A HIERARCHICAL ST-DBSCAN ALGORITHM FOR SPATIO-TEMPORAL DATA CLUSTERING

AMALIA MABRINA MASBAR RUS

UNIVERSITI KEBANGSAAN MALAYSIA

A HIERARCHICAL ST-DBSCAN ALGORITHM FOR SPATIO-TEMPORAL DATA CLUSTERING

AMALIA MABRINA MASBAR RUS

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE MASTER OF COMPUTER SCIENCE

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2018

ALGORITMA ST-DBSCAN HIERARKI UNTUK PENGELOMPOKAN DATA SPATIO-TEMPORAL

AMALIA MABRINA MASBAR RUS

DISERTASI YANG DIKEMUKAKAN UNTUK MEMENUHI SEBAHAGIAN DARIPADA SYARAT MEMPEROLEHI SARJANA SAINS KOMPUTER

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

DECLARATION

I hereby declare that the work in this dissertation is my own except for quotations and summaries which have been duly acknowledged.

06 August 2020

AMALIA MABRINA MASBAR RUS
P75880

ACKNOWLEDGEMENT

In the name of Allah, Most Gracious, Most Merciful. May the shalawat and greetings be upon the great Prophet (Muhammad SAW). Alhamdulillah and gratitude goes to the presence of God for his knowledge, mercy, and guidance given to me to complete this research.

Special appreciation and thanks to Assoc. Prof. Dr. Zulaiha Ali Othman, my supervisor who has guided, advised, and gave me all the ideas required in order to complete this research.

I also want to thank all lecturers and staff of Faculty of Information Science and Technology who have devoted their knowledge with high dedication to guide me throughout the course of this study.

Overall, I dedicate this work to my parents Prof. Raja Masbar and Dr. Normalina Arpi who always give me encouragement and spirit to continue working. Last but not least, this work is dedicated to my husband, Sangga Rima Roman Selia, who is fully committed and always accompany me in every step of this research. I am proud to acknowledge that all your sacrifices are very meaningful and priceless to the completion of this dissertation.

May all the good deed and the help given to me in the completion of this dissertation be rewarded highly by Allah in the Hereafter.

Thank you.

ABSTRAK

Pengelompokan data spatio-temporal adalah merupakan prosedur yang sangat mencabar berikutan kerumitannya dalam mengelompok data ruang dan waktu bersama. Terdapat pelbagai pendekatan pengelompokan telah dicadangkan dengan menambahbaik algoritma pengelompokan tradisional seperti algoritma berasaskan ketumpatan, algoritma berasaskan pembahagian, dan algoritma berasaskan hierarki. Algoritma berasaskan ketumpatan digunakan secara meluas kerana algoritma ini boleh mengendalikan kelompok untuk bentuk yang tidak teratur. Salah satu pengelompokan berasaskan ketumpatan untuk data spatio-temporal yang biasa digunakan ialah ST-DBSCAN. Walau bagaimanapun, algoritma ini mempunyai batasan untuk menangani elemen temporal dalam data. Oleh itu, kajian ini mencadangkan satu algoritma yang boleh menggabungkan bahagian temporal data ke dalam algoritma ST-DBSCAN dengan memperkenalkan jarak temporal maksimum yang dipanggil Eps 3. Dalam kajian ini, nilai Eps digunakan untuk mengehadkan jarak temporal ketika melaksanakan pengelompokan berasaskan ketumpatan. Keputusan kajian ini menunjukkan bahawa ST-DBSCAN dengan Eps 3 telah meningkatkan 4 daripada 6 nilai purata prestasi indeks sebanyak 27%, tidak termasuk indeks Ksq Det W. Walau bagaimanapun, bilangan kelompok yang dihasilkan adalah sangat besar, hingga mencapai 240 kelompok, berbanding ST-DBSCAN yang hanya mempunyai 14 kelompok. Oleh itu, eksperimen kedua dijalankan untuk mengurangkan bilangan kelompok. Percubaan ini dijalankan dengan mengagregasi kelompok yang serupa berdasarkan persamaan temporal yang dikira menggunakan masa dinamik pantas dan kemudiannya hierarki kelompok dihasilkan menggunakan kaedah hierarki Ward. Untuk memilih tahap pemotongan hierarki, dua nilai ambang dipilih, iaitu 0.3 dan 0.1. Kemudiannya, ST-DBSCAN dengan hierarki yang dikenali sebagai algoritma ST-HDBSCAN telah dibangunkan. ST-HDBSCAN dengan algoritma ambang 0.3 telah berhasil mengurangkan bilangan kelompok yang menggunakan ST-DBSCAN dengan Eps 3 dari 240 kelompok kepada 6 kelompok. Berbanding dengan algoritma ST-DBSCAN, algoritma ini telah dapat meningkatkan 5 daripada 6 indeks prestasi sebanyak 73% secara purata.

ABSTRACT

Clustering spatio-temporal data is a very challenging procedure due to the complexity of clustering the spatial and temporal data together. There have been various clustering approaches that have been proposed by enhancing traditional clustering algorithm, such as density-based, partition-based, and hierarchical-based algorithms. The density-based algorithm is widely used because the algorithm can handle irregular shape cluster. One of the commonly used spatio-temporal density-based clustering is ST-DBSCAN. However, the algorithm has its limitation on addressing temporal element of data. Therefore, this research proposes an algorithm that can incorporate temporal part of the data into ST-DBSCAN algorithm by introducing maximum temporal distance called Eps 3. In this study, the Eps value is used to limit the temporal distance when performing density-based clustering. The results of this study show that the ST-DBSCAN with Eps 3 increased 4 out of 6 performance indices values by 27% in average, excluding Ksq Det W index. However, the number of clusters generated is enormous, reaching 240 clusters, compared to ST-DBSCAN that only has 14 clusters. In view of this, a second experiment was conducted to reduce the number of clusters. This was done by aggregating similar cluster based on their temporal similarity calculation using a fast dynamic time warping after which a cluster hierarchy was generated using hierarchical Ward's method. To select the level of cutting the hierarchy, two threshold values were selected, 0.3 and 0.1. Thus, the ST-DBSCAN with a hierarchy that is otherwise known as ST-HDBSCAN algorithm was developed. While the ST-HDBSCAN with threshold 0.3 algorithm, on the other hand, reduce the number of clusters from ST-DBSCAN, including Eps 3 of 240 clusters to 6 clusters. Comparing this to the ST-DBSCAN algorithm, the latter was able to improve 5 out of 6 performance indices by 73% on average.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION

Spatio-temporal data mining is an emerging research area that is currently in demand in order to obtain important information from the enormous amount of geographic location and time data. This is because, technology has allowed humans the easier access to obtain spatial and temporal data using GPS, satellite, wireless technology, sensor networks and other devices that could transmit location and time stamped data. Moreover, the organization has also invested more on gaining hidden knowledge and information from spatio-temporal data which make research on this field more important. Several researchers, such as Bogorny and Shekar (2010), Mazimpaka and Timpf (2015), Yuan (2016), and Atluri et al. (2017), had reviewed spatio-temporal data mining and its application ranging from mobile commerce, hot spot detection, climate change, forest fire (Tonini et al. 2017), ecology, reservoir, neuroscience, health care, social media and so on.

One of spatio-temporal data mining area is clustering. It is a process to group data based on similarity distance, in which objects that are similar are grouped together. Nowadays, several clustering algorithms are already available for traditional datatype. Generally, traditional clustering algorithm is divided into 5 categories, density-based, partition-based, hierarchical-based, grid-based, and model-based algorithms (Jiawei et al. 2012).

However, spatio-temporal data is different from traditional dataset in a number of ways. Firstly, some of spatial or temporal information are not directly available and need to be calculated initially (Rao et al. 2012). Secondly, the scale or granularity level

of spatial and temporal data also have an impact on data mining results (Rao et al. 2012). Thirdly, spatio-temporal data have auto-correlation, which means that data collected together in terms of location and time will have similar values (Bogorny & Shekhar, 2010; Rao et al. 2012).

