uOttawa

L'Université canadienne
Canada's university

An Improved Density-Based Clustering Algorithm Using Gravity and Aging Approaches

By

Fadwa Gamal Mohammed Al-Azab

Thesis Submitted to the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electronic Business Technologies
(E-Technology Stream)

Supervisor: Dr. Bijan Raahemi

University of Ottawa

Ottawa, Ontario

January 2015

In the name of God,

the compassionate and the merciful

ABSTRACT

Density-based clustering is one of the well-known algorithms focusing on grouping samples according to their densities. In the existing density-based clustering algorithms, samples are clustered according to the total number of points within the radius of the defined dense region. This method of determining density, however, provides little knowledge about the similarities among points. Additionally, they are not flexible enough to deal with dynamic data that changes over time. The current study addresses these challenges by proposing a new approach that incorporates new measures to evaluate the attributes similarities while clustering incoming samples rather than considering only the total number of points within a radius. The new approach is developed based on the notion of ***Gravity*** where incoming samples are clustered according to the force of their neighbouring samples. The Mass (density) of a cluster is measured using various approaches including the number of neighbouring samples and Silhouette measure. Then, the neighbouring sample with the highest force is the one that pulls in the new incoming sample to be part of that cluster. Taking into account the attribute similarities of points provides more information by accurately defining the dense regions around the incoming samples. Also, it determines the best neighbourhood to which the new sample belongs. In addition, the proposed algorithm introduces a new approach to utilize the memory efficiently. It forms clusters with different shapes over time when dealing with dynamic data. This approach, called ***Aging***, enables the proposed algorithm to utilize the memory efficiently by removing points that are aged if they do not participate in clustering incoming samples, and consequently, changing the shapes of the clusters incrementally.

Four experiments are conducted in this study to evaluate the performance of the proposed algorithm. The performance and effectiveness of the proposed algorithm are validated on a synthetic dataset (to visualize the changes of the clusters' shapes over time), as well as real datasets. The experimental results confirm that the proposed algorithm is improved in terms of the performance measures including *Dunn Index* and *SD Index*. The experimental results also demonstrate that the proposed algorithm utilizes less memory, with the ability to form clusters with arbitrary shapes that are changeable over time.

ACKNOWLEDGEMENT

First of all, whatever blessings and achievements I am achieving is only from ***Allah***. All graces and thanks are due to him. Then, Prophet Mohammed peace be upon him the one who taught us that education is the most valuable thing in life.

I would like to express my deep feelings of appreciation and gratefulness to my supervisor, Professor. Dr. *Bijan Raahemi* for his guidance and help throughout this thesis work. This work would have not been done successfully without the unlimited support of Dr. *Mahdi Mohammadi*, and I am so grateful for each moment he spent teaching me many thing in data mining. I sincerely thank them for their kind support and guidance. Also, I would like to thank my dear colleagues at the KDD lab, I am very thankful for each moment we spent it together.

I would like to express the deepest appreciation and love to my beloved family, to whom I owe this achievement and without their support it would not have been accomplished. *Faiza* and *Gamal*, my parents, I am so pleased of having you, and I am grateful for each moment you spent teaching me that education comes first. You have sacrificed a lot to make me the person I am today. I would also like to thank my sisters (*Al-Shamma*, *Isra*, and *Zannobia*) and my brother *Mohammed*, who have always supported and encouraged me through my study. Special thanks to my beloved aunties *Entesar* and *Jameilah* (my second mothers) who always take care of me as their daughter. "Once again thank you to my beloved family for keeping me in your prayers, cheering me up, and stood by me through the good and bad time. I love you so much".

I also would like to express my warm thanks to my cousin *Gassan Al-Kibsi*. I am so thankful to you for encouraging and motivating me all the way to keep moving forward and the support you have granted me to pursue my education.

Finally, I would like to thank all my close relatives and my beloved friends who supported and motivated me with their wishes.

I am very fortunate for having all of you in my life.

"Thanks to ***Allah*** and may He bless you all***"***

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DM	:	Data Mining
DSR	:	Design Science Research Methodology
DBCSAN	:	Density-Based Spatial Clustering of Application with Noise
OPTICS	:	Ordering Points to Identify the Clustering Structure
GDBSCAN	:	Generalizing the Density-Based Algorithm DBSCAN
DENCLUE	:	DENsity-based CLUstEring
IL	:	Incremental Learning
PM	:	Performance Measures
DBNPR Algorithm	:	Density Computation using the Total Number of Points within Radius
GrDBNPR Algorithm	:	Gravity-Based Density Computation using the Total Number of Points within Radius
GrDBSil Algorithm	:	Gravity-Based Density Computation using the Silhouette Measure
GrDBAveD Algorithm	:	Gravity-Based Density Computation using Average Distance
Improved GrDBAveD		Gravity-Based Density Computation using Average Distance with some improvements in terms of evaluating the neighbours of the new incoming points
DBGrvAge Algorithm	:	An Improved Density-Based Clustering Algorithm using Gravity and Aging Approaches

"Understanding our world requires conceptualizing the similarities and differences between the entities that compose it."
(Tryon & Bailey, 1970)

CHAPTER 1. INTRODUCTION

In this new era of information technology, numerous data is generated by many applications carrying valuable information that needs to be analyzed (e.g. using data mining techniques) to extract their meaningful patterns. Clustering and classification are amongst the most popular data mining techniques. During the last decade, many studies have focused on the different methods of clustering algorithms, because of their significant role in allowing automatic identification of unlabeled records by grouping them into ***clusters*** based on similarity measurements (Han, Kamber, & Pei, 2011). Clustering methods are widely used in many applications including network, business and medical based applications. Figure 1-1 lists the variety of applications where clustering methods play a significant role (Han et al., 2011; Schaeffer, 2007).

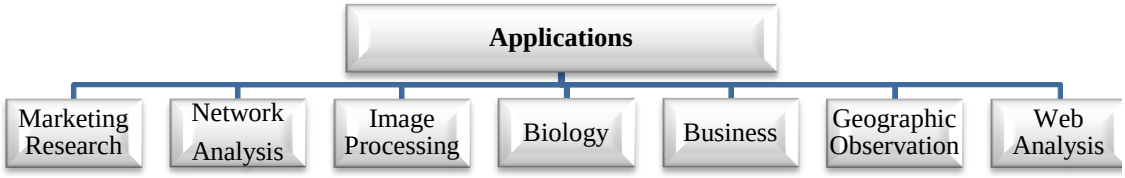

Figure 1-1: Application Areas of Clustering Methods

Clustering in data mining consists of many different algorithms including partitioning, hierarchical, grid, density, graph and model-based algorithms. Among these clustering methods, the density-based clustering is one of the well-known techniques mainly focusing on: 1) Minimizing the number of input parameters; 2) Discovering clusters with arbitrary shapes; 3) Clustering large data efficiently; 4) No need for a prior knowledge of the number of cluster; and 5) Handling noise (Parimala, Lopez, & Senthilkumar, 2011).

In this study, we propose a density-based clustering algorithm using gravity and aging approaches, which can be part of an incremental solution for clustering. By introducing the Gravity and Aging approaches, we present a density-based clustering algorithm that cluster dynamic data based on density and distance measurements. The ***Gravity*** approach focuses on the attribute similarities among samples when clustering

new incoming points. Also, it measures how distance affects the densities of neighbours. The other proposed concept, the ***Aging*** approach, performs the following tasks: 1) when clustering new incoming samples, the aging approach keeps track of the samples that participate in the clustering; 2) it deletes samples that are less frequently used over time; and 3) it enables the proposed algorithm to create clusters of *arbitrary shapes* (clusters of any shape such as "S" shape rather than round shape (Han et al., 2011)) and changeable over time.

The rest of this chapter is organized as follows: research motivation, problem statement, objectives, and scopes of the study which are discussed in Sections 1.1, 1.2, 1.3 and 1.4, respectively. The research methodology applied to the design, development, and the evaluation of the proposed algorithms is discussed in Section 1.5 in details, followed by Section 1.6, which highlights the contributions of the research. Finally, Section 1.7 provides a brief description of the following chapters of the thesis.

1.1 Research Motivation

Density-based clustering method is one of the well-known clustering algorithms. It is known for its ability to form clusters with arbitrary shapes, handle noise, and does not rely on a prior knowledge of the number of clusters. This type of algorithm is designed based on the notion of ***density*** (Han et al., 2011), by which samples are clustered based on the density of the regions. The dense regions are identified according to a density measure. For example, in some of the existing algorithms including DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) and OPTICS (Ankerst, Breunig, Kriegel, & Sander, 1999) density is measured by counting the number of points within a specific radius of a core-point. If the number of points within the radius exceeds a threshold, then the core-point becomes part of a particular cluster. Otherwise, it is classified as a noise or border-point (Ester et al., 1996).

The motivation of conducting this study is to investigate new measures that consider the attribute similarities among points. Attribute similarity is one of the fundamental concepts in Data Mining, which is used to find patterns in data by describing how points are far from each other by taking into account all the attributes of the data point (Ronkainen, 1998). By studying the attribute similarities among points, while measuring density, provides more information about how similar the points are, indicated

as by the distance (the smaller the distance among points, the more similar they are). However, attribute similarity is not taken into account in the existing density-based clustering algorithms including DBCSAN and OPTICS algorithms, which depend on counting the total number of points within a radius.

Furthermore, these new measurements may evaluate the density of the neighbourhood more accurately by examining how dense the neighbourhood of the identified neighbours of the new incoming points are, especially those neighbours that belong to different clusters. In addition, the algorithm should be able to decide which cluster pulls in the new incoming point, when there are variant densities of neighbourhoods. For instance, if there are several scenarios such as farthest neighbour with higher density or nearest neighbour with lower density and vice versa. For that, distance is employed to determine which cluster pulls in the new points since the distance has an impact on the density of the neighbourhood. Here, the Newton's law of gravity is adopted as it considers both factors: the masses (densities) of objects over their distances.

Additionally, we are motivated to improve the density-based clustering algorithm to cluster dynamic data. For that, we will introduce the gravity approach to insert new incoming points based on the neighbourhood with the higher force of gravity to deal with dynamic data call for algorithms that require as little memory as possible due to the large number of arriving data. Hence, this part of the study focuses on a new approach for incremental learning to be integrated into the proposed algorithm to limit the memory usage by removing samples that are less frequently used over time.

1.2 Problem Statement

Density-based clustering algorithms are one of the primary methods in clustering that have an advantage of creating clusters with arbitrary shapes defined by regions with high density. These regions are separated from each other by low-density regions that are essential to handle clusters with different sizes (Ashour & Sunoallah, 2011). Forming clusters in density-based clustering method mostly depends on the ***density*** measure that allows the algorithms to search for groups of samples that are dense (Tan, Steinbach, & Kumar, 2006).

Some of the algorithms of the density-based clustering method consider the minimum number of samples within the radius of a point to measure how dense its

neighbourhood is, in order to create a new cluster, or expand an existing one. For example, DBSCAN forms clusters based on the density-based connectivity. It considers a neighbourhood distance by specifying a radius. Then, it checks how many samples are distributed within that radius of the point. If the number of samples exceeds the threshold denoted by **"*MinPts*",** then the point is part of a cluster. Otherwise, the point is classified as a noise or a border-point (Ester et al., 1996).

However, density-based clustering method does focus mainly on the ***MinPts*** within the radius to measure how dense is the point's neighbourhood. By only considering the minimum number of points within the radius, it does not produce enough knowledge about how those points within the radius are similar to each other. In order to examine the similarities among points, new measurements need to be accommodated to define how dense is the neighbourhood of the point, for instance, calculating the distance among points. This criterion is extremely important in measuring the attributes similarities of the points to determine which cluster the samples belong to. This study intends to investigate new measures to define density and to study the attributes similarities among points in order to enhance the performance of the density-based clustering method.

Furthermore, the existing density-based clustering algorithms are not flexible enough to deal with a dynamic environment since they cluster the static dataset all at once. For that reason, this study investigates new methods of incremental learning (Aging approach) to enable the algorithm to work in a dynamic environment while considering memory usage.

1.3 Objectives

The objective of this study is to improve the performance of the density-based clustering method by investigating new measures besides the existing ones in order to group points into their proper clusters. Also, the proposed algorithm has to be able to deal with dynamic data by introducing new incremental learning methods.

1.3.1 Main Key Objectives

- Proposing new measures to study the similarities among points within the radius while grouping the new samples in order to improve the clustering performance of the enhanced density-based algorithm.

- Forming clusters with arbitrary shapes (clusters of any shape such as "S" shape rather than round shape) and changeable over time while maintaining the performance of the proposed algorithm.
- Maintaining a low computational complexity in terms of the running time and memory utilization of the proposed algorithm.
- Incorporating incremental learning, and in particular, the Aging approach to deal with dynamic data.

1.4 Scope of the Research

The focus of this study is mainly on grouping new incoming points into the existing clusters, and using the memory efficiently, specifically focusing on the testing phase. As for the training phase, we assume that the DBSCAN algorithm forms several clusters. The following are the list of the tasks to be accomplished.

- Improving the way the new samples are grouped into the clusters based on density, distance, and other criteria that might be needed to be incorporated with the gravity concept.
- Introducing new incremental learning methods, and specifically, the Aging approach, to deal with the dynamic environment while considering the memory usage.
- Validating the proposed algorithms on several datasets.
- Comparing the result of the proposed algorithms with other well-known algorithms (DBSCAN and K-means) using performance measures (PM) to highlight how the proposed algorithm has improved the performance compared with the existing algorithms.
- Applying the proposed algorithm on a real dataset, namely peer-to-peer (P2P) network traffic captured from the Internet traffic.

1.5 Research Methodology

The research methodology is a systematic way of solving a particular problem. It describes the general research strategy that outlines how the research is undertaken starting from problem formalization and ending up with the delivery of the research. The chosen methodology for this study is the ***Design Science Research Model***, described in details in Section 1.5.1. In addition, the research methodology takes a role in identifying

the methods needed in the study to achieve the study's objectives. As for this study, the method is ***Computer Simulation***, details in Section 1.5.2.

1.5.1 Design Science Research Methodology

Design Science Research (DSR) is the methodology used to design, develop, evaluate, and deliver the proposed algorithm to solve the stated problem in Section 1.1. The problem is identified either from the existing gaps in the literature view or identified by the organization (Vaishnavi, 2008). For this study, the DSR strategy is chosen to illustrate the main activities that contribute in identifying the research problem through the investigation of existing knowledge to discover the gaps. This study is more onto an improvement type research, where a new solution is implemented to improve a density-based clustering algorithm based on the identified problem (gaps) from the literature review. The DSR methodology follows the general DSR cycle described by Peffers et al., (2007) which is shown in Figure 1-2. As demonstrated in Figure 1-2, the DSR model consists of six activities which are: 1) Identify Problem and Motivation; 2) Define the Objectives of the Solution; 3) Design and Development; 4) Demonstration; 5) Evaluation; and 6) Communication.

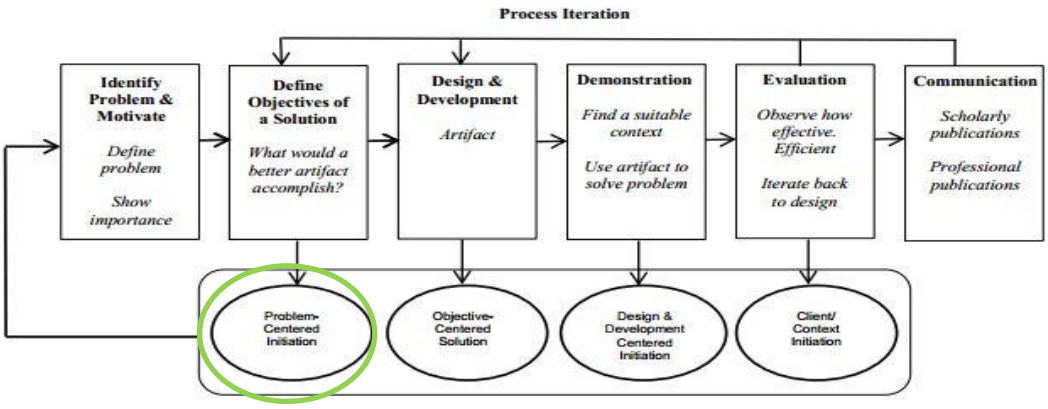

Figure 1-2: Design Science Research Model (DSR) (Peffers et al., 2007)

The sequential order of these activities depends on the researcher. If the study is objective-centered solution then the process begins with the activity (Design and Development), see Figure 1-2. As for this study, it is a problem-centered initiation that starts with the activity one (Identify Problem and Motivation) and it goes through all the activities until the activity six (Communication) in sequence order, see Figure 1-2. The description of each activity involved in the study is described below.

a) Identify Problem & Motivation

In this activity, the research problem is identified by reviewing the existing works in clustering algorithms, in particular the density-based clustering method which is the main focus of the study. Also, we review the incremental learning process, factors, and its tasks along with several incremental clustering algorithms. Furthermore, the gravity concept is discussed with some examples on how it is been adopted in data mining. Through the literature review, several problems are identified based on a number of requirements that are usually used to evaluate the clustering algorithms, and based on these, the study focuses on providing a justified solution for the selected problems from the identified gaps (see Section 1.2 for more details about the problem statement, and Chapter 2 for more details about the literature review).

b) Define Objectives

Based on the problem definition and the questions that are identified earlier in the previous step, the objectives are inferred from the problem in order to focus on providing a solution (see Section 1.3 for more details about the objectives of the study).

c) Design and Development

The design and development activity discusses the design process model of the proposed algorithm to solve the highlighted problems. The design of the proposed algorithm focuses on introducing new measures to enhance the performance of the density-based algorithm by studying the attribute similarities among points while clustering a new incoming sample, and how to make it incremental to cluster dynamic data. Also, this activity includes the description of the methods used to improve the proposed algorithm and how it works (see Chapter 3 for more details about the architecture of the proposed algorithm).

d) Demonstration

In the demonstration activity, the proposed algorithm is validated on a synthetic dataset (to visualize the changes of the clusters' shapes over time), and on several real datasets in order to show how well it performs in solving the stated problem. It includes the simulation and experimentation of the algorithm on the datasets.

e) Evaluation

In the evaluation activity, the proposed algorithm is assessed in order to see how well the proposed algorithm performs in solving the problem. It is done by comparing the results of the newly proposed algorithm with well-known density-based clustering algorithm including (DBSCAN and K-means algorithms) using the performance measures (PM) including Dunn Index and SD Index (see Section 4.1.1 for more details). Also, it is important to identify the requirements that clarify precisely the functionalities of the proposed algorithm. Some iteration might need to be done back to the activity (Design and Development) to improve the design and the development of the solution. Otherwise, we can proceed with last activity (Communication).

f) Communication

Once all the previous phases are completed and approved by the supervisor, the communication activity takes place by completing the following: report submission, thesis defense, and publishing paper in one of the well-known conferences or journals.

1.5.2 Research Method

The method used to build the proposed algorithm of the study is **Simulation**. Simulation runs on a single computer to examine the behaviour of the algorithm for a particular application. The simulation type selected for this study is ***Continues-Dynamic Simulation***[1], and it is conducted in ***Matlab*** version R2013a. The reason for choosing the continues-dynamic simulation is because this proposed algorithm has to be able to deal with dynamic data, and it needs to adjust itself over time when there is a new incoming sample that needs to be clustered.

1.5.3 Research Method and Steps

Based on the DSR methodology, the activities are performed through five phases; the details are presented in Table 1-1. First, the problem is identified through the Literature Review phase. Then the objectives desired to solve the problem are discussed in the second phase (Define Objectives of the Solution). The third phase (Designing the Proposed Algorithm) proposes the methods used to improve the algorithm. The fourth phase (Modeling, Implementation, and Validation) discusses the activities four and five

[1]https://www.extendsim.com/sols_simoverview.html#continuous

which are Demonstration and Evaluation of the proposed algorithm. Finally, the Communication activity is completed in the Knowledge Dissemination phase which provides the final delivery of the study.

Table 1-1: Research Phases and Activities

Phases	Activities
1 **Literature Survey** **Chapter 2)**	• Brief description of the different clustering methods with their algorithms. • Detailed description of the density-based clustering algorithms: ➢ Brief description of the well-known density-based clustering algorithms. ➢ Comparison of the density-based clustering algorithm to identify the gaps based on the list of requirements. • Brief descriptions of the incremental learning process: ➢ What is Incremental Learning (IL)? ➢ The importance of IL. ➢ Factors leading to integrate IL. ➢ Main tasks of IL. ➢ Existing Incremental Clustering Algorithms. • Brief description of the Newton's Law of Gravity: ➢ The notion of gravity. ➢ Existing clustering algorithms that adopt the gravity theory.
2 **Define the Objectives of the Solution** **(Section 1.3)**	• Investigating new measures to study the similarities among points within the radius while grouping samples, to improve the performance in terms of the performance measures, memory usage, and forming clusters with arbitrary shapes. • Incorporating incremental learning approach to deal with dynamic data generated by real-time applications.
3 **Designing the Proposed Algorithm**	• Studying the general structure of proposed algorithms along with its architecture and definitions employed to fulfill the objectives of the study. Also, introducing new

Phases	Activities
Chapter 3	approaches to make the proposed algorithm incremental using the ***Gravity*** concept and ***Aging*** strategy along with the Pseudo Code.
4 Modeling, Implementation, and validation **Chapter 4**	• The implementation of the algorithm is simulated using ***Matlab*** where the proposed algorithm is implemented, validated, and the results are evaluated based on: ➢ Improving the algorithm by considering new measures to study the attributes similarities among points. ➢ Validating the improved algorithm on synthetic dataset (to visualize the clusters with arbitrary shapes) and on real datasets. ➢ Evaluating the performance of the improved algorithm in terms of the performance measures, memory usage, running time, and clusters with arbitrary shapes. ➢ Highlighting the fulfilled requirements to indicate the contribution and the limitations of the study.
5 Knowledge Dissemination	• Publishing the resulted knowledge in conference, workshops, or journals. • Concluding and writing the thesis. • Thesis defense.

1.6 Contributions of the Thesis

We propose a density-based clustering algorithm that can be part of an incremental learning solution that incorporates the following methods:

- **Newton's Law of Gravity** is adopted to enable the density-based clustering algorithm to perform the following:
  - It studies the attribute similarities among points when clustering new incoming samples.
  - It inserts new samples into clusters while considering the variant densities (i.e. Number of points within a radius, Silhouette measure, and Average distance) and how the distance between the new incoming sample and its neighbours affects the force of gravity of those variant densities of the neighbours' neighbourhoods.
  - The gravity concept makes the proposed algorithm to be incremental by fulfilling the ***insertion task***.
- Introducing the **Aging strategy** that performs the following:
  - It enables the proposed algorithm to learn from the changes happening to the clusters over time by keeping track of the points that participate in clustering new incoming samples.
  - It removes points that are less frequently involved in clustering new incoming samples. Thus, it works efficiently in minimizing the memory usage.
  - It creates clusters with arbitrary shapes and changeable over time.
  - Aging strategy makes the proposed algorithm to be incremental by fulfilling the ***deletion task***.

