-7], [8-12], [13-18], [19-23]	[1-6], [7-10], [11-14], [15-23]	[1-7], [8-12], [13-18], [19-23]	[1-7], [8-12], [13-18], [19-23]	[1-7], [8-12], [13-18], [19-23]
2015	[1-8], [9-14], [15-23]	[1-8], [9-14], [15-23]	[1-8], [9-14], [15-23]	[1-7], [8-10], [11-14], [15-20], [21-23]	[1-7], [8-10], [11-14], [15-20], [21-23]	[1-7], [8-10], [11-14], [15-20], [21-23]
2016	[1-11], [12-14], [15-23]	[1-6], [7-14], [15-23]	[1-6], [7-14], [15-23]	[1-7], [8-14], [15-23]	[1-6], [7-12], [13-17], [18-23]	[1-6], [7-12], [13-17], [18-23]
2017	[1-6], [7-9], [10-23]	[1-6], [7-14], [15-23]	[1-6], [7-16], [17-23]	[1-3], [4-6], [7-12], [13-16], [17-23]	[1-3], [4-6], [7-12], [13-16], [17-23]	[1-3], [4-6], [7-12], [13-16], [17-23]
2018	[1-9], [10-15], [16-18], [19-23]	[1-4], [5-7], [8-11], [12-15], [16-23]	[1-6], [7-11], [12-17], [18-23]	[1-4], [5-7], [8-11], [12-15], [16-23]	[1-4], [5-7], [8-11], [12-15], [16-23]	[1-4], [5-7], [8-11], [12-15], [16-23]
2019	[1-3], [4-9], [10-23]	[1-7], [8-10], [11-15], [16-23]	[1-7], [8-13], [14-23]	[1-3], [4-9], [10-23]	[1-6], [7-9], [10-15], [16-23]	[1-6], [7-9], [10-15], [16-23]

Table A.4: North Morocco K-means time ranges results

Year	K-means SW	K-means EP1	K-means EP2	K-means CH	K-means WB	K-means XU
2004	[1-6], [7-12], [13-23]	[1-6], [7-12], [13-23]	[1-6], [7-12], [13-23], 16	[1-6], [7-10], [11-15], [16-23]	[1-6], [7-10], [11-15], [16-23]	[1-6], [7-10], [11-15], [16-23]
2005	[1-5], [6-15], [16-23]	[1-5], [6-11], [12-18], [19-23]	[1-5], [6-11], [12-18], [19-23]	[1-4], [5-8], [9-11], [12-15], [16-18], [19-23]	[1-4], [5-8], [9-11], [12-15], [16-18], [19-23]	[1-4], [5-8], [9-11], [12-15], [16-18], [19-23]
2006	[1-8], [9-11], [12-16], [17-23]	[1-11], [12-16], [17-23]	[1-8], [9-11], [12-16], [17-23]	[1-8], [9-11], [12-17], [18-23]	[1-8], [9-11], [12-17], [18-23]	[1-8], [9-11], [12-17], [18-23]
2007	[1-12], [13-19], [20-23]	[1-5], [6-12], [13-19], [20-23]	[1-5], [6-12], [13-19], [20-23]	[1-5], [6-12], [13-19], [20-23]	[1-5], [6-12], [13-19], [20-23]	[1-5], [6-12], [13-19], [20-23]
2008	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]
2009	[1-7], [8-15], [16-23]	[1-7], [8-15], [16-23]	[1-7], [8-12], [13-16], [17-23]	[1-7], [8-12], [13-16], [17-23]	[1-5], [6-13], [14-16], [17-23]	[1-5], [6-13], [14-16], [17-23]
2010	[1-4], [5-15], [16-23]	[1-4], [5-11], [12-16], [17-23]	[1-4], [5-11], [12-16], [17-23]	[1-4], [5-11], [12-16], [17-23]	[1-4], [5-11], [12-16], [17-23]	[1-4], [5-11], [12-16], [17-23]
2011	[1-7], [8-14], [15-23]	[1-7], [8-11], [12-15], [16-23]	[1-7], [8-11], [12-15], [16-23]	[1-7], [8-11], [12-15], [16-19], [20-23]	[1-7], [8-11], [12-15], [16-19], [20-23]	[1-7], [8-11], [12-15], [16-19], [20-23]

2012	[1-9], [10-15], [16-23]	[1-11], [12-15], [16-23]	[1-11], [12-15], [16-23]	[1-11], [12-15], [16-23]	[1-3], [4-11], [12-14], [15-19], [20-23]	[1-3], [4-11], [12-14], [15-19], [20-23]
2013	[1-9], [10-15], [16-23]	[1-10], [11-15], [16-19], [20-23]	[1-6], [7-11], [12-15], [16-23]	[1-10], [11-15], [16-19], [20-23]	[1-10], [11-15], [16-19], [20-23]	[1-10], [11-15], [16-19], [20-23]
2014	[1-9], [10-15], [16-23]	[1-9], [10-16], [17-23]	[1-9], [10-16], [17-23]	[1-11], [12-16], [17-20], [21-23]	[1-11], [12-16], [17-20], [21-23]	[1-11], [12-16], [17-20], [21-23]
2015	[1-9], [10-14], [15-23]	[1-8], [9-11], [12-14], [15-23]	[1-8], [9-11], [12-14], [15-23]	[1-9], [10-14], [15-23]	[1-7], [8-11], [12-14], [15-20], [21-23]	[1-7], [8-11], [12-14], [15-20], [21-23]
2016	[1-10], [11-15], [16-23]	[1-7], [8-10], [11-15], [16-23]	[1-7], [8-10], [11-15], [16-23]	[1-7], [8-10], [11-15], [16-23]	[1-4], [5-7], [8-15], [16-20], [21-23]	[1-4], [5-7], [8-15], [16-20], [21-23]
2017	[1-7], [8-14], [15-23]	[1-3], [4-7], [8-13], [14-17], [18-23]	[1-3], [4-7], [8-13], [14-17], [18-23]	[1-3], [4-7], [8-13], [14-17], [18-23]	[1-7], [8-10], [11-13], [14-17], [18-23]	[1-7], [8-10], [11-13], [14-17], [18-23]
2018	[1-7], [8-13], [14-23]	[1-3], [4-9], [10-13], [14-18], [19-23]	[1-7], [8-12], [13-18], [19-23]	[1-3], [4-9], [10-13], [14-18], [19-23]	[1-3], [4-9], [10-13], [14-18], [19-23]	[1-3], [4-9], [10-13], [14-18], [19-23]
2019	[1-11], [12-19], [20-23]	[1-9], [10-14], [15-19], [20-23]	[1-9], [10-14], [15-19], [20-23]	[1-5], [6-11], [12-16], [17-19], [20-23]	[1-5], [6-11], [12-16], [17-19], [20-23]	[1-5], [6-11], [12-16], [17-19], [20-23]

Table A.5: South Morocco K-means time ranges results

Year	K-means++ SW	K-means++ EP1	K-means++ EP2	K-means++ CH	K-means++ WB	K-means++ XU
2004	[1-3], [4-8], [9-23]	[1-3], [4-8], [9-14], [15-23]	[1-3], [4-8], [9-14], [15-23]	[1-3], [4-9], [10-14], [15-18], [19-23]	[1-3], [4-9], [10-14], [15-18], [19-23]	[1-3], [4-9], [10-14], [15-18], [19-23]
2005	[1-5], [6-10], [11-23]	[1-5], [6-10], [11-23]	[1-5], [6-9], [10-14], [15-23]	[1-5], [6-10], [11-14], [15-19], [20-23]	[1-5], [6-10], [11-14], [15-19], [20-23]	[1-5], [6-10], [11-14], [15-19], [20-23]
2006	[1-3], [4-10], [11-23]	[1-3], [4-10], [11-23]	[1-8], [9-13], [14-23]	[1-6], [7-10], [11-14], [15-20], [21-23]	[1-6], [7-10], [11-14], [15-20], [21-23]	[1-6], [7-10], [11-14], [15-20], [21-23]
2007	[1-9], [10-12], [13-23]	[1-9], [10-12], [13-23]	[1-5], [6-9], [10-12], [13-23]	[1-4], [5-9], [10-12], [13-15], [16-23]	[1-4], [5-9], [10-12], [13-15], [16-23]	[1-4], [5-9], [10-12], [13-15], [16-23]
2008	[1-5], [6-8], [9-23]	[1-4], [5-7], [8-13], [14-23]	[1-4], [5-7], [8-13], [14-23]	[1-11], [12-17], [18-23]	[1-11], [12-17], [18-23]	[1-11], [12-17], [18-23]
2009	[1-6], [7-12], [13-23]	[1-7], [8-12], [13-23]	[1-7], [8-12], [13-23]	[1-9], [10-13], [14-23]	[1-9], [10-13], [14-23]	[1-9], [10-13], [14-23]
2010	[1-6], [7-14], [15-23]	[1-4], [5-7], [8-15], [16-23]	[1-4], [5-7], [8-12], [13-17], [18-23]	[1-4], [5-7], [8-12], [13-17], [18-23]	[1-4], [5-7], [8-12], [13-17], [18-23]	[1-4], [5-7], [8-12], [13-17], [18-23]
2011	[1-3], [4-9], [10-23]	[1-7], [8-14], [15-23]	[1-7], [8-14], [15-23]	[1-7], [8-12], [13-17], [18-23]	[1-7], [8-12], [13-17], [18-23]	[1-7], [8-12], [13-17], [18-23]
2012	[1-6], [7-10], [11-23]	[1-6], [7-10], [11-16], [17-23]	[1-6], [7-10], [11-16], [17-23]	[1-6], [7-10], [11-16], [17-23]	[1-6], [7-10], [11-16], [17-23]	[1-6], [7-10], [11-16], [17-23]