Therefore, due to these unique characteristics of spatio-temporal data, it is required that data mining techniques for traditional data type is modified and adjusted in order to incorporate and exploit the pattern and relationships in spatio-temporal data (Rao, 2012). In this chapter, some research background about clustering spatio-temporal data, problem statements, objectives, and scope of this research would be explained.

1.2 RESEARCH BACKGROUD

The spatio-temporal dataset is different from traditional dataset in many ways. Firstly, it contains implicit spatial or temporal data that are initially needed to be calculated. Secondly, spatio-temporal data has a level of granularity and making its selection differently will have an impact on mining results. Thirdly, spatio-temporal data has an auto-correlation.

First, spatio-temporal data contains spatial and temporal data that are not explicitly stated in the data. For example, the distance, topological, seasonal and cycle are implicit information, which are not hard coded in the database but highly influence other attributes in the dataset (Rao et al. 2012). These data can be extracted by applying some calculation or preprocessing the data. For example, in spatial term, the data that are collected are in the form of latitude and longitude value which, if combined together can be used to pinpoint the location of the object in the world map. By using these two data, the distance between an object can be calculated through Euclidean distance or Haversian distance in order to get more information such as whether two objects are close to each other or not. This kind of information is not explicitly available, but need to be calculated at first hand.

The second reason is that the scale or granularity of spatial and temporal data also has an impact on data mining results. For example, in terms of spatial perspective, when mining the data regionally or globally, the result would be different. Similarly, in a temporal perspective, when the data are mined in hourly or daily basis, the result is also different and thus interpreted differently. This means that, it is contributing into the difficulty of mining spatio-temporal data properly (Rao et al. 2012).

The third reason is that the auto-correlation occurs for spatial and temporal data. In order to justify this in spatio-temporal data, the location and time were taken into account in the analysis process. This is because the value of an object is different if the location and time is not the same. This effect provides more insight when analysing the object (Bogorny & Shekhar, 2010; Rao et al. 2012). For example, the different positions of climate stations record different weather measurement. In other words, the closer the location of the two stations, the more similar the weather measurement would be. This auto-correlation of spatial data concept is called the Tobler's First Law of Geography (Faghmous & Kumar, 2014; Hsu & Lee, 2008; M. P. McGuire & Ziying, 2013).

Similarly, the concept applied to the temporal area, in which the values of measurement taken for the closer time interval are more similar. More importantly, these auto-correlations for time occur not only in a closer time interval, but also in a cyclic manner or repetitive pattern (Mazimpaka & Timpf 2015). For example, the weather in November, December, and January are quite similar yearly due to the occurrence of the season. Therefore, the spatial and temporal data have to be processed differently in order to obtain unbiased and hidden information related to spatial and temporal area.

Due to these three reasons, clustering spatio-temporal data is a challenge to many researchers in data mining field (Fischer & Nijkamp 2014). It is a process to group similar objects into the same cluster and different objects into different clusters (Jiawei et al. 2012; Ratanamahatana et al. 2010). The concept of similarity or dissimilarity is important in clustering algorithm (Rokach & Maimon 2010). Defining similarity or dissimilarity can be accommodated by using distance measure.

In general, there are five types of clustering algorithms, which are; density, partition, hierarchical, grid and model-based algorithm respectively (Jiawei et al. 2012). The density-based algorithms can cluster any shape of cluster and have quite good performance in terms of execution time. The algorithms are sensitive to the order of the input data, which means that if the input data are in different order, the result will also not be the same (Saraswathi & Sheela 2014). Nevertheless, some density-based algorithm, such as OPTICS (Ankerst et al. 1999), has deal with this problem by ordering the object during the clustering process.

As for hierarchical-based algorithms (Fionn & Pedro, 2012), it is computationally expensive if it is performed at a very low level of data granularity, since processing all input data together is required before generating the level (Ratanamahatana et al. 2010). However, the results can be found at several levels, which make it possible to see the relationship between the data and simplify the cluster according to appropriate level of interpretation (Saraswathi & Sheela 2014).

In terms of the grid-based algorithm, Ilango & Mohan, (2010) stated that, it is related more towards modelling the data before applying a clustering algorithm. Several examples of general grid-based clustering include STING (Wang et al. 1997), CLIQUE (Agrawal et al. 1998), and Wave Cluster (Sheikholeslami et al. 1998). The difference between grid-based and other algorithms is the time of gridding the data before applying the clustering process. The process is performed after the gridding. This is basically similar to partition-based or density-based algorithm. For example, ST-AGRID (Fitrianah et al. 2015), performed density-based clustering after the gridding process, and Geographically Robust Hotspot Detection (GRHD) (Eftelioglu et al. 2016), performed partition-based shared-nearest neighbor cluster after gridding the data, although it was only performed on spatial data.

Likewise, the model-based clustering algorithm, such as COB-WEB (Fisher 1987), incorporated similar algorithms to density-based and partition-based. An algorithm is considered to be model-based if it is incorporating statistic model for distance measure or as a mean to limit number of cluster in density-based or partition-

based algorithm. Therefore, the categories that are considered in this research are only density, partition, and hierarchical-based clustering algorithms respectively.

Among the five clustering algorithms, many of the clustering researches expanded the density-based algorithm compared to other clustering techniques (Birant & Kut, 2007; Chen et al. 2015; Hüsch et al. 2018; Kisilevich et al. 2010; Lee, 2012; Oliveira et al. 2013; Santos et al. 2015; Zhang & Eick, 2016). In addition, compared to partition-based clustering algorithm, which is only able to cluster convex or spherical cluster shape, density-based is able to handle irregular cluster shape (Agrawal et al. 2016). Therefore, in this research, density-based clustering algorithm is proposed to improve its ability to handle spatio-temporal data.

1.3 PROBLEM STATEMENT

One of spatio-temporal clustering algorithm that is widely used for comparing a newly develop spatio-temporal clustering algorithm is Spatio-temporal Density-based Spatial Clustering Application with Noise (ST-DBSCAN) Algorithm (Birant & Kut, 2007). This algorithm expands the Density-based Spatial Clustering Application with Noise (DBSCAN) Algorithm (Ester et al. 1996), which is one of the popular density-based algorithms. This DBSCAN algorithm uses an ops' value to limit maximum distance in deciding whether an object is a neighbor of another object. In ST-DBSCAN, the DBSCAN algorithm was enhanced by introducing a new distance limit for spatial data called Eps 2. In terms of its research, sea surface temperature, sea surface height and wave height in the Black Sea, the Marmara Sea, the Aegean Sea, and Mediterranean Sea are being clustered. The result of the clustering was presented in the form of map labeled with a cluster number for each area. By using cluster map, the result focused more on the spatial part of data and lacking in showing the temporal part of data.

The problem with ST-DBSCAN is the lack of the algorithm on the temporal aspect of spatio-temporal data. This is because the temporal data were separated manually by filtering the data that occurred on the consecutive day or the same day in different year (Birant & Kut, 2007). Thus, the algorithm lacked the identification of the cluster with a pattern that exists continuously in two different years. The cluster

generated took into account the spatial and non-spatio-temporal part of data; however, it lacks the usage of the temporal aspect of the data.

Therefore, in order to spatially and temporally cluster the data, there is a need to develop an algorithm that could incorporate and take into account the spatial and temporal aspect of data. Based on this information, in this research, the density-based algorithm ST-DBSCAN is enhanced by incorporating a temporal distance limit called Maximum Temporal Distance (Eps3). However, the number of cluster generated from this algorithm is enormous, reaching 240 clusters. Thus, second experiment is performed to reduce the number of clusters. In doing this, the clusters generated from previous experiment are aggregated using hierarchical method. This technique was used recently in ST-OPTICS where the algorithm combined the density-based and hierarchical based method to cluster spatio-temporal data (Agrawal et al. 2016).