1.7 The Structure of the Thesis

This section presents a brief description of the primary structure and contents of each chapter.

Chapter 2

This chapter covers the fundamental concepts of the density-based clustering method since this study focuses on improving a density-based clustering algorithm. It is necessary to investigate several algorithms related to the study area to have a clear understanding of the algorithms and how they work. Besides, there are several requirements that need to be identified to evaluate the performance of the algorithms. In addition, it provides a brief description of the incremental learning factors, tasks, and discusses several incremental clustering algorithms. Another concept called Newton's Law of Gravity is discussed here, clarifying how it is involved in data mining, particularly in clustering.

Chapter 3

This chapter discusses the general structure of the proposed algorithms along with their architecture and definitions that are employed to fulfill the objectives of the study. The chapter also provides a description of the methods used to make the proposed algorithm to be part of an incremental solution using the **Gravity** and **Aging** approaches along with the Pseudo Code.

Chapter 4

In this chapter, the proposed algorithm "the density-based clustering algorithm based on Gravity and Aging **(**DBGrvAge algorithm**)"** is evaluated by conducting four different experiments. These experiments are evaluated by validating the algorithm on several datasets types (synthetic dataset, real datasets and dataset of real-time application). The proposed algorithm is compared against several well-known algorithms (DBSCAN and K-means). Then their results are assessed by the performance measures, memory usage, and running time as discussed earlier in this chapter. These experiments are conducted to evaluate how well the proposed algorithm performed.

Chapter 5

This chapter summarizes the main idea and achievements of the study by providing a brief description of the study's outcomes based on Chapter 4. It is followed by highlighting the main contributions of the study. The chapter concludes with identifying the limitations of the proposed algorithm that can open a new direction for future works.

CHAPTER 2. LITERATURE REVIEW

Data mining (DM) is one of the most important concepts in knowledge discovery, and it has a great deal of importance in many industries due to the massive amount of data coming from different applications and the need to understand and act upon them. Industries can benefit from the data by analyzing it to obtain new hidden patterns for better understanding of business to make decisions. Data Mining is the practice that extracts useful information from datasets. There are several techniques in data mining including ***Classification*** (supervised learning) and ***Clustering*** (unsupervised learning). Each of these techniques analyzes the data in a different manner. However, this study focuses on the clustering technique that clusters unlabeled dataset in particular the density-based clustering methods that are discussed in details in the following sections. This literature survey is divided into four parts which are: 1) Brief description of clustering techniques, its methods and measurements used for clustering; 2) In particular, discussing the density-based clustering method and its well-known algorithms; 3) Description of the incremental learning (IL) factors leading to use it, and reviewing several studies that incorporate IL; finally, 4) The definition of the Gravity Theory along with some related works that adopted it.

2.1 Clustering

Clustering is the method used to divide data into groups, where the objects within a group are similar to one another and dissimilar with objects in other groups according to certain similarity measurement (Berkhin, 2006). Clustering techniques are categorized into a number of methods which are partitioning, hierarchical, grid, density, graph and model-based algorithms. These methods are different from one another on the way clusters are formed. However, the performance of these clustering methods is evaluated based on the same primary requirements that are discussed in several studies most of which focus on: 1) Minimizing the number of input parameters; 2) Discovering clusters with arbitrary shapes; 3) Clustering large data efficiently; 4) Handling noise; and 5) No prior knowledge of the number of cluster (Parimala et al., 2011). Figure 2-1 lists the clustering methods along with their well-known algorithms (Han et al., 2011).

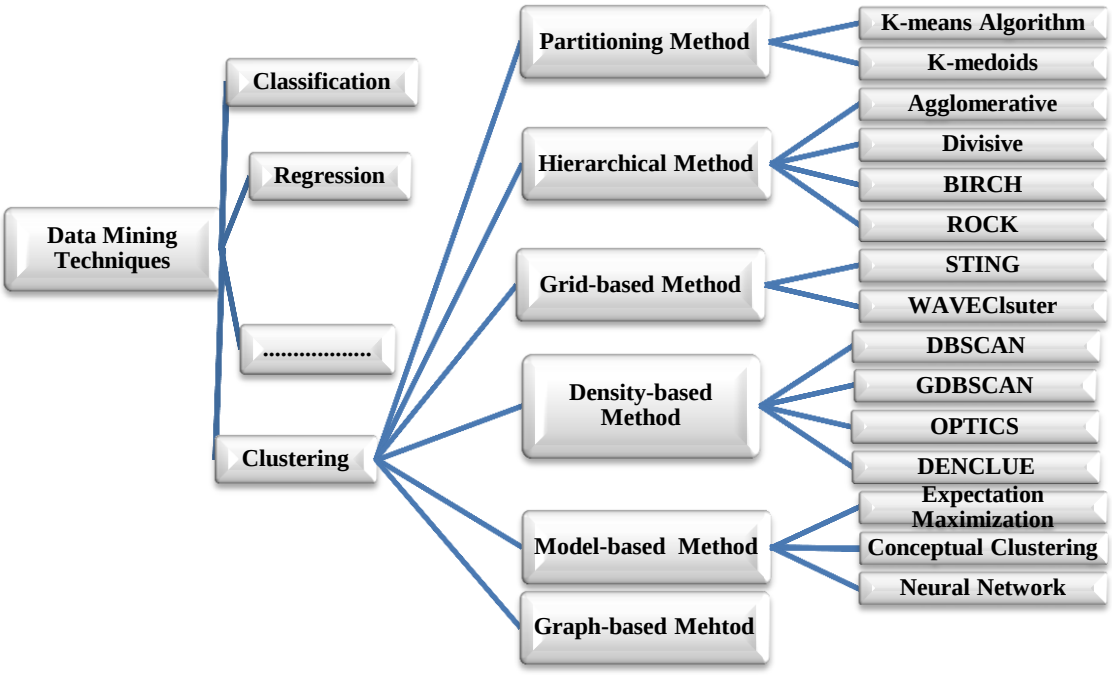

Figure 2-1: List of the Main Clustering Methods with their Well-known Algorithms

Each of those clustering methods highlighted in Figure 2-1 has its process for clustering data. The following section provides a brief description of how these methods work, along with their main algorithms, and more description of the density-based clustering method since it is the focus of this study.

2.1.1 Clustering Methods

Partitioning Clustering Method divides data into a set of clusters, where data is partitioned into several subsets by grouping samples into their closest center called "*centroid*"- this method called ***K-means*** algorithm (Berkhin, 2006). The other method is ***k-medoids*** algorithm; it replaces the means of the clusters with modes. ***K*** indicates the total number of clusters for both algorithms. The similarity measure used to group samples to its centroid is the distance measure. There are several types of distance measure including the Euclidean and Manhattan distance. The number of clusters needs to be known in advance in both algorithms K-means and K-medoids (Berkhin, 2006).

Hierarchical Clustering Method produces a multi-level clustering in which data is grouped into a sequence of partitions where the sub-cluster belongs to the super-cluster. Hierarchical clustering method consists of two algorithms are ***agglomerative*** "bottom-up" and ***divisive*** "top-down" (Schaeffer, 2007). The similarity or linkage measure used by the hierarchical method is the distance measurement. Distance is divided into four

measurements which are: 1) Minimum Euclidean distance between clusters called nearest-neighbour clustering algorithm; 2) Maximum Euclidean distance between clusters is called farthest-neighbour clustering algorithm; 3) Average distance; and 4) Mean distance (distance between centers) (Han et al., 2011). Another measure used in the hierarchical method is **cut-based measure** that measures the connectivity of the cluster with the rest of the clusters to determine when to split a cluster into several clusters (Berkhin, 2006; Schaeffer, 2007).

Grid-based Clustering Method quantizes the clustering space into a finite number of cells. Then a required operation is performed on the quantized space to detect which cell contains more number of points to determine whether the cell is dense or not (Ilango & Mohan, 2010). All dense cells are connected to each other to form clusters. According to the definition of the grid-based clustering method by Madhulatha (2012), the density measurement is used to determine how dense each cell is, by setting a threshold to determine how dense the cell is, and if the density of a cell exceeded the threshold then the cell forms a cluster.

Graph-based Clustering Method constructs a graph **G** by having a set of vertices **V** called nodes and a set of edges **E** that connect pairs of vertices. The primary task of the graph clustering method is to connect vertices with each other with edges to form clusters where there are many edges within a cluster and few between clusters (Schaeffer, 2007). To form clusters with vertices and edges, the measurements needed to connect vertices within a cluster and between clusters and these are: 1) Intra-density called *Relative Density* (connection within a cluster); 2) Inter-density called *Density-related* (connection between clusters); 3) k-nearest neighbour (K-nn) uses a distance measure such the *Euclidean Distance* (Lühr & Lazarescu, 2009). The cut-based measurements are used to determine when the clusters have to split into several clusters. There are two measurements to split a cluster which are; **Conductance** splits clusters based on the edge weight, and **Expansion** splits clusters based on the number of vertices (Z. Chen, 2010).

Density-Based Clustering Method is developed based on the density notion. This method is one of the most efficient methods for detecting clusters with arbitrary shapes and handling noise (Parimala et al., 2011). Section 2.2 discusses in depth the density-based clustering algorithms.

2.2 Density-based Clustering Algorithms

This study focuses on ***Density-Based Clustering Method***, which mainly depends on the notion of density. The algorithms in the density-based clustering method define their clusters based on the high dense regions separated by regions of low density (Cassisi, Ferro, Giugno, Pigola, & Pulvirenti, 2013). The clusters continue growing until the density exceeds a certain threshold (Cassisi et al., 2013). Based on that, the density-based clustering algorithm has an advantage in creating clusters with arbitrary shapes. Like any other algorithms, there is a major similarity measure needed to construct clusters that are *density* and *connectivity* that calculates the path between each pair of vertices using the distance measures (Schaeffer, 2007). Both are used to measure the local distribution of the nearest neighbours of points within a particular distance (Berkhin, 2006; Mann & Kaur, 2013).

1) ***Density-based Connectivity***

In this approach, density-based connectivity is "*a symmetric relation and all the points reachable from core-objects can be factorized into maximal connected components serving as clusters*" (Mann & Kaur, 2013). Both density and connectivity measures are used together to form clusters. The noncore-points inside a cluster represent the cluster's boundary, and the internal points represent core-points. On the other hand, the points that are not connected to any core-points are called outliers (Berkhin, 2006). The density-based clustering algorithms that represent this approach are DBSCAN (the most well-known density-based clustering algorithm), GDBSCAN, OPTICS, and DBCLASD.

The **DBSCAN** algorithm is the first and one of the well-known density-based clustering algorithms that aim at identifying high dense region to form clusters and low regions to separate them. The main idea of the DBSCAN is that each point in the dataset is scanned to determine the nearest neighbours within a particular distance (Ester et al., 1996). To determine a core-point, the number of neighbours should exceed the threshold. Otherwise, it might be a border-point or a noise. DBSCAN depends on two main parameters which are (Ester et al., 1996): 1) ***Eps*** (the radius of a point **P** to determine its neighbourhood; and 2) ***MinPts*** (the neighbourhood of a point **P** that should contain a minimum number of neighbours within the radius/Eps of point **P,** indicated by $\boldsymbol{N_{Eps}(P) \geq Minpts}$) based on the density and connectively approach, see Figure 2-2. The advantages of DBSCAN are its ability to detect clusters with arbitrary shapes, no need for any prior

knowledge for the number of clusters, and it handles noise very well. However, DBSCAN does not consider the attribute similarity resulting of little knowledge of how those points are similar to each other (Q. Liu, Deng, Shi, & Wang, 2012). Also, it cannot detect clusters with uneven densities (Q. Liu et al., 2012). Another disadvantage, it is a static algorithm that cannot cluster dynamic data.

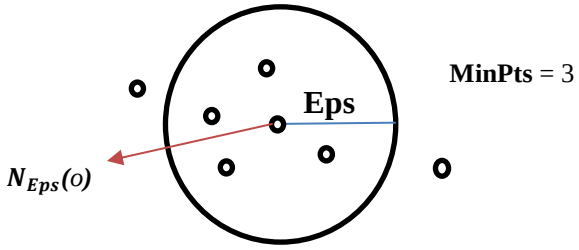

Figure 2-2: DBSCAN Algorithm Identifying a Neighbourhood of a Point based on MinPts

Two years after introducing DBSCAN, a new algorithm was implemented that considers clusters' points based on their spatial and non-spatial attributes, called- **GDBSCAN,** a version of generalized DBSCAN. It replaced Eps-neighbourhood with the notion of a *binary predicate*, instead of counting neighbours of a point within the Eps (radius), other measures have been used to define the neighborhood of a point. The idea of DBSCAN depends on the definition $\boldsymbol{N_{Eps}(P) \geq Minpts}$ where (the neighborhood of a point **P** should have a minimum number of points within Eps (radius) in order for **P** to be a core-point). On the other hand, the definition of neighborhood in GDBSCAN is $\boldsymbol{wCard(S) \geq MinCard}$. wCard is the weight of a set of points calculated by summing up the values of non-spatial attributes for a set of objects (Sander, Ester, Kriegel, & Xu, 1998). So the neighborhood is evaluated based on the weight function *wCard* and the minimum weight *MinCard* that is an input parameter to indicate the minimum weight of the neighbourhood of **P**. The rest of the definitions are similar to the DBSCAN. As a plus for GDBSCAN, it considers spatial attributes to study the similarity attributes compared with DBSCAN, OPTICS and DENCLE (Q. Liu et al., 2012). However, it does not discover clusters with uneven densities, and it cannot cluster dynamic data (Q. Liu et al., 2012).

Another density-based clustering algorithm introduced a year after GDBSCAN is called **OPTICS.** It extends the DBSCAN algorithm to generate augmented cluster ordering consisting of the ordering of the samples of the reachability-values and the core-values (Ankerst et al., 1999). It uses the same parameters as DBCSAN Eps (radius) and

MinPts, but it goes one step further by extending the infinite number of distance parameters ε_i which are smaller than the "generating distance ε" ($\varepsilon_i \leq \varepsilon$). For each point, OPTICS stores only two additional fields: the ***core-distance*** (the smallest distance ε_i between point and an object in its ε-neighbourhood), and the ***reachability-distances*** (the smallest distance such that **o** point is directly density-reachable from **P** if **P** is a core-object) and it cannot be smaller than the core-distance (Ankerst et al., 1999). OPTICS has the ability to discover clusters with uneven densities compared with DBSCAN, GDBSCAN and DENCLE by linearly ordering the points so the closest ones become neighbours in order. However, like the DBSCAN it does not consider the attribute similarity and does not produce clusters explicitly (Ankerst et al., 1999; Q. Liu et al., 2012). Also, this algorithm is static; it cannot cluster dynamic data.

2) ***Density Function***

In this approach, the density-based clustering algorithms form clusters by computing the density function (kernel function). The density function "*is defined over the attributes space and the overall density is modeled as the sum of the density functions of all objects. Clusters are determined by density attractors, where density attractors are local maxima of the overall density function. The influence function can be an arbitrary one.*" (Mann & Kaur, 2013). DENCLUE is the density-based algorithm that represents this approach of the density function.

DENCLUE is based on a set of density distribution functions that concentrates on *local maxima* of density functions called *density attractors.* The density attractor uses a *hill climbing* procedure to assigns the instance to a local maxima. A hill climbing procedure starts at a data point and iterates until the density does not grow anymore (Hinneburg & Gabriel, 2007). In addition, the arbitrary shapes clusters are defined as unions of local shapes along sequences of neighbours whose local densities are no less than a prescribed threshold ξ (indicates the noise threshold) (Berkhin, 2006). This algorithm works well with applications such as multimedia of high dimension and molecular biology data. On the other hand, it is very hard to determine the global parameter **σ** (density parameter) and **ξ** (noise threshold). Besides that, it is complicated to work with Kernel functions.

2.2.1 Evaluation of the Density-based Clustering Algorithms

The evaluation of the density-based clustering algorithms (DBSCAN, GDBSCAN, OPTICS and DENCLUE) is summarized in Table 2-1. For the evaluation, a list of seven requirements is presented on the left side of the table. The requirements are as follows: 1) ***Arbitrary Shapes:*** the algorithm should have the ability to form clusters of any shape such as "S" shape rather than round shape (Han et al., 2011); 2) ***Different Densities:*** the algorithm should have the ability to identify uneven densities of the cluster; 3) ***Handling Noise:*** ability to identify points that are noise in the dataset; 4) ***Attribute Similarity:*** algorithm should be able to study how close are points from each other by measuring the distance among points according to their attributes' values; 5) ***Managing High-dimensionality:*** is the ability of the algorithm to cluster a data of tens to many thousands of dimensions (attributes); 6) ***Computational Complexity:*** it is based on the running time and memory usage, where the algorithm should be faster in clustering points, and utilizing little memory; and 7) ***Input Parameters:*** evaluates the algorithm if it could automatically configure the input parameters or they are keyed manually.

These requirements are usually used to evaluate the performance of clustering algorithms, and they are obtained from the following articles as well as their information (Ankerst et al., 1999; Chaoji & Zaki, 2009; Q. Liu et al., 2012; Rui & Wunsch, 2005; Sander et al., 1998). In this study, they are utilized to identify the gaps among the different algorithms discussed in this section. From the Table 2-1, DBSCAN and GDBSCAN failed to detect different densities within the cluster. On the other hand, all algorithms can identify clusters with arbitrary shapes. Also, they can handle noise. However, DBSCAN, OPTICS and DENCLUE still lack the ability to study the attribute similarities of the samples like GDBSCAN. In terms of the input parameter, all the four algorithms need to set the input parameters manually.

Table 2-1: Evaluation of the Density-Based Clustering Algorithms based on Various Criteria

Evolution Matrices	Density-based Clustering Algorithms			
	DBSCAN	**GDBSCAN**	**OPTICS**	**DENCLUE**
Arbitrary Shapes	Yes	Yes	Yes	Yes
Clusters with Uneven Densities	No	No	Yes	No
Handling Noise	Yes	Yes	Yes	Yes
Attribute Similarity	No	Yes	No	No
Handling High Dimensionality	Not very well	N/A	N/A	Yes
Time Complexity	$O(n^2)$ or partial Index: O(nlogn)	$O(n^2)$ or partial Index: O(nlogn)	$O(n^2)$ or partial Index: O(nlogn)	$O(n^2)$ or partial Index: O(nlogn)
Parameters	**-Ept:** Radius. **-MinPts**: Minimum in number of points within a radius.	**-NPred:** neighbourhood predicate. **-WCard:** a weight function. **-MinCard:** a minimum Weight.	**-Ept:** Radius. **-MinPts:** Minimum in number of points within a radius. **-Core distance -Reachability Distance**	**σ:** density parameter **ξ:** noise threshold

This section studies the main density-based clustering algorithms that employ the idea of finding dense regions to form clusters. Each of those algorithms could fulfill several requirements in clustering, but it failed in achieving others. Moreover, they work best in the offline phase, since they are built to cluster static datasets (Han et al., 2011). Accordingly, they cannot cluster data coming from the dynamic environment (online). The following section discusses how clustering algorithms can be improved to cluster dynamic data.

2.3 Incremental Learning

Incremental Learning (IL) is one of the machine learning paradigms that take place whenever new samples emerge, and the overall learning is refined once the new samples become part of a group. IL is defined by Kulkarni (2012) as follows; *"it is the ability of the learning methodology to make effective use of new information and already formed features vectors or existing knowledge base which is generated in a previous phase of learning"*. There are several terms that are used interchangeably for incremental learning such as ***Online Learning***, ***Adaptive Learning***, and ***Transfer Learning***. The advantage of incorporating IL is to enable the algorithms to deal with dynamic datasets and to learn from the changes happening over time to update the clusters.

Incremental learning is essential when dealing with dynamic data including network traffic, security, telecommunication, and data management, since the data of those applications is evolving over time at high speed. In addition, it is important to identify the main factors that lead to accelerate the need to use incremental learning methods when clustering dynamic data (Kulkarni, 2012), see Figure 2-3**.**

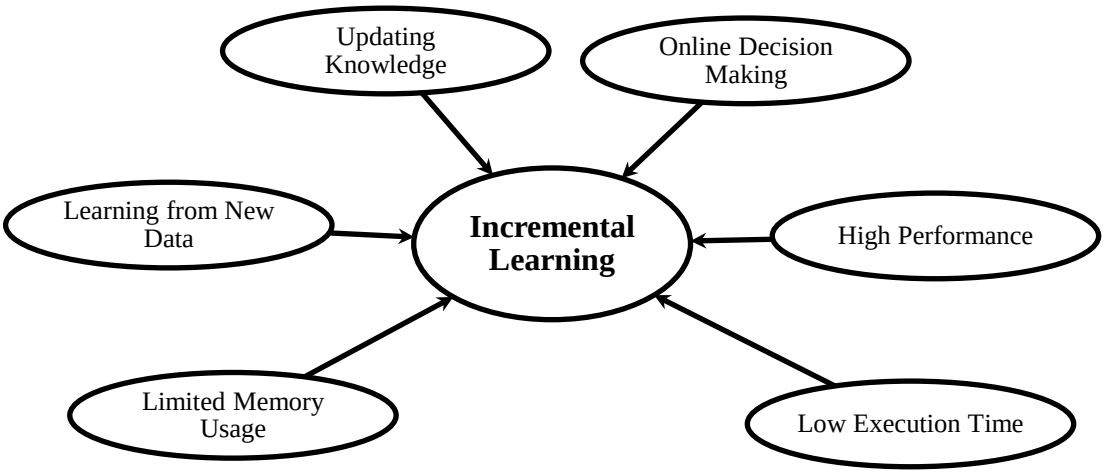

Figure 2-3:Factors Motivating Incremental Learning Methods (Kulkarni, 2012)

By incorporating IL, the algorithm can accommodate new knowledge over time and update it according to the new samples. Furthermore, it takes part in reducing the running time of the algorithm and limiting the use of the memory while maintaining a high performance in term of the accuracy.

Methods of IL are applied in both supervised and unsupervised learning, and the learning process of each one is conducted in a different manner. For instance, in the

unsupervised learning, the learning depends on the similarity, closeness, and distance among groups. As for the supervised learning, the learning responds to the information without retraining (Kulkarni, 2012). This study focuses on the clustering algorithm (unsupervised learning method) by grouping unlabeled data based on the similarity measures. The following section discusses the tasks required to incorporate incremental learning on clustering algorithms (unsupervised learning).