2013	[1-3], [4-9], [10-14], [15-23]	[1-3], [4-9], [10-14], [15-23]	[1-3], [4-9], [10-14], [15-23]	[1-3], [4-8], [9-11], [12-14], [15-20], [21-23]	[1-3], [4-8], [9-11], [12-14], [15-20], [21-23]	[1-3], [4-8], [9-11], [12-14], [15-20], [21-23]
2014	[1-3], [4-10], [11-13], [14-19], [20-23]	[1-10], [11-20], [21-23]	[1-3], [4-10], [11-19], [20-23]	[1-3], [4-10], [11-13], [14-19], [20-23]	[1-3], [4-10], [11-13], [14-19], [20-23]	[1-3], [4-10], [11-13], [14-19], [20-23]
2015	[1-3], [4-10], [11-23]	[1-3], [4-10], [11-23]	[1-3], [4-6], [7-12], [13-23]	[1-3], [4-6], [7-12], [13-19], [20-23]	[1-3], [4-6], [7-12], [13-19], [20-23]	[1-3], [4-6], [7-12], [13-19], [20-23]
2016	[1-5], [6-9], [10-23]	[1-6], [7-9], [10-23], 15	[1-6], [7-9], [10-23], 15	[1-5], [6-10], [11-16], [17-23]	[1-5], [6-10], [11-16], [17-23]	[1-5], [6-10], [11-16], [17-23]
2017	[1-5], [6-10], [11-23]	[1-5], [6-10], [11-23]	[1-5], [6-8], [9-14], [15-23]	[1-5], [6-10], [11-16], [17-23]	[1-5], [6-10], [11-16], [17-23]	[1-5], [6-10], [11-16], [17-23]
2018	[1-7], [8-13], [14-19], [20-23]	[1-7], [8-13], [14-19], [20-23]	[1-7], [8-13], [14-19], [20-23]	[1-7], [8-12], [13-19], [20-23]	[1-7], [8-12], [13-19], [20-23]	[1-7], [8-12], [13-19], [20-23]
2019	[1-4], [5-10], [11-23]	[1-4], [5-10], [11-23]	[1-4], [5-10], [11-16], [17-23]	[1-4], [5-7], [8-10], [11-16], [17-23]	[1-4], [5-7], [8-10], [11-16], [17-23]	[1-4], [5-7], [8-10], [11-16], [17-23]

Table A.6: Portugal K-means++ time ranges results

Year	K-means++ SW	K-means++ EP1	K-means++ EP2	K-means++ CH	K-means++ WB	K-means++ XU
2004	[1-5], [6-10], [11-16], [17-23]	[1-5], [6-8], [9-12], [13-16], [17-23]	[1-5], [6-10], [11-16], [17-23]	[1-5], [6-8], [9-12], [13-16], [17-23]	[1-5], [6-8], [9-12], [13-16], [17-23]	[1-5], [6-8], [9-12], [13-16], [17-23]
2005	[1-4], [5-10], [11-13], [14-17], [18-23]	[1-4], [5-10], [11-13], [14-17], [18-23]	[1-4], [5-10], [11-13], [14-17], [18-23]	[1-4], [5-7], [8-10], [11-13], [14-17], [18-23]	[1-4], [5-7], [8-10], [11-13], [14-17], [18-23]	[1-4], [5-7], [8-10], [11-13], [14-17], [18-23]
2006	[1-4], [5-12], [13-23]	[1-4], [5-10], [11-15], [16-18], [19-23]	[1-4], [5-11], [12-15], [16-18], [19-23]	[1-4], [5-10], [11-15], [16-18], [19-23]	[1-4], [5-10], [11-15], [16-18], [19-23]	[1-4], [5-10], [11-15], [16-18], [19-23]
2007	[1-6], [7-11], [12-23]	[1-6], [7-11], [12-23]	[1-6], [7-11], [12-14], [15-23]	[1-4], [5-12], [13-17], [18-23]	[1-4], [5-12], [13-17], [18-23]	[1-4], [5-12], [13-17], [18-23]
2008	[1-8], [9-11], [12-23]	[1-9], [10-15], [16-20], [21-23]	[1-9], [10-15], [16-20], [21-23]	[1-8], [9-11], [12-23]	[1-9], [10-15], [16-20], [21-23]	[1-9], [10-15], [16-20], [21-23]
2009	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]
2010	[1-10], [11-15], [16-23]	[1-10], [11-15], [16-23]	[1-9], [10-15], [16-23]	[1-7], [8-10], [11-15], [16-20], [21-23]	[1-7], [8-10], [11-15], [16-20], [21-23]	[1-7], [8-10], [11-15], [16-20], [21-23]
2011	[1-7], [8-13], [14-16], [17-23]	[1-7], [8-13], [14-16], [17-23]	[1-7], [8-13], [14-16], [17-23]	[1-7], [8-17], [18-23]	[1-7], [8-17], [18-23]	[1-7], [8-17], [18-23]

2012	[1-6], [7-14], [15-23]	[1-6], [7-14], [15-23]	[1-6], [7-14], [15-18], [19-23]	[1-6], [7-11], [12-14], [15-19], [20-23]	[1-6], [7-11], [12-14], [15-19], [20-23]	[1-6], [7-11], [12-14], [15-19], [20-23]
2013	[1-10], [11-16], [17-23]	[1-10], [11-16], [17-23]	[1-9], [10-12], [13-23], 17	[1-9], [10-12], [13-17], [18-23]	[1-9], [10-12], [13-17], [18-23]	[1-9], [10-12], [13-17], [18-23]
2014	[1-7], [8-14], [15-23]	[1-7], [8-14], [15-23]	[1-7], [8-14], [15-23]	[1-7], [8-12], [13-19], [20-23]	[1-7], [8-12], [13-19], [20-23]	[1-7], [8-12], [13-19], [20-23]
2015	[1-8], [9-14], [15-23]	[1-8], [9-14], [15-23]	[1-8], [9-14], [15-23]	[1-7], [8-10], [11-14], [15-20], [21-23]	[1-7], [8-10], [11-14], [15-20], [21-23]	[1-7], [8-10], [11-14], [15-20], [21-23]
2016	[1-11], [12-14], [15-23]	[1-7], [8-11], [12-14], [15-23]	[1-7], [8-11], [12-14], [15-23]	[1-7], [8-12], [13-23]	[1-7], [8-12], [13-23]	[1-7], [8-12], [13-23]
2017	[1-5], [6-8], [9-23]	[1-5], [6-8], [9-16], [17-23]	[1-5], [6-8], [9-16], [17-23]	[1-6], [7-12], [13-17], [18-23]	[1-6], [7-12], [13-17], [18-23]	[1-6], [7-12], [13-17], [18-23]
2018	[1-9], [10-12], [13-23]	[1-9], [10-12], [13-23]	[1-6], [7-11], [12-15], [16-23]	[1-4], [5-7], [8-11], [12-17], [18-23]	[1-4], [5-7], [8-11], [12-17], [18-23]	[1-4], [5-7], [8-11], [12-17], [18-23]
2019	[1-6], [7-9], [10-23]	[1-6], [7-9], [10-23]	[1-6], [7-9], [10-15], [16-23]	[1-6], [7-9], [10-23]	[1-5], [6-9], [10-17], [18-23]	[1-5], [6-9], [10-17], [18-23]

Table A.7: North Morocco K-means++ time ranges results

Year	K-means++ SW	K-means++ EP1	K-means++ EP2	K-means++ CH	K-means++ WB	K-means++ XU
2004	[1-6], [7-12], [13-23]	[1-6], [7-12], [13-23]	[1-6], [7-12], [13-23], 16	[1-4], [5-10], [11-15], [16-23]	[1-4], [5-10], [11-15], [16-23]	[1-4], [5-10], [11-15], [16-23]
2005	[1-4], [5-11], [12-18], [19-23]	[1-4], [5-11], [12-18], [19-23]	[1-4], [5-11], [12-18], [19-23]	[1-4], [5-8], [9-11], [12-15], [16-18], [19-23]	[1-4], [5-8], [9-11], [12-15], [16-18], [19-23]	[1-4], [5-8], [9-11], [12-15], [16-18], [19-23]
2006	[1-11], [12-16], [17-23]	[1-11], [12-16], [17-23]	[1-8], [9-11], [12-16], [17-23]	[1-9], [10-12], [13-17], [18-23]	[1-9], [10-12], [13-17], [18-23]	[1-9], [10-12], [13-17], [18-23]
2007	[1-12], [13-19], [20-23]	[1-5], [6-12], [13-19], [20-23]	[1-5], [6-12], [13-19], [20-23]	[1-5], [6-9], [10-14], [15-19], [20-23]	[1-5], [6-9], [10-14], [15-19], [20-23]	[1-5], [6-9], [10-14], [15-19], [20-23]
2008	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]
2009	[1-7], [8-15], [16-23]	[1-7], [8-15], [16-23]	[1-7], [8-12], [13-16], [17-23]	[1-7], [8-12], [13-16], [17-23]	[1-7], [8-10], [11-13], [14-16], [17-23]	[1-7], [8-10], [11-13], [14-16], [17-23]
2010	[1-4], [5-15], [16-23]	[1-4], [5-11], [12-16], [17-23]	[1-4], [5-11], [12-16], [17-23]	[1-4], [5-9], [10-15], [16-23]	[1-4], [5-9], [10-15], [16-23]	[1-4], [5-9], [10-15], [16-23]
2011	[1-7], [8-14], [15-23]	[1-7], [8-14], [15-23]	[1-7], [8-11], [12-15], [16-23]	[1-7], [8-11], [12-14], [15-19], [20-23]	[1-7], [8-11], [12-14], [15-19], [20-23]	[1-7], [8-11], [12-14], [15-19], [20-23]