The ST-OPTICS algorithm employs the used ST-DBSCAN to extract the clusters and then performs hierarchical clustering to aggregate the clusters. The result generated has better performance indices compared to ST-DBSCAN. However, comparing to a similar version of ST-DBSCAN, ST-OPTICS still performs clustering using spatial and non-spatio-temporal aspect of the data but lacks the temporal aspect of the data. Similarly, the results are also shown in the form of maps which limit the description of temporal part of data. Therefore, there is still a need of clustering algorithm that could cluster spatio-temporal data that will take into account not only spatial and non-spatio-temporal part of data but also the temporal part of data.

1.4 OBJECTIVE OF RESEARCH

The objectives of this research are:

i. To propose an improvement of Spatio-Temporal Density-based Spatial Clustering of Applications with Noise (ST-DBSCAN) algorithm that has a Maximum Temporal Distance (Eps 3) by including temporal element in clustering spatio-temporal data.

ii. To enhance the proposed algorithm by reducing the number of clusters using hierarchical Ward's method and Fast Dynamic Time Warping.

1.5 SCOPE OF RESEARCH

The scope of this research is limited to El-Nino dataset. It focuses on clustering process but excludes the understanding of the meaning of the cluster. The study is also conducted on the property that makes a cluster but rule out the explanation on why the cluster is created. This is because this research is purely conducted in order to generate algorithm that can cluster spatio-temporal data based on its spatial, temporal, and non-spatio-temporal properties. However, this clustering algorithm is limited to fixed location and does not handle shape, line or trajectory types of data. In terms of temporal, it can handle continuous type of data with daily level of granularity.

1.6 METHODOLOGY

In this research, experimental methodology is used which consists of five main activities, such as identifying problems, data pre-processing, development and experiment of proposed algorithm, development and experiment of enhanced algorithm, and evaluation as shown in Figure 1.1.

These experiments are divided into two. The first (Experiment 1) is conducted by adding maximum temporal distance (Eps 3) to ST-DBSCAN. The second (Experiment 2) is conducted by aggregating the clusters generated from Experiment 1 using hierarchical clustering algorithm. Detail explanation of this research methodology is described in Chapter III.

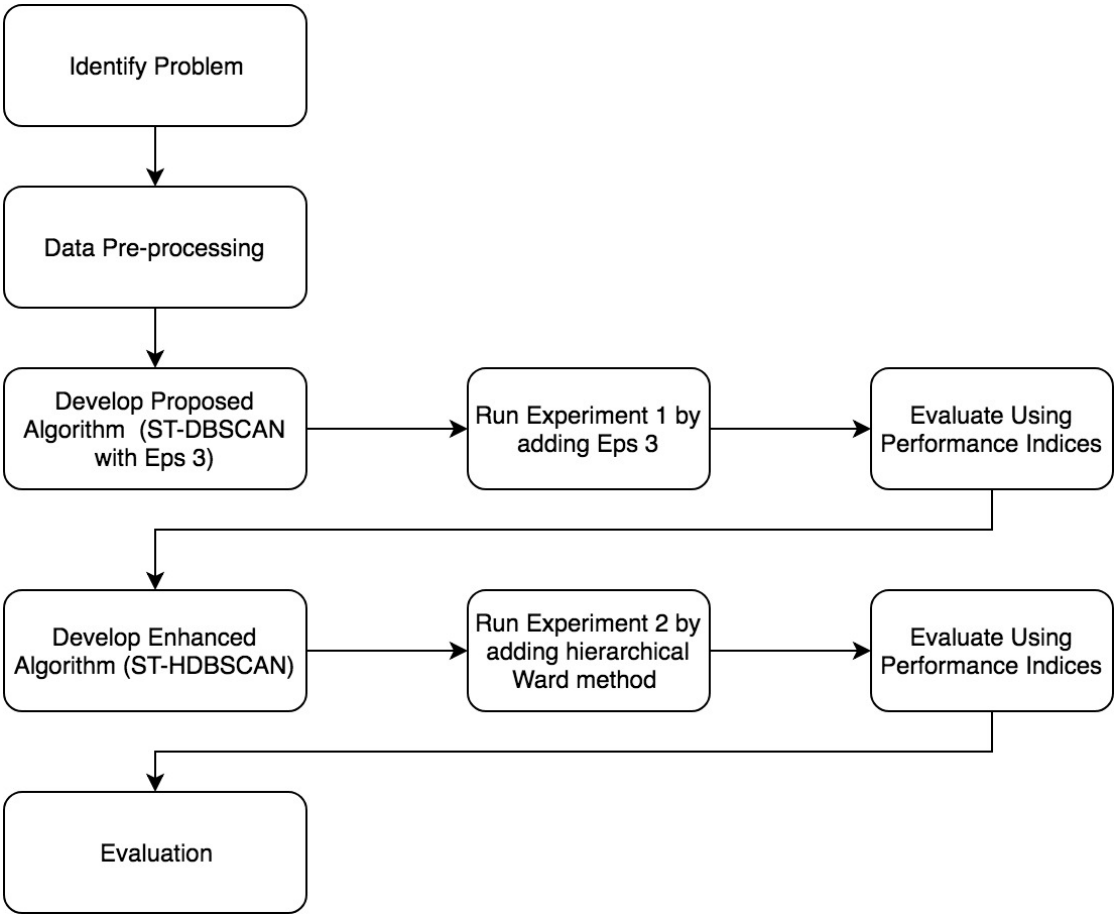

Figure 1.1 Experimental research methodology

1.7 DISSERTATION ORGANIZATION

This dissertation is organized into 7 chapters. The first chapter is the introduction, which explains the research background, problem statement, objectives of research, scope of works, methodology, and dissertation organization.

Chapter II describes the literature review of previous works that has been performed in the area of spatio-temporal clustering research. It explains the spatio-temporal data types, existing spatio-temporal clustering algorithms, comparative analysis of spatio-temporal clustering algorithms and several information backgrounds about ST-DBSCAN algorithm, ward linkage method, fast dynamic time warping, and performance indices.

Chapter III describes the experimental research methodology used in this study. The method consists of five activities which are identifying problems, data pre-processing, development and conduct of the experiment of proposed algorithm, development and conduct of the experiment of enhanced algorithm, and evaluation of the algorithm using several performance indices for spatio-temporal clustering algorithms. This research uses El-Nino dataset as the data sample to calculate performance indices of the proposed algorithm.

Chapter IV explains the first experiment which is ST-DBSCAN with maximum temporal distance. It explains the algorithm of ST-DBSCAN with maximum temporal distance and the stings of its parameter. Using similar format, Chapter V explains the second experiment which is ST-HDBSCAN Algorithm.

In Chapter VI, results of both experiments and evaluations using several performance indices are explained. This chapter also compares the three results of the algorithms, such as ST-DBSCAN, ST-DBSCAN with Eps 3, and ST-HDBSCAN.

Lastly, in Chapter VII the conclusion of the research is presented and some future works directions are explained. This chapter concludes the research process and highlighting its contributions.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

Spatio-temporal data mining is one of the challenging areas in research that demands high problem-solving skill. This is because the data itself combine two different types of data which need different techniques to handle. It is concerned with the location, movement, or shape of object. Therefore, clustering this type of data will result in objects to be close in distance, have similar movements or similar shapes in the same cluster. Comparing it to temporal data, this type of data is concerned with time-series, seasonality and cycles, which needs different types of treatments in order to get the pattern or cluster. The result of clustering temporal data would be a cluster of similar days or seasonality, including cycle that is repeated over time.

In order to understand spatio-temporal data, this chapter explains its types and classification based on the changes in spatial data and terms of temporal data. This results into six types of spatio-temporal data types.

Furthermore, this chapter explains several types of clustering algorithm that is available for spatio-temporal data. The clustering algorithms are categorized into five, namely; Density, Hierarchical, Partition, Grid, and Model-Based Algorithms respectively. After explaining types of clustering algorithms, comparative analysis is performed.

More so, this chapter discusses several topics related to the development of proposed and enhanced algorithms, such as ST-DBSCAN algorithm, Time Series, Fast

Dynamic Time Warping, Ward Linkage Method, and several performance indices that was used to evaluate the proposed and enhanced algorithms.

