NOVEL CLUSTERING ALGORITHMS: ENTROPY BASED NEIGHBORHOOD MERGING (ENM) AND SIMULTANEOUS FEATURE SELECTIVE CLUSTERING (SFSC)

Mustafa ÜNVER

Dissertation

Department of Industrial Engineering

Supervisor: Prof. Dr. Nihal ERGİNEL

Eskişehir

Eskişehir Technical University

Institute of Graduate Programs

March 2023

This thesis has been supported by the project 20DPR014, which is approved by the SRP Committee.

FINAL APPROVAL FOR THESIS

This thesis titled NOVEL CLUSTERING ALGORITHMS: ENTROPY BASED NEIGHBORHOOD MERGING (ENM) AND SIMULTANEOUS FEATURE SELECTIVE CLUSTERING (SFSC) has been prepared and submitted by Mustafa ÜNVER in partial fulfillment of the requirements in "Eskişehir Technical University Directive on Graduate Education and Examination" for the Degree of PhD in Industrial Engineering Department has been examined and approved on 24/03/2023.

Committee Members		**Title, Name and Surname**	**Signature**
Member	:	Prof. Dr. Kemal ÖZKAN	
Member	:	Prof. Dr. Tülin İNKAYA	
Member	:	Assoc. Prof. Dr. Özer ÖZDEMİR	
Member	:	Assoc. Prof. Dr. Duygu YILMAZ EROĞLU	
Member	:	Assoc. Prof. Dr. Ahmet ARSLAN	

Prof. Dr. Murat TANIŞLI

Director of the Institute of Graduate Programs

24/03/2023

SUPERVISOR APPROVAL

PhD student Mustafa ÜNVER, whom I supervise, has completed this thesis titled NOVEL CLUSTERING ALGORITHMS: ENTROPY BASED NEIGHBORHOOD MERGING (ENM) AND SIMULTANEOUS FEATURE SELECTIVE CLUSTERING (SFSC) . According to my inspections, the work is scientifically and ethically appropriate for the student to the thesis defense exam.

Supervisor

Prof. Dr. Nihal ERGİNEL

ABSTRACT

NOVEL CLUSTERING ALGORITHMS: ENTROPY BASED NEIGHBORHOOD MERGING (ENM) AND SIMULTANEOUS FEATURE SELECTIVE CLUSTERING (SFSC)

Mustafa ÜNVER

Department of Industrial Engineering

Eskişehir Technical University, Institute of Graduate Programs, March 2023

Supervisor: Prof. Dr. Nihal ERGİNEL

Clustering is an unsupervised machine learning approach that is frequently used by researchers in data science. It can be defined as the process of dividing a data set into subsets such that similar elements are in the same subset and/or dissimilar elements are in different subsets. In the state of art for clustering literature, clustering methodologies are challenged by some issues such as arbitrary geometric shaped clusters, density variations, multidimensional feature space and overcoming these issues within reasonable computational complexity and within a user friendly environment.

In this thesis, novel clustering methodologies (Entropy Based Neighborhood Merging – ENM and Simultaneous Feature Selective Clustering - SFSC), are proposed for arbitrary geometric shaped clusters along with the density variations and multidimensional feature space respectively. These algorithms are based on some statistical concepts such as Shannon's Entropy, Symmetric Relative Entropy and Isotropic Position. Experimental analysis of both ENM and SFSC are carried out on benchmark datasets to show their applicability and efficiency on clustering. Furthermore, two case studies are given (seismic zone detection and churn analysis) for the real-time applications of both algorithms. The experimental analyzes and the case studies showed that, ENM and SFSC are efficient methodologies for the relevant challenging issues and they have high levels of applicability, interpretability and user friendly character.

Keywords: Clustering, Arbitrary Geometric Shaped Clusters, Intra-Cluster and Inter-Cluster Density Variations, Multi-Dimensional Feature Space, Feature Selection.

ÖZET

ÖZGÜN KÜMELEME ALGORİTMALARI: ENTROPİ TABANLI KOMŞULUK BİRLEŞTİRME VE EŞ ZAMANLI ÖZNİTELİK SEÇİCİ KÜMELEME

Mustafa ÜNVER

Endüstri Mühendisliği Anabilim Dalı

Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Mart 2023

Danışman: Prof. Dr. Nihal ERGİNEL

Kümeleme, veri biliminde araştırmacılar tarafından sıklıkla kullanılan gözetimsiz bir makine öğrenmesi yaklaşımıdır. Bir veri kümesini, benzer öğeler aynı alt kümede ve/veya benzer olmayan öğeler farklı alt kümelerde olacak şekilde alt kümelere ayırma işlemi olarak tanımlanabilir. Kümeleme literatürünün en son durumunda, kümeleme metodolojileri keyfi geometrik şekilli kümeler, yoğunluk değişimleri, çok boyutlu öznitelik uzayı gibi bazı sorunlarla karşı karşıyadır ve bunları daha düşük hesaplama karmaşıklığı içinde ve kullanıcı dostu bir ortamda aşmak popüler araştırma alanlarıdır.

Bu tezde, yoğunluk değişimleri ile birlikte keyfi geometrik şekilli kümeler ve çok boyutlu öznitelik uzayı için özgün kümeleme algoritmaları (Entropi Tabanlı Komşuluk Birleştirme (ENM) ve Eşzamanlı Öznitelik Seçici Kümeleme (SFSC) geliştirilmiştir. Bu algoritmalar, Shannon Entropisi, Simetrik Bağıl Entropi ve İzotropik Konum gibi bazı istatistiksel kavramlara dayanmaktadır. Hem ENM hem de SFSC'nin, kümeleme problemlerinde uygulanabilirliği ve etkinliği göstermek için kıyaslayıcı veri setleri üzerinde deneysel analizi gerçekleştirilmiştir. Ayrıca, her iki algoritmanın da gerçek zamanlı uygulaması olarak iki vaka çalışması (sismik bölge tespiti ve müşteri kaybı analizi) verilmiştir. Deneysel analizler ve vaka çalışmaları, ENM ve SFSC'nin ilgili zorlayıcı konular için etkili metodolojiler olduğunu ve yüksek düzeyde uygulanabilirliğe, yorumlanabilirliğe ve kullanıcı dostu karaktere sahip olduğunu göstermiştir.

Anahtar Sözcükler: Kümeleme, Keyfi Geometrik Şekilli Küme Tespiti, Küme İçi ve Kümeler Arası Yoğunluk Değişimi Problemi, Çok Boyutlu Öznitelik Uzayı, Öznitelik Seçimi.

ACKNOWLEDGEMENTS

This thesis has been financially supported by Eskişehir Technical University Scientific Research Projects on Project: 20DPR014.

This thesis is also financially supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK), Science Fellowships and Grant Programme Directorate, 2211 - National PhD Scholarship Programme.

I would like to thank to my supervisor, Prof. Dr. Nihal Erginel, for her great support and encouragement during my PhD process. I also acknowledge to my thesis steering committee members; Assoc. Prof. Dr. Tülin İnkaya & Assoc. Prof. Dr. Özer Özdemir, for their valuable feedbacks and constructive critiques.

I also owe a lot; not only to; the sun, the moon, the stars, the rain, the snow, the wind, the sky, the clouds, the forests, the seas, the trees, the birds, the cats, the dogs, the cool summers, the warm winters…

But also; to bitter coffee, to sweet tea, to the songs in my playlist, to Covid19, to the trains, to my mugs, to my all knapsacks & outdoor outfits, to my veteran laptop and veteran smartphone…

To the gray of Ankara, to the brown of Eskişehir and to the blue of İstanbul.

For great motivation, inspiration and facilitation.

Mustafa ÜNVER

24/03/2023

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Eskişehir Technical University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Mustafa ÜNVER

CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

GLOSSARY OF SYMBOLS AND ABBREVIATIONS

2d	:	Two Dimensional (or Two Dimensions)
3d	:	Three Dimensional (or Three Dimensions)
β	:	Beta
γ	:	Gamma (as Discount Factor)
Γ	:	Gamma
#	:	Number
ABC	:	Artificial Bee Colony
ACO	:	Ant Colony Optimization
AGFC	:	Adaptive Grid Based Forest-Like Clustering
ANN	:	Artificial Neural Network
App.	:	Application
ARI	:	Adjusted Rand Index
CLIQUE	:	Clustering in QUEst
CH	:	Calinski-Harabasz
CURE	:	Clustering Using Representatives
DAN	:	Density Adaptive Neighborhood
DB	:	Davies-Bouldin
DBCLASD	:	Distribution-Based Clustering of Large Spatial Databases
DBSCAN	:	Density-Based Spatial Clustering Application with Noise.
Det.	:	Determinant
Dim.	:	Dimensions
Dim. Red.	:	Dimension Reduction
Dit	:	Decimal Digit
DL	:	Deep Learning
DMEL	:	Data Mining by Evolutionary Learning
ENM	:	Entropy based Neighborhood Merging
FC	:	Fractal Clustering
FMS	:	Fowlkes and Mallows Score
FRMVK	:	Feature-Reduction Multi-View k-Means
FS	:	Feature Selection

GA	:	Genetic Algorithms
GG	:	Gabriel Graphs
Heu.	:	Heuristics
JI	:	Jaccard Index
JS Dist.	:	Jensen-Shannon Distance
JS Div.	:	Jensen-Shannon Divergence
Kernel PCA	:	Kernel Principal Component Analysis
KD-Tree	:	K- Dimensional Tree
KL Div.	:	Kullback-Leibler Divergence
KNN	:	K- Nearest Neighbor
LC	:	Linear Classification
LDA	:	Linear Discriminant Analysis
LDP-MST	:	Local Density Peaks – Minimum Spanning Tree
LR	:	Logistics Regression
LSTM	:	Long-Short Term Memory
MDP	:	Markov Decision Process
MILBC	:	Mean Information Loss between Clusters
MILWC	:	Mean Information Loss within Clusters
MKNN	:	Mutual K-Nearest Neighbor
MinPts	:	Minimum number of points
MP	:	Multilayer Perceptron
MST	:	Minimum Spanning Tree
N\A	:	Not Applicable
Nat	:	Natural Unit of Information
NB	:	Naïve Bayes
NC	:	Neighborhood Construction
NCAR	:	Neighborhood Construction based on Apollonius Regions
Neigh	:	Neighborhood
NJW	:	Ng–Jordan–Weiss
NMI	:	Normalized Mutual Information
OD-ACO	:	Origin Destination Ant Colony Optimization
OPTICS	:	Ordering points to identify the clustering structure.
PCA	:	Principal Component Analysis

PDPC	:	Particle Swarm Intelligence based Density Peak Clustering
PSO	:	Particle-Swarm Optimization
Reg.	:	Regression
RF	:	Random Forest
RI	:	Rand Index
ROCK	:	Robust Clustering Algorithm For Categorical Attributes
SDSCM	:	Semantic-Driven Subtractive Clustering Method
SFSC	:	Simultaneous Feature Selective Clustering
SM	:	Simultaneous K-Way Cut with Multiple Eigenvectors
SOM	:	Self-Organizing Map
STING	:	Statistical Information Grid
SUBCLUE	:	Subspace Clustering Ensembles
SVC	:	Support Vector Clustering
SVM	:	Support Vector Machine
t-SNE	:	t-Stochastic Neighborhood Embedding
UCI	:	University of California Irvine
UMAP	:	Uniform Manifold Approximation and Projection
VarClus	:	Variable Clustering
VNS	:	Variable Neighborhood Search
w.r.t.	:	With Respect To
XGBoost	:	Extreme Gradient Boosting

1. INTRODUCTION

By the globalization, increasing competition and growing technical & economic opportunities; importance of the data science increased and it makes data science applications indispensable in making strategic, tactical and operational decisions in almost every sector and in almost all branches of science. On the other hand, the application areas of data science and its sub-fields have expanded immensely with the growing data production rates and data sizes, the spread and the development of data collection, data storage and data processing systems.

Data mining can be defined as the process of extracting useful and interesting information from huge amounts of raw data, in other words, it is an interdisciplinary field of extracting useful information from raw data. Since data mining is an interdisciplinary field, it uses methods from many fields such as operations research, computer science, machine learning and statistics. As another definition, data mining is a set of effective and popular methods that are used in solving problems in many branches of engineering, economics, health sciences, biology and so on. (Alakuş, 2018)

As another concept in data science, machine learning can be defined as a set of algorithms that are used to extract useful information from various databases. (Mitchell, 1997)

Machine learning algorithms can be generally classified in three groups: supervised learning algorithms, unsupervised learning algorithms and reinforcement learning algorithms. Supervised learning algorithms produce a function that maps inputs to target outputs. Classification and regression algorithms are included in the supervised learning category. Contrarily to supervised ones, unsupervised learning algorithms try to automatically reveal hidden data structures in the unlabeled data sets. Clustering and dimension reduction are the most common unsupervised learning techniques. As for the reinforcement learning algorithms, they are based on actions of agents in any environment and every action of the agent creates an effect and these effects are considered as rewards that guide the algorithm. In addition to these three types of machine learning approaches, there are semi-supervised learning approaches and “learning to learn” approaches. Some machine learning algorithms can be classified as semi-supervised learning because they are focused on producing results on both labeled and unlabeled data. As for the “learning to learn” approaches, they are the algorithms that are classified as learning to learn because they work by utilizing the experiences. (http-1)

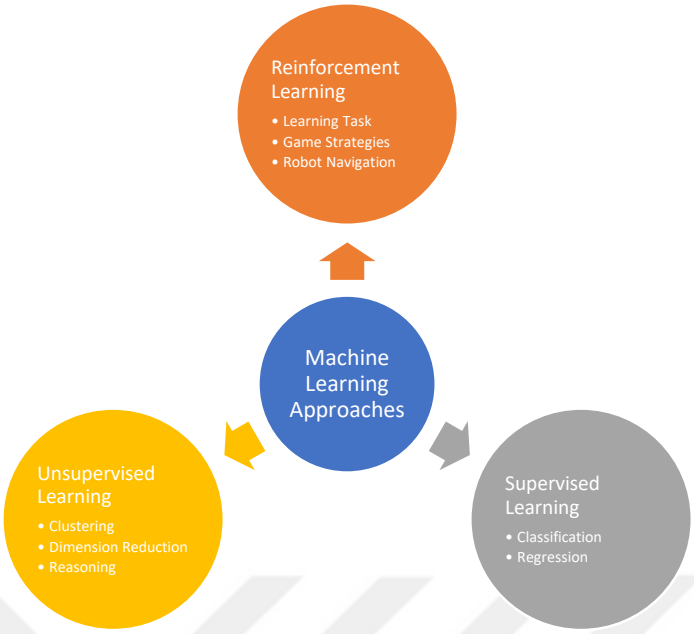

Figure 1.1. *Categorization of machine learning approaches*

Clustering is an unsupervised learning method that is frequently used by researchers in data science. Clustering is most commonly defined as the process of dividing a data set into subsets such that similar elements are in the same subset and/or dissimilar elements are in different subsets. According to another definition, it is the process of grouping each data in a data set according to its own characteristics such as spatial distance. (İnkaya, 2011)

There are many types of clustering algorithms in the literature. Traditional clustering algorithms can be classified according to their algorithmic structure as follows: partition-based approaches (k-means, PAM, etc.), hierarchical approaches (BIRCH, CURE etc.), density-based approaches (DBSCAN, Mean Shift, etc.), graph theory based approaches (CLICK, MST), grid-based approaches (STING, CLIQUE), probability distributions-based approaches (expectation maximization (EM), etc.) and fractal geometry-based (FC) approaches, and model-based approaches. (SOM, COBWEB, etc.) Furthermore, there are also modern clustering algorithms such as kernel based approaches, swarm intelligence based approaches, quantum theory based approaches and some other modern approaches. (Xu & Tian, 2015)

While some clustering approaches, like partition-based and probability distribution-based approaches, require the desired number of clusters in the output, which is defined a priori by the user; density approaches and grid-based approaches do not require such

prior knowledge and can perform clustering application through input parameters and unearth hidden structures in the dataset.

Apart from their algorithmic structures and the original number of cluster requirement in the dataset, another factor that can be considered in the classification of clustering approaches is whether the approach is fuzzy. Fuzziness in these approaches has arisen in two ways: The first way is the fuzziness of the clustering result. While each data can belong to only one cluster in crisp clustering approaches, each data may belong to more than one cluster with a certain membership degree in fuzzy clustering. In the second way, fuzzy logic principles are used to increase the strength of the algorithmic structure and make the algorithm more effective, rather than to obtain a fuzzy clustering result. (Ünver & Erginel, 2020)

The first and most popular fuzzy clustering approach in the literature is the fuzzy c-means method, which was developed by Bezdek et al. in 1984. This method is the fuzzy version of the k-means method, and unlike the k-means method, it aims to assign each data to all clusters with a certain membership degree which is in [0,1] interval. (Bezdek , Ehrlich , & Full , 1984) In addition to the fuzzy c-means method, fuzzy versions of some traditional and modern clustering methods can also be found in the literature. (Ünver & Erginel, 2020) (Nasibov & Ulutagay, 2009) (Bordogna & Dino, 2014)

1.1. Definition of Clustering Problem

Clustering can generally be defined as the division of a data set into subgroups according to a certain type of similarity measure such as spatial distance. More technically; it is the process of separating a data set that consists of M-dimensional N vectors $X_N = [X_{N1}, X_{N2}, X_{N3}, \dots X_{NM}]$ into c subsets like $C_i = \{X_a, X_b, X_c, \dots\}$, such that similarity between the elements in the same subset is the highest, and the similarity between the elements in different subsets is the lowest. In this process, the situation where $c=N$, that is, every element is a distinct cluster in the dataset, or the situation where $c=1$, that is, dataset itself is a subset, are called trivial solution. In most of the clustering methodologies c is assumed to be known and clusters are spherical.

Clustering problems are NP-hard combinatorial problems and require a huge computational burden since the solution space increased exponentially based on size of the dataset. Even for very small-scale datasets, as the solution space is also large in the process of assigning n data to c clusters. (İnkaya, 2011) (Xu & Tian, 2015)

In clustering algorithms, the similarities between the data are usually handled in a way that is inversely proportional to the spatial distances, when the data is quantitative. Manhattan, Minkowski, Euclidean, Canberra, Mahalanobis, Chebyshev and many other types of distance measures are generally used in clustering algorithms, but Euclidean distances are generally preferred. Manhattan, Euclidean and Chebyshev distances are the special type of Minkowski distance when n is 1, n is 2 and n goes to infinity, respectively. Where the data is qualitative, similarity measures such as simple matching coefficient, Jaccard coefficient or Hamming distance are used. In (Kazaz, 2019), some types of distance and similarity used in clustering algorithms and their formulas are summarized in details.

In the literature, there are many methodologies that are developed for clustering problems with varying computational complexity. The complexity of the k-means method, which is one of the most popular clustering approaches, is O(k*t*d*n), the complexity of the fuzzy c-means method is O(n), and the complexity of the DBSCAN algorithm, which is one of the density-based approaches is O(n*log(n)) and computational complexity of modern clustering approaches is higher than the traditional ones. Algorithms with the lowest computational complexities are; WaveCluster, which is a grid-based approach with O(n) complexity, BIRCH from hierarchical approaches, and STING from graph theory-based approaches. Most approaches based on neighborhood construction or spectral graph theory based ones have at least $O(n^2)$ complexity. (k: number of clusters, t: number of iterations, n: number of data, d: number of dimensions) (İnkaya, 2011) (Xu & Tian, 2015) (Berkhin, 2002)

Clustering problems are frequently encountered in many fields such as bioinformatics, image processing, supply chain, marketing, biomedical, geographic information systems, social media analysis, digital image processing, community/fraud detection, and it is a necessary tool in order to obtain meaningful and useful information from data stacks.

1.2. A General Clustering Process

For clustering problems, solution process consists of several steps. These steps are data preparation, definition of data matrices and parameters /variables, selection of a convenient clustering algorithm, implementation, validation analysis of the clustering output and interpretation of the result. This process is shown in Figure 1.2. (Kaya, 2018)

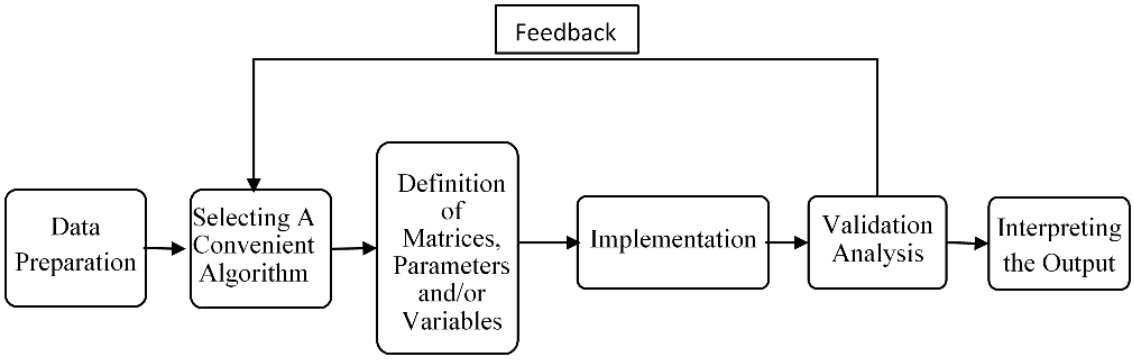

Figure 1.2. *A general clustering process*

1.3. Data Preparation (Preprocessing)

Pre-clustering data preparation processes may include, size reduction for attributes in the dataset, transformation and standardization of data, missing value imputations and etc., depending on the situation of the dataset. If the dataset is multidimensional, a dimension reduction process should be applied on the dataset, so redundant, less meaningful, or some of the correlated attributes can be eliminated. Moreover, clustering output may be visualized on 2d or 3d map by means of a certain dimension reduction technique. A more suitable dataset for clustering can be obtained by applying conversion and standardization processes such as conversion to Z scores, conversion to [0,1] range, conversion to [-1,1] range, if the data has different attribute standards. And if the dataset has some missing values, these values can be imputed by using some methods like interpolation. (Kazaz, 2019) (Halkidi, Batistakis, & Vazirgiannis, 2001)

1.4. Appropriate Clustering Algorithm Selection

The choice of the clustering algorithm is critical for the clustering outputs to be meaningful and useful in terms of quality and validity. Since the definition of a cluster is a relative, there are many clustering methods in the literature. However, not every method is capable of producing a meaningful result for every data set. The most important criterion in this selection is the data structures and possible cluster structures in the data set. In addition, the computational complexity of the algorithm, whether it can be applied on large-scale datasets and/or multidimensional datasets, the requirement of input parameter settings and the ability to produce fuzzy outputs etc. are also other important criteria in the selection of algorithm.

1.5. Clustering Validation

There are many validation indices that are used to evaluate the clustering outputs. In (Halkidi, Batistakis, & Vazirgiannis, 2001), validation indices are discussed in detail and they are classified in three main categories. If each data in the dataset has a label (a value that indicates original data class), external criteria are used, and if labels are not known a priori, internal validity indices are preferred. Relative criteria are used when testing the validity of multiple clustering outputs. However, since external indices are based on statistical true-false values, and internal indices are based on cluster centers (centroids) and accept clusters as circular, they are far from useful on data sets with non-convex - arbitrary shaped cluster structures. In addition to these criteria, there are also some fuzzy validity indices for fuzzy clustering techniques. However, since these criteria adopt the approach of the internal criteria, they accept the clusters as circular shapes and are meaningful only on data sets with non-convex arbitrary shaped cluster structures. (Alakuş, 2018)

- External Validation Indices:

Most of the popular external metrics are based on knowledge of whether or not each data is correctly classified, as the data labels are known. Related to this test result, true-positive (TP), true-negative (TN), false-positive (FP) and false-negative (FN) values are used as input for these validity indices. Higher values are desirable for most of the external validation indices and most of them are in [0,1] range. Some preliminary descriptions and related formulas for external validation indices are given below: (İnkaya, 2011), (Halkidi, Batistakis, & Vazirgiannis, 2001)

- True-Positive (TP): The number of data pairs belonging to the same cluster in the clustering output which are also originally in the same class.
- True-Negative (TN): The number of data pairs belonging to the different clusters in the clustering output which are also originally in the different class.
- False-Positive (FP): The number of data pairs belonging to the same class originally which are also belonging to different clusters in the clustering output.
- False-Negative (FN): The number of data pairs belonging to the same cluster in the clustering output, which are also belonging to different classes originally.

  - **Jaccard Index (JI):**

$$JI = TP/(TP + FP + FN) \tag{1.1}$$

- **Rand Index (RI):**

$$RI = (TP + TN)/(TP + TN + FP + FN) \tag{1.2}$$

- **Fowlkes ve Mallows Score (FMS):**

$$FMI = \sqrt{TP^2/((TP + FP) * (TP + FN))} \tag{1.3}$$

Another example of external validation indices is v-measure which is based on two statistical concepts: Homogeneity and Completeness. Full homogeneity basically means all the points within clusters has the same class label and completeness depicts all the points which has the same label fell into the same cluster in the clustering schema. Homogeneity and completeness indicators both take values from 0 and 1 and larger values are desirable. V- measure appears as harmonic mean of completeness and homogeneity measures in order to evaluate clustering result in a single indicator. V-measure naturally falls within 0-1 range and larger values are desirable as well. (Rosenberg & Hirschberg, 2007)

Related equations are given for the v-measure calculation as in (Rosenberg & Hirschberg, 2007): Equation 1.4. depicts calculation of class entropy, Equation 1.5. provide conditional entropy for a clustering result, Equation 1.6. & Equation 1.7. measures homogeneity & completeness respectively and finally Equation 1.8. shows the harmonic mean of homogeneity and completeness as v-measure. In these equations n refers to total number of points in the dataset, $|C|$ is the number of classes in the dataset, n_c refers to number of element in the relevant class, k refers to number of clusters in the cluster schema, $n_{c,k}$ refers to number of items which are labelled as class c and n_k refers to total number of items in cluster k.

$$H(C) = -\sum_{c=1}^{|C|} \frac{n_c}{n} * log\left(\frac{n_c}{n}\right) \tag{1.4}$$

$$H(C|K) = -\sum_{C=1}^{|C|}\sum_{K=1}^{|K|} \frac{n_{c,k}}{n} * log\left(\frac{n_{c,k}}{n_k}\right) \tag{1.5}$$

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{1.6}$$

$$c = 1 - \frac{H(K|C)}{H(K)} \tag{1.7}$$

$$v = 2 * \frac{h * c}{h + c} \tag{1.8}$$

Another external validation index is Normalized Mutual Information (NMI). (Pluim, Maintz, & Viergever, 2003) NMI is based on class entropy and predicted cluster entropy. Related equations are given in Equation 1.9. H(Class) and H(Cluster) are entropy of the original classes and predicted clusters, while H(Class | Cluster) is conditional entropy for original classes given by predicted clusters. On both formulas are based on classical entropy formula in information theory and 2 based logarithmic function is used. Entropy formula can be found in the next sections.

$$NMI(Class, Cluster) = \frac{H(Class) + H(Cluster)}{H(Class|Cluster)} \tag{1.9}$$

- Internal Validation Indices:

These criteria are based on attributes between data in the dataset without considering the original class labels of the data in the dataset. Some of the most commonly used internal metrics are provided below with their formulas. A higher Hubert index indicates compact clusters, a higher Dunn Index (DI) indicates compact and well-separated cluster structures, while a lower Davies-Bouldin (DB) index indicates a small mean similarity of each subset to the others, and a low root mean square standard deviation index indicates intra-cluster variability. (Halkidi, Batistakis, & Vazirgiannis, 2001)

From Equation 1.10 up to Equation 1.15 indicate the formula of each internal validation indices. (N: Total number of data in the dataset, P(i,j): Similarity matrix between element i and j, Q(i,j): Distance between the cluster centroids which element I and j belong to. nc: number of clusters, d(x,y): distance between x and y, $S_{i,j}$: mean within-cluster distance in the cluster i and j, SS_w: intra-cluster variability, SS_b: inter-cluster variability, SS_t: total variability. X_i refers to point i and $\bar{X}$ refers to mean value.)

- **Hubert (Γ) Statistics:**

$$\Gamma = \frac{2}{N * (N-1)} * \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} P(i,j) * Q(i,j) \tag{1.10}$$

- **Dunn Index (DI):**

$$DI = \min_{i=1,\dots,nc} \left\{ \min_{j=i+1,\dots nc} \left(\frac{\min\limits_{x \in Ci, y \in Cj} d(x,y)}{\max\limits_{k=1,\dots,nc} \left(\max\limits_{x,y \in Ck} d(x,y) \right)} \right) \right\} \tag{1.11}$$

- **Davies-Bouldin Index (DBI):**

$$DB = \frac{1}{nc} * \sum_{1}^{nc} \left(\max_{i=1,\ldots,nc\ (i \neq j)} \left(\frac{s_i + s_j}{d_{ij}} \right) \right) \quad (1.12)$$

- **Root Mean Square Standard Deviation (RMSSD) Index:**

$$SS_w \ or \ SS_b = \sum_{i=1}^{n} (X_i - \bar{X})^2 \quad (1.13)$$

$$SS_t = SS_w + SS_b \quad (1.14)$$

$$RMSSD = \sqrt{\frac{SS_w + SS_b}{SS_t}} \quad (1.15)$$

- Relative Validation Indices:

Relative criteria are used to find the best clustering scheme when evaluating the outputs of clustering processes with different parameters. They are used to determine the good clustering output of different parameter selections, especially in approaches such as the k-means method, which needs a priori defined cluster number by the users, or the input parameter dependent DBSCAN method.

They are divided into two groups: Where the number of clusters to be determined or input parameters for the algorithm. The selection is made according to any internal or external validity criterion in a knee method or knee graph. The value for the number of clusters or the value of input parameters that start to keep the validation indices (criteria) constant or starts to provide negligible and little variability for a long time is selected as the best clustering structure. “Knee” depicts a breakdown of the line in the graph constructed by different parameter or number of cluster settings. (Halkidi, Batistakis, & Vazirgiannis, 2001)

1.6. Challenging Issues in Clustering

In this section, recent challenging issues are summarized in the clustering literature like arbitrary shaped cluster detection, noise problems and inter-cluster - intra-cluster density variations. They frequently attract most of the researchers’ interest who study clustering across the globe.

1.6.1. Input Parameter Estimation

Most of the clustering algorithms require input parameters, preliminary knowledge for number of original classes in the dataset or both. That creates a non-user friendly environment and users of the algorithm may have to involve in a trial and error process. In partition based algorithm, k-means, k represents the number of partition on the dataset.

Likewise, epsilon (radius of spherical window) and MinPts (minimum number of points constraint within a spherical window to assign core points) are two input parameters in DBSCAN which is a popular and well-known density based algorithm. In both example, true parameter settings play key role in producing valid, meaningful and high quality clustering schema. But finding true parameter setting is not quite straightforward and time consuming.

1.6.2. Noise / Outlier Problem

In some datasets, there may be some data points which belongs to no cluster and represent itself as a distinct cluster due to infinitesimal similarity to all clusters. Except the density based approaches, most of the clustering approaches cannot handle noisy datasets. Alternatively, noise points can be ignored and can be treated as a single point cluster and by that way other methodologies can be convenient. There is no straightforwardly well definition of noise (or outliers) in general, since there is no threshold value for the similarity measure in order to declare a data point as a noise. In density based approaches, noise points are defined as the points for which have less points in its epsilon neighborhood than a MinPts threshold. In that concept, threshold is static and may cause low quality clustering schema, and noise points either may belong to a cluster or may cause declaration of too many noises that might be a member of a cluster.

1.6.3. Arbitrary Geometric Shaped Clusters

Most of the clustering algorithms such as partition based ones, hierarchical ones, probabilistic ones, are based on minimizing sum of squared errors assumes data clusters are spherical shaped around a centroid. In case, dataset has nonconvex cluster shapes or clusters which have overlapping convex hulls, a low quality clustering schema is obtained. A cluster may have been partitioned into sub clusters, some clusters may have been overlapped or partially or entirely integrated.

Figure 1.3, illustrates the k-means schema for a dataset the importance of the arbitrary shaped cluster existence. Spherical shaped cluster assumption reduces the efficiency of the any clustering algorithm. Some density based approaches, some hierarchical approaches or graph theoretical based ones can overcome this problem along with a parameter estimation problem or less convenience on high scale datasets.

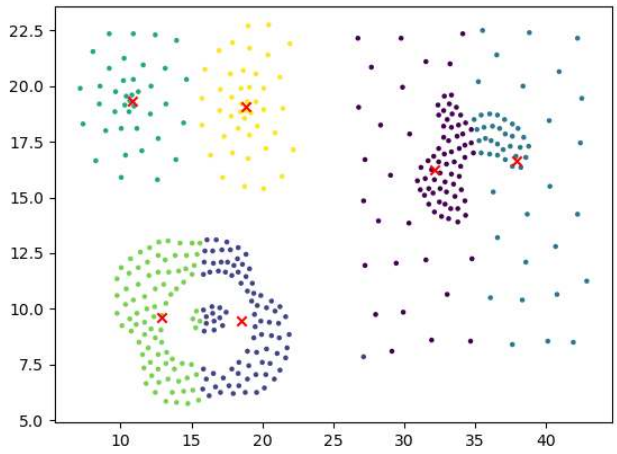

Figure 1.3. *Centroid based clustering output for arbitrary shaped clusters*

Density based ones are based on divide and conquer style, epsilon neighborhood merging. But a high quality clustering schema needs a long series of trial and error process with different input parameters. Some graph theoretical approaches can produce higher clustering schema quality, but most of them are not convenient for large scale datasets because of their complexities. Furthermore, these methodologies either uses epsilon neighborhood, KNN neighborhood or some special type of neighborhood.

These arbitrary shaped clusters may be come across in spatial datasets in the fields of pattern recognition, geographical information systems or computer graphics and so on.

1.6.4. Inter-Cluster and Intra-Cluster Density Variations

As introduced in (Xu & Wunsch, 2005), particularly in 2d or 3d datasets, one more challenging issue which may reduce robustness of clustering methodologies is density variations problem. Two types of data density variations may exist in the dataset which decreases clustering schema quality and may bears a long series of trial and error process: Inter-cluster density variations and intra-cluster density variations.

- Inter-Cluster Density Variations:

If there are both dense and sparse clusters in a dataset, most of the clustering algorithm fail on detecting these kind of clusters in high level too. It also causes lower validation index results and lower clustering quality since most of the original clusters may be defined as noises w.r.t. static density definition or there may be too many number of clusters detected. In Figure 1.4, an intra-cluster density variations example is given.

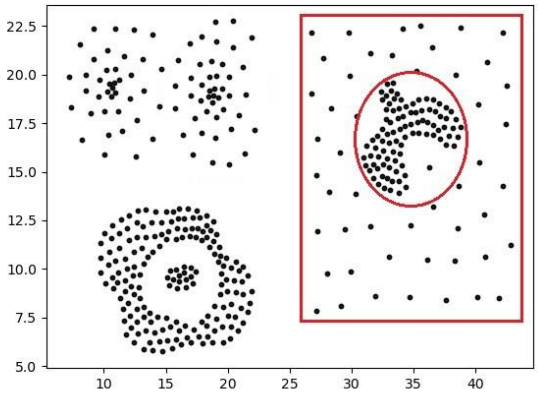

Figure 1.4. *Inter-Cluster density variation examples*

- Intra-Cluster Density Variations:

In case, data is scattered in different density levels within a cluster, most of the clustering algorithm fails detecting these kind of clusters in high level. It causes lower validation index results and lower clustering quality since density varies in the cluster and some points in the cluster either defined as noise or original cluster may be divided into separate clusters. In Figure 1.5, an inter-cluster density variations example is given.

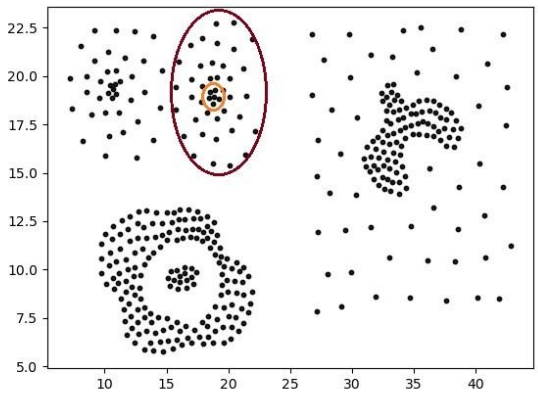

Figure 1.5. *Intra-Cluster density variation examples*

1.6.5. High Dimensional Feature Space

The concept “curse of dimensionality” is first introduced by R. E. Bellman in 1957 for dynamic programing problems and attract the interest of the researcher in the fields of databases, data mining, machine learning, combinatorics, sampling and so on. (Bellman & Rand Corporation, 1957) (http-2)

As we deal with high dimensional dataset in a clustering applications, distance functions fail in measuring the similarity/dissimilarity between data, since the datasets

fall too sparse and distance to nearest and farthest point in the dataset converges. Let P is a reference point and its maximum and minimum distance to the points among the set of points Q_1, Q_2, Q_3, …, Q_n fall indistinguishable as in Equation 1.16. as the dimension of the data grows.

$$\lim_{d\to\infty} E\left(\frac{dist_{max}(d)-dist_{min}(d)}{dist_{min}(d)}\right) \to 0 \tag{1.16}$$

Consequently, if a dataset contains n data with m attributes, with respect to larger values both for m and n; similarity/dissimilarity measures fail and nxn similarity matrix structure is required along with the huge computational burden for clustering applications. As the number of dimension grows, visualization dataset and clustering output becomes impossible as well.

As an example, a marketing dataset is visualized, that consists of customers and their attributes (income, spending score and age). Figure 1.6, depicts original dataset with clusters in distinct colors in 3d. At first look age attribute is redundant for that clustering, since age is a function of other two attribute. As a result, eliminating age from the dataset will not affect clustering output. In higher dimensional datasets, it is not easy to distinguish which attributes are redundant or not, since the visualization is impossible and for clustering application quality and validity, a dimensional reduction preprocessing step must be carried out.

In Figure 1.7, dataset is mapped in 2d by using only income and spending score attributes and proving redundancy of the age attribute for clustering.

Moreover, as a proof of similarity measure fails, geodesic distance of customer A, B and C can be given as an example. Distance from A to B and C are the same in Euclidean space. However, while A and B are in the same cluster and, C belongs to another cluster. But in 2d projected dataset, dissimilarity (distance) of customer A and B is less and dissimilarity of A and B to customer C is higher as desired, naturally A and B are grouped in the same cluster but C will fall to another cluster.

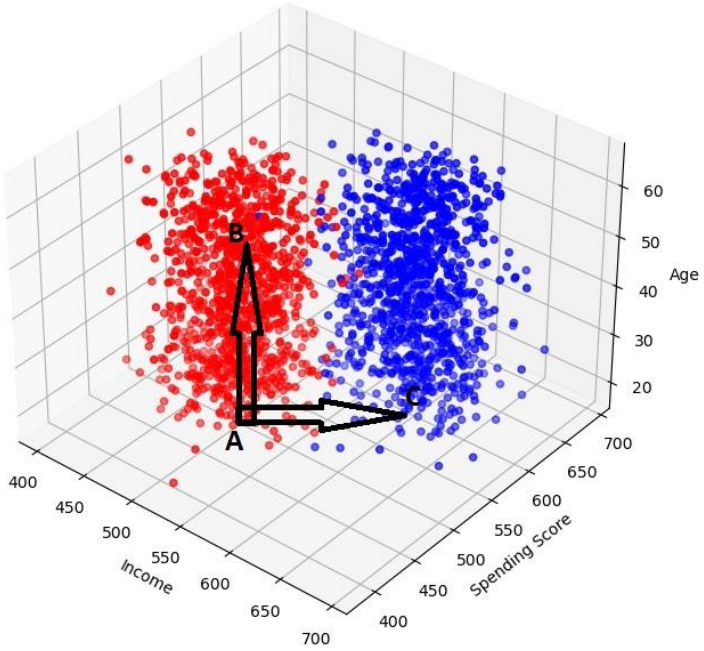

Figure 1.6. *Customer dataset in 3d (age, income, spending score)*

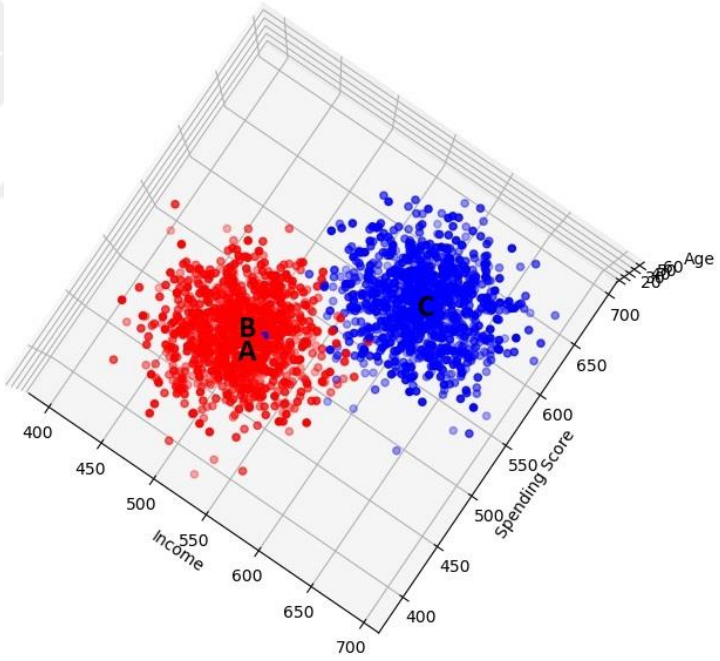

Figure 1.7. *Customer dataset is projected in 2d (income and spending score)*

As a quantitative example, consider A, B and C be arbitrary vectors as in Equation 1.17. The Euclidean distance between A & B and A and C is 5 in 3d. But in Equation 1.18, by dimensionality reduction of third redundant feature; the Euclidean distance between A & B is 0, while the Euclidean distance between A & C is 5. And that means; similarity between these three point must be described by first 2 features rather than all features in the dataset. Apparently, the number of redundant attributes cause more

convergence in the similarity measures as the number of attributes grows and they cannot be noticed straightforwardly in the datasets.

$$A = [3,4,0]\ , B = [3,4,5], C = [7,7,0];\ dist(A,B), dist(A,C) = 5 \qquad (1.17)$$

$$A = [3,4]\ , B = [3,4], C = [7,7]; dist(A,B) = 0, dist(A,C) = 5 \qquad (1.18)$$

Redundancy of attributes decreases the clustering quality much more as the dimension in the dataset increases since the similarity measure will be more pointless. In order to overcome this problem, a preprocessing application which does feature elimination/selection/extraction is proposed. But, in dimensional reduction loss of information are inevitable and it will apparently affect quality and validity of the clustering process. Moreover, if dataset have still more than 3 attributes after the dimensionality reduction, visualization of the clustering output is impossible and unless a feature extraction process is done, obtaining a sensible clustering visualization becomes impossible without decreasing clustering schema quality.

1.7. Research Questions, Motivation of the Dissertation and the Outline

In general, clustering problems that is considered in this dissertation are assumed to have the following features which are given in items:

1. The original number of cluster knowledge is assumed not to be known by the user in the problems and the purpose is to reveal hidden data structures in the dataset.
2. Clustering problems mainly on large scale and/or high dimensional datasets.
3. Clustering problems which have non-convex and/or overlapping convex hulls for multiple clusters, in other words, arbitrary geometric shaped clusters.
4. Clustering problems with inter-cluster and/or intra-cluster density variations.
5. Noise or outlier values may be included in the clustering problems.
6. Randomly ordered datasets may be used in the problem.

The clustering methodologies to be developed within this dissertation are planned to have the following features in terms of effectiveness and usability:

1. Based on an approach that builds neighborhoods and obtains cluster outputs by expanding neighborhood, as in density based algorithm like DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) or graph theoretical algorithm like NC (İnkaya, 2011).

2. No input parameter is required for the algorithm or it can work with default input parameters to fit every case, so as to create a user-friendly environment.
3. Able to handle variations within and/or between clusters.
4. Able to detect noise / outlier data type in the dataset, at least as a separate cluster.
5. Since they must also be convenient for large-scale data clustering problems, it must be an approach that keeps the computational complexity lower as much as possible.
6. Since they must also be convenient for high dimensional feature space, it must be an approach that can also make dimension reduction not to stuck in curse of dimensionality.