2.3.1 Incremental Learning Tasks for Clustering Technique (Unsupervised Learning)

The primary goal of integrating incremental learning methods with the clustering algorithm is to keep track of the potential changes that happen during the process of grouping new samples over time. It is necessary that the clustering algorithm can deal and learn from the changes by adjusting itself based on the newly received data. Here, the incremental learning process takes a role to enable the algorithm to learn and adjust itself over time by performing the following tasks, see Figure 2-4 (Kulkarni, 2012). Incremental learning allows the algorithm to generate clusters dynamically without prior knowledge of the number of clusters. Also, it can accommodate the new incoming samples to the existing clusters or form new clusters over time. Other important tasks are making decisions on whether to merge or split clusters, discarding samples that are not required further and fading clusters over time.

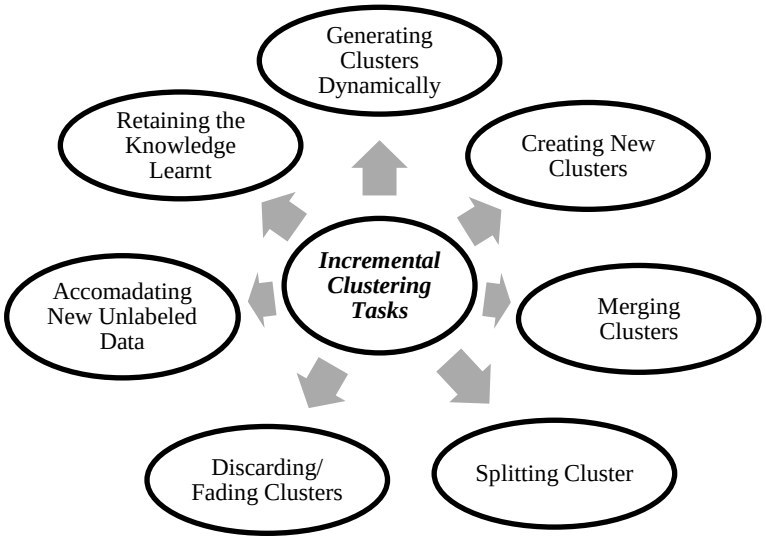

Figure 2-4:Tasks Carried Out By Incremental Clustering (Kulkarni, 2012)

While incorporating the IL methods into the clustering algorithms, a number of requirements need to be taken into accounts which are (Kulkarni, 2012): 1) No prior knowledge about the number of clusters when clusters are generated; 2) Handling noise; 3) Avoiding overlapping among clusters; 4) Be accurate and fast during the learning process. The following section reviews several clustering algorithms that incorporated incremental learning methods to cluster dynamic datasets.

2.3.2 Incremental Clustering Algorithms

This section discusses some of the existing studies that incorporate methods of incremental learning and highlights the tasks that are carried out by these incremental clustering algorithms to cluster dynamic data.

The first incremental clustering algorithm was introduced by Ester et al.(1998). It is based on the well-known DBSCAN algorithm. The main purpose of the proposed incremental DBSCAN was to update the clusters whenever new samples inserted and deleted at the data warehouse. One data sample is inserted and deleted from the existing cluster one at the time (Goyal, Goyal, Venkatramaiah, Deepak, & Sannop, 2011). For the insertion task, the new incoming sample denoted with **P** is classified to either one of the following, see Figure 2-5 (Ester et al., 1998). 1) Classified as a noise, if **P** has no neighbours; 2) **P** is inserted into a cluster, if it is a core-point in its neighbourhood, and the neighbouring samples belong to one cluster; 3) If it is a core-point and the neighbouring samples of **P** are from different clusters, then the **P** is inserted, and all clusters are merged as one cluster; 4) New cluster is created, if there is a core- point that did not belong to a cluster before inserting **P** (means they are noise), then a new cluster is created to group these noises. After inserting **P**, the knowledge update takes place to update clusters based on the newly inserted sample where some core-points may become either border-points or noise and vice versa.

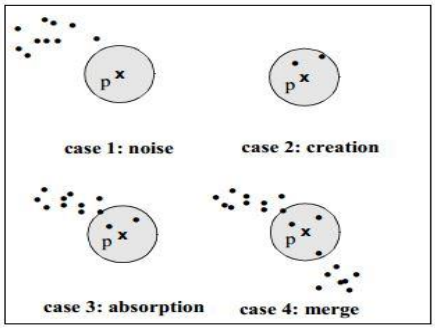

Figure 2-5:Different Cases of Insertion (Ester et al., 1998)

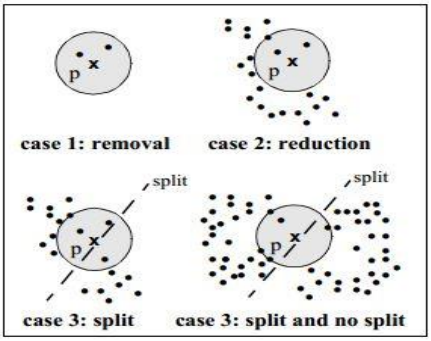

Figure 2-6: Different Cases of Deletion (Ester et al., 1998)

Another task fulfilled by the incremental DBSCAN is the deletion. Deleting **P** from the cluster can lead to either one of the following: 1) Removal of **P**, if it is not a core-point within the neighbourhood of **P**. Then **P** is removed and other samples become noise leading to remove the cluster entirely; 2) Reduction happens if all neighbours of **P** are directly density-researchable from each other. Then **P** is deleted, and some samples may become noise; 3) Splitting the cluster if the neighbour samples of **P** are not directly density-reachable from each other and they belong to one cluster before the deletion of **P**, see Figure 2-6 (Ester et al., 1998).

The incremental DBSCAN was compared with the DBSCAN and validated on synthetic datasets and WWW access logs. The performance of both algorithms was evaluated based on the maximum updates and speed-up factor. Incremental DBSCAN showed better performance in terms of the both.

Another study was proposed by Goyal et al. (2011) to improve the first incremental DBSCAN presented earlier which adds and deletes one sample one at a time. This process leads to more region queries. For that, this study improves the incremental DBSCAN by first collecting a number of samples and clusters them using the DBSCAN. Then the insertion task takes place by adding the newly generated clusters to the existing clusters instead of adding one sample at a time. Once the new clusters are added, the algorithm checks if the new points intersect with old ones. If so, then the incremental DBSCAN proposed by Ester et al. (1998) performs the required updates to the clusters memberships of the points. In addition, the improved algorithm can merge clusters based on the points presented at the intersection of the old and new clusters. The improved algorithm uses less region queries compared with the incremental DBSCAN at different levels of noise.

A study by Angel et al., (2012) proposed a density-based dynamic DBSCAN clustering algorithm based on the well-known DBSCAN introduced by Ester et al. (1996). This study focuses on developing an incremental DBSCAN that considers the problems of data insertion to find relations among the data in a real-time environment, radius is dynamically changed with every insertion, and also noise and border-points are considered unclassified and they are evaluated again with the new points. In the Incremental DBSCAN algorithm, the new incoming sample is classified to either one of

the following: 1) Classified as a noise, if the number of neighbours is not satisfied by the condition of MinPts, as described in the traditional DBSCAN in Section 2.2; 2) The new point is inserted into a cluster, if the neighbouring points belong to a cluster; 3) If the neighbouring points of the new samples belong to more than one cluster, then a new sample is inserted and all clusters are merged as one cluster; and 4) A new cluster is created, if the new sample is treated as a core point. The performance of the incremental DBSCAN was compared with the DBSCAN and Chameleon (a hierarchical clustering algorithm) using synthetic and real datasets. The results of the algorithms were evaluated by Generalized Dunn Index, Davies-Bouldin, and time taken for clustering. The results showed that the performance of the proposed algorithm is almost equal to the tradition DBSCAN based on Dunn Index and Davies-Bouldin. Moreover, the incremental DBSCAN was faster in comparison with DBSCAN and Chameleon.

The following section discusses another concept that is the ***Law of Gravity***, since this concept is one of the fundamental concepts of this study. It is essential to review the existing works in data mining and particularly clustering algorithms that adopted the Newton's theory of gravity and to examine to what extent the idea is feasible and valid.

2.4 Clustering Algorithms Adopted the Gravity Theory

Newton defined gravity as the universal gravitation between two objects, where the force of gravity depends on the weight of the objects, and it depends on the distance between objects. The gravity formula is presented in Equation (2-1) as follows:

$$f = G\frac{m_1 * m_2}{r^2} \tag{2-1}$$

Where f is the force, m_1 is the mass of the first object, m_2 is the mass of the second object, r^2 is the distance between the two objects, and G is gravitation constant; see Figure 2-7. The primary components of the gravity theory are: 1) ***Mass*** of an object: the bigger the object is, the greater the force of gravity to pull an object; and 2) ***Square Distance*****:** the distance between two objects, if the distance increases between the objects the force of gravity of the mass decreases.

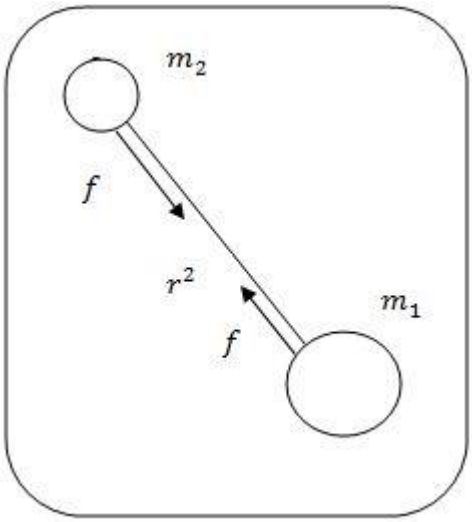

Figure 2-7: Newton's Law of Universal Gravitation

Newton's law of universal gravitation first inspired Wright et al. (1977), who proposed a gravitational clustering algorithm in data mining. Several studies later adopted the gravity theory of Newton to improve the existing clustering algorithms to become incremental. Several unrelated works to this study invoke the gravity for outlier detection. These works are reviewed in this literature review in order to highlight how the idea of gravity in physics can be involved in data mining. Also, to show how the idea of gravity can make a difference in overcoming existing problems in a particular domain. The following section discusses several studies that employed the concept of gravity to improve a particular clustering algorithm.

The GRIN algorithm is an incremental hierarchical data clustering algorithm based on the gravity theory that constructs clustering dendrograms (C. Chen, Hwang, & Oyang, 2002). The GRIN algorithm can add new incoming samples to the clusters, creating new clusters and detecting outliers. Furthermore, the gravity-based clustering algorithm for the Euclidean space (GRACE) introduced by Oyang et al.(2001), is invoked in the GRIN algorithm in order to check the possibility of merging clusters. The clusters merged when the distance between two clusters is less than the clusters' radius. The performance of the GRIN algorithm was compared with the BIRCH algorithm (Balanced Iterative Reducing and Clustering Using Hierarchies) proposed by Zang et al. (1996). The law of gravity was employed in both studies as an incremental method to add and merge clusters.

In 2008, a study by (Meng & Cheng, 2008) proposed a gravity-based outliers detection algorithm (GODA). This algorithm can identify outliers by considering two elements density-based and distance-based technique that both used the gravity formula, see Equation (2-1). One year later, a study by Wang et al. (2009) presented an algorithm called- Gravity-based Anomaly Intrusion Detection (GAIDA). This algorithm made use of the gravity concept according to the density-based and distance-based techniques as well. Both studies were able to detect outliers through the use of the gravity theory.

Fuzzy C-mean (FCM) clustering algorithm is sensitive to outliers and noise, and the initial cluster centers. A study by Zhong et al. (2010) introduced a solution to overcome these problems of the FCM by adopting the idea of gravity to remove outliers from the dataset. Also, by adopting the gravity concept, the algorithm can identify the initial cluster centers. The law of gravity consists of two primary components which are: 1) Distance (from object p to object q); and 2) The mass (the number of neighbours within the radius of p and q). The outliers are identified as follows: Set a threshold ε, then calculate the gravity of point x_i to other objects in the dataset and sum it, repeat the same process with each object in the dataset. The object with smaller sum of gravity than ε is called an outlier, and it is removed. In addition, the point with the largest sum of gravity than ε becomes an initial cluster center, then the sum of gravity is repeated to find the next initial cluster center that is larger than the threshold ε. The improved FCM was compared with the traditional FCM, and results showed that the accuracy of the improved FCM was slightly higher than the traditional FCM.

Many other studies are inspired by the law of gravity and used it in a variety of fields such as image edge detection in (Lopez-Molina, Bustince, Fernandez, Couto, & De Baets, 2010). The Gravitational Search Algorithm (GSA) is an optimization algorithm introduced in (Rashedi, Nezamabadi-pour, & Saryazdi, 2009). Also, the image segmentation algorithm based on the theory of gravity was proposed in (Rashedi & Nezamabadi-pour, 2012), and many more studies that involve the concept of gravity. It confirms that the law of gravity is feasible to be adopted in the area of data mining. As for this study, the Newton's law of gravity is used as an incremental method to cluster incoming samples and at the same time to consider the attribute similarities among samples.

2.5 Chapter Summary

This chapter covers the fundamental algorithms of the density-based clustering method since this study focuses on improving the performance of a density-based clustering algorithm. It was necessary to investigate several algorithms related to the study area to have a clear understanding of the algorithms and how they work. Besides that, there are several requirements identified to evaluate the performance of the algorithms. In addition, it provides a brief description of the incremental learning factors, tasks, and several incremental algorithms. According to the reviewed works in this section, Table 2-2 and Table 2-3 summarize the main outcomes of the literature review that highlights the gaps identified from the existing studies of the static and dynamic algorithms. The requirements with their information of the algorithms performance are identified from the surveys that are conducted by Q. Liu et al., (2012) and Rui et al. (2005). Moreover, some other information is obtained from the main articles (as highlighted in the tables) which discuss the algorithms in details.

Table 2-2: Summary of the Discussed Density-Based Clustering Algorithms

Evolution Matrices/ Requirements	**Density-based Clustering Algorithms**			
	DBSCAN	**GDBSCAN**	**OPTICS**	**DENCLUE**
Article	Ester et al. (1996)	Sander et al. (1998)	Ankrest et al. (1999)	Hinneburg et al. (2007)
Dataset	Static	Static	Static	Static
Arbitrary Shapes	√	√	√	√
Cluster with Uneven Densities	×	×	√	×
Handling Noise	√	√	√	√
Attribute Similarity	×	√	×	×
Handling High Dimensionality	×	N/A	N/A	√
Automatically Defined Parameters	×	×	×	×
Incremental	×	×	×	×

Table 2-3: Summary of the Discussed Incremental Clustering Algorithms

Evolution Matrices/ Requirements		Incremental Clustering Algorithms			
		Incremental DBSCAN	**Improved Incremental DBSCAN**	**Improved Incremental DBSCAN**	**Incremental Hierarchical Clustering**
Article		Ester et al. (1998)	Goyal et al. (2011)	Angel et al. (2012)	Chen et al.(2002)
Dataset		Dynamic	Dynamic	Dynamic	Dynamic
Arbitrary Shapes		√	√	√	√
Cluster with Uneven Densities		×	×	×	N/A
Handling Noise		√	√	√	√
Attribute Similarity		×	×	×	√
Handling High Dimensionality		×	N/A	N/A	√
Automatically Defined Parameters		×	×	√	×
Incremental Tasks	**Insert Points**	One points at a time	Bulks of points at a time	One points at a time	One points at a time
	Creating Cluster	√	√	√	√
	Merging Clusters	√	√	√	√
	Splitting Clusters	√	×	×	×
	Removing Points	√	√	×	√
	Removing Clusters	√	×	×	×
	Update Points	√	√	×	Not required

CHAPTER 3. THE ARCHITECTURE OF THE PROPOSED DENSITY-BASED CLUSTERING ALGORITHM USING GRAVITY AND AGING

This study proposes two approaches to improve the performance of the density-based clustering algorithm. The first approach accommodates new measurements to define the dense region by studying the attribute similarities of points. The second approach presents an incremental learning process that uses the memory efficiently. This section is divided into four subsections. First, Sections 3.1.2, 3.2.1, and 3.2.2 discuss several versions of the proposed algorithm that are taken into account to form the final solution that achieves most of the objectives. Followed by Sections 3.2.3 and 3.3**,** that provides an overview of the proposed algorithm and introduces the fundamental concepts taken into consideration to develop the proposed clustering algorithm. Second, Section 3.4 presents the approach that allows the proposed algorithm to utilize the memory efficiently and to deal with a dynamic behaviour. Section 3.5 provides the complete overview of the proposed algorithm. Lastly, Section 3.6 provides the pseudo-code of the improved clustering algorithm after employing new approaches of incremental learning for the insertion and deletion tasks.

3.1 Preliminaries

3.1.1 Initial Stage

Before discussing the architecture of the proposed algorithm, the study focuses on the testing phase where the dataset is divided into 40% for training and 60% for testing. The training set is used to create some initial clusters using the well-know DBSCAN algorithm (Ester et al., 1996), while the testing set acts as the new incoming points which arrive over time, see Figure 3-1.When the new incoming point P_i is arriving, the algorithm has to group it with one of the clusters according to the similarity measures. First, by considering the radius (*Eps*), the algorithm finds the neighbours of P_i, and checks how many neighbours are within that radius. If the new point P_i contains neighbours less than the minimum number of points within the radius (indicated by the condition: $Neighbour_{Eps}(P_i) \geq MinPts$), then the new incoming point is moved to a ***temporary bag***, and the algorithm proceeds with the next new incoming sample.

Otherwise, the algorithm starts evaluating the neighbourhood of the neighbouring samples to cluster the new incoming point.

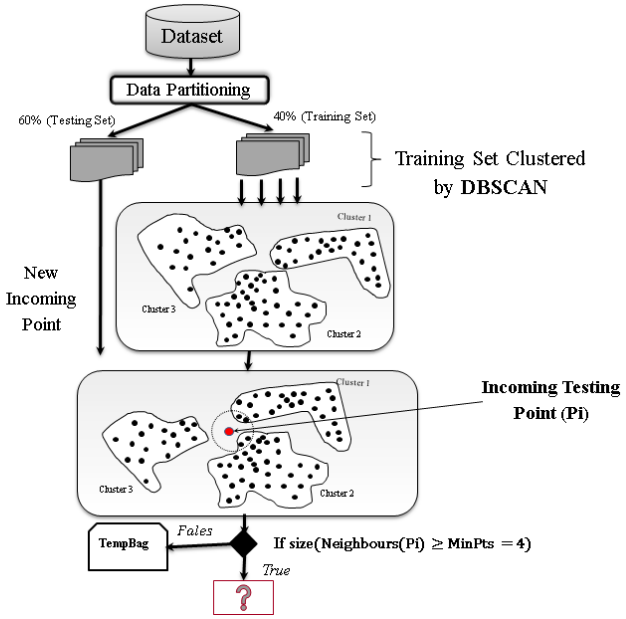

Figure 3-1: Initial Stage (Training Phase)

The new incoming point P_i is grouped into one of the clusters by evaluating the neighbourhood of the neighbouring samples based on several versions of the proposed algorithm which are discussed in details in the following sections.

3.1.2 Density Computation using the Total Number of Points within Radius (DBNPR algorithm)

The DBNPR algorithm employs the idea of DBSCAN to cluster the new incoming samples, where two different radiuses are introduced, the first radius (*Eps1*) finds the neighbours (denoted by $Neighbour_{Eps}$) of the new incoming point (denoted by P_i). If the new point P_i contains neighbours more or equal to the minimum number of points within the radius denoted by $MinPts$ (indicated by the condition: $Neighbour_{Eps}(P_i) \geq MinPts$), see Figure 3-2 (Step1). Then proceed with measuring how dense is the area surrounding the neighbours of P_i by introducing the second radius (*Eps2*), that counts how many points within the radius of each neighbour sample of P_i, see Figure 3-2 (Step 2). Consequently, if there is more than one neighbour belong to the same cluster sum their densities (see the three yellow points in cluster 1, as shown in Figure 3-2). Then, the cluster that has the higher density is the one that pulls in the new incoming sample, since its neighbourhood is denser than other neighbours of P_i in other clusters. Otherwise, if

the condition is false, then the new incoming point is moved to a *temporary bag*, and the algorithm proceeds with the next new incoming sample.

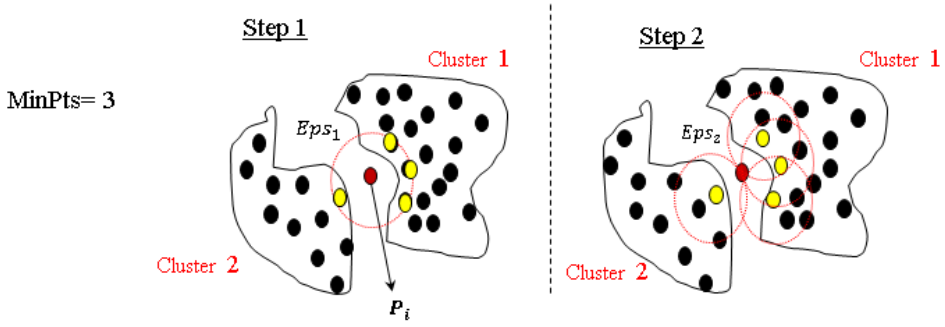

Figure 3-2: Clustering New Incoming Point P_i

The equation of the DBNPR algorithm is defined in the following Equations.

$$den(Neighbour_j) = Total\ number\ of\ points\ \ within\ the\ radius2\ of\ Neighbour_j \quad (3\text{-}1)$$

$$ForceC_n = \sum_{Neighbour_j \in\, c_n} den(Neighbour_j\ located\ in\ C_n) \quad (3\text{-}2)$$

$$WinningForce = \max_{c_n = c_1 \ldots c_k} (ForceC_n) \quad (3\text{-}3)$$

Where *den* represents density and *den*($Neighbour_j$) measures the density of each neighbour j^{th} of P_i by counting the total number of neighbouring samples of $Neighbour_j$ within the radius (radiua2). Then, Equation (3-2) sums up the densities of neighbours of P_i that are in the same cluster C to calculate the mass of each cluster. After that, Equation (3-3) indicates that the cluster with the maximum density is the one that has the winning force to pull in the new incoming point. In this equation, k is the number of clusters. This method, however, does not study the attribute similarity among points which is part of the objectives of the study.

3.1.3 Introducing the Gravity Concept

The intention of this study is to adopt and study the gravity theory, and incorporate the concept in clustering. ***Gravity Approach*** presents the idea of force that depends on two main elements which are: 1) **The Mass** (or density) represents the body (the bigger the body is, the larger the force to pull the body); and 2) **The Square distance** is the distance from the body to another body (the strength of the body's force of gravity

to pull the body is affected by the distance). Therefore, an increase in the distance leads the force of gravity of the body to decrease, see Equation (3-4).

$$f = G\frac{m_1 * m_2}{r^2} \tag{3-4}$$

In this study, according to Equation (3-4), the gravity concept is adopted as follows: 1) Masses (m_1 and m_2): m_1 indicates the density of the new incoming point P_i which is always equal to one, and m_2 represents how dense is the neighbourhood of the neighbouring samples of the incoming new sample. 2) The square distance (r^2) indicates the distance from the incoming sample P_i to its neighbouring samples. The algorithm computes the forces of each neighbourhood, and then the neighbourhood with the higher force is the one that pulls in the new incoming sample P_i to its cluster. Figure 3-3 highlights how the idea of gravity is adopted in clustering.