2.2 SPATIO-TEMPORAL DATA TYPE

There are 5 types of spatio-temporal data (Kisilevich et al. 2009). As shown in Figure 2.1, the data types are divided based on temporal and spatial extension, with spatial location. The spatial extension expands the shape from points into lines and then areas. But in temporal extension, it starts from single snapshot to updated one and lastly to the complete time-series. Meanwhile, the spatial location is divided into two categories, either the data is collected from fixed or dynamic location.

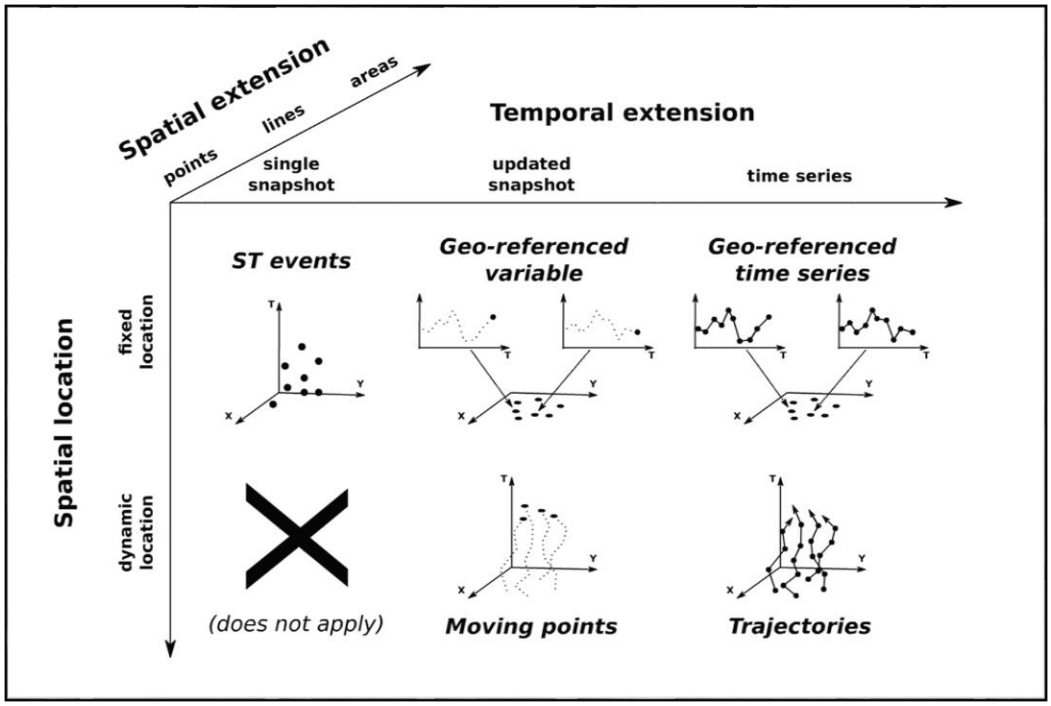

Figure 2.1 Spatio-temporal data type

Source : Kisilevich et al. (2009)

1) Spatio Temporal Events: this data type has fixed location and only store one snapshot of variable values.

2) Geo-referenced variable: this data type also has fixed location. For each location, it stores the updated value of variable but do not store historical values, in which it focuses only on the recent updated values.

3) Geo-referenced time series: this type of data also has fixed location. In which for each location, it stores the entire history of variable values as time-series data.

4) Moving points: in this type of data, the object changes its location overtime. However, only recent values are recorded.

5) Trajectories: this type of data also has dynamic location and stores each movement of the object as a complete time-series.

In this research, the dataset used is El-Nino and in tabular format. It consists of spatial attributes in the form of latitude and longitude values with almost fixed location. It also contains the temporal values (date, month, and year), and non-spatio-temporal values (sea surface temperature, zonal wind, meridional wind and several other attributes). Therefore, since it has almost fixed location and time series of non-spatio-temporal data, the data type for El-Nino dataset is Geo-referenced time series.

2.3 CLUSTERING SPATIO-TEMPORAL DATA

Traditional clustering algorithms in general are divided into 5 categories namely: density, hierarchical, partition, grid, and model-based algorithms respectively (Jiawei et al. 2012). Generally, clustering algorithm for spatio-temporal data is mostly based on traditional clustering algorithms that are adjusted to be applied for spatio-temporal data.

However, the categories of clustering algorithm considered in this research are density, partition and hierarchical-based respectively. This is because the grid and model-based algorithm are basically using either density, partition or hierarchical-based algorithm as the core of their algorithm.

Grid-based algorithm (Ilango & Mohan 2010) on the other hand, is related more on modeling the data before applying clustering algorithm. Several examples of general grid-based clustering, include STING (Wang et al. 1997), CLIQUE (Agrawal et al. 1998), and WaveCluster (Sheikholeslami et al. 1998). The difference between grid-based and other algorithm categories is in terms of gridding the data before applying the clustering process. The process performed after the gridding is basically similar to partition or density-based algorithm. For instance, ST-AGRID (Fitrianah et al. 2015) which performed density-based clustering after the gridding process and Geographically Robust Hotspot Detection (GRHD) which performed partition-based shared-nearest neighbor cluster after gridding the data, even though it was performed on spatial data only (Eftelioglu et al. 2016).

Likewise, the model-based clustering algorithm, such as COB-WEB (Fisher 1987), incorporated similar algorithms to density and partition-based. An algorithm is considered to be model-based if it incorporates statistical model for distance measure or as a mean to limit number of cluster in density or partition-based algorithm. In view of this, this research only considers density, partition, and hierarchical-based clustering algorithms categories respectively.

2.3.1 Density-based Algorithm

The algorithm is considered as density-based if it uses density threshold to search for the area where interesting data points are gathered and distinguished from noise (Jiawei et al. 2012). This type of algorithms looks for areas where many data are gathered or close to each other. One of the first density-based algorithm is DBSCAN (Ester et al. 1996). This algorithm scans all objects in dataset and labels each object as core object, border object, or noise based on distance range called Epsilon and minimum number of points, MinPts. If an object is in Epsilon range, then the object is considered as neighbor of the object, and could be grouped into one cluster. Conversely, if an object has a number of neighbor objects in its Epsilon range that is more than MinPts, the object is labeled as core object. But if the object is not labeled as belong to any cluster, then the object is considered as noise.

Other traditional density-based algorithm is OPTICS (Ankerst et al. 1999), which improve DBSCAN algorithm by ordering the data points based on its distance to another before searching for the core object, border object and noise. Another traditional algorithms are Shared Nearest Neighbours (SNN) (Ertöz et al. 2003), DENCLUE (Hinneburg & Gabriel 2007) and CURD (Ma et al. 2003).

In terms of the spatio-temporal data in density-based algorithm, most researches are based on DBSCAN algorithm (Ester et al. 1996), by improving or adding more adjustment to accommodate spatio-temporal data. Also, ST-DBSCAN adds minimum distance parameter to spatial data (Birant & Kut, 2007). This is because the algorithm uses a new epsilon variable for spatial data. Thus, two epsilons are used for determining the neighbor of core object, in which one is the epsilon to limit spatial data and the other is to limit the non-spatio-temporal data range. More detail explanation of ST-DBSCAN algorithm is available in Section 2.5 of this study.

Another example of density-based clustering spatio-temporal data is Clustering Dynamic Spatio Temporal Data (Chen et al. 2015), which also improves DBSCAN algorithm by adding temporal similarity distance. DBSCAN has also been adjusted and implemented in text mining area for Twitter Spatio-Temporal data to detect which event occurs and estimate where the event occurs, that is the area based on time-zone (Lee, 2012). Also, DBSCAN has been expanded to analyze places and events which is used to cluster geo-tagged photos in PDBSCAN algorithm (Kisilevich et al. 2010).