As for the research questions, following items are listed:

1. Is it possible to use some statistical concepts like Shannon's Entropy and its extensions or isotropic position in a clustering algorithm so as to come up with a statistical approach in clustering?
2. Is it possible to adapt Reinforcement Learning (RL) for clustering applications, so that awarded agents will automatically produce clusters in the datasets and intrinsically determine the number of clusters in the dataset itself?
3. Shannon's entropy and RL strategy may provide a convenient capability in arbitrary geometric shaped clusters?
4. Both statistical concepts (Shannon's entropy and isotropic position) and/ or RL strategy may provide more user friendly environment in terms of parameter-free way?
5. Shannon's entropy and its extensions and/or RL strategy may provide a noise detection capability and a sensitive density variation manner?
6. By means of Shannon's entropy and its extensions and/or RL strategy, is it possible to generate some new internal validation indices in order to evaluate non-convex, arbitrary geometric shaped clusters?
7. Can clustering and dimension reduction be applied simultaneously on datasets, so as to refrain a cascaded approach which include "first dimension reduction than clustering" and within lower computational burden?
8. Can simultaneous clustering and dimension reduction be effectively applied on real-life cases in that high dimensional data is naturally a subject?

9. Is it possible to generate an algorithm which has a reasonable computational burden, by means of statistical concepts and recursive structure of the RL strategy, in order to make the algorithms convenient for large-scale and/or high dimensional dataset?

The next parts of this PhD dissertation are composed of the following topics: Section 2 contains comprehensive literature survey for relevant subjects. Section 3 introduces a novel methodology (ENM) for arbitrary shaped cluster detection & density variations problems along with the applications of ENM on 2d & 3d challenging benchmark datasets. In Section 4, seismic zones map of Turkey is generated as a real-life application of ENM. Section 5 introduces another novel methodology for multi-dimensional feature space clustering along with the applications of SFSC on challenging UCI datasets. In Section 6, churn analysis is given as a real-life application of SFSC. And finally, Section 7 points out conclusion and future research issues.

2. LITERATURE SURVEY

In the section, state of art for clustering methodologies is examined and clustering algorithms are discussed along with some properties and their categorizations. First categorization is made according to algorithmic structure and the approach algorithms follow. The second categorization is done based on holistic or non-holistic manner of the algorithm. Moreover, arbitrary shaped cluster detection algorithms and high dimensional feature space algorithms are elaborated in the latter subsections.

In the literature, it is seen that many clustering approaches and various methods related to these approaches and their various versions are frequently used on both real-time datasets and synthetic datasets. Since each clustering approach is developed according to the structure of the dataset to be used in clustering and the definition of the cluster, these approaches can also be perceived as types of clustering problems. A broad literature review of these approaches and algorithms can be found in (Xu & Wunsch, 2005) and (Berkhin, 2002). In addition to these reviews, neighborhood-based approaches are examined in detail in (Pourbahrami & Khanli, 2018).

2.1. Clustering Algorithms According to Their Algorithmic Structure and Approach

As a state of art categorization of clustering algorithms, researchers generally make this classification: Traditional and modern clustering algorithms. Since traditional algorithms are mainly based on simple - classical computational way, they are easy to use and they have lower complexity, on the other hand their clustering schema quality is lower as well. As for the modern algorithms, contrarily to traditional ones, clustering schema quality is higher, but they have higher complexity and they are not easy to use and comprehensible by the users.

2.1.1. Traditional Clustering Algorithms

Traditional algorithms can be generally subcategorized as partitioning based, hierarchical, probability distributions-based, density-based, graph theory-based, grid-based, fractal geometry-based and model-based approaches according to their approaches which they use to produce clustering output. Some of the well-known methodologies are discussed below as an example to each approach category.

The most well-known and oldest of the partitioning-based approaches are the k-means (MacQueen, 1967) and k-medoid (Park & Jun, 2009) algorithms. Assuming that each data cluster in the data set is circular around a centroid (for k-means) or data group

(for k-medoid approaches), the centers of the data clusters are determined in an iterative way, and each data is assigned to the closest data centroid. These relatively low complexity approaches are not useful for datasets containing arbitrarily shaped cluster structures because they are centroid-based and assume clusters as convex spherical shapes. Fuzzy C-Means, is also non-crisp version of k-means algorithm. (Bezdek , Ehrlich , & Full , 1984).

Hierarchical approaches, on the other hand, are based on initially assuming each data as a cluster and creating new clusters by combining these initial clusters according to some certain similarity measures. BIRCH (Zhang, Ramakrishnan, & Livny, 1996) and CURE (Guha, Rastogi, & Shim, 1998) are examples of hierarchical approaches. While clustering in BIRCH is according to a feature tree, CURE uses random sampling for clustering. ROCK (Robust clustering using links) (Guha, Rastogi, & Shim, 1999) follow the same manner with CURE. A random data sample is drawn then clustering process is carried out on this sample with agglomerative way and then sub-clusters are merged based on their similarities. These approaches, need to specify the number of clusters as antecedent, just as in partitioning-based approaches. Most of the hierarchical algorithms suffer from the same problem, input parameter estimation and determination of right size of random samples. Their complexity is also higher than $O(n^2)$ in general.

In probability distributions based approaches, the main purpose is to distinguish these distributions from each other, since it is considered that there are data from different distributions in the data set. The most popular probability distribution based algorithm is expectation maximization (EM) (Rasmussen, 1999) algorithm is based on Gaussian distributions. These approaches are also fuzzy clustering approach in a way. Because each data point can belong to more than one cluster with a certain probability value. When these probabilities are considered as membership degrees, a fuzzy clustering output is obtained. These approaches need input parameters and have high computational complexity, do not have the ability to recognize arbitrarily shaped data sets, as in partitioning and hierarchical approaches.

Density-based approaches create data clusters by separating regions with higher data density in the data set from regions with lower density, and define data that remains alone in low-density regions as noise/outlier data. DBSCAN (Ester, Kriegel, Sander, & Xu, 1996), OPTICS (Ankerst, Breunig, Kriegel, & Sander, 1999) and Mean Shift (Fukunaga & Hostetler, 1975) are the most well-known density-based methods. In

contrast to partitioning, hierarchical and probability distribution-based approaches, they have the ability to reveal arbitrarily shaped cluster structures. In these approaches, which are usually lower in complexity, the quality of the clustering output is highly dependent on the correct selection of the input parameters. OPTICS (Ankerst, Breunig, Kriegel, & Sander, 1999) is the derivative of the DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) algorithm, in which the input parameters can be updated automatically. In the Mean Shift (Fukunaga & Hostetler, 1975) method, it is aimed to find the most dense centers by shifting the neighborhood (a circular window) center to the region along the way in which the density increases and to assign the data to these centers.

In graph theory-based approaches, data points are treated as nodes and relations between data as edges. CLICK (Sharan & Shamir, 2000) and minimum spanning tree-based clustering (MST) (Jain, Murty, & Flaynn, 1999) are two classical examples of graph theory-based approaches. CLICK (Sharan & Shamir, 2000) creates clusters by finding the graph segments with the least weight with an iterative structure, and MST-based clustering (Jain, Murty, & Flaynn, 1999) creates clusters by detecting the least spanning tree structures in the data set. Generally, the efficiency of these approaches drops dramatically as the complexity of the graph increases. The complexity of the graph is obviously increased as the number of data in the dataset increases.

Just as CLICK and MST, some other techniques which run on graph theoretical perspective are developed using pairwise point neighborhood as well. In Neighborhood Construction (NC) algorithm (İnkaya, Kayalıgil , & Özdemirel, 2015), based on Gabriel Graphs, density is measured within a sphere which has a diameter as distance of two points. As a result, to find the densest region, an n-square density matrix is needed, such that n is size of the dataset. Then direct and indirect neighborhood relations are determined to build clusters as an expansive way. Biggest advantage of NC is being input parameter free but having higher complexity relatively to epsilon neighborhood approaches.

Neighborhood Construction based on Apollonius Regions (NCAR) (Pourbahrami, Khanli, & Azimpour, 2019) is an extension methodology of NC algorithm. In NCAR, core points are similarly found as in NC and assigned as target points at first, then for each given target points, farthest points are located and used as counterpart to make pairwise point neighborhood and start forming primary Apollonius regions as clusters.

Likewise, NCAR also brings $O(n^2)$ complexity because of pairwise neighborhood density check at start.

One more algorithm is proposed as hybrid K-NN (CHKNN) based on graph theory. K-NN neighborhood strategy is followed in CHKNN and firstly sub-clusters are built up then they are merged by graph connectivity. (Qin, Yu, Wang, Gu, & Li, 2018) Similarly, Normalized Cuts (Shi & Malik, 2000) is another graph theoretical based clustering methodology which can be used for arbitrary shaped cluster detection. It deals with clustering as a graph partition problem and consider both total dissimilarity between clusters and similarity within clusters. Both CHKNN and Normalized Cuts suffer from larger scale datasets since the both need similarity matrix which brings $O(n^2)$ complexity burden.

Another graph theory based methodology is AMOEBA. (Estivil-castro & Lee, 2000) In that technique, clusters are constructed based on mean and standard deviation of the edges of Delaunay triangles. In computational geometry, Delaunay triangulation is division of a given dataset into triangles, of which outer circle is devoid of data points. As NC and NCAR, AMOEBA is parameter free but has lowered complexity to $O(nlog(n))$.

Grid-based approaches, on the other hand, work by partitioning the data set into a grid structure with certain sizes. While STING (Wang, Yang, & Muntz, 1997) uses the rectangular grid structure in a hierarchical way, CLIQUE (Agrawal, Gehrke, Gunopulos, & Raghavan, 1998) is based on density within the grid structure. However, in these approaches, the quality of the clustering output is highly dependent on the grid size and geometric shape, and the determination of the grid size and shape is not straightforwardly selected. As another instance of grid based clustering methodologies, Adaptive Grid Based Forest-Like (AGFC) can be given. (Cheng, Ma, Ma, Yuan, & Yan, 2022) In AGFC, by a startup parameter, automatic grid generation system is used and according to the gap between density peaks and valley. Then it merges, cells to build up clusters. It also eliminates small sized cluster at the end of the process.

In addition, fractal geometry-based approaches such as FC (Barbara & Chen, 2000) is based on data structures with the same characteristics as the whole, while model-based approaches cluster by learning through statistical learning or neural networks. For example, COBWEB (Fisher, 1987) is a statistical learning-based clustering method that

works according to heuristic criteria in which each attribute is treated as a different distribution.

There are some hybrid clustering methodologies like LDP-MST. (Cheng, Zhu, Huang, Wu, & Yang, 2021) It is both graph theoretical based and density based clustering methodology. It utilizes local density peaks to build up MST. Then it repeatedly cuts the longest edge until all a priori defined number of clusters are unearthed.

All the traditional clustering algorithms which are discussed above are summarized in Table 2.1, Table 2.2 and Table 2.3 along with their properties in detail.

Table 2.1. *Traditional Clustering Approaches and Algorithms*

Approach	Algorithm	Year	Complexity
Partition	K-Means	1967	O(k*n*t*d)
Partition	K-Medoids	2009	$O(k*(n-k)^2)$
Hierarchical	BIRCH	1996	O(n)
Hierarchical	CURE	1998	$O(s^2log(s))$
Distribution	EM	1999	$O(n^2*k*t)$
Density	DBSCAN	1996	O(n*log(n))
Density	OPTICS	1999	O(n*log(n))
Density	Mean Shift	1975	Kernel
Density	DBCLASD	1998	O(n*log(n))
Graph	CLICK	2000	O(k*f(v,e))
Graph	MST	1999	O(e*log(v))
Graph	NC	2015	$O(n^3)$
Graph	NCAR	2019	$O(n^2)$
Graph	Norm. Cuts	2000	$O(n^2)$
Graph	CHKNN	2018	$O(k*n^2)$
Graph	AMOEBA	2000	O(nlog(n)).
Graph	SM	2000	Eigenvector+Heu.
Graph	NJW	2002	Eigenvector
Graph	DAN	2015	N/A
Grid	STING	1997	O(n)
Grid	CLIQUE	1998	$O(n+k^2)$
Grid	ROCK	1999	$O(n^2*log(n))$
Grid	AGFC	2022	O(n)
Fractal	FC	2000	O(n)
Statistical	COBWEB	1987	O(distrubution)
Hybrid	LDP-MST	2021	$O(n^2)$

*Heu:Heuristics

Table 2.1. (Continued) *Traditional Clustering* Approaches *and Algorithms*

Algorithm	Preliminary Number of Cluster Requirement	Input Parameter Requirement	Arbitrary Shaped Cluster	Large Scale Data
K-Means	Yes	No	No	Yes
K-Medoids	Yes	No	No	No
BIRCH	Yes	No	No	Yes
CURE	Yes	Yes	No	Yes
EM	Yes	Yes	No	No
DBSCAN	No	Yes	Yes	Yes
OPTICS	No	Yes	Yes	Yes
Mean Shift	Yes	Yes	Yes	No
DBCLASD	Yes	Yes	No	Yes
CLICK	No	No	Yes	Yes
MST	No	Yes	Yes	Yes
NC	No	No	Yes	No
NCAR	No	Yes	Yes	No
Norm. Cuts	No	Yes	Yes	No
CHKNN	No	Yes	Yes	Yes
AMOEBA	No	No	Yes	Yes
SM	No	Yes	Yes	No
NJW	No	Yes	Yes	No
DAN	No	No	Yes	No
STING	No	Yes	Yes	Yes
CLIQUE	No	Yes	Yes	No
ROCK	No	Yes	Yes	No
AGFC	No	Yes	Yes	Yes
FC	Yes	No	Yes	Yes
COBWEB	Yes	No	Yes	Yes
LDP-MST	Yes	Yes	Yes	No

*"Yes" refers to "required" or "convenient"; "No" refers to "not required" or "not convenient"

Table 2.1. (Continued) *Traditional Clustering Approaches and Algorithms*

Algorithm	High Dim. Feature Space	Intra Cluster Density Variations	Inter Cluster Density Variations	Noise / Outlier
K-Means	No	No	No	No
K-Medoids	No	No	No	No
BIRCH	Yes	No	No	No
CURE	Yes	No	No	No
EM	No	No	No	No
DBSCAN	No	No	No	Yes
OPTICS	No	No	Yes	Yes
Mean Shift	No	No	No	No
DBCLASD	Yes	No	No	No
CLICK	No	Yes	Yes	Yes
MST	No	Yes	Yes	No
NC	No	No	No	Yes
NCAR	No	No	No	Yes
Norm. Cuts	No	Yes	Yes	Yes
CHKNN	No	Yes	Yes	Yes
AMOEBA	No	Yes	Yes	Yes
SM	Yes	Yes	Yes	No
NJW	Yes	Yes	Yes	No
DAN	Yes	Yes	Yes	Yes
STING	Yes	No	No	No
CLIQUE	Yes	No	No	No
ROCK	Yes	No	No	Yes
AGFC	Partially	No	No	Yes
FC	Yes	No	No	No
COBWEB	No	No	No	No
LDP-MST	No	No	No	Yes

*"Yes" refers to "convenient"; "No" refers to "not convenient"

2.1.2. Modern Clustering Algorithms

In addition to traditional approaches and their various derivatives, there are also approaches developed for different or more than one purpose and adapted to be used in the solution of clustering problems. Modern clustering approaches can be subcategorized as Neural Network based algorithms, kernel-based algorithms, herd algorithms, swarm

intelligence-based algorithms, quantum theory-based algorithms and other modern approaches. Some of the well-known methodologies are discussed below as an example to each approach category.

Self-organizing maps (SOM) (Kohonen, 1990) is artificial neural network based clustering and data visualization tool which is also a data visualization / mapping tool.

The support vector clustering (SVC) (Ben-Hur, Horn, Siegelman, & Vapnik, 2002) is one of the most efficient kernel-based approaches. The aim in SVC is to find the circle with the smallest radius that will cover all data points in the high-dimensional feature space, then reduce this circle to the original dimensions and determine the boundary of the clusters and assign data within that boundary in a cluster. Since, SVC has high computational complexity, it is not convenient for high-scale datasets, but convenient for arbitrarily shaped cluster structures and noise type data. It does not need number of original classes in the dataset as antecedent.

There are also some metaheuristic approaches used for clustering problems as well. The most common used ones among these approaches are evolutionary algorithms like Genetic algorithms (GA) (Maulik & Bandyopadhyay, 2000) or swarm intelligence-based algorithms like ant colony algorithms (ACO) (Lumer & Faieta, 1994), particle-swarm optimization (PSO) (der Merwe & Engelbrecht, 2003) and Artificial Bee Colony Optimization (ABC). In ACO, the data is first randomly distributed on the two-dimensional grid and each data is assigned or not assigned to a cluster according to the decision of each ant, until the stopping criterion is satisfied and a suitable clustering output is obtained. On the other hand, the particle-swarm optimization based clustering approach defines the data points as a particle and the cluster centers are updated according to the particle's location and speed based on the result of another clustering algorithm, and works iteratively until a suitable output is obtained. Artificial bee colony algorithms (ABC) (Karaboğa & Öztürk, 2011) is also one of the clustering approach based on swarm intelligence. The aim of this approach is to update the datasets to which data points are assigned to clusters at local and global level by mimicking the aging behavior of three kinds of bee species whose task is to find food sources.

One recent example of swarm intelligence based clustering technique is PSO based Density Peaks Clustering (PDPC). (Cai, Wei, Yang, & Zhao, 2020) In PDPC, particle swarm optimization is used by connected with density peak clustering algorithm to refrain the defects of density peaks clustering in terms of cut-off distance value.

Another metaheuristic based approach for clustering is a Origin Destination – Ant Colony Optimization (OD-ACO). (Song, et al., 2019) for real life case clustering. Flows between origin to destination are clustered "'residential-area to work-area" pattern discover problem in urban planning. ACO is used for clustering in that study and arbitrarily geometric shaped flow patterns are obtained.

The main advantage of swarm intelligence-based approaches is that they produce a good clustering result without being stuck on local optima. Clustering quality is not affected by the shape of the clusters, they have the ability to reveal noise type data, and they do not need a preliminary number of cluster. However, due to their high complexity, they are not effective on large-scale datasets and they are used to fine-tune the output of any other clustering method. A more detailed literature review is available in (Karaboğa & Akay, 2009).

Quantum theory-based approaches have also been developed for clustering problems. Dynamic quantum clustering (DQC) (Weinstein & Horn, 2009), inspired by quantum theory, aims to find the potential energy of each data point with the Schrodinger inequality. By determining the local minimum points with an optimization technique on the inequality, the data with the lowest energy is determined as the cluster center. Then, a clustering output is obtained by assigning each data to the predetermined cluster centers with a distance function. As in other modern clustering approaches, computational complexity is higher and output quality is dependent on the choice of input parameters. It is only suitable for datasets containing convex cluster structures.

Other similar modern clustering approaches also show similar features to the approaches discussed above.

All the modern clustering algorithms which are discussed above are summarized in Table 2.4, Table 2.5 and Table 2.6 along with their properties in detail.

Table 2.2. *Modern Clustering Approaches and Algorithms*

Approach	Algorithm	Year	Complexity
Neural Nets	SOM	1990	-
Kernel	SVC	2002	O(kernel)
Metaheuristics	GA	2000	GA - Complexity
Metaheuristics	OD-ACO	2019	$O(n^2)$
Metaheuristics	ACO	1994	ACO - Complexity
Metaheuristics	PSO	2003	PSO - Complexity
Metaheuristics	PDPC	2020	$O(n^2)$
Metaheuristics	ABC	2011	ABC - Complexity
Quantum	DQC	2009	O(kernel)

Table 2.2. **(Continued***) Modern Clustering Approaches and Algorithms*

Algorithm	Preliminary Number of Cluster Requirement	Input Parameter Requirement	Arbitrary Shaped Cluster?	Large Scale Data
SOM	No	No	Yes	No
SVC	No	Yes	Yes	No
GA	Yes	Yes	No	No
ACO	No	Yes	Yes	No
OD-ACO	No	Yes	Yes	No
PSO	No	Yes	Yes	No
PDPC	No	Yes	Yes	No
ABC	No	Yes	Yes	No
DQC	No	Yes	No	No

*"Yes" refers to "required" or "convenient"; "No" refers to "not required" or "not convenient"

Table 2.2. (Continued) *Modern Clustering Approaches and Algorithms*

Algorithm	High Dimensional Feature Space	Intra Cluster Density Variations	Inter Cluster Density Variations	Noise / Outlier
SOM	Yes	No	No	Yes
SVC	No	No	No	Yes
GA	No	No	No	No
ACO	No	Yes	Yes	Yes
OD-ACO	No	No	No	No
PSO	No	Yes	Yes	Yes
PDPC	No	No	No	No
ABC	No	Yes	Yes	Yes
DQC	Yes	Yes	Yes	Yes

*"Yes" refers to "convenient"; "No" refers to "not convenient"

2.2. Clustering Algorithms Based on Holistic and Non-Holistic Manner

Clustering algorithms can be classified in terms of their holistic or non-holistic algorithmic manner. Some methodologies (e.g. k-means) consider a dataset as a whole and produce a clustering output on that holistic basis. On the other hand, there are some methodologies which take only a small portion of a dataset and produce clustering output adaptively. In the next subsections, holistic and non-holistic categorizations are given in detail.

2.2.1. Holistic Algorithms

Partition based algorithms (k-means, fuzzy c-means, k-medoids…) are all holistic approaches. They consider dataset as an input and make a partitioning. But while making partitions, all the dataset is considered as well. Equation 2.1 is the formula of k-means objective function, while Equation 2.2a shows the formula of fuzzy c-means objective function and finally, Equation 2.2b is the membership degree constraint formula of each data in the dataset. ($w_{ij} = [0,1]$) (MacQueen, 1967) (Bezdek , Ehrlich , & Full , 1984) In these equations, k refers to cluster id, n refers to item index in cluster k, X_i^j refers to item i of cluster k, C_j is the center of cluster j and w_{ij} is the membership degree of item i to cluster j.

$$\arg\min\left(\sum_{j=1}^{k}\sum_{i=1}^{n}\left\|X_i^j - C_j\right\|^2\right) \tag{2.1}$$

$$\arg\min\left(\sum_{j=1}^{k}\sum_{i=1}^{n} w_{ij}\left\|X_i^j - C_j\right\|^2\right) \tag{2.2a}$$

$$where;\ w_{ij} = \frac{1}{\sum_{k=1}^{c}\frac{\left\|X_i^j - C_j\right\|^2}{\left\|X_i^j - C_k\right\|^2}} \tag{2.2b}$$

Hierarchical algorithms (BIRCH, CURE, ROCK…) are based on dendrogram, as a result they have to deal with the dataset in a holistic manner. Since dendrogram creation requires to use each data in the dataset, the hierarchical algorithms must be holistic like partition based algorithms and they usually need N^2 size similarity matrix, where n represent the number of data in the dataset.

Distribution based algorithms like Expectation Maximization (EM) also has to take a dataset as a whole into account, since they have to use all the data to fit a probability distribution.

Grid based approaches like STING, CLIQUE or AGFC follow some holistic manner, since the clustering is done using a certain kind of a holistic grid on dataset.

Similarly, to other approaches, Fractal Geometry based algorithms like FC and Statistical Learning based algorithms like COBWEB, treats a dataset in a holistic manner, since they share similar characteristics with other holistic algorithms.

Most of the modern clustering algorithms (SOM, SVC, DQC, GA, ABC, ACO, PSO and other metaheuristics) are also uses a holistic concern.

2.2.2. Non-holistic Algorithms

Density based algorithms like DBSCAN, OPTICS, Mean Shift… and graph theoretical based algorithms like CLICK, MST, LDP-MST, NC, Normalized Cuts, NCAR, CHKNN, AMOEBA follow a non-holistic concern contrarily to holistic approaches.

Most of these algorithms take only a small portion of a dataset (e.g. within a circular window or some connected neighbor data), and by using this small portion and neighborhood relations, clusters are constructed respectively in an adaptive way. This manner provides a capability of detecting arbitrary geometric shaped clusters and an

enhanced robustness. However, clusters are assumed as convex - circular shaped around a centroid in holistic approaches.

There are two types of neighborhood which is used in these approaches: Epsilon neighborhood and pairwise point neighborhood or k-nearest neighborhood. DBSCAN, OPTICS and Mean Shift prefer epsilon neighborhood, NC, NCAR, AMOEBA uses pairwise point neighborhood (like in Gabriel Graphs) and CLICK, MST, Normalized Cuts, CHKNN and LDP-MST follow a well-known graph theoretical concept, K-nearest neighbor.

2.3. Arbitrary Geometric Shaped Cluster Detection Algorithms

Some traditional and modern clustering techniques exist in the literature for detection of arbitrary shaped clusters, by using epsilon neighborhood, pair-wise point neighborhood and k-nearest neighbor relations.

DBSCAN is the earliest techniques in that context. In that methodology, by two input parameter (epsilon and MinPts), a dense neighborhood is found and expanded by connectivity relations until all the points in the dataset is visited and assigned to a cluster or declared as outlier. OPTICS is an improved version of DBSCAN, in which epsilon is determined dynamically by k-nearest neighbor distance of a core point. But nonetheless, OPTICS also relies on an input parameter: MinPts. Both methodology have the ability of detecting arbitrary shaped clusters but quality of the clustering schema is highly dependent on input parameters, it leads to some trial and error process.

CURE and ROCK are hierarchical clustering methodologies for arbitrary shaped cluster detection. In CURE, a random sample is drawn from the dataset, then sample is partitioned into sub-clusters and each sub cluster is identified by its centroids, then by means of this representative points sub-clusters are merged together by using same approach. ROCK follow the same manner with CURE. A random data sample is drawn then clustering process is carried out on this sample with agglomerative way and then sub-clusters are merged based on their similarities. Both of them suffers from the same problem, input parameter estimation and determination of right size of random samples. Their complexity is also higher than $O(n^2)$ in general.

In order to handle the same problem, some other techniques which run on graph theoretical perspective are developed using pairwise point neighborhood as well. In NC algorithm, based on Gabriel Graphs, density is measured within a sphere which has a diameter as distance of two points. As a result, to find the densest region, an n-square

density matrix is needed, such that n is size of the dataset. Then direct and indirect neighborhood relations are determined to build clusters as an expansive way. Biggest advantage of NC is being input parameter free but having higher complexity relatively to epsilon neighborhood approaches.

NCAR is an extension methodology of NC algorithm. In NCAR, core points are similarly found as in NC and assigned as target points at first, then for each given target points, farthest points are located and used as counterpart to make pairwise point neighborhood and start forming primary Apollonius regions as clusters. Likewise, NCAR also brings $O(n^2)$ complexity because of pairwise neighborhood density check at start.

One more arbitrary shaped cluster detection methodology is hybrid K-NN (CHKNN) which is based on graph theory. K-NN neighborhood strategy is followed in CHKNN and firstly sub-clusters are built up then they are merged by graph connectivity. Similarly, Normalized Cuts is another graph theoretical based clustering methodology which can be used for arbitrary shaped cluster detection. It deals with clustering as a graph partition problem and consider both total dissimilarity between clusters and similarity within clusters. Both CHKNN and Normalized Cuts suffer from larger scale datasets since the both need similarity matrix which brings $O(n^2)$ complexity burden. LDP-MST is convenient for arbitrarily shaped cluster detection and it is hybridization of density based and graph theoretical based. It has similar features with the algorithms in both class.

Another graph theory based methodology is AMOEBA. In that technique, clusters are constructed based on mean and standard deviation of the edges of Delaunay triangles. In computational geometry, Delaunay triangulation is division of a given dataset into triangles, of which outer circle is devoid of data points. As NC and NCAR, AMOEBA is parameter free but has lowered complexity to $O(n\log(n))$.

Grid based clustering methodologies are also convenient for detecting arbitrary shaped clusters like STING, CLIQUE or AGFC. Despite the fact that, they have lowest complexities among the traditional approaches, respectively as $O(n)$, $O(n+k^2)$ and $O(n)$, quality of the clustering schema is highly dependent on grid size and geometry and that determination is not straightforward like input parameter estimation.

Two modern approaches may be also exemplified for the capability of handling detecting arbitrary shaped clusters. SOM is a clustering and data visualization tool and SVC is kernel based clustering methodology which carry data into upper dimension then map them to the original dimensions with a minimal circle information covering data. In

contrast to traditional approaches, SOM and SVC has much higher complexity and also needs some kernel parameters.

Beside all clustering approaches above, another family of methodologies has also arbitrary shaped cluster detection capabilities: Heuristics. GA, ACO, OD-ACO, PSO, PDPC and ABC and other heuristic procedures are also used to solve clustering problems. They have also some drawbacks. GA, ACO, PSO and ABC are mostly uses the outputs of some other clustering techniques as input or they are used as a post clustering schema restoration. Input parameter determination and true operator selection bring estimation problem just as in many other approaches.

2.4. High Dimensional Feature Space Algorithms

Some traditional and modern techniques exist in the literature for handling high dimensionality for clustering. Beyond that clustering approaches, there are some dimension reduction techniques, which do not make any clustering, but as a preprocessing process, they may be applied on the datasets to reduce number of dimensions to a reasonable level.

A comprehensive review upon clustering methodology approaches and algorithms are given in (Xu & Tian, 2015) and (Xu & Wunsch, 2005). In that studies, within traditional approaches; CURE (Guha, Rastogi, & Shim, 1998) and ROCK (Guha, Rastogi, & Shim, 1999) as hierarchical methods, DBCLASD (Xu, Ester, Kriegel, & Sander , 1998) as a distribution based method, STING (Wang, Yang, & Muntz, 1997) and CLIQUE (Sharan & Shamir, 2000) as grid based methods, FC (Barbara & Chen, 2000) as a fractal geometry based method, SOM (Kohonen, 1990) as a model based method are exemplified for the convenience for the high dimensional feature space. Few spectral clustering methods, SM (Shi & Malik, 2000), NJW (Ng, Jordan, & Weiss, 2002) and DAN (İnkaya, 2015) are categorized as traditional approaches as well for the convenience for the high dimensional feature space by means of spectral graph theory.

Another hig dimensional clustering methodology is a feature-reduction multi-view k-means (FRMVK). (Yang & Sinega, 2019) FRMVK is a subspace clustering approach for multidimensional feature space and it is based on the idea that irrelevant features are eliminated in multi-views.

Table 2.7 and Table 2.8 summarizes that clustering algorithms which are convenient for high dimensional dataset with some of their properties. *“Yes” refers to “required” or “convenient”; “No” refers to “not required” or “not convenient”.

Table 2.3. *Clustering algorithms for high dimensional datasets and their properties*

Algorithm	Approach	Input Parameter Requirement	Preliminary # of Cluster Need
CURE	Hierarchical	Yes	Yes
ROCK	Hierarchical	Yes	Yes
DBCLASD	Distribution	Yes	Yes
STING	Grid Based	Yes	No
CLIQUE	Grid Based	Yes	No
FC	Fractal Geometry	Yes	No
SOM	Data Vis.	No	Yes
SM	Graph Theory	Yes	Yes
NJW	Graph Theory	Yes	Yes
DAN	Graph Theory	Yes	No
FRMVK	Partition Based	Yes	Yes

Table 2.3. (Continued) *Clustering algorithms for high dimensional datasets and their properties*

Algorithm	Approach	Complexity	Convenience For Large Datasets
CURE	Hierarchical	$O(s^2*\log(s))$	Yes
ROCK	Hierarchical	$O(n^2*\log(n))$	No
DBCLASD	Distribution	$O(n*\log(n))$	Yes
STING	Grid Based	$O(n)$	Yes
CLIQUE	Grid Based	$O(k^2+n)$	No
FC	Fractal Geometry	$O(n)$	Yes
SOM	Data Vis.	O(layers)	No
SM	Graph Theory	N/A	No
NJW	Graph Theory	N/A	No
DAN	Graph Theory	N/A	No
FRMVK	Partition Based	N/A	Yes

2.5. Dimensional Reduction Techniques as Auxiliary Methodologies

When dataset contains multi dimensions, some redundant or correlated features should be removed before clustering applications so as to increase the clustering schema quality, in the pre-processing step. There are two main approaches in dimensional reduction techniques: feature selection or feature elimination and feature extraction methodologies. But in each approach, datasets are considered within a holistic concern

and as a result of this holistic concern, information losses (loss of variance in dataset) are inevitable. (Velliangiria, Alagumuthukrishnanb, & Thankumar, 2019)

-Feature Selection and Elimination Approaches:

In feature selection or elimination approaches, general statistical criteria are used such as chi-square, Pearson's correlation, covariance matrixes or some heuristics such as Variable Neighborhood Search (VNS) and so on in order to determine the necessary and enough number of features or feature subsets.

PROC VarClus is one of the feature selection methodology in a statistical software called SAS. Contrarily to feature extraction approaches, original variables in data are preserved after the dimension reduction process. First features are clustered. Then principal components are selected among the clustered variables. Clustering of variables are done by means of a hierarchical clustering by maximizing explained variance. Number of variable clusters can be determined by the user like setting the value of k in k-means. VarClus can be also defined as orthoblique centroid component cluster analysis. (Harris & Kaiser, 1964)

-Feature Extraction Approaches:

In feature extraction, new features are generated by using original features in the dataset. Principal Component Analysis (PCA), Kernel Principal Component Analysis (Kernel PCA) and Linear Discriminant Analysis (LDA), Auto-encoder, T-distributed Stochastic Neighbor Embedding (t-SNE), Self-Organizing Maps (SOM), Uniform manifold approximation and projection (UMAP) can be given as an example of explicitly feature extraction and/or data visualization techniques. In Table 2.9 (Velliangiria, Alagumuthukrishnanb, & Thankumar, 2019) and Table 2.10 (Velliangiria, Alagumuthukrishnanb, & Thankumar, 2019); some of these methods are given along with some of their properties.

Table 2.4. *Common feature extraction methodologies and their properties*

Algorithm	Linear/Nonlinear	Exploration Type
PCA	Linear	Sequential
Kernel-PCA	Non-Linear	Sequential
LDA	Linear	Partition
Auto Encoder	Linear	Sequential
t-SNE	Non-Linear	-
SOM	Non-Linear	Random
UMAP	Non-Linear	-

Table 2.4. (Continued) *Common feature extraction methodologies and their properties*

Algorithm	Running Criteria	Ending Rules	Scalability
PCA	Min. Inf. Loss	Ranking	Low
Kernel-PCA	Min. Inf. Loss	Ranking	Low
LDA	Predictive Manner	Partition	Low
Auto Encoder	Non-noise Signals	Partition	Middle
t-SNE	KL-Divergence	Fitting	Middle
SOM	Similarity	Iteration Limit	Middle
UMAP	Cross Entropy	Iteration Limit	Middle

For an instance to feature extraction methodologies, PCA is a dimensional reduction methodology in which all the features are extracted into fewer ones while preserving variations on the data set as much as possible in order to obtain lower dimensional feature space. It provides visualization possibility if the number of principal components are set to 2 or 3, or at least lower dimensional feature space. (Pearson, 1901)

PCA can be described as orthogonal linear transformation that transforms the data to a new dimensional space such that the greatest variance by some scalar projection of the data will lie onto the first principal component, the second greatest variance on the second principal component, and so on.

Another example for feature extraction methodologies is t-SNE. It is a nonlinear dimensional reduction technique in which N dimensional data is mapped into lower dimension (2 or 3 is preferred) by preserving similarities between data points. New locations of each data is found by minimizing Kullback-Leibler Divergence level by a well-known optimization method called gradient descent algorithm. The output is a lower dimensional map that reflect the real similarities in original dimension. Gauss kernel function is used in measuring similarities. (Van der Maaten & Hinton, 2008)

2.6. Consequence of the Literature Survey

All the clustering algorithms which are discussed above are summarized from Table 2.1 up to Table 2.6, along with their properties in detail.

Consequently, literature survey shows that neighborhood construction / merging strategy to handle arbitrary shaped clusters has never been discussed from a statistical viewpoint, within a parameter free manner to the best of the authors knowledge. By that point of view, it is expected that clustering schema quality will be higher relatively to other methodology implementations without increasing computational complexity too

much and parameter estimation-free manner. Search and merge procedures will be carried out in a reinforcement learning framework for applicability on density variations problems. And just like other arbitrary shaped cluster detection methodologies, there will be no requirement of preliminary number of original classes is the dataset.

One more consequence of the literature survey is that, except some spectral clustering techniques, no clustering algorithm exist in the literature which is effectively capable of handling high dimensional feature space without preliminary number of clusters, with parameter-free manner and interpretable and visualizing. Moreover, no method exists which can make simultaneously dimensionality reduction and clustering to the best of our knowledge even the methodologies not discussed in that section too.

In Figure 2.1., the proposed algorithm for the arbitrary shaped cluster detection is illustrated in a literature survey tree along with the existing arbitrary shaped cluster detection methodologies.

Similarly, other proposed algorithm which is for the high dimensional feature space is illustrated in a literature survey tree along with the existing high dimensional feature space clustering methodologies in Figure 2.2.

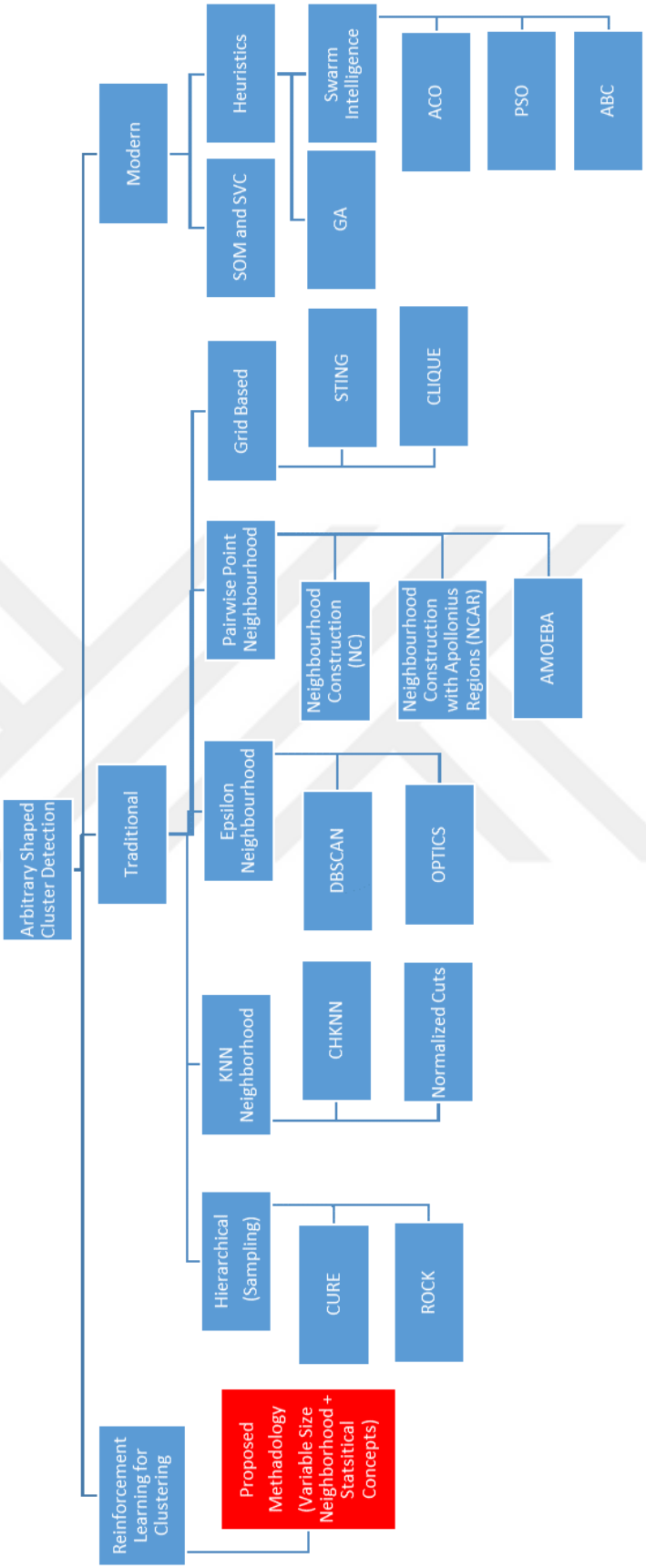

Figure 2.1. *Methodologies for arbitrary shaped cluster detection*

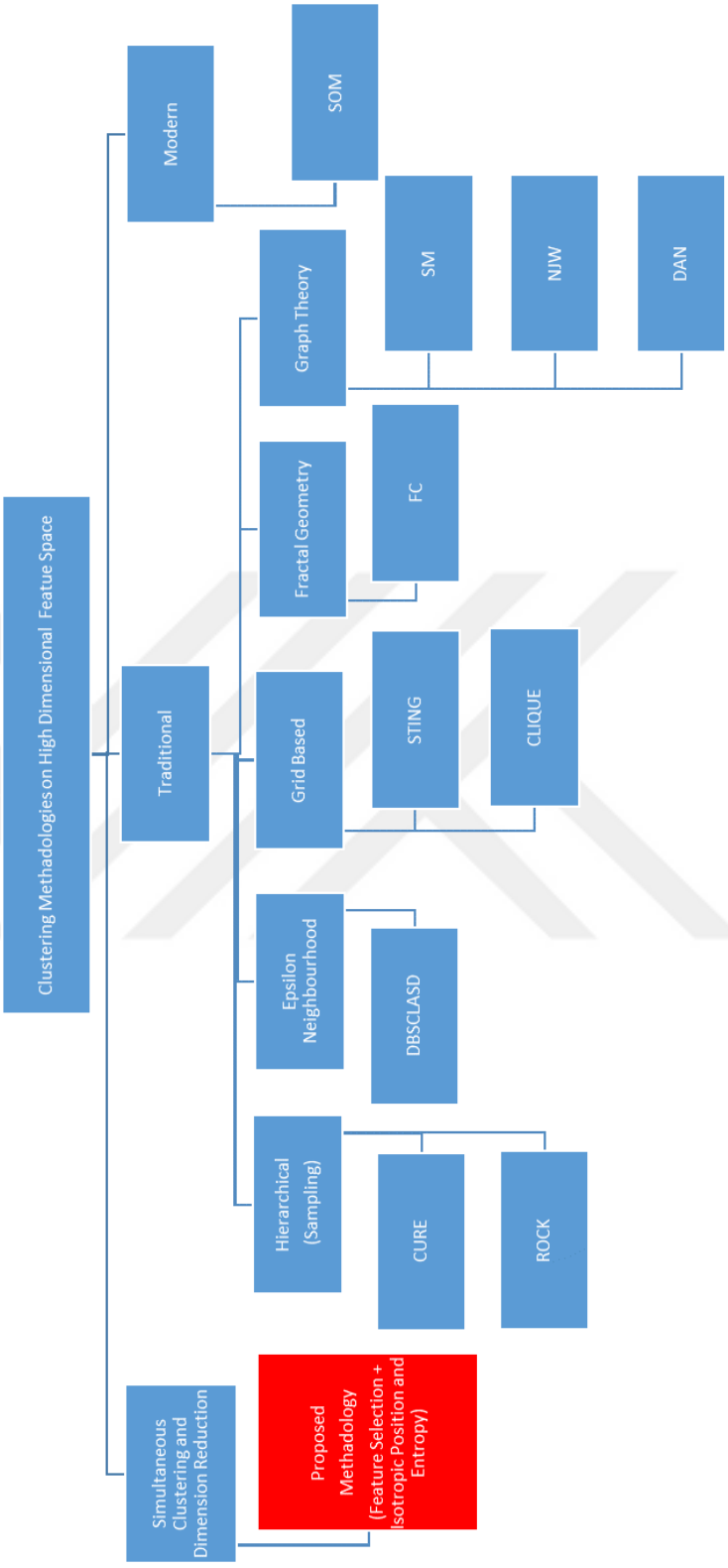

Figure 2.2. *Methodologies for high dimensional feature space clustering*

3. THE PROPOSED METHODOLOGY FOR ARBITRARY GEOMETRIC SHAPED CLUSTER DETECTION

In the section, proposed methodologies for arbitrary geometric shaped cluster detection which is called entropy based neighborhood merging and abbreviated as ENM is introduced in detail respectively, with its properties.

3.1. Basic Terminology

In this subsection, some well-known statistical terminologies and a machine learning concept are introduced: Entropy, KL Divergence (Relative Entropy) and JS Divergence (Symmetric Relative Entropy) and Reinforcement Learning (RL). The statistical terms are used in the proposed methodology in a joint way for detecting arbitrary shaped cluster detection.