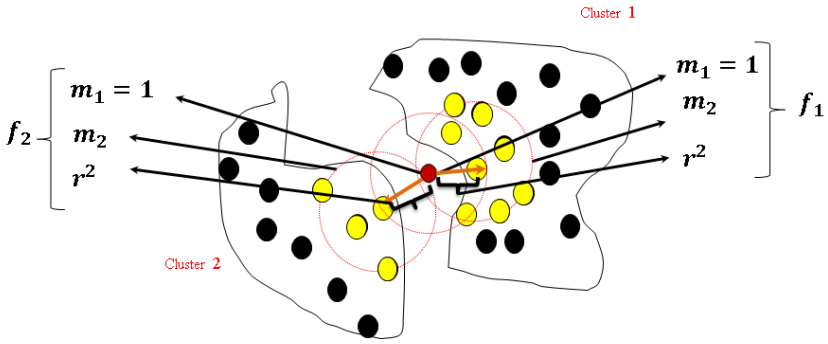

Figure 3-3: Adopting the Gravity Concept in Clustering

It is possible and feasible to apply this concept in clustering since clustering deals with density and distance as well. For that, the following sections discuss several versions of the proposed algorithm, which introduce different aspects of obtaining new measures to calculate density and incorporating them with the idea of gravity to create a new method for clustering dynamic data. The evaluation of these versions is discussed in Chapter 4.

3.2 Proposed Algorithms

3.2.1 Gravity-based Density (Mass) Computation using the Number of Points within Radius (GrDBNPR Algorithm)

This version considers the idea of counting the number of points within a radius to measure the density (mass). The mass (or density) of each neighbour of (P_i) is measured as discussed in 3.1.2. To apply gravity, the density (mass) of each neighbouring sample of (P_i) is measured, and then the density of each neighbouring sample is divided with the

square distance to identify the force of gravity for each neighbour. The concept of GrDBNPR algorithm is defined in the following Equations.

$$den(Neighbour_j) = Total\ number\ of\ points\ within\ the\ radius2\ of\ Neighbour_j \quad (3\text{-}5)$$

$$ForceGrv\ (Neighbour_j) = den(Neighbour_j)/[dist(P_i, Neighbour_j)]^2 \quad (3\text{-}6)$$

$$ForceGrvC_n = \sum_{Neighbour_j \in\ c_n} ForceGrv\ (Neighbour_j\ located\ in\ C_n) \quad (3\text{-}7)$$

$$WinningForce = \max_{c_n = c_1 \ldots c_k} (ForceGrvC_n) \quad (3\text{-}8)$$

Where *den*($Neighbour_j$) measures the density of each neighbour j^{th} of P_i by counting the total number of neighbouring samples of $Neighbour_j$ within the radius (radius2). According to Equation (3-6), the force of gravity of each $Neighbour_j$ is computed by dividing the *den*($Neighbour_j$) to the square distance from P_i to neighbour j^{th} of P_i, indicated by $[dist(P_i, Neighbour_j)]^2$. Then, sum up the forces of the neighbours of P_i that are in the same cluster C, as indicated in Equation (3-7), and the cluster with the highest force of gravity is the one with the winning force that pulls in the new incoming point, as indicated in Equation (3-8). In this equation, k is the number of clusters. This method, however, does not study the attributes similarities among points as well.

3.2.2 Gravity-based Density (Mass) Computation using the Silhouette Measure (GrDBSil Algorithm)

In this version, the silhouette concept is employed to measure the mass (or density) of each neighbouring sample of P_i. The reason of employing such concept is because of its ability to study the attribute similarities among points using the distance measurement. The concept of the GrDBSil algorithm that uses the **Silhouette** measure to compute density of the neighbours is defined as follows.

To employ the silhouette measure to compute the density of the gravity formula is by the following steps:

1. Identify the neighbouring samples of P_i based on the given radius1.
2. Identify the points x of each neighbour j^{th} of P_i based on the given radius2.

3. Group all the points x within the radius2 of the neighbour j^{th}, and calculate the distance from the point x^{th} to all other points x^{th} of neighbour j^{th}. Then, sum up the distances of x^{th} to other points, and to measure the average similarity score of x^{th}, divide the sum distances of point x^{th} by the total number of points of neighbour j^{th} minus one, and apply the same for each x.
4. After measuring the average similarity scores of each point x, the neighbourhood density of neighbour $j^{th}$of P_i denoted by $den(Neighbour_j)$ is measured by summing up the average similarity scores of all points x of neighbour j^{th}, and then divides it by the total number of points x of neighbour j^{th}.
5. Compute the force of gravity for each neighbouring sample by dividing the density of each neighbouring sample of P_i denoted by $den(Neighbour_j)$ with $[dist\left(P_i, Neighbour_j\right)]^2$ is the square distance from the new incoming point P_i to $Neighbour_j$, as defined in Equation (3-9)

$$ForceGrv\left(Neighbour_j\right) = den(Neighbour_j)/[dist(P_i, Neighbour_j)]^2 \quad (3\text{-}9)$$

6. Sum up the force of gravity of the neighbours in the same cluster, as indicated in Equation (3-10). Then, the cluster with the highest value of force is the winning force that pulls in the new incoming sample, as defined in Equation (3-11). In this equation, *K* is the number of clusters.

$$ForceGrvC_n = \sum_{Neighbour_j \in c_n} ForceGrv\left(Neighbour_j\ located\ in\ C_n\right) \quad (3\text{-}10)$$

$$WinningForce = \max_{c_n = c_1 \ldots c_k} (ForceGrvC_n) \quad (3\text{-}11)$$

This version of the proposed algorithm studies the attribute similarities among points and introduces a new measurement incorporated with the gravity concept to improve the performance of the algorithm as highlighted in the objectives. Nevertheless, it might have a drawback in terms of the running time which is too long and is proved in Chapter 4. As a result, the following algorithm is proposed in order to fulfill the objectives of the study by overcoming the drawbacks of the versions discussed earlier.

3.2.3 Gravity-based Density (Mass) Computation using Average Distance (GrDBAveD Algorithm)

The improved density-based clustering algorithm adopts the notion of gravity concept to cluster incoming samples during the online testing phase. It introduces new measurements that take into account the similarities among points according to the distance measurements. In this study, the ***Euclidean distance*** is employed to measure the attribute similarities among the points since the dataset's attributes are numeric (Q. Liu et al., 2012). Similar to versions 2 (as discussed in 3.2.1) and 3 (as discussed in 3.2.2), mass is represented as how dense is the neighbourhood of the neighbouring samples of the incoming new sample, and the square distance indicates the distance from the incoming sample to its neighbouring samples. However, in this version the density is computed differently.

Figure 3-4 illustrates the main diagram of the proposed method based on the gravity concept integrated with new measures for the density. This study is focusing on the testing phase; accordingly, we assume that some clusters are available, as discussed in Section 3.1.1. First, the dataset is partitioned into 40% for the training set and 60% for the testing set. Then several clusters are available already by clustering the training set of 40% of the data using the DBSCAN. Then the proposed algorithm is applied on the testing phase to cluster the incoming samples by studying the attribute similarities among the points of the clusters. The process goes as follows: in which $\boldsymbol{P_i}$ is a new incoming testing sample.

1. Finding the neighbouring samples of $\boldsymbol{P_i}$ within the $radius_1$, denoted by $Neighbours_{Eps}(P_i)$.
2. If the number of points within the radius is greater or equal than a threshold "$Total\ Neighbours_{Eps}(P_i) \geq MinPts$", then proceed with step 3. Otherwise, the sample is moved and kept in temporary bag for later analysis, and then the algorithm proceeds with next incoming sample and starts over from step 1.
3. When the total number of neighbours of $\boldsymbol{P_i}$ is equal or greater than the threshold, then proceed with the second level to identify the neighbours of $Neighbour_j$ using radius2.
4. The distance from $\boldsymbol{P_i}$ to its neighbouring samples and the distance from each neighbours samples of $\boldsymbol{P_i}$ to its neighbours are calculated to measure the attribute similarities among the neighbours.

5. Then, the improved density-based clustering algorithm using gravity concept is applied to measure the density of the neighbouring samples over their distances to determine which cluster the new incoming point $\boldsymbol{P_i}$ belongs to. *(More description of the algorithm is presented in the following section)*
6. After applying the proposed algorithm, then the new incoming sample $\boldsymbol{P_i}$ emerge into the cluster of the neighbourhood with highest the force of gravity.
7. The process keeps looping whenever a new incoming sample is available to be clustered.

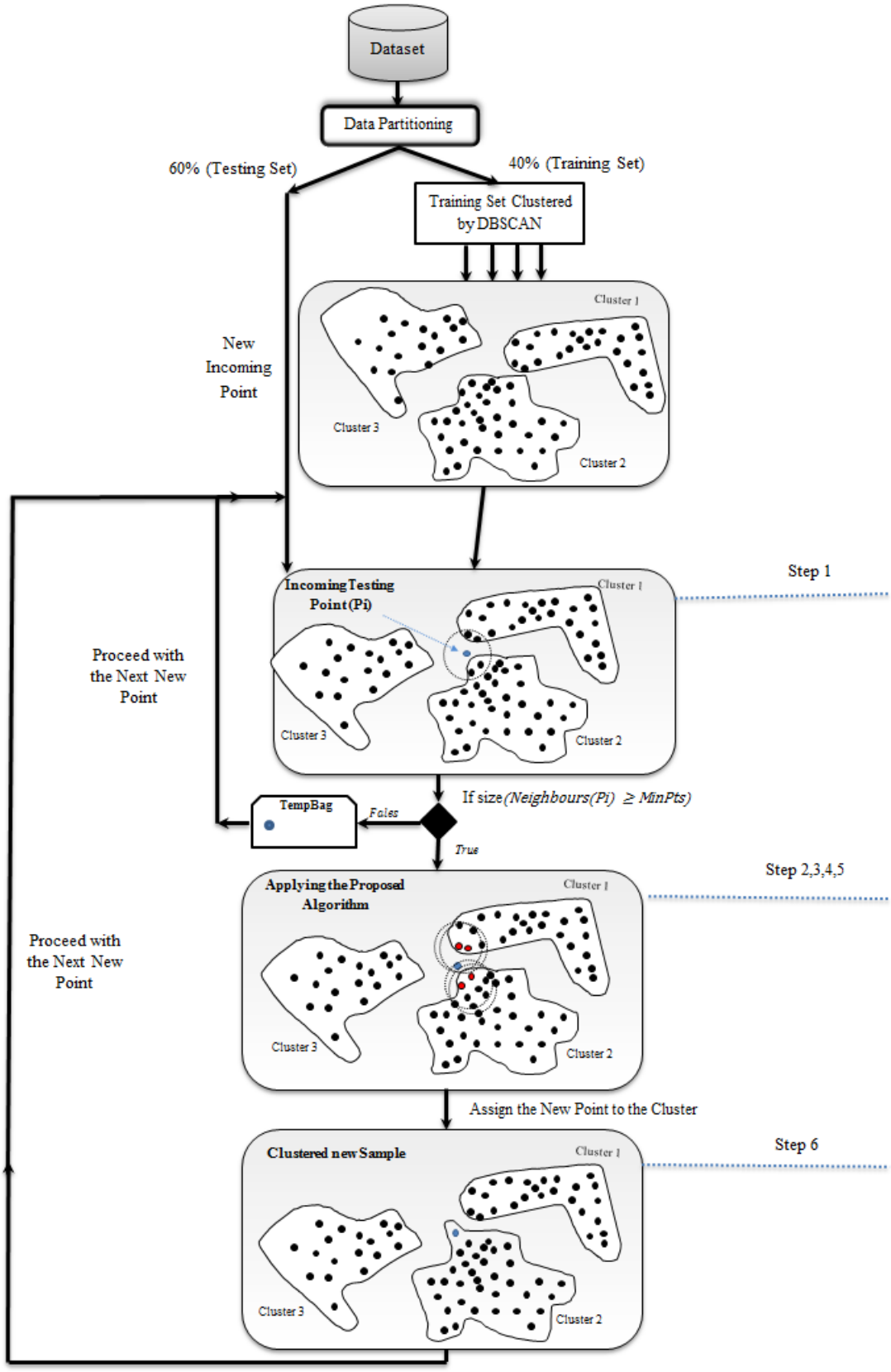

Figure 3-4: Overview of GrDBAveD Algorithm

After presenting the overview of the GrDBAveD algorithm, the following section describes the fundamental concepts and the definitions used to improve the density-based clustering algorithm.

Step 1) Neighbouring Samples of P_i

A sample P_i is a new incoming sample, and to identify the neighbouring samples of P_i, radius1(Eps_1) is given to contain at least a minimum number of points within the neighbourhood of P_i, and it is denoted by (*MinPts)* as introduced by (Ester et al., 1996).

1) If total $Neighbours_{Eps}(P_i) \geq MinPts$" Total number of neighbours within P_i radius is greater or equal than *MinPts*"
2) {Then proceed with the next phase (Step 2) "If true"
3) }
4) else { P_i is moved and placed in a temporary bag to be evaluated later }

Step 2) Neighbouring Samples of $Neighbour_j$ of P_i

After identifying the neighbours of P_i when condition (1) is true, the areas surrounding the neighbouring samples of P_i are scanned in order to measure their densities. For that, Radius2 is introduced and denoted by (Eps_2) to identify the neighbours of the $Neighbour_j$ to examine how dense is the neighbourhood of each $Neighbour_j$ of P_i, as shown in Figure 3-5

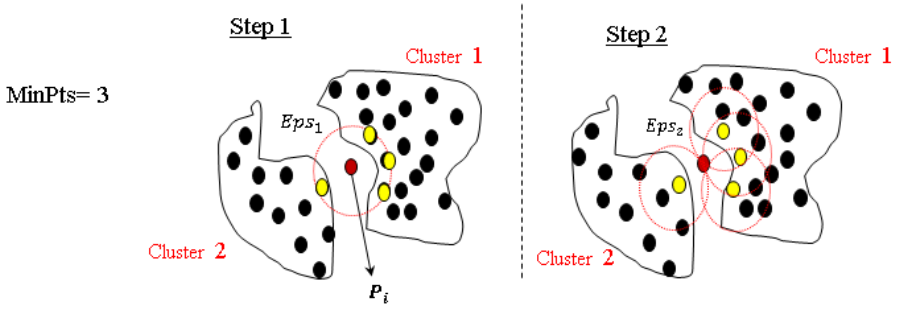

Figure 3-5:Identifying the Neighbouring Samples of P_i

Step 3) Attribute Similarities of the Neighbouring Samples of P_i

Once the neighbourhood of each neighbouring sample of P_i is identified as shown in Figure 3-5, the distances from P_i to its neighbouring samples, and the distances from the neighbouring samples of P_i to their neighbours are recorded. The distance is calculated using Euclidean distance to study the attribute similarities among the points (Q. Liu et al., 2012). Also, it is used in the process of identifying the densities of the

neighbouring samples as described in Step 4. The selection of the distance measurement is varying depending on the attributes data type. Euclidean Distance is defined as Equation (3-12) (Han et al., 2011)

$$d(x_i, x_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{in} - x_{jn})^2} \quad (3\text{-}12)$$

Where $d(x_i, x_j)$ is the distance between point x_i and point x_i and $(x_{i1} - x_{j1})$ indicates the distance between the first attribute of each point, $(x_{i2} - x_{j2})$ indicates the distance between the second attribute of each point and it goes on for the rest of attributes.

Step 4) Measuring the Density of the Area Surrounding the Neighbouring Samples of P_i

The neighbouring samples of $\boldsymbol{P_i}$ is surrounded by samples and in order to measure their densities. The following equation is introduced to measure density of the area surrounding each neighbouring sample of $\boldsymbol{P}_i$, and to study their attributes similarities as well, see in Equation (3-13).

$$AvgDis(Neighbour_j) = \frac{1}{\|x\|} \sum_{x_u \in Neighbour_j} dist(x_u, Neighbour_j) \quad (3\text{-}13)$$

Where $\|x\|$ is the total number of points within the radius (radius2) of $Neighbour_j$. $dist(x_u, Neighbour_j)$ is the distance from each point x^{th} to $Neighbour_j$. After, computing the distances from all points x to $Neighbour_j$, then sum up those distances and divide it by the total number of points within the radius (radius2) of $Neighbour_j$ to find the average distance of $Neighbour_j$, see Figure 3-6.

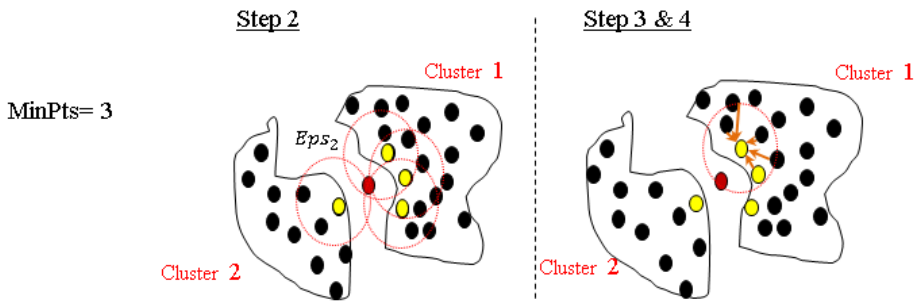

Figure 3-6: Measuring the Density of the Area Surrounding the Neighbouring Samples of P_i

According to Equation (3-13), the larger ***Average distance*** of $Neighbour_j$ is the lower the density. In order to indicate the higher density, Equation (3-14) shows that the smallest the average distance among the neighbouring samples of (P_i) is the more dense the $Neighbour_j$.

$$den(Neighbour_j) = \frac{1}{AvgDis(Neighbour_j)} \quad (3\text{-}14)$$

Step 5) Gravity Concept to Cluster P_i

The basic concept of gravity depends on two core elements:

- ***Mass of an object*:** the bigger the object is, the greater the force of gravity to pull an object.
- ***Distance***: The distance between two objects, if the distance increases between the objects the force of gravity of the mass decreases.

  Gravity formula is donated as Equation (3-15).

$$f = G\ \frac{m_1 * m_2}{r^2} \quad (3\text{-}15)$$

Where f is the force, m_1 is the mass of first object, m_2 is the mass of the second object, and finally r^2 is the distance between the two objects. However, G is the gravitational constant, which has no influence in the proposed clustering problem (Zhong et al., 2010).

The notion of gravity is adopted to examine the force of each neighbouring sample of $\boldsymbol{P_i}$ to determine which neighbouring sample pulls $\boldsymbol{P_i}$ to its region. In this study, mass is denoted by density (*den)* and r^2 is denoted by $[dist(..)]^2$. According to Equation (3-15), m_1 represents the density of any new incoming sample $\boldsymbol{P_i}$ which is always equal to 1, m_2 represents the density of each neighbouring sample of $\boldsymbol{P_i}$ denoted by the Equations (3-13 and 3-14) of $den(Neighbour_j)$. Based on Equations (3-13) and (3-14) the following Equation is defined

$$ForceGrv\left(Neighbour_j\right) = 1 * den\left(Neighbour_j\right)/[dist\left(P_i, Neighbour_j\right)]^2 \quad (3\text{-}16)$$

$$ForceGrvC_n = \sum_{Neighbour_j \in c_n} ForceGrv\left(Neighbour_j \; located \; in \; C_n\right) \quad (3\text{-}17)$$

$$WinningForceGrv = \max_{c_n = c_1 \ldots c_k} (ForceGrvC_n) \quad (3\text{-}18)$$

According to Equation (3-16), 1 represents the density of the incoming sample $\boldsymbol{P_i}$, and $den(Neighbour_j)$ is the density of each neighbouring sample ($Neighbour_j$) of $\boldsymbol{P_i}$ which is calculated by the Equations (3-13 and 3-14). Then, $[dis\left(P_i, Neighbour_j\right)]^2$ is the square distance between the new incoming sample $\boldsymbol{P_i}$ and $Neighbour_j$.

Step 6) Clustering $\boldsymbol{P_i}$

According to step 5, each neighbouring sample of $\boldsymbol{P_i}$ has its force. If there is more than one neighbour belong to the same cluster, then the forces of gravity those neighbours are summed up, as indicated in Equation (3-17). After summing up, the cluster with highest force among other clusters is the one with winning force that pulls in the new incoming sample $\boldsymbol{P_i}$ to be part of its cluster, as indicated in Equation (3-18). In this equation, *K* is the number of clusters.

3.3 Improving the GrDBAveD Algorithm[2]

As discussed in Section 3.2.3, the proposed algorithm studies the attribute similarities of points by employing the Euclidean distance. However, there is a possibility to have more than one neighbouring sample of P_i that belong to the same cluster. In addition, evaluating each one of those to measure their densities is time consuming even though it is faster than the gravity algorithm where density (mass) is based on silhouette measure. Longer time to cluster a new incoming sample is not functional especially if the algorithm is used for online clustering. For that, the proposed algorithm is improved to be faster in clustering new incoming points and to fulfill the objectives of the study. The details of the improvements are discussed below.

1. If the total number of neighbours of $\boldsymbol{P_i}$ is greater than the threshold (MinPts), then the algorithm identifies how many neighbouring samples of $\boldsymbol{P_i}$ belong to the same cluster.
    i. The algorithm checks if there is more than one neighbour in the same cluster. If so, then it calculates the mean of those neighbours in the same

[2]**GrDBAveD Algorithm:** Gravity-Based Density (Mass) Computation using Average Distance, see Section 3.2.3.

cluster to find a mid-point that represents them by the cluster, as shown in Figure 3-8. This process reduces the computation time compared with the idea of evaluating each neighbouring sample of $\boldsymbol{P_i}$, as discussed in details in step 2, see Figure 3-7.