2012	[1-9], [10-15], [16-23]	[1-9], [10-15], [16-23]	[1-9], [10-15], [16-23]	[1-5], [6-9], [10-14], [15-19], [20-23]	[1-5], [6-9], [10-14], [15-19], [20-23]	[1-5], [6-9], [10-14], [15-19], [20-23]
2013	[1-9], [10-15], [16-23]	[1-9], [10-15], [16-23]	[1-6], [7-11], [12-15], [16-23]	[1-6], [7-11], [12-15], [16-19], [20-23]	[1-6], [7-11], [12-15], [16-19], [20-23]	[1-6], [7-11], [12-15], [16-19], [20-23]
2014	[1-9], [10-15], [16-23]	[1-9], [10-16], [17-23]	[1-9], [10-16], [17-23]	[1-11], [12-16], [17-20], [21-23]	[1-11], [12-16], [17-20], [21-23]	[1-11], [12-16], [17-20], [21-23]
2015	[1-9], [10-14], [15-23]	[1-8], [9-11], [12-14], [15-23]	[1-8], [9-11], [12-14], [15-23]	[1-9], [10-14], [15-23]	[1-4], [5-11], [12-14], [15-20], [21-23]	[1-4], [5-11], [12-14], [15-20], [21-23]
2016	[1-10], [11-15], [16-23]	[1-7], [8-10], [11-15], [16-23]	[1-7], [8-10], [11-15], [16-23]	[1-7], [8-10], [11-15], [16-20], [21-23]	[1-7], [8-10], [11-15], [16-20], [21-23]	[1-7], [8-10], [11-15], [16-20], [21-23]
2017	[1-8], [9-14], [15-23]	[1-3], [4-7], [8-13], [14-17], [18-23]	[1-3], [4-7], [8-13], [14-17], [18-23]	[1-3], [4-7], [8-13], [14-17], [18-23]	[1-3], [4-7], [8-13], [14-17], [18-23]	[1-3], [4-7], [8-13], [14-17], [18-23]
2018	[1-7], [8-13], [14-23]	[1-9], [10-13], [14-18], [19-23]	[1-9], [10-13], [14-18], [19-23]	[1-3], [4-9], [10-13], [14-18], [19-23]	[1-3], [4-9], [10-13], [14-18], [19-23]	[1-3], [4-9], [10-13], [14-18], [19-23]
2019	[1-9], [10-14], [15-19], [20-23]	[1-9], [10-14], [15-19], [20-23]	[1-9], [10-14], [15-19], [20-23]	[1-5], [6-11], [12-14], [15-19], [20-23]	[1-5], [6-11], [12-14], [15-19], [20-23]	[1-5], [6-11], [12-14], [15-19], [20-23]

Table A.8: South Morocco K-means++ time ranges results

Year	DBA SW	DBA EP1	DBA EP2	DBA CH	DBA WB	DBA XU
2004	[1-3], [4-9], [10-23]	[1-3], [4-9], [10-14], [15-18], [19-23]	[1-3], [4-8], [9-23], 18	[1-3], [4-9], [10-14], [15-18], [19-23]	[1-3], [4-9], [10-14], [15-18], [19-23]	[1-3], [4-9], [10-14], [15-18], [19-23]
2005	[1-5], [6-12], [13-23]	[1-5], [6-12], [13-23]	[1-5], [6-9], [10-16], [17-23]	[1-5], [6-9], [10-14], [15-19], [20-23]	[1-5], [6-9], [10-14], [15-19], [20-23]	[1-5], [6-9], [10-14], [15-19], [20-23]
2006	[1-3], [4-10], [11-23]	[1-3], [4-10], [11-23]	[1-8], [9-13], [14-23]	[1-3], [4-8], [9-13], [14-20], [21-23]	[1-3], [4-8], [9-13], [14-20], [21-23]	[1-3], [4-8], [9-13], [14-20], [21-23]
2007	[1-9], [10-12], [13-23]	[1-9], [10-12], [13-23]	[1-4], [5-9], [10-12], [13-23]	[1-4], [5-9], [10-12], [13-15], [16-23], 21	[1-4], [5-9], [10-12], [13-15], [16-23], 21	[1-4], [5-9], [10-12], [13-15], [16-23], 21
2008	[1-5], [6-9], [10-23]	[1-5], [6-13], [14-23]	[1-5], [6-13], [14-23]	[1-7], [8-12], [13-18], [19-23]	[1-7], [8-12], [13-18], [19-23]	[1-7], [8-12], [13-18], [19-23]
2009	[1-4], [5-11], [12-23]	[1-7], [8-12], [13-23]	[1-7], [8-12], [13-23]	[1-7], [8-10], [11-14], [15-23]	[1-7], [8-10], [11-14], [15-23]	[1-7], [8-10], [11-14], [15-23]
2010	[1-4], [5-8], [9-23]	[1-4], [5-7], [8-12], [13-17], [18-23]	[1-4], [5-7], [8-12], [13-17], [18-23]	[1-4], [5-7], [8-12], [13-17], [18-23]	[1-4], [5-7], [8-12], [13-17], [18-23]	[1-4], [5-7], [8-12], [13-17], [18-23]
2011	[1-5], [6-12], [13-23]	[1-5], [6-8], [9-14], [15-23]	[1-5], [6-8], [9-14], [15-23]	[1-5], [6-8], [9-14], [15-23]	[1-7], [8-10], [11-15], [16-23]	[1-7], [8-10], [11-15], [16-23]
2012	[1-6], [7-10], [11-23]	[1-6], [7-10], [11-15], [16-23]	[1-6], [7-10], [11-15], [16-23]	[1-6], [7-10], [11-15], [16-23]	[1-6], [7-10], [11-15], [16-23]	[1-6], [7-10], [11-15], [16-23]

2013	[1-3], [4-14], [15-23]	[1-3], [4-11], [12-14], [15-23]	[1-3], [4-11], [12-14], [15-23]	[1-3], [4-8], [9-11], [12-14], [15-20], [21-23]	[1-3], [4-8], [9-11], [12-14], [15-20], [21-23]	[1-3], [4-8], [9-11], [12-14], [15-20], [21-23]
2014	[1-3], [4-10], [11-23]	[1-3], [4-10], [11-23]	[1-3], [4-10], [11-23]	[1-3], [4-7], [8-12], [13-19], [20-23]	[1-3], [4-7], [8-12], [13-19], [20-23]	[1-3], [4-7], [8-12], [13-19], [20-23]
2015	[1-3], [4-10], [11-23]	[1-3], [4-10], [11-23]	[1-3], [4-6], [7-12], [13-23]	[1-3], [4-6], [7-12], [13-19], [20-23]	[1-3], [4-6], [7-12], [13-19], [20-23]	[1-3], [4-6], [7-12], [13-19], [20-23]
2016	[1-5], [6-9], [10-23]	[1-5], [6-10], [11-16], [17-23]	[1-4], [5-9], [10-15], [16-23]	[1-5], [6-10], [11-16], [17-23]	[1-5], [6-10], [11-16], [17-23]	[1-5], [6-10], [11-16], [17-23]
2017	[1-5], [6-10], [11-23]	[1-5], [6-10], [11-23]	[1-5], [6-9], [10-15], [16-23]	[1-5], [6-10], [11-14], [15-18], [19-23]	[1-5], [6-10], [11-14], [15-18], [19-23]	[1-5], [6-10], [11-14], [15-18], [19-23]
2018	[1-3], [4-12], [13-23]	[1-7], [8-13], [14-19], [20-23]	[1-7], [8-13], [14-19], [20-23]	[1-7], [8-12], [13-19], [20-23]	[1-7], [8-12], [13-19], [20-23]	[1-7], [8-12], [13-19], [20-23]
2019	[1-4], [5-11], [12-23]	[1-4], [5-11], [12-23]	[1-4], [5-10], [11-16], [17-23]	[1-6], [7-10], [11-16], [17-23]	[1-6], [7-10], [11-16], [17-23]	[1-6], [7-10], [11-16], [17-23]