3.1.1. Entropy

Entropy refers to disorder or randomness. Shannon introduced entropy as level of surprisal or uncertainty of a stochastic data source and it is measured in unit of bits (alternatively called "shannons") (Shannon, 1948). Entropy value is bounded between [0, ∞). The generalized entropy formula (H_X) is given in Equation 3.1 for any given discrete random variable "X". $P(X_i)$ refers to probability of event i in the sample and $P(X_i)$ needs to satisfy the axioms of probability. The unit of the output value which is coming from the formula is bits when n is 2, dits or hartley when n is 10 and nats when n is e.

$$H_X = -\sum_{x_i \in X} P(x_i) \log_n(P(x_i)) \quad \text{i=1,2,3,…,m} \tag{3.1}$$

Just like in (Chattopadhyay, Pratihar, & De Sarkar, 2011), by adapting Shannon's entropy formula, Equation 3.3 returns the entropy level of all the regions in the dataset, which is constructed by two initial points, by assuming the points in that region approximately bivariate normally distributed with the mean (μ) and covariance matrix (Σ) (see in Equation 3.2.) The region (N_{mn}) is defined as a sphere which has a diameter by the distance of the initial points and points inside that sphere. Since desired entropy levels ($H_{N_{mn}}$) must be independent from the size of the region, each entropy level is normalized by dividing the entropy level by radius^d where d refers to the dimension of the data. (see in Equation 3.4.) The densest region is going to be determined as the region which has maximal entropy level. $P(X_i)$ refers to probability of point X_i.

$$N_{mn} \sim N(\mu_{mn}, \Sigma_{mn}) \tag{3.2}$$

$$H_{N_{mn}} = -\sum_{X_i \in N_{mn}} P(X_i) \log_2 P(X_i) \qquad N_{mn} \in Dataset \tag{3.3}$$

$$H_{N_{mn}} = -\sum_{i} P(X_i) \log_2 P(X_i) \; / radius^d \qquad N_{mn} \in Dataset \tag{3.4}$$

3.1.2. Kullback-Leibler Divergence (D_{KL}) or relative entropy

Kullback-Leibler divergence (D_{KL}) or relative entropy level is a measure of how far or how similar a statistical distribution to reference distribution. (Kullback & Leibler, 1951) If it is zero, two distributions, namely P(x) and Q(x), are identical. The formula of D_{KL} is given in Equation 3.5. To adapt that formula on sets, it is assumed that they are discrete multivariate Gaussians with a mean value (μ) and covariance matrix (Σ). As a result of the adaptation, two different sets, P and Q, are similar in $D_{KL}(P|Q)$ level as in Equation 3.6 where d is the number of data dimensions and tr represents trace of the matrix. Just like entropy formula, DKL is bounded in [0, ∞).

$$D_{KL}(P|Q) = \sum_{x} P(x) * \log\left(\frac{P(x)}{Q(x)}\right) \tag{3.5}$$

$$D_{KL}(P|Q) = \frac{1}{2}\left(\log\left(\frac{|\Sigma_Q|}{|\Sigma_P|}\right) - \mathrm{d} + tr\left(\Sigma_Q^{-1}\Sigma_P\right) + (\mu_q - \mu_p)^T \Sigma_2^{-1}(\mu_q - \mu_p)\right) \tag{3.6}$$

3.1.3. Jensen-Shannon Divergence (D_{JS}) (symmetric relative entropy) and Jensen-Shannon Distance ($Dist_{JS}$)

Since D_{KL} is bounded by [0, ∞), Jensen-Shannon Divergence (D_{JS}) is preferred which is a symmetric generalization of D_{KL}. (Lin, 1991) D_{JS} is bounded between 0 and 1, when 2 based log function is used instead of 10 based log function on D_{KL} formula. The JS distance ($Dist_{JS}$) is formulized as in Equation 3.8 as the square root of D_{JS} (see in Equation 3.7) where M=(P+Q)/2. Moreover; $D_{JS}(P|Q) = D_{JS}(Q|P)$, since D_{JS} computes information loss in symmetric way.

$$D_{JS}(P|Q) = \sqrt{\frac{1}{2}\left(D_{KL}(P|M) + D_{KL}(Q|M)\right)} \;\; (where = (P+Q)/2) \tag{3.7}$$

$$Dist_{JS}(P|Q) = \sqrt{D_{JS}} \tag{3.8}$$

3.1.4. Reinforcement Learning (RL)

Reinforcement Learning (RL) (i.e. Rewarded Learning) is a process in which an agent decides among a set of actions to increase long term rewards and for each action state of the environment changes. A general framework for RL is given in Figure 3.1 (Kaelbling, Littman, & Moore, 1996) RL can be applied in deterministic cases as well as

probabilistic cases. A_t,S_t and R_t refer to action of agent at step t, state of environment at step t and reward for action at step t.

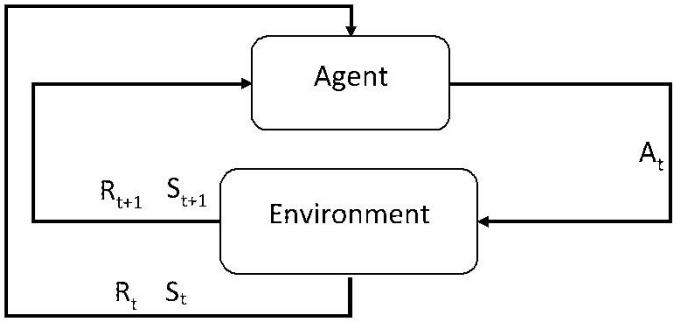

Figure 3.1. *General framework for classical Reinforcement Learning*

RL strategy is used to merge connected regions which have similar compactness levels in the proposed algorithm. To adapt the RL strategy with clustering which is a deterministic case, items which are described above considered as following: Environment is considered as dataset, agent is considered as clustering algorithm itself, states are considered as current elements in the currently forming cluster, actions are considered as merging decision of candidate regions and rewards are considered as (1-D_{JS}) level which are coming from merging decision.

RL works in a recursive way just like in Equation 3.9. Here, Q refers to Q value for each state, S refers to states, R refers to rewards and A refers to actions. Gamma is the discount factor in [0,1] and generally is chosen as 0.9 for a slow convergence. In other words γ is the discount factor which makes future rewards less valuable.

$$Q_i(S) = \max_a R(S,a) + \gamma * Q_{i+1}(S', a') \tag{3.9}$$

Q-Learning is a methodology for evaluating Q function (see in Equation 3.9) for all states. It is a model-free RL algorithm and starts with the initial state. (Watkins, 1989) Q value is expected to converge in the next recursions for the consecutive states. Unlike Q-Learning in which Q values are updated by backpropagation; core region will be initial state for the bellman equation and a forward propagation policy will be applied. For the second iteration, former Q value will be factorized by gamma and summed up a new reward coming from the next merging process.

3.2. The Proposed Methodology: Entropy Based Neighborhood Merging (ENM)

A novel clustering methodology which is based on neighborhood merging strategy by means of statistical frameworks and reinforcement learning is proposed unlike

the papers summarized in the survey. In (Chattopadhyay, Pratihar, & De Sarkar, 2011), an entropy based clustering methodology is developed based on partition approach like k-means, so it assumes that data clusters are in spherical shaped around the centroids. For that reason, it fails in detection of arbitrary shaped clusters. Following a similar way in that work, entropy is used to measure the level of compactness. But this measurement is done for pairwise point neighborhoods, not for all data points in the dataset, to locally determine the minimal disorder region on the dataset. And unlike that work, instead of computing entropy around each data point in the dataset, the entropy level within the pairwise point neighborhood is computed for some certain pairwise points. Then, it finds a couple of points which construct neighborhood with minimal entropy level in the dataset as the locally most compact region. One or both of that initial couple of points are reflected in the direction of mean vector in search for locally minimal entropic region Just like in other neighborhood merge based methodologies, these core neighborhoods are extended by using symmetric relative entropy level (Jensen-Shannon divergence) which is a generalization of Kullback-Leibler divergence level, until all the points are visited and assigned to a cluster or declared as noise. In merging process, reinforcement learning (specifically Q-Leaning) is used in terms of value iteration. Algorithm is awarded by ($1-D_{JS}$) levels between two regions that are merged in each step.

3.3. Attributes of ENM

The proposed algorithm has the following attributes:

- The number of original classes in the datasets is assumed as unknown.
- Arbitrary shaped clusters and/or clusters which have overlapping convex hulls can be detected by the proposed algorithm.
- The algorithm is based on statistical concepts: Entropy and Symmetric Relative Entropy.
- Density is described by the entropy instead of # of points in the epsilon neighborhood and that will provide better clustering schema.
- It has no threshold value for merging compact regions. (Full parameter free)
- There are three types of termination conditions for the proposed algorithm:
    - All the points are visited and assigned to cluster. (For main function)
    - When Q_i (S) and Q_(i+1) (S') convergence. (For merging)
    - No candidate points to be merged for a core region. (For merging)

- It is more sensible to inter-cluster and intra-cluster density variations due to slow convergence.
- Pairwise point neighborhood is preferred in the proposed algorithm instead of epsilon neighborhood in other density based approaches.

3.4. Pseudocodes for ENM

The proposed algorithm works in a recursive way. There is a main function for the detection of locally core region in the dataset and inside the main function there is an expand cluster function, for the merging process of the regions that have similar compactness. The main function is pseudo coded as in Table 3.1. In that table, indentation levels refer to inner statements and noise situations are ignored.

Table 3.1. *Algorithmic Structure for the main function*

While (There is more than two unvisited points in the dataset):
Main step: Select two random points among unvisiteds and set entropy level as 0 and i as 0.
Step 1. While the region has less than 2 points:
Find two nearest neighbor points of these initial points until the region has at least 2 points and set them as new initials.
Endwhile
Step 2. Fit the points inside the region into bivariate normal distribution and compute its entropy.
Step 3. Compute Mean vector of the region excluding visited points.
Step 4. Reflect the initial points in the direction of the mean vector (to the nearest point)
Step 5. If entropy level for one of the candidate regions is higher than the previous one:
Update the initials and Go to step 2.
Endif
Step 6. Set cluster id as i and Run expand-cluster function.
Step 7. If there is no unvisited point left in the dataset:
Terminate.
Endif
Step 8. Else: Increase i by 1 and Go to Main Step.
Endwhile

As for the expand-cluster function, the following pseudocodes are given in Table 3.2. which explains how it works:

Table 3.2. *Algorithmic Structure for the expand-cluster function*

Main step: Set the locally most compact region as core region and find its Radius. **Set j as 0 and QList as [0,0,…,0] (len(levellist)=# of points in the core region.**

For each unvisited point k in the core region:
Step 1. Search for new points (if there is) at most in the diameter distance to the core region points (excluding the core ones and visited ones)
Step 2. If there is no candidate points:
Set i as cluster label of the points in the core region and return.
Endif
Step 3. Else:
Construct new regions with these new points and compute D_{JS} between core region & the new regions and select maximum among them.
Step 4. if levellist[k]=0:
Step 5. Merge new points with the core region, set them as visited and append the Q value for the new state into QList into by the number of new points.
Step 6. Else: If Q levels converge for the two consecutive connected merge:
Stop merging, set i as cluster label of the points in the core region and return.
Endif
Step 7. Else:
Update cluster list and go to step 1 and try for new merges.
EndFor

3.5. Flowchart of ENM

The flowcharts of the ENM are also provided in the appendix part, both for main function and Expand-Cluster function in Figure 0.1 and Figure 0.2 in Appendix.

3.6. Time Complexity of ENM

As explained in pseudocodes in the previous subsection, the proposed algorithm has two functions: Main function and expand cluster function. If the length of the dataset (N) is considered as the input size, time complexity of the proposed algorithm can be deducted as follow:

In main function, distribution fitting, entropy calculation, candidate points search has O(N) complexity at most. The rest of the operations within this function has constant time complexity: O(1). In summary, main function has O(N) complexity.

In expand cluster function, search for the candidate points, computing D_{JS} levels, determination of minimum of the D_{JS} list has O(N) complexity at most. Similarly, the rest

of the operations like Q value computations has constant time (O(1)) complexity. The expand-cluster function is recursive and # of points for that function decreases in each iteration in logarithmic level, moreover Q value calculations for different states is done in recursive way. As a result, overall complexity O(nlog(n)) complexity on average. However, in the worst case, all the pair of points may have to be visited for both locally maximal entropic regions and merging process, the overall complexity may increase up to $O(n^2)$ level.

Consequently, overall time complexity of the proposed algorithm is O(nlog(n)) for the average case and $O(n^2)$ for the worst case.

3.7. A Simple Implementation of ENM

ENM clustering algorithm is designed as a nested function structure. Main function is to determine the maximal entropic neighborhood in the dataset and Expand-Cluster Function is nested in the main function to establish clusters respectively by merging connected similar neighborhoods. By that way, increase in clustering schema quality is aimed without increasing computational complexity relatively to other arbitrary shaped cluster detecting methodologies. Moreover, it has no input parameter setting requirement. An illustrative example is given on a simple spatial dataset which has 17 elements. In Figure 3.2, the original dataset is visualized.

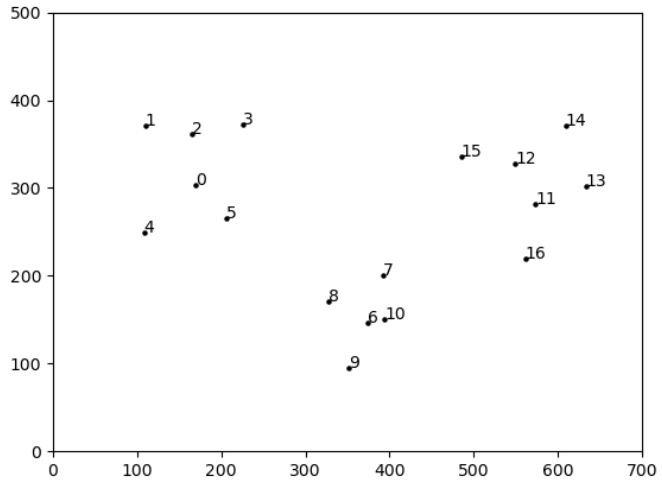

Figure 3.2. *A toy dataset for illustrative example*

As a result of the proposed algorithm, it is aimed to obtain such as clustering schema. C_1={0,1,2,3,4,5}, C_2={6,7,8,9,10} and C_3={11,12,13,14,15,16}. And in Figure

3.3, resultant clustering schema is visualized. Each distinct color represents a distinct cluster.

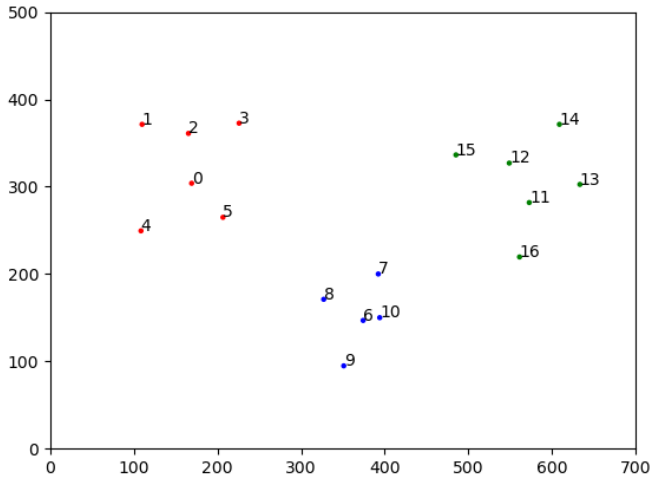

Figure 3.3. *Desired Clustering Schema (Distinctive by colors)*

- Finding the most compact region in the dataset:

To find the most compact region in the dataset, pairwise point neighborhood is used. For a two arbitrary couple of initial points, region is determined as inside the circle which has a diameter in the length of the distance between the points. Firstly, center and radius is calculated. Then the determination of the points within that spherical region is done. If there are other points in that region, a search procedure is applied in order to find the most compact region (region which has highest entropy) in the dataset. Search procedure is based on reflecting initial points in the direction of mean vector. Mean vector is the vector which is the arithmetic average of the points in both axes. Search procedure is typical local search, because initial point is dragged in the direction of the mean vector, candidate points will be nearest point to that new point.

Let Points 1 and 13 be the initial neighborhood constructing regions as in the Figure 3.4. Neigh(1,13) ={0,2,3,4,5,6,7,8,9,10,11,12,14,15,16} and mean vector of the elements inside the region is depicted as an arrow which is precisely mean vector = [384.45, 257.22]. (see as an arrow in the following figures)

Then, as mentioned in pseudocodes, search for the maximal entropic region begins in the direction of mean vector. When Point 1 is reflected in the direction of mean vector, nearest point is 4, similarly when Point 13 is reflected in the direction of mean vector, nearest point is 16. As a result, 4 and 16 are candidate points and Neigh(4,13),

Neigh(1,16) and Neigh(4,16) are candidate regions. Entropy levels are given in 10^5 times to be comparable. Among the candidate regions, maximal entropic region is Neigh(1,16) as seen in Figure 3.4, thus search will continue using new region as initial region, until no more candidate region which is "better in entropy levels" are left in the dataset.

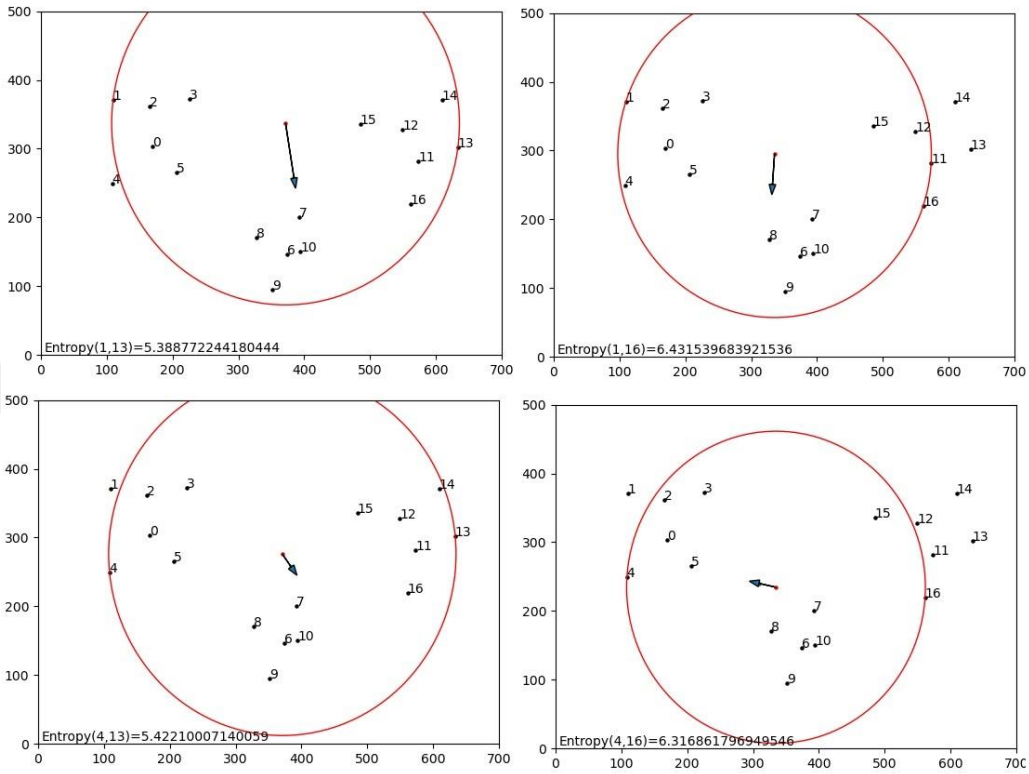

Figure 3.4. *Initial region and candidate regions in the first iteration*

For Neigh(1,16), new candidates will be found by following the same procedure. Point 4 and Point 10 will be candidates, if Point 1 and Point 16 are reflected in the direction of the mean vector of the region. New candidate regions will be Neigh(4,16), Neigh(1,10) and Neigh(4,10) as in Figure 3.5. And Neigh(4,10) will be new core region.

In the third iteration, initial region will be Neigh(4,10) and algorithm will check for the new candidate regions. Neigh(5,10), Neigh(4,9) and Neigh(Neigh(5,9) will be new candidate regions. And Neigh (5,10) will be selected as new region since it has highest entropy level among the candidate regions as in Fig 3.6.

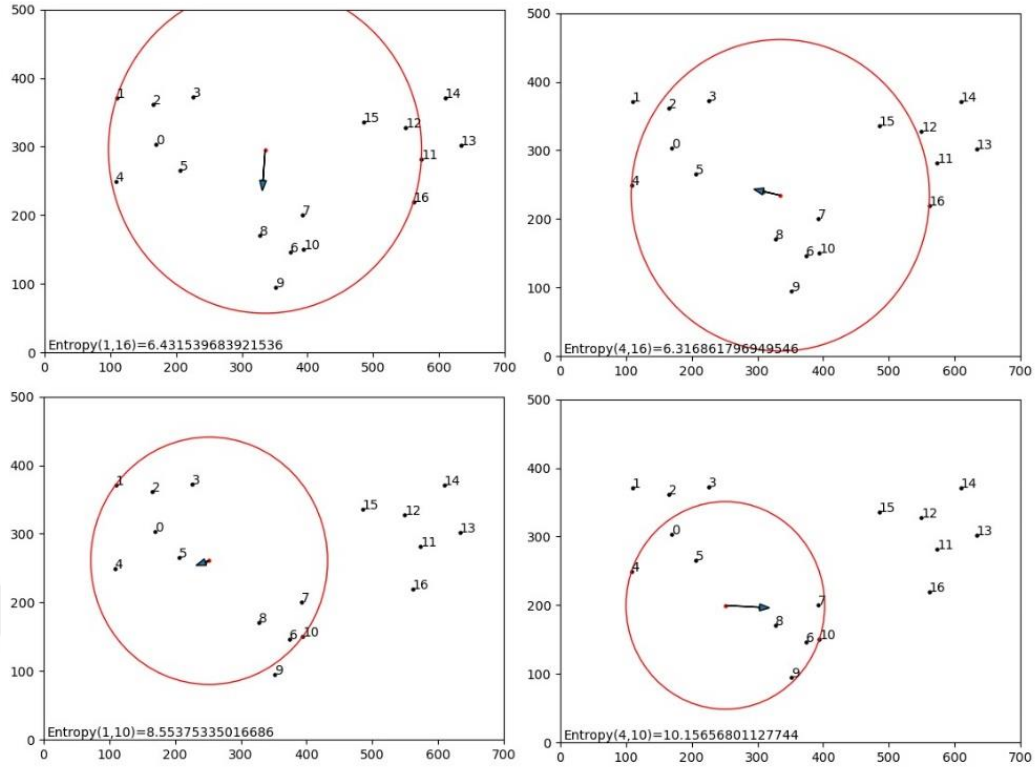

Figure 3.5. *Initial region and candidate regions in the second iteration*

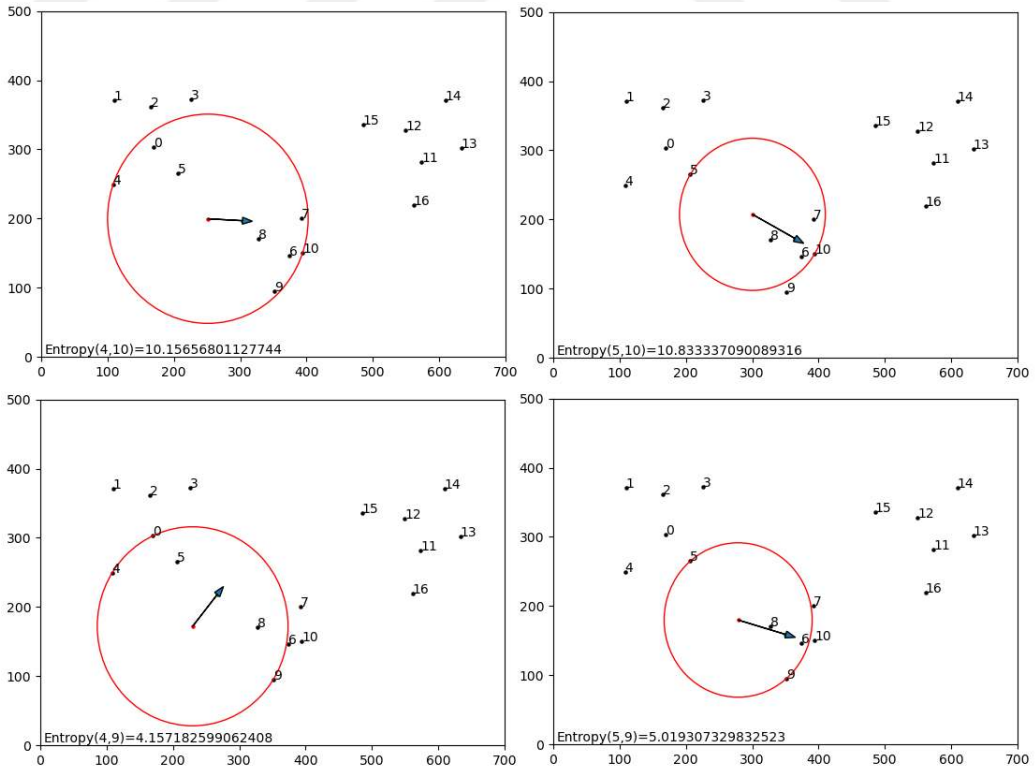

Figure 3.6. *Initial region and candidate regions in the third iteration*

In the fourth iteration, initial region will be Neigh(5,10); Neigh(8,10), Neigh(5,9) and Neigh(Neigh(8,9) will be new candidate regions. And Neigh (5,9) will be selected as new region since it has highest entropy level among the candidate regions as in Figure 3.7. Other region has too less points to compute entropy levels even.

In the fifth iteration, initial region will be Neigh(5,9) and candidate regions will only be Neigh(7,9), since no candidate points is left to reflect Point 5 and Point 9 except Point 7 in the direction of mean. Neigh(7,9) has higher entropy (almost 39.77) than the Neigh(5,9). Consequently, new region will be Neigh(7,9). Since there will be no more candidate points which will provide a better region in entropy level, algorithm stops and determine Neigh(7,9) as the most entropic region and start forming Cluster 1. But, search procedure to expand Neigh(7,9) will not be carried out, because there will be no more candidate points for the inner points in Neigh(7,9) at most in diameter of Neigh(7,9). As a result, clustering process for Cluster1 is complete and Cluster1 will include all the points inside the Neigh(7,9) including Point 7 and Point 9. Cluster1=6,7,8,9,10 as depicted in Figure 3.8. in blue color. These points will not be visited again, moreover in the determination of the next core region, they will be ignored element in region if they exist.

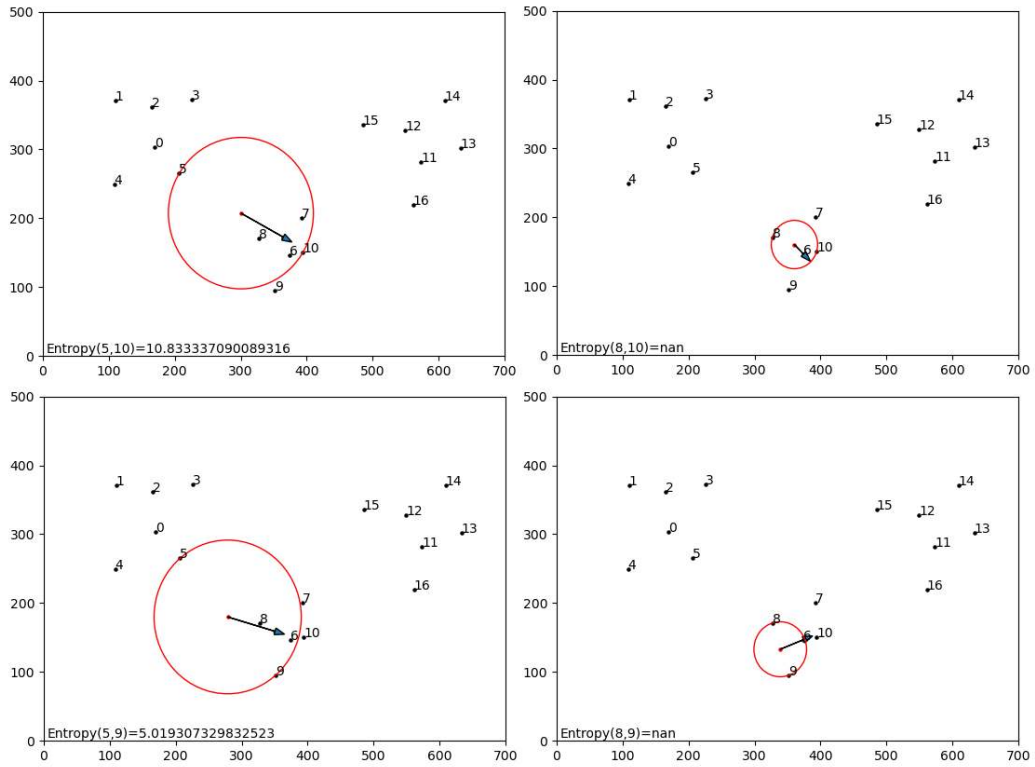

Figure 3.7. *Initial region and candidate regions in the fourth iteration*

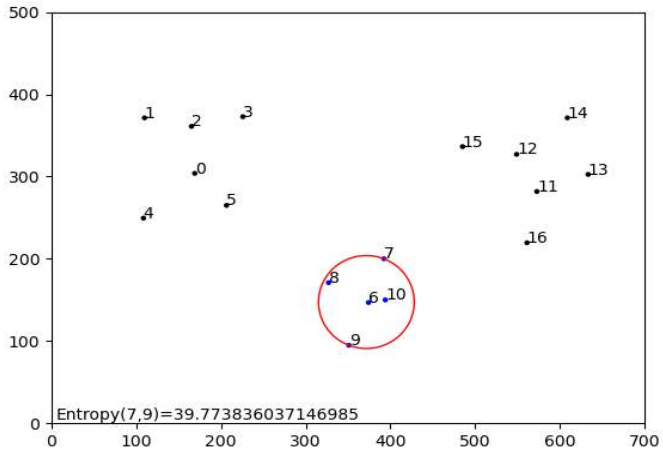

Figure 3.8. *First Cluster Formation (blue colored)*

Since not all points are visited in the dataset yet, by excluding the points in the first cluster, new initials will be set by random just like in the first iteration. Let, Point 3 and Point 15 are the new initials. Since there is no element in the Neigh(3,15), each point will be reflected into Point 2 and Point 12. Neigh(2,15) has highest entropy among the candidate regions; Neigh(2,12), Neigh(2,15) and Neigh(3,12) as in Figure 3.9.

In the second iteration new initial region will be Neigh(2,15) and candidate points will be 4 and 5. Neigh(4,15) has highest entropy among the candidate regions and core region: Neigh(2,15), Neigh(2,5), Neigh(2,4) and Neigh(4,15) as in Figure 3.10. In the third iteration, Neigh(4,3) has highest entropy among the candidate regions: Neigh(4,15), Neigh(1,15), Neigh(4,3) and Neigh(1,3) as in All the intermediary steps are given in Figure 3.11. Just after that iteration, no candidate region will provide a better entropy level and algorithm stops. Similarly in the first cluster, Neigh(3,4) will not get expanded, since there will be no candidate point in the diameter of Neigh(3,4). As a result Cluster 2 will include all the elements in Neigh(3,4). Cluster 2 = {0,1,2,3,4,5}.

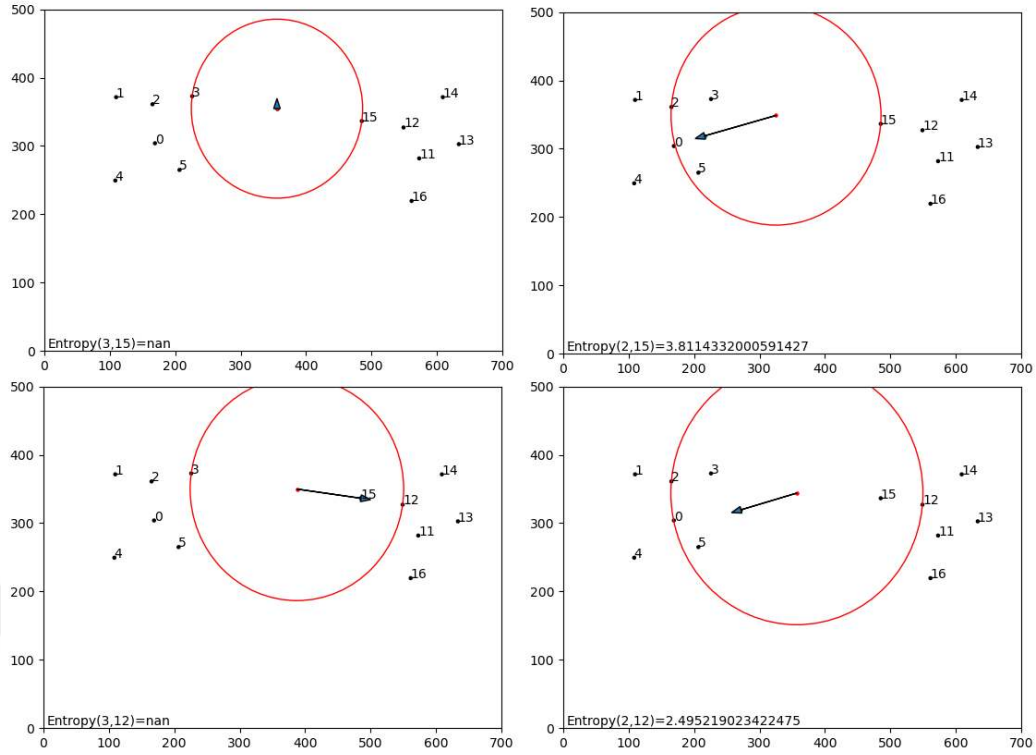

Figure 3.9. *Determination of second core region and first iteration for Cluster2*

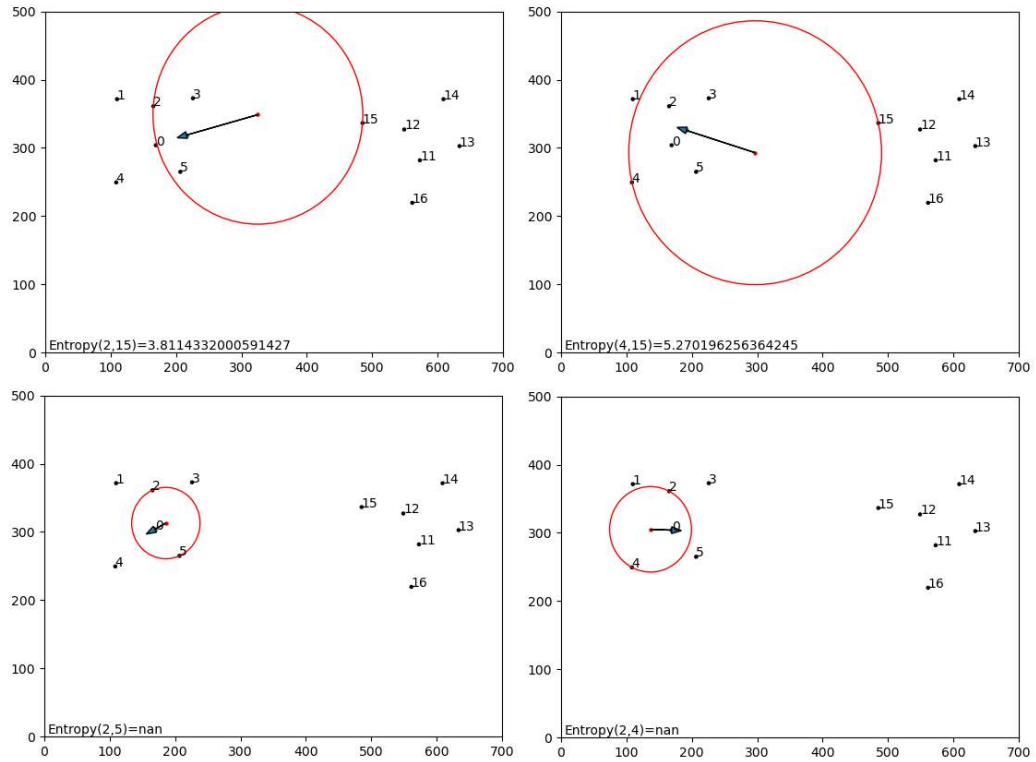

Figure 3.10. *Second iteration for Cluster2*

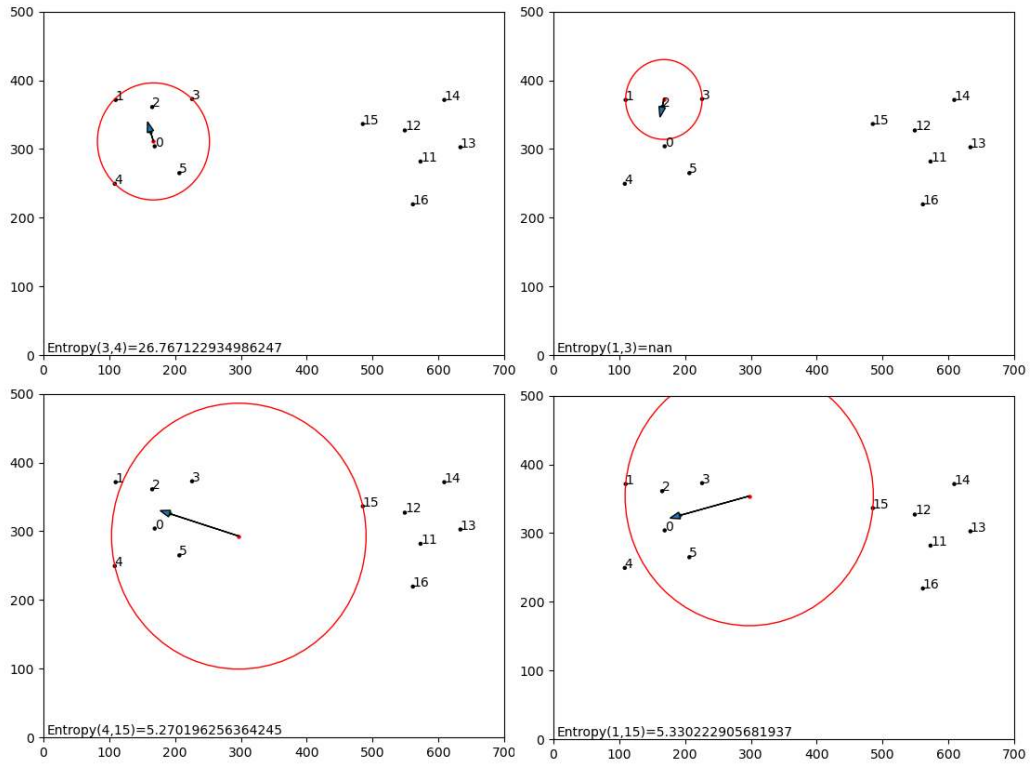

Figure 3.11. *Third iteration for Cluster2*

As for the remaining elements in the dataset, let Points 14 and 16 be the new random initials. In that case, candidate points are Point 11 and Point 12; no better entropy level can be achieved from Neigh(14,11), Neigh(12,16) and Neigh(11,12). Only remaining element in the dataset which is not visited yet is Point 15. But, Neigh(14,15) and Neigh(15,16) will provide no better entropic regions. As a result, Neigh(14,16) will be the core region of the Cluster 3 as in Figure 3.12.

In other words, elements of Neigh(14,16)=[11,12,13] will be the core region. Contrarily to the first two cluster, formation of Cluster 3 hasn't completed yet. Because, merging neighborhoods is possible for the third cluster by searching within the diameter of Neigh(14,16). Figure 3.12. depicts the core region determination of cluster 3. In the next subsection, formation of cluster 3 will be completed.

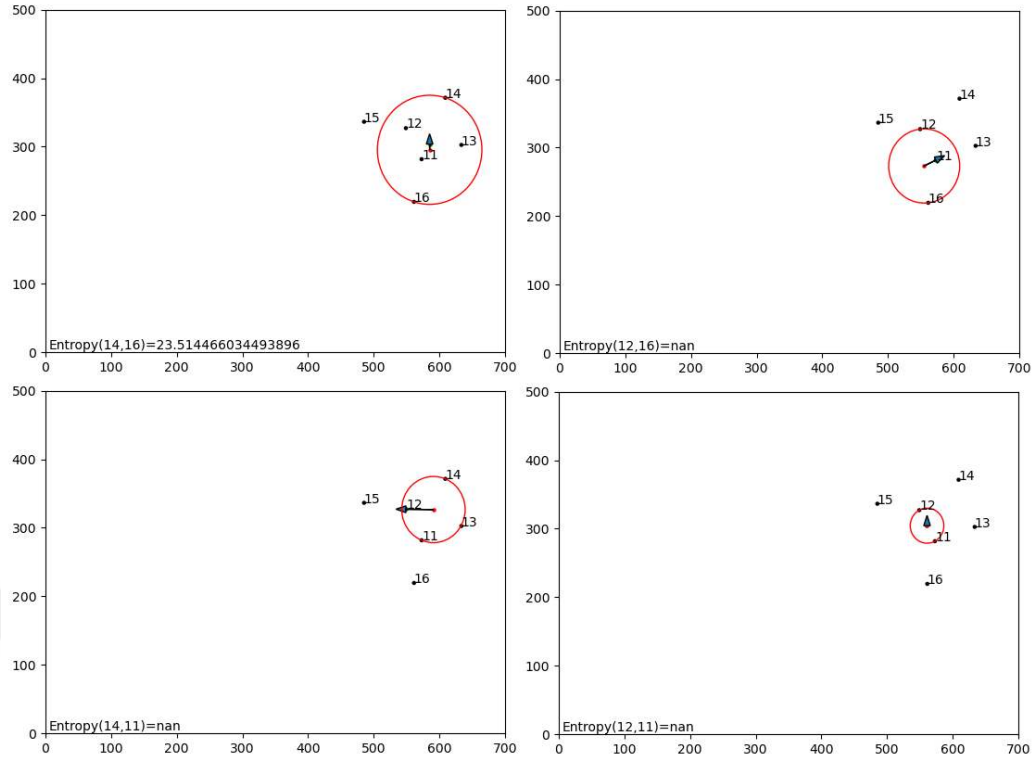

Figure 3.12. *Determination of third core region for Cluster3*

- Merging connected regions that have similar compactness:

As mentioned in the previous sections, merging compact regions will be done according to D_{JS} levels between regions just after the determination of the core regions for each cluster. Firstly, new candidate points will be searched within the distance of the dimeter of the core region to each inner point. After determination of the candidate points for each inner points, new regions will be constructed and D_{JS} levels between core region and candidate region will be computed.

Merging process will be carried out by means of immediate reinforcement algorithm. Bellman Equation (see in Equation 3.10) is used to determine cluster centers (Barbakh & Fyfe, 2007) and (Likas, 1999). Unlike them, immediate reinforcement learning procedure is carried out to merging process of the compact regions to build up clusters in the proposed algorithm. Rewards will be 1-D_{JS} for merging in the iterations, since D_{JS} falls in between 0 and 1. Inıtial state will be the set of elements in the core region and the other states in the latter iterations will be the set of elements in merged compact regions. Actions (a) refers to candidate points which will construct candidate regions with inner points of the core region.

$$Q_i(S) = \max_a R(S,a) + \gamma Q_{i+1}(S',a') \tag{3.10}$$

By means of value iteration, q values for each state will be computed and if two q value converges for consequent iterations, algorithm will stop. Alternatively, two cases will be terminating conditions for the algorithm: no candidate regions to be merged or no unassigned point remains.

To be more illustrative, formation of cluster 3 will be given as an example of merging process. As mentioned in the previous subsection, a core region is set for cluster 3, but there is one more point which is not assigned to any cluster yet: Point 15.

As depicted in Figure 3.13, after a search for a candidate point within the dimeter distance of the core region, Point 15 is found as candidate points. Neigh (11,15), Neigh(12,15), Neigh(13,15) , Neigh(14,15) and Neigh(15,16) will be candidate regions to be merged with Neigh(14,16). Since a reward is coming from the merging process of Neigh(14,16) and Neigh(3,15), algorithm decides to merge them. As a result points in the Neigh(3,15) will be merged with Neigh(14,16.) Eventually, all the points are visited and assigned into a cluster which provides a termination condition. Since the merging process is done in only one iteration, there will be no need to check the convergence of the Q value.