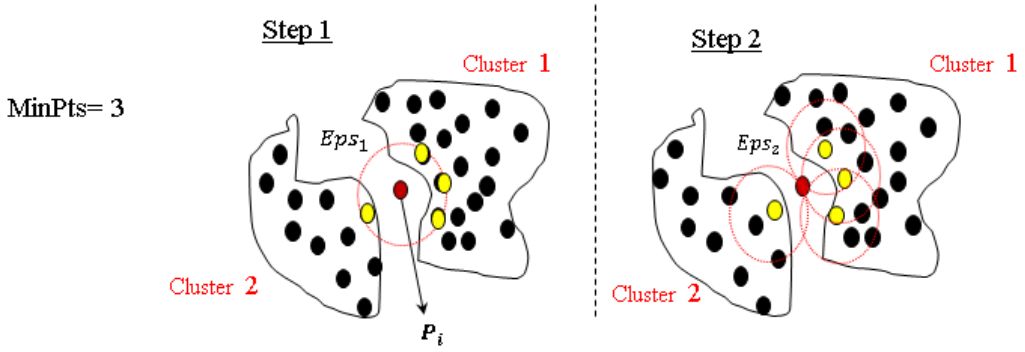

Figure 3-7: Proposed Algorithm(Step 1 and Step 2)

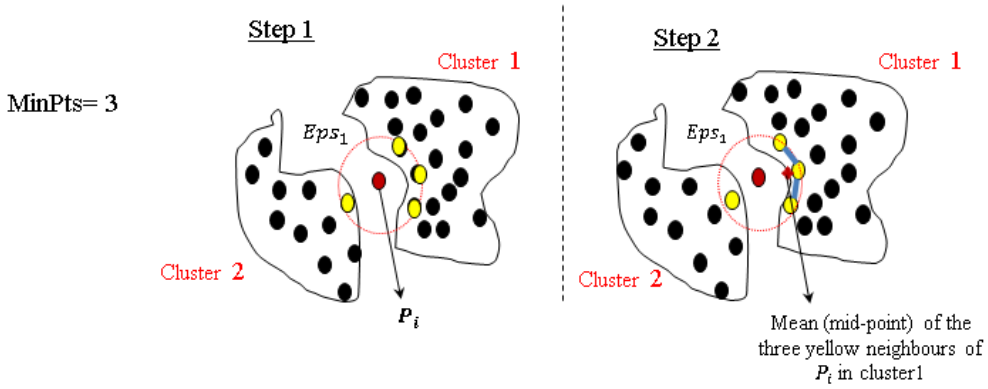

Figure 3-8: Proposed Algorithm After the Improvements

2. **In step 3 and 4**, the distances from $\boldsymbol{P_i}$ to its neighbouring samples, and the distances from the neighbouring samples of $\boldsymbol{P_i}$ to their neighbours are recorded. This process is performed for each neighbouring sample $\boldsymbol{P_i}$. As shown in Figure 3-9, three neighbours of $\boldsymbol{P_i}$ in cluster 1 their densities are calculated separately based on the recorded distance. This process might get longer if there are tens or more neighbours need to be evaluated. However, by finding the mid-point of three neighbours in cluster 1 that saves time by only evaluating one neighbourhood rather than three, as shown in Figure 3-10**. Note:** Equation (3-17) is not performed after the new improvements.

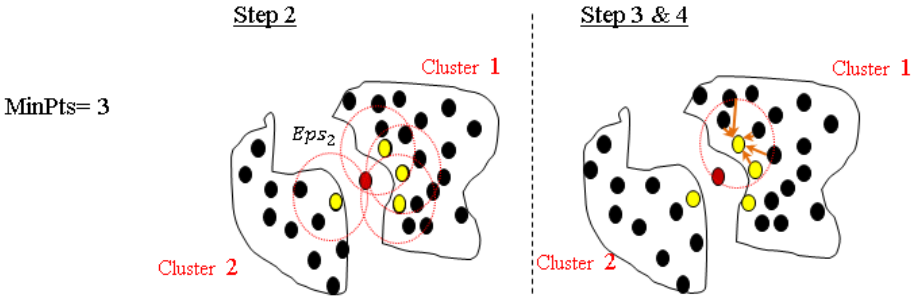

Figure 3-9: Proposed Algorithm(Step 3 and Step 4)

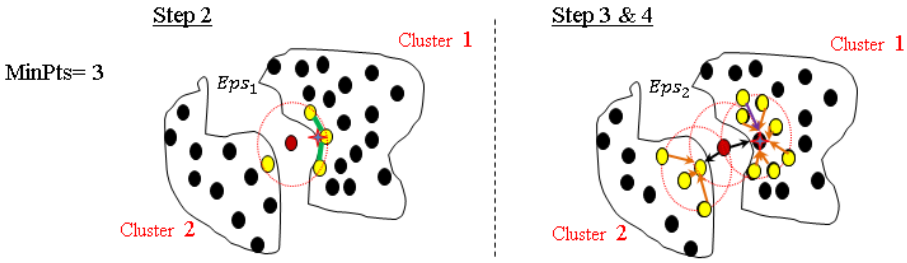

Figure 3-10: Proposed Algorithm After the Improvements

3. The temporary bag is checked every several iteration and if the size of the temporary bag exceeded a threshold. Then, the proposed algorithm creates new clusters by applying DBSCAN clustering algorithm that need to be merged into the existing ones. Otherwise, the algorithm proceeds to the next testing point.

The discussed proposed algorithm adopts the idea of gravity to enable the algorithm to be incremental by inserting new incoming points into the cluster that belongs to. Also, it studies the attributes similarities among points. Another approach is introduced to enable the proposed algorithm to perform another incremental task which is the deletion. The new proposed approach is called Aging, and it is discussed in details in the following section.

3.4 Applying Aging Strategy

Aging concept is used as a method for incremental learning to enable the proposed algorithm to deal with the dynamic environment by adjusting what is learned after grouping the new sample. The basic idea of aging is to assign an age to each point in the clusters. The points' ages are either decreased by one or remain the same when a new incoming sample is clustered. The description of the aging process is presented as follows.

1. A value for the age variable is declared in the beginning.
2. Each point in the clustered training set is assigned an age, see Figure 3-11.
3. A new incoming sample $\boldsymbol{P_i}$ is clustered according to the discussed concepts in 3.2.3 and 3.3, see Figure 3-12.
4. Once $\boldsymbol{P_i}$ is emerged into a cluster then the aging process starts learning about the following:
   a. Identifying which points participated in clustering $\boldsymbol{P_i}$ and to which cluster they belong to.
   b. Which points did not participate in clustering $\boldsymbol{P_i}$.
5. Then Aging method identifies the points that participated in clustering $\boldsymbol{P_i}$ which are all the neighbouring samples of $\boldsymbol{P_i}$ and the neighbours of the neighbouring samples of $\boldsymbol{P_i}$. The age of those points is kept the same without being decrement (the yellow points in Figure 3-13).
6. The rest of points that are in the same clusters as neighbouring samples of $\boldsymbol{P_i}$ and the neighbours of the neighbouring samples of $\boldsymbol{P_i}$ are decreased by 1, as they did not participate in clustering $\boldsymbol{P_i}$ (the black points in same cluster as those participated points in Figure 3-13)
7. If there are other clusters that none of their points among the neighbouring samples of $\boldsymbol{P_i}$, then their ages are the same and do not decrement (as cluster 3 in Figure 3-13).
8. Finally, the new incoming sample $\boldsymbol{P_i}$ is given an age, once it belongs to a cluster, see Figure 3-13.
9. If any of the points' ages become zero, then those points are removed.
10. The process repeats until the last new incoming sample $\boldsymbol{P_i}$.

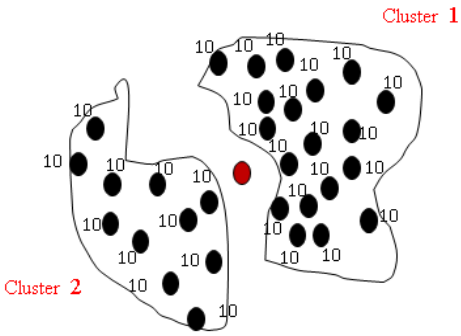

Figure 3-11: Assigning Age to Each Point in the Clusters

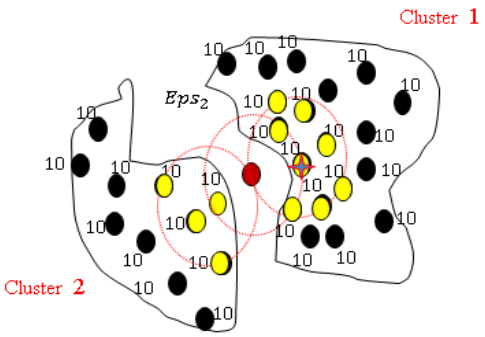

Figure 3-12:Clustering A New Point Based on the Improved GrDBAveD

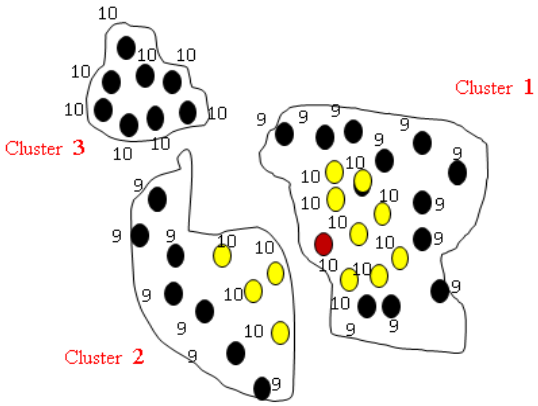

Figure 3-13: Aging Process After Clustering the New Point

The advantages of employing such method in the study are: 1) to give a time window for each point by keeping the points that are frequently participating in clustering the new incoming samples as a reward. In addition, less frequently used points are removed; 2) By removing aged samples, the memory usage decreases, which mean the aging method does utilize the memory efficiently by keeping only the most frequently used points in clustering; and 3) Aging process is part of the incremental learning process, since the proposed algorithm used for clustering incoming data (Online). It enables the algorithm to learn about the clusters behaviours over time. Additionally, the most important points are kept while the rest are removed if they are aged.

3.5 The Complete Architecture of the Proposed Algorithm

Figure 3-14 shows the complete flowchart of the proposed algorithm after applying the improvements to steps 2, 3, and 4 as discussed in Section 3.3. These improvements speed up the clustering process to group the new incoming points faster. The improved **GrDBAveD**[3] algorithm is integrated with aging approach to improve the

[3] **GrDBAveD** Algorithm: Gravity-Based Density (Mass) Computation using Average Distance, see Sections 3.2.3 and 3.3

density-based clustering algorithm using gravity and aging approaches, and it is denoted by DBGrvAge algorithm, see Figure 3-14.

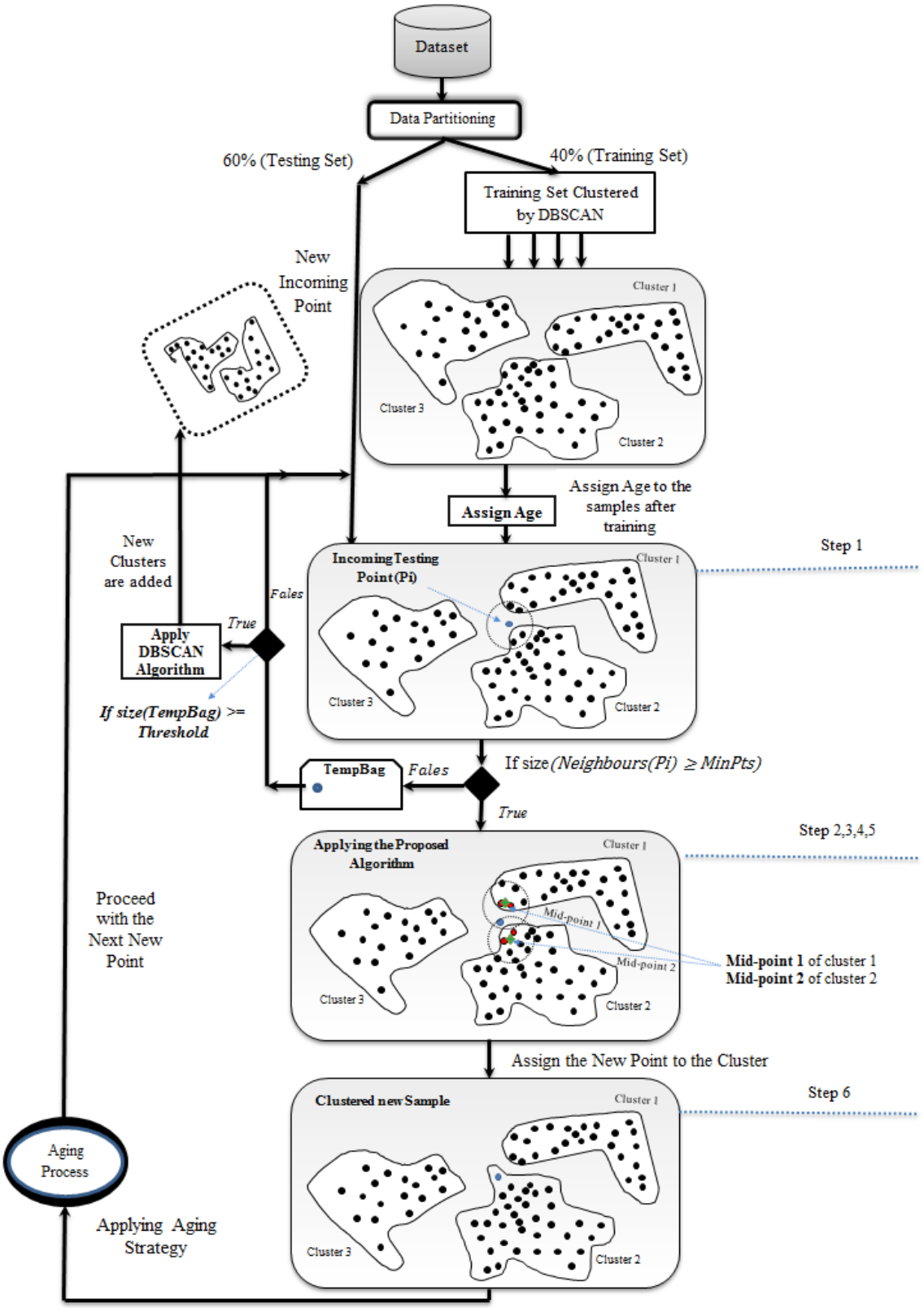

Figure 3-14: The Complete Flowchart of the DBGrvAge Algorithm After Integrating the Aging Approach

3.6 The Pseudo Code of the Proposed Algorithm

According to the Sections 3.2.3 and 3.3, the algorithm Pseudo Code of the improved density-based clustering using gravity concept for the insertion task is proposed in details as follows:

Table 3-1: The Pseudo Code of the Gravity Approach for the DBGrvAge[4] Algorithm

Pseudo Code 1- Gravity approach

Input:

Testing sample = [$\boldsymbol{P_1}$; $\boldsymbol{P_2}$; … ; $\boldsymbol{P_n}$] *// A set of testing samples*

MintPts = ... *// Minimum number of points*

radius1 = ...*// Radius1 of the new incoming sample* $\boldsymbol{P_i}$ *to identify neighbours*

radius2 = ...*//Radius2 of the neighbouring samples of* $\boldsymbol{P_i}$ *to find their neighbours*

// Adding New Incoming Points

Foreach new incoming sample $\boldsymbol{P_i}$ ∈ Testing sample
// calculating the distance from $\boldsymbol{P_i}$ *to all points in the clusters to find the neighbours of* $\boldsymbol{P_i}$
$\boldsymbol{Neighbours_{Eps}(P_i)}$ = Finding the neighbours $\boldsymbol{P_i}$ ≤ radius1
d($\boldsymbol{Neighbours_{Eps}(P_i)}$ **)** *// Distance of all the neighbours of* $\boldsymbol{P_i}$ *are* recorded
size $\boldsymbol{Neighbours_{Eps}(P_i)}$*// and count the total Neighbouring samples of* $\boldsymbol{P_i}$ *within Radius1*
If size($\boldsymbol{Neighbours_{Eps}(P_i)}$) ≥ $\boldsymbol{MinPts}$ **then**
If the condition TRUE
Check if there is more than one neighbouring sample of $\boldsymbol{P_i}$ belonge to the same cluster
Foreach cluster i
If cluster i contains more than one neighbouring sample of $\boldsymbol{P_i}$
Find the **Mean (mid-point)** of the neighbouring samples belongs to the same cluster
Calculate the distance from mid-point to $\boldsymbol{P_i}$
Identify the neighbours of the mid-points within radius 2
Measure the density of the area surrounding the mid-
Point using *den*($\boldsymbol{Neighbour_j}$) // As Equation (3-13 & 3-14)

[4]**DBGrvAge algorithm**: Density-Based Clustering Algorithm Using Gravity and Aging Approaches.

// Find the force of gravity for each neighbour of $\boldsymbol{P_i}$ that represent a cluster

$\boldsymbol{ForceGrv}\left(\boldsymbol{Neighbour_j}\right)$// As Equations (3-16)

Save $\boldsymbol{ForceGrv}$ of mid-point represented by cluster i

ElseIf the cluster i has one neighbouring sample of $\boldsymbol{P_i}$

Find the neighbouring samples of $\boldsymbol{Neighbour_j} \leq$ radius2

Measure the density of the area surrounding the neighbour of $\boldsymbol{P_i}$ by *den*($\boldsymbol{Neighbour_j}$) // As Equation (3-13 & 3-14)

// Find the force for each neighbouring sample of $\boldsymbol{P_i}$

$\boldsymbol{ForceGrv}\left(\boldsymbol{Neighbour_j}\right)$// As Equations (3-16)

Save $\boldsymbol{ForceGrv}$ of the neighbouring sample of $\boldsymbol{P_i}$ represented by its cluster i

Else if cluster i does not have any neighbouring sample for $\boldsymbol{P_i}$ then

Set $\boldsymbol{ForceGrv}$ of cluster i as Zero

End

End

Based on the maximum force of the cluster, add the new incoming point $\boldsymbol{P_i}$ to the cluster with the highest value of force, as Equation (3-18)

Update the IDX, Training, and testing matrix with the new information of $\boldsymbol{P_i}$

Else

Put the $\boldsymbol{Neighbour_j}$ that has less neighbours than MinPts in temporary Bag for later analysis

Then

Delete $\boldsymbol{Neighbour_j}$ from the testing set

End

// Creating new cluster

Every # of testing samples

If temporary Bag exceeds a threshold

Apply DBSCAN algorithm to create new clusters

End

Proceed to the next new incoming point

End

When the improvements are applied on the density-based clustering algorithm, the aging method is employed right after the incoming sample $\boldsymbol{P_i}$ is added to a cluster. Before proceeding with the next incoming sample, the aging method updates the age of all the points in the clusters after clustering $\boldsymbol{P_i}$, also the aging process checks if there is any point with age zero to be removed. The Table 3-2 presents the pseudo code for the steps taken to perform the aging process for the deletion task.

Table 3-2: Pseudo Code of the Aging Approach for the DBGrvAge Algorithm

Pseudo Code 2- Aging Approach

Input:

age = ... *// give a value for age*
Foreach sample in the training set *// declared at the beginning of the program*
    aging(training Set) = age *// ageing matrix that keep track of the points age*
End

1 ***Within "if*** size($\boldsymbol{Neighbours_{Eps}(P_i)}$) $\geq$ $\boldsymbol{MinPts}$" *condition the aging method take*
2 *place*
3 All the points that participated in clustering $\boldsymbol{P_i}$ are saved in the following variables
4 allPointsParticClust = [$\boldsymbol{Neighbour_j}$; $\boldsymbol{neighbours_{(Neighbour_j(P_i))}}$]
5 *// Put all points in the matrix*
6 unique(allPointsParticClust) *// To remove any redundant point*
7 The points that belong to the same cluster as $\boldsymbol{Neighbour_j}$ of $\boldsymbol{P_i}$, but they are not among
8 $\boldsymbol{neighbours_{(Neighbour_j(P_i))}}$
9 means do not participate in clustering $\boldsymbol{P_i}$
10 **Then**
11 All points except *allPointsParticClust* in same cluster = Age -1 (minus 1)
12 *allPointsParticClust* same age is the same (no decrement)
13 clusters that none of their points among $\boldsymbol{Neighbour_j}$ of $\boldsymbol{P_i}$ then their ages are
14 the same (no decrement).
15 Add the age of the new clustered point $\boldsymbol{P_i}$
16 Check if any of the rows of the aging matrix contain value 0
17 indexNo = find(ageing == 0);*// find the index number of point that aged or expired*
18 trainingSet(indexNo,:)= [];*// Delete the point that is indicated by index from training set*
19 IDXTr(indexNo,:)= [];*// Delete the record that belongs for that point based on index*
20 ageing(indexNo,:)= [];*// Delete the record that belongs for that point based on index*

End

In this section, we have discussed our proposed algorithm that improves the performance of the density-based clustering algorithm using two approaches that are gravity and aging. Both approaches enhance the density-based clustering to be flexible while dealing with a dynamic behaviour (online incoming data). The experimental result and evaluation of the proposed algorithm are presented in the next Chapter 4.

3.7 Chapter Summary

This chapter discussed the architecture of proposed algorithms along with their definitions employed to achieve the objectives of the study. Besides their definitions, the chapter also provides a description of the methods used to make the proposed algorithm to be part of an incremental solution for clustering using the gravity and aging approaches. Also, it provides the Pseudo Code that provides a description of how the algorithm works. It highlights the final proposed algorithm after integrating the aging strategy, and it is called as "An Improved Density-Based Clustering Algorithm using Gravity and Aging Approaches" denoted by **DBGrvAge** algorithm.

CHAPTER 4. EXPERIMENTAL RESULTS

This chapter presents the experimental results of the proposed algorithm along with a comparison of the performance according to the evaluations metrics discussed in Section 4.1. The evaluation metrics section is divided into two sub-sections: 1) The ***performance measures*** which are used to evaluate the algorithms including Dunn Index and SD Index; 2) The ***algorithm performance*** based on execution time, memory usage, and sensitivity. Then Section 4.2 provides a description of the datasets that are involved in the experiments. Section 4.3 presents experimental results on several datasets. Finally, Section 4.4 provides a discussion of the experimental results.

4.1 Evaluation Metrics

The performances of the clustering algorithms are different from each other even if they are tested on the same dataset. Liu et al. (2010) discussed two main criteria to evaluate the goodness of the clustering algorithm which are: 1) ***Compactness***: the data points within each cluster should be close to each other as much as possible; and 2) ***Separation***: clusters should be well-separated from each other. In this study, the proposed algorithm is evaluated according to the clusters performance measurements and the algorithm overall performance. The details are discussed in Sections 4.1.1 and 4.1.2.

4.1.1 Performance Measurements

The performance of the proposed algorithm is evaluated by employing the following four internal performance measures:

- ***Dunn Index***

  Dunn Index is an internal validity index that measures the compactness and how well-separated the clusters are. The index formula is presented in Equation (4-1) (Legány, Juhász, & Babos, 2006).

$$Dunn = \min_{i=1\ldots n_c} \left\{ \min_{j=i+1\ldots n_c} \left(\frac{d(c_i, c_j)}{\max_{K=1\ldots n_c} \left(diam(c_k) \right)} \right) \right\} \quad (4\text{-}1)$$

  Where n_c is the number of clusters, and d is the distance between two objects. d (c_i, c_j) is the dissimilarity function that measure the distance between two clusters c_i and c_j, as defined by the following Equation (4-2), and $diam(c_k)$ is the

diameter of the cluster c, measures the clusters' dispersion. The diameter of a cluster c is defined in Equation (4-3) (Halkidi, Batistakis, & Vazirgiannis, 2002):

$$\boldsymbol{d(c_i, c_j) = \min_{x \in c_i, y \in c_j} \{d(x, y)\}} \quad (4\text{-}2)$$

$$\boldsymbol{diam(c_i) = \max_{x, y \in c_i} \{d(x, y)\}} \quad (4\text{-}3)$$

When clusters are compact, their diameters are small, hence the higher value of the Dunn Index. Similarly, the larger the distance between the two clusters, the most separated the two clusters are, and the higher value of the Dunn index. Accordingly, a high value of the Dunn Index indicates the presence of compact and well-separated clusters (Halkidi et al., 2002).