Table A.9: Portugal DBA time ranges results

Year	DBA SW	DBA EP1	DBA EP2	DBA CH	DBA WB	DBA XU
2004	[1-10], [11-16], [17-23]	[1-5], [6-10], [11-16], [17-23]	[1-5], [6-10], [11-16], [17-23]	[1-4], [5-8], [9-12], [13-16], [17-23]	[1-4], [5-8], [9-12], [13-16], [17-23]	[1-4], [5-8], [9-12], [13-16], [17-23]
2005	[1-4], [5-10], [11-13], [14-17], [18-23]	[1-4], [5-7], [8-10], [11-13], [14-17], [18-23]	[1-4], [5-10], [11-13], [14-17], [18-23]	[1-4], [5-7], [8-10], [11-13], [14-17], [18-23]	[1-4], [5-7], [8-10], [11-13], [14-17], [18-23]	[1-4], [5-7], [8-10], [11-13], [14-17], [18-23]
2006	[1-4], [5-11], [12-23]	[1-4], [5-10], [11-15], [16-18], [19-23]	[1-4], [5-10], [11-15], [16-18], [19-23]	[1-4], [5-10], [11-15], [16-18], [19-23]	[1-4], [5-10], [11-15], [16-18], [19-23]	[1-4], [5-10], [11-15], [16-18], [19-23]
2007	[1-4], [5-11], [12-23]	[1-4], [5-11], [12-23]	[1-4], [5-11], [12-17], [18-23]	[1-4], [5-12], [13-17], [18-23]	[1-4], [5-12], [13-17], [18-23]	[1-4], [5-12], [13-17], [18-23]
2008	[1-8], [9-11], [12-23]	[1-9], [10-15], [16-20], [21-23]	[1-9], [10-16], [17-23]	[1-8], [9-11], [12-23]	[1-9], [10-15], [16-20], [21-23]	[1-9], [10-15], [16-20], [21-23]
2009	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-6], [7-9], [10-16], [17-19], [20-23]
2010	[1-10], [11-15], [16-23]	[1-10], [11-15], [16-23]	[1-9], [10-15], [16-23]	[1-7], [8-10], [11-15], [16-23]	[1-7], [8-10], [11-15], [16-23]	[1-7], [8-10], [11-15], [16-23]
2011	[1-7], [8-13], [14-16], [17-23]	[1-7], [8-13], [14-16], [17-23]	[1-7], [8-13], [14-16], [17-23]	[1-7], [8-13], [14-16], [17-23]	[1-7], [8-13], [14-16], [17-23]	[1-7], [8-13], [14-16], [17-23]

2012	[1-6], [7-14], [15-23]	[1-6], [7-14], [15-23]	[1-6], [7-14], [15-19, [20-23]	[1-6], [7-11], [12-14], [15-19], [20-23]	[1-6], [7-11], [12-14], [15-19], [20-23]	[1-6], [7-11], [12-14], [15-19], [20-23]
2013	[1-11], [12-16], [17-23]	[1-11], [12-16], [17-23]	[1-8], [9-12], [13-16], [17-23]	[1-8], [9-12], [13-16], [17-23]	[1-8], [9-12], [13-16], [17-23]	[1-8], [9-12], [13-16], [17-23]
2014	[1-7], [8-13], [14-23]	[1-7], [8-12], [13-19], [20-23]	[1-7], [8-13], [14-23]	[1-7], [8-12], [13-19], [20-23]	[1-7], [8-12], [13-19], [20-23]	[1-7], [8-12], [13-19], [20-23]
2015	[1-8], [9-14], [15-23]	[1-8], [9-14], [15-23]	[1-8], [9-14], [15-23]	[1-7], [8-10], [11-14], [15-23]	[1-7], [8-10], [11-14], [15-23]	[1-7], [8-10], [11-14], [15-23]
2016	[1-11], [12-14], [15-23]	[1-7], [8-14], [15-23]	[1-7], [8-14], [15-23]	[1-6], [7-11], [12-23]	[1-6], [7-11], [12-23]	[1-6], [7-11], [12-23]
2017	[1-6], [7-9], [10-23]	[1-8], [9-15], [16-23]	[1-8], [9-15], [16-23]	[1-6], [7-12], [13-17], [18-23]	[1-6], [7-12], [13-17], [18-23]	[1-6], [7-12], [13-17], [18-23]
2018	[1-9], [10-12], [13-23]	[1-9], [10-12], [13-23]	[1-6], [7-11], [12-15], [16-23]	[1-4], [5-7], [8-11], [12-15], [16-23]	[1-4], [5-7], [8-11], [12-15], [16-23]	[1-4], [5-7], [8-11], [12-15], [16-23]
2019	[1-9], [10-13], [14-23]	[1-9], [10-13], [14-23]	[1-9], [10-13], [14-23]	[1-9], [10-13], [14-23]	[1-5], [6-9], [10-14], [15-23]	[1-5], [6-9], [10-14], [15-23]

Table A.10: North Morocco DBA time ranges results

Year	DBA SW	DBA EP1	DBA EP2	DBA CH	DBA WB	DBA XU
2004	[1-6], [7-12], [13-23]	[1-6], [7-12], [13-23]	[1-5], [6-10], [11-15], [16-23]	[1-4], [5-12], [13-15], [16-23]	[1-4], [5-12], [13-15], [16-23]	[1-4], [5-12], [13-15], [16-23]
2005	[1-4], [5-11], [12-18], [19-23]	[1-4], [5-11], [12-18], [19-23]	[1-4], [5-11], [12-18], [19-23]	[1-4], [5-10], [11-14], [15-18], [19-23]	[1-4], [5-10], [11-14], [15-18], [19-23]	[1-4], [5-10], [11-14], [15-18], [19-23]
2006	[1-11], [12-17], [18-23]	[1-11], [12-17], [18-23]	[1-8], [9-12], [13-17], [18-23]	[1-8], [9-12], [13-18], [19-23]	[1-8], [9-12], [13-18], [19-23]	[1-8], [9-12], [13-18], [19-23]
2007	[1-5], [6-12], [13-19], [20-23]	[1-5], [6-12], [13-19], [20-23]	[1-5], [6-12], [13-19], [20-23]	[1-5], [6-9], [10-14], [15-19], [20-23]	[1-5], [6-9], [10-14], [15-19], [20-23]	[1-5], [6-9], [10-14], [15-19], [20-23]
2008	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]
2009	[1-7], [8-15], [16-23]	[1-7], [8-15], [16-23]	[1-7], [8-12], [13-16], [17-23]	[1-7], [8-12], [13-16], [17-23]	[1-7], [8-10], [11-13], [14-16], [17-23]	[1-7], [8-10], [11-13], [14-16], [17-23]
2010	[1-10], [11-16], [17-23]	[1-4], [5-11], [12-16], [17-23]	[1-4], [5-11], [12-16], [17-23]	[1-4], [5-9], [10-13], [14-16], [17-23]	[1-4], [5-9], [10-13], [14-16], [17-23]	[1-4], [5-9], [10-13], [14-16], [17-23]
2011	[1-7], [8-14], [15-23]	[1-7], [8-14], [15-23]	[1-7], [8-11], [12-15], [16-23]	[1-7], [8-11], [12-15], [16-19], [20-23]	[1-7], [8-11], [12-15], [16-19], [20-23]	[1-7], [8-11], [12-15], [16-19], [20-23]

2012	[1-6], [7-12], [13-23]	[1-11], [12-15], [16-23]	[1-11], [12-15], [16-23]	[1-6], [7-11], [12-14], [15-19], [20-23]	[1-6], [7-11], [12-14], [15-19], [20-23]	[1-6], [7-11], [12-14], [15-19], [20-23]
2013	[1-9], [10-15], [16-23]	[1-6], [7-11], [12-15], [16-23]	[1-6], [7-11], [12-15], [16-23]	[1-6], [7-11], [12-15], [16-23]	[1-8], [9-11], [12-15], [16-19], [20-23]	[1-8], [9-11], [12-15], [16-19], [20-23]
2014	[1-9], [10-16], [17-23]	[1-9], [10-16], [17-23]	[1-9], [10-16], [17-23]	[1-11], [12-16], [17-20], [21-23]	[1-11], [12-16], [17-20], [21-23]	[1-11], [12-16], [17-20], [21-23]
2015	[1-8], [9-14], [15-23]	[1-5], [6-11], [12-14], [15-23]	[1-5], [6-11], [12-14], [15-23]	[1-8], [9-14], [15-23]	[1-4], [5-11], [12-14], [15-17], [18-23]	[1-4], [5-11], [12-14], [15-17], [18-23]
2016	[1-8], [9-15], [16-23]	[1-7], [8-10], [11-15], [16-23]	[1-7], [8-10], [11-15], [16-23]	[1-7], [8-10], [11-14], [15-23]	[1-7], [8-10], [11-14], [15-23]	[1-7], [8-10], [11-14], [15-23]
2017	[1-8], [9-14], [15-23]	[1-3], [4-7], [8-13], [14-17], [18-23]	[1-3], [4-7], [8-13], [14-17], [18-23]	[1-3], [4-7], [8-13], [14-17], [18-23]	[1-3], [4-7], [8-11], [12-14], [15-17], [18-23]	[1-3], [4-7], [8-11], [12-14], [15-17], [18-23]
2018	[1-8], [9-13], [14-23]	[1-8], [9-13], [14-23]	[1-3], [4-9], [10-13], [14-18], [19-23]	[1-3], [4-9], [10-13], [14-18], [19-23]	[1-3], [4-9], [10-13], [14-18], [19-23]	[1-3], [4-9], [10-13], [14-18], [19-23]
2019	[1-11], [12-19], [20-23]	[1-9], [10-14], [15-19], [20-23]	[1-9], [10-14], [15-19], [20-23]	[1-5], [6-11], [12-15], [16-19], [20-23]	[1-5], [6-11], [12-15], [16-19], [20-23]	[1-5], [6-11], [12-15], [16-19], [20-23]

Table A.11: South Morocco DBA time ranges results

Year	Mean-Shift Flat	Mean-Shift Gaussian
2004	[1-3], [4-9], [10-14], [15-23]	[1-3], [4-8], [9-23]
2005	[1-5], [6-8], [9-14], [15-23]	[1-5], [6-9], [10-15], [16-23]
2006	[1-3], [4-8], [9-13], [14-20], [21-23]	[1-4], [5-9], [10-13], [14-20], [21-23]
2007	[1-6], [7-9], [10-12], [13-23]	[1-5], [6-9], [10-12], [13-23]
2008	[1-5], [6-8], [9-23]	[1-7], [8-12], [13-23]
2009	[1-7], [8-13], [14-23]	[1-6], [7-9], [10-13], [14-23]
2010	[1-4], [5-7], [8-15], [16-23]	[1-4], [5-8], [9-14], [15-23]
2011	[1-5], [6-9], [10-23]	[1-5], [6-9], [10-23]
2012	[1-6], [7-10], [11-23]	[1-6], [7-10], [11-23]
2013	[1-3], [4-10], [11-14], [15-23]	[1-3], [4-11], [12-14], [15-23]
2014	[1-3], [4-11], [12-19], [20-23]	[1-3], [4-11], [12-19], [20-23]
2015	[1-3], [4-8], [9-23]	[1-3], [4-8], [9-23]
2016	[1-5], [6-10], [11-23]	[1-7], [8-12], [13-23]
2017	[1-5], [6-12], [13-23]	[1-5], [6-13], [14-23]
2018	[1-7], [8-13], [14-19], [20-23]	[1-7], [8-13], [14-23]
2019	[1-4], [5-12], [13-23]	[1-4], [5-8], [9-13], [14-23]