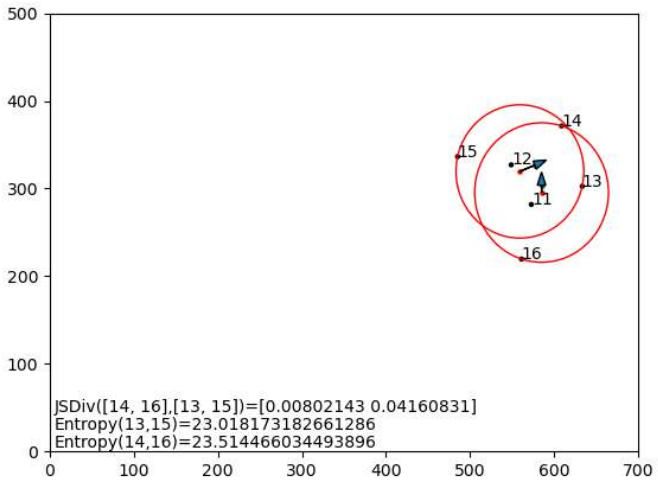

Figure 3.13. *Merging process for the formation of Cluster3*

Consequently, the proposed algorithm ends up a clustering schema which is illustrated in Figure 3.3. With a mathematical saying, Cluster1={6,7,8,9,10}, Cluster2={0,1,2,3,4,5} and Cluster3={11,12,13,14,15,16}.

In case of new candidate points for Point 15, algorithm will follow the same manner and check the terminating conditions respectively. But for the next iterations, new set of

actions, a new state, a new core region and its diameter is going to be defined and they will be used for the consecutive iterations. This framework will provide detection of arbitrary shaped clusters and will be robust to density variations.

3.8. Experimental Analysis of ENM

In this section, the proposed algorithm (ENM) is tested on several spatial datasets to illustrate the its applicability and robustness on arbitrary geometric shaped cluster detection along with noise and density variations problematic.

3.8.1. The Benchmark Datasets

The proposed algorithm is experimentally analyzed on several 2d and 3d benchmark datasets that have arbitrary shaped clusters.

Table 3.3. *Main characteristics of the datasets which are used for experimentation*

Dataset	Size	# of Clust. (n), Dim. (d) n-d	Noisy?*	Overlap?*	Inter Cluster Density Variations*	Intra Cluster Density Variations*
Aggregation	788	7-2	N	P	N	N
Atom	800	2-3	N	N	N	Y
Chainlink	1000	2-3	N	P	N	N
DS31	3100	31-2	P	H	N	N
Flame	240	2-2	P	H	N	N
Jain	373	2-2	N	N	Y	Y
Path Based	300	3-2	P	P	Y	N
R15	600	15-2	N	P	N	N
Spiral	312	3-2	N	N	N	Y
t4.8k	8000	6-2	Y	N	N	N
t5.8k	8000	6-2	Y	P	N	N
t7.1k	10000	8-2	Y	P	N	N
t8.8k	8000	8-2	Y	N	Y	N
Zahn Comp.	399	6-2	N	N	Y	Y

*(N: No, Y: Yes, P: Partially, H: Highly, Clust.: Cluster, Dim.: Dimensions)

These benchmark datasets are acquired from Clustering Basic Benchmark (http-3), Clustering Benchmark/Deric (http-4) and Clustering Datasets-GITHUB Milaan (http-5) websites. The main characteristics of these spatial datasets are summarized in Table 3.3 along with some of their properties.

3.8.2. Validation Indices

The proposed algorithm is evaluated via both external and internal validation indices. For external validation, existing indicators are used but for internal validation, new indicators are developed as well since existent internal measures are not so convenient to evaluate nonconvex arbitrary shaped clusters. In the subsections, these indicators are elaborated.

External Validation: Since ground truth labels of data is given in every dataset which is used in experimentation, a bunch of well-known external validation indices are used. Firstly, v-measure (Rosenberg & Hirschberg, 2007) is used which takes both homogeneity and completeness metrics into account. Related equations are given for the v measure calculation as in Introduction Section. Moreover, adjusted rand index (ARI) (Hubert & Arabie, 1985) and Fowlkes & Mallows Score (FMS) are used as an additional external evaluation indices. Higher values are desirable for both indices. Adjusted RAND Index (ARI) and Fowlkes-Mallows Score (FMS) are defined based on pairwise similarity as in (Hubert & Arabie, 1985) and (Fowlkes & Mallows, 1983) uses the following measures: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

Internal Validation: Since the existing and commonly used internal validation indices such as Hubert index, Dunn Index (DI), Davies-Bouldin (DB) or root mean square standard deviation (RMSSD) is all centroid based internal validation indices and fall too far in accurately measuring the validation of arbitrarily geometric shaped clusters, because of overlapping convex hulls, new intrinsic indices are required.

Therefore, new intrinsic validation measures are also proposed based on entropy increase (or information loss): Mean information loss within clusters(MILWC) and mean information loss between clusters (MILBC). Each measure is computed as in Equation 3.11 and Equation 3.12. Lower values are desirable for both measures. Since a beta threshold is used for neighborhood merging, MILWC takes values less or equal then beta threshold and lower bound will be 0. Information loss levels are computed by using symmetric relative entropy levels ($D_{JS}(P|Q)$) in both measures. P and Q refer to consecutive neighborhood which are merged within cluster k. MILBC returns mean D_{JS} between first core neighborhoods (densest parts) of each consecutive clusters.

$$MILWC = \sum_{i=1}^{k} \sum_{P}^{Q} D_{JS}(P|Q)/|Q| \quad (\forall\, P, Q \in cluster\ k) \tag{3.11}$$

$$MILBC = \sum_{i=1}^{k} D_{JS}(P_k|Q_{k+1})/n \;\; P_k \in clust\_k \; and \; Q_{k+1} \in clust\_k + 1 \tag{3.12}$$

3.8.3. Open Source Library Usage and Computational Environment

Experiments are performed in Python 3.8.5 by using Core i7, 2.8 GHz processer and 8 GB Ram as a computational environment.

For the experimental analysis of the proposed methodology (ENM), some open source libraries are used. “Numpy” is used for mathematical, matrix and list operations, “Time” module is used to measure computation times. “SciPy” is used for to calculate spatial distances and “sklearn” is used for performance indicators (homogeneity, completeness, v-measure, ARI and FMS). Finally, “matplotlib.pyplot” is used for the visualization of the clustering schema results.

3.8.4. Experimental Results and Discussion

Proposed methodology is executed in a single time by setting gamma=0.9 as a discount factor. The entropy level of the regions which have less than 2 elements are assumed as infinity. Results of the experimental analysis is summarized on three different aspects: Clustering Schema Quality by visual check, both internal and external validation index results and execution times for every dataset.

- Clustering Schema Quality:

High quality clustering schema is obtained for all dataset. Quality is proved by visual check. For each benchmark dataset, original dataset and clustering schema of the proposed algorithm are given from Figure 3.14 up to Figure 3.27. To be more quantitative, Table 3.4 is given which summarizes match between original number of clusters and how many of them are correctly (at least as a distinct cluster) unearthed. Clusters are distinctive

by colors while black corresponds to noise in every dataset. γ is set as 0.9 for the proposed algorithm for all datasets.

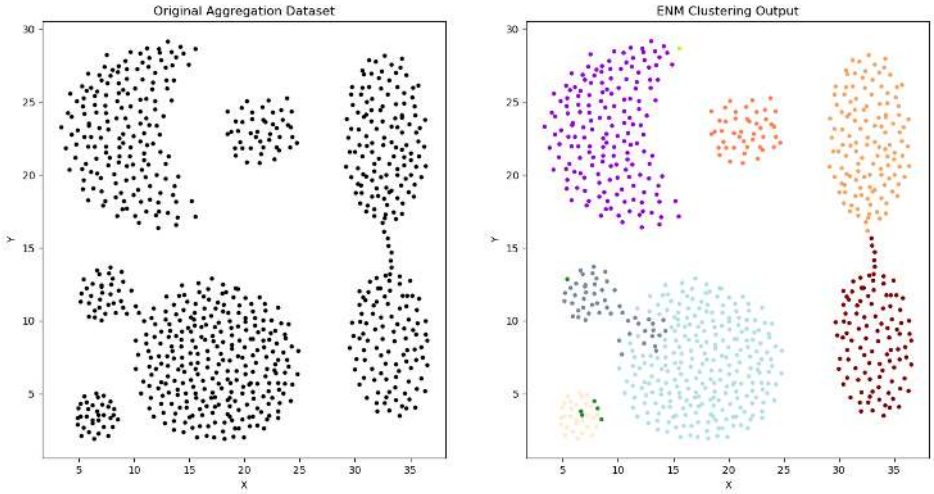

Figure 3.14. *Original Aggregation Dataset vs. ENM Clustering Output**

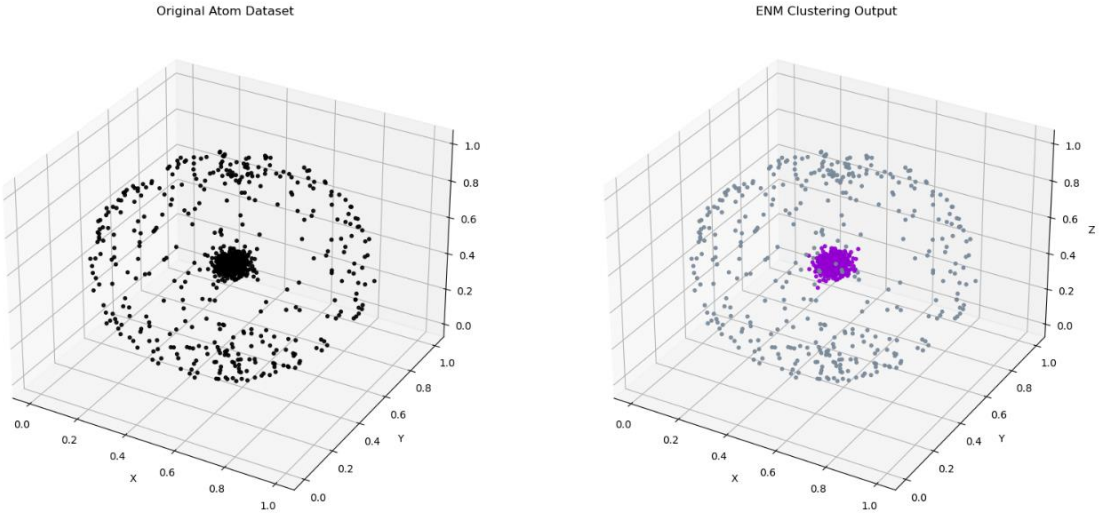

Figure 3.15. *Original Atom Dataset vs. ENM Clustering Output**

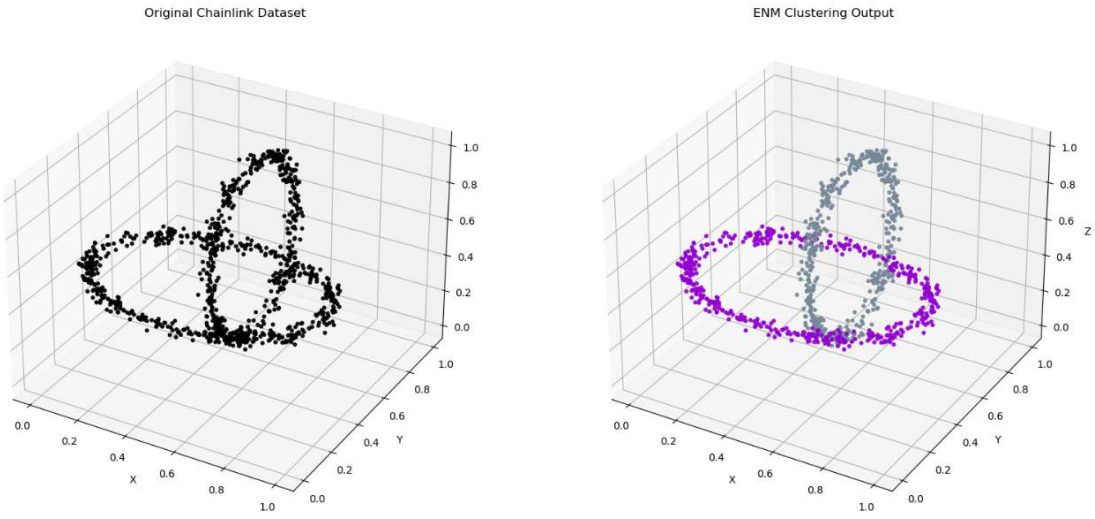

Figure 3.16. *Original Chainlink Dataset vs. ENM Clustering Output**

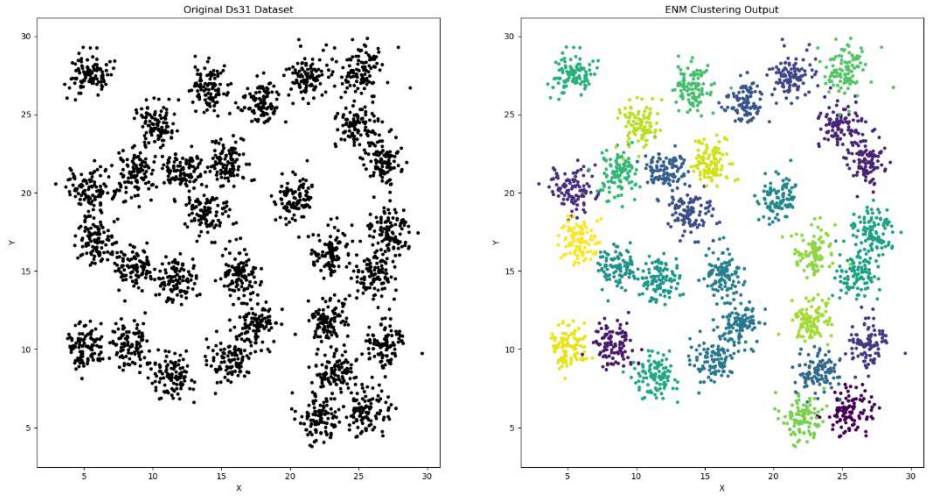

Figure 3.17. *Original DS31 Dataset vs. ENM Clustering Output**

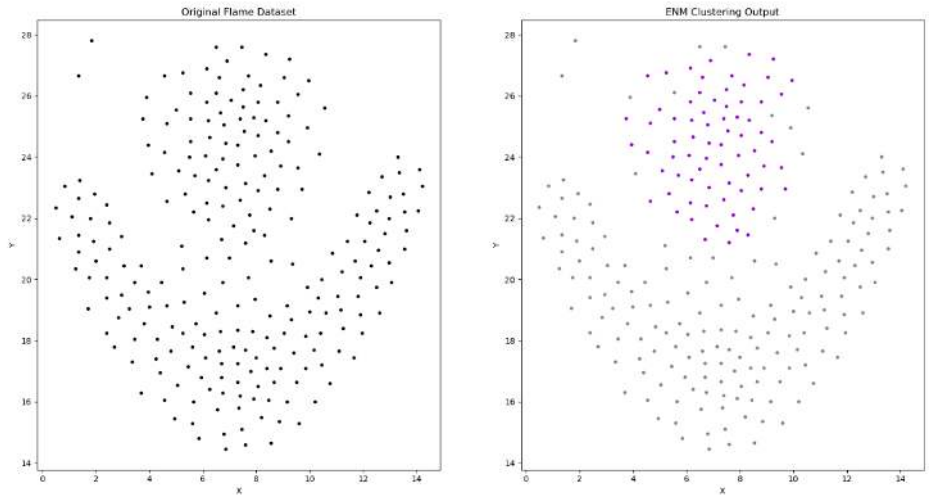

Figure 3.18. *Original Flame Dataset vs. ENM Clustering Output**

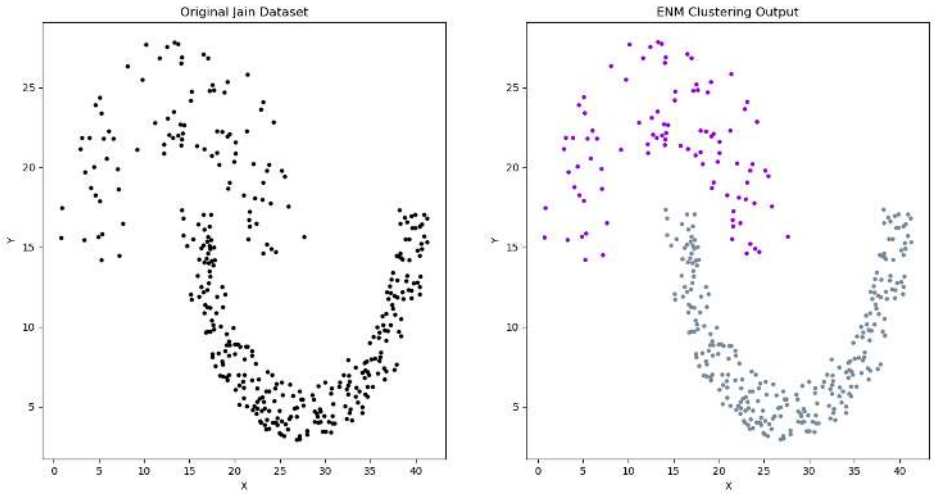

Figure 3.19. *Original Jain Dataset vs. ENM Clustering Output**

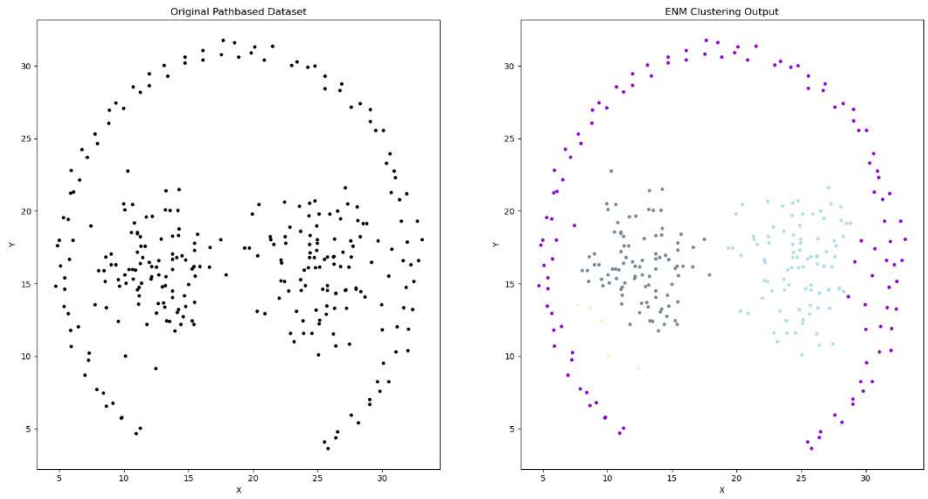

Figure 3.20. *Original Path Based Dataset vs. ENM Clustering Output**

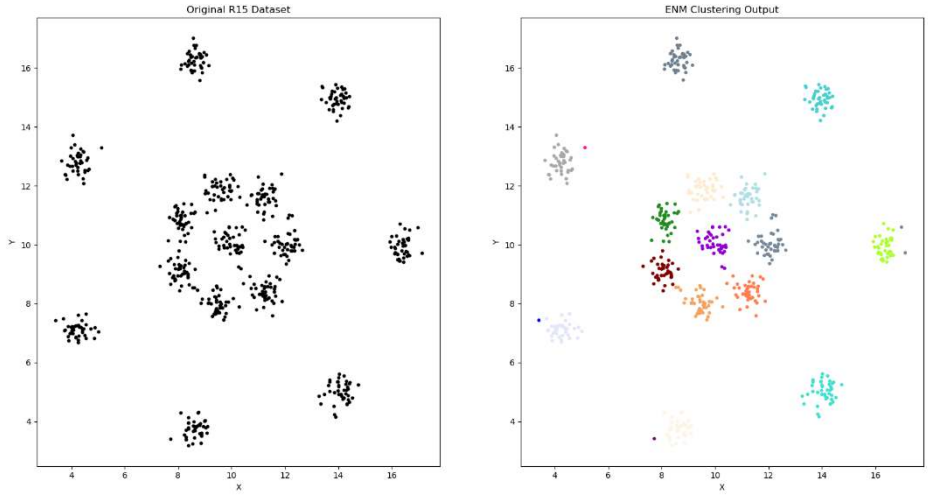

Figure 3.21. *Original R15 Dataset vs. ENM Clustering Output**

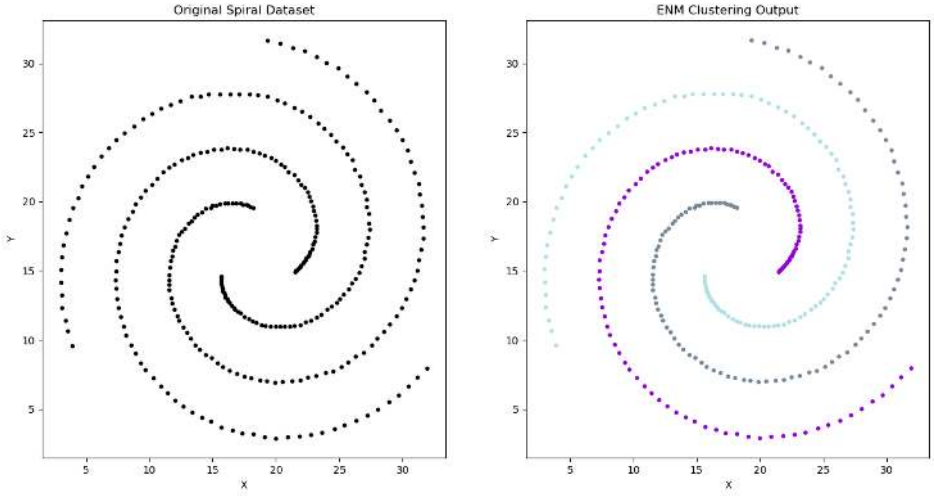

Figure 3.22. *Original spiral Dataset vs. ENM Clustering Output**

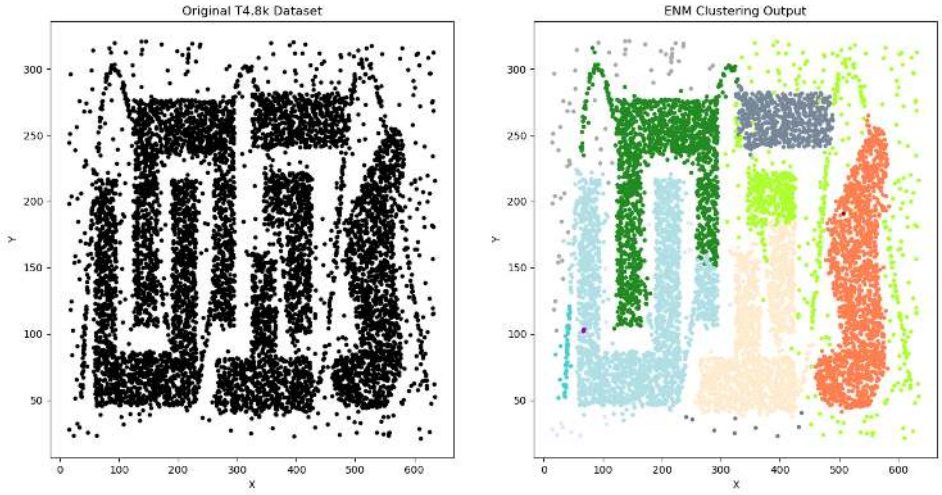

Figure 3.23. *Original t4.8k Dataset vs. ENM Clustering Output**

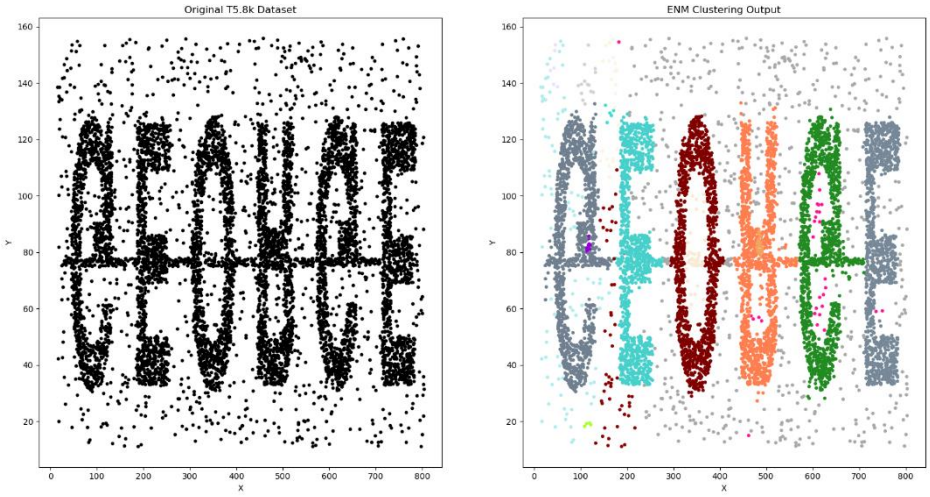

Figure 3.24. *Original t5.8k Dataset vs. ENM Clustering Output**

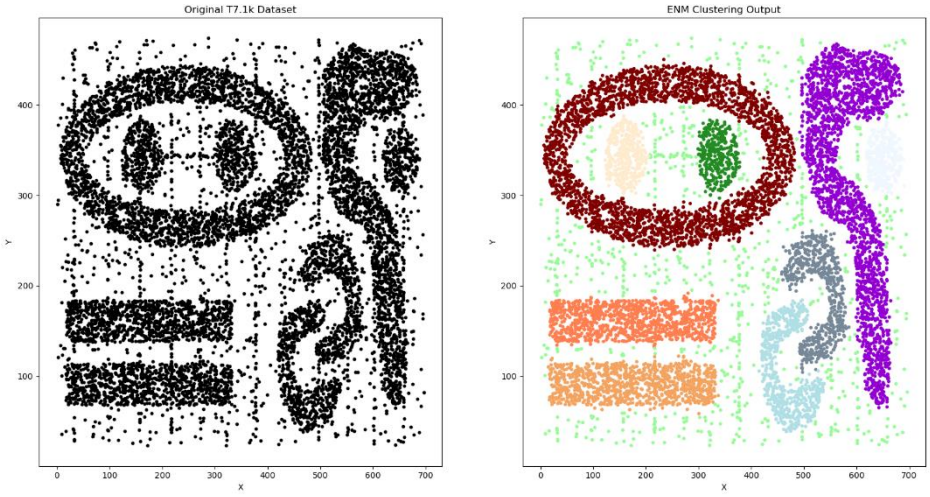

Figure 3.25. *Original t7.1k Dataset vs. ENM Clustering Output**

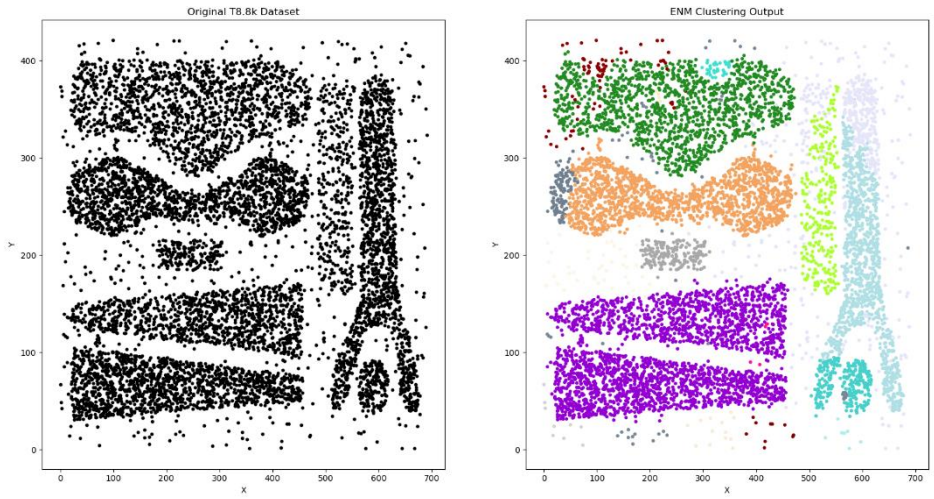

Figure 3.26. *Original t8.8k Dataset vs. ENM Clustering Output**

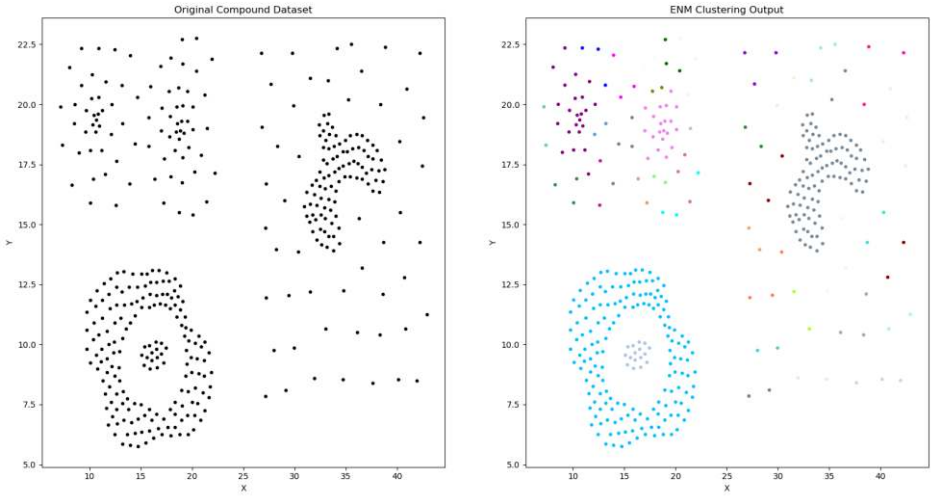

Figure 3.27. *Original Zahn's Compound Dataset vs. ENM Clustering Output**

Table 3.4. *Clustering Schema results for all the benchmark datasets*

Dataset	**Agg.**	**Atom**	**Cha.**	**DS31**	**Fla.**	**Jain**	**Path.**
# of Or. Clu.	7	2	2	31	2	2	3
# of Cor. Det. Clu.	7	2	2	27	2	2	3
Dataset	**R15**	**Spiral**	**t4.8k**	**t5.8k**	**t7.1k**	**t8.8k**	**Comp.**
# of Or. Clust.	15	3	6	6	8	8	6
# of Cor. Det. Clu.	15	3	6	6	8	7	6

*(Or.: Original, Clu.:Clusters, Cor.:Correctly, Det.:Detected)

- Validation Index Values:

When proposed methodology (ENM) is tested through several well-known external validation indices (homogeneity, completeness and v-measure indicators), we also achieved high quality results on all performance indicators Best results are given in bold characters. (see Table 3.5) Lower number of original cluster and overlapping clusters are thought to be main reasons of some lower scores on some datasets. ENM is also tested by some new internal evaluation indices and their result can be found also in Table 3.5.

Table 3.5. *External and Internal validation index values of the ENM experiments on each dataset*

Dataset	Homog.	Comp.	V-M.	ARI	FMS	MILWC	MILBC
Aggregation	0,96	0,92	0,94	0,94	0,95	0,039	0,058
Atom	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0,076	0,220
Chainlink	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0,058	0,090
DS31	0,93	0,92	0,93	0,89	0,91	**0,021**	**0,008**
Flame	0,50	0,72	0,59	0,69	0,86	0,039	0,031
Jain	**1,00**	**1,00**	**1,00**	**1,00**	**1,00**	0,033	0,031
Path Based	0,97	0,98	0,98	0,97	0,98	0,056	0,061
R15	0,98	0,99	0,99	0,97	0,98	0,028	0,015
Spiral	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0,009	0,031
t4.8k	0,89	0,84	0,88	0,90	0,90	0,142	0,050
t5.8k	0,92	0,85	0,88	0,88	0,90	0,028	0,016
t7.1k	0,95	0,93	0,94	0,92	0,91	0,144	0,060
t8.8k	0,82	0,81	0,81	0,71	0,77	0,062	0,057
Zahn Comp.	0,99	0,69	0,81	0,92	0,94	0,022	0,021

*(Homog.: Homogeneity, Comp.: Completeness, V-M.: V-Measure)

- CPU Execution Times:

As an empirical analysis to the time complexity of the proposed methodology, execution times of the proposed algorithm for all datasets are given in Figure 3.28. Just as the clustering schema quality and validation index results, proposed methodology presents low execution times and it makes proposed methodology to be convenient to the larger scale spatial datasets.

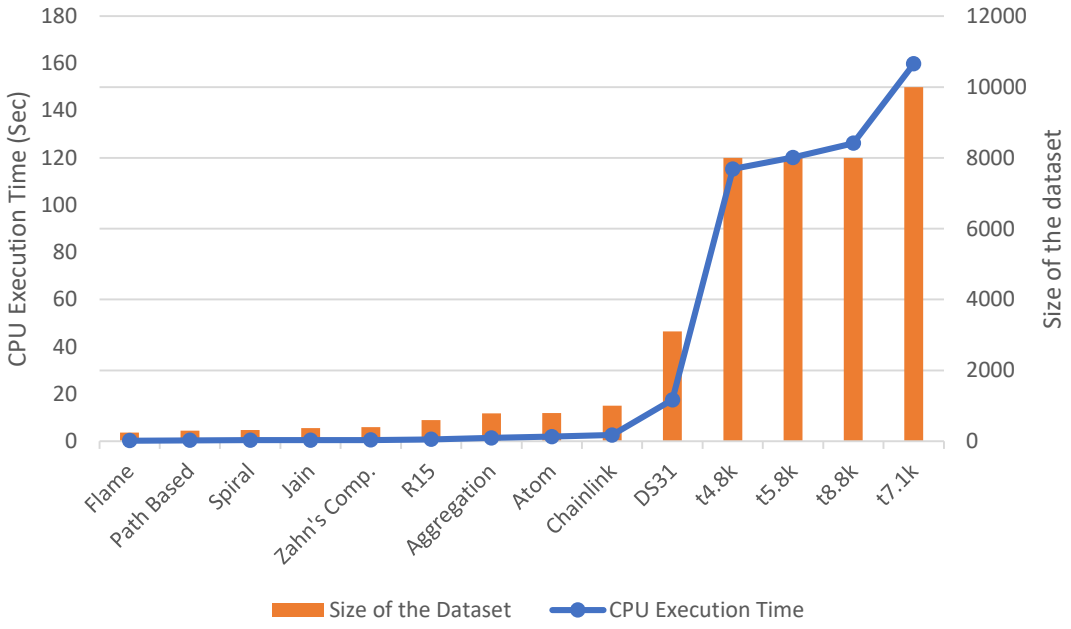

Figure 3.28. *Execution Times of the proposed algorithm*

- Comparative Analysis:

The proposed methodology (ENM) is compared with some well-known clustering methodologies like k-means, DBSCAN and Spectral Clustering. The comparison is made via # of correctly detected clusters by visual check, by some validation index scores: Homogeneity, Completeness, V-Measure, Adjusted Rand Index (ARI) and Fowlkes & Mallows Score (FMS) and by CPU times. Validation score comparisons can be found in Table 3.6 and Table 3.7 for each dataset and clustering algorithm. Moreover, in Figure 3.29, Figure 3.30 and Figure 3.31 comparison graphs are provided for some validation index results of the experimental analysis. Original # of classes vs. # of correctly detected clusters vs is given in Table 3.8. and Figure 3.32. Moreover, CPU Times are compared and the comparison can be found in Table 3.9 and Figure 3.33.

For K-Means and Spectral Clustering k is set as original number of clusters in the datasets. In DBSCAN epsilon is set as 3 and MinPts is set as 2 while preliminary number of original classes in the dataset is not required. As can be seen in from Table 3.6 and Table 3.7, Spectral Clustering is not applicable for large scale datasets because of similarity matrix requirement. Since t.4.8k, t5.8k, t7.1k and t8.8k contain at least 8000 data, spectral clustering applications are skipped for them. Sklearn's DBSCAN algorithm is preferred and it utilizes KD-TREE structure.

Table 3.6. *Performance of ENM and existing methodologies w. r. t. validation scores on the datasets*

Dataset	Algorithm	Homogeneity	Completeness	V-M.	ARI	FMS
Aggregation	K-Means	0,93	0,83	0,88	0,76	0,82
	DBSCAN	0,72	**1,00**	0,84	0,73	0,82
	Spect. Clust.	**0,99**	0,98	**0,99**	**0,99**	**0,99**
	ENM	0,96	0,92	0,94	0,94	0,95
Atom	K-Means	0,21	0,31	0,25	0,13	0,65
	DBSCAN	0,00	1,00	0,00	0,00	0,71
	Spect. Clust.	0,21	0,31	0,25	0,13	0,65
	ENM	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Chainlink	K-Means	0,07	0,07	0,07	0,09	0,55
	DBSCAN	0,00	1,00	0,00	0,00	0,71
	Spect. Clust.	0,08	0,08	0,08	0,10	0,55
	ENM	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
DS31	K-Means	**0,97**	0,94	**0,95**	**0,88**	0,88
	DBSCAN	0,00	**1,00**	0,00	0,00	0,20
	Spect. Clust.	**0,97**	0,93	**0,95**	0,87	0,88
	ENM	0,93	0,92	0,93	**0,89**	**0,91**
Flame	K-Means	0,36	0,46	0,40	0,44	0,72
	DBSCAN	0,00	**1,00**	0,00	0,00	0,69
	Spect. Clust.	0,06	**1,00**	0,10	0,03	0,70
	ENM	**0,50**	0,72	**0,59**	**0,69**	**0,86**
Jain	K-Means	0,40	0,36	0,38	0,32	0,70
	DBSCAN	0,00	**1,00**	0,00	0,00	0,78
	Spect. Clust.	0,92	**1,00**	0,96	0,99	**1,00**
	ENM	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Path Based	K-Means	0,48	0,58	0,52	0,43	0,64
	DBSCAN	0,00	**1,00**	0,00	0,00	0,57
	Spect. Clust.	0,71	0,77	0,74	0,65	0,77
	ENM	**0,97**	0,98	**0,98**	**0,97**	**0,98**
R15	K-Means	**0,98**	0,99	0,98	**0,98**	**0,98**
	DBSCAN	0,50	**1,00**	0,67	0,20	0,41
	Spect. Clust.	**0,98**	0,99	0,98	**0,98**	**0,98**
	ENM	**0,98**	0,99	**0,99**	0,97	**0,98**
Spiral	K-Means	0,02	0,02	0,02	0.00	0,32
	DBSCAN	0,93	1,00	0,96	0,98	0,98
	Spect. Clust.	0,93	1,00	0,96	0,98	0,98
	ENM	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
t4.8k	K-Means	0,60	0,62	0,61	0,52	0,60
	DBSCAN	**0,89**	0,27	0,42	0,03	0,10
	Spect. Clust.	N\A	N\A	N\A	N\A	N\A
	ENM	**0,89**	**0,84**	**0,88**	**0,90**	**0,90**

Table 3.7. *Performance of ENM and existing methodologies w. r. t. validation scores on the datasets (Cont'd)*

Dataset	Algorithm	Homogeneity	Completeness	V-M.	ARI	FMS
t5.8k	K-Means	0,79	**0,86**	0,82	0,78	0,81
	DBSCAN	**0,96**	0,80	0,87	**0,91**	**0,92**
	Spect. Clust.	N\A	N\A	N\A	N\A	N\A
	ENM	0,92	0,85	**0,88**	0,88	0,90
t7.1k	K-Means	0,59	0,58	0,59	0,39	0,48
	DBSCAN	0,79	0,26	0,39	0,00	0,09
	Spect. Clust.	N\A	N\A	N\A	N\A	N\A
	ENM	**0,95**	**0,93**	**0,94**	**0,92**	**0,91**
t8.8k	K-Means	0,57	0,54	0,55	0,35	0,44
	DBSCAN	0,71	0,24	0,36	0,00	0,10
	Spect. Clust.	N\A	N\A	N\A	N\A	N\A
	ENM	**0,82**	**0,81**	**0,81**	**0,71**	**0,77**
Zahn's C.	K-Means	0,76	0,70	0,73	0,54	0,64
	DBSCAN	0,41	**1,00**	0,58	0,43	0,67
	Spect. Clust.	0,78	0,74	0,76	0,52	0,63
	ENM	**0,99**	0,69	**0,81**	**0,92**	**0,94**

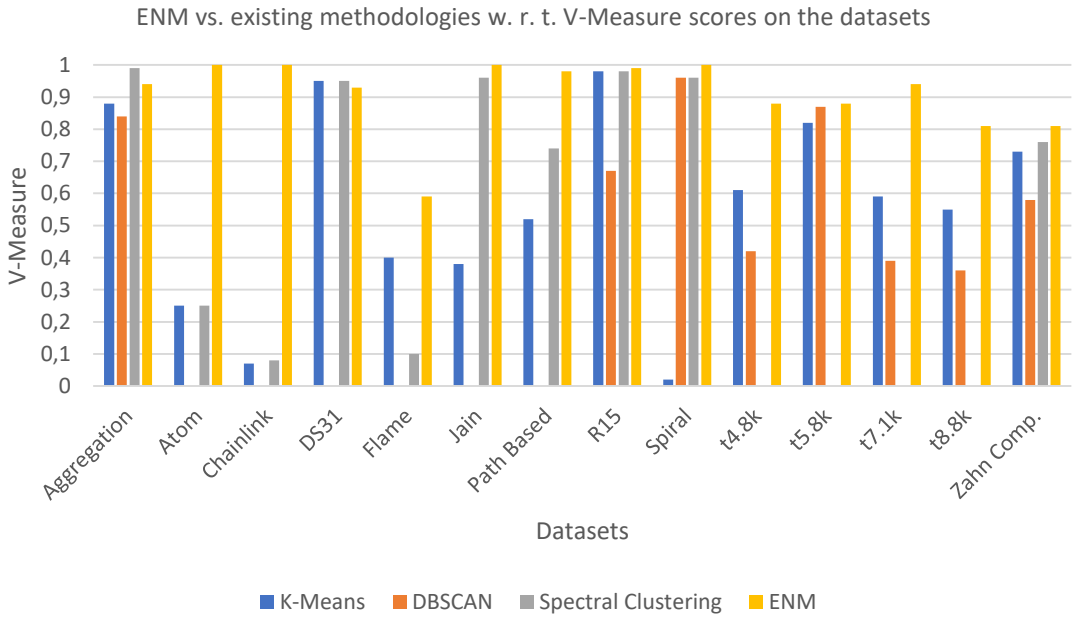

Figure 3.29. *V-Measure scores for ENM and other methodologies*

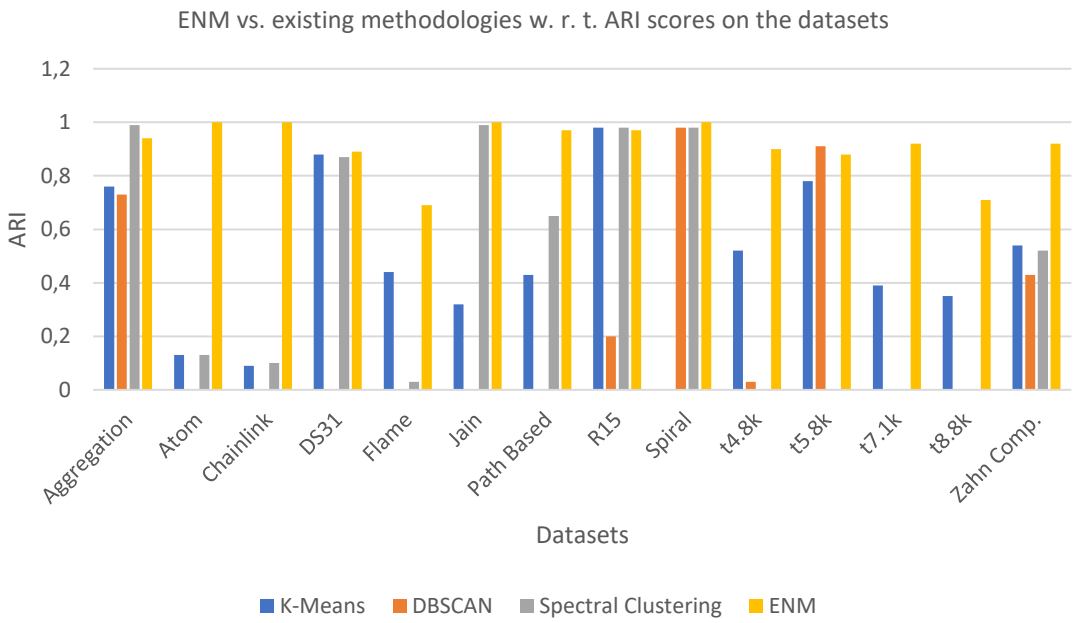

Figure 3.30. *ARI scores for ENM and other methodologies*

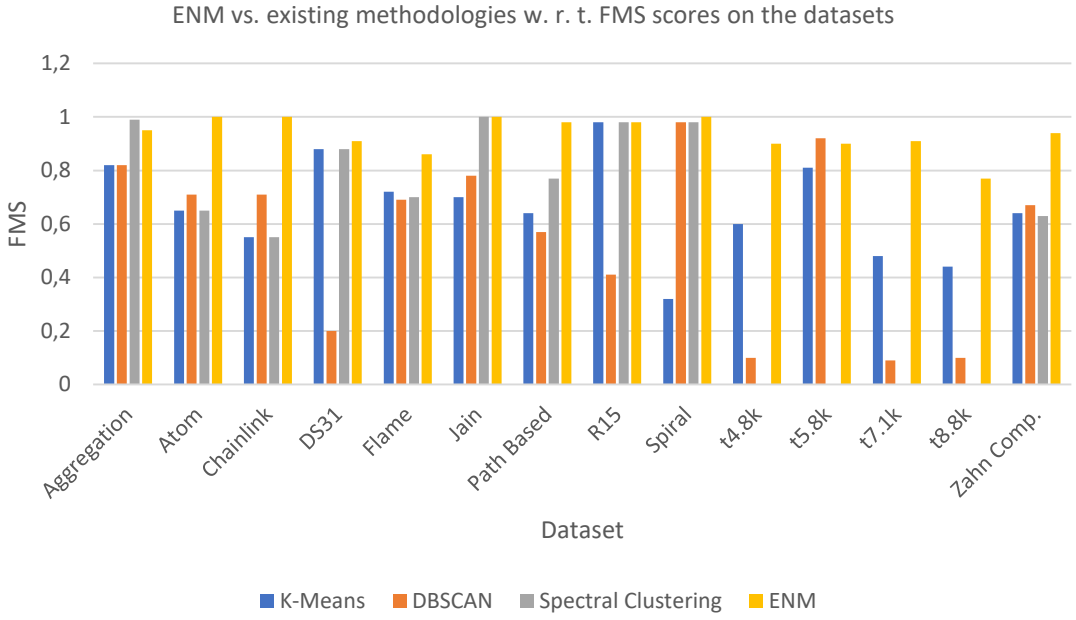

Figure 3.31. *FMS scores for ENM and other methodologies*

Table 3.8. *Original # of clusters vs. # of correctly detected clusters*

Dataset	Original # of clusters	# of correctly detected clusters			
		K-Means	DBSCAN	Spec. Clust.*	ENM
Aggregation	7	5	2	7	7
Atom	2	0	0	0	2
Chainlink	2	0	0	0	2
DS31	31	31	0	31	27
Flame	2	0	0	0	2
Jain	2	0	0	2	2
Path Based	3	0	0	1	3
R15	15	15	7	15	15
Spiral	3	0	3	3	3
t4.8k	6	1	0	N\A	6
t5.8k	6	6	0	N\A	6
t7.1k	8	0	0	N\A	8
t8.8k	8	0	0	N\A	7
Zahn Comp.	6	2	0	2	6