- ***Davies-Bouldin Index (DB)***

Davies-Bouldin Index is another internal validity index that is based on the similarity measure $\boldsymbol{R_{ij}}$ between clusters $\boldsymbol{c_i}$ and $\boldsymbol{c_j}$. It is defined according to the measure of dispersion $\boldsymbol{s_i}$ of cluster $\boldsymbol{c_i}$. In addition, $\boldsymbol{d_{ij}}$ is the dissimilarity measure between two clusters. The $\boldsymbol{R_{ij}}$ is defined to satisfy the following conditions (Halkidi et al., 2002; Legány et al., 2006):

- $\boldsymbol{R_{ij} \geq 0}$	Indicates that $\boldsymbol{R_{ij}}$ is non-negative
- $\boldsymbol{R_{ij} = R_{ji}}$	Indicates symmetric relation
- $\boldsymbol{if\ s_i = 0}$ **and** $\boldsymbol{s_j = 0}$ **then** $\boldsymbol{R_{ij} = 0}$	Indicates 2 points are always unique
- $\boldsymbol{if\ s_j > s_k}$ **and** $\boldsymbol{d_{ij} = d_{ik}}$ **then** $\boldsymbol{R_{ij} > R_{ik}}$	A well-dispersed cluster is more similar to all other clusters than a tight cluster.
- $\boldsymbol{if\ s_j = s_k}$ **and** $\boldsymbol{d_{ij} < d_{ik}}$ **then** $\boldsymbol{R_{ij} > R_{ik}}$	The closer the clusters are, the more similar.

To satisfy these conditions, $\boldsymbol{R_{ij}}$ is measured by the following Equation (4-4).

$$\boldsymbol{R_{ij} = \frac{(s_i + s_j)}{d_{ij}}} \quad (4\text{-}4)$$

The dissimilarity measure $\boldsymbol{d_{ij}}$ between two clusters and measure of dispersion $\boldsymbol{s_i}$ of cluster $\boldsymbol{c_i}$ are defined in the following Equations (4-5) and (4-6)

$$\boldsymbol{d_{ij} = d(v_i, v_j)} \tag{4-5}$$

$$\boldsymbol{s_i = \frac{1}{\|c_i\|} \sum_{x \in c_i} d(x, v_i)} \tag{4-6}$$

Where $d(v_i, v_j)$ is the distance between the center point of the cluster i^{th} and cluster j^{th}. And $\|c_i\|$ is the number of elements in the cluster i^{th}. $\boldsymbol{d(x, v_i)}$ is the distance from each point in cluster i^{th} to the cluster's center. Then the DB index is defined as Equation (4-7)

$$\boldsymbol{DB = \frac{1}{n_c} \sum_{i=1}^{n_c} R_i} \tag{4-7}$$

Where n_c is the number of clusters, and R_i is defined in Equation (4-8)

$$\boldsymbol{R_i = \max_{j=1 \ldots nc, i \neq j} (R_{ij}) \ , i = 1 \ldots n_c} \tag{4-8}$$

The lower the Davies-Bouldin index means the clusters are not very similar to each other indicating that they are well-separated.

- ***Sum Square Error (SSQ)***

  SSQ measures the distance from each point in the cluster to the center of its cluster to assess how dense the cluster is. The SSQ formula is presented in Equation (4-9) (Quan, Xiao, & Zhang, 2013).

$$\boldsymbol{SSQ = \frac{\sum_{i=1}^{K} \sum_{j=1}^{N} |d(x_{ij}, v_i)|}{K}} \tag{4-9}$$

  Where $\boldsymbol{x_{ij}}$ is the point j^{th} in the cluster i^{th}, $\boldsymbol{v_i}$ is the clusters' center, and $\boldsymbol{K}$ is the total number of clusters. $\boldsymbol{d(x_{ij}, v_i)}$ is the distance from each point in cluster i^{th} to it is centers $\boldsymbol{v_i}$. The best performance of SSQ is indicated by a lower value of SSQ, which indicates a higher the concentrations of the clusters.

- ***SD Validity Index***

  SD Index is measured based on the average scattering of the clusters and the total separation of the clusters.

- ✓ **Average Scattering** is calculated according to the variance of the clusters and the variance of the entire dataset. Through the average scattering, the SD index can measure the homogeneity and the compactness of the clusters. The average scattering is defined in Equation (4-12) (Kovács, Legány, & Babos, 2005; Mohammadi, Raahemi, Akbari, Nassersharif, & Moeinzadeh, 2012)

$$\textbf{\textit{Dataset Variance}} = \|\boldsymbol{\sigma}(\boldsymbol{X})\| \tag{4-10}$$

$$\boldsymbol{Cluster\ Variance} = \sum_{i=1}^{c} \|\boldsymbol{\sigma}(\boldsymbol{v_i})\| \tag{4-11}$$

$$\textbf{\textit{AvgScatt}} = \frac{1}{\boldsymbol{Total\ num\ of\ clusters}} (\boldsymbol{Cluster\ Variance\ /Dataset\ Variance}) \tag{4-12}$$

Where σ is the variance of X (the whole dataset). $\|\sigma(v_i)\|$ is the variance of each cluster, in which v_i is the center of the cluster i^{th}.

- ✓ **Total separation** of clusters is based on the distance between the clusters' center points. It is defined by Equation (4-13) (Mohammadi et al., 2012).

$$\boldsymbol{TotalSeparation} = \frac{\boldsymbol{D_{max}}}{\boldsymbol{D_{min}}} \sum_{i=1}^{c} \left(\sum_{n=1}^{c} \|\boldsymbol{v_i} - \boldsymbol{v_n}\| \right)^{-1} \tag{4-13}$$

where

$$\boldsymbol{D_{max}} = \boldsymbol{max}\|\boldsymbol{v_i} - \boldsymbol{v_n}\|$$

$$\boldsymbol{D_{min}} = \boldsymbol{min}\|\boldsymbol{v_i} - \boldsymbol{v_n}\|$$

D_{max} and D_{max} are the maximum and minimum distance between clusters' centers. According to that, the SD Index is then defined as follows (4-14):

$$\textbf{\textit{SD}} = \boldsymbol{\alpha} * \textbf{\textit{AvgScatt}} + \textbf{\textit{TotalSeparation}} \tag{4-14}$$

Based on Equation (4-14), α is the weighting factor is equal the total separation parameter in case of maximum number of clusters. Lower SD index indicates the presence of compact and well-separated cluster (Kovács et al., 2005; Mohammadi et al., 2012).

4.1.2 Algorithm Performance

The overall performance of the proposed algorithm is evaluated in terms of the algorithm's execution time, memory usage, and sensitivity (Amini, Wah, & Saboohi, 2014).

- **Execution Time**: The algorithm performance is evaluated based on the total time taken to cluster the new incoming samples, and since the proposed algorithm is used online, then the faster the algorithm is the better.
- **Memory Usage:** It is the amount of memory used by the algorithm to keep points. Since the proposed algorithm is used for online clustering, then it should be able to minimize the memory usage.
- **Sensitivity:** It measures how sensitive is the algorithm to the input parameters of the algorithm. Also, how it affects the clustering performance and what is the best ranges of the algorithm's parameters are (Amini et al., 2014).

Having briefly described the performance metrics that are used to evaluate the proposed algorithm, the following section explains the structure of the datasets used to perform the experiments.

4.2 Data Description

The result of the study is obtained by validating the performance of the proposed algorithm on several datasets. Validating the algorithm on several datasets is essential in order to measure the performance of the proposed algorithm and whether the results are achieved according to the objectives discussed earlier. There are two types of datasets which are:

- ✓ **Synthetic Dataset** is used to validate the performance of the proposed algorithms. It is generated from populations not from direct measurements. For this study, we created a synthetic data with the following properties; attributes are numeric, the data consists of 1500 observations with two attributes (features) and the third attribute is the label (that consists of three classes that represent 3 different clusters). It is mainly used to visualize the changes happening to the clusters' shapes over time.

- ✓ **Real Datasets** are obtained from UCI machine learning repository that contains around 264 datasets of different applications from

(*http://archive.ics.uci.edu/ml/index.html*). The datasets that are considered from the repository for the study are Iris, Wine, Ecoli, Diabet and Segment. Another dataset is P2P Internet traffic dataset (generated at the University of Ottawa). Each dataset contains a number of properties. These properties are data type, number of instances, number of attributes, and the number of classes. Table 4-1 discusses in details those properties to provide a better understanding of the structure of datasets being used.

Table 4-1: Datasets Description

	Dataset Properties					
Properties / Datasets	**Data Type**	**No. Instances**	**No. Attributes**	**No. Classes**	**Missing Values**	**Area /Application**
Iris	Numeric (Real)	150	4	3	No	Life
Wine	Numeric (Real & Integer)	178	13	3	No	Physical
Ecoli	Numeric (Real)	327	8	5	No	Life
Diabet	Numeric (Real & Integer)	768	8	2	Yes	Life
Segment	Numeric (Real & Integer)	2310	19	7	No	N/A
P2P	Numeric (Real & Integer)	32,767	4	2	No	Network Traffic

These datasets are used to evaluate the proposed algorithm in order to examine how the algorithm performs with datasets of various sizes and high/low dimensions (attributes). Before conducting the experiments, the datasets need to be pre-processed first by transforming the raw dataset into a structured format and understandable by the algorithm to guarantee high-quality results. More details are presented in Section 4.2.1.

4.2.1 Data Pre-processing

The data pre-processing phase lists all the tasks that are required to construct the raw data into structural datasets to improve the efficiency of the proposed algorithm. Figure 4-1 shows the two primary tasks needed to prepare the datasets to be ready for clustering:

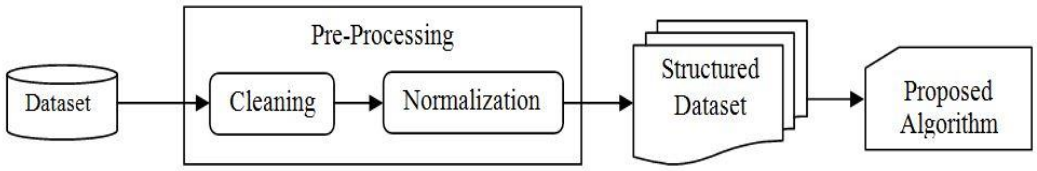

Figure 4-1: Data Pre-processing

- ✓ **Data Cleaning:** The datasets are cleaned from any missing and inconsistent values.
- ✓ **Data Transformation Method:** is performed by applying normalization to make all the data values to fall within a small range between either (0-1) or (0-100). It enables the proposed algorithm to perform efficiently. Normalization is very useful in this study, since it uses the distance measurements. The selected normalization technique is **min-max normalization.** It performs a linear transformation on the original data by setting the minimum value of 0 and maximum value of 1 or 100. "*The minimum and maximum values are represented in the formula* (4-15) *with* min_A *and* max_A *of an attribute A. Min-max normalization maps a value,* v_i *of A to* v_i' *in the range [*new_max_A*,*new_min_A*]*" (Han et al., 2011)

$$\boldsymbol{v'} = \frac{\boldsymbol{v - min_A}}{\boldsymbol{max_A - min_A}} (\boldsymbol{new\_max_A} - \boldsymbol{new\_min_A}) + \boldsymbol{new\_min_A} \qquad (4\text{-}15)$$

After pre-processing tasks, the structured datasets are ready to be used by the proposed algorithm to evaluate its performance. Datasets are used by the algorithm one at a time to validate the performance of algorithms. More description of the experimental results is in the following section.

4.3 Experimental Results

Several versions of the proposed algorithm are introduced in Chapter 3 that are employed to investigate the different aspects and examine which one fulfills most of the study's objectives. Through these versions, we are trying to obtain new measurements that enable the proposed algorithm to study the attributes similarities among points, and to allow it to cluster dynamic data. The performance of the proposed versions is evaluated in this section through a series of experiments conducted on 6 real datasets and a synthetic dataset.

The experiments focus on evaluating the proposed algorithm for its: 1) arbitrary shapes; 2) memory usage; 3) running time; and 4) performance measures. The experimental results section is divided into several sub-experiments which are: 1) experiment-1 presents the comparison among the four versions of the proposed algorithm to select the one with best performance; 2) Experiment-2 presents a comparison between the selected proposed algorithm and the well-known algorithms K-means and DBSCAN; 3) Experiment-3 visualizes the arbitrary shapes when the proposed algorithm is integrated with the aging strategy and validated on the synthetic dataset. Moreover, experiment-3 illustrates the results of the proposed algorithm integrated with the aging strategy on the real datasets; 4) Finally, experiment-4 that applies the proposed algorithm on dataset of real-time application to show how the algorithm can work incrementally and compare it with the results of DBSCAN and K-means algorithms. The setup of the experiments is presented in Figure 4-2.

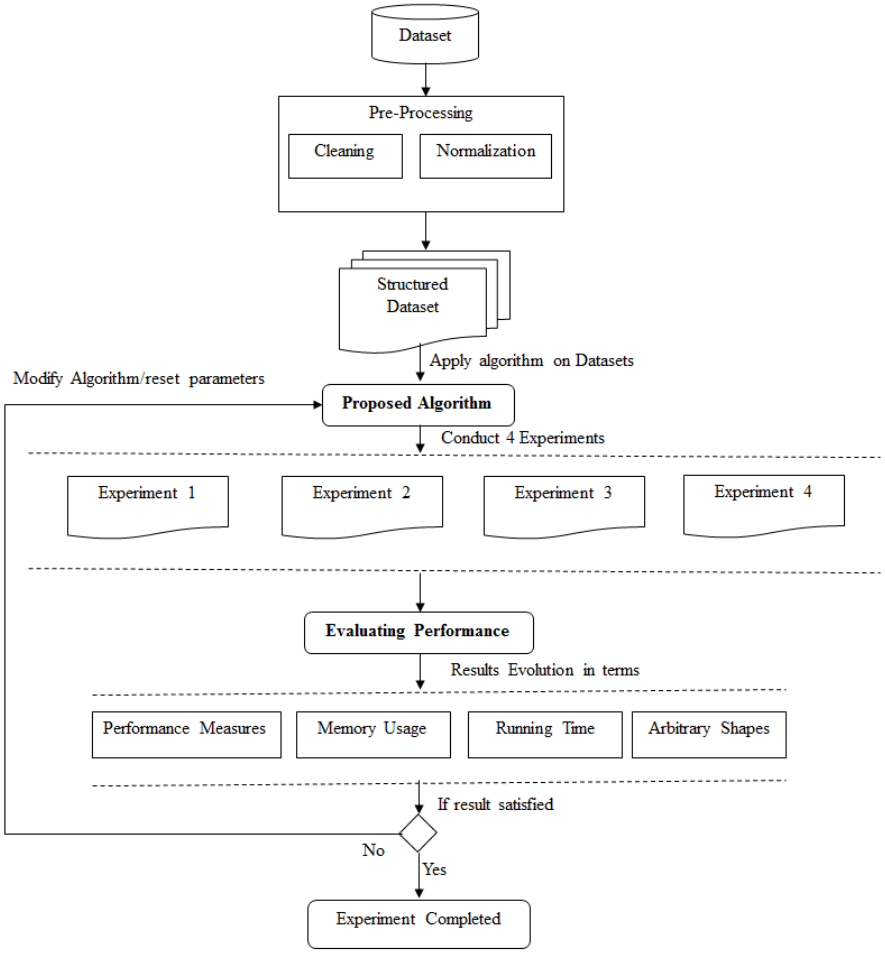

Figure 4-2: The Setup of the Experiments

According to the experimental setup of Figure 4-2, the experiments are conducted on the datasets after being pre-processed, and results are assessed according to the evaluation metrics. If the results of the algorithm are reasonable, then the experiments are considered to be completed. Otherwise, several modifications might be made to the input parameters or to the algorithm itself to adjust the performance of the algorithm. The first experiment is conducted, and details are presented as follows.

4.3.1 Experiment -1

This section compares the performance of the first four versions of the proposed algorithm discussed in Sections 3.1 and 3.2 with the final selected algorithm. The selected algorithm implies Newton's law of gravity to study the similarities among points and to insert new incoming points into their clusters. By adopting the gravity concept, we can build a density-based clustering algorithm that can be part of an incremental solution for clustering.

Experiment-1 is conducted on five datasets that are Iris, Wine, Ecoli, Diabet, and Segment. Then, the results of each algorithm are assessed by the performance measures (Dunn Index, Square Sum Error (SSQ), Davies-Bouldien (DB), and SD Index) and the running time, as discussed in 4.1.1 and 4.1.2. Table 4-2 describes the abbreviations used through the experimental tables.

Table 4-2: The Abbreviations Used in the Tables throughout the Experiments

Abbr.	Indicates
PM	It means Performance Measures.
DBSCAN Algorithm	It stands for the well-known algorithm density-based spatial clustering of applications with noise. It defines its clusters based on the high dense regions separated by regions of low density, see Section **2.2**.
K-means Algorithm	It divides data into a set of clusters, where data is partitioned into several subsets by grouping samples into their closest center called "*centroid*".
DBNPR Algorithm	The *density* is calculated based on the total number of points within the radius, see Section 3.1.2.
GrDBNPR Algorithm	Gravity is introduced where the *density* (mass of gravity) is calculated based on the total number of points within the radius, and dividing it by the square distance, see Section **3.2.1**.

Abbr.	Indicates
GrDBSil Algorithm	Gravity is introduced where the *density* (mass of gravity) is calculated based on the silhouette measure, and dividing it by the square distance, see Section **3.2.2**.
GrDBAveD Algorithm	Gravity is introduced where the *density* (mass of gravity) is calculated based on the average distance of each neighbouring samples to its neighbours, and dividing it by the square distance, see Section **3.2.3.**
Improved GrDBAveD Algorithm	It represents the final improved algorithm where Gravity is introduced, and the *density* (mass of gravity) is calculated based on the average distance of the mid-point to its neighbours. Mid-point is the mean of the neighbours presented in the same cluster. Then, dividing it by the square distance, see Section **3.3**
DBGrvAge Algorithm	It represents the improved GrDBAveD algorithm after integrated with the aging approach, see Sections **3.2.3, 3.3, 3.4,** and **3.5**.
MinPts	It is the minimum number of points that must be available within the radius. If the total number of points within the radius is equal or more than four, then the algorithm proceeds with the next operation. Otherwise, the point is moved to a temporary bag for later evaluation (which then might be classified as noise).
Radius	Three radiuses are considered for each dataset: Radius-1 is used for the DBSCAN applied at the training phase; Radius-2 is to identify the neighbours of the new incoming sample; and Radius-3 is to determine how dense is the neighbourhood of the new point's neighbours
K	Number of clusters

Note: *more description of the algorithms are presented in chapter 3 (Section 3.1,3.2, and 3.3).*

Table 4-3 highlights the input values used by the algorithms for each dataset. The MinPts is the minimum number of points within the radius used by each algorithm, and it is the same for all the datasets. The dataset is divided into 40% for training phase to

create the initial clusters for the offline using the DBSCAN algorithm, and the rest of the dataset (60% for the testing) acts as the new incoming samples (as a dynamic data) that are generated online to test if the algorithms can cluster them incrementally.

Table 4-3: Parameters Setup of the Algorithms for Each Dataset in Experiment-1

Inputs / Datasets	Proposed Algorithms			
	Parameter		Data Segment	
	MinPts	Values for Radiuses 1,2,3	Training 40%	Testing 60%
Iris	4	0.25, 0.25 & 0.3	60	90
Wine	4	50, 65 & 70	71	107
Ecoli	4	0.18, 0.34 & 0.4	131	196
Diabet	4	30, 38 & 40	307	461
Segment	4	0.45, 0.5 & 0.55	924	1386

Table 4-4 shows how to interpret the values of each performance measures (PMs) in order to evaluate the performance of the algorithms.

Table 4-4: Performance Indication of the Evaluation Metrics

PM	High/Low	Performance Indication
Dunn Index	↑	The higher the value of Dunn Index, the higher the performance of the algorithm.
SSQ	↓	The lower the value of SSQ, the higher the performance of the algorithm.
DB Index	↓	The lower the value of DB Index, the higher the performance of the algorithm.
SD Index	↓	The lower the value of SD Index, the higher the performance of the algorithm.
Time	↓	The lower the value of time, the faster the algorithm.

The following tables present the results of experiment-1 based on the various performance measures employed to evaluate the algorithms.

Each value in the tables represents the average of 20 iterations (or runs) per algorithm on a dataset. The **bolded and underlined** values represent the best results achieved for each dataset according to the values of the performance measures of each algorithm (it is a column-based comparison for each row). While the **bolded** values are the second best results in a row.

- **Dunn Index:** According to Table 4-5**,** the Improved GrDBAveD has the highest value of Dunn Index compared with other algorithms. Also, it shows that the Improved GrDBAveD performs well in clustering all the datasets of various sizes and number of features. It is obvious from the results that Improved GrDBAveD has clusters that are compact and well-separated compared with the other algorithms.

Table 4-5: Experiment-1 the Average Dunn Index of the Proposed Algorithms

Dunn Index					
Algorithms / Datasets	**DBNPR**	**GrDBNPR**	**GrDBSil**	**GrDBAveD**	**Improved GrDBAveD**
Iris	0.424	**0.429**	0.419	0.404	**0.430**
Wine	0.263	0.275	**0.300**	0.296	**0.310**
Ecoli	**0.113**	**0.109**	0.083	0.097	**0.113**
Diabet	0.230	**0.230**	0.192	0.208	**0.263**
Segment	0.259	0.186	**0.259**	0.254	**0.265**

- **Davies–Bouldin Index:** According to Table 4-6, the Improved GrDBAveD has either the same or higher value of Davies–Bouldin Index compared with the other algorithms for most of the dataset. However, it does not have a lower value for the Diabet dataset, but overall performance is still better in most datasets.

Table 4-6: Experiment-1 the Average Davies–Bouldin Index of the Proposed Algorithms

Davies–Bouldin Index					
Algorithms / Datasets	**DBNPR**	**GrDBNPR**	**GrDBSil**	**GrDBAveD**	**Improved GrDBAveD**
Iris	**0.464**	**0.462**	0.466	0.465	**0.462**
Wine	1.126	1.209	1.119	**1.090**	**1.081**
Ecoli	0.865	**0.789**	1.358	0.927	**0.858**
Diabet	1.036	**1.018**	1.038	**0.997**	1.026
Segment	0.761	0.807	**0.726**	0.789	**0.629**

- **Sum Square Error (SSQ):** According to Table 4-7**,** the Improved GrDBAveD has either the same or lower value of SSQ compared with the other algorithms for Iris, Wine, and Segment. However, it does not perform well with Ecoli and Diabet dataset as **GrDBAveD** algorithm. But overall performance is reasonable.

Table 4-7: Experiment-1 the Average Sum Square Error (SSQ) of the Proposed Algorithms

Sum Square Error (SSQ)					
Algorithms / Datasets	**DBNPR**	**GrDBNPR**	**GrDBSil**	**GrDBAveD**	**Improved GrDBAveD**
Iris	**16**	**16**	**16**	**16**	**16**
Wine	3831	3261	3697	**3153**	**3060**
Ecoli	39	**25**	49	**32**	38
Diabet	**13151**	13458	13469	**12487**	13374
Segment	299	235	**234**	**234**	**230**

- **SD Index:** According to Table 4-8**,** the Improved GrDBAveD has either the same or lower value of SD index compared with other algorithms for most of the dataset. However, it does not have a lower value for the Wine dataset, but overall performance is still better. Similar to the Dunn Index, it shows that the clusters are well-separated and compact in most of the datasets in comparison with the rest of the algorithms.

Table 4-8: Experiment-1 the Average SD Index of the Proposed Algorithms

SD Index					
Algorithms / Datasets	**DBNPR**	**GrDBNPR**	**GrDBSil**	**GrDBAveD**	**Improved GrDBAveD**
Iris	**1.184**	**1.183**	1.185	1.185	**1.183**
Wine	**1.551**	1.590	1.578	**1.575**	1.579
Ecoli	**1.415**	1.541	1.606	1.545	**1.406**
Diabet	1.734	**1.724**	**1.726**	1.789	**1.724**
Segment	**1.437**	1.987	1.853	1.836	**1.825**

- **Running Time:** According to Table 4-9**,** the Improved GrDBAveD is faster in clustering incoming points in comparison with the other algorithm. Since, the Improved GrDBAveD checks if there are more than one neighbour that belongs to the same cluster. If there are, then it calculates the mean of those neighbours in the same cluster to find a mid-point that represents them, instead of evaluating each neighbour as DBNPR, GRDBNPR, GrDBSil, and GrDBAveD are doing. That is what makes the performance of the Improved GrDBAveD faster than others.

Table 4-9: Experiment-1 the Average Running Time of the Proposed Algorithms

Running Time						
Datasets \ **Algorithms**		**DBNPR**	**GrDBNPR**	**GrDBSil**	**GrDBAveD**	**Improved GrDBAveD**
Iris	**All Points**	**2.614**	5.372	17.527	4.054	**0.354**
	Per-Point	**0.029**	0.060	0.194	0.045	**0.0039**
Wine	**All Points**	**3.123**	6.791	23.695	4.865	**0.887**
	Per-Point	**0.029**	0.063	0.221	0.045	**0.0082**
Ecoli	**All Points**	**23.81**	40.70	716.28	34.62	**2.50**
	Per-Point	**0.121**	0.208	3.65	0.176	**0.013**
Diabet	**All Points**	215	362	18070	**190**	**8.25**
	Per-Point	0.466	0.785	39.20	**0.412**	**0.018**
Segment	**All Points**	**1842**	1950	113470	2165	**42**
	Per-Point	**1.329**	1.407	81.87	1.562	**0.030**

All Points- indicates the total time taken to cluster all the new incoming points in seconds.

Per-Point- indicates the time taken to cluster a single new incoming point in seconds.

Experiment-1 concludes that the Improved GrDBAveD provides a better performance in terms of how well the clusters are separated and compact according to the performance measures in most datasets. Most importantly, it clusters samples faster than other algorithms. The total number of clusters produced by the algorithms is the same for each dataset. Based on the outcomes, the Improved GrDBAveD is chosen as the proposed

algorithm for this study because it can cluster dynamic data faster and with a better performance, while studying the attributes similarities among points.

The next experiment compares the performance of the selected Improved GrDBAveD with other well-known clustering algorithms. Details are presented in the following experiment.

4.3.2 Experiment -2

This section compares the performance of the selected algorithm (Improved GrDBAveD.) discussed in Section 4.3.1 with two well-known clustering algorithms DBSCAN and K-means algorithms. The selected Improved GrDBAveD algorithm implies Newton's law of gravity to study the similarities among points, and to insert new incoming points into their clusters one at a time when dealing with a dynamic environment. On the other hand, DBSCAN and K-means cluster the entire dataset at once.

Table 4-10 lists the parameters set up of the algorithms for each dataset. The MinPts is the minimum number of points within the radius used by DBSCAN and Improved GrDBAveD, and the Radius is specified for both DBSCAN and Improved GrDBAveD algorithms. K-means only needs K (i.e. the number of clusters). The DBSCAN and K-means algorithms cluster the whole dataset at once without data partitioning. While, the improved GrDBAveD has to divide the datasets into 40% of the total dataset size for training to create the initial clusters using the DBSCAN algorithm, and the rest of the dataset acts as the new incoming samples generated online to validate if the algorithms are incremental.

Table 4-10: Parameters Setup of the Algorithms for Each Dataset in Experiment-2

	Algorithms				
	K- means	**DBSCAN**		**Improved GrDBAveD**	
Parameters / Datasets	**K**	**MinPts**	**Radius**	**MinPts**	**Radius**
Iris	2	4	0.25	4	0.25, 0.25 &0.3
Wine	5	4	50	4	50, 65 & 70
Ecoli	3	4	0.123	4	0.18, 0.34 &0.4
Diabet	4	4	30	4	30, 38 & 40
Segment	4	4	0.5	4	0.45, 0.5 &0.5

Table 4-11 to Table 4-14 show the results of experiment-2 to evaluate which algorithm performs better in terms of the performance measures. Each value in these tables represents the average of 20 iterations (or runs) for each algorithm.

The **bolded and underlined** values represent the best results achieved for each dataset according to the values of the performance measures of each algorithm (it is a column-based comparison for each row). While, the **bolded** values are the second best results in a row. The comparison between the proposed algorithm (Improved GrDBAveD that clusters new points incrementally) and (K-means and DBSCAN algorithms) is discussed according to the performance measures.

- **Dunn Index:** According to Table 4-11**,** the Improved GrDBAveD has almost the highest value of Dunn Index compared with the other algorithms. It shows that Improved GrDBAveD performs well in clustering dataset of various sizes and number of features compared with K-means and DBSCAN. It is obvious from the results that Improved GrDBAveD has clusters that are compact and well-separated.

Table 4-11: Experiment-2 the Average Dunn Index of the Improved GrDBAveD and (K-means & DBSCAN Algorithms)

Dunn Index			
Algorithms / Dataset	**K-mean**	**DBSCAN**	**Improved GrDBAveD**
Iris	**0.358**	**0.358**	**0.426**
Wine	0.189	**0.309**	**0.310**
Ecoli	0.087	**0.146**	**0.113**
Diabet	0.060	**0.247**	**0.263**
Segment	0.020	**0.259**	**0.266**

- **Davies–Bouldin Index:** According to Table 4-12**,** the Improved GrDBAveD performs well with Iris and Wine datasets by having lower values compared with the K-means and DBSCAN algorithms. On the other, DB index for Ecoli dataset is slightly higher than K-means and DBSCAN. Even though, DBSCAN shows a lower values of DB index when applied on Diabet and Segment datasets compared with Improved GrDBAveD and K-means. But, the Improved GrDBAveD is still better than K-means since it has a lower value in comparison

to the K-means. The overall performance of Improved GrDBAveD is still better, since K-means and DBSCAN algorithms cluster the whole data at once (static clustering algorithm not dynamic).

Table 4-12: Experiment-2 the Average Davies–Bouldin Index of the Improved GrDBAveD and (K-means & DBSCAN Algorithms)

DB Index			
Algorithms / Dataset	**K-mean**	**DBSCAN**	**Improved GrDBAveD**
Iris	**0.486**	**0.486**	**0.465**
Wine	1.282	**1.177**	**1.081**
Ecoli	**0.848**	**0.832**	0.858
Diabet	1.351	**0.742**	**1.026**
Segment	0.806	**0.580**	**0.629**

- **Sum Square Error (SSQ):** According to Table 4-13**,** the Improved GrDBAveD has a lower value of SSQ compared with DBSCAN for most datasets. However, it does not have a higher SSQ for Ecoli in comparison to K-means and DBSCAN.

Table 4-13: Experiment-2 the Average Sum Square Error of the Improved GrDBAveD and (K-means & DBSCAN Algorithms)

Sum Square Error SSQ Index			
Algorithms / Dataset	**K-mean**	**DBSCAN**	**Improved GrDBAveD**
Iris	**18**	**18**	**16**
Wine	**2957**	4276	**3060**
Ecoli	**25**	**19**	38
Diabet	**6333**	14556	**13374**
Segment	262	**241**	**230**

- **SD Index:** According to Table 4-14, Improved GrDBAveD performs well in terms of SD index in most datasets. It has the lowest values (as shown in Iris and Ecoli) and the second lower values of SD index (for Diabet and Segment).

However, it does not have a lower value for the Wine dataset, but its overall performance is still good.

Table 4-14: Experiment-2 the Average SD Index of the Improved GrDBAveD and (K-means & DBSCAN Algorithms)

SD Index			
Algorithms / Dataset	**K-mean**	**DBSCAN**	**Improved GrDBAveD**
Iris	**1.203**	**1.203**	**1.185**
Wine	**1.562**	**1.567**	1.579
Ecoli	1.681	**1.637**	**1.406**
Diabet	2.034	**1.682**	**1.725**
Segment	**1.628**	1.940	**1.824**

The three algorithms produced the same number of clusters for each dataset. According to the results, the Improved GrDBAveD performs either better or almost the same as K-means and DBSCAN based on the performance measures. Overall, the Improved GrDBAveD performs well throughout the datasets, and we have to keep in mind that if DBSCAN and K-means show better performance at some points, because they both cluster the whole dataset at once. Improved GrDBAveD clusters one point at a time by adopting the gravity concept. Another advantage of adopting gravity is that we built a density-based clustering algorithm that can be part of an incremental solution for clustering compared with the DBSCAN and K-means, which only cluster static datasets.

The next experiment provides the results after integrating the Improved GrDBAveD with the aging approach to create the ***DBGrvAge Algorithm*** (that was discussed in details in 3.2.3, 3.3, 3.4, and 3.5).

4.3.3 Experiment -3

In experiment-3, a new method is introduced to add another feature for the proposed algorithm to satisfy one of the incremental learning tasks which is removing aged samples from the clusters. These samples are removed if they are not involved in clustering the new incoming samples frequently. This strategy called Aging, is integrated with the Improved GrDBAveD algorithm to create (***DBGrvAge algorithm*** that stands for Density-based clustering algorithm using the gravity and aging approaches). The Aging approach is introduced for three reasons: 1) to add a new incremental task which is

"deletions"; 2) To limit the memory usage; and 3) It may slightly reduce the running time. The Aging approach enables the proposed algorithm to delete points beside the insertion of the new points, which satisfies another task of the incremental learning process.

This experiment is divided into two sub-experiments which are: 1) Applying the **DBGrvAge algorithm** on a synthetic dataset in order to visualize the clusters' shapes. It is important to check if the proposed algorithm can produce clusters with arbitrary shapes since it is one of the primary requirements of the density-based clustering algorithms. Also, it shows if clusters are changeable over time as well; and 2) Applying the **DBGrvAge algorithm** on a real dataset to validate the performance in terms of time, memory and performance measures. The discussion of both sub-experiments is presented in the following section.

- **Synthetic Dataset**

  The synthetic data is created according to the following properties; attributes are numeric, data consists of 1500 observations with two features and a label attribute (consists of three classes that indicate the number of clusters). It is mainly used to visualize the changes happening to the clusters' shapes over time. The clusters' shapes were captured in six different slots during the iterations in order to demonstrate how they change over time.

  Before that, the dataset is divided into two parts: 30% for training as illustrated in Figure 4-3 that shows three different clusters. The other 70% of the dataset acts as the new incoming points that need to be clustered. Every 210 points are arrived and clustered; a screen shot of the clusters is captured. In additional, an age variable is declared to represent how long the point is valid for. The age for this experiment is 100 iterations for each point.

  Figure 4-4 shows a slight change in the clusters' shape after adding 210 new incoming samples to the clusters. It indicates that some points are inserted, and others are removed if they are not frequently used. After adding another 210 samples, Figure 4-5 shows new shapes for the clusters. It goes the same with Figure 4-6 and Figure 4-7, until the last 210 samples are inserted as shown in Figure 4-8**.** It shows that the proposed DBGrvAge algorithm that employs aging

can produce clusters with arbitrary shapes and changeable over time, including adding and deleting points.

1.

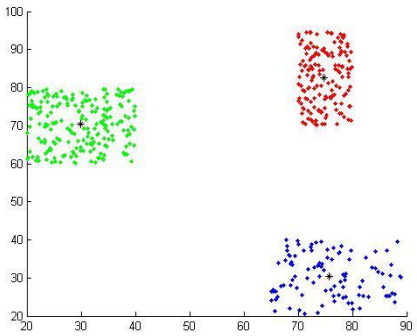

Figure 4-3: Training Samples for the Synthetic Data Before Clustering

2.

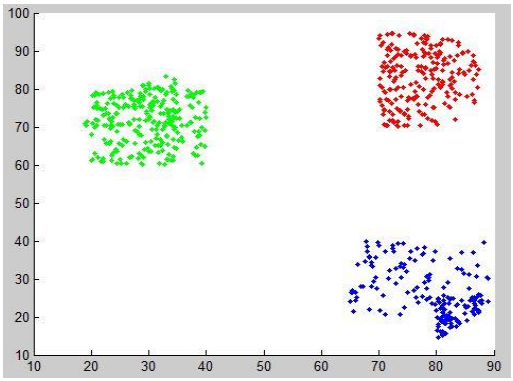

Figure 4-4: First Iteration- 210 New Incoming Samples are Clustered Based on the Gravity & Aging

3.

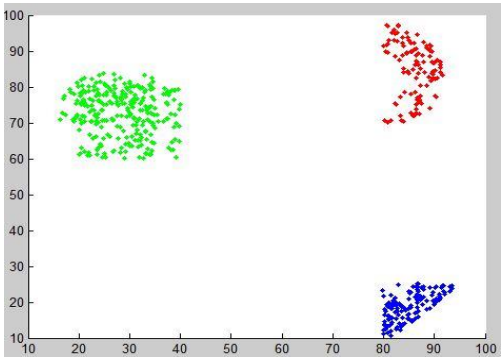

Figure 4-5: Second Iteration- 210 New Incoming Samples are Clustered Based on the Gravity & Aging

4.

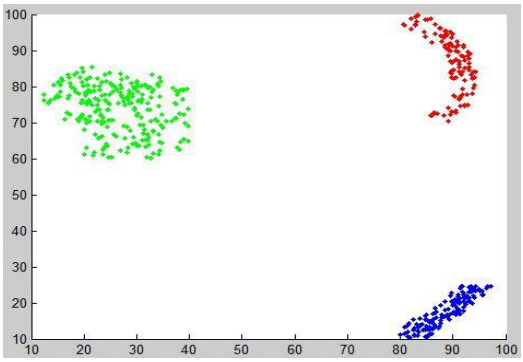

Figure 4-6: Third Iteration- 210 New Incoming Samples are Clustered Based on the Gravity & Aging

5.

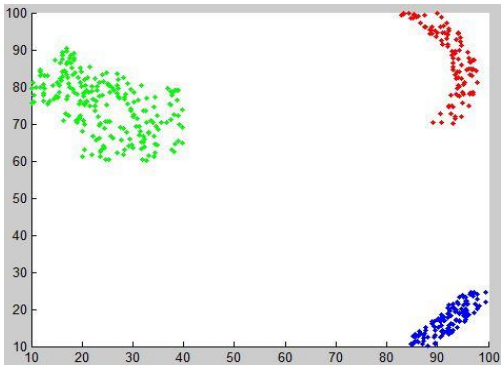

Figure 4-7: Fourth Iteration- 210 New Incoming Samples are Clustered Based on the Gravity & Aging

6.

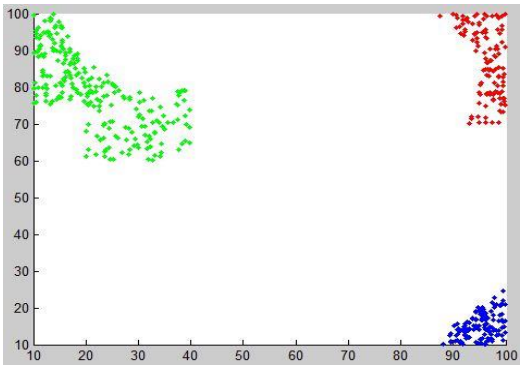

Figure 4-8: Fifth Iteration- 210 New Incoming Samples are Clustered Based on the Gravity & Aging

These Figures exhibit the learning process from the changes happening when inserting and deleting points over time according to the clusters shapes. However, we still need to show how the proposed algorithm can limit the memory usage, and whether it

reduces the running time while maintaining a high performance according to the performance measures. We will discuss this in the next section.

- **Real datasets**

In this experiment, the Iris and Segment datasets are used to validate the proposed algorithm after the aging approach is integrated and whether it limits the memory usage and reduces the time while maintaining the performance. The parameters of the proposed algorithm are specified as shown in Table 4-15**.** Besides**,** the aging parameter for each dataset is also specified, from the smallest age to the infinite values (i.e. the age value that does not cause deletion of any sample points, keeping the same memory size regardless). The age values are indicated in the first column of each table, see Table 4-17 and Table 4-19.

Table 4-15: Experiment-3 Parameters Setup of DBGrvAge Algorithm for Iris and Segment Datasets

DBGrvAge Algorithm		
Parameters / **Datasets**	**MinPts**	**Radius**
Iris	4	0.25,0.25&0.3
Segment	4	0.45 , 0.5 & 0.55

Table 4-17 illustrates the performance of the proposed DBGrvAge algorithm at different ages on Iris dataset. The age starts with the lowest value of 5 and the highest value of 70 that indicates the infinite value. The infinite value is the value of age that keeps the results of the performance measurements; memory size and the running time of the algorithm the same as if the aging strategy is not incorporated. According to the table, the DBGrvAge algorithm removes almost half of the points or less when the age is between 5 and 15, which is efficient because they limit the memory usage. Besides, the running time is between 0.215 and 0.294 (as presented in Table 4-17) which is lower than 0.354 (as shown in Table 4-16). Also, the DBGrvAge Algorithm shows a better performance in terms of the performance measurements, in which the Dunn index values after integrating aging are between 0.976 and 0.633 (as shown in Table 4-17), and they are higher than 0.426 (before aging as shown in Table 4-16). Also, the values of SSQ, DB Index and SD Index are lower than the values before integrating aging, which is better, see Table 4-16 and Table 4-17. On the other hand, when the age is between 55 and 70, the memory still contains most of the data. It means that these are the infinite values of age that keep the size of the memory the same as the original size of 141-142.

Table 4-16: Based on Experiment-1 the Results of the Improved GrDBAveD without Aging on Iris Dataset

Improved GrDBAveD Algorithm on Iris dataset before applying Aging						
Memory Size	**Running Time**		**Dunn Index**	**SSQ**	**DB Index**	**SD Index**
142	0.354		0.426	16	0.465	1.185

Table 4-17: Experiment-3 DBGrvAge Algorithm (after Integrating the Improved GrDBAveD Algorithm with Aging Strategy) on Iris Dataset

Iris Dataset Size: 150 (60 for Training & 90 for Testing)						
Age	**Memory Size**	**Running Time**	**Dunn Index**	**SSQ**	**DB Index**	**SD Index**
5	78	0.215	0.976	6	0.336	1.079
10	85	0.263	0.812	7	0.364	1.099
15	94	0.294	0.696	8	0.385	1.111
20	104	0.309	0.633	9	0.404	1.125
25	111	0.314	0.593	11	0.415	1.133
30	119	0.317	0.541	12	0.428	1.145
35	126	0.325	0.490	14	0.439	1.157
40	133	0.319	0.500	14	0.445	1.161
45	138	0.327	0.459	15	0.454	1.171
50	139	0.327	0.454	16	0.460	1.178
55	141	0.336	0.426	16	0.463	1.184
60	142	0.322	0.441	16	0.462	1.181
65	142	0.317	0.425	16	0.464	1.185
70	142	0.312	0.434	16	0.462	1.182

Table 4-19 illustrates the performance of the DBGrvAge algorithm at different ages on the Segment dataset. The age starts with the lowest value of 35 and the highest value of 1200. According to the table, the proposed algorithm removes a lot of points when the age is 35 and 150 which is not practical. On the other hand, when the age is between 900 and 1200, the memory still contains most of the data. It means that these are the infinite values of age that keep the size of the memory the same as the original size. However, when the age is between 200 and 270, then the results are reasonable because they limit the memory usage by keeping almost half of the dataset. Also, when the age is between 300 and 500, they also limit the memory usage by having more than half of the dataset. Moreover, the running time is between 30 and 42 (as shown in Table 4-19), which is slightly lower than 42 in some cases (as shown in Table 4-18). Furthermore, the DBGrvAge Algorithm shows lower values of SD Index between 1.626 and 1.714 when age is between 200 and 500 (as shown in Table 4-19) in comparison to 1.824 (before aging as shown in Table 4-18). The same with SSQ, it shows a better performance by having lower values (between 87 and 142) than 230, see Table 4-18 and Table 4-19. However, the values of the Dunn Index are between 0.226 and 0.257 (as shown in

Table 4-19) which are not higher than 0.266 (as shown in Table 4-18). Also, the DB index is higher when integrating aging with the values between 0.714 and 0.866. But, the overall performance of the proposed algorithm with aging is good, since it uses the memory efficiently.

Table 4-18: Based on Experiment-1 the Results of the Improved GrDBAveD without Aging on Segment Dataset

Improved GrDBAveD on Segment dataset before applying Aging					
Memory Size	**Running Time**	**Dunn Index**	**SSQ**	**DB Index**	**SD Index**
2217	42	0.266	230	0.629	1.824

Table 4-19: Experiment-3 DBGrvAge Algorithm (after Integrating the Improved GrDBAveD Algorithm with Aging Strategy) on Segment Dataset

Segment Dataset Size: 2310 (924 for Training & 1386 for Testing)						
Age	**Memory Size**	**Running Time**	**Dunn Index**	**SSQ**	**DB Index**	**SD Index**
35	558	23	0.292	38	0.761	1.297
40	570	22	0.343	38	0.677	1.684
45	598	23	0.193	42	0.845	1.500
50	625	22	0.297	44	0.671	1.735
55	642	24	0.195	37	0.802	1.835
60	653	23	0.290	46	0.711	1.697
65	681	23	0.330	49	0.699	1.724
70	695	23	0.309	51	0.726	1.991
80	750	24	0.238	57	0.826	1.848
85	763	24	0.217	58	0.769	1.700
90	770	25	0.280	60	0.755	1.871
95	786	25	0.308	61	0.665	1.967
100	817	23	0.322	84	1.033	1.346
110	838	25	0.279	65	0.720	1.558
115	821	24	0.311	81	0.698	1.665
120	861	26	0.203	69	0.921	1.699
150	931	28	0.240	84	0.888	1.544
200	1013	30	0.226	87	0.807	1.714
220	1047	30	0.245	90	0.823	1.726
250	1071	32	0.260	95	0.719	1.626
270	1104	33	0.251	107	0.720	1.646
300	1167	35	0.227	106	0.784	1.711
320	1183	35	0.230	108	0.773	1.673
350	1242	36	0.247	118	0.849	1.657
400	1316	38	0.255	128	0.729	1.656
450	1419	39	0.228	137	0.866	1.711
500	1527	42	0.234	142	0.714	1.713
600	1726	44	0.246	172	0.786	1.732
700	1990	43	0.241	206	0.799	1.828
800	2174	43	0.223	234	0.866	1.843
900	2242	43	0.224	245	0.861	1.802
1000	2253	42	0.233	227	0.786	1.831
1100	2253	43	0.244	235	0.840	1.893
1200	2257	44	0.226	236	0.837	1.884

Experiment-3 demonstrates how the DBGrvAge algorithm performed after integrating the aging strategy with the Improved GrDBAveD. It shows that the DBGrvAge algorithm can be applied on real datasets that need to be clustered in real-time. For that, another experiment is conducted to validate the DBGrvAge algorithm on data from a real application. More details are presented in the following section.

4.3.4 Experiment -4

Experiment-4 is performed in order to evaluate the performance of the DBGrvAge algorithm named as "*a density-based clustering algorithm based on Gravity and Aging*" on a real application. The selected application is Peer to Peer network traffic dataset. It is a quiet large dataset compared with what considered earlier in terms of the size. The result of the proposed algorithm is compared with K-means and DBSCAN algorithms based on the performance measures and memory usage. Details on the parameters of each algorithm are listed in Table 4-20, and the results of the performance measures are presented in Table 4-21.

Table 4-20: Experiment-4 Parameters Setup of the Datasets

	Algorithms					
	K- means	**DBSCAN**		**DBGrvAge**		
Parameters / Dataset	**K**	**MinPts**	**Radius**	**MinPts**	**Radius**	**Age**
P2P Traffic Network	4	3	19	4	21,10&12	15000

Table 4-21 shows the results of experiment-4 based on the performance measures that are employed to evaluate the algorithms to highlight which algorithm performs better according to the performance measures and memory usage. This table compares the performance of the DBGrvAge algorithm with DBSCAN and K-means when applied on the Peer to Peer network traffic application.

Each value in the tables represents the average of 20 iterations (or runs) for each algorithm (it is a column-based comparison for each row). The **bolded and underlined** values represent the best results achieved for each dataset according to the values of the performance measures and memory size of each algorithm while the **bolded** values are the second best results in a row.

Table 4-21: Experiment-4 a Comparison between DBGrvAge Algorithm and (K-means & DBSCAN) on P2P Network Traffic Application

P2P Traffic Network Size: 32767 (13107 for Training & 19660 for Testing)			
Algorithms / PM	**K-means**	**DBSCAN**	**DBGrvAge**
Dunn Index	0.0000014	**0.010**	**0.014**
SSQ	**264345**	521625	**269897**
DB Index	0.813	**0.212**	**0.508**
SD Index	**1.827**	3.153	**2.661**
Memory Size	32767	**32741**	**18627**
Cluster Number	4	4	4

As demonstrated in Table 4-21, the DBGrvAge algorithm performs well based on the Dunn Index and memory usage in comparison to K-means and DBSCAN algorithms. However, DBSCAN performs better than DBGrvAge and K-means according to the DB Index measure, but still DBGrvAge is lower than K-means, which specifies it as the second best value of DB. According to the value of SSQ measure, the K-means has the best result, whereas DBGrvAge and DBSCN have higher SSQ than K-means. But, DBGrvAge algorithm has much lower SSQ than DBSCAN, which make it the second best value for SSQ. On the other hand, DBGrvAge is the best in reducing the memory usage compared with K-means and DBSCAN algorithms while maintaining an excellent performance.

Overall, the DBGrvAge performs well throughout the datasets. We would like to highlight that DBSCAN and K-means cluster the entire dataset at once, while DBGrvAge clusters one point at a time according to the gravity concept. That is to say, the advantage of adopting gravity is that we built a density-based clustering algorithm that can be part of an incremental solution for clustering compared with the DBSCAN and K-means, which only cluster static datasets.

In chapter 3, we discussed the temporary bag used to move the incoming samples that do not satisfy the condition of the MinPts (minimum number of points within a radius). If the size of the temporary bag exceeds the threshold then a DBSCAN algorithm is applied offline to create new clusters out of the points in the temporary bag. However, the proposed algorithm is not able yet to merge those new clusters to the existing ones.

4.4 Discussion

We conducted several experiments to evaluate the proposed algorithm that incorporates the following: 1) the gravity approach (that allow the algorithm to insert samples incrementally at a time and studies the attribute similarities among neighbours while clustering new points); and 2) the aging approach (that removes aged samples and creates arbitrary shapes of clusters over time). We demonstrated that our proposed algorithm performs well in many cases.

The performance of the proposed algorithm was measured in terms of performance measures, running time, arbitrary shapes, and memory usage. Table 4-22 presents a summary of the experimental results.

Table 4-22: Summary of the Experimental Results

Experiment No.	Summary
Experiment -1	Compares the performance of the Improved GrDBAveD algorithm with the other versions of the proposed algorithm, and the outcomes are: • It is faster in terms of the running time. • It studies the attribute similarity among points while clustering new points. • It shows a better performance in terms of the performance measures in comparison with the other versions
Experiment -2	Compares the performance of the Improved GrDBAveD algorithm with well-known algorithms like K-means and DBSCAN, and the outcomes are: • It shows a better performance in most of the performance measures. • It inserts points incrementally one at a time in comparison with the static K-means and DBSCAN algorithms which cluster points all at once.

Experiment No.	Summary
Experiment -3	Demonstrates the aging strategy after integrating it with Improved GrDBAveD to create the DBGrvAge algorithm, and the outcomes are: • It can create clusters with arbitrary shapes and changeable over time. • It limits the memory usage. • It removes aged samples over times that are not frequently involved in clustering new incoming points.
Experiment -4	Compares the performance of the DBGrvAge algorithm with the well-known algorithms K-means and DBSCAN when applied on a real application, and the outcomes are: • It shows a better performance in most of the performance measures in comparison to K-means and DBSCAN algorithms. • It inserts points incrementally one at a time in comparison to the static K-means and DBSCAN which clusters points all at once. • The DBGrvAge algorithm uses the memory efficiently in comparison to K-means and DBSCAN algorithms by removing old samples.

Another important observation is that through the experiments, it was noticeable that the algorithm is **sensitive** to the input parameters. It shows that the selection of the parameter values has an effect on the clustering performance, and one needs to go through several experiments with different parameters. Table 4-23 highlights the fulfilled requirements by the proposed algorithm to identify what it could be done to improve it later.

Table 4-23: Fulfilled Requirement by the Proposed Algorithm

Evolution Matrices		***Density-based Clustering Algorithm based on Gravity and Aging* (DBGrvAge)**
Different Arbitrary Shapes		√
Neighbourhood with Different Densities		√ (Gravity Concept)
Attribute Similarity		√
Handling High Dimensionality		(Needs to be validated with high-dimensional dataset)
Automatically Defined Parameters		X
Incremental Tasks	**Insert Points**	√
	Creating Cluster	√
	Merging Clusters	X
	Splitting Clusters	X
	Removing Points	√

4.5 Chapter Summary

In this chapter, the proposed algorithm, DBGrvAge Algorithm (stands for *A Density-Based Clustering Algorithm Based on Gravity and* Aging) was evaluated by conducting four different experiments. These experiments were evaluated by validating the algorithm on several datasets types (Synthetic dataset, real datasets and dataset of real-time application). Then their results were assessed by the performance measures, memory usage, and running time as discussed earlier in this chapter. We demonstrated that, according to these measures, the proposed algorithm performed well throughout the experiments.

CHAPTER 5. CONCLUSIONS

5.1 A Summary of the Research

This study proposes a density-based clustering algorithm that can be part of an incremental solution in clustering, inspired by the Newton's Law of Gravity and Aging Strategy. Each of these approaches focuses on providing a solution to resolve the stated problems. The ***Gravity approach*** is adopted to allow the proposed algorithm to be part of an incremental solution for clustering by inserting the new incoming samples to their clusters. Furthermore, the ***Aging Approach*** is introduced to utilize the memory efficiently by removing less frequently used samples.

In this study, the gravity is employed to computes the force of gravity for each neighbouring sample of the new incoming samples, and it is done by measuring the density of the area surrounding each neighbouring sample and dividing it by the square distance (distance from the new incoming sample to its neighbour). There are two primary requirements need to be considered during the process of insertion, which are: 1) the ability to detect neighbourhoods with uneven densities around the new incoming points; and at the same time; and 2) the ability to study the attribute similarities among points within a single neighbourhood. These requirements are measured by adopting the gravity approach that accommodates new measurements to compute the distance and density. For that, the Euclidean distance is used to study the similarities among points, and to identify the densities of their neighbourhoods.

Several measurements to compute the density (Mass) of the gravity equation were introduced in Chapter 3 in order to insert the new incoming samples while considering the variant densities and attribute similarities among points (using Number of points within a radius as discussed in Sections 0 and 3.2.1, Silhouette measure as presented in Section 3.2.2, Average distance discussed in Sections 3.2.3 and 3.3). Each of those measurements computes the density of the neighbouring samples. Once the density is identified, the distance from the new incoming sample to its neighbour is computed. Then, the neighbourhoods' force of gravity for the neighbours is computed by dividing the density over the square distance. If the distance increases between the new incoming samples to its neighbour, the force of the neighbourhood is decreased. Thus, the neighbour with the higher force among other neighbours is the one that pulls in the new incoming sample to its cluster, indicating the ***insertion task***.

The other proposed concept, the ***Aging* approach**, is introduced to enable the proposed algorithm to accomplish another incremental task which is deletion. It assigns an age for each sample in the clusters to indicate the lifetime of the sample according to how long does the sample participates in clustering new points. It performs the followings: 1) when clustering new incoming samples, the aging approach keeps track of the samples that participate in the clustering; 2) it deletes samples that are less frequently used over time; and 3) it enables the proposed algorithm to form clusters of arbitrary shapes and changeable over time. Aging approach removes the points that have completed their life cycle according to the assigned age. The removal of points from the clusters causes some changes in the clusters' shapes, and they keep changing over time. Having an algorithm able to form clusters of arbitrary shapes and changeable over times is important, because it is one of the main requirements that distinguish the density-based clustering methods from other methods.

The experimental results evaluated the performance of the improved GrDBAveD algorithm by comparing it with the proposed versions discussed in Chapter 3 and with the well-known clustering algorithms such as DBSCAN and K-means algorithms. These algorithms along with the proposed one were validated on several datasets; including synthetic and real datasets. The experimental evaluation was focused on comparing the algorithms according to the performance of clustering by employing several performance measures including Dunn Index, SD Index, SSQ, and DB Index clusters. These measures evaluate the clusters in terms of how compact the clusters are and how well-separated they are. Also, the DBGrvAge algorithm (is the improved GrDBAveD integrated with aging approach) was evaluated in terms of clusters' shapes, memory usage, running time, and parameter sensitivity.

As demonstrated in the four conducted experiments in Chapter 4, the proposed algorithm performed well. However, we also noticed that the algorithm is sensitive to the input parameters.

In summary, the proposed DBGrvAge algorithm clusters new samples based on the similarities among points by employing a new definition of density and distance measurements. Also, it deletes less frequently used points (aged ones) leading to efficient use of the memory, and creates clusters of arbitrary shapes changeable over time. Accordingly, we conclude that our proposed density-based clustering algorithm using

Gravity and Aging approaches (DBGrvAge algorithm) can be part of an incremental solution to cluster data dynamically.

5.2 Contributions of the Thesis

This study contributes to the field of clustering in data mining by proposing a density-based clustering algorithm that introduces two new approaches as follows:

- **Gravity** is adopted to enable the density-based clustering algorithm to perform the following:
  - It studies the attribute similarities among points when clustering new incoming points.
  - It inserts new points into clusters while considering the variant densities (i.e. Number of points within a radius, Silhouette measure, and Average distance) and how the distance between the new incoming point and its neighbours affects the force of gravity of those variant densities of the neighbours' neighbourhood.
  - Gravity concept makes the proposed algorithm incremental by fulfilling the ***insertion task*.**
- Introducing the **Aging strategy** that performs the following:
  - It enables the proposed algorithm to learn from the changes happening to the clusters over time by keeping track of the points that participate in clustering new incoming points.
  - It removes points that are less frequently involved in clustering new incoming points. Thus, it works efficiently in minimizing the memory usage.
  - It creates clusters with arbitrary shapes and changeable over time.
  - Aging strategy makes the proposed algorithm incremental by fulfilling the ***deletion task***.

5.3 Limitations of the Thesis

We can identify the limitations of the proposed algorithm by highlighting the fulfilled requirements. These requirements represent the criteria that need to be characterized by the optimal algorithm. They are identified through the literature review as presented in Table 2-2 and Table 2-3. According to the experimental results, the proposed algorithm was evaluated based on the requirements discussed in Table 4-23

in Chapter 4. Table 5-1 is reproduced here from Table 4-23 indicating the criteria that were satisfied with symbol (√) and the unsatisfied one with the symbol (X).

As shown in Table 5-1, there are four features that were not satisfied by the proposed algorithm. For instance, the proposed algorithm is unable to automatically define its input parameters; it cannot merge clusters if they are close, and when new clusters are created and need to be merged with existing ones. Also, it cannot split a cluster if the points within the cluster are sparse. It is not capable of fading clusters, if no insertion and deletion happened over time.

Table 5-1: Fulfilled Requirement by the Proposed Algorithm

Evolution Matrices		***A Density-based Clustering Algorithm Based on Gravity and Aging***
Different Arbitrary Shapes		√
Neighbourhood with Different Densities		√ (Gravity Concept)
Attribute Similarity		√
Handling High Dimensionality		(Needs to be validated with high-dimensional dataset)
Automatically Defined Parameters		X
Incremental Tasks	**Insert Points**	√
	Creating Cluster	√
	Merging Clusters	X
	Splitting Clusters	X
	Removing Points	√

5.4 Future work

According to the limitations discussed in Section 5.3, there are some works for future research which we highlight as follows:

- Validating the proposed algorithm in high-dimensional data. This is challenging especially if the data are very sparse.
- Configuring the parameters of the proposed algorithm automatically by finding the optimal values. Although challenging, however, this can offer a better and faster performance.
- Finding new measurements to enable the algorithm to split and merge clusters; maybe by using the same concept of gravity and make it fit into those cases.

REFERENCES

Amini, A., Wah, T. Y., & Saboohi, H. (2014). On density-based data streams clustering algorithms: A survey. *Journal of Computer Science and Technology, 29*(1), 116-141.

Angel, L. M., & Shankar, K. R. (2012). A density based dynamic data clustering algorithm based on incremental dataset. *Journal of Computer Science, 8*(5), 656-664. doi:10.3844/jcssp.2012.656.664

Ankerst, M., Breunig, M. M., Kriegel, H., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. *SIGMOD '99 Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data,* 49-60.

Berkhin, P. (2006). A survey of clustering data mining techniques. *In Grouping multidimensional data.* (pp. 25-71) . Springer Berlin Heidelberg. doi:10.1007/3-540-28349-8_2

Cassisi, C., Ferro, A., Giugno, R., Pigola, G., & Pulvirenti, A. (2013). Enhancing density-based clustering: Parameter reduction and outlier detection. *Journal of Information Systems, 38*(3), 317-330. doi:http://dx.doi.org/10.1016/j.is.2012.09.001

Chaoji, V., & Zaki, M. (2009). *Efficient algorithms for mining arbitrary shaped clusters.* (Doctoral Dissertation, Rensselaer Polytechnic Institute Troy).

Chen, C., Hwang, S., & Oyang, Y. (2002). An incremental hierarchical data clustering algorithm based on gravity theory. *Advances in knowledge discovery and data mining.* (pp. 237-250) Springer.

Chen, Z. (2010). Graph-based clustering and its application in coreference resolution. *In Proceedings of the 2010 Workshop on Graph-Based Methods for Natural Language Processing, ACL 2010,* 1-9.

Ester, M., Kriegel, H., Sander, J., Wimmer, M., & Xu, X. (1998). Incremental clustering for mining in a data warehousing environment. *VLDB '98 Proceedings of the 24rd International Conference on very Large Data Bases,* 323-333.

Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *In Kdd*, 226-231.

Goyal, N., Goyal, P., Venkatramaiah, K., Deepak, P., & Sannop, P. (2011). An efficient density based incremental clustering algorithm in data warehousing environment. *2009 International Conference on Computer Engineering and Applications, IPCSIT, (2)*

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2002). Clustering validity checking methods: Part II. *ACM Sigmod Record, 31*(3), 19-27.

Han, J., Kamber, M., & Pei, J. (2011). In Kamber M., Pei J. (Eds.), *Data mining: Concepts and techniques* (3rd ed.). Burlington: Morgan Kaufmann.

Hinneburg, A., & Gabriel, H. (2007). DENCLUE 2.0: Fast clustering based on kernel density estimation. *In Proceedings of the 7th International Symposium on Intelligent Data Analysis*, 70-80. doi:10.1007/978-3-540-74825-0_7

Ilango, M., & Mohan, V. (2010). A survey of grid based clustering algorithms. *International Journal of Engineering Science and Technology, 2*(8), 3441-3446.

Kovács, F., Legány, C., & Babos, A. (2005). Cluster validity measurement techniques. *Symposium of Hungarian Researchers on Computational Intelligence,* Hungary. 18-19.

Kulkarni, P. (2012). Incremental learning and knowledge representation. *Reinforcement and systemic machine learning for decision making* (pp. 177-208) Wiley Online Library.

Legány, C., Juhász, S., & Babos, A. (2006). Cluster validity measurement techniques. *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, 388-393.

Liu, Q., Deng, M., Shi, Y., & Wang, J. (2012). A density-based spatial clustering algorithm considering both spatial proximity and attribute similarity. *Computers & Geosciences, 46*, 296-309.

Liu, Y., Li, Z., Xiong, H., Gao, X., & Wu, J. (2010). Understanding of internal clustering validation measures. *Data Mining (ICDM), 2010 IEEE 10th International Conference On*, 911-916.

Lopez-Molina, C., Bustince, H., Fernandez, J., Couto, P., & De Baets, B. (2010). A gravitational approach to edge detection based on triangular norms. *Pattern Recognition, 43*(11), 3730-3741. doi:http://dx.doi.org/10.1016/j.patcog.2010.05.035

Lühr, S., & Lazarescu, M. (2009). Incremental clustering of dynamic data streams using connectivity based representative points. *Journal of Data & Knowledge Engineering, 68*(1), 1-27.

Madhulatha, T. S. (2012). An overview on clustering methods. *IOSR Journal of Engineering, 2*(4), 719-725.

Mann, A. K., & Kaur, N. (2013). Survey paper on clustering techniques. *International Journal of Science, Engineering nd Technology Research, 2*(4), 0803-0806.

Meng, J., & Cheng, W. (2008). Research of gravity-based outliers detection. *IIHMSP '08 International Conference on Intelligent Information Hiding and Multimedia Signal Processing.* 590-593. doi:10.1109/IIH-MSP.2008.48

Mohammadi, M., Raahemi, B., Akbari, A., Nassersharif, B., & Moeinzadeh, H. (2012). Improving linear discriminant analysis with artificial immune system-based evolutionary algorithms. *Journal of Information Sciences, 189*(0), 219-232. doi:http://dx.doi.org/10.1016/j.ins.2011.11.032

Oyang, Y., Chen, C., & Yang, T. (2001). A study on the hierarchical data clustering algorithm based on gravity theory. *Principles of data mining and knowledge discovery* (pp. 350-361) Springer.

Parimala, M., Lopez, D., & Senthilkumar, N. (2011). A survey on density based clustering algorithms for mining large spatial databases. *International Journal of Advanced Science and Technology, 31*(1)

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45-77.

Quan, Q., Xiao, C., & Zhang, R. (2013). Grid-based data stream clustering for intrusion detection. *International Journal of Network Security, 15*(1), 1-8.

Rashedi, E., & Nezamabadi-pour, H. (2012). A stochastic gravitational approach to feature based color image segmentation. *Engineering Applications of Artificial Intelligence,* doi:10.1016/j.engappai.2012.10.002

Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences, 179*(13), 2232-2248. doi:http://dx.doi.org/10.1016/j.ins.2009.03.004

Ronkainen, P. (1998). *Attribute similarity and event sequence similarity in data mining.* (PhD Thesis, University of Helsinki, Department of Computer Science).

Rui, X., & Wunsch, D. (2005). Survey of clustering algorithms. *Neural Networks, IEEE Transactions On, 16*(3), 645-678. doi:10.1109/TNN.2005.845141

Sander, J., Ester, M., Kriegel, H., & Xu, X. (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery, 2*(2), 169-194.

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review, 1*(1), 27-64. doi:http://dx.doi.org/10.1016/j.cosrev.2007.05.001

Tryon, R., & Bailey, D. (1970). Cluster analysis. McGraw-Hill.

Vaishnavi, V. (2008). *Design science research methods and patterns: Innovating information and communication technology*. MA, USA: Auerbach Publications Boston.

Wang, B., Jin, R., Zhang, S., & Zhao, X. (2009). Research on gravity-based anomaly intrusion detection algorithm. *IAS '09. Fifth International Conference on Information Assurance and Security. , 2* 350-353. doi:10.1109/IAS.2009.283

Wright, W. E. (1977). Gravitational clustering. *Pattern Recognition, 9*(3), 151-166. doi:http://dx.doi.org/10.1016/0031-3203(77)90013-9

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. *ACM SIGMOD Record, 2(25)* 103-114.

Zhong, J., Liu, L., & Li, Z. (2010). A novel clustering algorithm based on gravity and cluster merging. *Advanced data mining and applications* (pp. 302-309) Springer.