Table A.12: Portugal Mean-Shift time ranges results

Year	Mean-Shift Flat	Mean-Shift Gaussian
2004	[1-5], [6-10], [11-16], [17-23]	[1-7], [8-11], [12-16], [17-23]
2005	[1-4], [5-10], [11-13], [14-17], [18-23]	[[1-4], [5-10], [11-13], [14-17], [18-23]
2006	[1-4], [5-11], [12-15], [16-18], [19-23]	[1-4], [5-11], [12-15], [16-18], [19-23]
2007	[1-6], [7-11], [12-23]	[1-4], [5-11], [12-23]
2008	[1-9], [10-15], [16-20], [21-23]	[1-9], [10-20], [21-23]
2009	[1-6], [7-9], [10-16], [17-19], [20-23]	[1-5], [6-9], [10-16], [17-19], [20-23]
2010	[1-9], [10-12], [13-15], [16-23]	[1-7], [8-11], [12-15], [16-23]
2011	[1-7], [8-13], [14-16], [17-23]	[1-7], [8-13], [14-16], [17-23]
2012	[1-6], [7-14], [15-19], [20-23]	[1-6], [7-14], [15-19], [20-23]
2013	[1-9], [10-12], [13-16], [17-23]	[1-8], [9-12], [13-16], [17-23]
2014	[1-7], [8-13], [14-23]	[1-7], [8-15], [16-23]
2015	[1-7], [8-10], [11-14], [15-23]	[1-7], [8-10], [11-14], [15-23]
2016	[1-6], [7-14], [15-23]	[1-5], [6-11], [12-14], [15-23]
2017	[1-8], [9-12], [13-17], [18-23]	[1-8], [9-15], [16-23]
2018	[1-7], [8-11], [12-23]	[1-7], [8-11], [12-23]
2019	[1-5], [6-9], [10-23]	[1-8], [9-13], [14-23]

Table A.13: North Morocco Mean-Shift time ranges results

Year	MS Ranges flat	MS Ranges gaussian
2004	[1-5], [6-12], [13-23]	[1-5], [6-12], [13-23]
2005	[1-8],[9-18],[19-23]	[1-4], [5-8], [9-17], [18-23]
2006	[1-11], [12-16], [17-23]	[1-8], [9-11], [12-17], [18-23]
2007	[1-11], [12-19], [20-23]	[1-12], [13-19], [20-23]
2008	[1-4], [5-10], [11-13], [14-16], [17-23]	[1-4], [5-10], [11-13], [14-16], [17-23]
2009	[1-7], [8-15], [16-23]	[1-7], [8-11], [12-15], [16-23]
2010	[1-4], [5-15], [16-23]	[1-11], [12-16], [17-23]
2011	[1-7], [8-13], [14-23]	[1-5], [6-12], [13-23]
2012	[1-9], [10-12], [13-23]	[1-9], [10-12], [13-23]
2013	[1-9], [10-15], [16-23]	[1-8], [7-11], [12-15], [16-23]
2014	[1-10], [11-16], [17-23]	[1-9], [10-16], [17-23]
2015	[1-11], [12-14], [15-23]	[1-11], [12-14], [15-23]
2016	[1-10], [11-15], [16-23]	[1-10], [11-15], [11-23]
2017	[1-10], [11-16], [17-23]	[1-8], [9-12], [13-16], [17-23]
2018	[1-7], [8-13], [14-23]	[1-7], [8-13], [14-23]
2019	[1-12], [13-19], [20-23]	[1-12], [13-19], [20-23]

Table A.14: South Morocco Mean-Shift time ranges results

B

Appendix 2

B.1 IAP Contribution and Cardinality Plots

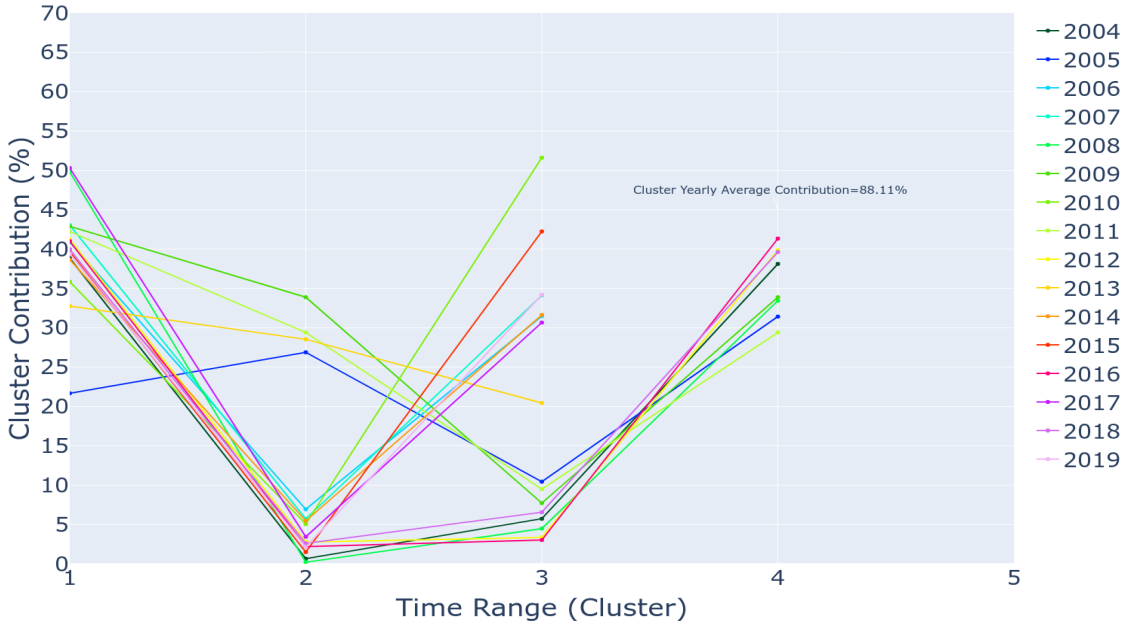


Figure B.1: Cluster Contribution of the clusters in the region of northern Morocco

Figure B.2: Cluster Cardinality of the clusters in the region of northern Morocco

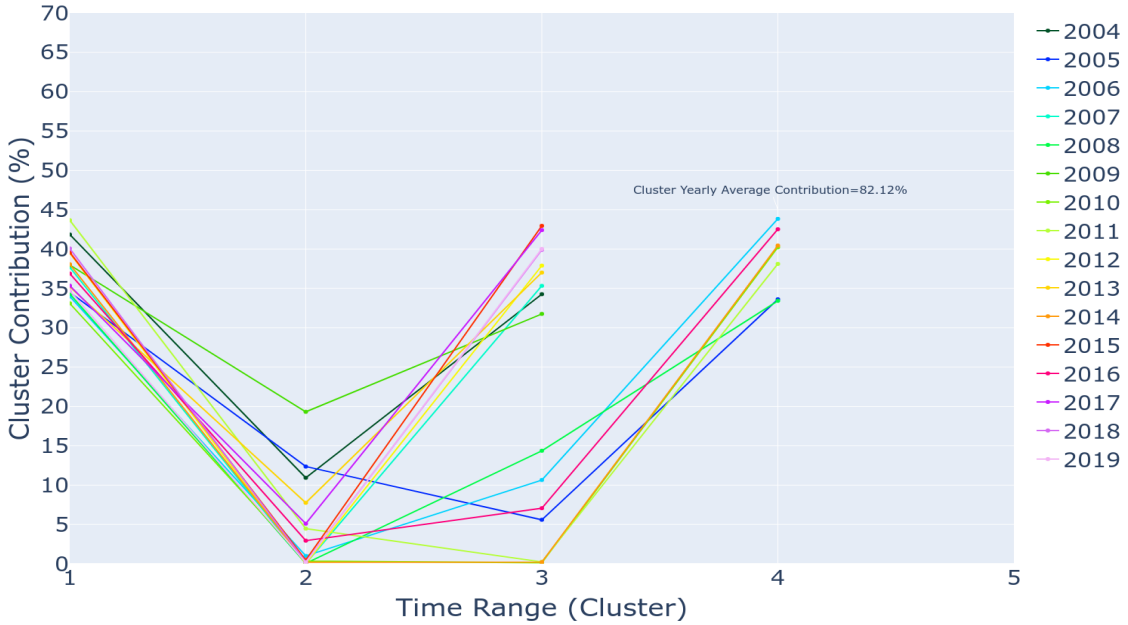


Figure B.3: Cluster Contribution of the clusters in the region of southern Morocco

Figure B.4: Cluster Cardinality of the clusters in the region of southern Morocco