*(Spec. Clust.: Spectral Clustering)

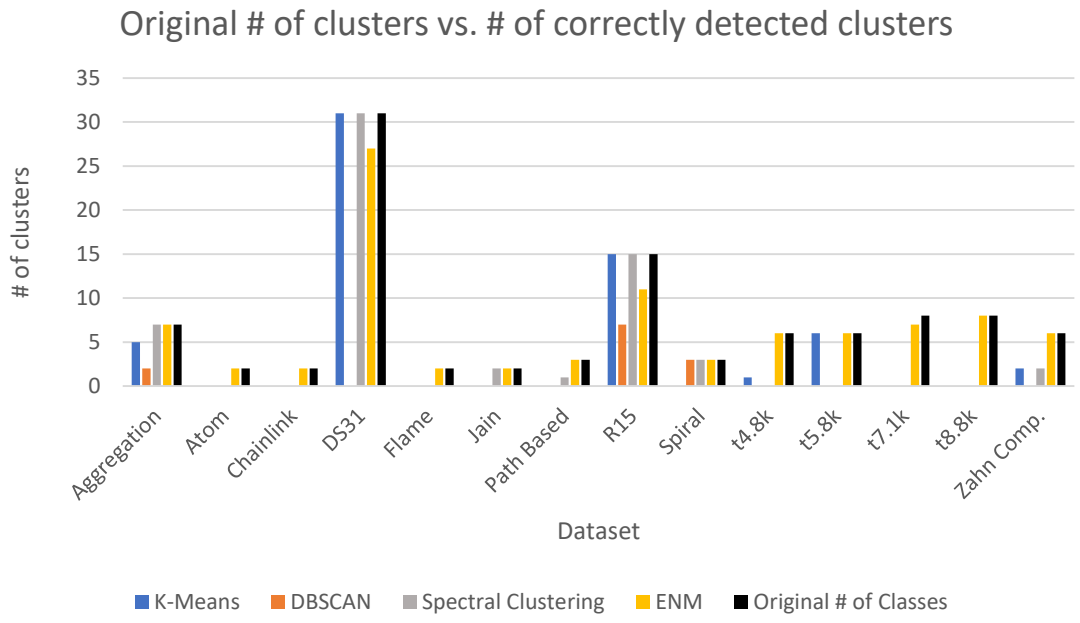

Figure 3.32. *Cluster detection results for ENM and other methodologies*

Table 3.9. *CPU execution time (seconds) comparison*

Dataset	Size	K-Means	DBSCAN	Spectral Clustering	ENM
Aggregation	788	0,050	0,009	0,406	1,37
Atom	800	0,037	0,004	0,107	1,95
Chainlink	1000	0,306	0,049	3,417	2,59
DS31	3100	0,020	0,002	0,054	17,55
Flame	240	0,029	0,006	1,695	0,25
Jain	373	0,023	0,004	0,148	0,45
Path Based	300	0,067	0,007	0,180	0,38
R15	600	0,047	0,002	0,208	0,75
Spiral	312	0,210	0,059	-	0,42
t4.8k	8000	0,123	0,067	-	115,3
t5.8k	8000	0,327	0,071	-	120.15
t7.1k	10000	0,251	0,062	-	159,81
t8.8k	8000	0,050	0,009	-	126,18
Zahn Comp.	399	0,037	0,004	0,107	0,53

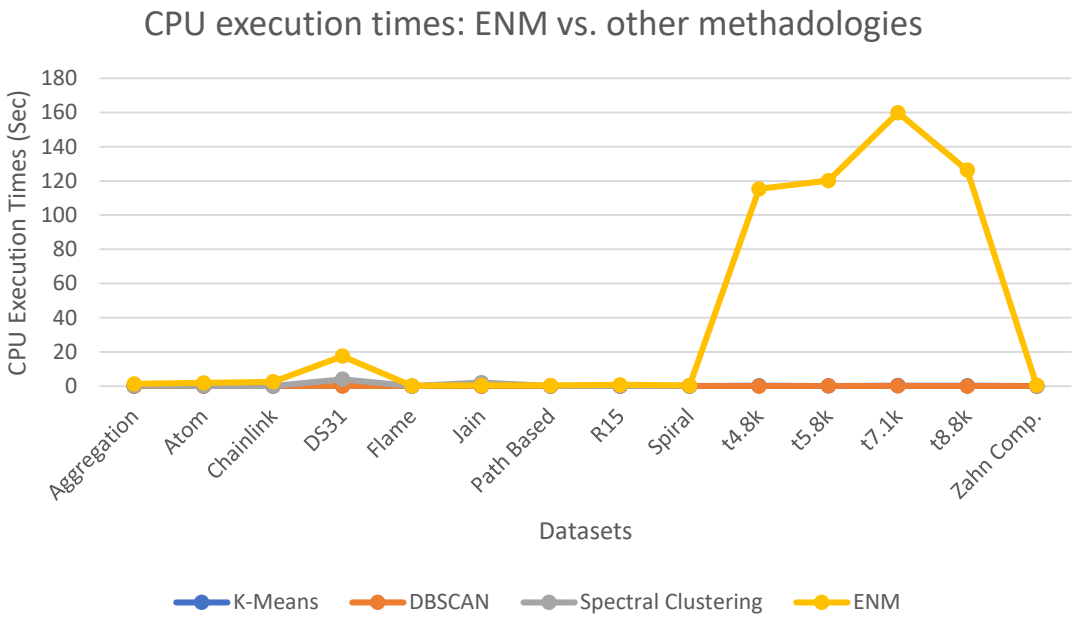

Figure 3.33. *CPU times for ENM and other methodologies*

Main finding from the comparison tables is that the proposed methodology outperforms K-Means, DBSCAN and Spectral Clustering according to all validation index scores and clustering schema quality. As for the execution times, it is observed that the proposed methodology presents higher computational times contrarily to other methodologies but still it is about seconds on most of the datasets or reasonable on large scale datasets. Thus it can be disserted that the proposed methodology, the ENM, is also

convenient for larger scale datasets due to reasonable computational burden and less execution time requirement while considering its high quality clustering schema output. Moreover, computational times may be decreased by means of more efficient programming as a future research issue.

4. ENM APPLICATION: GENERATING SEISMIC ZONES MAP

In this part, a real life application of the proposed methodology (ENM) is given. Since the ENM is developed for arbitrarily geometric shaped cluster detection, planer datasets are convenient as the real life case such as geospatial datasets. In that context, historical earthquake data provide planer datasets which have 2d coordinates and magnitude of the earthquakes and eventually it provides arbitrary shaped seismic zones.

Some earthquake zoning via several clustering methodologies exist in the literature for some counties / locations. These papers are summarized in Table 4.1.

Table 4.1. *Earthquake zoning literature via clustering*

Paper	Country / Location	Clustering Method
(Ansari, Noorzad, & Zaf, 2009)	Iran	Fuzzy C-Means &
(Konstantaras, 2020)	Hellenic and Aegean	Agglomerative Clust.
(Georgoulas, Konstantaras, Katsifarakis, Stylios, & Maravelakis, 2013)	Hellenic and Aegean	DBSCAN & Agglomerative Clust.
(Morales-Esteban, Martínez-Álvarez, Scitovski, & Scitovski, 2014)	Croatia and Iberian	K- Partition
(Gvishiani, Dobrovolsky, Agayan, & Dzeboev, 2013)	Caucasus	Discrete Perfect Sets
(Di Giuseppe, Troiano, Troise, & De Natale, 2014)	Italy	K-Means
(Scitovski, 2018)	Croatia	Rough DBSCAN
(Karri, Ansari, & Pathak, 2018)	India	DBSCAN

*Clust.: Clustering

4.1. The Earthquake Datasets

The earthquake datasets are generated and retrieved from "B.Ü. Kandilli Rasathanesi BDTİM Deprem Sorgulama Sistemi" (http-8). The datasets include historical earthquake data of TURKEY and some nearby places.

For scientific coherence 4 different earthquake datasets are generated from (http-8) for specific time ranges like 50 years or 100 years and magnitude ranges from 3.0-9.0 or 4.0-9.0. All the generation properties of the raw datasets are given in Table 4.2. Depth range is chosen as 0-500 km as default, longitude range is chosen as 26°-45° and latitude ranges is chosen as 35°-42° for the true geographical position of Turkey. Each dataset includes some number of historical earthquake data, given in size column.

Table 4.2. *The earthquake datasets generation and their size*

Dataset	Date	Magnitude	# of Earthquakes
Earthquake1	1972-2023	4.0 – 8.0	4965
Earthquake2	1972-2023	3.5 – 8.0	16364
Earthquake3	1922-2023	4.0 – 8.0	7001
Earthquake4	1922-2023	3.5 – 8.0	18335

The properties of each dataset can be summarized as in Table 4.3 and Table 4.4. Each raw dataset is filtered in accordance with the dataset description in Table 4.4.

Table 4.3. *The earthquake datasets earthquake distribution for each magnitude range*

Dataset	M<4	4<M<5	5<M<6	6<M<7	M>7
Earthquake1	0	4351	403	27	4
Earthquake2	11164	4760	409	27	4
Earthquake3	0	5909	997	79	16
Earthquake4	11334	5909	997	79	16

Table 4.4. *The Feature Descriptions of the Earthquake Datasets.*

Feature	Unit	Data Type
Epicenter Latitude	Degree	Float
Epicenter Longitude	Degree	Float
Magnitude	Richter	Float
Depth	Km	Float

The general classification, occurrence frequency and the possible effects of the earthquakes are given in Table 4.5. Since the earthquakes which has less than 3.0 xM, are discarded from the datasets in the generation process. Because, they are geologically assumed as infinitesimal for seismic zone detection. (xM refers to the highest values of magnitudes among all the measurements.)

Table 4.5. *The earthquake classification and earthquake properties*

Magnitude	Effects	Occurrence	Description
0.0 – 2.0 xM	Not felt by people.	Daily	Small
2.0 – 3.0 xM	Little felt by people.	Daily	Small
3.0 – 4.0 xM	Ceiling lights swing.	Monthly	Small
4.0 – 5.0 xM	Walls crack.	Monthly	Moderate
5.0 – 6.0 xM	Furniture moves.	Yearly	Moderate
6.0 – 7.0 xM	Some building collapses.	Yearly	Great
7.0 – 8.0 xM	Many building collapses.	Yearly	Great
8.0 – 9.0 xM	Catastrophic destructions.	Rare	Great

4.2. Seismic Zones Detection by Using ENM

Before implementing ENM on datasets, each dataset is visualized on 2d. Each point refers to a single earthquake. In each visualization, exponent of magnitude is assumed the size of the points on the maps. Similarly, each point (earthquake) is colored in accordance with its size. As the color gets darker, the magnitude is higher. The earthquakes which are lower than 4.0 xM is depicted as yellow, the earthquakes which are in between 4.0 and 5.0 xM is depicted as orange, the earthquakes which are in between 4.0 and 5.0 xM is depicted as orange, the earthquakes which are in between 5.0 and 6.0 xM is depicted as red and the earthquakes which are higher than 6.0 xM is depicted as brown. From Figure 4.1 up to Figure 4.4. each visualization is given.

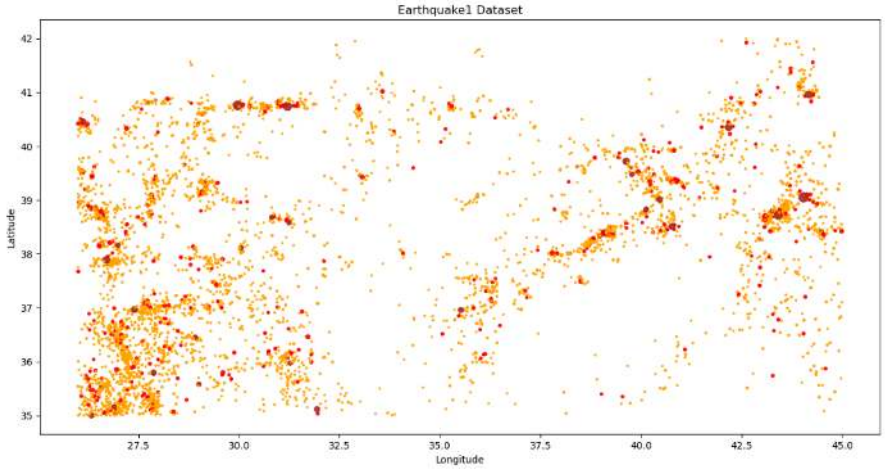

Figure 4.1. *Visual of Earthquake1 Dataset*

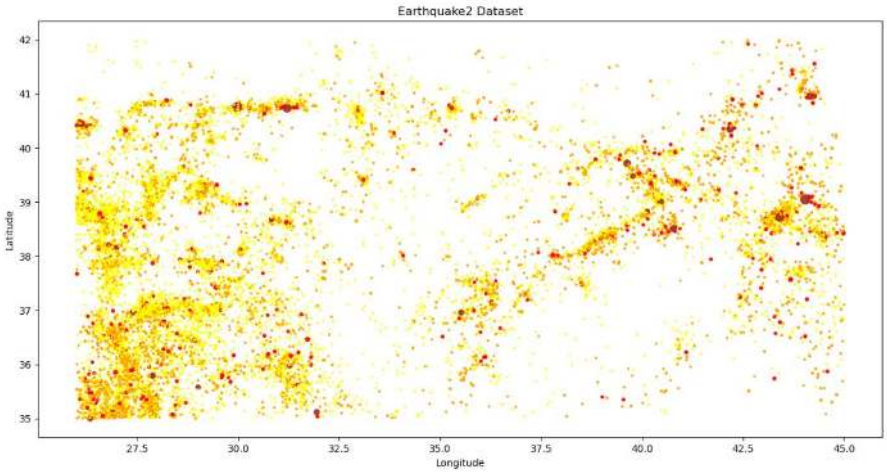

Figure 4.2. *Visual of Earthquake2 Dataset*

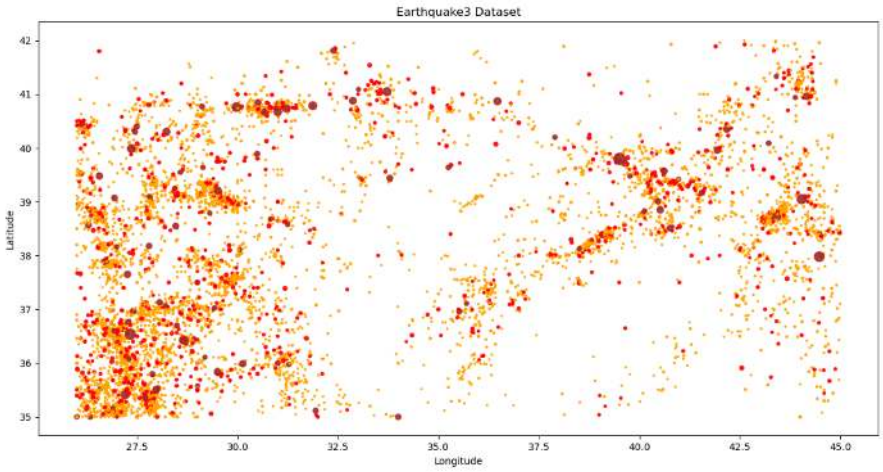

Figure 4.3. *Visual of Earthquake3 Dataset*

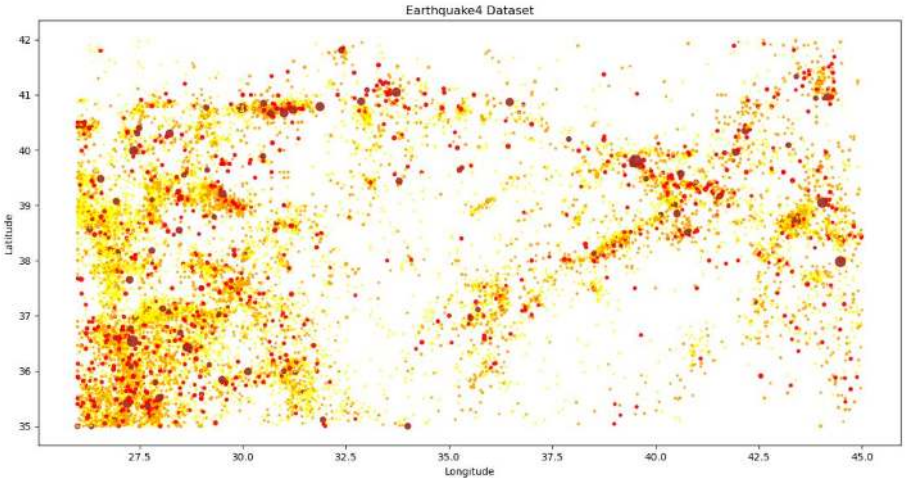

Figure 4.4. *Visual of Earthquake4 Dataset*

The proposed algorithm (ENM) is then implemented on 4 different earthquake datasets. In order to unearth the seismic zones of Turkey. Since the proposed algorithm (ENM) has no requirement of preliminary number of clusters in the dataset, number of seismic zones are determined by the ENM automatically.

4.3. Result of the Experimentation and Discussion

After implementing ENM on 4 different datasets by setting γ as 0.9, clustering schemas are obtained. In the clustering output of ENM, each earthquake belongs to a single cluster (seismic zone) and labelled by some integer numbers.

All the clustering outputs are visualized from Figure 4.5 up to Figure 4.8. Clusters are represented by distinct color in all clustering schema. Since external validation is

impossible because of unlabeled datasets, clustering output is validated by visual checks and compared with the real seismic zone map of Turkey in Figure 4.9 and earthquake risk map which is given in Figure 4.10. Visual checks and comparisons shows that, ENM has detected similar seismic zones with the original map and Earthquake1 dataset has higher similarity with the real seismic zone map.

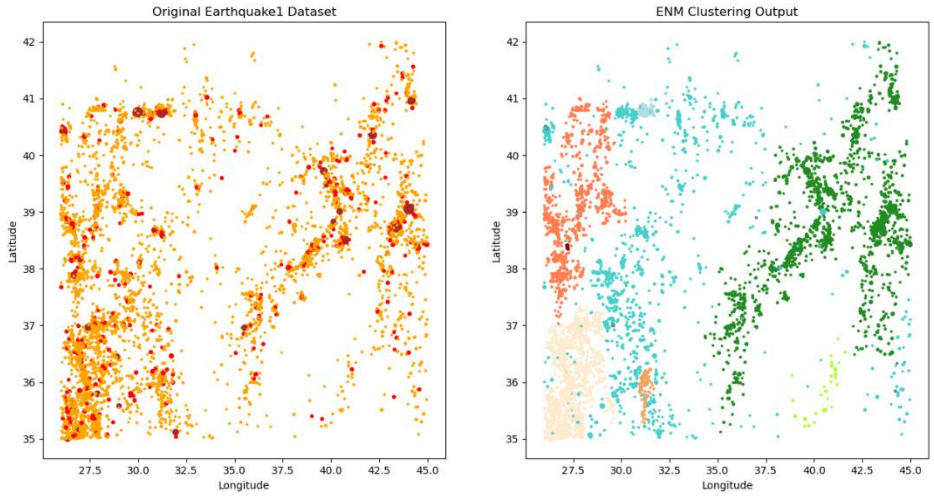

Figure 4.5. *ENM clustering output for Earthquake1 Dataset*

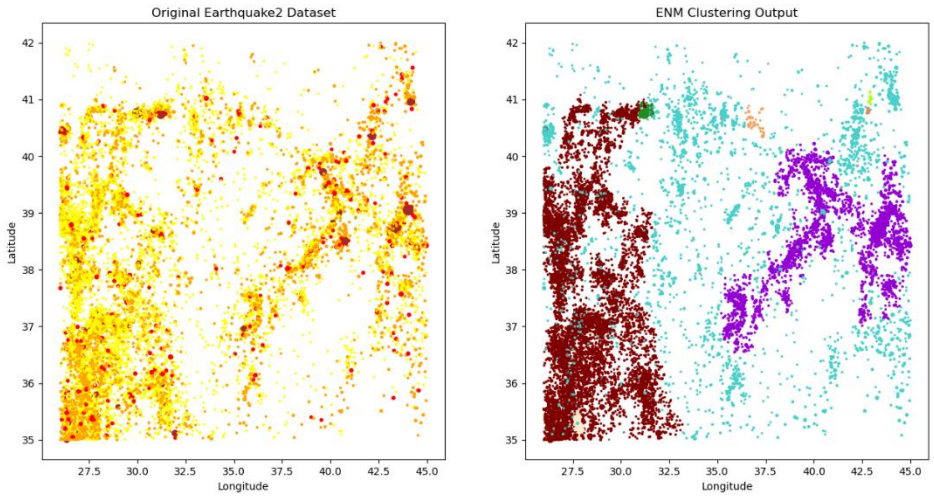

Figure 4.6. *ENM clustering output for Earthquake2 Dataset*

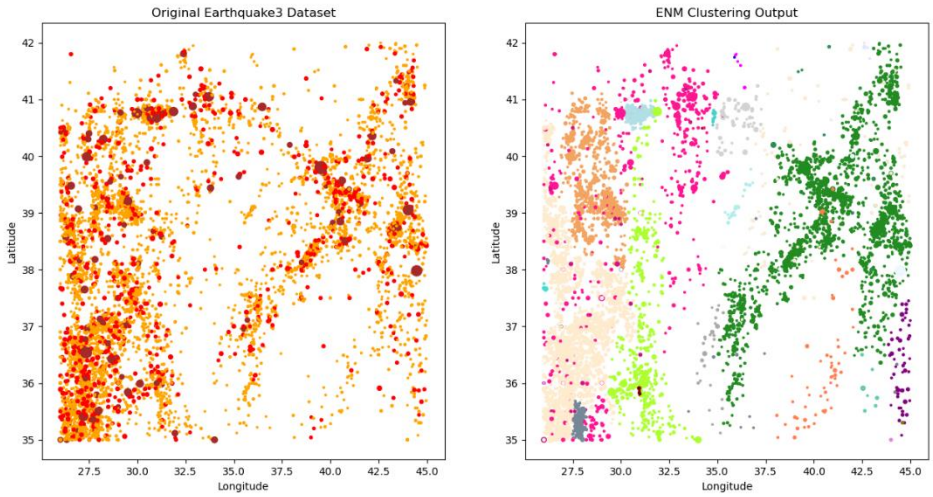

Figure 4.7. *ENM clustering output for Earthquake3 Dataset*

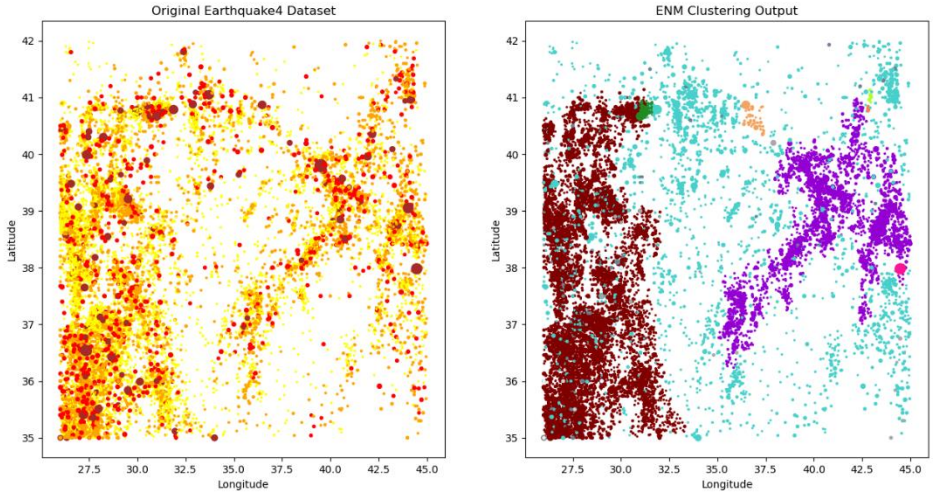

Figure 4.8. *ENM clustering output for Earthquake4 Dataset*

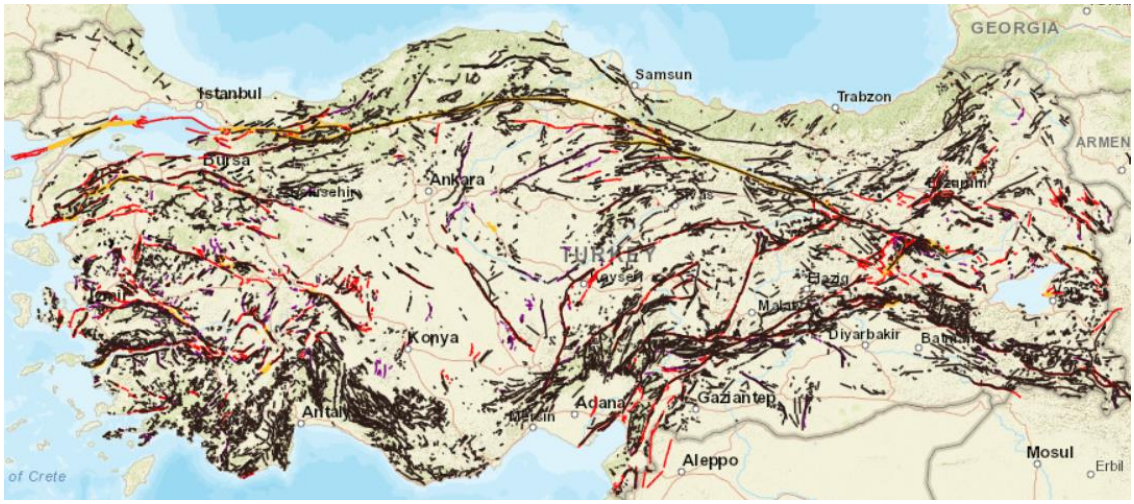

Figure 4.9. *Seismic zones map of Turkey (http-9a)*

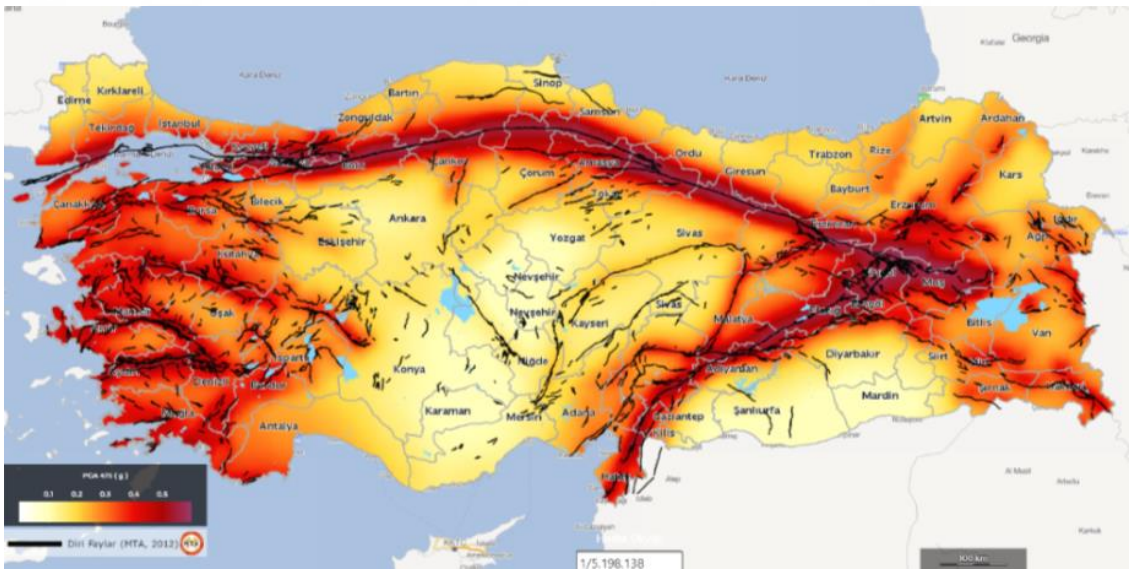

Figure 4.10. *Earthquake Risk map of Turkey. (http-9b)*

Since all the dataset include unlabeled data, it is not possible to use external validation indices. Instead of external evaluation, two internal validation indices are used: MILWC and MILBC. They are developed for validation of the clustering outputs which have arbitrary geometric shaped clusters.

Table 4.6. *Internal validation index results of ENM output.*

Dataset	MILWC	MILBC	# of Clusters (Seismic Zones)
Earthquake1	0,002	0,002	6
Earthquake2	0,003	0,002	4
Earthquake3	0,006	0,005	10
Earthquake4	0,004	0,005	5

In Table 4.6, validation index results are given. It shows lower MILWC and MILBC values and it means clusters are compact in every clustering output. But Earthquake1 dataset presents the best clustering validation index results just as in visual checks and visual comparison.

4.4. Comparison with the Existing Methodologies

Seismic zone detection problem is solved via some well-known existing methodologies for the comparison with ENM: K-Means, DBSCAN and Spectral Clustering. K-Means are centroid based algorithm, DBSCAN is density based algorithm and spectral clustering is based on spectral graph theory.

When existing methodologies are preferred, clustering outputs are given from Figure 4.11 up to Figure 4.19. For K-Means and Spectral Clustering, original number of clusters is set as 3 while epsilon and MinPts are set as 1 and 200 respectively for DBSCAN for all datasets. But for Earthquake2, Earthquake3 and Earthquake4 datasets, spectral clustering is not applicable due to the fact that the size of the datasets is too large to construct similarity matrix.

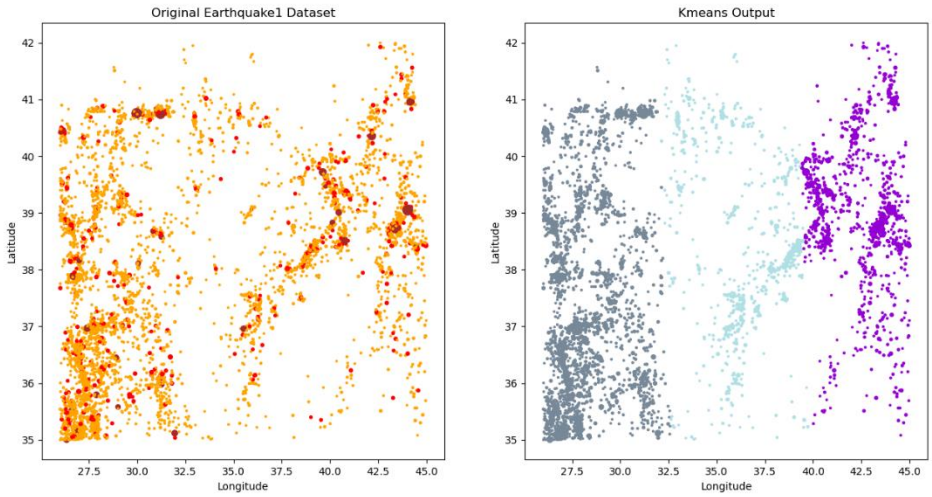

Figure 4.11. *K-Means clustering output for Earthquake1 Dataset*

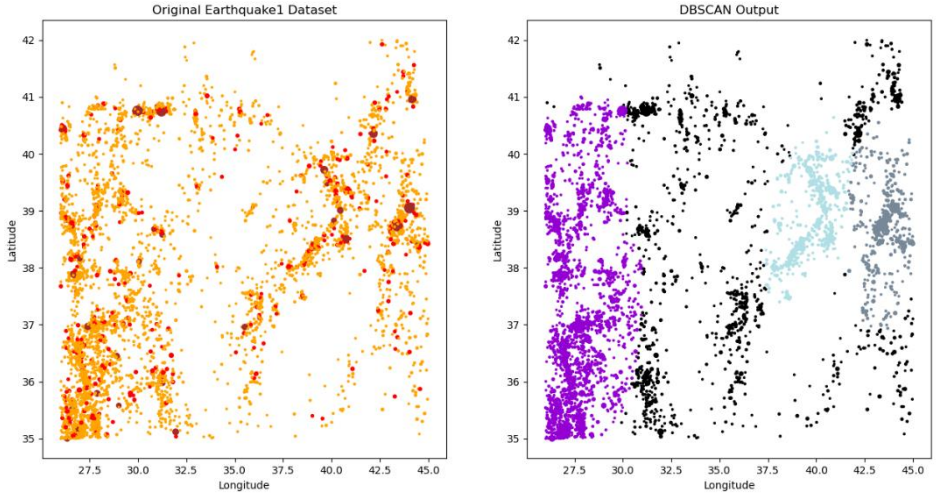

Figure 4.12. *DBSCAN clustering output for Earthquake1 Dataset*

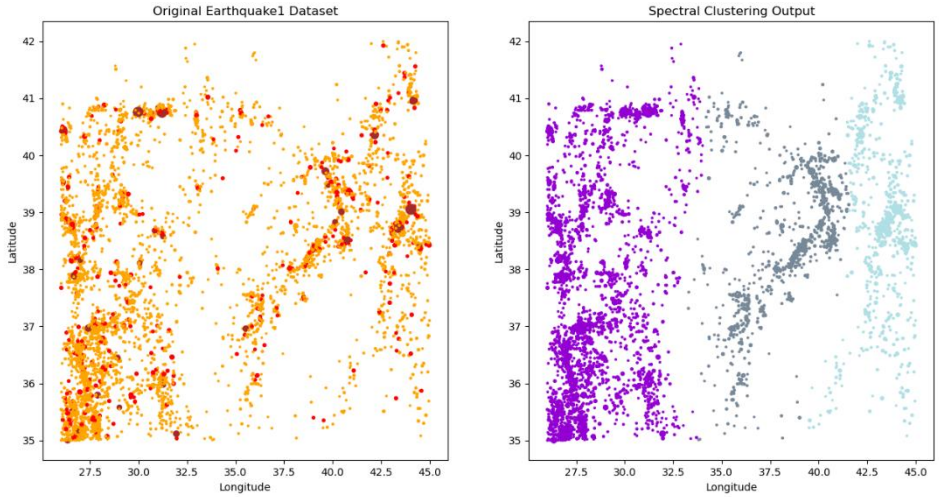

Figure 4.13. *Spectral clustering output for Earthquake1 Dataset*

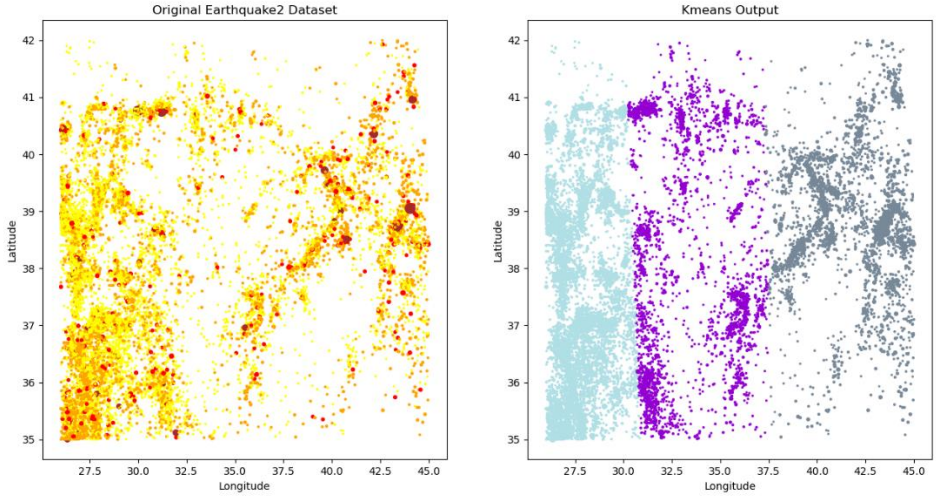

Figure 4.14. *K-Means clustering output for Earthquake2 Dataset*

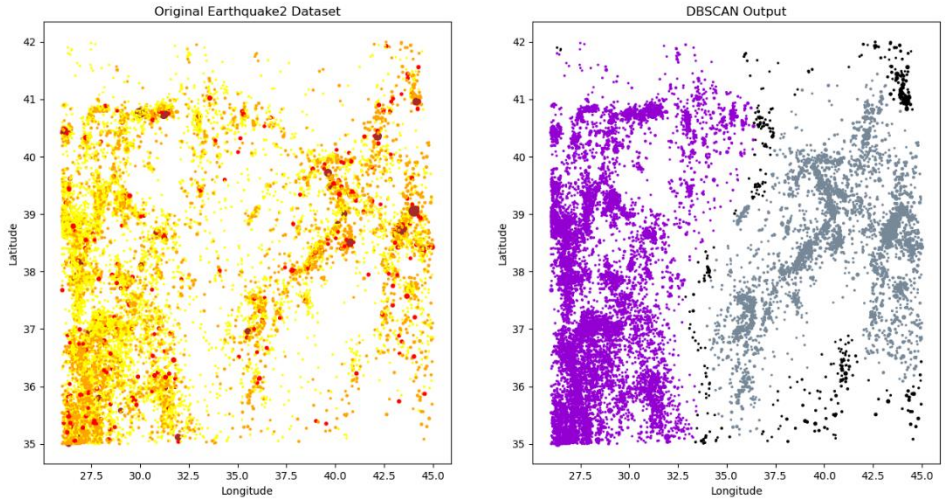

Figure 4.15. *DBSCAN clustering output for Earthquake2 Dataset*

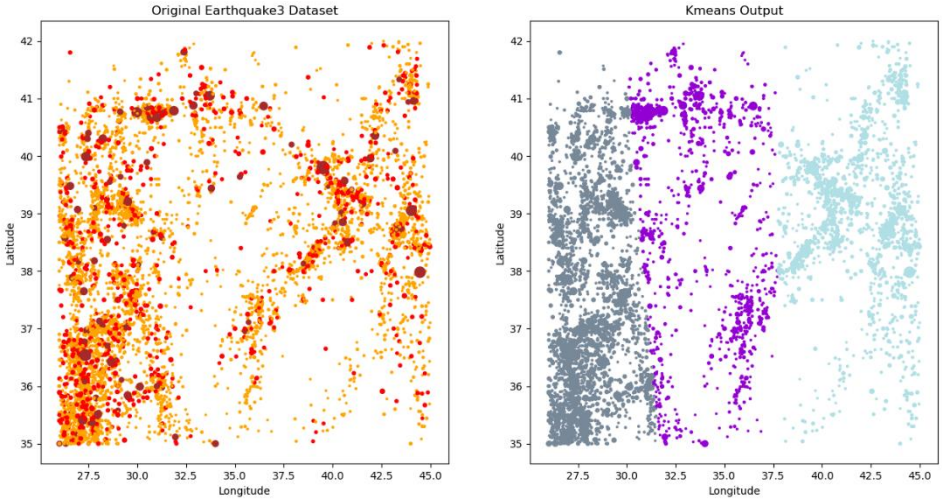

Figure 4.16. *K-Means clustering output for Earthquake3 Dataset*

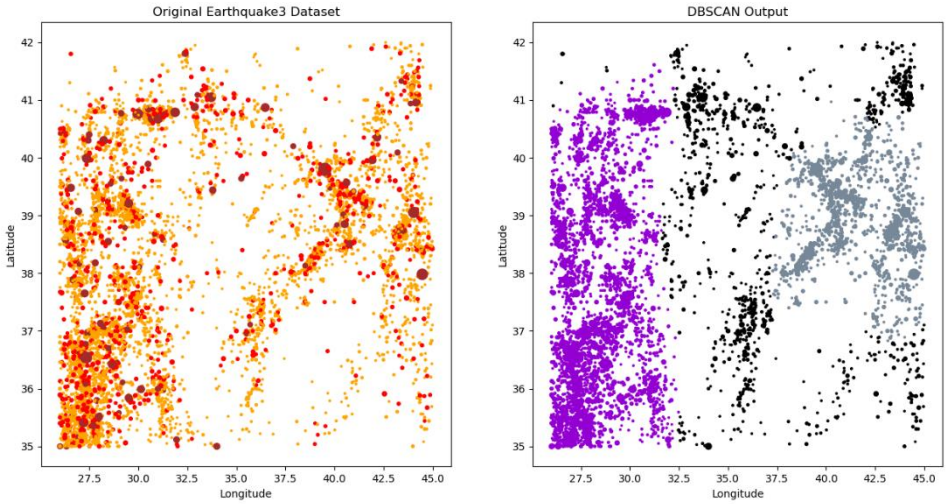

Figure 4.17. *DBSCAN clustering output for Earthquake3 Dataset*

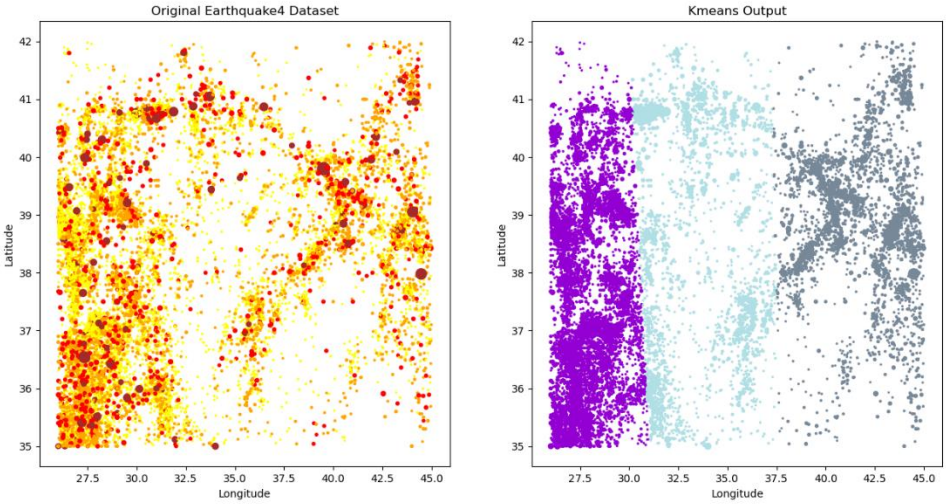

Figure 4.18. *K-Means clustering output for Earthquake4 Dataset*

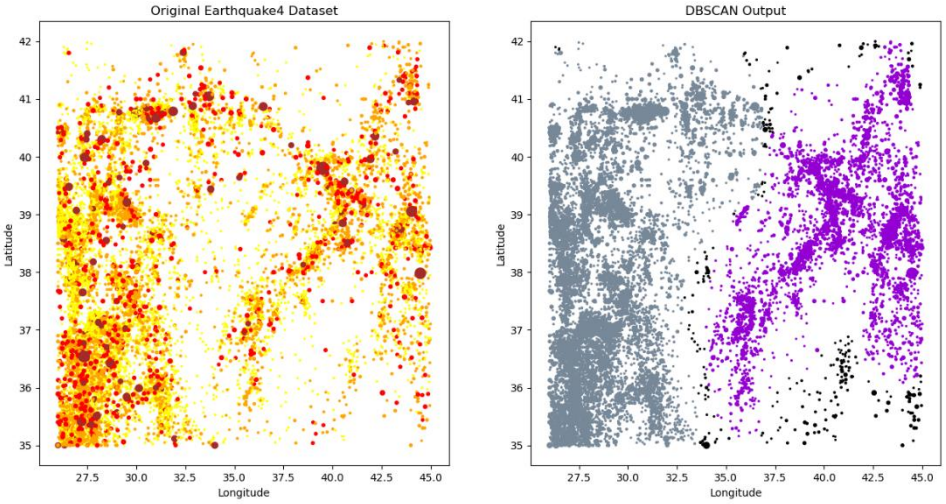

Figure 4.19. *DBSCAN clustering output for Earthquake4 Dataset*

The visual comparison points out that ENM outperforms all other existing methodologies on every dataset w.r.t. clustering schema quality and the number of clusters (seismic zones) that are detected. Furthermore, second best quality among the clustering schemas are DBSCAN outputs in general relatively to others.

5. THE PROPOSED METHODOLOGY FOR CLUSTERING ON MULTI DIMENSIONAL FEATURE SPACE

The ability of handling high dimensional feature space is one of the challenging issues in the state of art clustering methodologies, since all of the machine learning algorithms suffers from the same concept: curse of dimensionality. Most of the real-life clustering datasets contains high dimensional data such as in the marketing, text mining, bioinformatics and so on. Therefore, ability of handling multidimensional feature space effectively plays key role for the robustness of a clustering algorithm and/or approach.

As mentioned in the introduction part, curse of dimensionality could be one of the obstacle for a clustering algorithm on the producing a valid, meaningful and reliable output. The phenomena "curse of dimensionality" is first introduced by Richard E. Bellman in 1957 and refers to so fast increment rate of space that makes data so sparse in datasets as the number of dimension of data increases and the phenomenon can be come across in the fields of statistics, machine learning and databases. The sparsity in data prevents obtaining statistically significant clustering results since every data represent itself rather than to be a member of a data heap in that case. But clustering is based on detecting similar & dissimilar data elements to produce data cluster. Consequently, in high dimensional datasets, statistical significance of the clustering output decreases because of exponential convergence of similarity / dissimilarity measures in order to support the output. (Bellman & Rand Corporation, 1957)

As the number of dimension grows in datasets, distance to farthest and nearest point converge and distance concept fail in recognizing similarities. Beyond that issue, multiple dimensions are hard to imagine and impossible to be visualized. To tackle this issue, researches come up different approaches: Subspace clustering, projected clustering, hybrid approaches and correlation clustering. (Kriegel, Kröger, & Zimek, 2009)

In subspace clustering, main idea is that analyzing dataset in one dimensional subspaces and identify clusters according to that analysis. CLIQUE (Agrawal, Gehrke, Gunopulos, & Raghavan, 1998) and SUBCLUE (Kriegel, Kröger, & Zimek, 2009) are typical examples of subspace clustering. On the other hand, in projected clustering approaches, typical clustering algorithms are used by taking special distance function into considerations such as Mahalanobis distance, but with an additional complexity burden. There are also hybrid approaches in which the main idea falls in between for subspace and projected clustering. Finally, adaptive correlation between attributes are considered

in correlation clustering instead of considering a global correlation value between attributes. Most of them suffers from input parameter estimation or high complexity.

In literature, a general consensus can be seen on the clustering process in unsupervised machine learning: If the data is high dimensional, as a preprocessing step, first the number of dimensions are reduced by using dimension reduce reduction techniques, then cluster data. However, such a kind of cascaded way may cause information loss and reduce the validity of the clustering output and may yield low quality clustering schema. Moreover, most of the dimension reduction methodologies are inconvenient for the real life datasets in terms of interpretability of the output, since they are generating new dimensions, rather than using original dimensions.

The applicability of the isotropic position is studied to overcome clustering difficulties in multidimensional space by constructing clusters in a certain subspace.

5.1. Isotropic Position

Isotropic position is a statistical concept which is introduced in (Rudelson, 1999), when vectors are fitted to a probability distribution, if covariance matrix of this distribution is identity matrix then these vectors is said to be in isotropic position. In other words, isotropic position can be used to describe uniformity of vectors around the centroid data. Isotropic measure is defined as how close is covariance matrix to identity matrix. The bivariate gauss distribution is an example of isotropic distribution. As illustrated in Figure 5.1, dataset includes vectors which are in isotropic position since the covariance matrix is close to identity matrix. It means, data is spherically distributed around the centroid in the current dimensions.

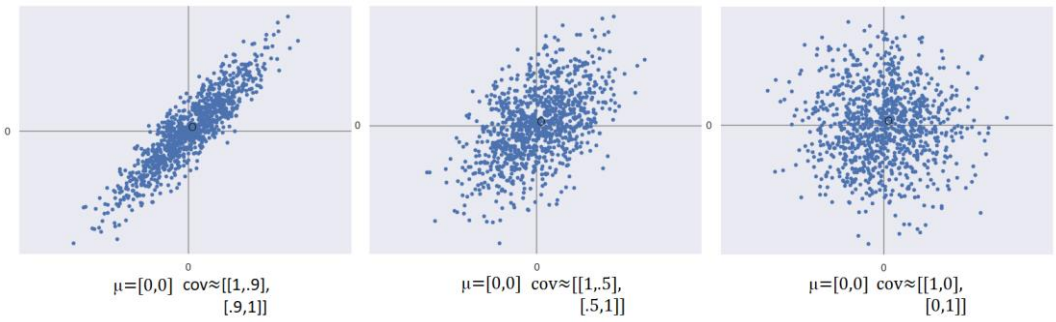

Figure 5.1. *Some example of 2d vector sets in different levels of isotropic position with their mean and their relevant covariance matrices*

5.2. The Simultaneous Feature Selective Clustering (SFSC) Algorithm

In this thesis; a hybrid approach is proposed for high dimensional data clustering which is mainly based on subspace clustering and projected clustering using the term,

isotropic position. More precisely, proposed algorithm deals with clustering, simultaneously reducing number of dimensions by means of isotropic positon concept. For any dimension, if data is invariant, this dimension considered as a redundant feature in the proposed methodology. Additionally, since the proposed methodology will be executed in neighborhood merging manner, low complexity burden will be incurred and cluster output visualization will be possible by dynamic feature selection for each cluster.

In the proposed algorithm (SFSC), data is assumed as invariant by some features by means of isotropic position by inspiring from (Brubaker & Vempala, 2018) which is isotropic PCA and determination of invariant subspaces (affine). If data is invariant (or lower levels of invariance) by some features, these features can be neglected, as a result only principal features will be considered. As an extension of (Brubaker & Vempala, 2018), clustering will be defined as union of highest isotropic neighborhood by dynamic principal features.

The formula of the frobenius norm for a covariance matrix($\|COV\|_F$) , which is also known as the Euclidean norm, can be found in Equation 5.1. a_{ij} refers to the component of the matrix in row i and column j. In SFSC, isotropic position level is based on how close the covariance matrix of a vector set (Cov(Set)) to identity matrix (I) and then it is used for determination of principal attributes for the dense neighborhoods. Closeness to identity matrix is assumed as the gap between frobenius norm of the covariance matrix for a set of vectors (COV(Set)) and frobenius norm of identity matrix (I). (see in Equation 5.2)

$$\|COV\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|a_{ij}\right|^2} \tag{5.1}$$

$$Isolevel = \|COV(Set)\|_F - \|I\|_F \tag{5.2}$$

5.3. Features of SFSC

The proposed algorithm(SFSC) has the following attributes:

- The SFSC doesn't need the number of clusters knowledge that is defined a priori by the user.
- The SFSC is convenient on multidimensional datasets.
- The algorithm is based on statistical concepts: Entropy and Isotropic Position.
- The SFSC, simultaneously makes feature selection and clustering.

- The SFSC makes feature selection dynamically. It means all the clusters may have different set of principal features which indicates, cluster members are similar w.r.t. that principal features.
- All the principal features are selected by k-means clustering of the features. (By means of clustering the transposed the dataset.)

5.4. Pseudocodes for SFSC

Algorithmic structure of the SFSC is introduced in the Table 5.1.

Table 5.1. *Algorithmic Structure of the Proposed Methodology*

SFSC (Dataset, k)
1: Initial Step: Set all points in the dataset as unassigned and unvisited and construct KNN neighborhoods for all the points in the dataset and set c as 1.
2: Repeat:
3: Select initial KNN neighborhood (IN) by means of highest entropy among unassigneds.
4: Set IN as core neighborhood
5: Execute K-means where k=2, by transposing IN data.
6: Select one of the features from two feature clusters by means of maximal isotropy level
7: Expand_cluster (IN,PF,kmax)
8: c=c+1
9: Until all points in the dataset is assigned to a cluster.
10: Return Cluster id list and PF list
11: Expand_cluster (IN,PF)
12: for all unvisited elements in IN:
13: compute new Isolevel by adding its KNN points.
14: if new Isolevel>old Isolevel:
15: add these KNN points into IN
16: mark the current point as visited.
17: endif
18: else:
19: mark the current point as visited.
20: assign IN elements into cluster c and return

Both for cluster forming and determination of the principal features, isotropic position level of the neighborhood is used. Termination condition is the fact that all the points in the dataset are visited and assigned into a cluster.

5.5. Flowchart of SFSC

The flowcharts of the SFSC are also provided in the appendix part, both for main function and Expand-Cluster function in Figure 0.3 and Figure 0.4.

5.6. Time Complexity of SFSC

As explained in pseudocodes in the previous subsection, the proposed algorithm has two functions: Main function and expand cluster function. If the length of the dataset (n) is considered as the input size, time complexity of the proposed algorithm can be deducted as follow:

In main function, generating KD-Tree structure for KNN has O(nlog(n)) complexity on average and $O(n^2)$ on worst case. Finding the most compact region in the dataset has O(n) complexity, since there will be n-sized entropy list. K-Means for feature clustering yields O(n) complexity since K-Means has O(n) complexity. Finally selecting PF's O(n) complexity and dependent on size of IN. In summary, main function has O(nlog(n)) complexity on average. Complexity of computing covariance matrix for a d dimensional n sized set is O(n*d*min(n,d)). Since the covariance matrix computation is done for 2 featured PF set, it will bring O(n) complexity while selecting principal features.

In expand cluster function, since the covariance matrix computation and computing isotropic levels are done for 2 featured IN set, it will bring O(n) complexity. Visited markings, cluster assignments and adding new points into IN has constant time complexity: O(1).

Consequently, overall time complexity of the SFSC is O(nlog(n)) for the average case and $O(n^2)$ for the worst case, because of nested function structure and non-holistic concern on the dataset.

5.7. Application of SFSC on UCI Benchmark Datasets

The SFSC is carried out on some UCI benchmark datasets as an experimental analysis, in order to show its computational efficiency and clustering output quality by means of two validation indices such as rand index (RI), normalized mutual information score (NMI).

5.7.1. UCI Benchmark Datasets

In order to test the performance of the SFSC, 8 UCI (University of California - Irvine) (http-6) benchmark datasets are used. These multidimensional datasets and some of their characteristics are summarized in Table 5.2. Additionally, ground truth labels for each data is known a priori in every dataset. Since, these datasets are used to test different clustering algorithms before by means of several performance indices such as in (İnkaya, 2015), they are chosen to facilitate the comparison of the proposed methodology relatively to other methodologies.

Table 5.2. *Multidimensional datasets for experimentation and their characteristics*

Dataset	# of classes	# of features	# of data
Banknotes	2	5	1372
Breast Cancer	2	9	286
Hepta	7	3	212
Iris	3	4	150
Liver	2	7	341
Seeds	3	7	210
User K.	4	5	258
Vertebral	2	6	310

5.7.2. Clustering Output and Comparison

Proposed methodology is tested on each multidimensional dataset. In Table 5.3, clustering output is given. The performance of the proposed methodology is examined through 3 dimensions: Detection rate of the original clusters, (RI) and (NMI). Table 5.3, points out, the SFSC can present high quality clustering schema and promising validation scores for high dimensional datasets. In Table 5.3, CPU times for the experiment on each dataset are also given.

Table 5.4 summarizes the comparison w.r.t. RI scores for the proposed methodology and others. When clusters in infinitesimal sizes are assumed as noise in every SFSC experiment, they are declared as noise as a fine tune on clustering outputs. Therefore, RI can be increased up to outperforming values and Table 5.5. is given to depict it. When the SFSC is compared to several spectral clustering techniques which are summarized in (İnkaya, 2015) w.r.t. RI levels, some best practice level is achieved as in Hepta, Iris, Liver, User K. and Vertebral datasets. Since SFSC is not a supervised approach, KNN based classification methods or spectral graph theory based clustering methods are performing better than SFSC on some datasets.

Table 5.3. *Clustering Output of the SFSC*

Dataset	# of original classes	# of detected clusters	RI	NMI	CPU Times (sec)
Banknotes	2	6	0,527	0,067	92,74
Breast C.	2	4	0,562	0,044	14,1
Hepta	7	8	**0,997**	**0,991**	0,8
Iris	3	7	0,792	0,538	0,9
Liver	2	6	0,501	0,009	14,9
Seeds	3	6	0,651	0,299	3,2
User K.	4	13	0,648	0,114	3,6
Vertebral	2	7	0,486	0,135	11,6

Table 5.4. *The comparison of the SFSC and the others w.r.t. RI scores*

Dataset	SFSC	Best of KNN's	Best of MKNN's	Best of MST	DAN
Banknotes	0,527	**1.000**	**1.000**	0,887	**1.000**
Breast C.	0,562	0.891	0,891	0,841	**0,907**
Hepta	**0,997**	0,947	0,861	0,937	0,956
Iris	0,792	0,864	0,864	0,702	**0,876**
Liver	0,501	0,504	0,504	0,504	**0,512**
Seeds	0,651	**0,911**	0,910	0,904	0,891
User K.	0,648	0,675	**0,718**	0,699	0,701
Vertebral	0,486	0,534	0,534	0,559	**0,559**

Table 5.5. *The comparison of the SFSC and the others w.r.t. RI scores after noise declaration of infinitesimal sized clusters.*

Dataset	SFSC	Best of KNN's	Best of MKNN's	Best of MST	DAN
Banknotes	0,726	1.000	1.000	0,887	1.000
Breast C.	0,894	0.891	0,891	0,841	0,907
Hepta	**0,997**	0,947	0,861	0,937	0,956
Iris	**0,894**	0,864	0,864	0,702	0,876
Liver	**0,522**	0,504	0,504	0,504	0,512
Seeds	0,861	0,911	0,910	0,904	0,891
User K.	**0,729**	0,675	0,718	0,699	0,701
Vertebral	**0,644**	0,534	0,534	0,559	0,559

6. SFSC APPLICATION: CHURN ANALYSIS

Churn analysis is an important tool in service system for analyzing customer churn. Since the cost of a new subscription is getting higher than the retaining a current customer, churn analysis provides useful managerial insight for the service system companies in terms of predicting a churn or defining churn patterns and then providing them promotional activities to prevent the customer churn.

Churn analysis is important in service provider sector such as banking, insurance or in business which runs on subscriptions like telecom or internet service providers. (Celik & Osmanoglu, 2019) (Shaaban, Helmy, Khedr, & Nasr, 2012) Churn of a customer basically can be identified as being inclined to quit from a service provider. (Hashmi, Butt, & Iqbal, 2013)In other words, probable customer loss due to some dissatisfaction problems. Thus, churn rate, number of churners over total number of customer for a given time period, has a vital role for a customer relationships management (CRM) systems as a feedback.

There are two types of churns in general: Voluntarily churns and non-voluntarily churns. The non-voluntarily churns are those who are unsubscribed by the companies without any customer permission for some certain reasons. Voluntarily churns are the ones who voluntarily unsubscribed from the company for some deliberate reasons such as high price, competitors image and quality, low customer satisfaction or some incidental reasons such as change in social life, change in economic conditions or location change. (Shaaban, Helmy, Khedr, & Nasr, 2012) (García, Nebot, & Vellido, 2017) In Figure 6.1, a general classification for the churn reasons is given.

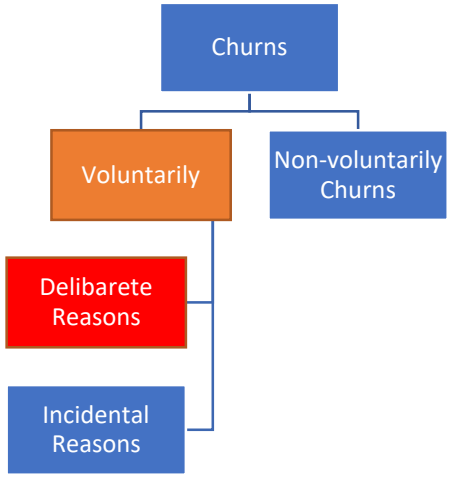

Figure 6.1. Why do customers churn? *(Shaaban, Helmy, Khedr, & Nasr, 2012)*

Churn analysis mostly deals with deliberate and voluntarily churns. The main goal of churn analysis is that determining the reasons of already churned customers, predicting the most likely churned ones and prevent them to quit. Thus, in Figure 6.1, voluntarily reasons are depicted in orange color and among them deliberate reasons are depicted in red color.

Not all types of churners are bad for a company such as churn of a nonpaying-bill types of customers and some already churned customers may have considered as to be returned in the future. A nonpaying-bill types of customer is a typical example of non-voluntarily churns and customers who switched to competitors for lower prices are typical example of a voluntarily churns.

Customer retention is the basic issue for a CRM system. The essential condition for retention of a customer is that increasing customers' perception of being satisfied as much as possible by some certain activities such as one-to-one marketing, loyalty programs and complaints management. (Ngai, Xiu, & Chau, 2009)

Customer segmentation is basically clustering the customer portfolio with respect to their similarities such as value, preferences, demand frequency etc. for targeted market and establishing convenient marketing. (Wu & Lin, 2005) Since, different customer segments may have different churn reasons (like price sensitive churners or quality sensitive churners), customer segmentation, which is mostly via clustering, may be a necessity for an accurate churn analysis.

Cluster analysis is the process of discovering data patterns or describing data heaps in unlabeled data sets by using a collection of techniques. In general, clustering approaches are classified into two groups: Traditional approaches mainly derived from classical computational manner, simple to implement but mostly not superior to others with respect to clustering schema quality. On the other hand, some modern approaches which might be superior to traditional approaches in clustering schema quality along with mostly increased computational burden relatively to traditional approaches. (Xu & Tian, 2015)

6.1. Research Questions and Contributions

The research questions of this paper can be listed as follows:

- Is it possible to make a churn analysis without using a predictive model such as classification and regression?
- Does only clustering along with dimension reduction provide a churn analysis?

- Do different customer segments have different churn reasons?
- Can clustering and dimension reduction be applied simultaneously?
- Can feature selection provide better conditions than feature extraction methods for clustering?
- Can dynamic feature selection provide customer churn reason?

As for the contributions of this study, we can list the following issues:

- A new approach is proposed for churn analysis.
- The new approach is based on generating churn map via simultaneous clustering and feature selection (SFSC).
- Since churn data sets are multidimensional, a churn map is provided by defining different customer segments and their most important two principal features.
- The new approach and algorithm is also experimented on a real-time churn data set in order to demonstrate their applicability and their benefits on churn analysis.

In summary, case study proposes a new churn analysis approach by generating a churn map via simultaneous clustering and dynamic feature selection and demonstrating applicability of them through experimental analysis on a real-time telecom churn data set.

6.2. Churn Literature

Machine learning algorithms are frequently used in churn analysis in supervised and unsupervised way: Classification algorithms such as logistics regression and support vector machines or clustering techniques such as K-Means or Hierarchical Clustering algorithms or some neural network models.

In (Celik & Osmanoglu, 2019) churn analysis is introduced and a comparison of techniques in churn analysis is given, but mostly supervised ones like Support Vector Machines (SVM), Regression Models or Artificial Neural Networks. Customer segmentation in marketing field is discussed in (Shaaban, Helmy, Khedr, & Nasr, 2012) and clustering applications are applied on a customer data set for credit card consumption by using a certain clustering application based on customer retention period, frequency and monetary features.

Moreover, a broader survey on the machine learning algorithms on churn analysis is given in (García, Nebot, & Vellido, 2017). In that study, mostly papers in which classification algorithms are used for churn analysis as predictive models are taken into account. But in a few of them, clustering techniques are used for churn analysis.

In (Wu & Lin, 2005), both classification and clustering algorithms are used for churn analysis to build a convenient marketing strategy. A real time data set which contains 23 attributes and 5000 data are used for the case study. After churn prediction, most likely churners are clustered into 3 groups as low, medium and high risks. K-means is used for clustering and decision tree, artificial neural networks and support vector machine is used for classification.

A broad review for clustering algorithms are given in (Xu & Tian, 2015). Churn determinants in the Korean mobile telecommunications service industry is investigated in a statistical way by hypothesis testing in (Ahn, Han, & Lee, 2006). Churn analysis in telecom industry is carried out by using logistics regression and decision trees in (Dahiya & Bhatia, 2015) by using software called WEKA.

In (Almana, Aksoy, & Alzahrani, 2014), common predictive data mining techniques to identify churners are surveyed such as neural networks, decision trees, statistical concepts and covering algorithms and future directions are given for these applications.

A churn analysis is done in (Ge, Xiong, & Brown, 2017) by using classification algorithms such as regression, extreme gradient boost classification (XGBoost) and random forests for a online software company. Principal component analysis is used for dimension reduction on the data set. A predictive churn analysis is done in (Huang, Kechadi, & Buckley, 2012) by using 7 techniques (Logistic Regressions, Linear Classifications, Naive Bayes, Decision Trees, Multilayer Perceptron Neural Networks, Support Vector Machines and the Evolutionary Data Mining Algorithm (DMEL)) in telecom industry.

In (Yang, Shi, Jie, & Han, 2018), a pipeline which include clustering and long-short term memory based deep learning is carried out on the data set of a mobile applications. New customers are clustered into six groups by using K-Means based on a single and multi-features. Then a predictive model is applied on each cluster. Clustering and classifications models are combined in (Huang & Kechadi, 2013) for the churn analysis in telecom industry. Logistic regression, decision trees and K-Nearest Neighbor classifiers are used for prediction and K-Means is used for customer clustering before prediction of churners. Moreover, a rule inductive model is used for each customer clusters.

In (Rajamohamed & Monokaran, 2018) a credit card churn analysis is done with respect to k-means, rough k-means and fuzzy c means customer segmentation and for each customer segment decision tree, support vector machine, random forest, Naive Bayes and K-Nearest neighbor classifiers are used for churn prediction in order to decrease classification error. A novel clustering algorithm, Semantic-driven subtractive clustering method (SDSCM), is proposed in (Bi, Cai, Liu, & Li, 2016) and used for telecom customer clustering along with k-means and fuzzy c-means on a telecom data set which include 1.8 million subscribers and 42 feature for each customer as a big data. Moreover, clustering results are used for convenient management solution for marketing and campaigns.

In some papers, churn analysis is not directly named but a churn-like analysis is done. For instance, in (Bose & Chen, 2015), churn analysis is not named, but customers in telecommunication sector is segmented into groups by using extended fuzzy c-means algorithm, to determine the customer migration with respect to their revenue patterns. In (Jonker, Piersma, & Van den Poel, 2004), customers are segmented into groups according to their customer retention period, frequency and monetary features for a charity organization. Then by using MDP and GA techniques, optimal marketing policy is determined for retention in each segment. Finally, in (Shin & Sohn, 2004), stock traders are clustered by using k-means, fuzzy c-means and SOM, then a decision tree process is used to determine different brokerage commission rates for each cluster.

A summary of the literature survey is given in Table 6.1 and Table 6.2 as well.

Table 6.1. *Related Papers for Churn Analysis*

Paper	Churn Sector	Clustering Models
(Celik & Osmanoglu, 2019)	-	-
(Shaaban, Helmy, Khedr, & Nasr, 2012)	Credit Card	Hierarchical
(Wu & Lin, 2005)	Telecom	K-Means
(Ahn, Han, & Lee, 2006)	Telecom	-
(Almana, Aksoy, & Alzahrani, 2014)	Telecom	-
(Dahiya & Bhatia, 2015)	-	-
(Ge, Xiong, & Brown, 2017)	Software	-
(Huang, Kechadi, & Buckley, 2012)	Telecom	-
(Yang, Shi, Jie, & Han, 2018)	Mobile App.	K-Means
(Huang & Kechadi, 2013)	Telecom	K-Means
(Rajamohamed & Monokaran, 2018)	Credit Card	K-Means
		Rough K-Means
		Fuzzy C- Means
(Bi, Cai, Liu, & Li, 2016)	Telecom	SDSCM, K-Means
		Fuzzy C- Means
(Bose & Chen, 2015)	Telecom	Fuzzy C-Means,
		Extended Fuzzy C-Means
(Jonker, Piersma, & Van den Poel, 2004)	Charity	Optimization
(Shin & Sohn, 2004)	Stock Market	K-Means,
		Fuzzy C-Means, SOM

Table 6.2. *Related Papers for Churn Analysis (Cont'd)*

Paper	Predictive Models	Dim. Red.
(Celik & Osmanoglu, 2019)	SVM, Reg., ANN	-
(Shaaban, Helmy, Khedr, & Nasr, 2012)	-	-
(Wu & Lin, 2005)	SVM, ANN, DT	-
(Ahn, Han, & Lee, 2006)	Statistical	-
(Almana, Aksoy, & Alzahrani, 2014)	LR, DT	-
(Dahiya & Bhatia, 2015)	ANN, DT, Statistical,	-
	Covering Algorithms	
(Ge, Xiong, & Brown, 2017)	RF, Reg. , XGBoosts	PCA
(Huang, Kechadi, & Buckley, 2012)	LR, RC, DT, ANN, SVM,	-
	MP, NB, DMEL	
(Yang, Shi, Jie, & Han, 2018)	DL, LSTM	FS
(Huang & Kechadi, 2013)	LR,DT, KNN	-
(Rajamohamed & Monokaran, 2018)	DT, SVM, RF, NB, KNN	-
(Bi, Cai, Liu, & Li, 2016)	-	-
(Bose & Chen, 2015)	-	-
(Jonker, Piersma, & Van den Poel, 2004)	MDP	-
(Shin & Sohn, 2004)	DT	-

Consequently, literature survey shows that both clustering and classification models are necessary for churn analysis, but dimension reduction (feature extraction or feature selection models) is necessary for data sets which is going to be used in churn analysis. Moreover, no study exists in the literature which deals with churn analysis in simultaneous clustering and dimension reduction algorithm along with the predictive models.

6.3. The Proposed Framework for Churn Analysis

Churn analysis generally relies on either one of segmentation and prediction or combination of them as in related papers, which are summarized in Table 6.1. For segmentation of the customers, clustering applications such as k-means, fuzzy c-means or hierarchical algorithms are generally carried out. As for the churn prediction, classification algorithms are preferred like DT, SVM, RL, ANN, RF etc...

Figure 6.2. represents the approaches in the studies in which only churn prediction models are used as in (Ahn, Han, & Lee, 2006), (Celik & Osmanoglu, 2019), (Dahiya & Bhatia, 2015), (Almana, Aksoy, & Alzahrani, 2014), (Ge, Xiong, & Brown, 2017), (Huang, Kechadi, & Buckley, 2012). Moreover, as in (Shaaban, Helmy, Khedr, & Nasr, 2012) and (Bi, Cai, Liu, & Li, 2016), only clustering process is used instead of a classification model. But in (Ge, Xiong, & Brown, 2017), PCA is also applied on as a dimension reduction stage, before classification models are carried out.

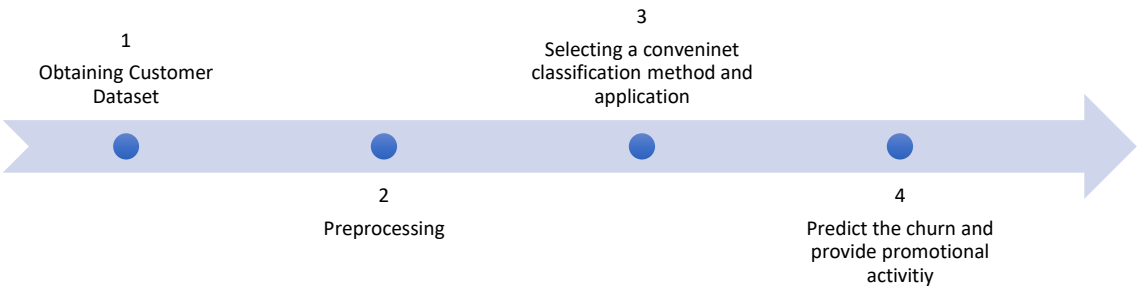

Figure 6.2. *General framework which is preferred by the researchers*

Figure 6.3. represents the approaches in studies that customer segmentation and churn prediction models are combined as in (Wu & Lin, 2005), (Yang, Shi, Jie, & Han, 2018), (Huang, Kechadi, & Buckley, 2012), (Rajamohamed & Monokaran, 2018). In only (Yang, Shi, Jie, & Han, 2018) a feature selection algorithm is also used for dimension reduction process.

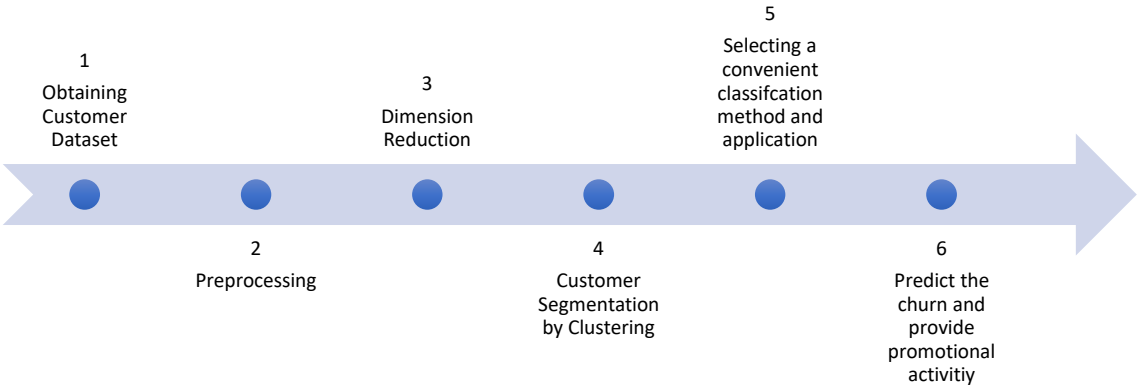

Figure 6.3. *Pipeline of Clustering and Classification for Churn Analysis*

In this study, churn analysis is done, with respect to a customer segmentation process by simultaneous feature selection and clustering algorithm (SFSC) and for each segment churn prediction is done by KNN classifier, contrarily to reference works. The new approach is represented as in Figure 6.4.

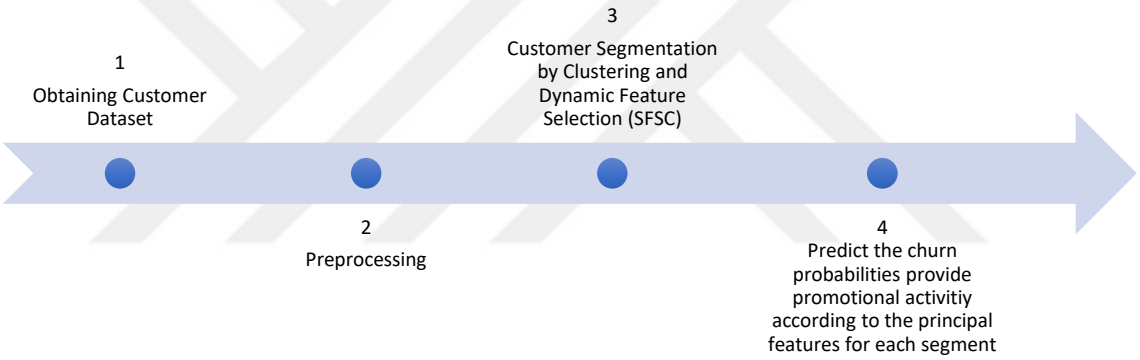

Figure 6.4. *Simultaneous Feature Selective Clustering Algorithm and Classification Pipeline*

By the proposed approach, customer clusters are expected to be found and more accurate churn prediction to be done. Moreover, for each customer segment, it will be possible to see why customers churn with respect to dynamic features.

6.4. Customer Dataset for Churn Analysis and Preprocessing

Churn analysis is done by using a customer portfolio dataset. The dataset is retrieved from (http-7) and it is a customer portfolio of an anonymous telecom company based in USA for annual data in one of the past years. It contains 10.000 customer data and for each customer 37 attributes. There are 2650 churns and 7350 non-churns in the data set and it also exists in the dataset as labels. Basic description of all the attributes and their types (categorical, discrete, continuous or binary) are given in the Table 6.3.

Table 6.3. *Feature Description and Data Types in the Churn Dataset*

Attribute	Description	Type
City	Customer's city	Categorical
State	Customer's state	Categorical
County	Customer's country	Categorical
Zip	Customer's zip code	Categorical
Latitude	Customer's latitude location	Continuous
Longitude	Customer's longitude location	Continuous
Population	Population of customer's city	Continuous
Area	Area type of customer's city	Categorical
Time Zone	Time-zone of customer's city	Categorical
Job	Customer's job	Categorical
Children	# Of children of a customer	Continuous
Age	Customer's age	Discrete
Income	Customer's annual income	Continuous
Marital	Customer's marital status	Categorical
Gender	Customer's gender	Binary
Outage per week	Outage per week in seconds	Continuous
Email	# Of daily emails for the customer	Discrete
Contacts	# Of customer contact	Discrete
Yearly equip failure	# Of yearly equip failures	Discrete
Techie	Is customer is fond of tech?	Binary
Contract	Customer's contract	Categorical
Port modem	Does customer have port modem?	Binary
Tablet	Does customer have tablet?	Binary
Internet Service	Type of internet service?	Categorical
Phone	Does customer have phone?	Binary
Multiple	Does customer have multiple lines?	Binary
Online Security	Does customer have online-security?	Binary
Online Backup	Does customer have online-backup?	Binary
Device Protection	Does customer have device-protection?	Binary
Tech Support	Does customer have tech support?	Binary
Streaming TV	Does customer have tv streaming?	Binary
Streaming Movies	Does customer stream movies?	Binary
Paperless Billing	Does customer have paperless billing?	Binary
Payment Method	Which payment method does customer do?	Categorical
Tenure	How many years the customer subscribes?	Continuous
Monthly Charge	Mean monthly charge of the customer	Continuous
Bandwidth GB Year	Mean annual GB usage of the customer	Continuous

As a data cleaning step, all of the categorical and binary variables are discarded from the data set. Only some discrete and continuous variables are taken into account such as Population, Children, Age, Income, Outage per week, Email, Contacts, Yearly equip failure, Tenure, Monthly Charge, Bandwidth(GB) Year, since the others do not provide a valuable information and not convenient for dimensional reduction and clustering process. At the final stage of cleaning process, only these 11 variables are used.

As the scaling step, standard scaler is used, to make data set convenient for dimension reduction and clustering. Thus, each attribute is transformed as normal distribution.

In Figure 0.5, a pair plot is given which visualize the scatter of data between for all pair of variables in all to all mode. High correlations between tenure and bandwidth variables can be easily seen, but there are infinitesimal correlations between other pairs. Diagonal plots are kernel density estimation for each variable. In Figure 0.6, a heat map which summarizes the correlation between variables in all to all mode. High correlations between tenure and bandwidth variables can be easily seen, but there are infinitesimal correlations between other pairs as in the pair plot. Diagonal values (ones) represent the correlation between a variable and itself.

6.5. Churn Map Generation

A churn map is generated by using the proposed approach and SFSC. In Figure 6.5. and Figure 6.6., churn maps are given. Each point represents a distinct customer. Every cluster (customer segment) has different principal features and it represents why a customer belongs to that cluster. Number of clusters (customer segments) is determined by the algorithm, not by the user. As a result, churn map can be used for determination of churn probabilities by such KNN (Where k=50) classifier methodology for all existing customers and each most likely churn customer will be provided promotional activity in accordance with its principal feature which makes him/her into his segment.

In Figure 6.5, all the clusters (customer segments) are given in 2d (Different Principal Features for each cluster by SFSC) as the output of SFSC and that bears an overlapping cluster visual, since 11 dimensional churn datasets is embedded into 2d by SFSC. In Figure 6.5, each customer segment is represented by a distinct color and each customer segment (27 clusters) may have different set of principal features. Black dots represent noise points. In Figure 6.6., clustering output is colored in terms of original

customer label (churn- yellow & non-churn-purple). Churn and non-churn customers seem to be distributed homogenously in the dataset.

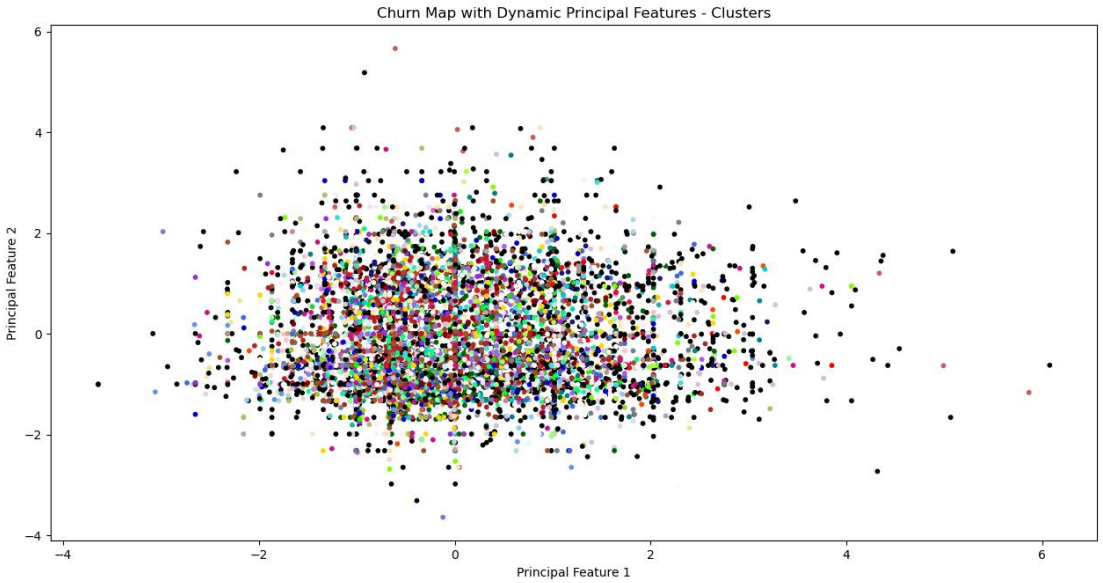

Figure 6.5. *Clustering Output of the SFSC.*

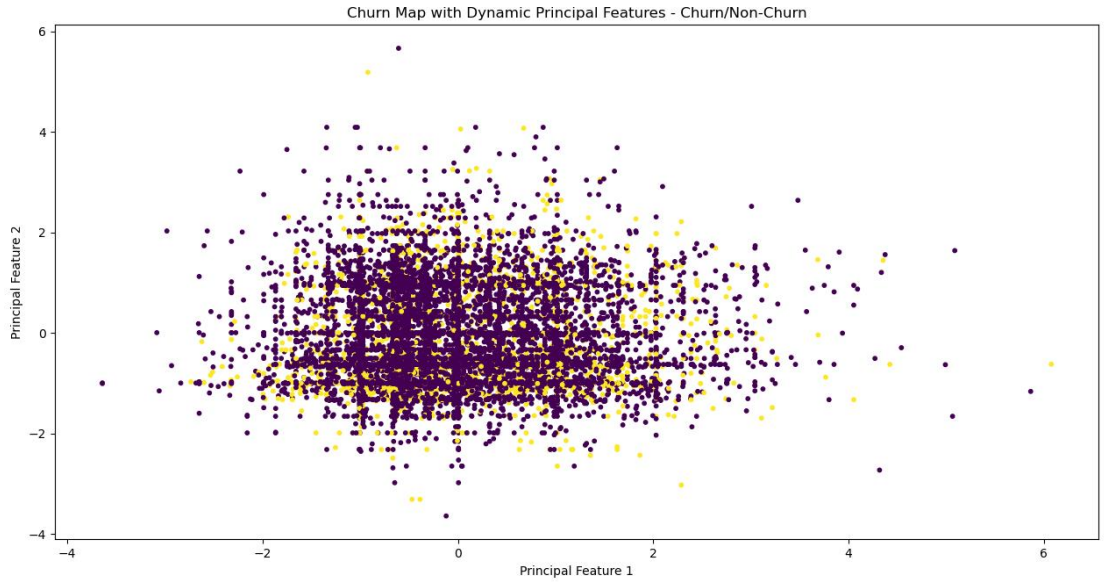

Figure 6.6. *Clustering Output of the Proposed Methodology for original churn/non-churn representation. Yellow points represent churned customers and others are non-churners*

To avoid overlapping cluster visual, each cluster can be visualized respectively. For instance, Cluster1 (493 customers) in Figure 6.7 and Cluster_2 (246 customers) in Figure 6.8 and Cluster_3 (317 customers) in Figure 6.9.

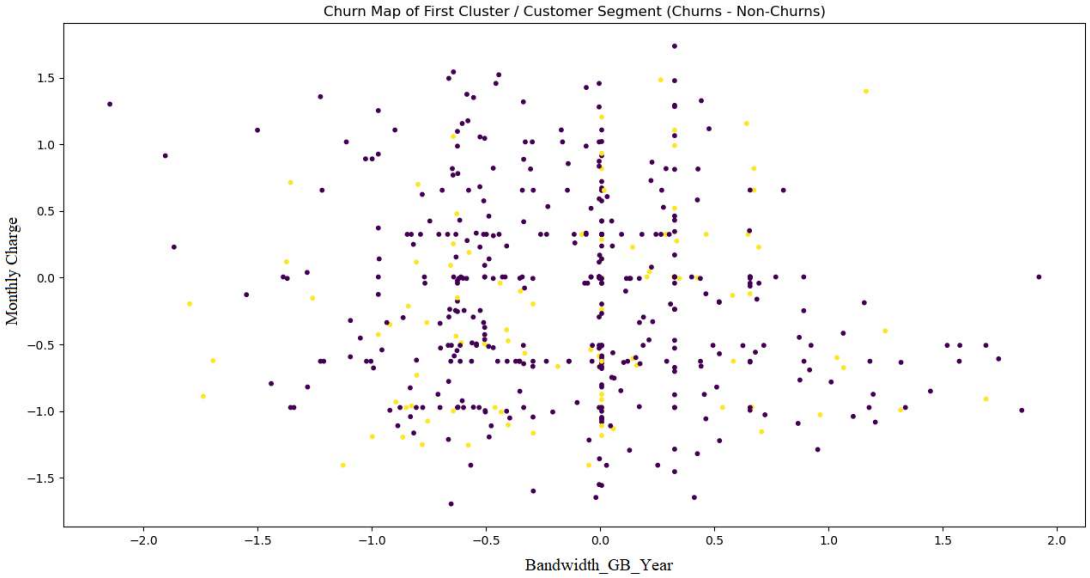

Figure 6.7. *Cluster_1 visualization by SFSC (Yellow represents churned customers and purple represents non-churned customers.)*

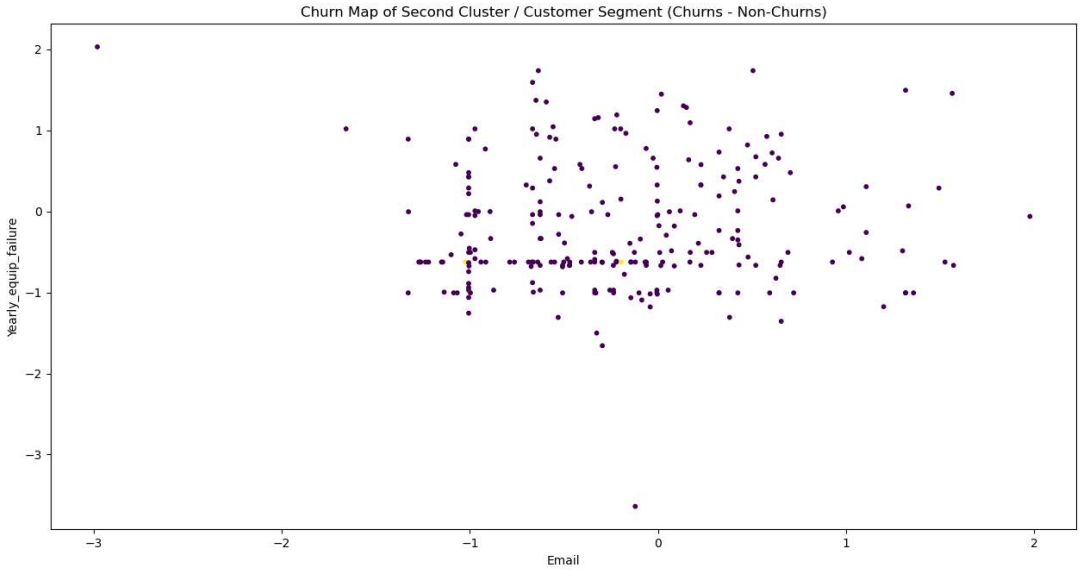

Figure 6.8. *Cluster_2 visualizations by SFSC (Yellow represents churned customers and purple represents non-churned customers.)*

On the output of the proposed methodology, some descriptive samples can be obtained as well, such as cluster ID's of customers and their principal features which make them a member of their segment. Such lists represent cluster id's and principal features: A sample cluster id list= [11,13,3,11,1,7,...,18].

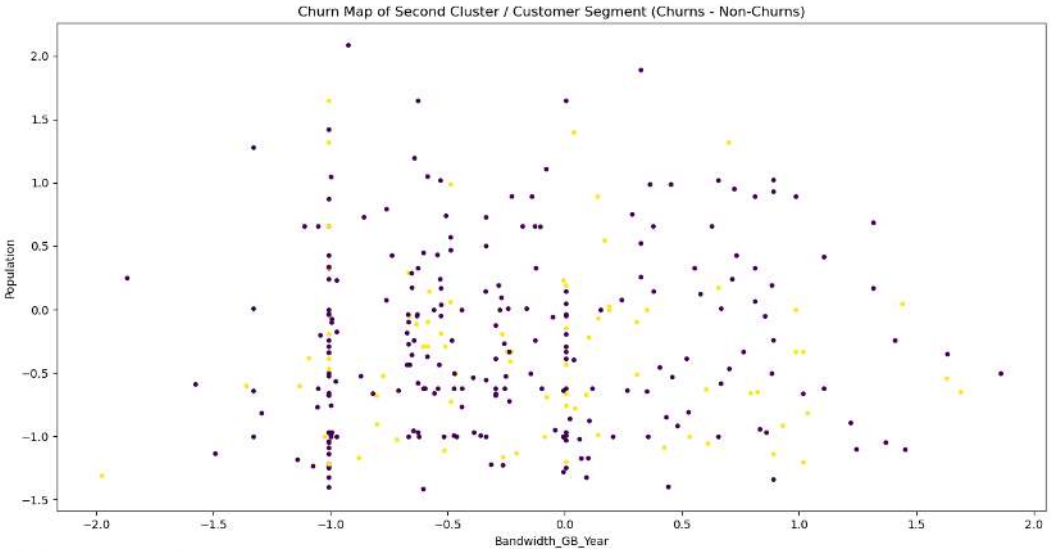

Figure 6.9. *Cluster_3 visualizations by SFSC (Yellow represents churned customers and purple represents non-churned customers.)*

The principal feature (PF) set is like PF = [[children, Yearly equip failure], [income, Yearly equip failure], [Population, Bandwidth], [children, Yearly equip failure], [Age, Income], [Tenure, Monthly Charge], [Tenure, Age], [Age, Contacts], [Population, Outage],...,[Children, Bandwidth]] on the output. PF's for some clusters are given in Table 6.4.

Table 6.4. *Principal Feature Sets for Some Customer Segments and Possible Promotional Activities*

Cl__ID	Principal Features	Possible Promotional Activity
1	Bandwidth & Monthly Charge	Bandwidth Increase or Discounts
2	Email & Yearly equip failure	Enhanced Customer Service Quality
…	…	…
26	Income & Tenure	Discounts, Long Term Contracts
27	Tenure & Yearly Equip Failure	Long Term Contracts or Enhanced Customer Service Quality

For instance, a customer belongs to cluster 11 depending on feature sets: children and Yearly equip failure or second customer belongs to cluster 13 depending on income, Yearly equip failure. Likewise, the another customer belongs to cluster 3 depending on feature sets Population, Bandwidth. Just like first instance, another customer belongs to cluster 11, depending on feature sets children and Yearly equip failure.

When each point (customer) is analyzed via KNN classifier, we also get a churn probability for every customer in accordance with number of churn rate among their KNN where k is hypothetically chosen as 10, 20 or 50. The churn probability list for all

customers is [0.6,0.4,0.4,0,8,0.2, ... ,0.1] (where k=10), [0.6,0.5,0.5,0,8,0.3,0.1, ..., 0.2] (where k=20) and [0.65,0.50,0.50,0,75,0.30,0.15, ..., 0.25] (where k=50). In Figure 6.10 churn probabilities are depicted as color intensity for each customer. As the point color gets darker, the churn probability gets higher.

The promotional activities are going to be offered to the customers by the company, in accordance with the PF of the segment that each customer belong to and churn probabilities which is computed for each customer by assuming only within his/her customer segment respectively. For instance, customer1 belong to cluster1 and his/her churn probability is %80, then the company will present bandwidth increase or discounts in monthly charge, since the PF of cluster1 are Bandwidth & Monthly Charge. Similarly, customer2 belongs to cluster2 and his/her churn probability is %5, then no promotional activity will be offered for email and yearly equip failure.

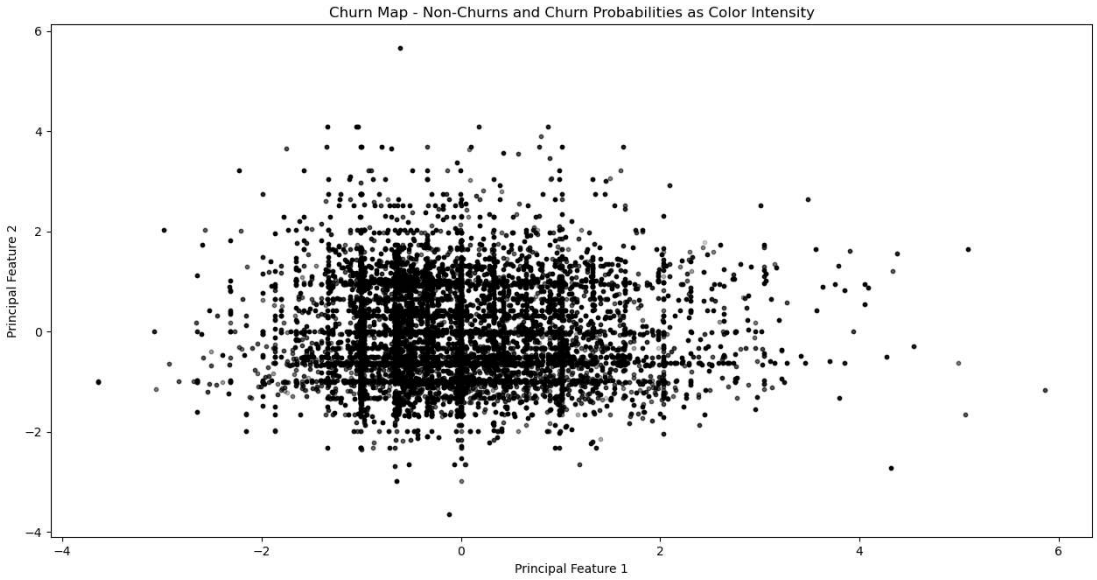

Figure 6.10. *Churn Probabilities for non-churn customers as color intensity (where k=50)*

6.6. Customer Segmentation via Existing Methods

Customer segmentation is also discussed via existing dimension reduction and clustering methods for comparison with the churn map via SFSC.

6.6.1. Dimension Reduction via Existing Methods

Since the data set have 11 features for each customer, visualization of the clustering output is impossible and there may be some redundant features. For instance, tenure and bandwidth usage of customers are highly correlated, but others not. And visualization of the churn data set is impossible because number of dimensions is not 2

or 3. In this case three different approach is used here: PCA and t-SNE as feature extraction and PROC VarClus as feature selection.

Firstly, PCA which is a feature extraction algorithm is used for the churn data set. For PCA, number of dimension is tested on the interval as [2,11], and for each item in this interval, explained variance ratio is given. In Figure 6.11., explained ratio for given each number of component is illustrated. Because of elbow method rule, number of new dimensions is selected as 2. After the second component, explained variance ratio does not drastically fall, that's why 2 new principal feature can be selected in PCA. The same output and inference can be seen in (http-7) as well. And loadings for each new principal feature is illustrated in terms of original feature vectors in Figure 6.12. PCA_x is highly positively dependent on tenure and bandwidth features, on the other hand PCA_y on both positively and negatively dependent on other features.

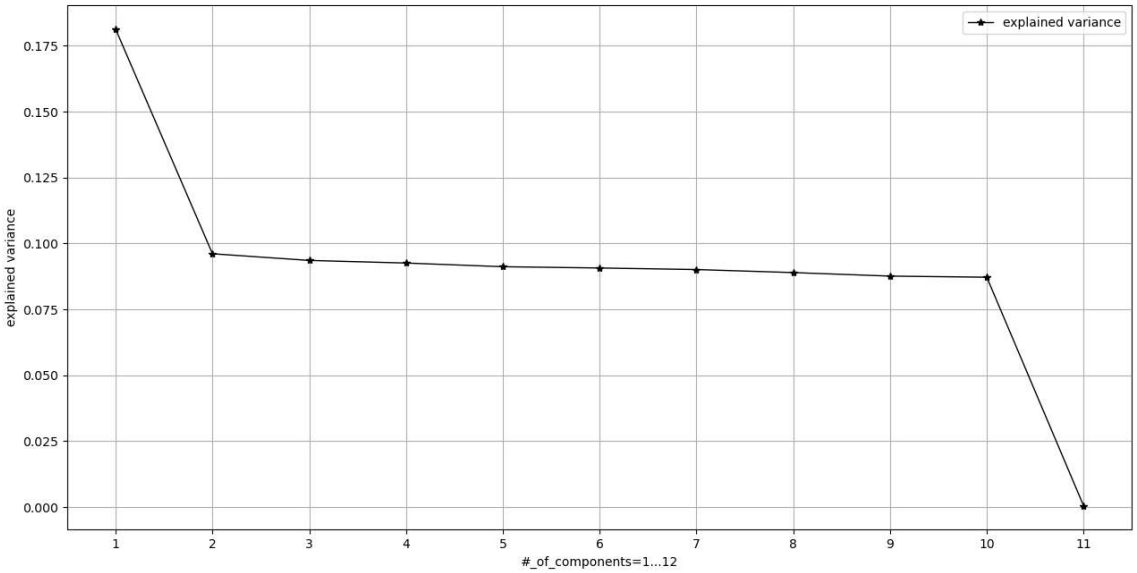

Figure 6.11. *Explained Variance vs. Number of Component for PCA*

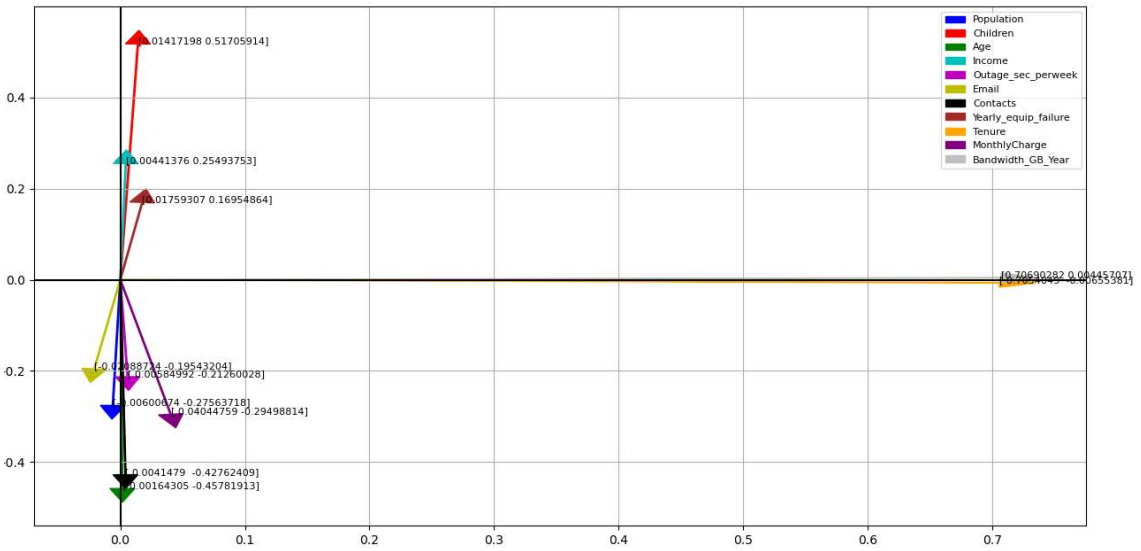

Figure 6.12. *Loadings for each new principal component (PCA_x and PCA_y) in terms of original features*

After PCA result, data set is reduced in 2 dimensions and plotted as in Figure 6.13. Since we know churn variables, red dots represent churned customers and blues represent non-churners.

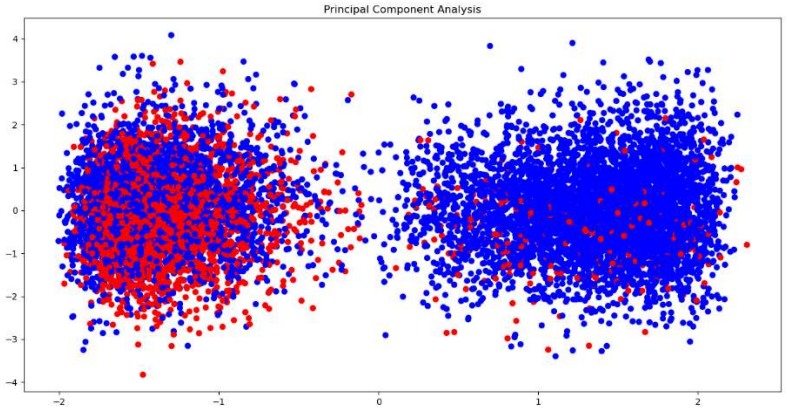

Figure 6.13. *Churn Dataset Plot for PCA Output*

Secondly, t-SNE is used which is a non-linear feature extraction algorithm is used for the churn data set. For t-SNE, number of new dimension is set as 2 accordingly to the PCA result. And perplexity parameter is set as 30 as a default setting. In Figure 6.14., t-SNE visualization is given. Since we know churn variables, red dots represent churned customers and blues represent non-churners.

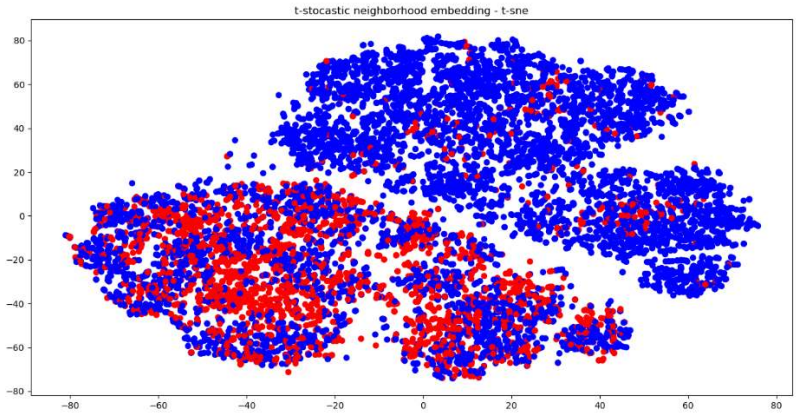

Figure 6.14. *Churn Dataset Plot for t-SNE Output*

Thirdly PROC VarClus is used which is a feature selection methodology. In VarClus, original variables in data are preserved after the dimension reduction process and clustering of variables are done by means of a hierarchical clustering by maximizing explained variance. R-Square Own Cluster, R-Square Next Cluster and RS_Ratio and cluster id of each feature is given in Table 6.6. According to the Table 6.6., tenure or bandwidth can be selected as the representative features for the first cluster since they have the same RS_Ratio values as 0.004253 but tenure is selected arbitrarily. For the second cluster, best representative feature is children since it has the lowest RS_Ratio value as 0.719891. So the principal component of the data set will be tenure and children for the VarClus applied churn data set.

For VarClus, eigenvalues and variance proportion is also given for the new feature clusters in Table 6.5. When PROC VarClus is applied on churn data set, plot for churn data set is given in Figure 6.15. Contrarily to previous methods, Varclus applied data set is much more categorical, since children attribute has only discrete values such as 0,1,2,3,4,6 etc. As a result, difference from PCA and t-SNE is that original features are preserved instead of linearly or non-linearly combining some of them into new ones.

Table 6.5. *VarClus Output Eigenvalues and Variance Proportion for Each Feature Cluster*

Cluster	# of Variables	Eigval1	Eigval2	Var_Prop
0	2	1.991.45	0.008505	0.995747
1	9	1.057.38	1.029.01	0.117483

Table 6.6. *VarClus Output Variable Clusters with some values*

Cluster	Variable	RS_Own	RS_NC	RS_Ratio
0	Tenure	0.99577	0.0001	0.004253
0	Bandwidth(GB)	0.99577	0.0008	0.004253
1	Population	0.08017	0.0005	0.919816
1	Children	0.28014	0.0003	0.719891
1	Age	0.21845	0.0001	0.781597
1	Income	0.06901	0.0008	0.931007
1	Outage(sec/week)	0.04895	0.0002	0.951097
1	Email	0.04094	0.0004	0.959222
1	Contacts	0.19334	0.0009	0.806653
1	Yearly Equip F.	0.03101	0.0003	0.969137

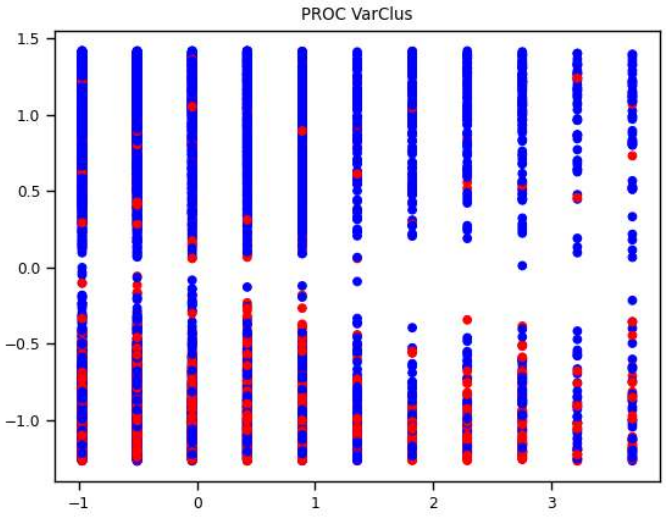

Figure 6.15. *Churn Dataset Plot for VarClus Output*

6.6.2. Clustering via Existing Methods

After the dimensional reduction process, dimensionally reduced data set is obtained with 2 dimensions. In clustering process k-means and fuzzy c-means algorithms are used to unearth the data heaps in churn data set. Both algorithms are used for the data set coming from the dimensional reduction process. In summary, there will be 2x3=6 clustering schema output: PCA+K-Means, PCA+Fuzzy C-Means, t-SNE+K-Means, t-SNE+Fuzzy CMeans, VarClus+K-Means, VarClus+Fuzzy C-Means.

From Figure 6.16. up to Figure 6.24., knee method graphs are provided for choosing best resultant clustering output (based on different dimension reduction processes) with respect to k or c values according to Davies-Bouldin Score (DB),

Calinski-Harabasz Score (CH), Within Cluster Sum of Square(WCSS) for k-means and fuzzy partition coefficient for fuzzy c-means. For PCA+Kmeans, k is set as 2, PCA+Fuzzy C-Means, c is set as 2, for t-SNE+K-Means, k is set as 4, for t-SNE+Fuzzy C-Means, c is set as 4, for VarClus+K-Means,k is set as 6 and for VarClus+Fuzzy CMeans, c is set as 6 according to validation scores.

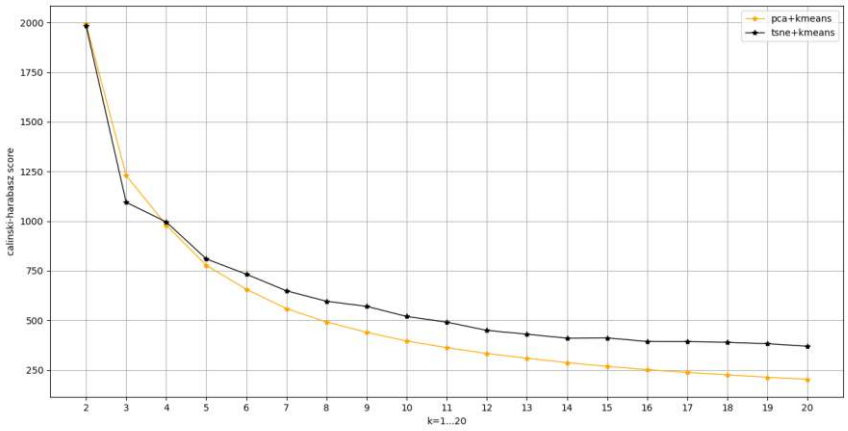

Figure 6.16. *CH Scores for (t-SNE or PCA) and K-Means*

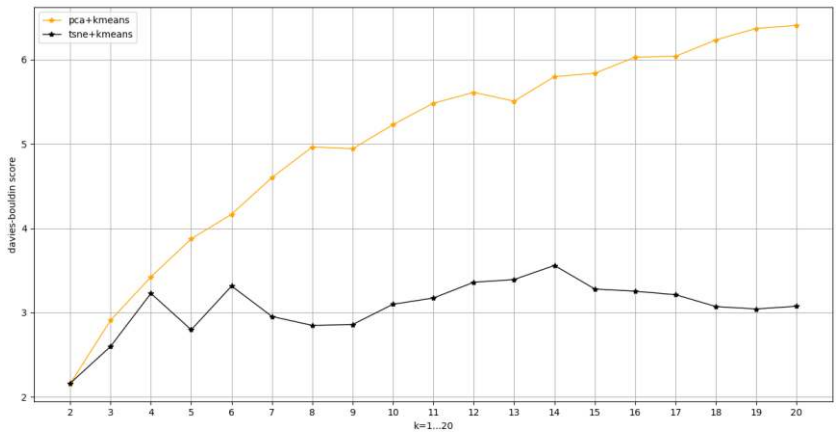

Figure 6.17. *DB Scores for (t-SNE or PCA) and K-Means*

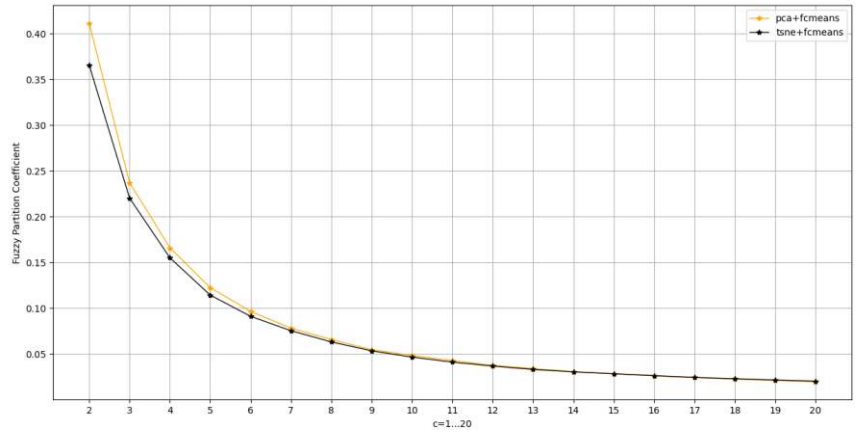

Figure 6.18. *Fuzzy Partition Coefficient for (tSNE or PCA) and Fuzzy C-Means*

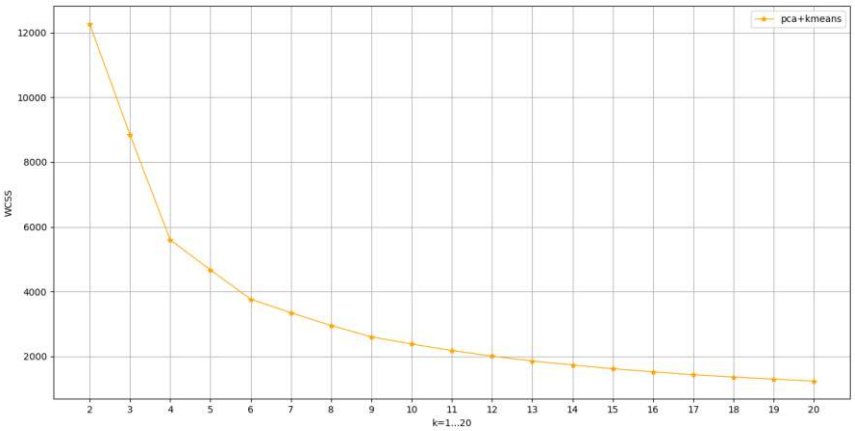

Figure 6.19. *WCSS for PCA + K-Means*

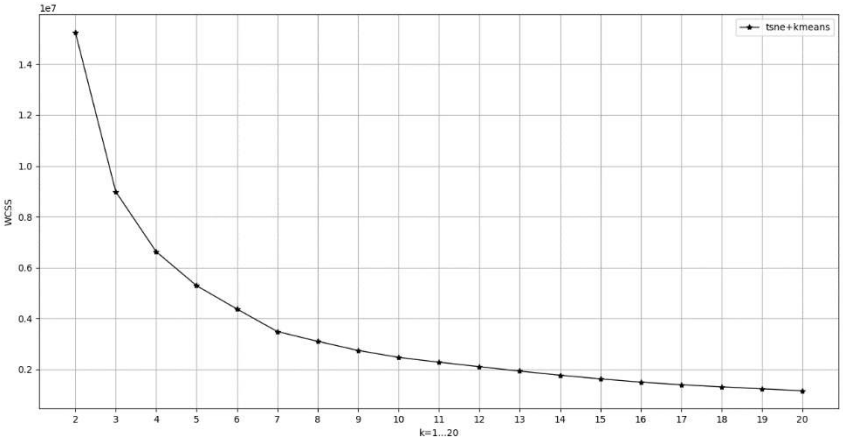

Figure 6.20. *WCSS for t-SNE + K-Means*

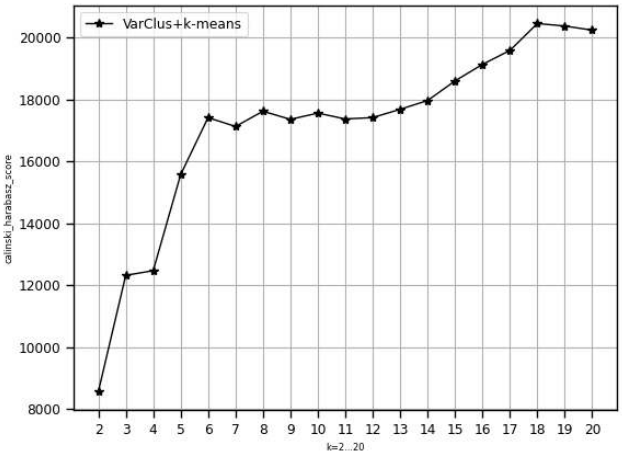

Figure 6.21. *CH Scores for VarClus + K-Means*

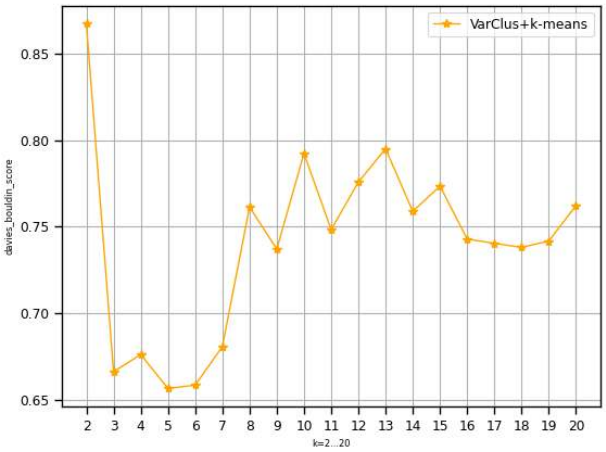

Figure 6.22. *DB Scores for VarClus + K-Means*

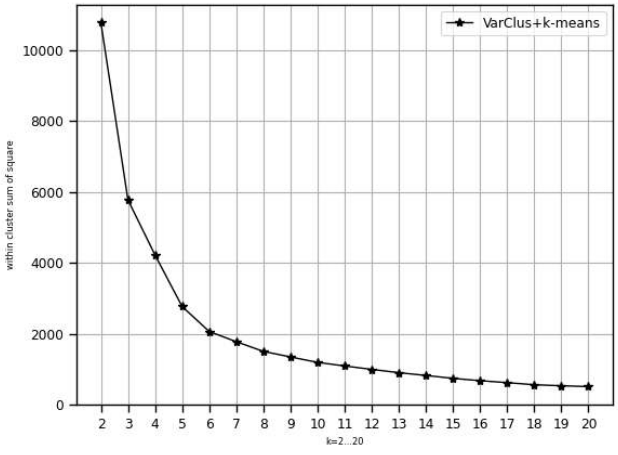

Figure 6.23. *WCSS for VarClus + K-Means*

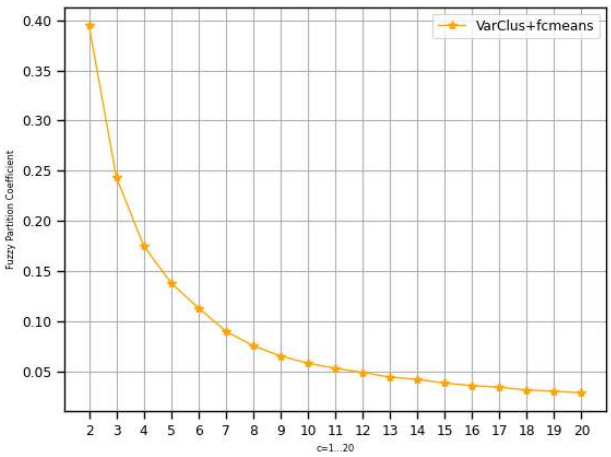

Figure 6.24. *Fuzzy Partition Coefficients for VarClus + Fuzzy C-Means*

For visual check, clustering schemas are also provided from Figure 6.25. up to Figure 30. just for best resultant k or c values according to validation graphs. Thus, customers are segmented in 2 dimensions but clustering schemas do not provide meaningful insights to the users for a comprehensive customer segments knowledge for all algorithmic pipelines.

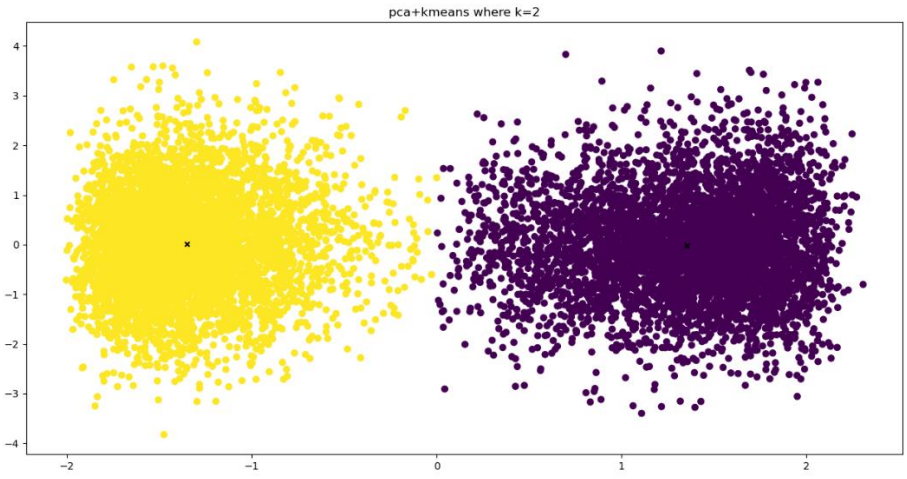

Figure 6.25. *Clustering Schema for PCA + K-Means for k=2*

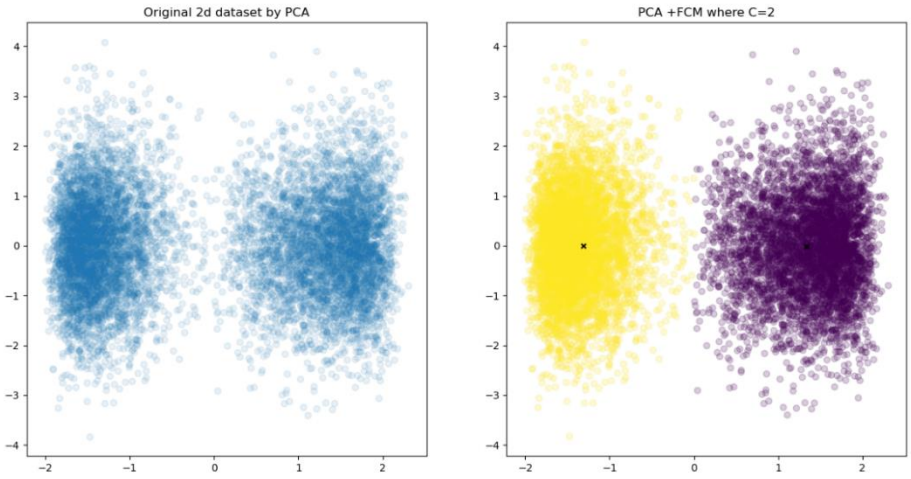

Figure 6.26. *Clustering Schema for PCA + Fuzzy C-Means for c=2*

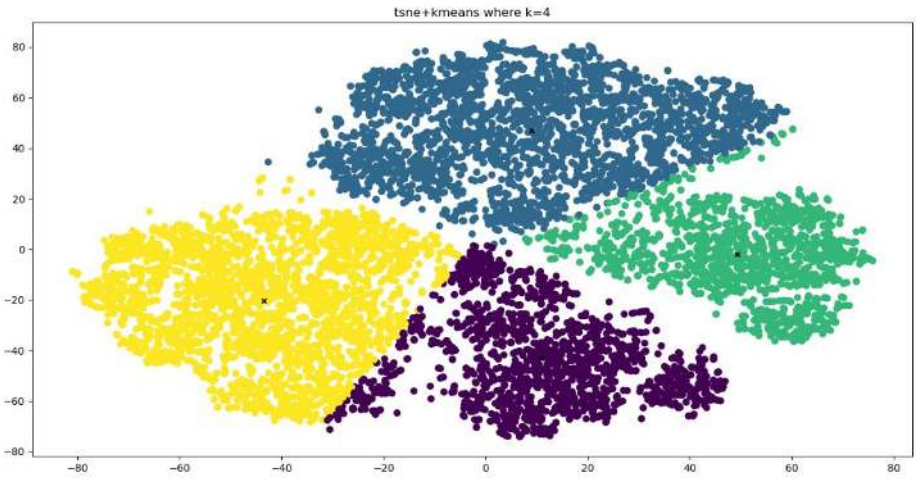

Figure 6.27. *Clustering Schema for t-SNE + K-Means for k=4*

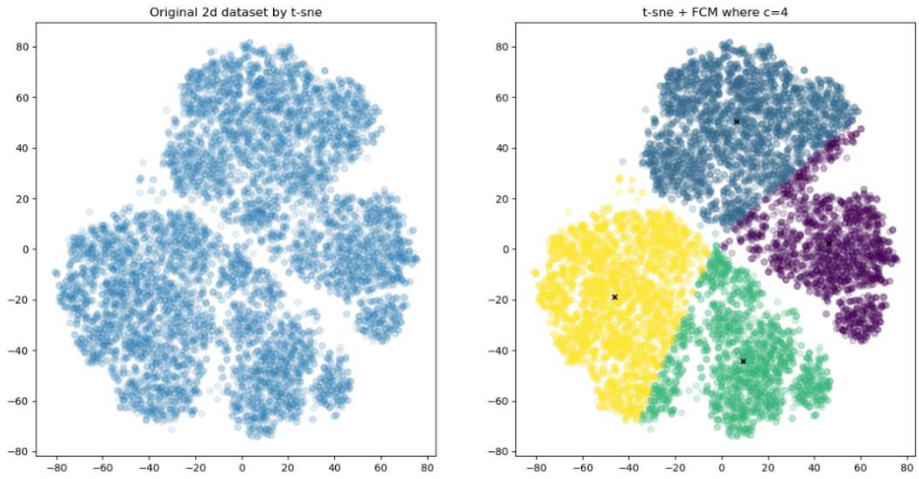

Figure 6.28. *Clustering Schema for t-SNE + Fuzzy C-Means for c=4*

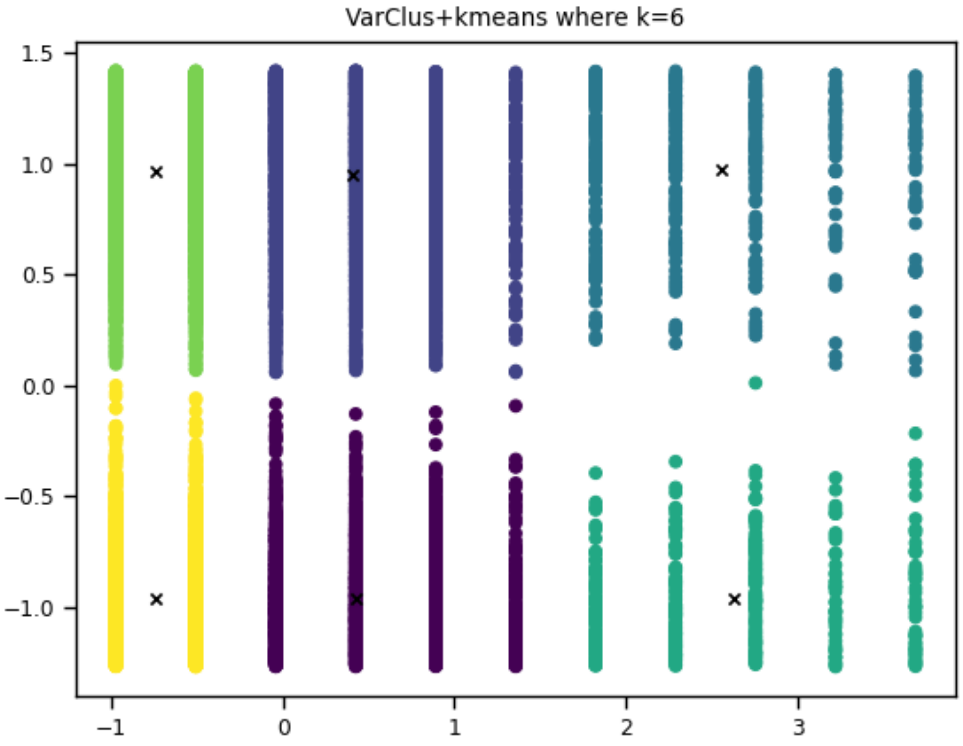

Figure 6.29. *Clustering Schema for PROC VarClus + K-Means for k=6*

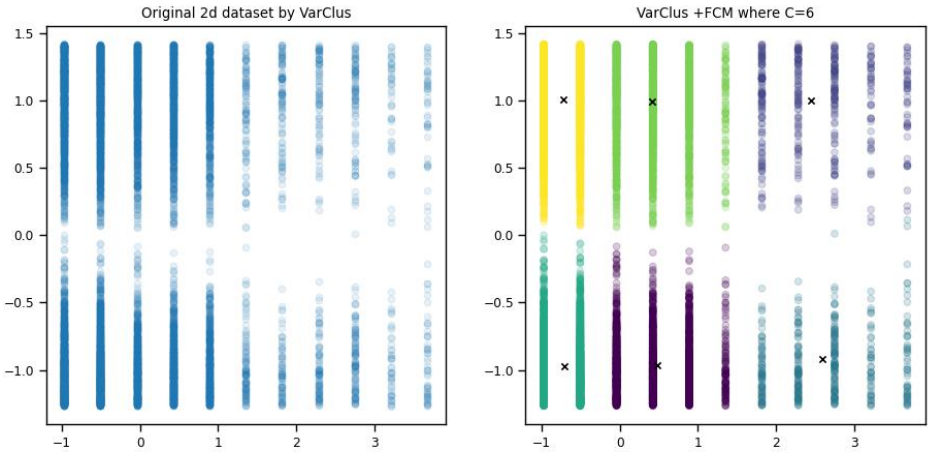

Figure 6.30. *Clustering Schema for PROC VarClus + Fuzzy C-Means for c=4*

7. CONCLUSION

In the thesis, novel clustering algorithms based on entropy, symmetric relative entropy and isotropic position concepts are developed for both arbitrary shaped cluster detection (ENM) and multidimensional feature space (SFSC). Experimental analysis is carried out on benchmark datasets which include arbitrary geometric shaped clusters along with some density variations and noise problems to illustrate the robustness of ENM. Another experimental analysis is done on multidimensional UCI benchmark datasets to show the efficiency of SFSC. Moreover, a seismic zone detection problem is solved and a churn analysis is done to illustrate the applicability and robustness of ENM and SFSC on real time problems. The main concluding remarks of this thesis are in the following paragraphs:

Firstly, a novel algorithm (ENM) based on reinforcement learning concept for clustering is introduced. The proposed concept is capable of unearthing arbitrary shaped clusters in the datasets along with the sensitivity to both inter-cluster and intra-cluster density variations. The biggest advantage of the ENM is that preliminary number of clusters knowledge is not required and being parameter-free. Thus, more user-friendly clustering environment is created with lower computational burden. Density is described by maximal normalized entropy in the ENM, instead of number of points in the epsilon neighborhood. Instead of a threshold value for merging compact regions, a reinforcement learning methodology is used in dynamical programing framework. It is capable of handling arbitrary shaped cluster detection by utilizing divide and conquer manner and, it is more sensible to density variations because of slow convergence in RL framework. A simple implementation of the proposed methodology is given. Then, the ENM is tested on some planer benchmark datasets including larger scale ones and 3d benchmark datasets. The experimental analysis and comparison points out that, the ENM outperforms some well-known, robust methodologies like k-means, DBSCAN or Spectral Clustering in terms of clustering schema quality, validations index scores and applicability on larger scale datasets. Moreover, proposed methodology provides relatively lower computational complexity, thus it can produce the clustering output within reasonable times.

Furthermore, a real-life problem, seismic zone detection of Turkey, is solved via ENM. In this case study, ENM presented better clustering schema then existing clustering methodologies like k-means, DBSCAN and Spectral Clustering with respect to real-life seismic zone map.

Secondly, SFSC, a parameter free statistical method for simultaneous clustering and dimensional reduction based on isotropic position and entropy is proposed. The SFSC has no requirement of preliminary number of cluster knowledge. In SFSC, every cluster may have different principal features and it provides easier labelling of cluster items and eventually more interpretable output. The experimentation of SCSC on multidimensional UCI benchmark dataset point outs that the SFSC presented promising result for the clustering applications on multidimensional datasets with respect to clustering schema quality and validation indices.

Furthermore, a novel approach which also uses SFSC for churn analysis is also proposed. In this approach, churn analysis is done in fully unsupervised way by clustering and dimension reduction contrarily to other works in the literature. Since every customer churn data set has high dimensional feature space, we have done clustering by using dynamic feature selection for different clusters. Thus, high dimensional feature space is handled by our novel simultaneous clustering and dynamic feature selection methodology - SFSC. It has no need for preliminary number of clusters in the data set. Features are selected dynamically for different clusters. Clustering and variable selection is done at the same step. In summary, proposed approach and algorithm generate a churn map. A case study (churn analysis) on a telecom churn data set is studied as a real case application of the SFSC and the proposed approach. Churn map generation present beneficial descriptive output for true CRM management as managerial insight, since it explains the principal features for each customer segment and churn probability for each customer. Thus retention activities on customers can be done much more accurately and efficiently by using the churn map.

As for the future research directions of this thesis, they are in the following paragraphs:

For ENM, usage of other divergence methodologies (if possible, generalized forms as statistical distances) which are bounded in between [0,1] are planned to be discussed. Furthermore, other reinforcement learning methodologies or approaches like policy iteration other than q-learning for merging similar compact regions are thought to enhance the ENM. Large scale 3d dataset applications or real-life cases of the ENM will be other interesting future research issues.

For SCSC, a new algorithmic design not by using KNN is planned. Thus, SFSC will not be dependent on KNN architecture and K-Means feature clustering. Furthermore,

integration ENM and SFSC can be another future extension issue. Moreover, SFSC can be extended to a new structure which is capable of producing 3d clustering output.

REFERENCES

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. *Proceedings of ACM sigmod international conference on management of data*, *27*, pp. 94–105.

Ahn, J.-H., Han, S.-P., & Lee, Y.-S. (2006). Customer churn analysis: Churn determinants and mediation effects of partial defection in the Korean mobile telecommunications service industry. *Telecommunications Policy, 30*(10-11), 552-568.

Alakuş, C. (2018). *Neighborhood construction-based multi-objective evolutionary clustering algorithm with feature selection.* Ankara: Master Thesis, Middle East Technical University, Graduate School of Natural and Applied Sciences.

Almana, A., Aksoy, M., & Alzahrani, R. (2014). A survey on data mining techniques in customer churn analysis for telecom industry. *International Journal of Engineering Research and Applications, 4*(5), 165-171.

Ankerst, M., Breunig, M., Kriegel, H., & Sander, J. (1999). OPTICS: ordering points to identify the clustering structure. *Proceedings of ACM SIGMOD International Conference on Management of Data*, *28*, pp. 49–60.

Ansari, A., Noorzad, A., & Zaf, H. (2009). Clustering analysis of the seismic catalog of Iran. *Computers & Geosciences, 35*(3), 475-486. doi:https://doi.org/10.1016/j.cageo.2008.01.010.

Barbakh, W., & Fyfe, C. (2007). Clustering with Reinforcement Learning. *IIDEAL 2007, Lecture Notes in Computer Science* (pp. 48-81). Berlin, Heidelberg: Barbakh W., Yin H., Tino P., Corchado E., Byrne W., Yao X. (eds): Intelligent Data Engineering and Automated Learning.

Barbara, D., & Chen, P. (2000). Using the fractal dimension to cluster datasets. *Proceedings of the sixth ACM SIGMOD international conference on knowledge discovery and data mining*, (pp. 260–264).

Bellman, R., & Rand Corporation. (1957). *Dynamic Programming*. Princeton University Press.

Ben-Hur, A., Horn, D., Siegelman, H., & Vapnik, V. (2002). Support vector clustering. *J. Mach. Learn. Res. , 2*, 125–137.

Berkhin, P. (2002). A Survey of clustering data mining techniques. *Recent Advances in Clustering, 10*.

Bezdek , J., Ehrlich , R., & Full , W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences, 10*(2-3), 191-203.

Bi, W., Cai, M., Liu, M., & Li, G. (2016). A big data clustering algorithm for mitigating the risk of customer churn. *IEEE Transactions on Industrial Informatics, 12*(3), 1270-1281.

Bordogna, G., & Dino, I. (2014). Fuzzy core DBSCAN clustering algorithm,. *Communications in Computer and Information Science, 444*, 100-109.

Bose, I., & Chen, X. (2015). Detecting the migration of mobile service customers using fuzzy clustering. *Information Management, 52*, 227–238. doi:https://doi.org/10.1016/j.im.2014.11.00

Brubaker, S., & Vempala, S. (2018). Isotropic PCA and Affine-Invariant Clustering. *49th Annual IEEE Symposium on Foundations of Computer Science*, (pp. 551-560). Philadelphia, PA,.

Cai, J., Wei, H., Yang, H., & Zhao, X. (2020). A Novel Clustering Algorithm Based on DPC and PSO. *IEEE Access, 8*, 88200-88214. doi:10.1109/ACCESS.2020.2992903

Celik, O., & Osmanoglu, U. (2019). Comparing to Techniques Used in Customer Churn Analysis. *Journal of Multidisciplinary Developments, 4*(1), 30-38.

Chattopadhyay, S., Pratihar, D., & De Sarkar, S. (2011). A comparative study of fuzzy c-means algorithm and entropy-based fuzzy clustering algorithms. *Computing and Informatics, 30*, 701–720.

Cheng, D., Zhu, Q., Huang, J., Wu, Q., & Yang, L. (2021). Clustering with Local Density Peaks-Based Minimum Spanning Tree. *IEEE Transactions on Knowledge and Data Engineering, 33*(2), 374-387. doi:10.1109/TKDE.2019.2930056

Cheng, M., Ma, T., Ma, L., Yuan, J., & Yan, Q. (2022). Adaptive grid-based forest-like clustering algorithm. *Neurocomputing, 481*, 168-181. doi:https://doi.org/10.1016/j.neucom.2022.01.089.

Dahiya, K., & Bhatia, S. (2015). Customer churn analysis in telecom industry. *4th (IEEE) International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions)*, (pp. 1-6).

der Merwe, D., & Engelbrecht, A. (2003). Data clustering using particle swarm optimization. *The 2003 Congress on Evolutionary Computation CEC '03*, *1*, pp. 215–220.

Di Giuseppe, M., Troiano, A., Troise, C., & De Natale, G. (2014). k-Means clustering as tool for multivariate geophysical data analysis. An application to shallow fault zone imaging. *Journal of Applied Geophysics, 101*, 108-115. doi:https://doi.org/10.1016/j.jappgeo.2013.12.004.

Ester, M., Kriegel, K., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, (s. 226-231). Portland, Oregon, USA.

Estivil-castro, V., & Lee, I. (2000). Amoeba: Hierarchical clustering based on spatial proximity using delaunay diagram. *Proceedings of the 9th International Symposium on Spatial Data Handling*, (pp. 7-26).

Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning, 2*, 139–172.

Fowlkes, E., & Mallows, C. (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association, 78*, 553–569. Retrieved from http://www.jstor.org/stable/2288117

Fukunaga, K., & Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory, 21* (1), 32–40.

García, D., Nebot, À., & Vellido, A. (2017). Intelligent data analysis approachesto churn as a business problem: a survey. *Knowledge and InformationSystems, 51*, 719–774.

Ge, Y., Xiong, J., & Brown, D. (2017). Customer churn analysis for a software-as-a-service company. *Systems and Information Engineering Design Symposium (SIEDS)*, (pp. 106-111).

Georgoulas, G., Konstantaras, A., Katsifarakis, E., Stylios, C., & Maravelakis, G. (2013). Seismic-mass" density-based algorithm for spatio-temporal clustering. *Expert Systems with Applications, 40*(10), 4183-4189. doi:https://doi.org/10.1016/j.eswa.2013.01.028.

Guha, S., Rastogi, R., & Shim, K. (1998). CURE: an efficient clustering algorithm for large databases. *ACM SIGMOD Rec.*, *27*, s. 73–84.

Guha, S., Rastogi, R., & Shim, K. (1999). Rock: a robust clustering algorithm for categorical attributes. *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*, (s. 512-521). doi:10.1109/ICDE.1999.754967

Gvishiani, A., Dobrovolsky, M., Agayan, S., & Dzeboev, B. (2013). FUZZY-BASED CLUSTERING OF EPICENTERS AND STRONG EARTHQUAKE-PRONE AREAS. *Environmental Engineering & Management Journal (EEMJ), 12*(1).

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems, 17*(2001), 107-145.

Harris, C., & Kaiser, H. (1964). Oblique Factor Analytic Solutions by Orthogonal Transformation. *Psychometrika, 32*, 363–379.

Hashmi, N., Butt, N., & Iqbal, M. (2013). Customer churn prediction in telecommunication a decade review and classification. *International Journal of Computer Science Issues (IJCSI), 10*(5), 271.

Huang, B., Kechadi, M., & Buckley, B. (2012). Customer churn prediction in telecommunications. *Expert Systems with Applications, 39*(1), 1414-1425.

Huang, Y., & Kechadi, T. (2013). An effective hybrid learning system fortelecommunication churn prediction. *Expert Systems with Applications, 40*, 5635–5647.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*, 193–218.

İnkaya, T. (2011). *A methodology of swarm intelligence application in clustering based on neighborhood construction.* Ankara: Phd Dissertation, Middle East Technical University, Graduate School of Natural and Applied Sciences.

İnkaya, T. (2015). A parameter-free similarity graph for spectral clustering. *Expert Systems With Applications, 42*, 9489–9498.

İnkaya, T., Kayalıgil , S., & Özdemirel, N. (2015). An adaptive neighbourhood construction algorithm based on density and connectivity. *Pattern Recognition Letters, 52*, 17–24. doi:https://doi.org/10.1016/j.patrec.2014.09.007

Jain, A., Murty, M., & Flaynn, P. (1999). Data clustering: a review. *ACM Computer Survey (CSUR), 31*, 264–323.

Jonker, J., Piersma, N., & Van den Poel, D. (2004). Joint optimization ofcustomer segmentation and marketing policy to maximize long-termprofitability. *Expert Systems with Applications, 27*, 159–168.

Kaelbling, L., Littman, M., & Moore, A. (1996). Reinforcement learning: A survey. *J. Artif. Int. Res. , 4*, 237-285.

Karaboğa, D., & Akay, B. (2009). A survey: algorithms simulating bee swarm intelligence. *Artifical Intelligence Review, 31*, 61–85.

Karaboğa, D., & Öztürk, C. (2011). A novel clustering approach:Artificial bee colony (abc) algorithm. *Applied Soft Computing, 11*, 652–657. doi:https://doi.org/10.1016/j.asoc.2009.12.025

Karri, N., Ansari, M., & Pathak, A. (2018). Identification of Seismic Zones of India using DBSCAN. *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, (pp. 65-69). Greater Noida, India. doi:10.1109/GUCON.2018.8674964.

Kaya, A. (2018). *Bulanık Kümeleme Analizinde Bulanık Kümeleme Algoritmalarının Karşılaştırılması.* Eskişehir: Master Thesis, Anadolu University, Graduate School of Natural and Applied Sciences.

Kazaz, N. (2019). *Veri Madenciliğinde Kümeleme Analizi Yöntemlerinin İncelenmesi Ve Sağlık Bilimleri Alanındaki Uygulamaları.* İstanbul: Master Thesis, İstanbul University, Institute of Health Sciences.

Kohonen, T. (1990). The self-organizing map. *Proceedings of IEEE, 78*, 1464–1480.

Konstantaras, A. (2020). Deep Learning and Parallel Processing Spatio-Temporal Clustering Unveil New Ionian Distinct Seismic Zone. *Informatics, 7*(4), 39. doi:https://doi.org/10.3390/informatics7040039

Kriegel, H., Kröger, P., & Zimek, A. (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transaction Knowledge Discovery Data 3, 1*(1), 1-58.

Kullback, S., & Leibler, R. (1951). On information and sufficiency. *The Annals of Mathematical Statistics, 22*, 79–86. Retrieved from http://www.jstor.org/stable/2236703

Likas, A. (1999). A Reinforcement Learning Approach to Online Clustering. *Neural Computation, 11*(8), 1915-1932.

Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory, 37*, 145-151. doi:10.1109/18.61115

Lumer, E., & Faieta, B. (1994). Diversity and adaptation in populations of clustering ants. *Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3: From Animals to Animats 3* (pp. 501-508). Cambridge, MA, USA: MIT Press.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations,. *Proceedings of Fifth Berkeley Symp. Math. Stat. Probab*, *1*, s. 281–297.

Maulik, U., & Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern Recognition, 33*, 1455–1465. doi:https://doi.org/10.1016/S0031-3203(99)00137-5.

Mitchell, T. (1997). *Machine Learning,*. New York NY.: McGraw-Hill.

Morales-Esteban, A., Martínez-Álvarez, F., Scitovski, S., & Scitovski, R. (2014). A fast partitioning algorithm using adaptive Mahalanobis clustering with application to seismic zoning. *Computers & Geosciences, 73*, 132-141. doi:https://doi.org/10.1016/j.cageo.2014.09.003.

Nasibov, E., & Ulutagay, G. (2009). Robustness of density-based clustering methods with various neighborhood relations. *Fuzzy Sets and Systems, 160*(24), 3601-3615.

Ng, A., Jordan, M., & Weiss, Y. (2002). On spectral clustering: analysis and an algorithm. *Adv. Neural Inf. Process Syst., 2*, 849–856.

Ngai, E., Xiu, L., & Chau, D. (2009). Application of data miningtechniques in customer relationship management: A literature review and classification. *Expert Systems with Applications, 36*, 2592–2602. Retrieved from https://www.sciencedirect.com/science/article/pii/S09S0957417408001243, doi:https://doi.org/10.1016/j.eswa.2008.02.021.

Park, H., & Jun, C. (2009). A simple and fast algorithm for K-medoids clustering. *Expert Systems and Applications, 36*, 3336–3341.

Parmar, M. (2022). *Clustering dataset on github milaan*. Clustering dataset on github milaan: https://github.com/milaan9/Clustering-Datasets adresinden alındı

Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine, 2*(11), 559–572.

Pluim, J., Maintz, J., & Viergever, M. (2003). Mutual-information-based registration of medical images: a survey. *IEEE transactions on medical imaging, 22*(8), 986-1004.

Pourbahrami, S., & Khanli, L. (2018). *A survey of neighbourhood construction algorithms for clustering and classifying data points.*

Pourbahrami, S., Khanli, L., & Azimpour, S. (2019). A novel and efficient data point neighborhood construction algorithm based on apollonius circle. Retrieved from ArXiv abs/1809.10702.

Qin, Y., Yu, Z., Wang, C., Gu, Z., & Li, Y. (2018). A Novel clustering method based on hybrid K-nearest-neighbor graph. *Pattern Recognition, 74*, 1–14. doi:10.1016/j.patcog.2017.09.008

Rajamohamed, R., & Monokaran, J. (2018). Improved credit card churn prediction based on rough clustering and supervised learning techniques. *Cluster Computing, 21*(1), 65-77.

Rasmussen, C. (1999). The infinite gaussian mixture model. *Adv. Neural Inf. Process Syst., 12*, 554–560.

Rosenberg, A., & Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. *2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* (pp. 410-420). Prague, Czech Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D07-1043

Rudelson, M. (1999). Random Vectors in the Isotropic Position. *Journal of Functional Analysis, 164*(1), 60-72.

Scitovski, S. (2018). A density-based clustering algorithm for earthquake zoning. *Computers & Geosciences, 110*, 90-95. doi:https://doi.org/10.1016/j.cageo.2017.08.014.

Shaaban, E., Helmy, Y., Khedr, A., & Nasr, M. (2012). A Proposed Churn Prediction Model. *International Journal of Engineering Research and Applications, 2*(4), 693-697.

Shannon, C. (1948). Shannon, C.E., 1948. A mathematical theory of communication. *The Bell System Technical Journal, 27*, 379–423. doi:10.1002/j.1538-7305.1948.tb01338.x.

Sharan, R., & Shamir, R. (2000). CLICK: a clustering algorithm with applications to gene expression analysis. *Proceedings of international conference intelligent systems molecular biolgy*, (pp. 307-316).

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*, 888-905. doi:10.1109/34.868688

Shin, H., & Sohn, S. (2004). Segmentation of stock trading customersaccording to potential value. *Expert Systems with Applications, 27*, 27–33. doi:https://doi.org/10.1016/j.eswa.2003.12

Song, C., Pei, T., Ma, T., Du, Y., Shu, H., Guo, S., & Fan, Z. (2019). Detecting arbitrarily shaped clusters in origin-destination flows using ant colony optimization. *International Journal of Geographical Information Science, 33*(1), 134-154. doi:DOI: 10.1080/13658816.2018.1516287

Ünver, M., & Erginel, N. (2020). Intuitionistic fuzzy density based spatial clustering of applications with noise: IFDBSCAN. *Proceedings of International Conference of Intelligent and Fuzzy Systems*, (pp. 56-64). İstanbul, Turkey.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research, 9*(11).

Velliangiria, S., Alagumuthukrishnanb, S., & Thankumar, S. (2019). A review of dimensionality reduction techniques for efficient computation. *International Conference On Recent Trends In Advanced Computing.*

Wang, W., Yang, J., & Muntz, R. (1997). STING: a statistical information grid approach to spatial data mining. *VLDB*, 186–195.

Watkins, C. (1989). *Learning from Delayed Rewards.* Phd Dissertation, Cambridge University.

Weinstein, M., & Horn, D. (2009). Dynamic quantum clustering: a method for visual exploration of structures in data. *Physical Review E, 80*, 66-117.

Wu, L., & Lin, Z. (2005). Research on customer segmentation model by clustering. *Proceedings of the 7th International Conference on Electronic Commerce, Association for Computing Machinery,*, (pp. 316–318). New York, NY, USA. doi:316–318

Xu, D., & Tian, Y. (2015). A Comprehensive survey of clustering algorithms. *Annals of Data Science, 2*(2), 165-193.

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks, 16*, 645–678. doi:10.1109/TNN.2005.845141

Xu, X., Ester, M., Kriegel, H., & Sander , J. (1998). A distribution-based clustering algorithm for mining in large spatial databases. *Proceedings of the fourteenth international conference on data engineering*, (pp. 324-331).

Yang, C., Shi, X., Jie, L., & Han, J. (2018). I know you'll be back: Interpretable new user clustering and churn prediction on a mobile social application. *24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, (pp. 914-922).

Yang, M.-S., & Sinega, K. (2019). A Feature-Reduction Multi-View k-Means Clustering Algorithm. *IEEE Access, 7*, 114472-114486. doi:10.1109/ACCESS.2019.2934179

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. *ACM SIGMOD Rec.*, *25*, s. 103–104.

http-1: https://en.wikipedia.org/wiki/Machine_learning (Date of access: 01.08.2022)

http-2: https://en.wikipedia.org/wiki/Curse_of_dimensionality (Date of access: 01.08.2022)

http-3: http://cs.joensuu.fi/sipu/datasets/ (Date of access: 15.03.2020)

http-4: https://github.com/deric/clustering-benchmark (Date of access: 15.03.2020)

http-5: https://github.com/milaan9/Clustering-Datasets. (Date of access: 15.03.2020)

http-6: https://archive.ics.uci.edu/ml/datasets.php (Date of access: 15.10.2020)

http-7: https://github.com/microbhai/CustomerChurnAnalysis/ (Date of access: 15.10.2022)

http-8: http://www.koeri.boun.edu.tr/sismo/zeqdb/ (Date of access: 01.20.2023)

http-9a: http://yerbilimleri.mta.gov.tr/anasayfa.aspx (Date of access: 01.20.2023)

http-9b: https://www.afad.gov.tr/turkiye-deprem-tehlike-haritasi (Date of access: 01.20.2023)

APPENDIX

APPENDIX-1

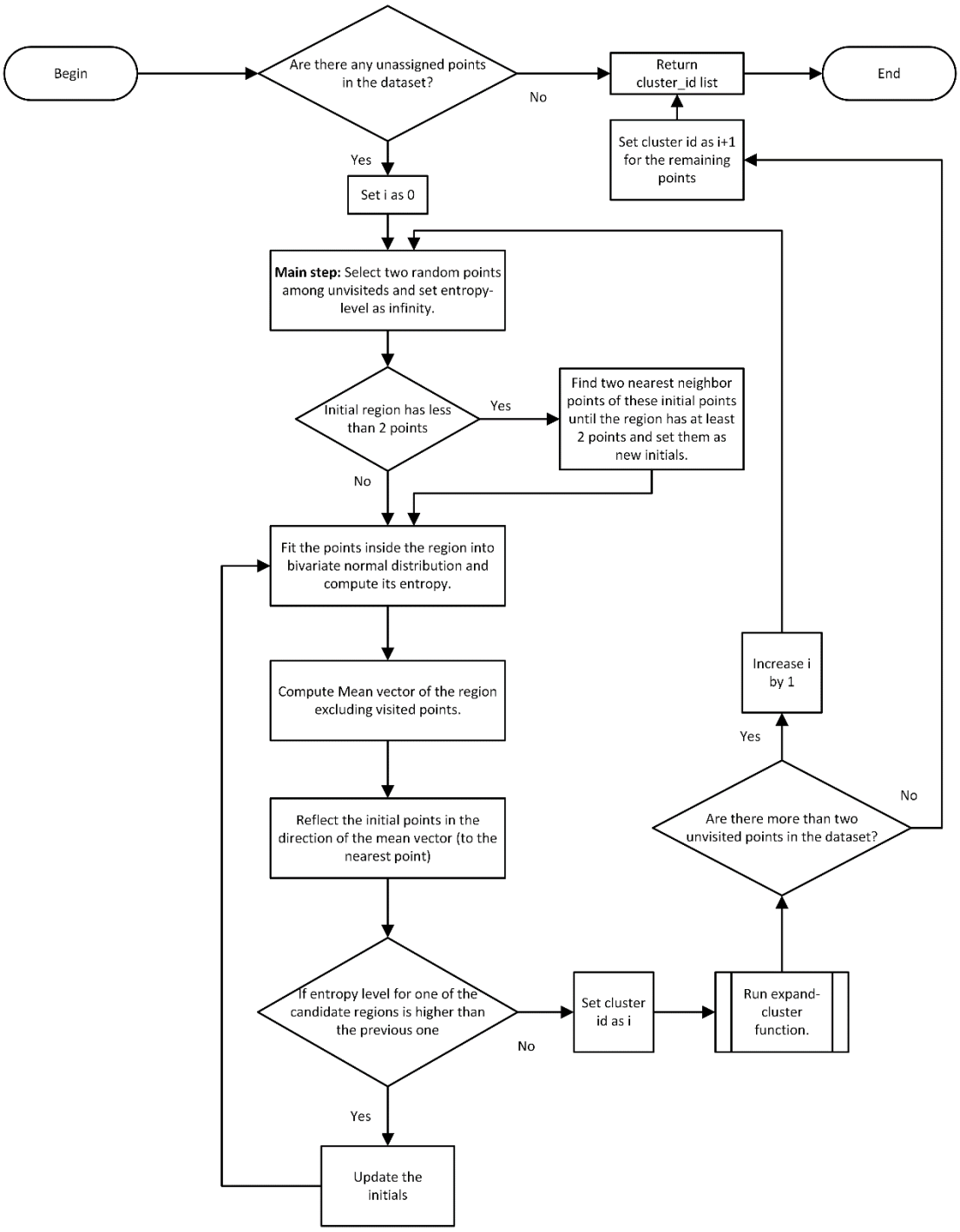

Figure A.1.1. *Flowchart for the main function of ENM*

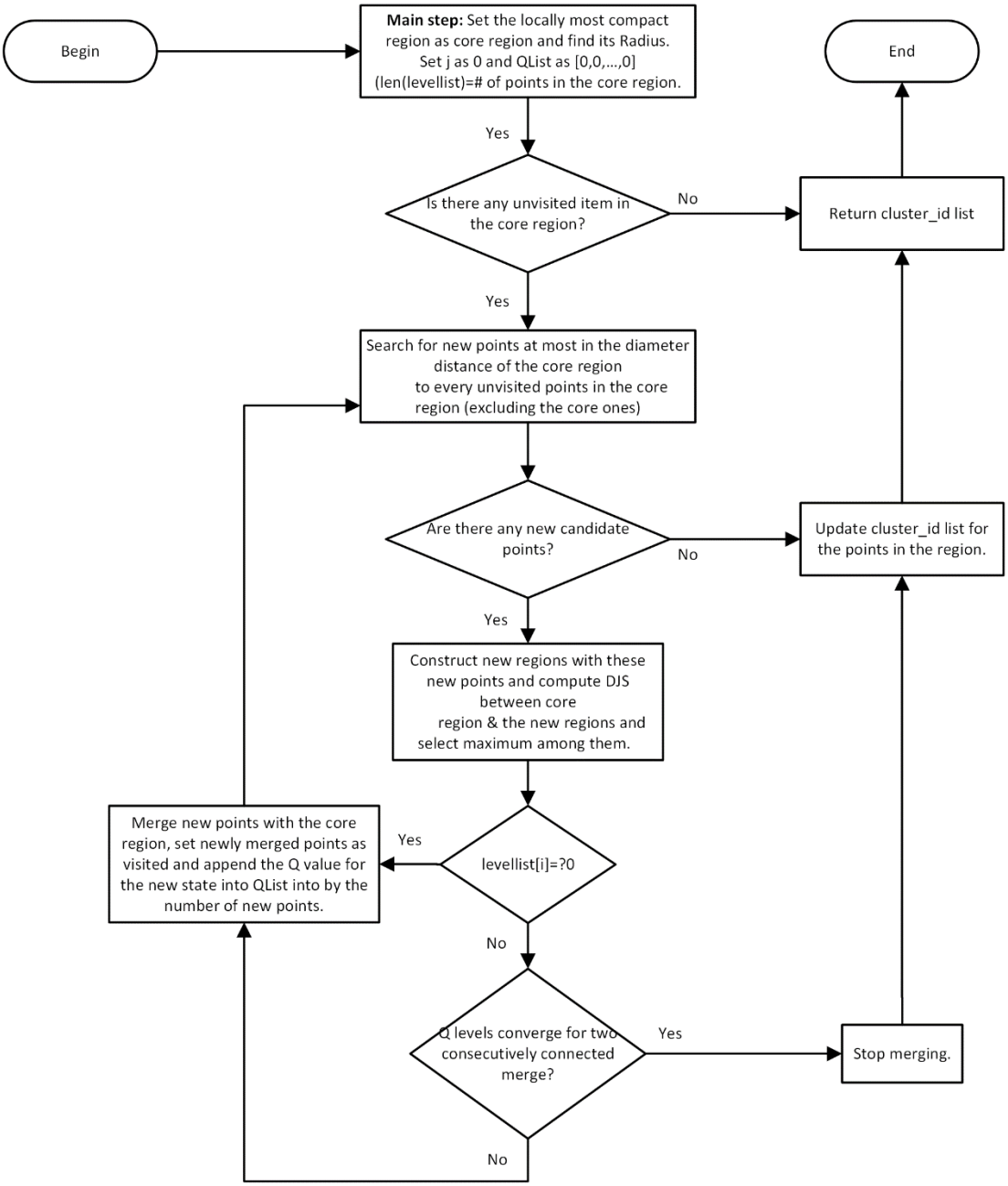

Figure A.1.2. *Flowchart for the expand_cluster function of ENM*

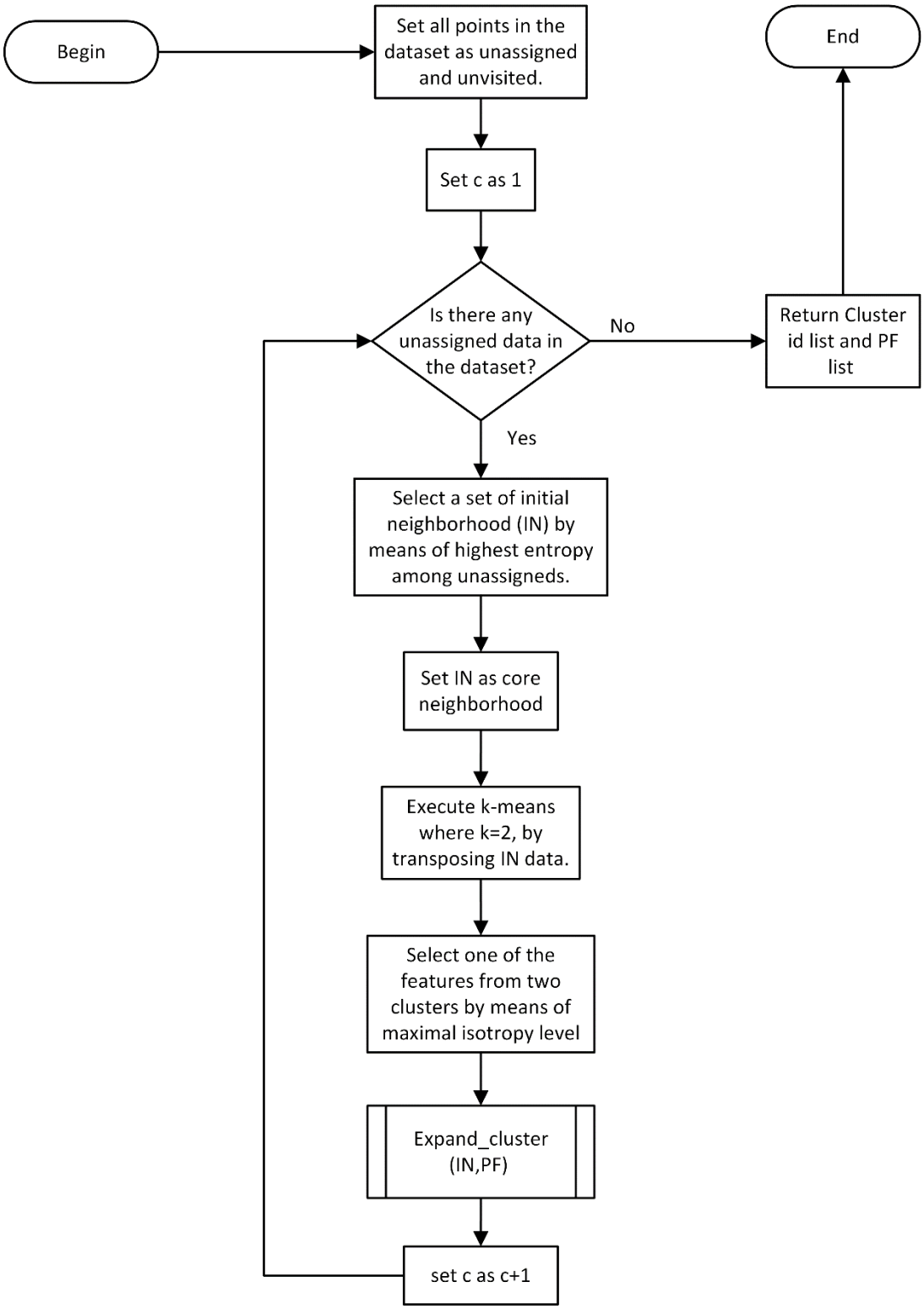

Figure A.1.3. *Flowchart of SFSC Main Function*

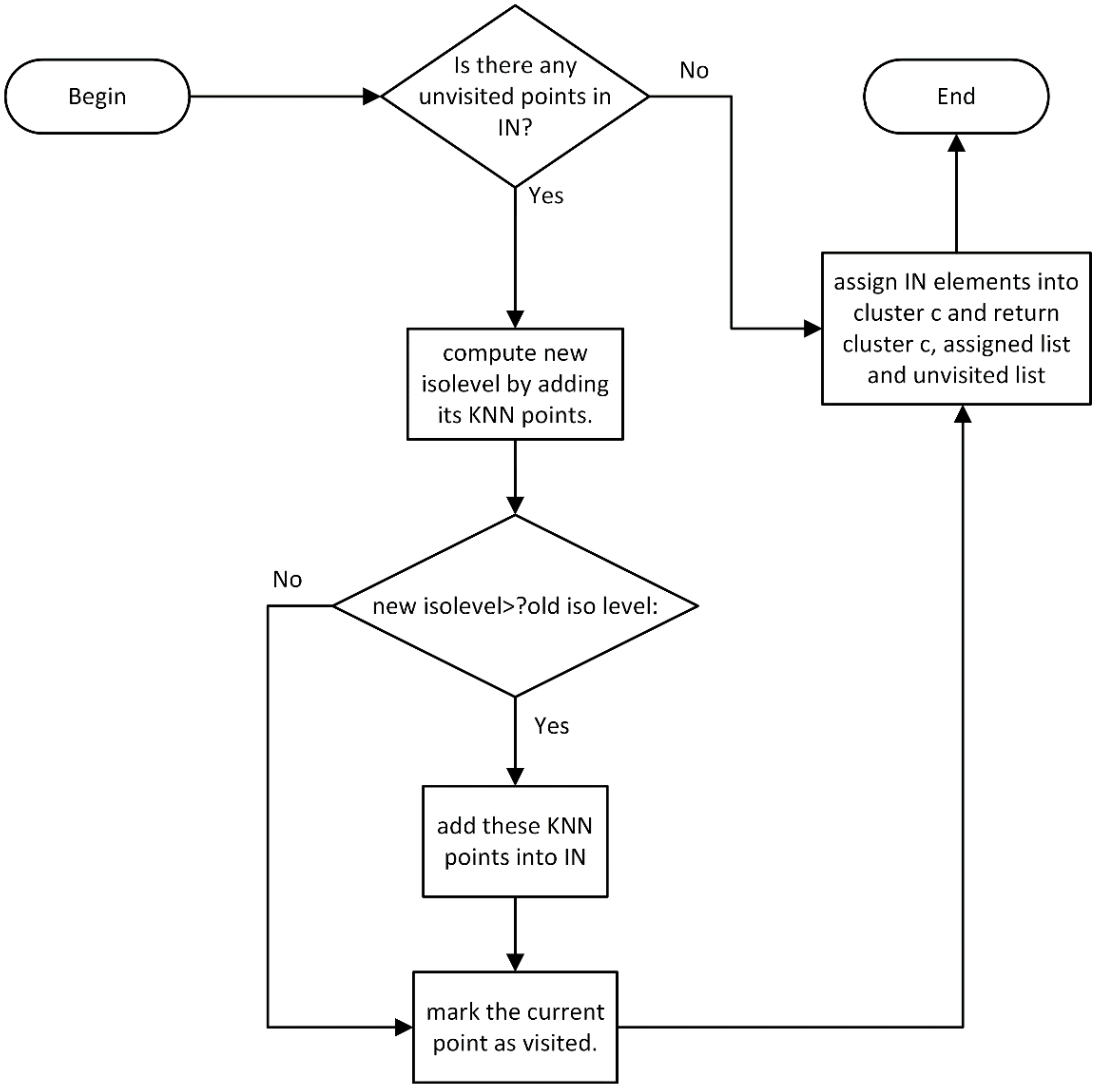

Figure A.1.4. *Flowchart of SFSC – Expand Cluster Function*

APPENDIX-2

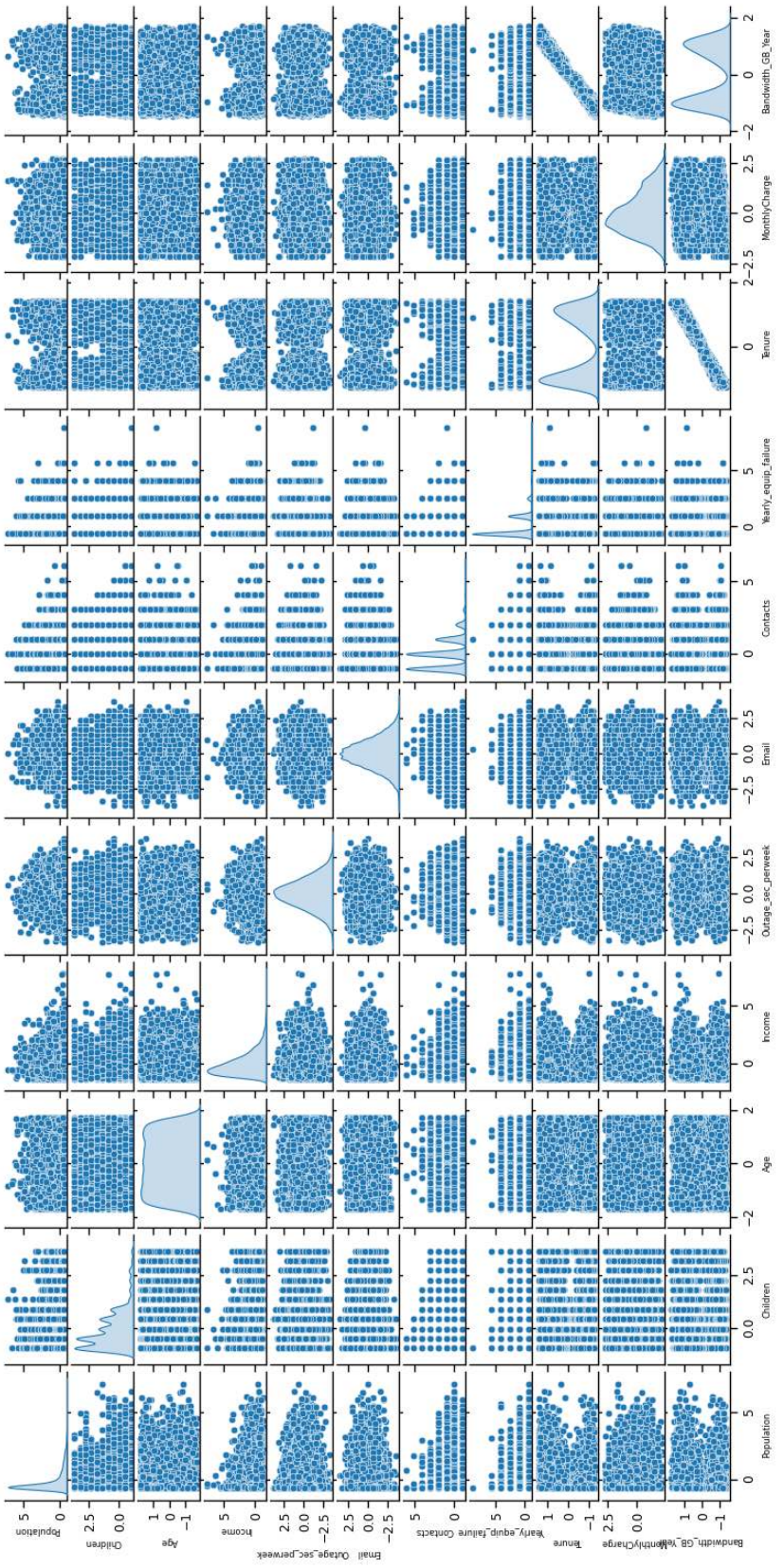

Figure A.2.1. *Projected scatter plot of the churn data set*

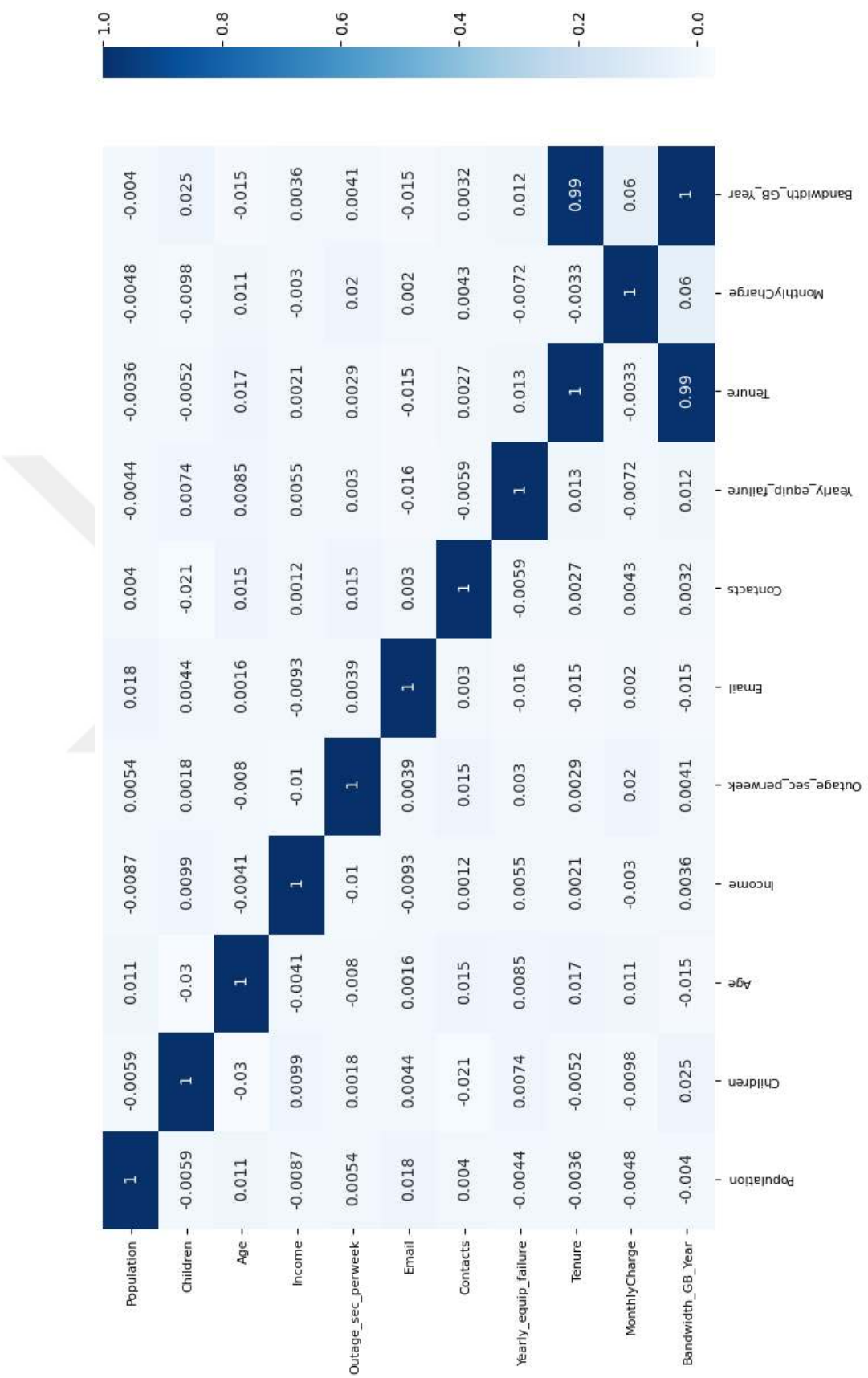

Figure A.2.2. *Correlation between variables in a heat map*

CURRICULUM VITAE

ORCID ID: 0000-0002-3988-0231

Name Surname **:** Mustafa ÜNVER

Foreign Language **:** English

Education and Professional Background:

- 2011-2012, Assistant Specialist of Scientific Programs, TUBİTAK, ARDEB
- 2017-2023, Eskişehir Technical University, Institute of Graduate Science, Industrial Engineering (Integrated Ph. D.)
- 2020-Present, Researcher, Gebze Technical University, Industrial Engineering Department

Publications and/or Scientific/Artistic Activities:

- Ünver, M., Erginel, N., 2020, Clustering applications of IFDBSCAN algorithm with comparative analysis, Journal of Intelligent and Fuzzy Systems, 39(5), 6099-6108.
- Ünver, M., Erginel, N., 2019, Intuitionistic Fuzzy Density Based Spatial Clustering of Applications with Noise: IFDBSCAN,International Conference on Intelligent and Fuzzy Systems (INFUS-2019),Advances in Intelligent Systems and Computing, Vol 1029, pp. 56-64, İSTANBUL.
- Ünver, M., Erginel, N., 2021, A Neighborhood Merging Policy Based on Shannon's Entropy and Symmetric Relative Entropy for Non-convex Cluster Detection, International Conference on Intelligent and Fuzzy Systems (INFUS-2021), Lecture Notes in Networks and Systems, 307(1), pp. 44-51.
- Ünver, M., 2021, Simultaneous Clustering and Dynamic Feature Selection Methodology Based on Dimensional Isotropy Preservation, 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), pp. 1-6, doi: 10.1109/ICECET52533.2021.9698560.

Awards:

- TÜBİTAK, 2211-A - National PhD Scholarship

Professional Union/Association/Organization Memberships:

- -