Other technique employed is by expanding other traditional density-based clustering algorithm such as Shared Nearest Neighbor (Ertöz et al. 2003). Spatio-temporal Shared Nearest Neighbor (ST-SNN) and Spatio-temporal Separated Shared Nearest Neighbor (ST-SEP-SNN) but add polygon distance formula and algorithm to search core polygon (Wang, Cai, & Eick, 2013). Their result has been compared with PDBSCAN. 4D+SNN also added weighting for spatial, temporal and other attributes (Oliveira, Santos, & Pires, 2013). Another recent research that was carried out on clustering spatio-temporal data is the use of SNN as the basis algorithm that emphasize on the batching process of ST-DCONTOUR (Zhang & Eick, 2016).

Recently, one of density-based algorithm for spatio-temporal data was extended to ST-DBSCAN algorithm by using correlation function called CorClust (Hüsch et al. 2018). The algorithm clusters spatial data that has been used over time is Pearson correlation as a similarity distance to generate spatio-temporal cluster. Thus, the algorithm cluster results can be compared to various period of time.

2.3.2 Partition-based Algorithm

In partition-based algorithms, minimum distance threshold is used to cluster the data point (Jiawei et al. 2012). It is different from density-based clustering in which it is calculated based on the minimum number of point in order to decide whether a set of data point can be considered as core cluster or not (Jiawei et al. 2012). On the other hand, in the partition-based algorithm, the minimum number of data point in one cluster is not decided. Each data point is assigned to one of the closest cluster center (Jiawei et al. 2012). Thus, in such an instance, it is possible to have a cluster that contains only one data point. This also means that, the algorithm is sensitive to outlier data. Also, due to the fact that the algorithm is quite simple, it could perform faster compared to other type of algorithms (Jiawei et al. 2012). There are several examples of general partition-based algorithms, such as K-Means (Macqueen, 1967) and its variants (Baboo & Tajudin, 2013; Garg & Jain, 2006; Geng Zhang et al. 2018), K-Medoid (Vinod, 1969), Partitioning Around Medoids (PAM) (Kaufman & Rousseeuw, 1990) and CLARANS (Ng & Han, 1994).

Partition-based Algorithms for spatio-temporal data types are mostly an adaptation of k-means or fuzzy c-means algorithm that is into spatio-temporal data. K-means algorithm uses centroid and calculates distance of each point to another centroid and assigns each point into the closest centroid (Macqueen, 1967). Similarly, fuzzy c-means (Dunn, 1973) uses the same concept but allows a data point to be assigned to several clusters based on some degree of similarities. For example, one data point can belong to a cluster and up to 20% similarity and other cluster up to 80% similarity.

One of the recent partition-based algorithms that modified k-means algorithm to be able to process and incorporate spatial and temporal part of data is Spatio-temporal

Neighborhood Discovery for Sensor Data (McGuire, Janeja, & Gangopadhyay, 2013). In the study, the use of Geo-referenced ST data in tabular form with fixed location for spatial data and time series data for temporal was employed. It generated spatial neighborhood by using discretized temporal interval, and the spatial neighborhood is generated using graph model with the node as location of data and the arc/line as the similarity measure distance. The temporal data also is discretized by using agglomerative clustering with sum of error as the similarity measure (McGuire, Janeja, & Gangopadhyay, 2013). The results of neighborhood graph and agglomerative temporal cluster were then combined to get spatio-temporal graph cluster, in which the data point was close in spatial neighborhood and the same level of temporal discretization. This allowed overlapping, which means there are data points that belong to two clusters, which are positioned in the boundaries of the two close clusters (McGuire, Janeja, & Gangopadhyay, 2013). This algorithm combines partition-based and hierarchical-based algorithm.

Another example of partition-based algorithm that modifies fuzzy c-means is Augmented Fuzzy C-Means (Izakian et al. 2013). In this method, the cluster is generated by using line to separate regions. Its prediction is quite similar to original data but may highly fit the data. This is because, the algorithm is compared with Robust Fuzzy C-Means (RFCM) (Pham, 2001). This is done by comparing three time-series data model, such as Discrete Fourier Transform (DFT), Piecewise Aggregate Approximation (PAA), and Discrete Wavelet Transform (DWT).

2.3.3 Hierarchical-based Algorithm

In hierarchical-based algorithm, data are divided into several groups, in which in each step, the algorithm aggregates or separates the data into several clusters according to location or time. For example, spatial data can be grouped from several cities into a state, several states into regions, and several regions into countries. But for temporal data, it can be grouped based on time interval, such as daily, weekly, monthly, or yearly. The main problem in hierarchical based is deciding which level to be used in order to get the best clustering results. Several examples of hierarchical clustering algorithm include ROCK (Guha et al. 2001), Chameleon (Karypis et al. 1999), CURE (Guha et

al. 1998) and BIRCH (Tian Zhang et al. 1996). Lee and Crawford also used hierarchical clustering with Bayesian to classify image (2005).

Hierarchical-based clustering algorithm implementation in spatio-temporal data is mostly combined with density-based or partition-based algorithm. ST-OPTICS includes ST-DBSCAN and Agglomerative Hierarchical Clustering since the cluster produced are always too many and small (Agrawal et al. 2016). MR-DBSCAN (MapReduce-DBSCAN), also tried to use MapReduce model and combine the DBSCAN algorithm with hierarchical agglomerative algorithm (He et al. 2011). Conversely, SM-DBSCAN used density-based and then hierarchical agglomerative to cluster earthquake magnitude (Georgoulas et al. 2013). A recent research in hierarchical-based clustering also combined the use of a partition-based clustering for neonatal data (Mago et al. 2018).

2.4 COMPARATIVE ANALYSIS OF CLUSTERING ALGORITHMS

In this research, extensive literature reviews were performed in order to select the best clustering algorithm for spatio-temporal data. Table 2.1, 2.2 and 2.3 respectively summarize the comparison of clustering algorithm for spatio-temporal data using density, partition, and hierarchical-based algorithm respectively.

The data formats used for the algorithm in testing are basically stored in tabular format, or from the map they are converted to grid and then to tabular format. As mentioned in Table 2.1, 2.2, and 2.3 respectively, some of the data are modeled using other data structure model such as R-Tree Indexing Model, Vector, Polygon, Dynamic Term Weighting, STARIMA, or bivariate Gaussians.

The density-based algorithms are able to cluster any shape and have quite good performance in terms of execution time. However, the algorithms are sensitive to the order of the input data, which means if the input data is in different order, the result will also be different (Saraswathi & Sheela 2014). Nevertheless, some density-based algorithm, such as OPTICS (Ankerst et al. 1999), has dealt with this problem by ordering the object while performing the clustering process.

In terms of partition-based algorithms, the execution time is better than other algorithms, since the algorithm is quite simple. It can handle fuzziness when using fuzzy c-means and process the data in any order (Izakian et al. 2013; Shamshirbanda et al. 2015). However, it is very sensitive to outliers and can only detect spherical shape cluster (Saraswathi & Sheela 2014). Spatio-temporal data on the other hand, are mostly irregular in shape, thus, difficult to enhance the partition-based algorithm for handling spatio-temporal data (Agrawal et al. 2016).

As for hierarchical-based algorithms (Fionn & Pedro 2012), it is computationally expensive, if it is performed with a very low level of granularity data because it requires processing the combination of all input data before the level is generated (Ratanamahatana et al. 2010). However, the results can be seen in several levels, which make it possible to get the relationship among the data and simplify the cluster according to which level appropriate for interpretation (Saraswathi & Sheela 2014).

Moreover, if the hierarchical process is combined with density-based or partition-based algorithm, the number of object to be hierarchically-clustered can be minimized. Therefore, performing clustering using density or partition-based algorithm before applying hierarchical algorithm could produce a more advanced algorithm than using only density or partition-based algorithm (Agrawal et al. 2016).