B.2 Mean-Shift Plots

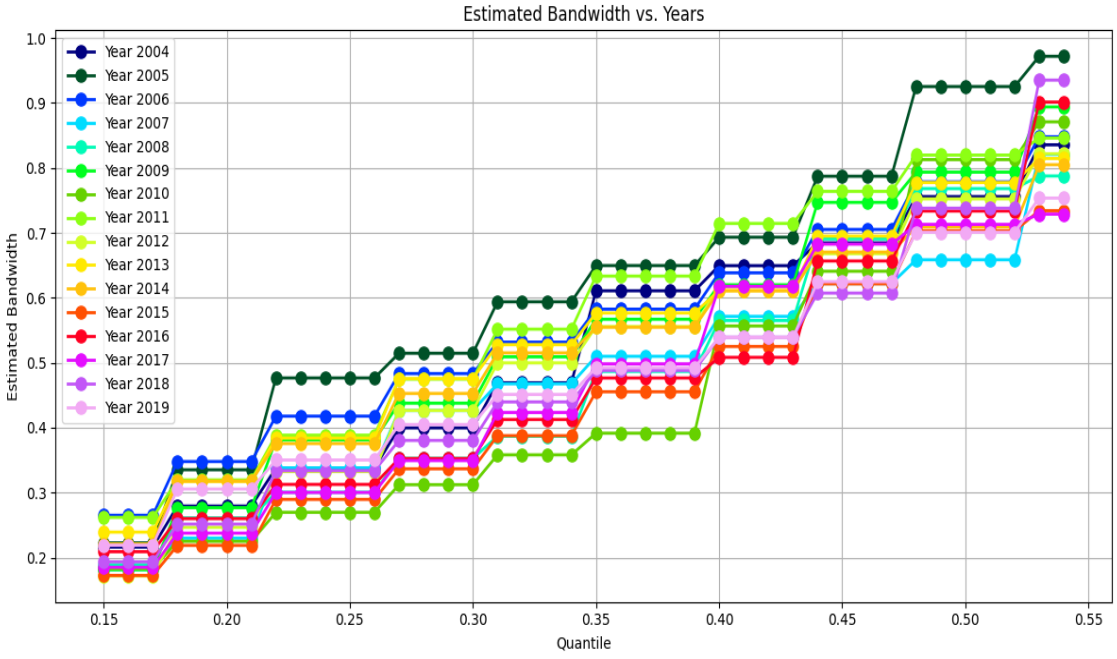
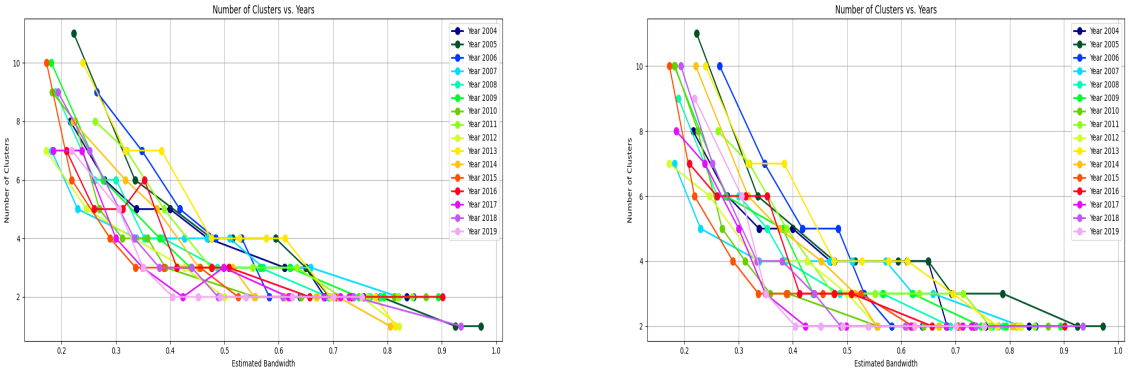


Figure B.5: Estimated bandwidth for different quantile values across the years



(a) Gaussian kernel (b) Flat kernel

Figure B.6: Bandwidth analysis for the region of North Morocco

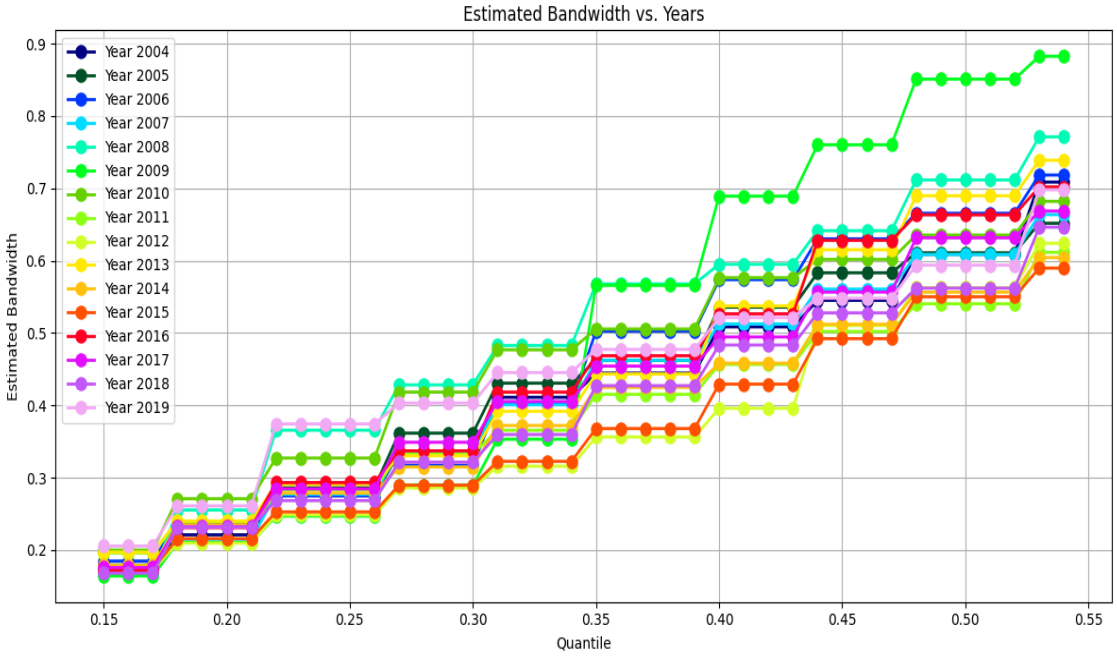
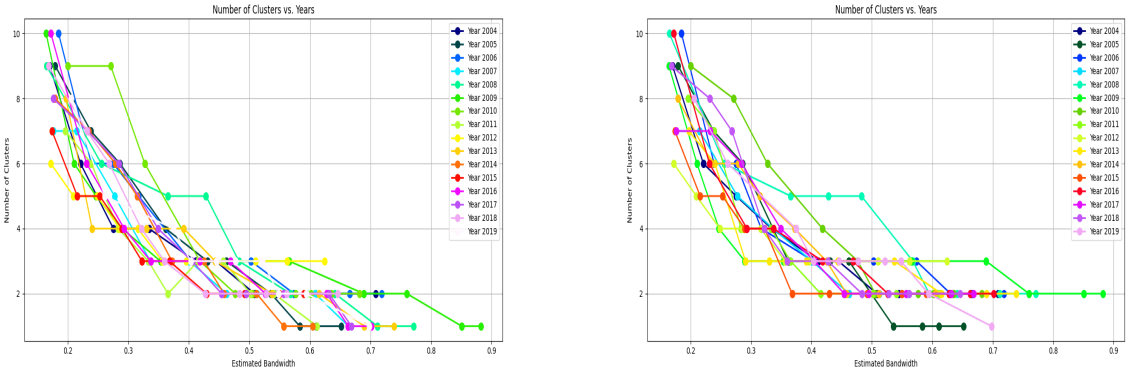


Figure B.7: Estimated bandwidth for different quantile values across the years



(a) Gaussian kernel (b) Flat kernel

Figure B.8: Bandwidth analysis for the region of South Morocco

B.3 Analysis of Core-Shell features

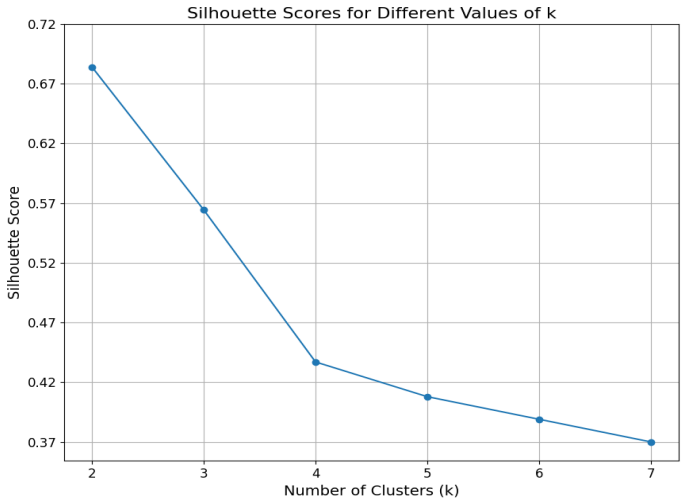


Figure B.9: Silhouette scores for different values of K using DBA with the Core temperature for the region of North Morocco

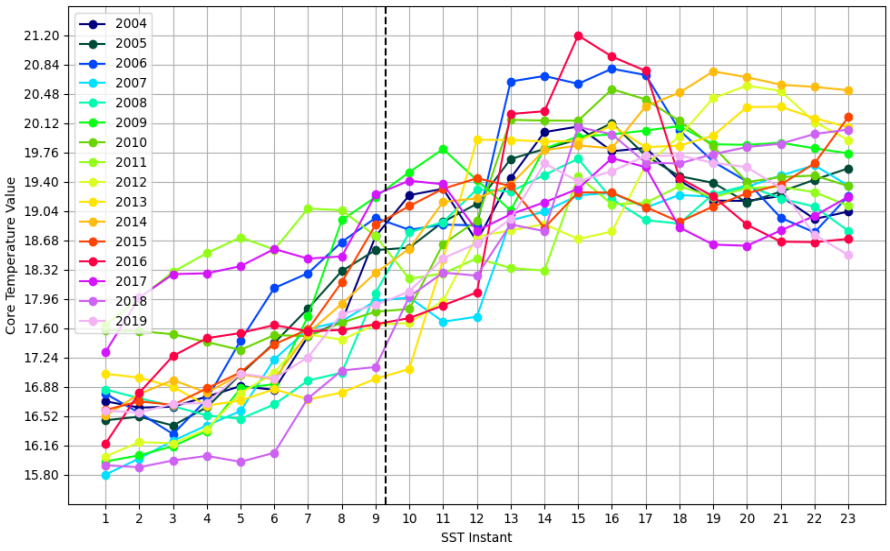


Figure B.10: Core temperature values across the 23 SST instants with each of the 16 years represented and "border" separating the different clusters for the region of North Morocco

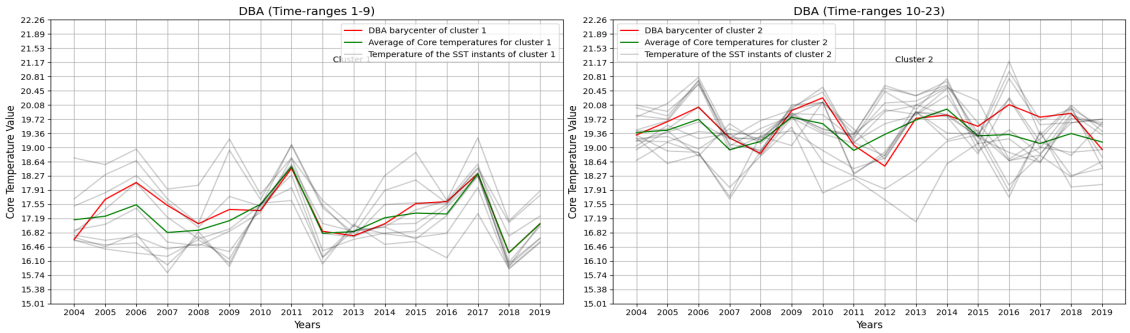


Figure B.11: Core temperature values across 16 years divided by each cluster obtained by the DBA algorithm, with the SST instant temperatures of each year, the value of the DBA barycenter and the average of the core temperatures for the region of North Morocco

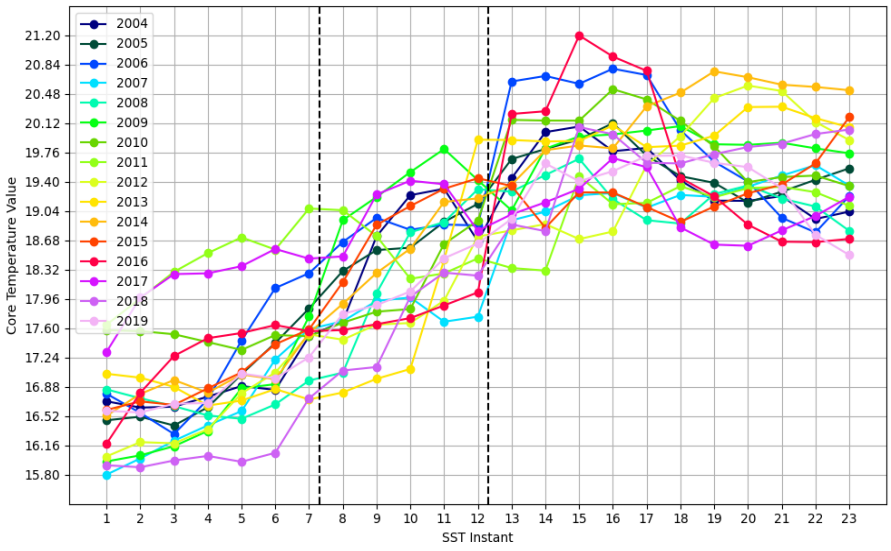


Figure B.12: Core temperature values across the 23 SST instants with each of the 16 years represented and "border" separating the different clusters for the region of North Morocco for K = 3

Figure B.13: Silhouette scores for different values of K using DBA with the Core temperature for the region of South Morocco

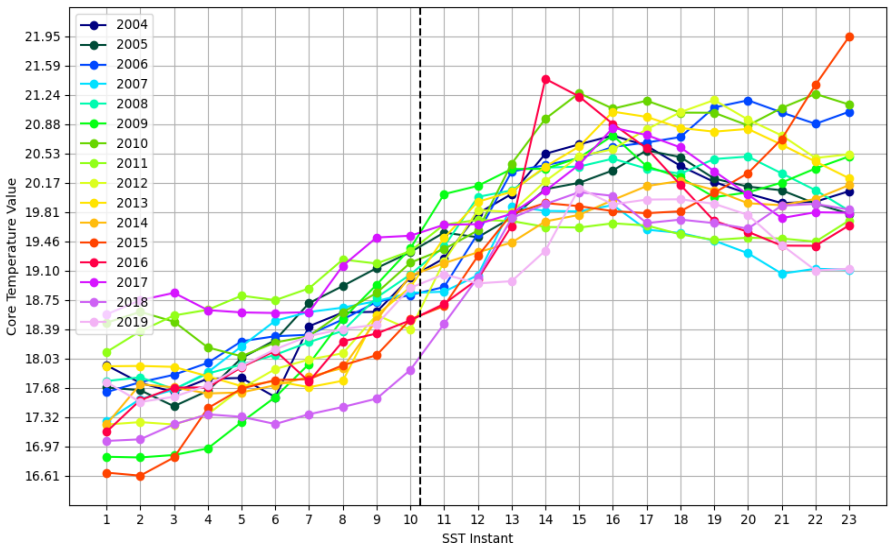


Figure B.14: Core temperature values across the 23 SST instants with each of the 16 years represented and "border" separating the different clusters for the region of South Morocco

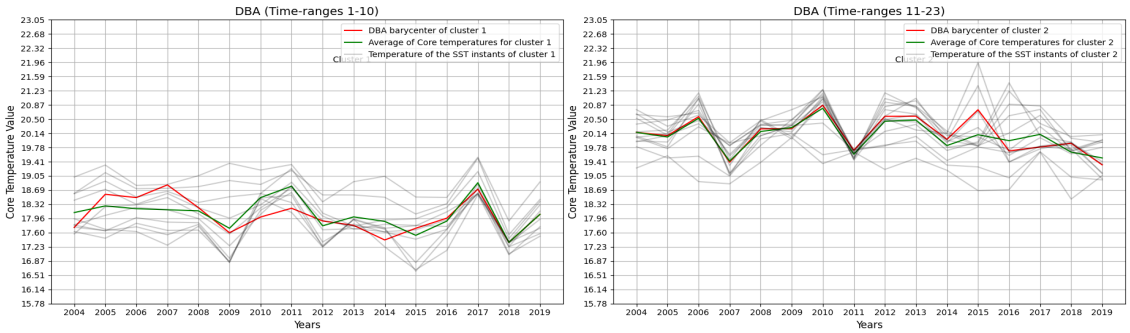


Figure B.15: Core temperature values across 16 years divided by each cluster obtained by the DBA algorithm, with the SST instant temperatures of each year, the value of the DBA barycenter and the average of the core temperatures for the region of South Morocco

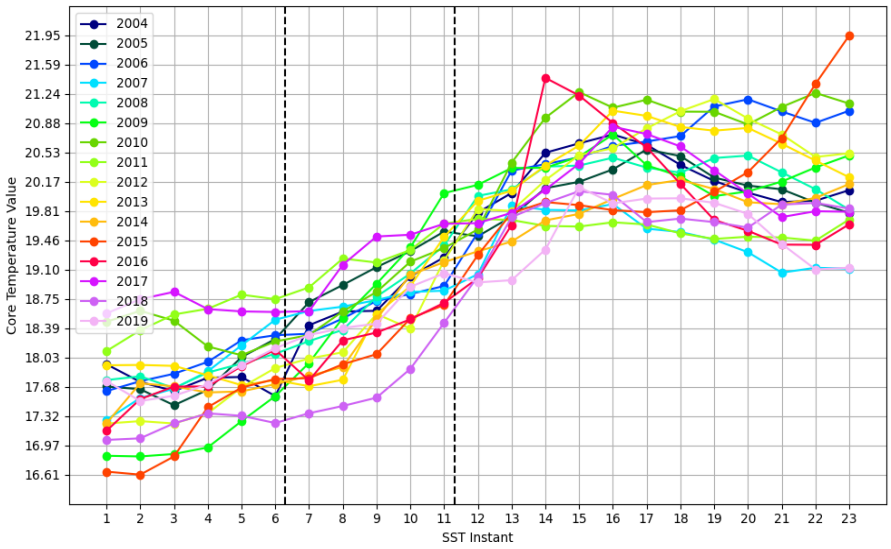


Figure B.16: Core temperature values across the 23 SST instants with each of the 16 years represented and "border" separating the different clusters for the region of South Morocco for K = 3

Figure B.17: Silhouette scores for different values of K using DBA with the Core areas for the region of North Morocco

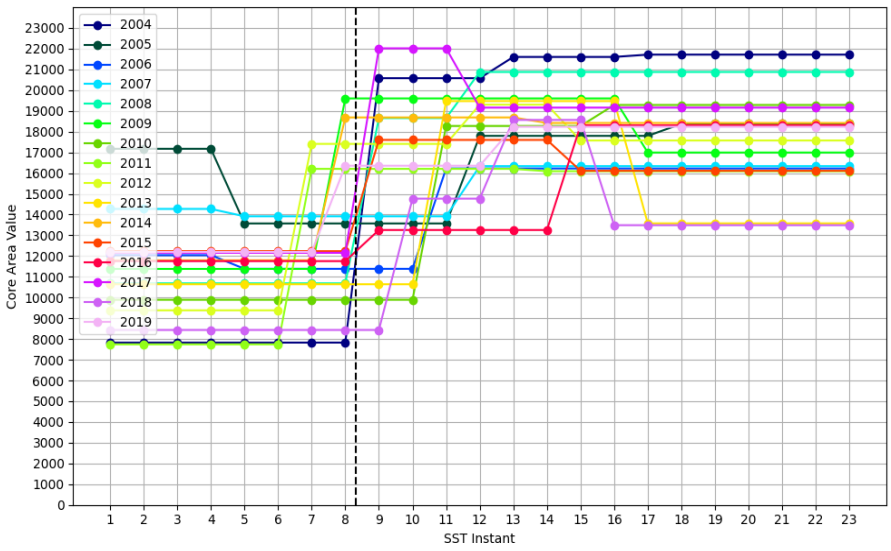


Figure B.18: Core area values across the 23 SST instants with each of the 16 years represented and "border" separating the different clusters for the region of North Morocco

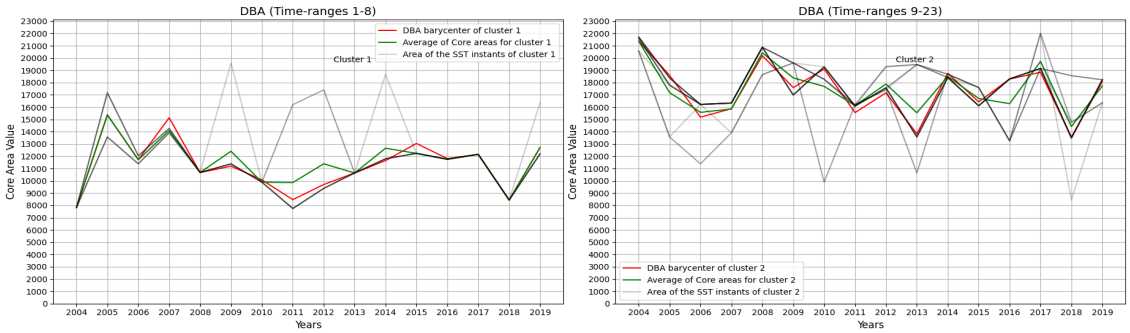


Figure B.19: Core area values across 16 years divided by each cluster obtained by the DBA algorithm, with the SST instant areas of each year, the value of the DBA barycenter and the average of the core areas for the region of North Morocco

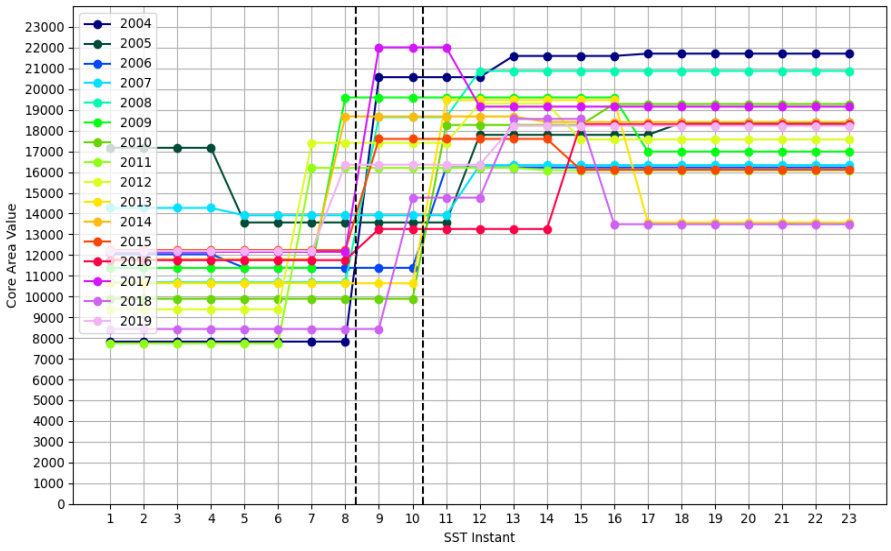


Figure B.20: Core area values across the 23 SST instants with each of the 16 years represented and "border" separating the different clusters for the region of North Morocco for K = 3

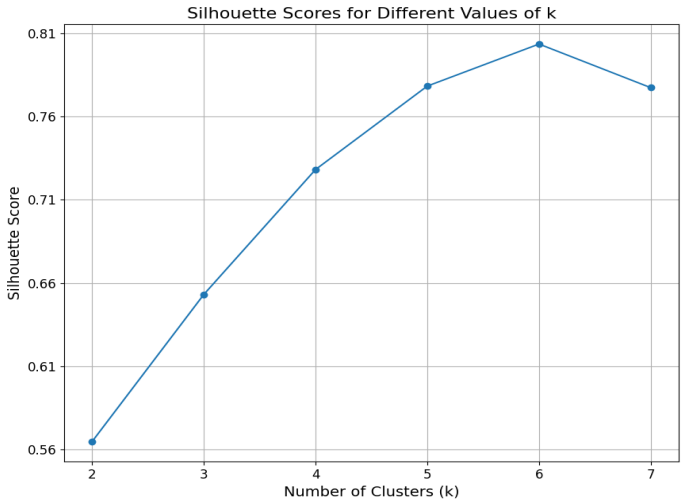


Figure B.21: Silhouette scores for different values of K using DBA with the Core areas for the region of South Morocco

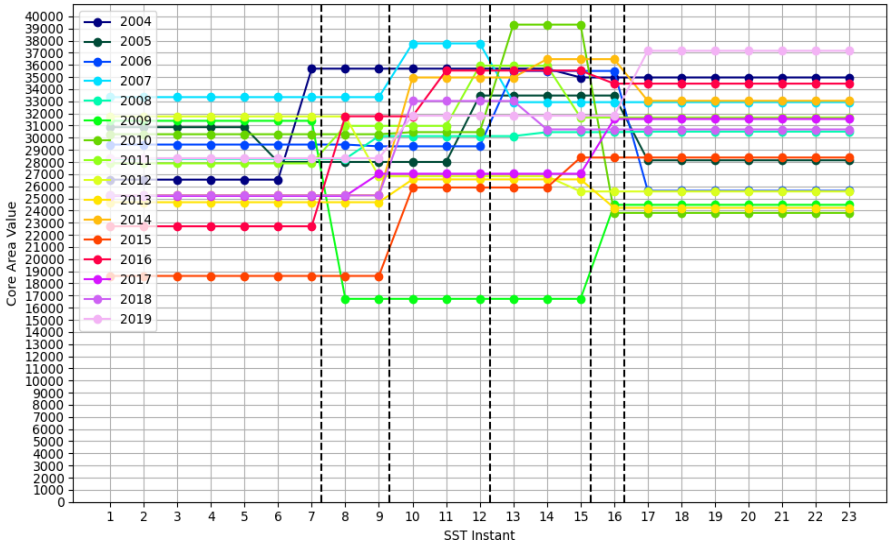


Figure B.22: Core area values across the 23 SST instants with each of the 16 years represented and "borders" separating the different clusters for the region of South Morocco

Figure B.23: Core area values across 16 years divided by each cluster obtained by the DBA algorithm, with the SST instant areas of each year, the value of the DBA barycenter and the average of the core areas for the region of South Morocco

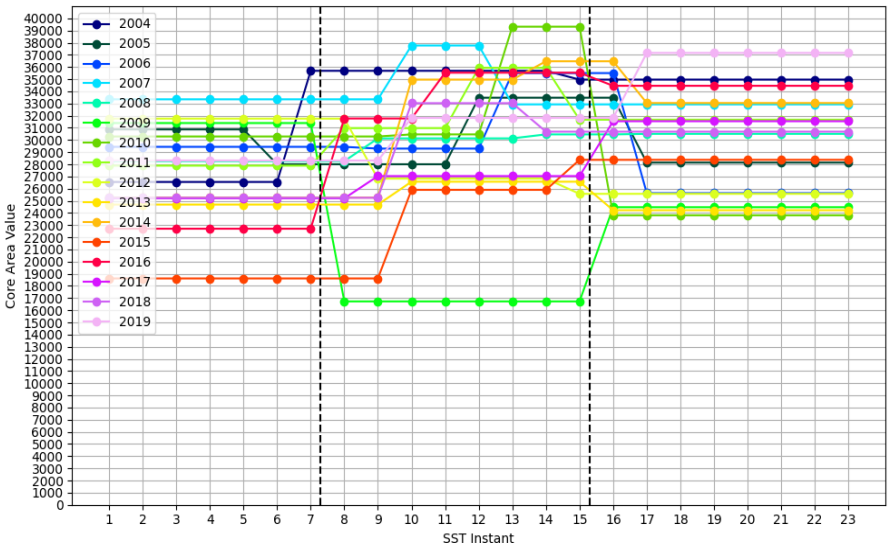


Figure B.24: Core area values across the 23 SST instants with each of the 16 years represented and "border" separating the different clusters for the region of South Morocco for K = 3