Table 2.1 Comparison of density-based algorithm

Density-based Algorithm	Base Algorithm	Application	Data Type	Data Source Model	Data Representation Model	Size data	Distance Measure	Order of Data	Shape of Cluster
ST-DBSCAN (Birant & Kut 2007)	DBSCAN	Climate	ST Events	Tabular	R-Tree Indexing Model	large	Euclidean Distance	in order	irregular shape
P-DBSCAN (Kisilevich et al. 2010)	DBSCAN	Geo-tagged Photo	Spatial	Tabular	N/A	small	N/A	N/A	irregular shape
Mining ST Info (C. H. Lee 2012)	DBSCAN	Microblogging Streams	ST	Tabular	Dynamic Term Weighting	N/A	cosine similarity	N/A	irregular shape
Clustering Dynamic ST Patterns (Chen et al. 2015)	DBSCAN	Climate , global water body	ST Events	Pixel (grid)	N/A	small	N/A	in order	irregular shape
4D+ SNN (Oliveira et al. 2013)	SNN	Fire in Portugal 2011	ST Events	Tabular	Vector	medium	Euclidean Distance and 3D distance formula	any order	irregular shape
ST-DCONTOUR (Y. Zhang & Eick n.d.)	SNN	Taxi trips data	ST Events + Contour	Tabular and Contour	Bivariate Gaussians	large	N/A	N/A	irregular shape
ST-SNN and ST-SEP-SNN (Santos et al. 2015)	SNN	Climate, ozone	ST Georeferenced Time-series	Tabular and Contour	Polygon	large	Euclidean Distance	any order	irregular shape
CorClustST (Hüsch et al. 2018)	ST-DBSCAN	Climate	ST Georeferenced Time-series	Map	Grid	medium	Correlation Value	in order	irregular shape

Table 2.2 Comparison of partition-based algorithm

Partition-based Algorithm	Base Algorithm	Application	Data Type	Data Source Model	Data Representation Model	Size data	Distance Measure	Order of Data	Shape of Cluster
CS-FCM-STT	Fuzzy C-Means	Social Economic Indicator in Italy	ST Georeferenced Data	Tabular	Vector	small	N/A	any order	convex or spherical
CFFCM (Cloud Fuzzier Fuzzy C-Means) (Qin et al. 2010)	Fuzzy C-Means	Climate, Sea Surface Temperature	Time Series	Tabular	Interval Type 2 FCM	medium	Pearson's correlation function	any order	convex or spherical
FIRST Cluster Analysis-Based (Mills et al. 2011)	K-Means	Climate, Phenology, Forest Fire Warning	ST Georeferenced	Map	Map	medium	Euclidean Distance	N/A	convex or spherical
Parallel k-Means Clustering (Kumar et al. 2011)	K-Means	Ecology, Phenology	ST Georeferenced	Map	Map	large	Euclidean Distance	N/A	convex or spherical
An Augmented Fuzzy C-Means (Izakian et al. 2013)	Fuzzy C-Means	Climate, Temperature Data of Alberta	ST Georeferenced	Tabular	DFT, PAA, and DWT	N/A	Enhanced Euclidean Distance	N/A	convex or spherical
Clustering Centroid Finding Algorithm (CCFA) (Baboo & Tajudin 2013)	K-Means	Climate, Hurricane, India	ST Events	Tabular	N/A	small	Euclidean Distance	any order	convex or spherical
ST Neighborhood (Michael P. McGuire et al. 2010)	K-Means	Climate, Sea Surface Temperature and Traffic	ST Georeferenced	Tabular	Graph Model	medium	Euclidean Distance	N/A	convex or spherical

Table 2.3 Comparison of hierarchical-based algorithm

Hierarchical-based Algorithm	Base Algorithm	Application	Data Type	Data Source Model	Data Representation Model	Size data	Distance Measure	Order of Data	Shape of Cluster
MR-DBSCAN (He et al. 2011)	DBSCAN	Taxi GPS Location	Spatial	Text Format	MapReduce Model	large	N/A	N/A	irregular shape
Seismic Mass DBSCAN (Georgoulas et al. 2013)	DBSCAN	Seismic Earthquake	ST Events	Tabular	N/A	medium	N/A	in order	irregular shape
ST-OPTICS (Agrawal et al. 2016)	OPTICS, ST-DBSCAN	Environmental, Forest	ST Georeferenced Time series	Map into Tabular	N/A	large	Euclidean Distance	any order	irregular shape

2.5 ST-DBSCAN ALGORITHM

ST-DBSCAN Algorithm is developed by modifying DBSCAN algorithm. The idea is to add a new epsilon to limit the distance of spatial data and include it as the criteria to calculate similarity of data in terms of spatial distance.

Previously, DBSCAN only has 1 distance function and epsilon (*Eps*). The distance function was applied for all variables, including the spatial data. The Eps was used to limit distance of objects in order to decide whether other object could be considered as a neighbor.

In order to improve DBSCAN, ST-DBSCAN algorithm incorporated the spatial part of data to the clustering algorithm, and thus being considered as spatio-temporal clustering algorithm. As a result, there are two Epsilons in ST-DBSCAN, in which one is for spatial data and the other is for non-spatial data.

However, ST-DBSCAN algorithm did not incorporate the temporal part of data into the clustering process. Based on this, this research includes the temporal part of data into the clustering process; thereby improves the ability of the algorithm to truly cluster the data according to the space and time.

There have been several concepts in ST-DBSCAN algorithm that has been used in various studies. Their definitions are as shown below (Birant & Kut 2007). These definitions are based on Birant and Kut (2007) with some minor improvements and different representations.

a. Definition 1: Distance

The first definition of distance is that it can be calculated by using formula, such as Manhattan Distance, Euclidean Distance, Haversine Distance and so on. Also, distance between two objects (o and p) can be denoted using function dist (*o,p*). In ST-DBSCAN, distances are calculated for both spatial data (*spatial_dist*) and non-spatial data (*var_dist*).

b. Definition 2: Neighbourhood

The second definition views it as neighborhood of object o and can be defined *as* $\{p \in D | dist(o,p) \leq Eps\}$ where p is another object in database *D* and *Eps* is value to define minimum distance between object o and p. In ST-DBSCAN, there are two Eps, in which one is for spatial data (*Eps1*) and the other is for non-spatial data (*Eps2*).

$$(spatial\_dist(o,p) \leq Eps1) \wedge (non\_st\_dist(o,p) \leq Eps2) \longrightarrow Neighbour(o,p) \quad \ldots(2.1)$$

c. Definition 3: Core object

The third definition sees it as a core object that can satisfy minimum number of neighborhood within radius of Eps that should be larger than *MinPts*.

$$NumNeighbour(o) \geq MinPts \longrightarrow CoreObject(o) \quad \ldots(2.2)$$

d. Definition 4: Directly density-reachable

The fourth definition says it is an object o that is Directly Density-Reachable (DDR) from object *p*, in which it believes that if *p* is the neighbor of *o*, then *o* is a core object.

$$Neighbour(o,p) \wedge CoreObject(o) \longrightarrow DirectDensityReachable(o,p) \quad \ldots(2.3)$$

e. Definition 5: Density-reachable

The fifth definition believes that an object *o* is density reachable from *p*, that if there is a path between object *o* and *p* with minimum distance of path, it is then *Eps* and *MinPts* of each object and traveled is satisfied.

$$DDR(o,q) \wedge DDR(q,p) \wedge CoreObject(q) \longrightarrow DensityReachable(o,p) \quad \ldots(2.4)$$

f. Definition 6: Density-connected

The sixth definition says that an object *o* is a density that is connected to object *p*, and believes that if there is an object *q*, such density will be reachable from both *o* and *p*.

$$DensityReachable(o,q) \wedge DensityReachable(p,q) \longrightarrow DensityConnected(o,p) \quad \ldots(2.5)$$

g. Definition 7: Density-based cluster

The seventh definition believes that a density-based cluster is a cluster where an object is a core object, a neighbor of core object, or density-reachable of a core object.

$$CoreObject(o) \vee (Neighbor(p,o) \wedge CoreObject(p)) \vee (DensityReachable(p,o) \wedge CoreObject(p)) \longrightarrow o \in C \quad \ldots(2.6)$$

The argument here is that if object *o* is a member of cluster *C*, and if object *p* is density-reachable from *o*, then object *p* is also a member of cluster *C*.

$$(o \in C) \wedge DensityReachable(o,p) \longrightarrow (p \in C) \quad \ldots(2.7)$$

h. Definition 8: Border object

The eighth definition believes that a border object is not a core object but has a number of neighbors less than *MinPts* and still density reachable from a core object. Hence, object *o* is a border object if an object is not a core object and it is density-reachable from object *p* and object *p* is a core object.

$$\sim CoreObject(o) \wedge DensityReachable(p,o) \wedge CoreObject(p) \longrightarrow BorderObject(o) \quad \ldots(2.8)$$

ST-DBSCAN uses definitions 1 to 8 as described in its algorithm. Figure 2.2 shows the original ST-DBSCAN algorithm (Birant & Kut 2007). In general, ST-DBSCAN algorithm process is described below as follows:

1) Select the first object *o* in database

2) Search neighbor of *o* by calculating distance between *o* and other object in database. But if the distance satisfies radius (*Eps1* and *Eps2*), then set this object as neighbor of *o*.

3) If *o* has number of neighbor larger than *MinPts*, then set *o* as core object.

4) Since *o* is a core object, create new cluster and add all neighbors of *o* as members of the same cluster.

5) For all neighbor of *o*, search their neighbor that satisfy radius (*Eps1* and *Eps2*) and add this neighbor to the same cluster.

6) Repeat the process until all object in database are labeled with Cluster ID.

For example, object O (*s1, s2, t1, t2*) and P (*s3, s4, t3, t4*) has 4 variables, latitude, longitude, air temperature, and sea surface temperature respectively. Therefore, in order to calculate the distance between O and P, there is need to calculate *spatial_dist* and *var_dist* using Euclidean distance, thus the formula would be:

$$spatial\_dist = \sqrt{(s1 - s3)^2 + (s2 - s4)^2} \quad \ldots(2.9)$$

$$var\_dist = \sqrt{(t1 - t3)^2 + (t2 - t4)^2} \quad \ldots(2.10)$$

```
Algorithm ST_DBSCAN (D, Eps1, Eps2, MinPts, Δε)
   // Inputs:
       // D={o1, o2, ..., on}  Set of objects
       // Eps1 : Maximum geographical coordinate (spatial) distance value.
       // Eps2 : Maximum non-spatial distance value.
       // MinPts : Minimum number of points within Eps1 and Eps2 distance.
       // Δε : Threshold value to be included in a cluster.
   // Output:
       // C={C1, C2, … Ck} Set of clusters

 Cluster_Label = 0

 For i=1 to n                                        //(i)
   If oi is not in a cluster Then                    //(ii)
      X=Retrieve_Neighbors(oi , Eps1, Eps2)          //(iii)

      If |X| < MinPts Then
            Mark oi as noise                         //(iv)
      Else                              //construct a new cluster (v)
         Cluster_Label = Cluster_Label + 1

         For j=1 to |X|
           Mark all objects in X with current Cluster_Label
         End For

         Push(all objects in X)                      //(vi)

         While not IsEpmty()
            CurrentObj = Pop()
            Y= Retrieve_Neighbors(CurrentObj, Eps1, Eps2)

            If |Y| >= MinPts Then
               ForAll objects o in Y                 //(vii)
                  If (o is not marked as noise or it is not in a cluster) and
                     |Cluster_Avg() - o.Value| <= Δε Then
                     Mark o with current Cluster_Label
                     Push(o)
                  End If
               End For
            End If
         End While
      End If
   End If
 End For
End Algorithm
```

Figure 2.2 ST-DBSCAN algorithm

2.6 TIME SERIES

The focus of this research is to add temporal part of data into consideration when performing spatio-temporal clustering. As mentioned before, in order to incorporate time aspect of data into clustering, the addition of maximum temporal distance is introduced to set the time boundary of object in spatio-temporal data. Thus, some basic knowledge of time series is explained in this section.

Time series is a collection of observation recorded chronologically. Its specific characteristics include the large size of data, high dimensionality, and constantly updated data. In time series, data are analyzed as a whole rather than as individual values (Fu 2011).

Generally, time series studies have been involved only in areas such as representing and indexing, time series similarity measure, segmentation, visualization, and recently data mining (Fu 2011). But there has not been attempt towards the area in which this research is focusing, which are more on time series similarity measure and data mining.

Similarity measure in time series means focusing towards the way of comparing two time series and quantifying their similarity value. Similarity comparison can be performed in two ways; (i) comparing the whole sequence matching and (ii) comparing subsequence matching (Fu 2011).

The first one means comparing a pattern to the whole length of timeline. In this comparison, several techniques are considered as whole matching, these include using Euclidean Distance to compare Discrete Fourier Transform coefficient (Agrawal et al. 1993), computational geometry (Goldin & Kanellakis 1995), generalized Markov Model (Ge & Smyth 2000), Dynamic Time Warping (Berndt & Clifford 1994), Longest Common Sequence (Vlachos et al. 2002), and Bag of Pattern (Lin & Li 2009).

Another technique could be subsequence matching, which involves comparing a pattern of time series data to segments of timeline. This subsequence matching techniques include DualMatch (Moon et al. 2001), GeneralMatch (Moon et al. 2002), Using Sequence of Linear Segments (Morinaka et al. 2001), Hierarchical Similarity Search (Li et al. 1996) and so on. In order to performe subsequence matching technique, segmentation process of the time series data is required. This is because the additional process will increase the execution time of subsequence matching technique. As a result, the amount of time for conducting subsequence matching technique will also be increased in exchange of accuracy and precision of similarity values.

Therefore, in this research, the technique used for similarity comparison is the whole sequence matching using Dynamic Time Warping (DTW). This technique is selected because it is an extremely efficient and elastic similarity measure that is able to deal with temporal shifting and has better accuracy than Euclidean distance (Kleist 2015). It is also easy to understand and commonly available in several programming libraries in data mining area.

2.7 FAST DYNAMIC TIME WARPING (FASTDTW)

Also in this study, in order to incorporate temporal aspect into ST-HDBSCAN, similarity between time series of clusters is compared using Fast Dynamic Time Warping (FastDTW) (Salvador & Chan 2007). Therefore, when the cluster hierarchy is formed, the similarity distance that would be used is FastDTW. The scenario is that, if a time series of two clusters is similar when calculated using FastDTW, then this cluster can be grouped into one larger cluster hierarchically.

Dynamic Time Warping (DTW) is one of commonly used formula that could intuitively calculate similarity between two time series through ignoring the global and local shifts that appear in the time series. Thus, if two time series are almost similar but shifted, Dynamic Time Warping would still consider them as the same. Although, this is not possible if the similarity function used is Euclidean distance.

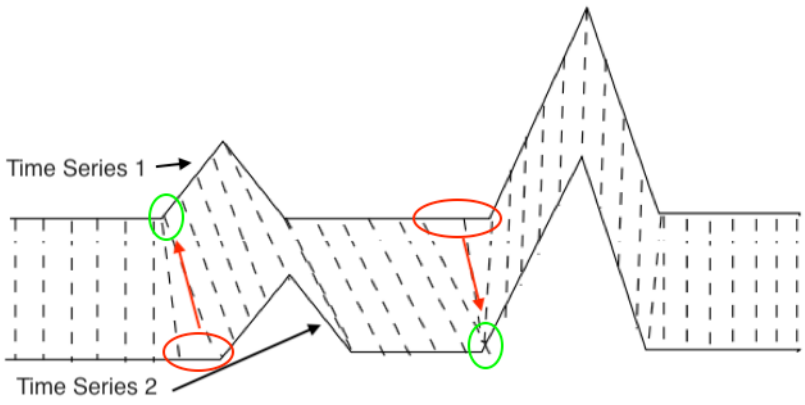

Figure 2.3 Dynamic time warping illustration of two time series with warping value

DTW is able to recognize similarity even though there is a shifting in time series because the time series is mapped into a warp path, where it is possible to map several points of a time series and warp it into one point of other time series (Figure 2.3). There are two time series as shown in Figure 2.3, Time Series 1 and Time Series 2. The first example shows that several points in Time Series 2 (in red circle on the left) are warped into one point of Time Series 1 (green circle on the left). While in the second example, the figure shows several points in Time Series 1 (in red circle on the right) are warped into one point in Time Series 2 (green circle on the right).

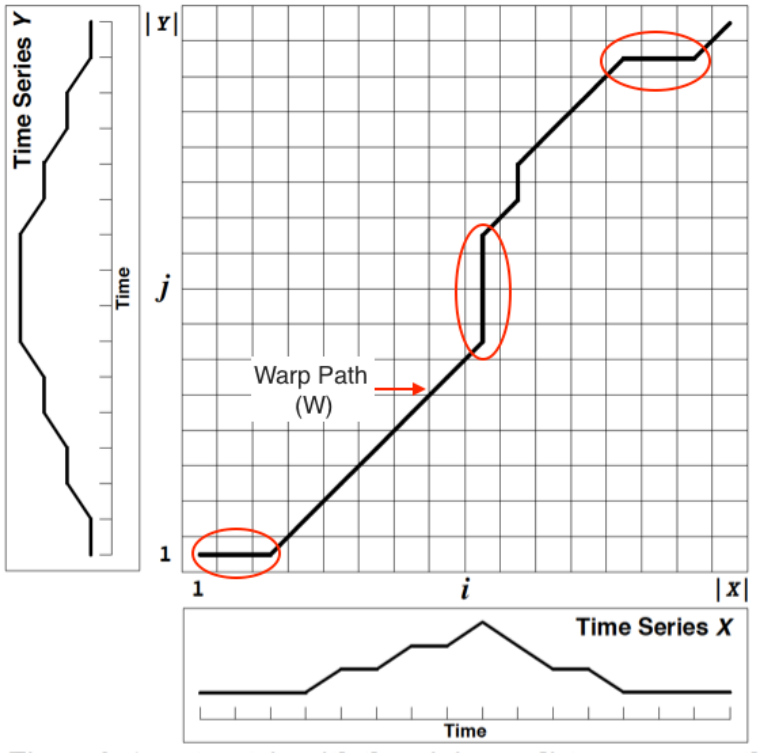

Figure 2.4 Example of warp path between two time series

Source : Salvador & Chan (2007)

Another example is when considering two time series dataset of $X = (x_1, x_2, .. x_i)$ and $Y = (y_1, y_2, y_j)$ where i and j are not in a similar length as shown in Figure 2.4. The warp path of the two time series would be $W = (w_1, w_2, \ldots w_k)$ where $w_1 = (x_1,y_1)$, $w_2=(x_2,y_2)$, *and* $w_k = (x_i,y_j)$. The k index value depended on the number of i , j and whether there is a point that could be warped into single point or not. Hence if there is a warping, the k value would be larger than i or j.

Figure 2.4 example shows that the warp process is indicated by red circle, which is the first red circle on the left corner. Also, the first three data in Time Series *X* is warped into 1 data of *Y*. While in the second circle (the middle), four data in Time Series *Y* are warped into 1 data of *Y*.

Additionally, as shown in formula 2.11 below, *D* is the minimum distance warp path between point *x* and *y* calculation using Euclidean distance or other distance measure (Salvador & Chan 2007). This formula (*D*) is used in order to calculate the minimum distance warp path between point *x* and *y*,

$$D(i, j) = Dist(i, j) + \min[D(i\text{-}1, j), D(i, j\text{-}1), D(i-1, j\text{-}1)] \quad \ldots(2.11)$$

Where:

D(i,j) : minimum distance warp path of point x_i and y_j

Dist(i,j) : distance between point x_i and y_j, calculation using distance measure

2.8 LINKAGE WITH WARD'S METHOD

In this research, the ST-DBSCAN algorithm was combined with hierarchical algorithm in order to produce a more advanced spatio-temporal algorithm that can be corporated to not only spatial and non-spatio-temporal but also temporal aspect of the data. As earlier mentioned, ST-OPITCS has already performed the process of combining density and hierarchical-based spatio-temporal data clustering. The algorithm has produced better cluster performance indices compared to ST-DBSCAN (Agrawal et al. 2016). However, ST-OPTICS algorithm does not take into account the temporal part of data.

Hierarchical algorithm on the other hand involves a process called linkage. This is a process to create hierarchical cluster from raw data or several clusters by grouping similar data or cluster into a larger one that will be in levels based on some similarity distance.

Also, in order to create linkage among cluster, the linkage process will be used to calculate similarity distance between clusters that have not been included in

hierarchy. Therefore, if two clusters are similar, they are combined into a new larger one and added to the hierarchy, in which these two combined clusters are removed from the hierarchy. Thus, the new one replaced the combined cluster in the hierarchy. The effect is that, the algorithm will stop when all of the clusters are combined into the largest one thereby becomes the root of the hierarchy.

There are several linkage methods that can be used to generate hierarchical clusters, namely; single, complete, average, weighted, centroid, median, and ward (SciPy n.d.). In this research, the method used is Ward variance minimization method because it is considered as one of the widely used method to generate hierarchical cluster (Murtagh & Legendre 2014). This is described in the formula 2.12 and 2.13 respectively.

$$d(u,v) = \sqrt{\frac{|v|+|s|}{T}d(v,s)^2 + \frac{|v|+|t|}{T}d(v,t)^2 + \frac{|v|}{T}d(s,t)^2} \quad \ldots(2.12)$$

$$T = |v| + |s| + |t| \quad \ldots(2.13)$$

where

u: is the new developed cluster that contains cluster *s* and *t*

v: is the random cluster that have not been joined with any cluster

s: is the cluster that has been joined into cluster *u*

t: is another cluster that has been joined into cluster *u*

|*v*|: is the cardinality or number of object in cluster *v*

Thus if *d(u,v)* is small, it indicates that cluster *v* is similar to other cluster in that cluster *u* and *v* could be joined to make cluster *u*. If *d(u,v*) is larger, the effect is that *v* will not be joined into cluster *u*. Thus, *v* will be joined into cluster that has minimum value of *d(u,v)*.

2.9 EVALUATION INDICES

In order to compare the performance of the developed algorithm, several evaluation indices are calculated. This is because, not all the indices are suitable for calculating performance of spatio-temporal clustering algorithm. Agrawal (2016) performed a study to determine whether an index could be used. The study selected 11 evaluation indices, such as Ball-Hall, Det Ratio, Dunn, Gamma, G+, GDI31, GDI51, Ksq Det W, Log Det Ratio, Point Biserial, Tau and Trace W. However, in this research, only 6 indices are selected, namely; Ball-Hall, Det Ratio, GDI51, Ksq Det W, Log Det Ratio, and Trace W. This is because after performing the calculation, the other indices do not provide any result (the indices calculation result is 0) or took too long to calculate. Therefore, those indices are omitted in this research.

Generally, there are two methods to calculate the mining accuracy for clustering algorithm (extrinsic method and intrinsic method). Extrinsic method is possible only if the ground truth or number of cluster and cluster membership predicted by human expert is available. This is because; clustering is an unsupervised algorithm that does not have number of class or cluster availability, in order to compare and check whether the data are actually grouped into the correct cluster, while the intrinsic method is used when human expert opinion is not available. This method evaluates how intense the density of a cluster is (Jiawei et al. 2012).

In order to calculate these indices values, clusterCrit package in RStudio is used in this research. The definition of each algorithm is based on clusterCrit documentation (Desgraupes 2013).

2.9.1 Ball-Hall

Ball-Hall index is one of the index that can be used to calculate performance of clustering for spatio-temporal data (Ball & Hall 1967). The mean dispersion of a cluster is the mean of the squared distances of the points of the cluster with respect to their barycenter. Invariably, the Ball-Hall index is the mean, through all the clusters of their mean dispersion: