DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

ADVANCED STATISTICAL METHODS FOR INTELLIGENT TRANSPORTATION SYSTEMS

by
Büşra GÜNGÖR

July, 2022
İZMİR

ADVANCED STATISTICAL METHODS FOR INTELLIGENT TRANSPORTATION SYSTEMS

A Thesis Submitted to the
Graduate School of Natural And Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Statistics

by
Büşra GÜNGÖR

July, 2022
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**ADVANCED STATISTICAL METHODS FOR INTELLIGENT TRANSPORTATION SYSTEMS**" completed by **BÜŞRA GÜNGÖR** under supervision of **PROF. DR. SELMA GÜRLER** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

.....

Prof. Dr. Selma GÜRLER

Supervisor

.....	.....
Prof. Dr. Burcu HÜDAVERDİ	Prof. Dr. Serhan TANYEL
Thesis Committee Member	Thesis Committee Member
.....	.....
Prof. Dr. Gözde ULUTAGAY	Assoc.Prof. Dr. Sevcan DEMİR ATALAY
Examining Committee Member	Examining Committee Member

Prof.Dr. Okan FISTIKOĞLU
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my doctoral thesis advisor, Prof. Dr. Selma GÜRLER, for all her efforts during the whole period of my dissertation. For me, she is not just a thesis advisor or an academic coach, she is someone who touched my whole life. In addition to sharing her academic knowledge, she has made the greatest contribution to completing my thesis by lifting me up every time I fall and ensuring not to lose my motivation. I am also grateful to her for the valuable advices she shared not only for my thesis but also for other areas of my life, and for all the values she taught. I feel honored and privileged to be a student of her.

I would like to thank Prof. Dr. Burcu HÜDAVERDİ, a member of the thesis committee, for her comments and contributions that helped me develop my thesis. I would also like to thank Prof. Dr. Serhan TANYEL, a member of the thesis committee, for his contributions and knowledge transfer in the field of intelligent transportation systems.

I would like to thank to Higher Education Council of Turkey (YÖK) 100/2000 Ph.D. scholarship programme and the Scientific and Technological Research Council of Turkey (TÜBİTAK) 2211/A fellowship programme for supporting this thesis. Moreover, thanks to Turkey Directorate General of Security for sharing with me the data set used in the application chapter of my thesis. Another thanks for the Seeds data set, which is used in this thesis, to the Institute of Agrophysics of the Polish Academy of Sciences in Lublin.

I would like to thank to my academic sister and dear friend Melek ESEMEN for all her supports whenever I needed it. I would also like to thank to all my dear friends, especially Yeliz SAVUT, Elif YILMAZ and Can YURDAKUL, who were with me whenever I needed them.

I also owe thanks to my parents, Lüveyza & Azmi SEVİNÇ, and my sister Ayşe Nur SEVİNÇ for their love and endless support through the years and for their confidence in me.

Finally, I would like to express my deepest thanks to my dear husband Serdar GÜNGÖR for his endless patience during this difficult process. His support, love and belief in me gave me the strength to succeed.

Büşra GÜNGÖR

ADVANCED STATISTICAL METHODS FOR INTELLIGENT TRANSPORTATION SYSTEMS

ABSTRACT

The use of machine learning techniques and statistical methods for intelligent transportation systems has gained importance recently. In this thesis, two different methods are proposed for clustering and modeling that can serve the needs of the transportation field.

Clustering methods are used to group data points quickly and easily for further analysis of clusters. Also, most clustering methods require complex computations or have an iterative procedure that makes the algorithm time-consuming, especially when the data are relatively large. In this thesis, we propose a new clustering method called spatial adaptive clustering (SAC) based on the idea of adaptive cluster sampling (ACS) design. ACS is a sampling method, which is based on neighborhood search on a grid structure, has an adaptive selection process of units and recursively added units reveal the batched individuals easily and quickly. The SAC algorithm forms clusters based on neighborhood search using grid structures and can detect noise points. The performance of the proposed algorithm is evaluated through comparisons with the results from well-known density-based clustering approaches in the literature using real and artificial data sets. Also, hot spots of accident locations in Birmingham/England and Buca/Turkey are investigated using the SAC algorithm. Additionally, to reduce the number of accidents, hotspot locations are examined in terms of the factors causing the accident.

Vehicle headway modeling has also been one of the other important topics for traffic signal optimization and flow modeling. In this thesis, we estimate the parameters of Exponentiated Weibull (EW) distribution using the maximum likelihood method under the assumption that all parameters are unknown. We deal with the performance of ranked set sampling and simple random sampling methods by a simulation study in R-software in terms of mean-squared error. Finally, we illustrate the flexibility and

usefulness of EW distribution by analyzing simulated data from a real application study in the transportation field.

Keywords: Intelligent transportation systems, Clustering methods, Hotspot detection, Vehicle headway modeling, Adaptive cluster sampling, Ranked set sampling, Exponentiated Weibull distribution, Parameter estimation.

AKILLI ULAŞIM SİSTEMLERİ İÇİN İLERİ İSTATİSTİKSEL YÖNTEMLER

ÖZ

Akıllı ulaşım sistemleri için makine öğrenmesi teknikleri ve istatistiksel yöntemlerin kullanımı son zamanlarda önem kazanmıştır. Bu tezde, kümeleme ve modelleme için ulaştırma alanının ihtiyaçlarına hizmet edebilecek iki farklı yöntem önerilmiştir.

Kümeleme yöntemleri, kümelerin ileri analizi için veri noktalarını hızlı ve kolay bir şekilde gruplamak amacıyla kullanılmaktadır. Çoğu kümeleme yöntemi karmaşık hesaplamalara ihtiyaç duymaktadır. Ayrıca, yöntemler özellikle veri hacmi büyük olduğunda algoritmayı zaman alıcı hale getiren yinelemeli bir prosedüre sahiptir. Bu tezde, uyarlanabilir küme örneklemesi (ACS) tasarımı fikrine dayanan uzamsal uyarlamalı kümeleme (SAC) adı verilen yeni bir kümeleme yöntemi önerilmektedir. Izgara yapısı üzerinde komşuluk aramasına dayanan bir örnekleme yöntemi olan ACS, uyarlanabilir birim seçim sürecine sahiptir ve yinelemeli olarak eklenen birimler toplu haldeki birimleri kolay ve hızlı bir şekilde ortaya çıkarmaktadır. SAC algoritması, ızgara yapılarını ve komşuluk aramasını kullanarak kümeler oluşturmakta ve gürültü noktaları tespit edebilmektedir. Önerilen algoritmanın performansı, gerçek ve yapay veri setleri kullanılarak literatürde iyi bilinen yoğunluk tabanlı kümeleme yöntemleri ile karşılaştırılarak değerlendirilmiştir. Ayrıca, SAC algoritması kullanılarak Birmingham/İngiltere ve Buca/Türkiye'deki kaza yerlerinin sıcak noktaları tespit edilmiştir. Ek olarak, kaza sayısını azaltmak amacıyla sıcak nokta lokasyonlarında kazaya neden olan faktörler incelenmiştir.

Araç takip aralığı modellemesi, trafik sinyali optimizasyonu ve akış modellemesi için önemli konulardan biridir. Bu tezde, tüm parametrelerin bilinmediği varsayımı altında, maksimum olabilirlik yöntemini kullanarak Üstel Weibull (EW) dağılımının parametreleri tahmin edilmektedir. Sıralı küme örneklemesi ve basit rastgele örnekleme yöntemlerinin performansı, R-yazılımında bir simülasyon çalışması ile ortalama hata kareler açısından ele alınmaktadır. Son olarak, ulaşım alanındaki gerçek bir uygulama çalışmasından elde edilen simüle edilmiş verileri analiz ederek

EW dağılımının esnekliği ve kullanışlılığı gösterilmiştir.

Anahtar kelimeler: Akıllı ulaşım sistemleri, Kümeleme yöntemleri, Sıcak nokta tespiti, Araç takip aralığı modelleme, Uyarlanabilir küme örneklemesi, Sıralı küme örneklemesi, Üstel Weibull dağılımı, Parametre kestirimi.

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF SYMBOLS

k	: Number of cluster
MinPts	: Minimum number of points required
ϵ	: Radius of a neighbourhood
dist(p,q)	: Distance function for two points p and q
D	: Data set
$N_\epsilon(p)$	: The ϵ-neighborhood of a point p
$\mathbb{Z}^+$	: Set of positive integers
B	: Cluster
coreDist(p)	: Core-distance of a point p
reachabilityDist(p,q)	: Reachability-distance of a point p to point q
mpts	: Minimum data point number a cluster has to contain
dcore(p)	: Core distance of a point p
$d_m reach$(p,q)	: Mutual reachability distance between p and q
G_{mpts}	: Mutual reachability graph
S^2	: Number of square
n_1	: Initial sample size
C	: Condition
m	: Number of grid
N	: Number of data point
Ω	: Cluster
$\emptyset$	: Empty set
$G_{m_1,m_2,...,m_d}$	: Grid with the coordinates of $(m_1, m_2, ..., m_d)$
s	: Class number
TP	: True Positive
TN	: True Negative
FP	: False Positive
FN	: False Negative
#cluster	: Number of cluster obtained
λ	: The scale parameter of Exponentiated Weibull distribution

α, β	: The shape parameters of Exponentiated Weibull distribution
$g(y; \alpha, \beta, \lambda)$	: The pdf of Exponentiated Weibull distribution
$G(y; \alpha, \beta, \lambda)$	: The cdf of Exponentiated Weibull distribution
$r(y; \alpha, \beta, \lambda)$	: The hazard function of Exponentiated Weibull distribution
$l(\alpha, \beta, \lambda)$	: Loglikelihood function
$\hat{\alpha}_{srs}$	: Estimator of α based on simple random sampling
$\hat{\beta}_{srs}$	: Estimator of β based on simple random sampling
$\hat{\lambda}_{srs}$	: Estimator of λ based on simple random sampling
$\hat{\alpha}_{rss}$	: Estimator of α based on ranked set sampling
$\hat{\beta}_{rss}$	: Estimator of β based on ranked set sampling
$\hat{\lambda}_{rss}$	: Estimator of λ based on ranked set sampling

ABBREVIATIONS

ITS	: Intelligent Transportation System
K-S	: Kolmogorov-Smirnov
ML	: Maximum Likelihood
SAC	: Spatial Adaptive Clustering
ACS	: Adaptive Cluster Sampling
EW	: Exponential Weibull
RSS	: Ranked Set Sampling
PAM	: Partitioning Around Medoids
CLARA	: Clustering Large Applications
DBSCAN	: Density-based Spatial Clustering of Application with Noise
ADBSCAN	: Adaptive Density-based Spatial Clustering of Application with Noise
GRIDBSCAN	: Grid Density-based Spatial Clustering of Application with Noise
HDBSCAN	: Hierarchical Density-based Spatial Clustering of Application with Noise
RI	: Rand Index
ARI	: Adjusted Rand Index
HR	: Hit Rate
UK	: United Kingdom
SRS	: Simple Random Sampling
PDF	: Probability Density Function
CDF	: Cumulative Distribution Function
MSE	: Mean Squared Error

CHAPTER 1
INTRODUCTION

The Intelligent Transportation System (ITS) has been playing an important role in the global world since the last decades. It is responsible for personal mobility; provides access to services, jobs, and leisure activities. Like any other disciplines, transportation needs statistical analysis for solving problems. Some of the research fields are speed estimation, behavior analysis, estimation of trip matrices, modeling of vehicle headway data and traffic flow, analysis of transportation performance, safety analysis, hotspot detection and mobility analysis. In the next sections, clustering and modeling, which play an important role in solving some problems, will be discussed.

1.1 Clustering in ITS

The volume and availability of data in ITS gives rise to the need for data mining approaches. Data mining algorithms in ITS have a wide range of applications, including but not limited to object detection, signal recognition, traffic flow prediction, travel route planning, travel time planning, and vehicle and road safety. One of the important ITS field that uses clustering techniques is accident data analysis. The main purpose of analyzing accident data is to identify the main factors associated with a road and traffic accident. In order to recommend safe driving, a precise analysis of road traffic data is essential to discover the elements involved in accidents. In this way, the loss of life and property can be reduced to a great extent by working to eliminate or minimize the accident factors. However, the heterogeneous nature of road accident data complicates the analysis phase. Data segmentation is a widely used method to overcome this heterogeneity of accident data, and cluster analysis aids in the segmentation of road accidents.

In the literature, there are many studies used clustering techniques on accident data. Kim & Yamashita (2007) propose the use of k-Means algorithm for analyzing pedestrian crash patterns in Honolulu, Hawaii and they divided the accident locations

into 10 clusters based on their coordinates. Kumar & Toshniwal (2015) proposed to segment the road accident locations using k-modes clustering method. They used the data on road accidents that occurred in Dehradun, India between 2009 and 2014. The accident locations are segmented into 6 clusters using k-modes, and then each cluster is examined based on their characteristics. Additionally, clusters are further analyzed to find factors behind clustered accident locations using association rule mining methods.

Kumar & Toshniwal (2016) used k-means algorithm in order to divide the accident locations into 3 groups in India. The clusters are categorised based on their accident frequency counts, as low, moderate and high frequency locations. They used association rule mining methods to understand the relationship between the set of features that often occur together when an accident occurs. The rules indicate that highways are the locations with a high frequency accidents, and intersections in the marketplace area of highways are more inclined to serious accidents. Alotaibi (2018) used DBSCAN algorithm in order to cluster the accident locations in India based on their accident frequencies. The data set they used in analysis are for 6 years period from 2012 to 2016. After clustering the accident locations, they discovered the factors behind the different locations.

Maltaş et al. (2019) clustered the accident locations in Istanbul, Turkey, based on three years of data, using k-means clustering method. They divided the accident locations into 5 clusters based on their level of danger from very dangerous to very safe. Shen et al. (2019) used grid clustering and k-medoids method as a novel approach in order to segment the accident data in rural areas of China. Ganjali Khosrowshahi et al. (2021) suggested to use of GriDBSCAN, which is a grid-based DBSCAN algorithm, for identifying the accident hotspots. They detected the hotspots in Gebze and Izmit, Turkey, using the accident data between 2013 and 2014. According to the results of the study, the factors contributing to injury accidents in both cities were determined as speed limit, collision and intersection types. For more studies, one can see, Anderson (2009), Le et al. (2020), Shinde et al. (2022) and references therein.

1.2 Vehicle Headway Modeling in ITS

Vehicle headway modeling is a very important concept for intelligent transportation in traffic engineering. It is useful for the traffic signal optimization and flow modelling by characterizing the distribution of vehicles on a road. Vehicle headway is defined as the time interval between the successive arrival of vehicles in a lane and it is a fundamental topic for many traffic flow optimization study and vehicle simulation issues. Statistical modeling of vehicle headway data has received great attention along the last two decades, and there are a variety of distributions that can be used for different traffic conditions.

There are several studies in the literature for modelling and analyzing the time headway. Luttinen (1992) used histogram method for the estimation of the headway distribution. Al-Ghamdi (2001) analyzed the headway data for different traffic flow states at freeway and arterial sections in Saudi Arabia. He used chi-square test to find best fitted distribution that described the headway data. He found that negative exponential, shifted exponential and Erlang distributions are the best models to describe headway at freeway for low, middle and high flow rates, respectively. However, Gamma distribution provides best fit for the data obtained from arterial sites for all cases of traffic flow.

Jang et al. (2011) used Kolmogorov-Smirnov (K-S) test to examine the best fit model for time headway for a suburban arterial in Korea. Riccardo & Massimiliano (2012) estimated the density of headways obtained at four automatic traffic recorder sites of Venice on two-lane two-way road. They used K-S test to find the best fitted distribution, and it is found that Inverse Weibull distribution fits better than commonly used models (Log-logistic, Pearson 5, Pearson 6, Inverse Gaussian) for almost all situations of traffic flow. Caliskanelli & Tanyel (2010) applied Anderson-Darling test for determining the simple headway distribution in İzmir and used method of moment method for estimating the parameters for mixed headway distribution. Kong & Guo (2016) found out the best fit for the headway distribution in

China by using K-S test and then used maximum likelihood (ML) estimation method for building the parameter functions.

Badhrudeen et al. (2016) compared some distributions, which are commonly used in the literature, for the data obtained from Automated Sensor Data in India. The comparison is done in R-software based on the log-likelihood values and the ML estimation method is used for parameter estimation. They found that Weibull distribution is the best model to describe headway data for all category of leader-follower pairs. Rossi et al. (2017) applied Kolmogorov-Smirnov test for fitting time headway and speed distributions of bicycles in Bologna. Touhbi et al. (2018) used Pareto IV model to represent the headway data obtained in Morocco for different flow levels, and they also compared Pareto IV distribution with log-normal, shifted log-normal, exponential, shifted exponential and Pearson III distributions by performing K-S test. For more details, see Li & Chen (2017) and Roy & Saha (2018).

1.3 Motivation and Outline of the Thesis

The main aim of a clustering algorithm is grouping the data points in a fast and easy way. Most of the clustering methods need complex calculations or they have an iterative procedure which makes the algorithm time-consuming especially when the data is relatively large. In the literature, there are some studies which combine the clustering algorithms with sampling methods to improve the computational efficiency for large-scale data. But, most of the sampling-based approaches in the literature use sampling techniques for initial step adjustment of the algorithm or as a part of an existing method. Borah & Bhattacharyya (2004) used sampling technique to decrease the number of neighbourhood query operations. Ayech & Ziou (2015) proposed ranked-k-means method which uses ranked set sampling in order to obtain representative initial points for k-means algorithm. Zhao et al. (2019) combined stratified sampling method and fuzzy c-means algorithm, which is a modification of k-means. However, adapting the steps of algorithms of sampling methods to clustering algorithms is also important as it affects their simplicity and efficiency.

Furthermore, use of the algorithm of sampling methods may result in minimizing the runtime and faster running. The first aim of this thesis is to propose a new clustering method, spatial adaptive clustering (SAC). The proposed SAC method is using the main idea behind the adaptive cluster sampling (ACS) which is an effective sampling method based on adaptive selection process. Since ACS uses neighbourhood search in grid structure and recursively added units reveals the batched individuals easily and quickly, SAC has some advantages such as being easy, flexible and practical to use. Also, the SAC algorithm is able to identify the arbitrary shaped clusters and detect noise points.

The second aim of this thesis is to model vehicle headway data, which is fundamental for intelligent transportation applications in traffic engineering. It is useful for the traffic signal optimization and flow modelling. In this thesis, Exponential Weibull (EW) which is one of the best flexible models is proposed to use for characterizing the uncertainty in vehicle headway data. In addition, ranked set sampling (RSS), which is an efficient sampling method, is used to estimate unknown model parameters.

This thesis mainly focuses on statistical approaches used to solve some problems in transportation systems; clustering, estimation and modelling. It consists of seven chapters that present some well known clustering algorithms, methodology of the SAC algorithm, experimental studies for clustering, hotspot analysis with real data sets and headway modeling with EW distribution based on RSS. Chapter 2 briefly introduces DBSCAN, OPTICS and HDBSCAN algorithms which are used for numerical comparisons of clustering methods in the thesis. In Chapter 3, we present the SAC algorithm with illustration of steps for simulated data points. In Chapter 4, we discuss the results of experimental studies designed to evaluate the performance of the SAC in terms of external validation measures such as the Rand index, adjusted Rand index, purity, and computational complexity. In addition, the performance of the algorithm is compared with the well-known algorithms using different real and artificial data sets. Also, the run times of the algorithms are compared for data points of different sizes. In Chapter 5, we present an application study in the field of

intelligent transportation systems to identify the accident hotspots to reveal the factors behind the accidents by detecting the accident-heavy points. Using Birmingham/UK (32 features) and Buca/Turkey (44 features) real accident datasets, accident zones are clustered by SAC method and hot spots were determined to prevent accidents. In Chapter 6, we obtained maximum likelihood equations for parameters of EW distribution based on RSS when all parameters are unknown. The Monte Carlo simulation study is conducted in R software and an application study for vehicle headway is presented to show the flexibility of EW distribution. In Chapter 7, the conclusions are given.

CHAPTER 2
CLUSTERING ALGORITHMS

Data mining is, in simple term, the process of extracting useful information from the large amount of data by using the algorithms. One of the fundamental task in data mining is clustering. Clustering methods basically aim to group similar data points in the same group and dissimilar data points among groups. Because of the challenging goal of clustering and the usefulness of mined patterns, clustering methods can be applied into different fields. For example, clustering methods can be applied in astronomy in order to group stars into galaxies automatically. In military field, they can be used to detect clusters of mines. In business contexts, clustering can be used for grouping customers into classes with a group profile. Moreover, the areas can be characterized based on their geologic features in geography using clustering methods.

Clustering algorithms are categorized into four classes, namely, partitioning-based, density-based, grid-based and hierarchical methods. Partitioning-based methods define clusters by grouping data points into k partitions, which is defined by the user. Each partition respresents a cluster and a point is determined to be similar with the points in its partition. K-means algorithm is one of the simplest clustering method proposed by MacQueenet al. (1967). It aims to divide the data points into k clusters where each data point belongs to the closest cluster. It is easy to understand and implement and is highly used in scientific and industrial applications. However, the number of cluster (k) needed to be predicted by user which is a difficult task. Another disadvantage of the algorithm is the distance measure which affects the speed of algorithm especially when the sample size is getting larger. Also, initial data points have a strong impact on the final cluster form. Moreover, the performance of the k-means algorithm is affected by noise and outliers and it is not appropriate for clusters with arbitrary shapes. In the literature there are a variety of modified k-means method in order to eliminate the disadvantages. One of the modification of k-means algorithm is k-medoids, which helps to reduce sensitivity to outliers. The procedure is similar with the k-means algorithm, the only difference is that the center of the cluster

(medoid) is one of the data points instead of the mean of data points in cluster. Partitioning around medoids (PAM) is a representative k-medoids clustering method proposed by Rousseeuw & Kaufman (1990) . Its main idea basis on selecting k representative points in order to form initial clusters. PAM is a popular among many k-medoids clustering method beecause of being more powerful. However, PAM is not efficient for large scale data since its time complexity. In the aim of removing some of the disadvantages, Rousseeuw & Kaufman (1990) proposed clustering large applications (CLARA) algorithm. Park & Jun (2009) proposed a new algorithm for selecting most middle objects as initial centroids in k-medoids clustering. The performance of the proposed method is compared with the k-means algorithm by using the accuracy measure. The numerical experiment results show that the accuracy of the proposed method is higher than k-means algorithm and it has significantly less computation time than PAM. Zadegan et al. (2013) used the ranked-based similarity to find the medoids faster. They introduced new function that ranks objects according to their similarities. It is figured out that ranked k-medoids method is faster and more accurate than its counterparts for large datasets.

Grid-based methods are firstly formed a grid structure by dividing the data space into a finite number of cells. Then, the all clustering operations are performed on that grid structure instead of data points. They are computationally efficient for large datasets because of being independent from the number of data points. The first grid-based approach for clustering is Grid-Clustering proposed by Schikuta (1996). The algorithm divides the data space into d-dimensional blocks, which are non-empty and cover all data space. The density of each block is calculated and then the blocks are sorted by their density index. The blocks which has the highest density index are chosen as the cluster centers. Then, neighbour searching algorithm is applied in order to merge blocks as clusters. Schikuta & Erhart (1997) proposed another algorithm called BANG in the aim of reducing the inefficiencies in Grid-Clustering algorithm. They adressed the grid size, neighbour search and density issues in Grid-Custering. One of the most popular grid based method bases on statistical information for spatial data is the statistical information grid-based algorithm proposed by Wang et al.

(1997).

The density-based clustering approach is based on the idea that the density of points within a cluster is higher as compared to those outside of it. One of the most popular algorithm is density-based spatial clustering of application with noise (DBSCAN) is presented by Ester et al. (1996) in the aim of discovering clusters of arbitrary shaped data with noise. The main idea of the algorithm is that for each point of a cluster the neighborhood of a given radius has to contain at least a minimum number of points. In other words, the density in the neighborhood of a point has to exceed some threshold. By this way, the points which are density reachable from each other are assigned to the same cluster.

The predetermination of the number of cluster does not required in DBSCAN algorithm and it is capable of handling the noise effectively. Moreover, the performance of DBSCAN algorithm in discovering clusters is efficient in convex shaped data as well as in arbitrary shaped data. However, ϵ and $MinPts$ parameters are needed to be specified by user and it is difficult to predict an appropriate ϵ value manually. Because of the some disadvantages of the algorithm, such as the inability to find clusters with different density and computational complexity, the extensions of DBSCAN are of great interest in last decades. OPTICS is proposed by Ankerst et al. (1999) is an augmented ordering of the data set and represents its density-based clustering structure. Since the augmented ordering contains the same information as the density-based clustering obtained from a broad range of parameter settings, the difficulty of determining the parameter values is overcome. Campello et al. (2013) introduced HDBSCAN, which provides a hierarchical density-based clustering and bases on extracting only the most significant clusters. Campello et al. (2013) proposed DBSCAN* method which assignes data points in border as noise points. In addition, Khan et al. (2018) proposed Adaptive DBSCAN (ADBSCAN) algorithm in order to determine an appropriate ϵ and $MinPts$ values. In numerical experiments, it is found that ADBSCAN algorithm performs better than DBSCAN algorithm in terms of accuracies when the clusters have different densities. More examples of such results can be found in Birant & Kut (2007), Kumar & Reddy (2016), Bryant & Cios

(2017) and Wang et al. (2018).

In addition, some studies have modified the DBSCAN method with a grid-based approach to reduce the complexity of the DBSCAN algorithm. The motivation behind the grid-based DBSCAN approach is to divide the data space into equal-sized grids based on ϵ value and the data dimension to reduce the runtime of the algorithm. Uncu et al. (2006) presents a modification of DBSCAN algorithm called Grid DBSCAN (GRIDBSCAN) to provide solution for inability of DBSCAN to recognize cluster with different densities. The proposed algorithm is compared with DBSCAN algorithm and the results show that GRIDBSCAN has higher accuracy. However, it is found that GRIDBSCAN is not suitable for large dataset because of computational complexity. Boonchoo et al. (2019) combine the Grid-based DBSCAN with Cluster Forest algorithm, which indexes the non-empty grids and prevents unnecessary merging operations. More additional studies about grid-based DBSCAN methods can be found in Yanchang & Junde (2001), Gan & Tao (2015), Sakai et al. (2017), Chen et al. (2018) and references therein.

Hierarchical algorithms can form clusters in two different ways; top-down (divisive) approach or bottom-up (agglomerate) approach. Divisive techniques first define a broader cluster that contains all data points, then split these cluster into child clusters. These child clusters splitted into their own child clusters. This procedure continues until each point represents a cluster by own. The agglomerate method is the reverse of the divisive method. Each data point represents a cluster first and then they merged until the all points are contained in one cluster. Some of the clustering algorithms use the ideas of more than one clustering method. Therefore, it is not an easy task to assign a particular algorithm to a single clustering method category. Moreover, the main idea of some clustering methods may be the combination of clustering methods in different categories.

In the clustering methods above, each data point is assigned to exactly one cluster. There is also a clustering approach called fuzzy clustering, where each data point can belong to more than one cluster. For more details about fuzzy clustering, one can see

the sudies of Ulutagay (2009), Ulutagay & Nasibov (2012), Ruspini et al. (2019) and references therein.

In the following, some well-known clustering methods mentioned throughout the thesis are briefly explained.

2.1 DBSCAN Method

Density-based Spatial Clustering of Application with Noise (DBSCAN) is presented by Ester et al. (1996) in the aim of discovering clusters of arbitrary shaped data. The main idea of the algorithm is that for each point of a cluster the neighborhood of a given radius has to contain at least a minimum number of points. In other words, the density in the neighborhood of a point has to exceed some threshold. The size of a neighborhood is determined using a distance function denoted by $dist(p, q)$ for the points p and q. The definitons to formalize the notion of the algorithm are given below.

Definiton 1: (ϵ-neighborhood) The ϵ-neighborhood of a point p in a data set D, denoted by $N_\epsilon(p)$, is defined $N_\epsilon(p) = \{q \in D | dist(p, q) \leq \epsilon\}$.

Definiton 2: (core point) The point p is a core point if $|N_\epsilon(p)| \geq MinPts$, where $MinPts \in \mathbb{Z}^+$.

Definiton 3: (border point) The point p is a border point if it is not a core point but is a neighbour of a core point.

Definition 4: (directly density-reachable) A point p is directly density-reachable from a point q if $p \in N_\epsilon(q)$ and $\mid N_\epsilon(q) \mid \geq MinPts$ (core point condition).

Definition 5: (density-reachable) A point p is called density reachable from a point q if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i.

Definition 6: (density-connected) A point p is density connected to a point q if there

is a point o such that both, p and q are density-reachable from o.

Definition 7: (cluster) A cluster B in a data set D is a non-empty subset of D satisfying the following conditions:

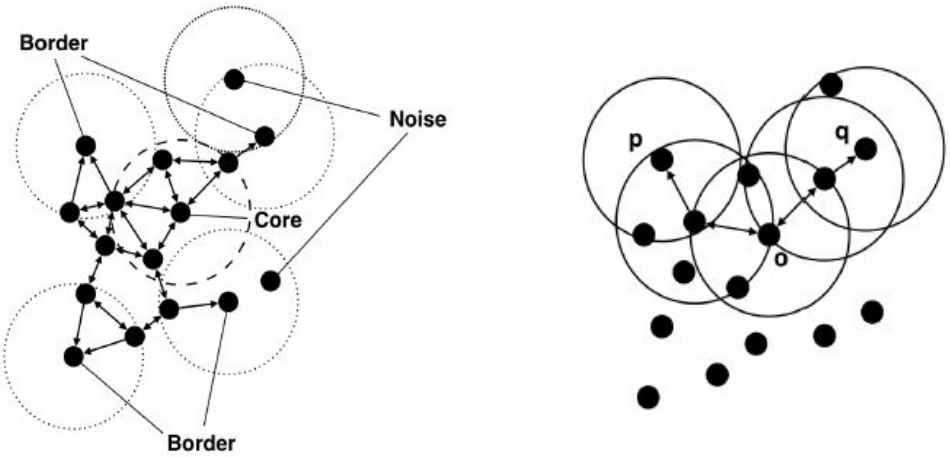
1) $\forall p, q$: if $p \in B$ and q is density-reachable from p then $q \in B$ (Maximality).

2) $\forall p, q \in B$: p is densityity-connected to q (Connectivity).

Definition 8: (noise) A point p is defined as noise if is not belonging to any cluster B_i, where , $i = 1, \ldots, k$ (noise = $\{p \in D | \forall i : p \notin B_i$).

A cluster is defined to be a set of density connected points and noise is simply the set of points in D not belonging to any of its clusters.

The definitions in DBSCAN algorithm for $\epsilon = 1$ and $MinPts = 4$ are illustrated in Figure 2.1 by Hahsler et al. (2019). Examples of the core, border and noise points are shown in Figure 2.1 (a). The p and q given in Figure 2.1 (b) are density-connected points, since p and q are both density-reachable from o.



(a) (b)

Figure 2.1 Illustration of the definitions in DBSCAN algorithm.

In the following, the algorithm of DBSCAN is given.

Step 1 The parameters ϵ and $MinPts$ are determined.

Step 2 Select a random data point p and retrieve the points which are density reachable from p.

Step 3 If p is a core point, then the cluster is formed. If p is a border point, then another point in the dataset is visited.

Step 4 Continue the process until all of the points have been processed.

Step 5 The remaining data points are marked as noise points.

After applying the five steps above, the data points will be clustered successfully.

2.2 OPTICS Method

OPTICS is the extended version of DBSCAN algorithm and is basis on an augmented ordering of the data set developed by Ankerst et al. (1999). It does not cluster the dataset; but instead it creates an augmented ordering of the database that represents the density-based clustering structure. It contains the same information as density-based clustering from a wide variety of parameter settings. By this way, the difficulty of determining the parameter values in DBSCAN is overcome.

Parameters ϵ and $MinPts$ are required in OPTICS as in DBSCAN. The ϵ is used in the aim of reducing the runtime of the algorithm so it is not needed except for practical purposes. OPTICS works in principle like the DBSCAN algorithm for an infinite number of distance parameters ϵ_i, where $0 \leq \epsilon_i \leq \epsilon$. The main difference here is that instead of assigning cluster memberships to data points, the order in which data points are processed and the information that will be used by an extended DBSCAN algorithm to assign cluster membership is stored. The core-distance and the reachability-distance are the two pieces of information stored for each object.

There are two more definitions below, in addition to those given in DBSCAN.

Definition 9: (core-distance) The core-distance of a data point p is defined as;

$$coreDist(p) = \begin{cases} undefined & if\ |N_\epsilon(p)| < MinPts \\ MinPts - dist(p) & otherwise \end{cases}$$

where $MinPts - dist(p)$ represents the distance from the point p to its $MinPts$' neighbour. Intuitively, it is the smallest distance (ϵ') between the point p and another point in its neighbourhood.

Definition 10: (reachability-distance) The reachability-distance of a data point p to point q, where $p, q \in D$ is defined as;

$$reachabilityDist(p,q) = \begin{cases} undefined & if\ |N_\epsilon(p)| < MinPts \\ max(coreDist(p), dist(p,q)) & otherwise \end{cases}$$

In simple term, the reachability-distance between a core point p and another point q can be defined as the smallest distance as which p can be directly density-reachable from q.

The structure of the clusters can be obtained by using a reachability-plot, which is the visualization of orders of the points and their corresponding reachability distances. The data points which have low reachability-distance to their nearest neighbour belongs to a cluster. Therefore, the clusters appear as valleys on the reachability-plot. The depth of the valley indicates that the density of the cluster is high.

The density-based clusters can be extracted from the generated augmented order by scanning the order and assigning the data points to the clusters based on the reachability-distance and the core-distance values. Clusters created by OPTICS are very similar to clusters created by DBSCAN. If the border data points are processed before a core object of the corresponding set is found in OPTICS, then some of them may be missed.

2.3 HDBSCAN Method

The Hierarchical Density-based Cluster Application with Noise (HDBSCAN) is proposed by Campello et al. (2013) as a modificated version of DBSCAN* for varying ϵ values. One of its features that makes it stand out is that it only requires one parameter, which is $mpts$, in order to find clusters in a data set. The $mpts$ parameter represents the minimum data point number a cluster has to contain.

HDBSCAN uses a simplified clustering hierarchy of only the most significant clusters can be easily extracted. In HDBSCAN, the hierarchy of DBSCAN* clusterings are computed and the optimal cutting point in that hierarchy is extracted based on a stability-based extraction method.

The additional definitions for HDBSCAN are as follows.

Definition 1: (Core Distance) The core distance of a data point p is represented by $dcore(p)$ and is the distance between p and its $mpts$ nearest-neighbours.

Definition 2: (ϵ-Core Object) The data point p is called as ϵ-core object if $dcore(p) < \epsilon$.

Definition 3: (Mutual Reachability Distance) The mutual reachability distance of data points p and q is the maximum value of the core distance of p, core distance of q, and the distance between p and q.

$$d_m reach(p, q) = max(dcore(p), dcore(q), dist(p, q))$$

Definition 4: (Mutual Reachability Graph) The mutual reachability graph (G_{mpts}) is a weighted graph based on the mutual reachability distance between any two data points p and q.

The steps of the HDBSCAN algorithm can be summarised as below.

Step 1 The mutual reachability distances between data points are computed.

Step 2 The minimum spanning tree based on mutual reachability distance is constructed.

Step 3 The minimum spanning tree is pruned.

Step 4 The clusters are extracted.

CHAPTER 3
THE SPATIAL ADAPTIVE CLUSTERING METHOD

In this chapter, a new clustering method, spatial adaptive clustering (SAC), is introduced using the main idea behind the adaptive cluster sampling (ACS) which is an effective sampling method based on an adaptive selection process. Since ACS uses neighborhood search in the grid structure and uses recursively added units that reveal grouped individuals easily and quickly, SAC has some advantages, such as being easy to use, flexible and practical. In the following subsection the ACS procedure is given in details.

3.1 The Adaptive Cluster Sampling Design

The ACS is firstly proposed by Thompson (1990) to obtain a sample in an easy and effective way for rare and clustered populations. The ACS method is widely used in biology, ecology, epidemiology, geology. Moreover, more specific examples of the application areas of ACS include the epidemiology of rare diseases, rare and endangered animal/plant populations, pollution concentrations, uneven mineral surveys and hotspot investigations. Roesch (1993) used the ACS method for an investigation about the forest which tend to be clustered. Thompson & Seber (1995) mentioned about the examples of rare diseases, rare species, and environmental pollution studies where the use of an adaptive sampling method can be extremely beneficial. Smith et al. (2003) applied ACS in a survey of freshwater mussels which tend to be spatially clustered. Davis et al. (2011) used ACS design on the research about a rare and endangered fish living in the Cumberland River. They proved that the ACS design is an efficient sampling method and suitable for the researches of fish that spatially clustered. For more study about ACS and its applications, see Turk & Borkowski (2005), Patummasut & Borkowski (2014), Zhu & Fu (2018), Nolau et al. (2021) and references therein.

The ACS design relies on neighborhood search to select additional units based on the

initial sampling units selected. In ACS design, firstly an initial sample is obtained with a traditional sampling methods. Whenever the interested variable of the sampled unit in the sample satisfies a specific condition, units in the neighborhood of that sampled unit are also added to the sample. This process continues until there is no unit in the neighborhood which meets the specified condition. All collected units in the union of the initial sampled units and their neighbors added to the sample are called as a "cluster". Units that do not satisfy the condition, but are in the neighborhood of a unit that does are called 'edge units'. The term 'network' is a subcollection of units in a cluster isolated from edge units. Selecting any unit in the network to sample ensures that other units in the network are also included in the sample. Thus, the population can be uniquely divided into clusters.

The procedure of ACS method can be described as below.

Step 1: The research area is partitioned into S^2 squares.

Step 2: An initial sample of squares of size n_1 is chosen by using a conventional sampling design.

Step 3: If the neighbours (left-right-top-bottom) of the units which satisfy the condition in the sample satisfies the condition C, they are added to the ACS sample.

Step 4: Step 3 is repeated until sample includes all the neighbours of any unit satisfying the condition C.

All units that meet the condition C in the neighborhood of each other form a network. Units in the sample that do not meet the condition C are called edge units. A network and its associated edge units forms clusters; hence the name 'adaptive cluster sampling'.

The illustration of the sampling procedure is given in Figure 3.1. The initial sample, which is represented using the dark bordered units, is of size $n_1 = 3$. Units that represented with a dotted border are adaptively added units. A network of size 4 is formed with units with values of 2, 5, 3, 1. Units with a value of zero, represented

by a dotted border in the neighborhood of this network, are the edge units. A cluster is formed by the union of the edge units and the network of size 4. Other units with a value of zero in the sample are network of size 1.

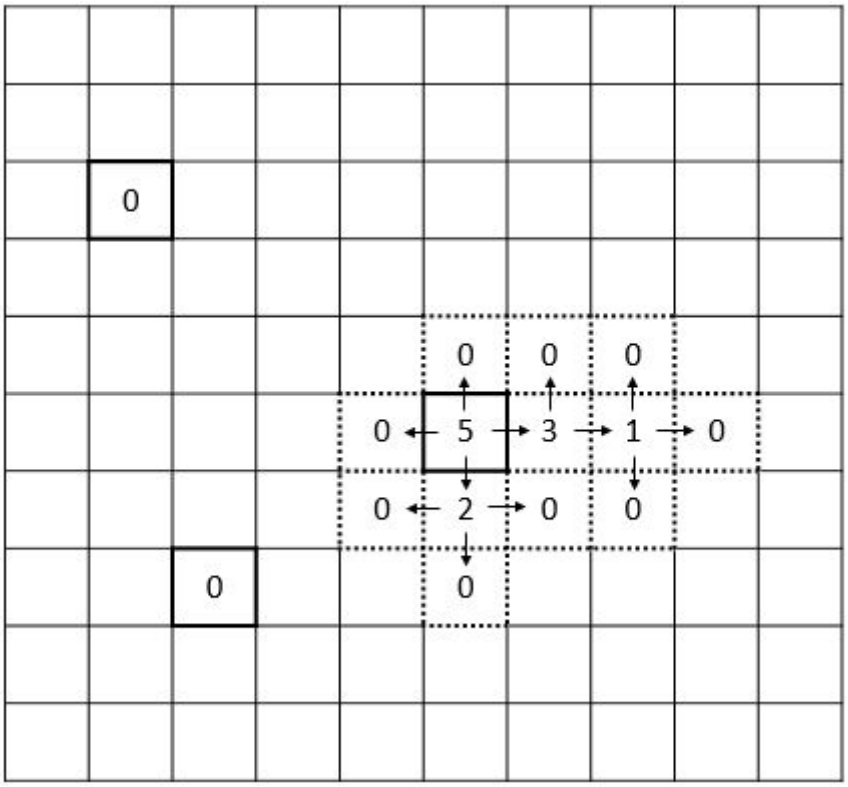


Figure 3.1 Illustration of ACS design for $n_1 = 3$ and $C = 1$.

The primary application of the sampling scheme is for spatially clustered populations. The steps of the ACS design give a chance to discover clusters in a data. The recursively added units in the procedure of ACS desing reveals the batched individuals easily and fastly. Therefore, this method can be adapted to a new form to use as an efficient clustering method.

3.2 Properties and Definitions

The SAC algorithm consists of four main parts; constructing the grid structure, calculating the density of each grid, forming the clusters using neighbourhood search and finally marking the noise points. In constructing grid structure part; the each

dimension of the data space is split into m grids. The grid width, together with m, depends on the minimum and maximum values in each dimension. After creating the grid structure, the grid densities are calculated by counting the data points assigned to each grid. Clusters are created depending on the densities of the networks by using the neighborhood search. At the end, the noise points are marked and the clustering process is completed.

Let **D** is a data set contains of N data points. $\{\omega_1, \omega_2, ..., \omega_k\}$ is the set of k clusters obtained by SAC algorithm and ω_0 consists of noise points marked by SAC, then the following properties are satisfied.

- $\omega_i \neq \emptyset$ and $\omega_i \subset D, \forall i, i = 1, 2, ..., k$
- $\omega_i \cap \omega_j = \emptyset, \forall i \neq j, i, j = 1, 2, ..., k$
- $\omega_i \cap \omega_0 = \emptyset, \forall i, i = 1, 2, ..., k$
- $(\cup_{i=0}^{k} \omega_i) = D$

Before describing the steps of the SAC algorithm, we give some definitions. Let $G_{m_1,m_2,...,m_d}$ represent a grid with the coordinates of $(m_1, m_2, ..., m_d)$ in a d dimensional grid structure.

Definition 3.2.1 (dense grid) *$G_{m_1,m_2,...,m_d}$ is a dense grid if it contains at least C data points.*

Definition 3.2.2 (non-dense grid) *$G_{m_1,m_2,...,m_d}$ is a non-dense grid if it is not a dense grid but contains at least one data point.*

3.3 Steps of the SAC Algorithm

The main idea of the SAC method is based on the neighbourhood search of grids as mentioned previously in ACS design. The SAC algorithm needs two input parameters,

say, the grid number m and the condition C. In this algorithm, from each dimension of the data space to be partitioned into m intervals, m^d grids and the number of data points in each grid are obtained. Thus, all clustering operations are performed on that grid structure based on neighbourhood search instead of data points. The steps of the SAC algorithm for d-dimensional dataset are given as below:

Step 1 The grid number m and the condition C are selected.

Step 2 Each dimension is partitioned into m intervals and m^d grids are obtained.

Step 3 The number of data points in each grid is calculated then the dense grids and non-dense grids are labelled.

Step 4 A dense grid is randomly chosen, the dense grid and all its non-dense grid neighbours are grouped together.

Step 5 The group is expanded if the dense grid has dense grid neighbours, randomly select one and repeat step 4 until there are no dense grid neighbours anymore, then all the grids in the group are labeled as a cluster.

Step 6 If there is any non-labeled dense grids, repeat steps 4 and 5.

Step 7 All non-labeled non-dense grids are grouped into the noise cluster.

After applying the 7 steps above, the points are clustered successfully. In addition, it is recommended that a cluster should consist of at least a certain number of data points in order to obtain meaningful results.

3.4 Illustrative Example

The steps executed for an example dataset are shown in Figure 3.2 (a). First, the parameters m and C are determined by the researcher as $m = 10$ and $C = 2$, respectively. After, the grid structure is constructed by segmenting the space into

$m^2 = 100$ cells. The number of data points in each grid is calculated and dense and non-dense grids are labelled. A dense grid is randomly selected as shown with red square in Figure 3.2 (b). Then, its non-empty top-bottom-left-right neighbourhoods are added to the cluster. The neighbours are checked for the condition C in Figure 3.2 (c) and it is seen that top neighbour contains data points more than C, so its non-empty neighbourhoods are also added to cluster. The first cluster is constituted as seen in Figure 3.2 (d). Another cell is chosen randomly and all the steps are applied until second cluster is constituted. At the end, the remaining non-dense grids are marked as noise. After applying all the steps of the SAC algorithm, the data set is successfully segmented into two clusters with three noise points as it is seen in Figure 3.2 (g).

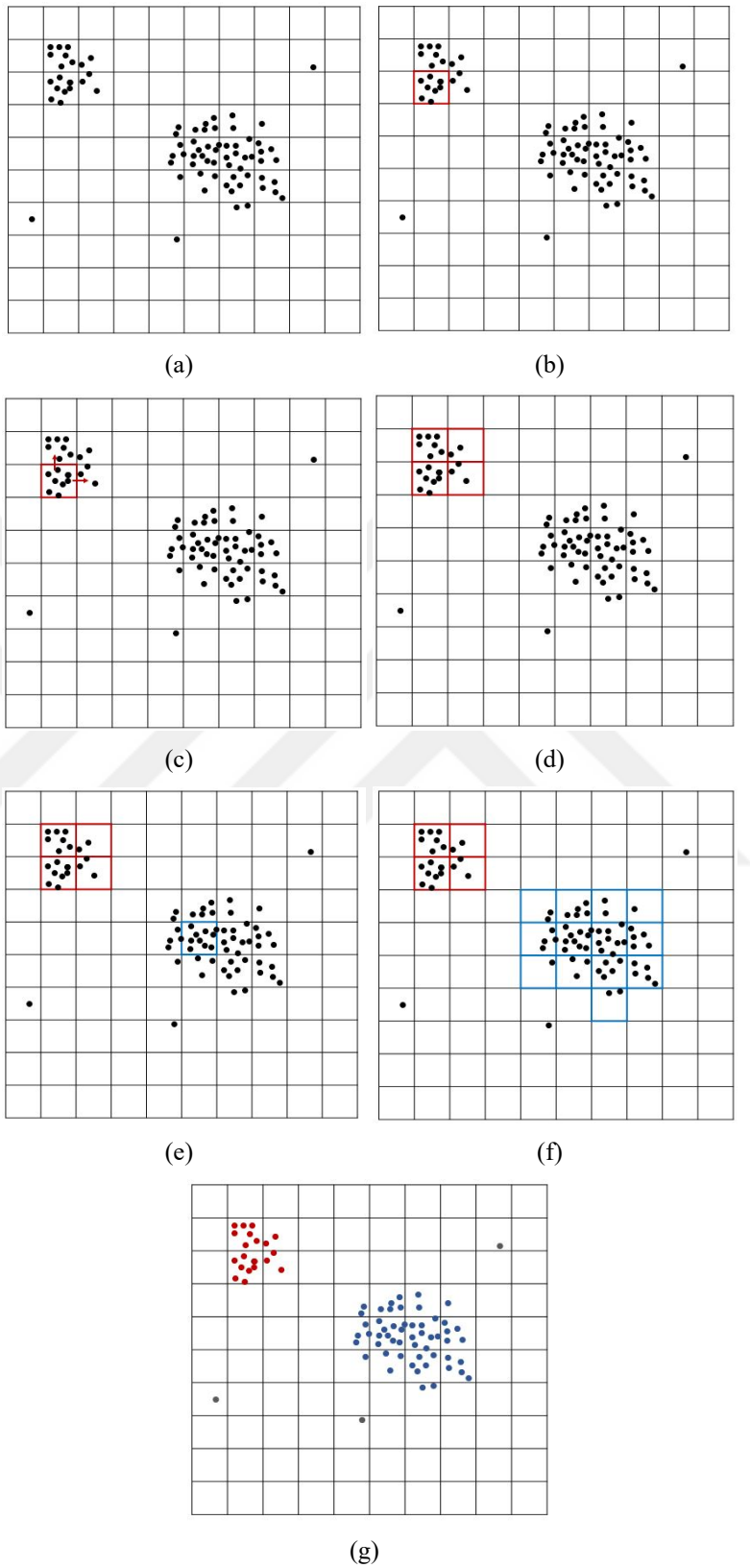


Figure 3.2 Illustrative example for the steps of SAC algorithm ($m = 10$ and $C = 2$)

CHAPTER 4
PERFORMANCE EVALUATION OF THE SAC METHOD

In this chapter, experimental studies are performed in order to demonstrate the performance of the proposed method using different real and artificial data sets. The experiments are conducted in R Core Team (2020) 4.0.3 and the operating system of the computer is Intel Core i7 2.50 GHz processor with 8 GB RAM. We experimentally compare the proposed algorithm with the DBSCAN method and its modifications, namely, OPTICS and HDBSCAN.

The performances of algorithms are evaluated from two different perspectives: external validation measures and runtime. In the first part, we performed experiments on eight different real datasets of various sizes and dimensions obtained from UCI repository of Dua & Graff (2019). In the second part, we performed experiments on six state-of-the-art synthetic data sets. In addition, the SAC algorithm is compared with DBSCAN and OPTICS in terms of their runtimes on five different synthetic data sets of sizes in varied range.

DBSCAN, OPTICS and HDBSCAN methods are implemented using the `DBSCAN` package by Hahsler et al. (2019) in R. For each of the algorithm, the parameters with the maximum ARI value are selected corresponding to their search ranges from the Table 4.1.

Table 4.1 The search ranges of parameters required for algorithms.

Algorithm	Search Range
DBSCAN	$minPts \in \{5, 10, 15, 20\}$, $\epsilon \in [0, 2]$
OPTICS	$minPts \in \{5, 10, 15, 20\}$, $\epsilon' \in [0, 10]$
HDBSCAN	$minPts \in \{2, 3, 4, \ldots, 20\}$
SAC	$m \in \{2, 3, 4, \ldots, 20\}$ $C \in \{1, 2, 3, 4\}$

In the experiments, we compare the performances of these methods in terms of three validation measures namely, purity, Rand index (RI) and adjusted Rand index (ARI) given as follows.

Purity: The purity is a commonly used external validation measure that evaluates the "purity" of the clusters based on the given class labels. The value of purity can be in the interval $[0, 1]$ and the higher purity value indicates a better clustering result. Assume that **D** is the data set of size N and consists of s classes, denoted by $\{D_1, D_2, ..., D_s\}$. After applying a clustering algorithm, k clusters obtained as $\Omega = \{\omega_1, \omega_2, ..., \omega_k\}$. The purity of a clustering method can be defined as

$$\text{purity}(\Omega) = \frac{1}{N} \sum_{i=1}^{k} \max_{j} |\omega_i \cap D_j|.$$

Rand Index (RI): The RI measure is proposed by Rand (1971) as the percentage of the correctly clustered data points and its value ranges from 0 to 1. The value of 0 indicates that the method works as a random clustering assignment and the value of 1 indicates that there is a perfect clustering. The rand index of a clustering method can be computed by

$$\text{RI}(\Omega) = \frac{TP + TN}{TP + FP + TN + FN},$$

where TP and TN represent the number of true positives and true negatives, and FP and FN represent the number of false positives and false negatives, respectively.

Adjusted Rand Index (ARI): The ARI measure is proposed by Hubert & Arabie (1985) as an improved version of RI. It can be calculated by using the following formula,

$$\text{ARI}(\Omega) = \frac{\binom{N}{2}(TP + TN) - [(TP + FN)(TP + FP) + (FP + TN)(FN + TN)]}{\binom{N}{2}^2 - [(TP + FN)(TP + FP) + (FP + TN)(FN + TN)]},$$

where TP and TN represent the number of true positives and true negatives, and FP and FN represent the number of false positives and false negatives, respectively.

4.1 Experimental study on real data sets

In this section, we compare the SAC, DBSCAN, OPTICS and HDBSCAN algorithms using eight different real data sets from UCI repository of Dua & Graff (2019). We give some details of data sets as follows.

Iris: This is one of the best known databases in the literature. Iris is a 4 dimensional data set with 50 instances per 3 different type of species, namely, Setosa, Versicolour and Virginica. It contains 4 features of the species: Sepal length, sepal width, petal length, and petal width.

Glass: The Glass data set is motivated by the criminological studies in order to identify the crime scene. It is very important to identify the glass found at the crime scene in order to use them as evidence. The data set of size 214 contains the oxide content of 7 types of glass, namely, refractive index, Sodium, Magnesium, Aluminum, Silicon, Potassium, Calcium, Barium and Iron.

Seeds: It includes measurements of the geometrical properties of wheat kernels of the Kama, Rosa, and Canadian type obtained by a soft X-ray technique. Seven geometric parameters including area, perimeter, compactness, length, and width of the kernel, asymmetry coefficient, length of kernel groove belonging to 210 observations are measured.

Banknote: The data set is obtained from images from genuine and forged banknote samples. The digitization is done with a camera used in print inspection. The features are extracted from the images created in grayscale with the Wavelet Transform. The dataset contains 1372 observations of different banknotes. Classes indicate whether a banknote is genuine or counterfeit. The features in the data set are the some of the characteristcis of Wavelet Transformed image, namely variance, skewness, curtosis and, entropy.

Wine: The data includes the chemical analysis results of three different wines in

Italy. Alcohol, malic acid, ash, Alcalinity of ash, magnesium, phenols, total phenols, flavanoids, nonflavanoid, proanthocyanins, color intensity, hue, dilution, and proline are the features of 178 sample in data set.

E.coli: It contains the information of bacteria of the genus Escherichia coli living in the lower intestine. The clustering algorithms are performed on 7 features of 336 observations.

HTRU2: The HTRU2 data set (Lyon et al. (2016)) contains the information about the candidates pulsar which are a rare type of Neutron star that produce radio emissions. It contains 8 features in total, including mean, standard deviation, kurtosis, and skewness of both the integrated profile and the DM-SNR curve. The 17898 observations in the sample are obtained from the negative and positive classes of pulsar candidates.

PageBlocks: The dataset consists of 5473 data points and their 10 features that define the page layouts of the documents determined by a segmentation operation.

In addition, the characteristics of the real data sets are summarized in Table 4.2. By the algorithms, purity, RI, ARI, runtime and observed cluster number (#cluster) are presented in Table 4.3. Based on the results, we can conclude that SAC gives best results in terms of external validation measures for all data sets, except Glass. The observed cluster number of the SAC method is close to the real number of class, except Banknote, HTRU2 and PageBlocks data sets.

For Iris, Banknote and E.coli data sets, the SAC performs slightly better than the others in terms of ARI, RI and purity. For PageBlocks, the SAC algorithm gives a much superior result than the others according to ARI . For Glass data set, the SAC performs slightly better than the rest in terms of RI and purity, while it performs better than OPTICS and HDBSCAN algorithms in terms of ARI. The DBSCAN algorithm could not produce any cluster for Wine and HTRU2 data sets with given parameter search ranges and OPTICS produce only one cluster for E.coli data set, which is not approved. In addition, RI measure could not calculated for algorithms because of the

higher sample size of the HTRU2 data set.

Table 4.2 The characteristics of real data sets used in experiments.

Data set	Size	Dimension	Number of Class
Iris	150	4	3
Glass	214	9	7
Seeds	210	7	3
Banknote	200	6	2
Wine	178	13	3
E.coli	336	7	8
HTRU2	17898	8	2
PageBlocks	5473	10	5

Table 4.3 Comparisons on algorithms for real data sets based on external validation measures.

Data set		DBSCAN	OPTICS	HDBSCAN	SAC
Iris	ARI	0.5898	0.5681	0.5678	0.5916
	RI	0.8339	0.7763	0.7729	0.8341
	purity	0.8800	0.6667	0.6667	0.9133
	runtime	0.0005	0.0086	0.0094	0.1585
	#cluster	4	2	2	6
Glass	ARI	0.2845	0.2558	0.2491	0.2607
	RI	0.6371	0.6135	0.5939	0.6413
	purity	0.5132	0.5000	0.4813	0.5140
	runtime	0.0012	0.0144	0.0142	0.4114
	#cluster	3	2	2	6
Seeds	ARI	0.3678	0.3165	0.3421	0.5708
	RI	0.7351	0.6726	0.7204	0.8214
	purity	0.7810	0.6190	0.7571	0.8667
	runtime	0.0006	0.0127	0.0129	1.4621
	#cluster	4	2	4	4
Banknote	ARI	0.5584	0.4471	0.4788	0.6109
	RI	0.7790	0.7220	0.7380	0.8047
	purity	0.8965	0.9308	0.9366	0.9701
	runtime	0.0050	0.0831	0.1193	1.2689
	#cluster	6	7	10	21
Wine	ARI	-	0.1239	0.1093	0.2797
	RI	-	0.5931	0.5889	0.7058
	purity	-	0.5730	0.5730	0.7247
	runtime	-	0.0105	0.0072	5.7686
	#cluster	-	2	3	5
E.coli	ARI	0.4873	0.0513	0.4094	0.4951
	RI	0.7725	0.3588	0.7384	0.7954
	purity	0.6339	0.4464	0.6012	0.6845
	runtime	0.0018	0.0253	0.0223	3.5393
	#cluster	2	1	2	4
HTRU2	ARI	-	0.3520	0.3064	0.6072
	RI	-	-	-	-
	purity	-	0.91374	0.9107	0.9399
	runtime	-	29.5955	41.3147	58.2068
	#cluster	-	13	4	70
PageBlocks	ARI	0.1442	0.2231	0.0420	0.7110
	RI	0.8225	0.8227	0.5113	0.9170
	purity	0.9066	0.9063	0.9320	0.9424
	runtime	0.0085	2.8063	1.7096	4.1597
	#cluster	11	3	106	11

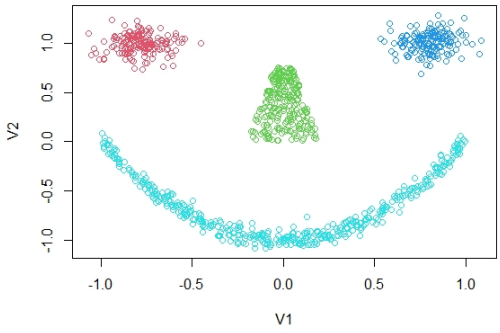
4.2 Experimental study on artificial data sets

In this section, two experimental studies are conducted using artificial data sets in the aim of examining the performance of SAC in two aspects; the first one is the clustering performance and the second one is runtime. Figure 4.1 shows the characteristics of the six artificial data sets with the number of data points (n) given under the plots in brackets. The x-axis (v1) and the y-axis (v2) in each plot respresent the dimension 1 and dimension 2 of the data set, respectively.

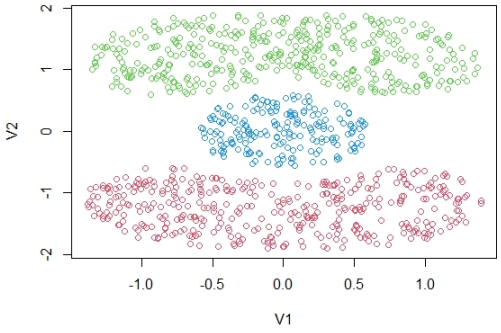
The ARI, RI and purity values of the algorithms for each data set are given in Table 4.4. From these results we can conclude that, all of the algorithms resulted with perfect clustering for Smiley and Shapes data sets. SAC gives better results than the rest for Jain data set, while it performs better than OPTICS and HDBSCAN algorithms in terms of all measures for Compund and Multishapes data sets. Furthermore, as shown in Table 4.4, the SAC algorithm works better than the OPTICS for all data sets.

Table 4.4 The external validation measures of the algorithms for artificial data sets.

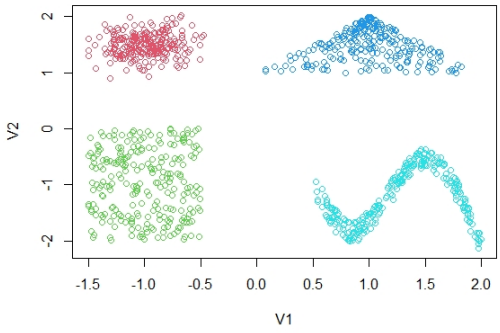
Data set		DBSCAN	OPTICS	HDBSCAN	SAC
Smiley	ARI	1	1	1	1
	RI	1	1	1	1
	purity	1	1	1	1
Cassini	ARI	1	0.9922	1	0.9983
	RI	1	0.9964	1	0.9992
	purity	1	0.9970	1	1
Shapes	ARI	1	1	1	1
	RI	1	1	1	1
	purity	1	1	1	1
Multishapes	ARI	0.9648	0.4491	0.7467	0.9497
	RI	0.9855	0.7071	0.9055	0.9793
	purity	0.9764	0.6091	0.9545	0.9673
Jain	ARI	0.9887	0.9885	0.9212	0.9971
	RI	0.9946	0.9939	0.9619	0.9986
	purity	0.9973	0.9971	1	1
Compound	ARI	0.9571	0.8205	0.8939	0.9146
	RI	0.9840	0.9290	0.9613	0.9682
	purity	0.9624	0.8847	0.9248	0.9373



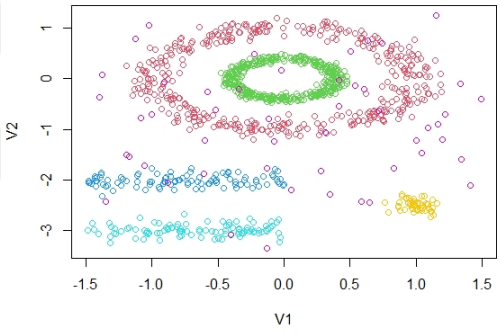
(a) Smiley ($n = 1000$)



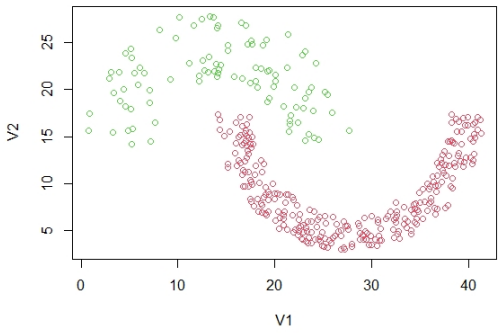
(b) Cassini ($n = 1000$)



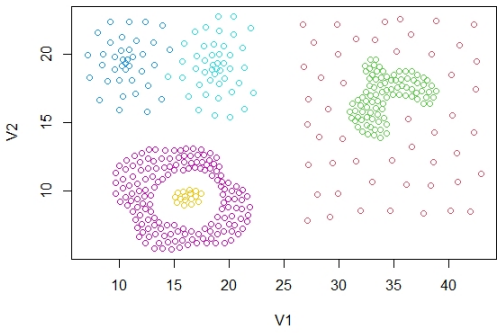
(c) Shapes ($n = 1000$)



(d) Multishapes ($n = 1100$)



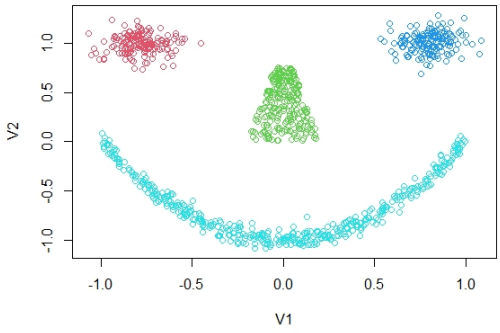
(e) Jain ($n = 373$)



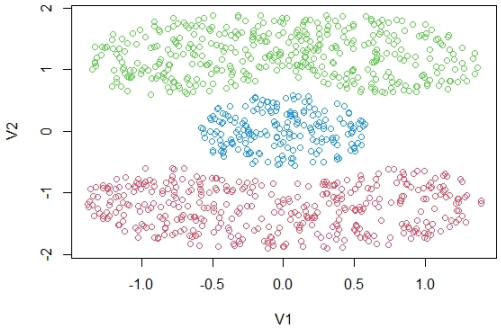
(f) Compound ($n = 399$)

Figure 4.1 The artificial data sets used for the comparison of measures.

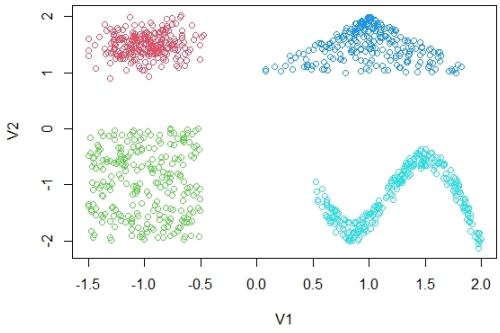
The clustering results of the DBSCAN, OPTICS, HDBSCAN, and SAC algorithms for each data set are shown in Figure 4.2, Figure 4.3, Figure 4.4, and Figure 4.5, respectively. In the plots, each detected cluster is represented by a different color and the data points shown in black represent the noise points marked by the algorithm. The clustering results show that, the performances of detecting the clusters with OPTICS and HDBSCAN algorithms is lower for multishapes data set. These algorithms are not succesfull in detecting clusters truely as in DBSCAN and SAC algorithms. DBSCAN and OPTICS algorithms detect only one cluster and HDBSCAN detects three clusters in Jain data. The only method among the algorithms, which is the most successfull one, is the SAC algorithm.



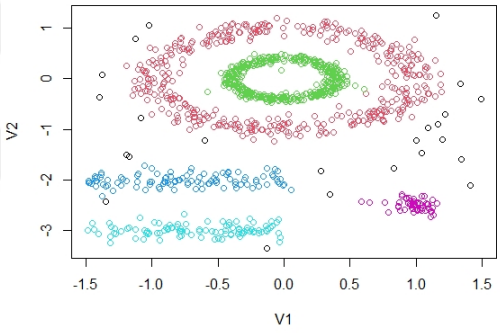
(a) Smiley



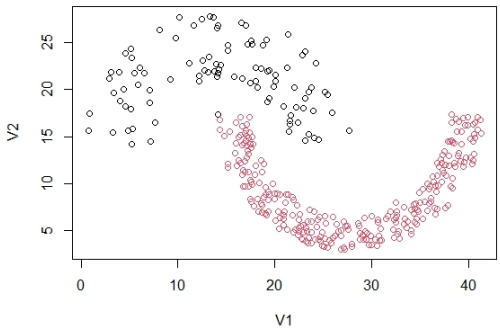
(b) Cassini



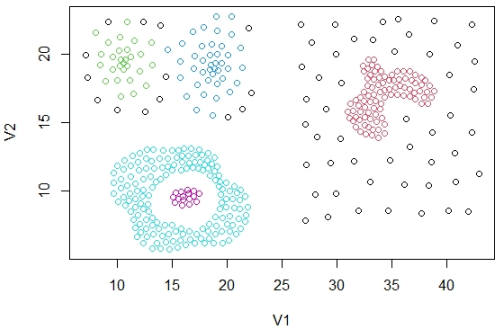
(c) Shapes



(d) Multishapes

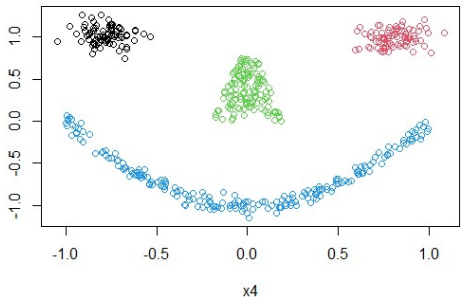


(e) Jain

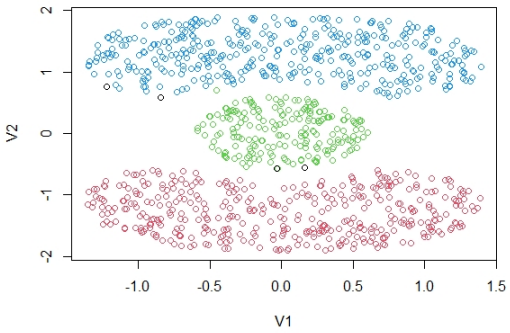


(f) Compound

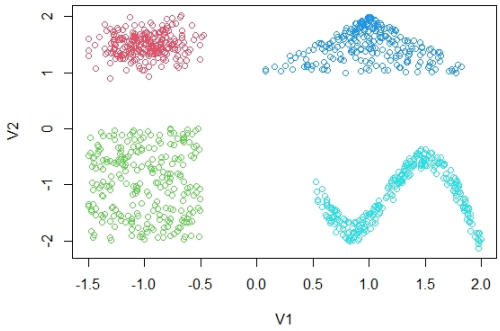
Figure 4.2 The clustering results of DBSCAN algorithm.



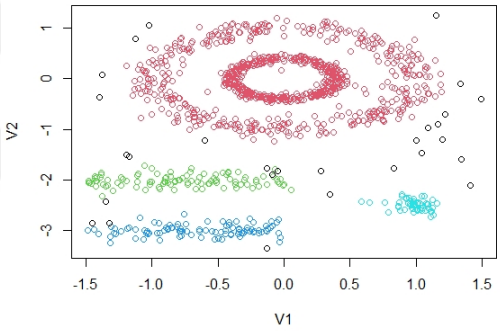
(a) Smiley



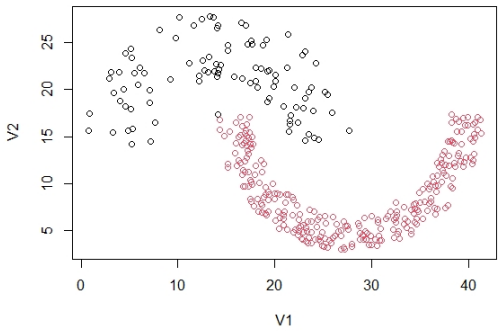
(b) Cassini



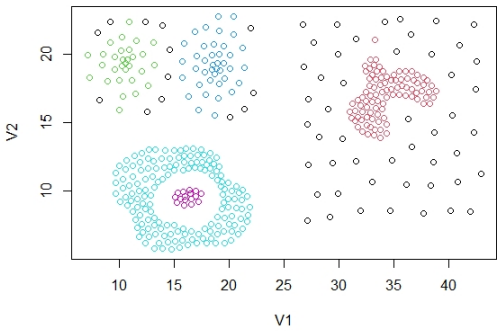
(c) Shapes



(d) Multishapes

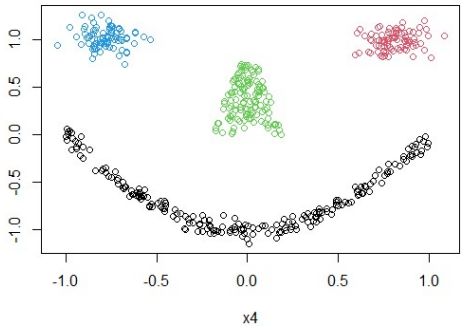


(e) Jain

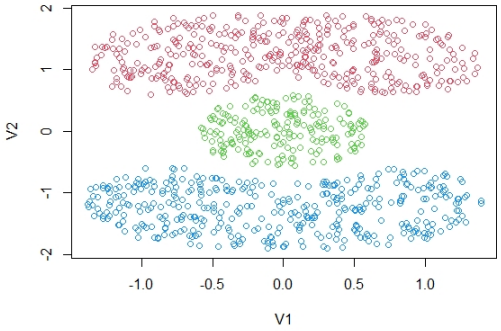


(f) Compound

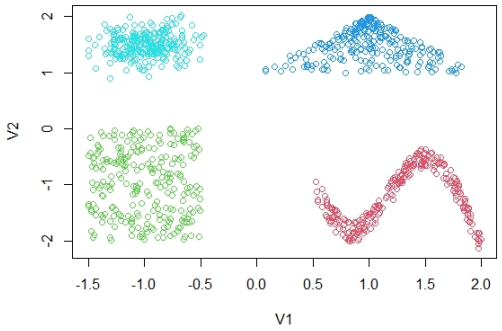
Figure 4.3 The clustering results of OPTICS algorithm.



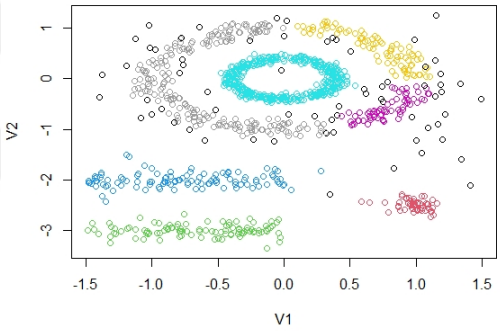
(a) Smiley



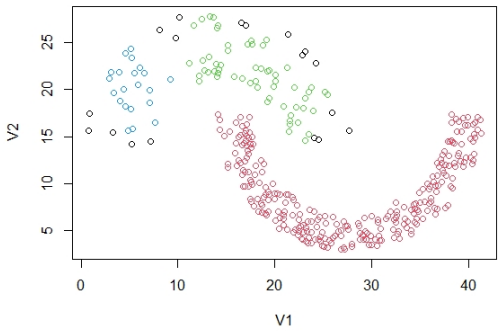
(b) Cassini



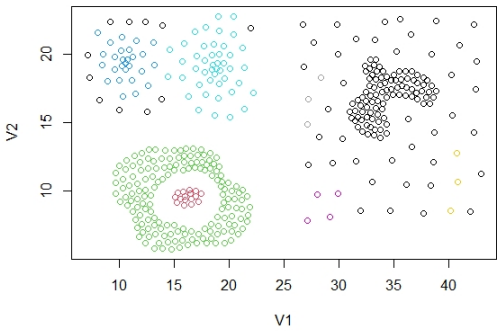
(c) Shapes



(d) Multishapes

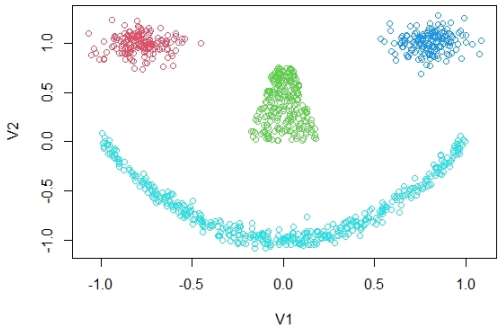


(e) Jain

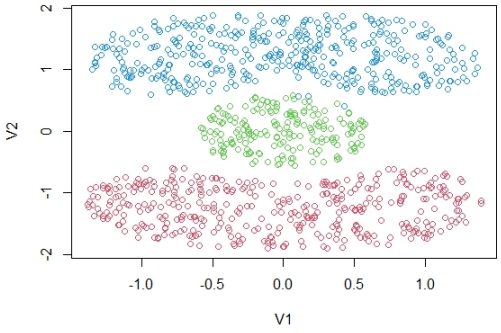


(f) Compound

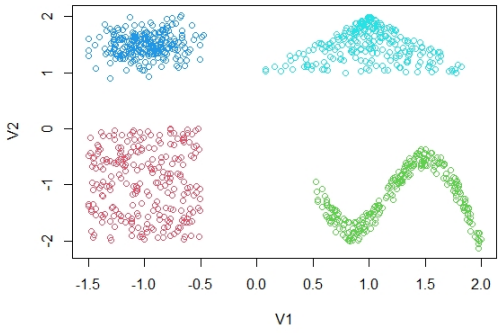
Figure 4.4 The clustering results of HDBSCAN algorithm.



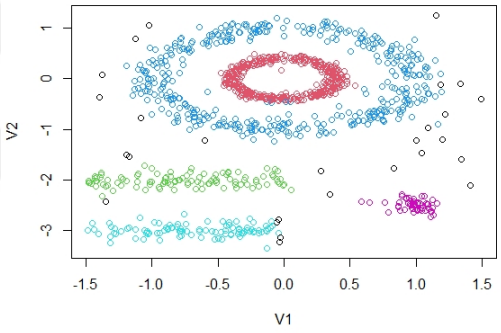
(a) Smiley



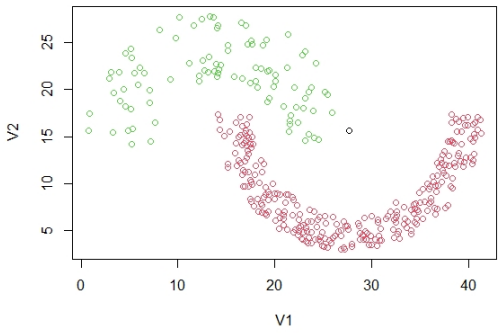
(b) Cassini



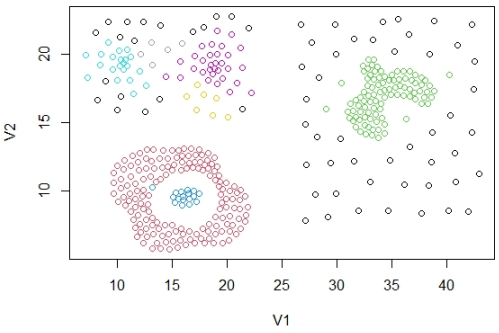
(c) Shapes



(d) Multishapes



(e) Jain



(f) Compound

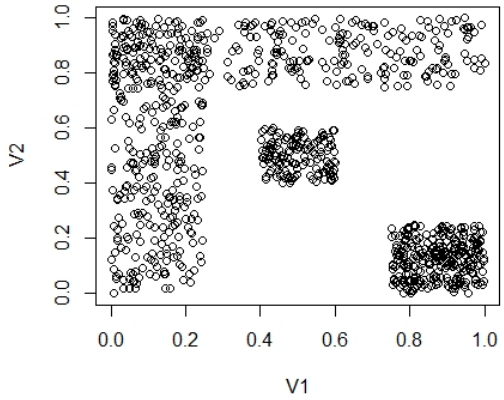
Figure 4.5 The clustering results of SAC algorithm.

4.2.1 Runtime comparisons of algorithms

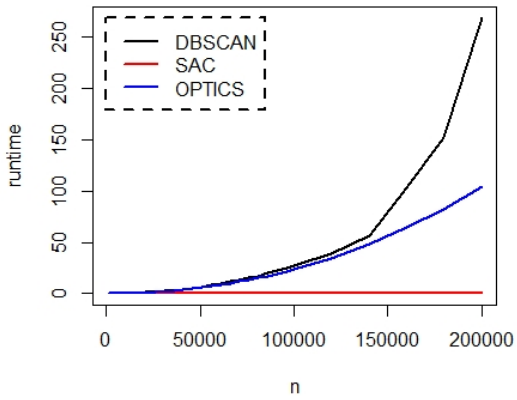
The second experimental study of this section is conducted in order to compare the speed of the algorithms on cuboids, smiley, spirals, and hypercubes data sets. The runtimes of the algorithms were recorded in seconds by increasing the number of data points from 1,000 to 200,000 throughout the experiment. Since the HDBSCAN algorithm does not work for large-scale data in R, it is not included in this part of the experimental study. For this reason, the SAC algorithm has only been compared with the DBSCAN and OPTICS algorithms. In Figure 4.6 (a), Figure 4.7 (a), and Figure 4.8 (a), the datasets for 1,000 sample sizes are given in order to visualize the datasets used for comparison of runtimes. The plots in Figure 4.6 (b), Figure 4.7 (b), and Figure 4.8 (b), are created using the recorded runtimes (in seconds) of the algorithms when the number of data points changes between 1,000 and 200,000. The red, black and blue lines represent the runtimes of SAC, DBSCAN and OPTICS algorithms, respectively.

According to the Figure 4.6 (b), there does not seem to be any difference between the runtimes of the algorithms when the data size is less than 40,000 for the cuboids data set. It is observed that when n increases, the runtimes of DBSCAN and OPTICS increase exponentially, while the runtime of SAC remains constant. The gap between OPTICS and DBSCAN increases as the n grow from 140,000. In addition, for the cuboids data set, the SAC algorithm performs in 0.55 seconds for the case of $n = 100,000$ while DBSCAN and OPTICS performs in 27.26 and 23.73 seconds, respectively. Thus, the SAC algorithm is about 50 times faster than the DBSCAN and 43 times faster than OPTICS.

The efficiency of the SAC algorithm in terms of runtime increases after $n = 30,000$ as seen in Figure 4.7 (b) for smiley data set. While the SAC algorithm is not affected by the change in data size, OPTICS and DBSCAN algorithms show a similar behavior and increase in their runtimes. For the smiley data, the SAC algorithm runs in 0.42 seconds for the case of $n = 200,000$. The DBSCAN and OPTICS algorithms run in 55.69 and 53.62 seconds, respectively. If we compare the difference between speeds



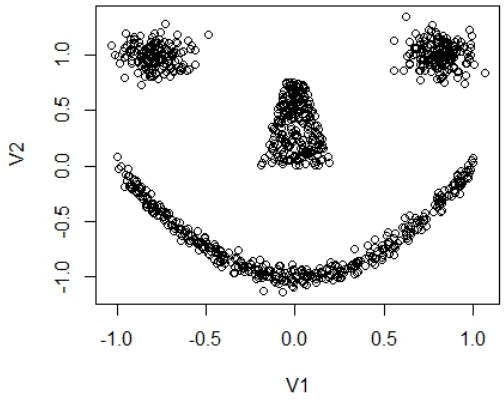
(a) Cuboids ($n = 1000$)



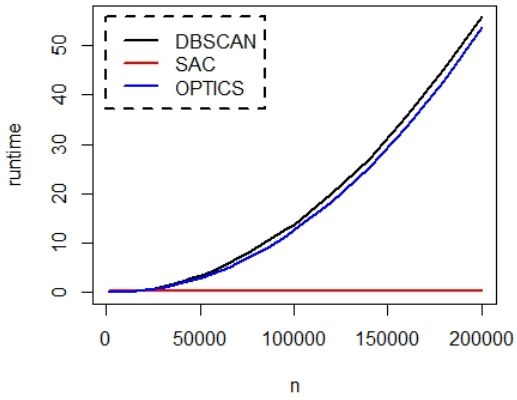
(b) Runtimes of the algorithms for different values of n in Cuboids.

Figure 4.6 The Cuboids data set and the runtime results of the algorithms.

of the algorithms, the SAC algorithm performs about 133 times faster than DBSCAN and 128 times faster than OPTICS.



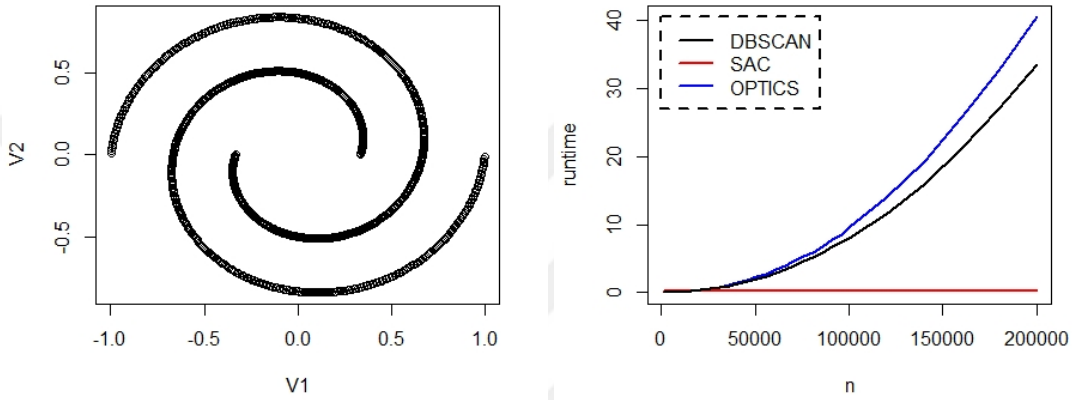
(a) Smiley ($n = 1000$)



(b) Runtimes of the algorithms for different values of n in Smiley.

Figure 4.7 The Smiley data set and the runtime results of the algorithms.

In Figure 4.8 (b), it is seen that the DBSCAN algorithm runs faster than OPTICS algorithm for spirals data set. The performance of the SAC algorithm is superior to those compared. For spirals data, the SAC algorithm runs in 0.34 seconds for the case of $n = 200,000$ while DBSCAN and OPTICS runs in 33.50 and 40.47 seconds, respectively. Thus, the SAC algorithm is about 97 times faster than the DBSCAN and 117 times faster than OPTICS.

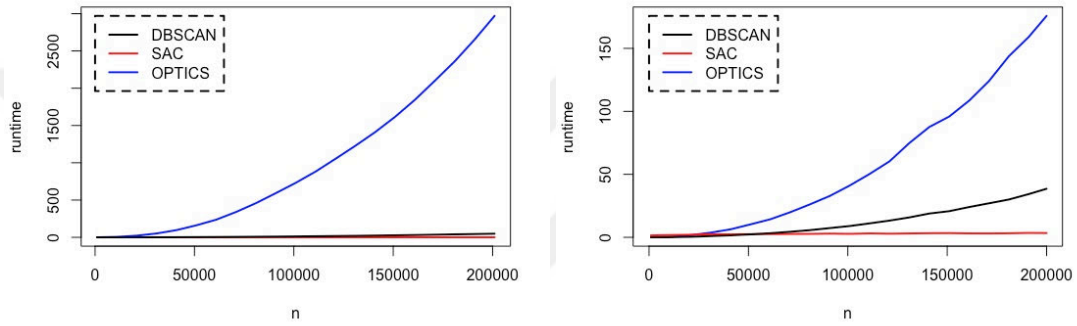


(a) Spirals ($n = 1000$)

(b) Runtimes of the algorithms for different values of n in Spirals.

Figure 4.8 The Spirals data set and the runtime results of the algorithms.

In addition, Figure 4.9 shows the recorded runtimes of the algorithms for cuboids and hypercubes data sets with dimensions 3 and 5, respectively. It is seen that the runtime of SAC remains stable while the runtime of the DBSCAN increases exponentially. OPTICS has a dramatically higher runtime than the others for both data sets. Moreover, for the cuboids data set, the SAC algorithm runs in 0.40 seconds for the case of $n = 200,000$ while DBSCAN runs in 48.56 seconds. It can be said that the SAC has the best runtime results in this comparison due to its minimal runtime.



(a) Cuboids (d=3) (b) Hypercubes (d=5)

Figure 4.9 Runtime (in secs) results of the algorithms in 3-dimensional and 5-dimensional data sets for different values of n.

CHAPTER 5
ANALYSIS OF ROAD ACCIDENT DATA BY THE SAC METHOD

In this chapter, we detected hotspots using the coordinates of accident locations for two different accident data sets with the SAC method. First, we examine accident data for Birmingham, second largest city of the United Kingdom and identify accident hotspots for the region with the highest number of accidents. In addition, we evaluated the factors that may cause the accident by examining the determined accident locations. As a second analysis, we used 3-year accident data obtained for İzmir, Turkey, and determined the accident hotspots in Buca for the regions where the accidents are intense. Then, we determine the potential factors causing the accident by analyzing the accident zones. All analyzes were performed using R statistical software and the `ggmap` (Kahle & Wickham (2013)) and `ggplot2` (Wickham (2011)) packages were used to visualize the results.

5.1 Analysis of UK accident data

The data set used in the analysis are obtained by the Department for Transport (2019) as ”Road Safety Data” and it includes the road accidents in UK from 2019. In the UK, accidents are reported to the police only if at least one person is injured or dead. The data set consists of 117,536 rows and 32 features. Some of the features are; accident index, date, longitude, latitude, police force, number of vehicle, accident severity, local authority, road type, light conditions, etc. The data set is preprocessed in order to make it appropriate for the analysis. The missing values are removed and some demographic variables related with drivers are merged in the data set. The features of the road accident data are presented in Table 5.1.

Table 5.1 The features of UK road accident data used in the analysis.

Feature	Subcategories	Feature	Subcategories
Accident Index	-	Light conditions	Daylight
Longitude	-		Darkness-lights lit
Latitude	-		Darkness-lights unlit
Accident Severity	Fatal		Darkness-no lighting
	Seious	Weather conditions	Fine no high winds
	Slight		Raining no high winds
Number of vehicles	1-17		Snowing no high winds
Number of casualties	1-52		Fine + high winds
Date	-		Raining + high winds
Day of week	-		Snowing + high winds
Time	-		Fog or mist
Local authority	1-941	Road surface conditions	Dry
Junction detail	Not at junction		Wet or damp
	Roundabout		Snow
	Mini-roundabout		Frost or ice
	T/staggered junction		Flood over 3 cm deep
	Slip road		Oil or diesel
	Crossroads		Mud
	More than 4 arms	Urban/rural area	Urban
	Private drive/entrance		Rural
	Other junction	Age of casualty	-
Junction control	Not at junction	Sex of casualty	Male
	Authorised person		Female
	Auto-traffic signal		
	Stop sign		
	Give way/uncontrolled		
	NA or out of range		

5.1.1 Descriptives of the data

In order to identify the most problematic local authority district in UK, the accident frequencies are examined as shown in Figure 5.1. The local authority district with the highest number of accident is Birmingham with 2,116 accidents. Therefore, Birmingham accident data is analysed in further for identifying the potential problematic zones. The accident locations are shown on the map as seen in Figure 5.2.

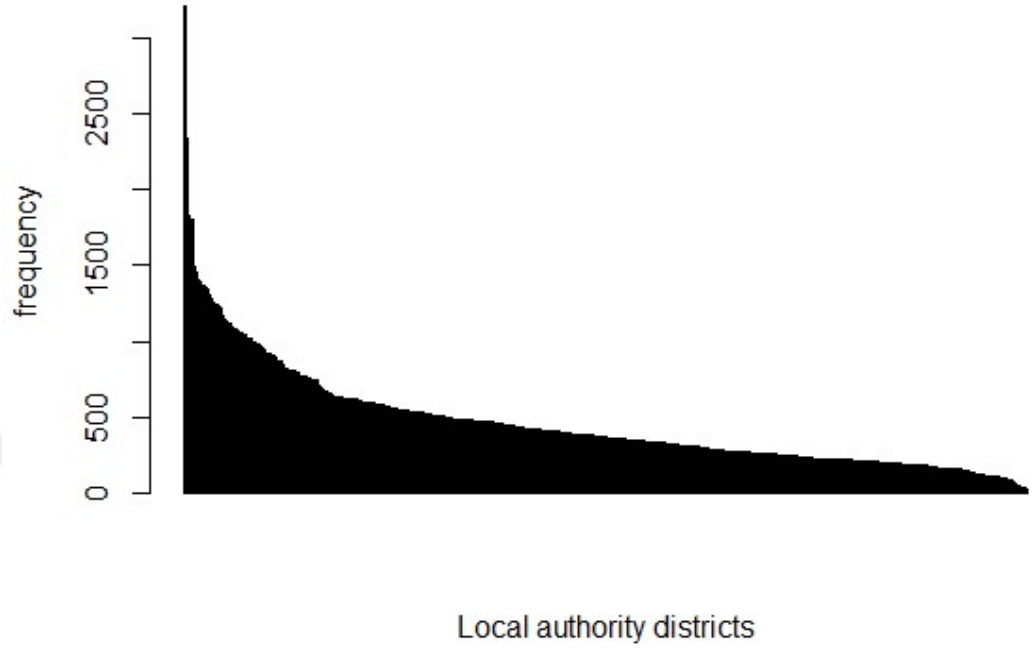


Figure 5.1 Accident frequency of local authority districts in descanding order.

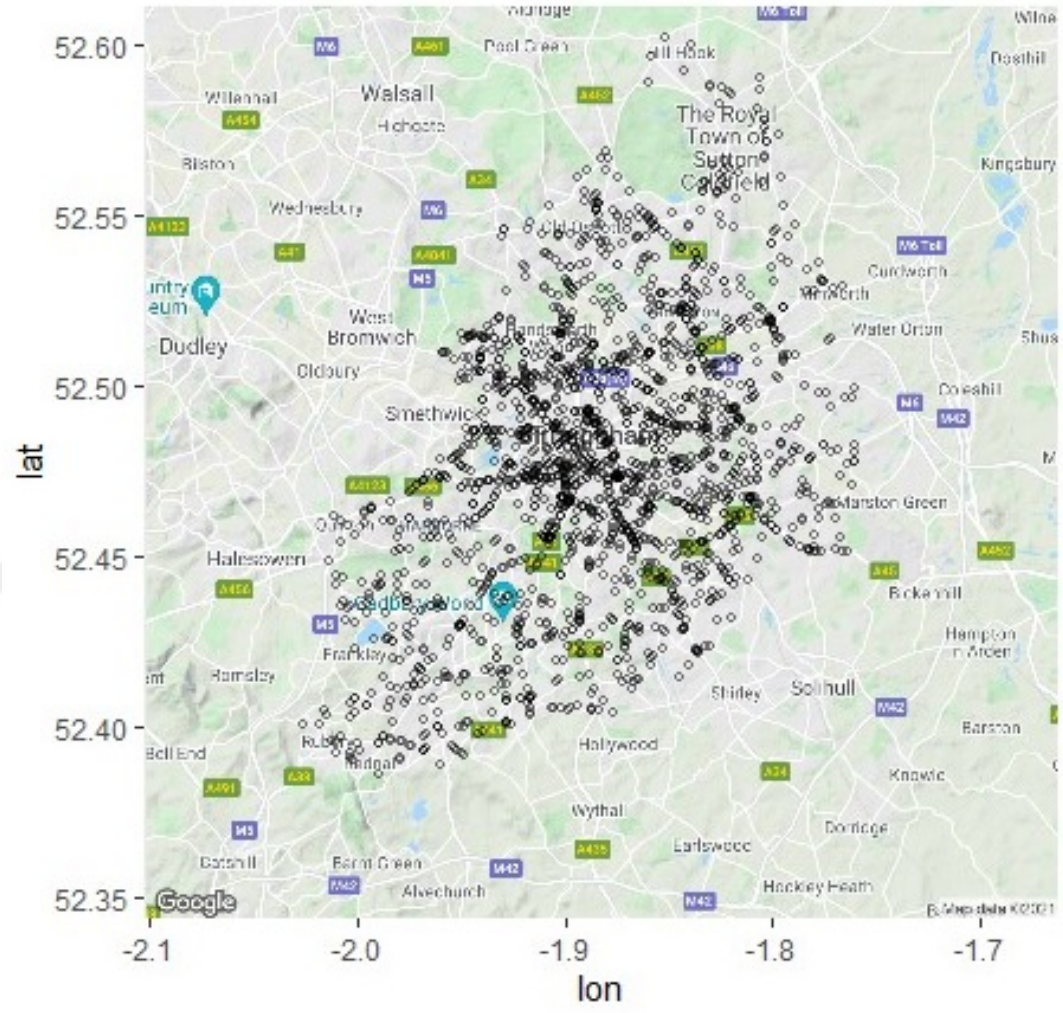
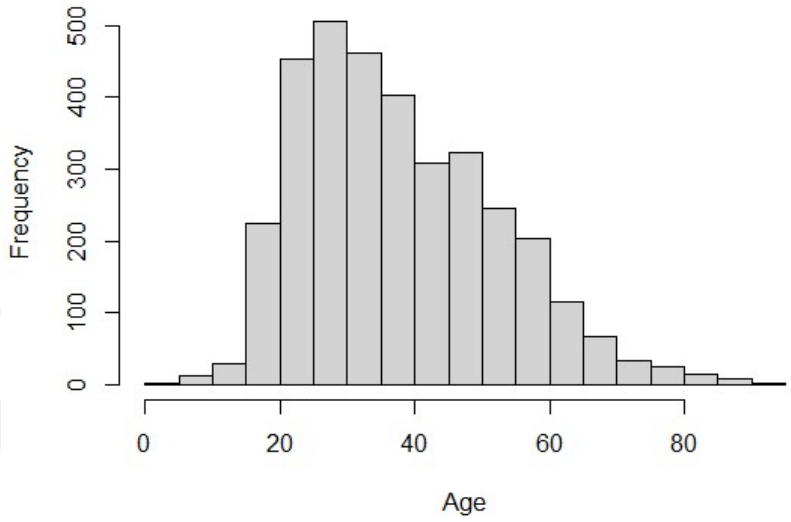


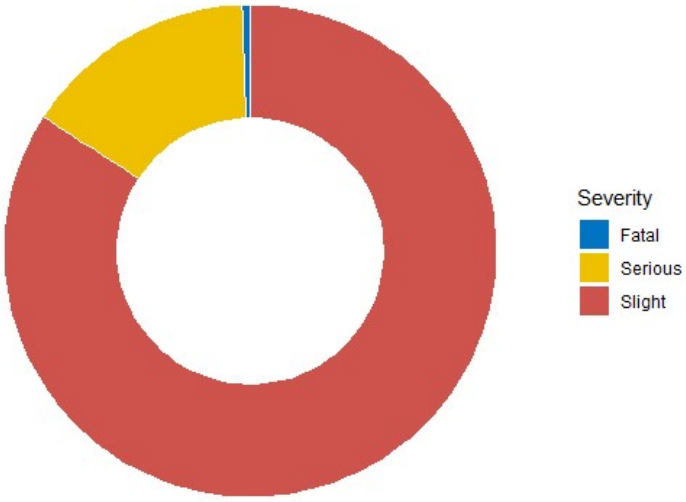
Figure 5.2 Accident locations in Birmingham.

The histogram of the age of casualties are shown in Figure 5.3 (a). The mean and median values for age of casualty is 38.25 and 36, respectively. The accidents are shown in Figure 5.3 (b) based on their severities. It can be seen that the most of the accidents resulted with slightly injuries.

Accident frequencies according to the day of the week and the time of the day are shown with Figure 5.4 (a) and Figure 5.4 (b), respectively. The number of accidents on Sunday appears to be lower (1:Sunday, … , 7:Saturday). It also appears that accidents tend to occur during business hours when people commute.

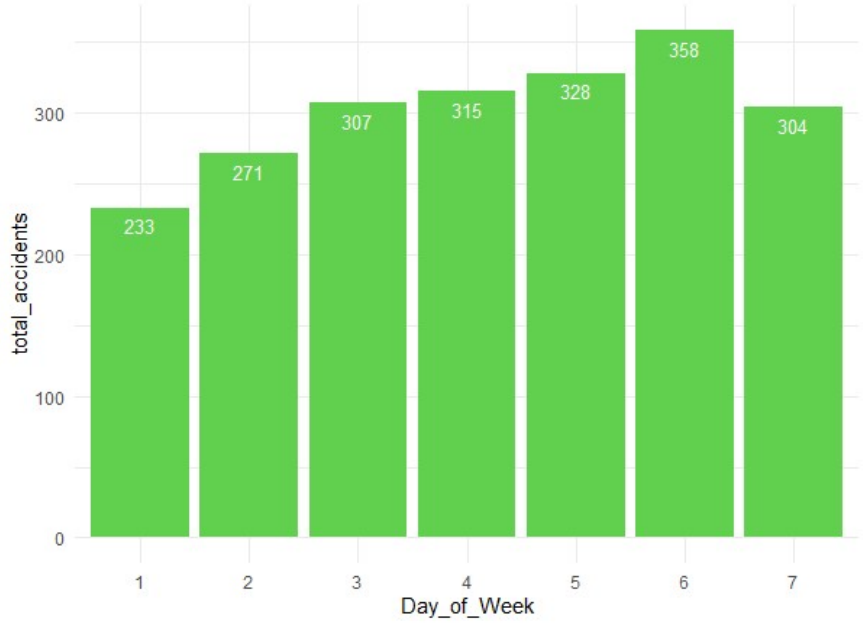


(a) Histogram of the age of casualty

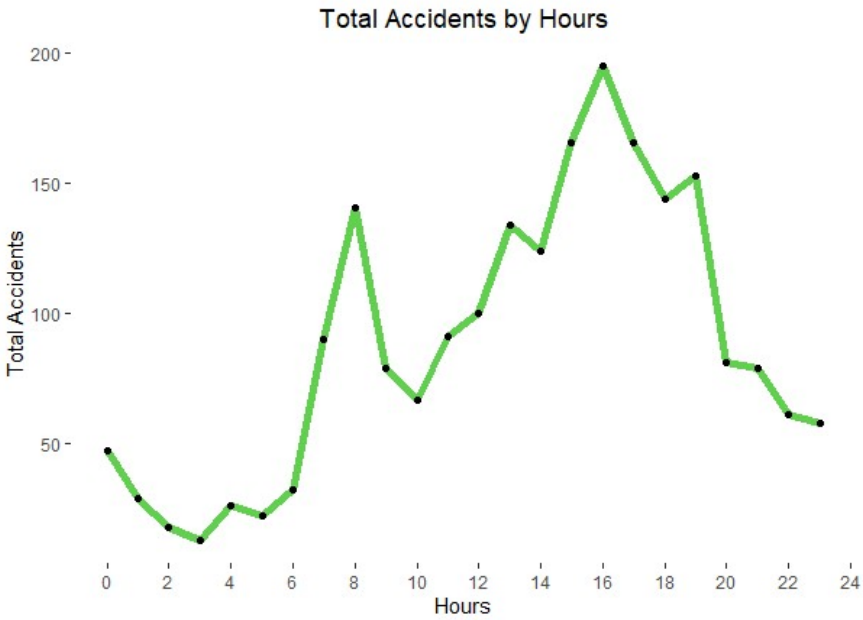


(b) Piechart of accident severity

Figure 5.3 Plots for the age of casualty and the accident severity features



(a) Number of accidents based on the day of the week



(b) Number of accidents based on the hour of the day

Figure 5.4 Plots for the day of the week and the hour features

5.1.2 Hotspot detection for Birmingham/UK dataset

In order to find the road segments and intersections where there may be a systemic safety problem, the accident locations were clustered by using the SAC method according to the number of accidents. The reasons behind hotspots are then identified using features in Table 5.1.

Among the different parameter settings for the SAC method, the appropriate parameters were determined based on the information in the accident data analysis literature. For this reason, the hit rate (HR) value, which is the ratio of the clustered points in the research area to the number of all points, was used. The HR values are displayed in Figure 5.5 and the parameter setting with the maximum HR value is selected as m=250 and c=3 for further analysis. In this way, the maximum distance between accidents is 96 meters and the number of accident criteria is 3. Accident locations are divided into 36 groups as shown in Figure 5.6 according to their frequency and spatial characteristics. There are 1884 accidents marked as noise.

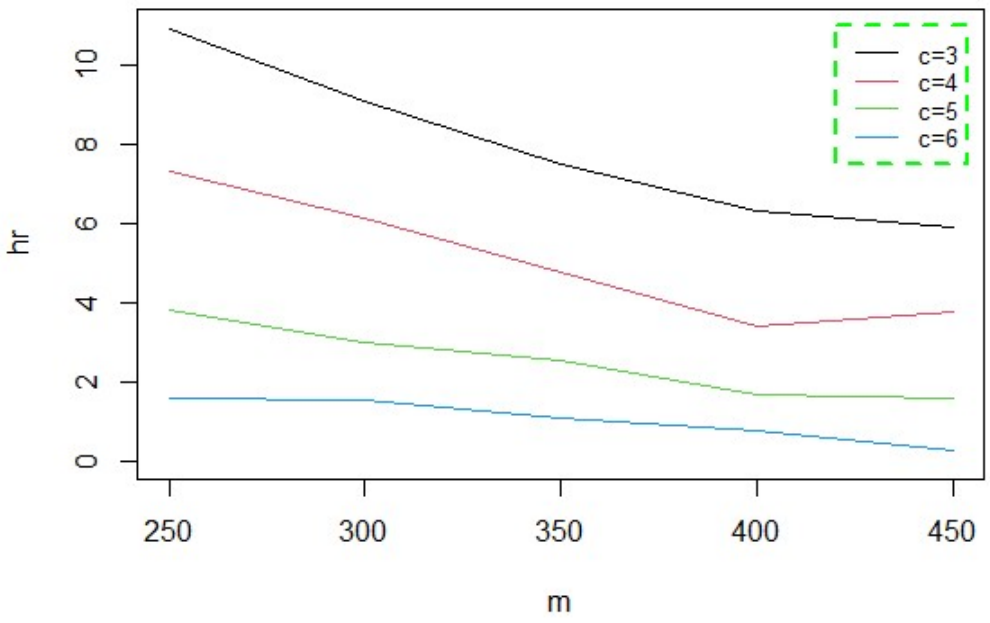


Figure 5.5 Hit rates for different settings of parameters in SAC

Figure 5.6 Detected hotspots using the SAC method

In this analysis, three of the detected hotspots were selected for further investigation. One of the hotspots obtained using SAC is an intersection denoted by Figure 5.7. It was seen that five separate traffic accidents occurred at this intersection in one year. While 11 vehicles were damaged in the accident, 8 people were slightly injured. All of the accidents occurred during the daytime and in good weather conditions. In addition, when we examined these locations in terms of intersection controls, it was observed that 4 of them did not have traffic lights. Therefore, the number of accidents can be reduced by controlling these locations with traffic lights.

Figure 5.7 The map image of an intersection hotspot obtained by SAC

Another notable hotspot in the analysis is a junction, as shown in Figure 5.8. At this point, 7 accidents occurred and 10 people were injured. The common feature of the accidents that occur here is that the accidents occur with two vehicles. When we analyzed the data at that accident location, we found that the intersection was uncontrolled, and that seems to be the potential reason behind the crashes at that hotspot.

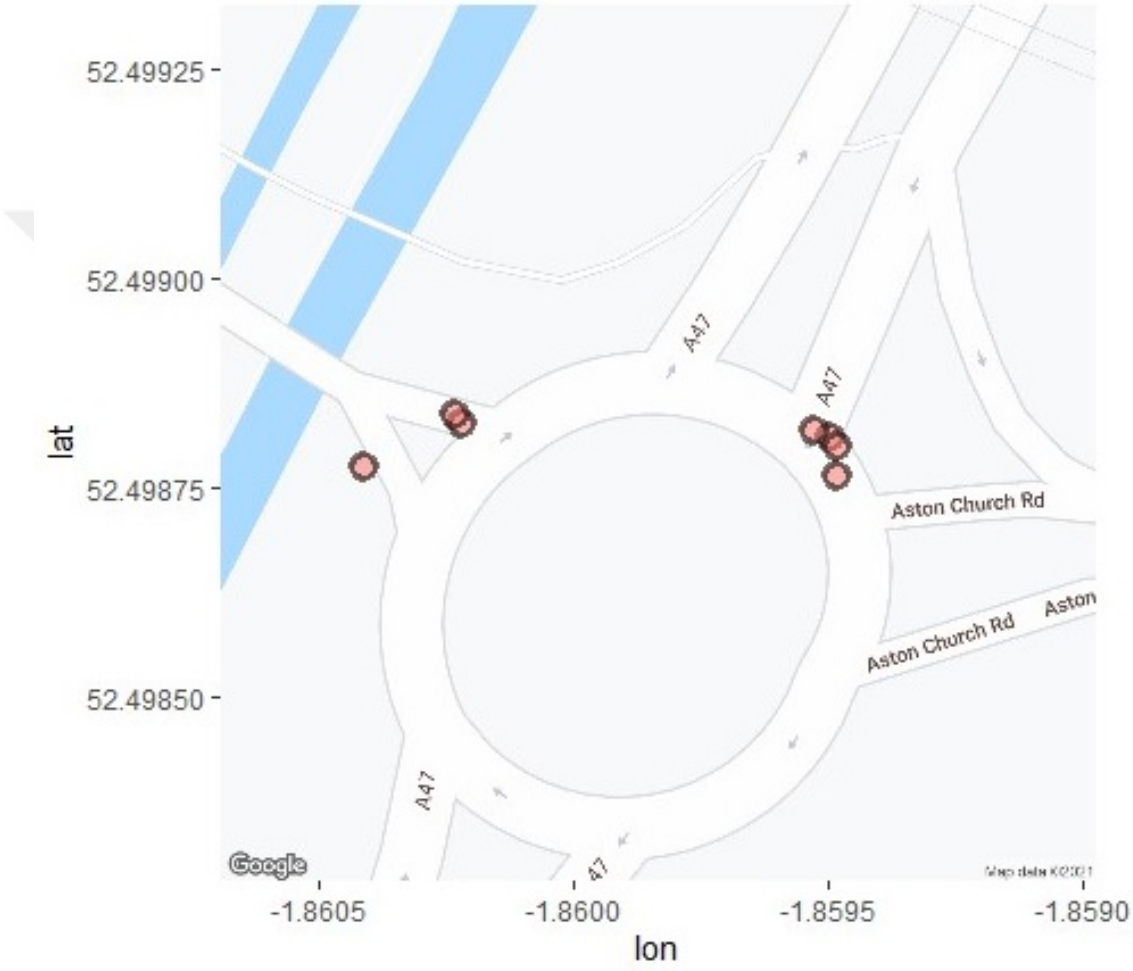


Figure 5.8 The map image of a junction hotspot obtained by SAC

Poets Corner is also determined as one of the hotspots in Birmingham, which is shown in Figure 5.9. There are five accidents with minor injury and 80% of the accidents occurred at night when the traffic lights were on. It is figured out from the data that the Poets Corner is an uncontrolled roundabout. An advertising sign for a company offering car accident management services that assists customers involved in traffic accidents caused by others has been identified at Poets corner. This supports that the number of accidents in that corner is high.

Figure 5.9 The map image of Poets Corner

5.2 Analysis of Turkey accident data

Izmir is the third largest city in Turkey and it is located in the western part. Moreover, it is one of the most important cities of the country in terms of economic, historical and socio-cultural aspects. Izmir has 11906.85 km2 areas including 30 counties namely; Aliağa, Balçova, Bayındır, Bayraklı, Bergama, Beydağ, Bornova, Buca, Çeşme, Çiğli, Dikili, Foça, Gaziemir, Güzelbahçe, Karabağlar, Karaburun, Karşıyaka, Kemalpaşa, Kınık, Kiraz, Konak, Menderes, Menemen, Narlıdere, Ödemiş, Seferihisar, Selçuk, Tire, Torbalı and Urla.

According to the Turkish Statistical Institute (Türkiye İstatistik Kurumu - TÜİK (2021)), a total of 983,808 traffic accidents occurred in Turkey in 2020. 833,533 of these accidents resulted with material damage and 150,275 of them resulted with death or injury. As a result of fatal or injured traffic accidents, 2,197 people lost their lives at the accident site, 2,669 people died within 30 days due to the cause and effect of the accident, after they were injured and transferred to the hospital. In this section, the accidents data for Izmir is analysed and the Buca district is examined in terms of hotspots.

The data set used in analysis obtained from General Directorate of Security of Turkey and it includes the road accidents in Izmir between 2018/01/01-2020/12/31. Accidents are reported to the police only if at least one person is injured or dead. In total, 24,596 accidents were recorded for Izmir between 2018–2020. The accident data set contains 44 features. Some of the features are; accident ID, longitude, latitude, date, district, junction detail, light conditions, weather conditions, number of people injured, etc. Features of the road accident data are presented in Table 5.2.

Table 5.2 Features of Izmir road accident data used in the analysis.

Feature	Subcategories	Feature	Subcategories
Accident Id	-	Lighting	Yes
Longitude	-		Yes (Broken)
Latitude	-		No
Date	-	Weather conditions	Fine
Time	-		Cloudy
Number of deaths	0-19		Rain
Number of injuries	0-36		Fog or mist
District	-		Snowing
Accident site	residential		Hail
	non-residential		High wind
Time of day	Day	Road surface conditions	Dry
	Night		Wet or damp
	Twilight		Snow
Junction detail	Not at junction		Frost or ice
	Roundabout		Flood
	Three-way (T) junction	Traffic sign	Yes
	Three-way (Y) junction		No
	Crossroads	Traffic light	Yes
	Interchange junction		Yes (Broken)
	Level crossing		No
	Other junction	Pedestrian crossing	Yes
Type of crossing	Controled railway		No
	Uncontroled railway		
	School crossing		
	Pedestrian crossing		
	Not at crossing		

5.2.1 Descriptives of the data

The number of fatal or injury accidents recorded in İzmir in 2018, 2019 and 2020 is 8788, 8388 and 7420, respectively. It is decreasing over the years, as shown in Figure 5.10. When we examine the number of accidents according to the days of the week, Figure 5.11 shows a decrease on an annual basis. The colors green, orange and purple represent 2018, 2019 and 2020, respectively. In addition, the days of the week, from Sunday to Saturday, are numbered from 1 to 7 on the horizontal axis. It is seen that the number of accidents is less on Sunday. While a large number of accidents occurred on Saturdays in 2018 and 2019, there seems to be a decrease in 2020. The reason for this decrease may be weekend curfews during the Covid-19 pandemic that started in 2020.

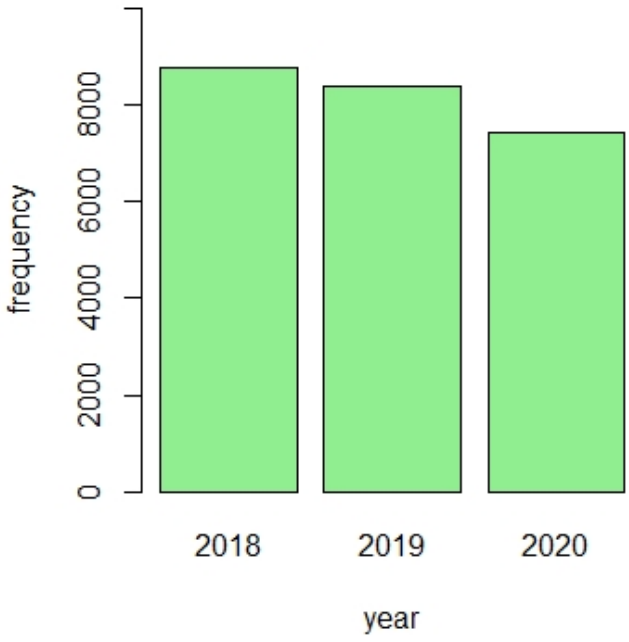


Figure 5.10 Number of accidents by year

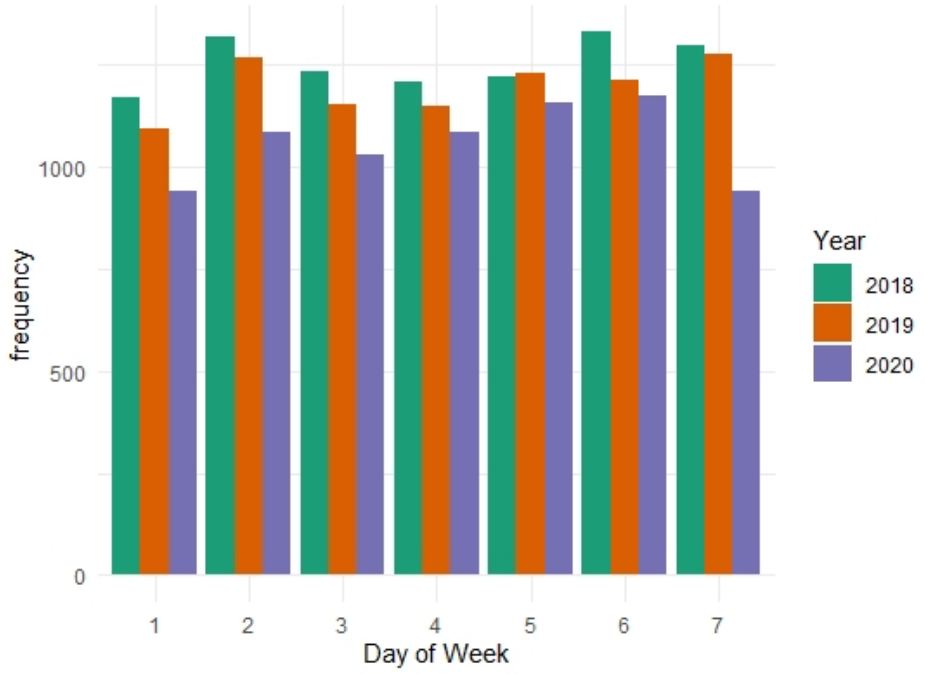


Figure 5.11 Number of accidents by day of the week

When we analyze the distribution of accidents by months and years given in Figure 5.12, we can say that there is a serious decrease in the number of accidents in March and especially in April and May in 2020. The reason for this is the serious restrictions and closures implemented in İzmir on these dates due to the pandemic. In general, it is noted that more accidents occur in the spring and summer seasons. Also, it is observed from the Figure 5.13 that accidents tend to occur more on the business hours when people commute to work.

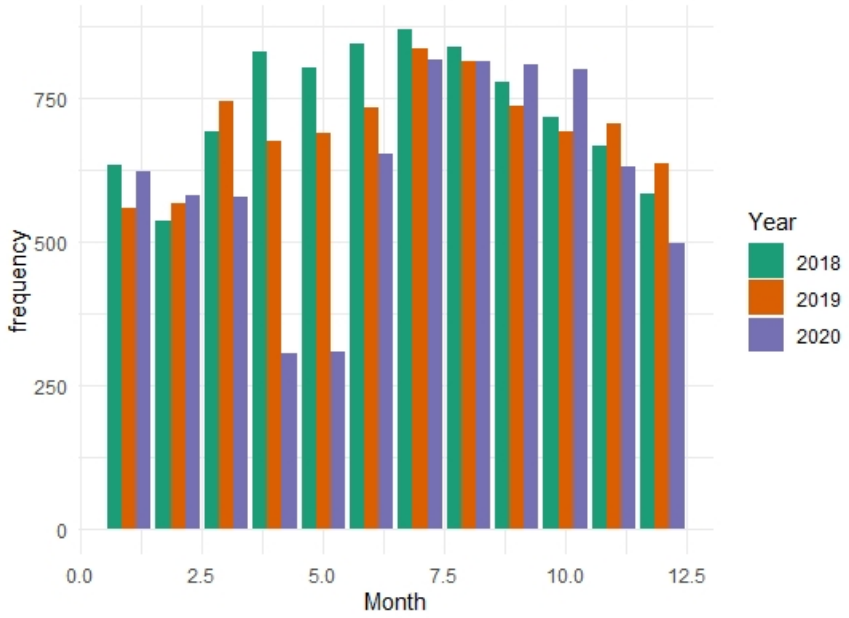


Figure 5.12 Number of accidents by month

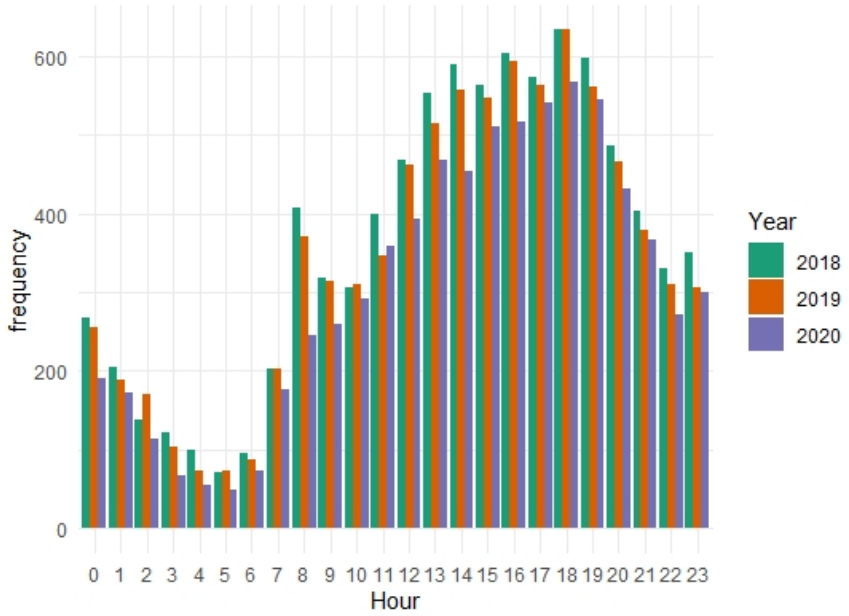


Figure 5.13 Number of accidents by hour

In order to identify the problematic districts in Izmir, the accident frequencies examined as shown in Figure 5.14. The three districts with the highest number of accidents are Konak, Bornova and Buca in descending order. The accident locations are shown on the map given in Figure 5.15.

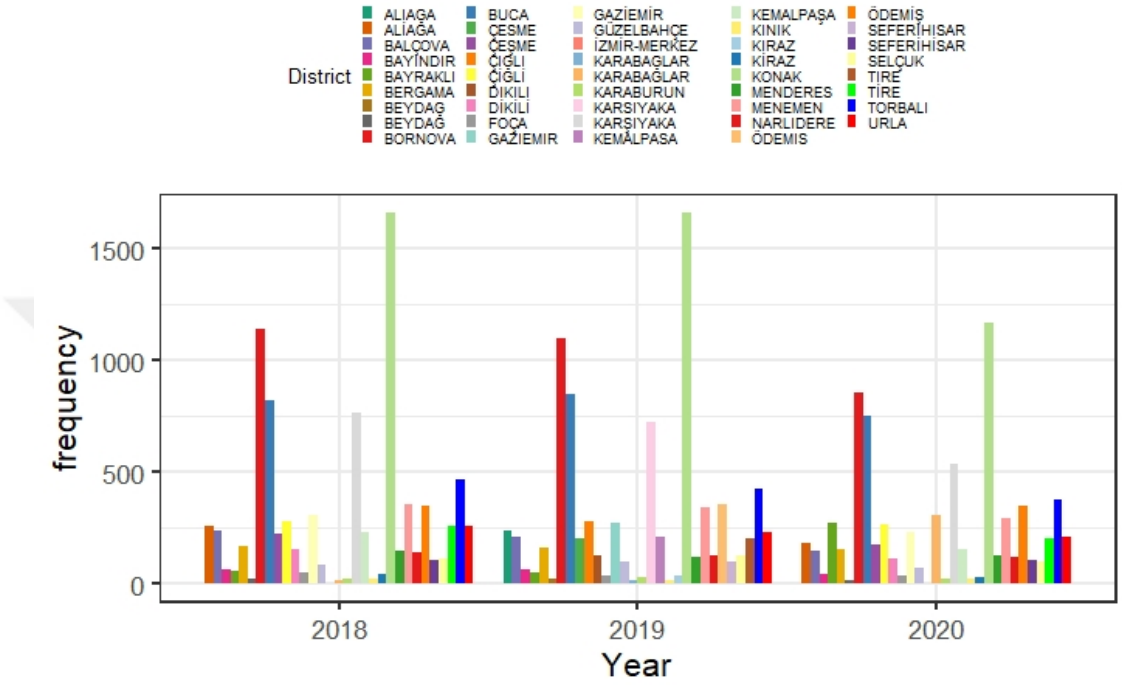


Figure 5.14 Accident frequency of districts by years.

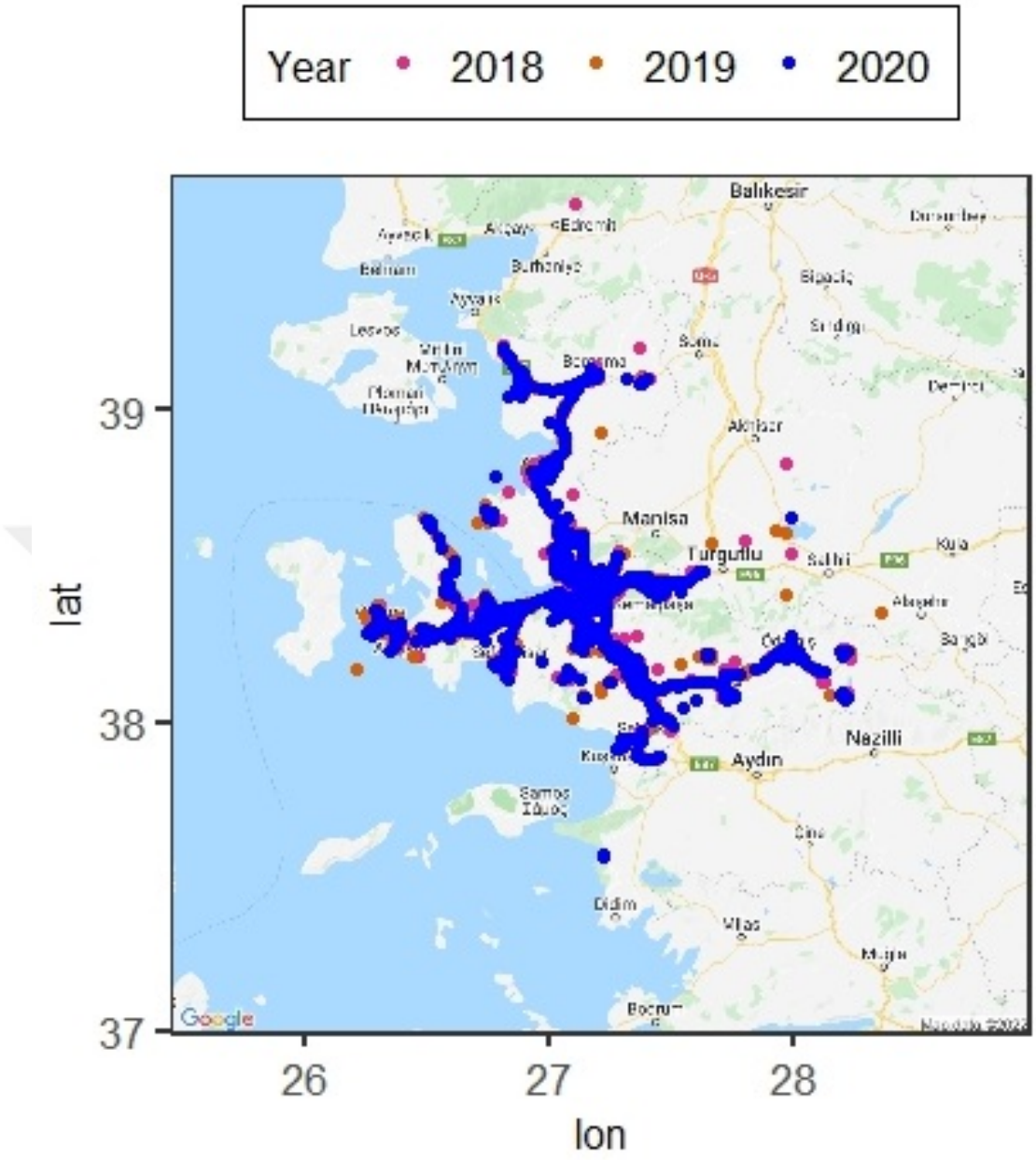


Figure 5.15 Accident points in Izmir for three years period.

5.2.2 Descriptives for Buca/Izmir dataset

In this section, we analysed the accident data in 2019-2020 for Buca, one of the districts with the highest number of accidents, to identify possible problematic locations. Accident locations in Buca are shown in Figure 5.16 and Figure 5.17 for 2019 and 2020, respectively. The hourly accident numbers are examined from Figure 5.18, where the pink and blue lines represent the number of accidents in 2019 and 2020, respectively. It is seen that there are three peak hours in 2019. However, a significant decrease is observed in the morning and night peak hours in 2020 compared to 2019.

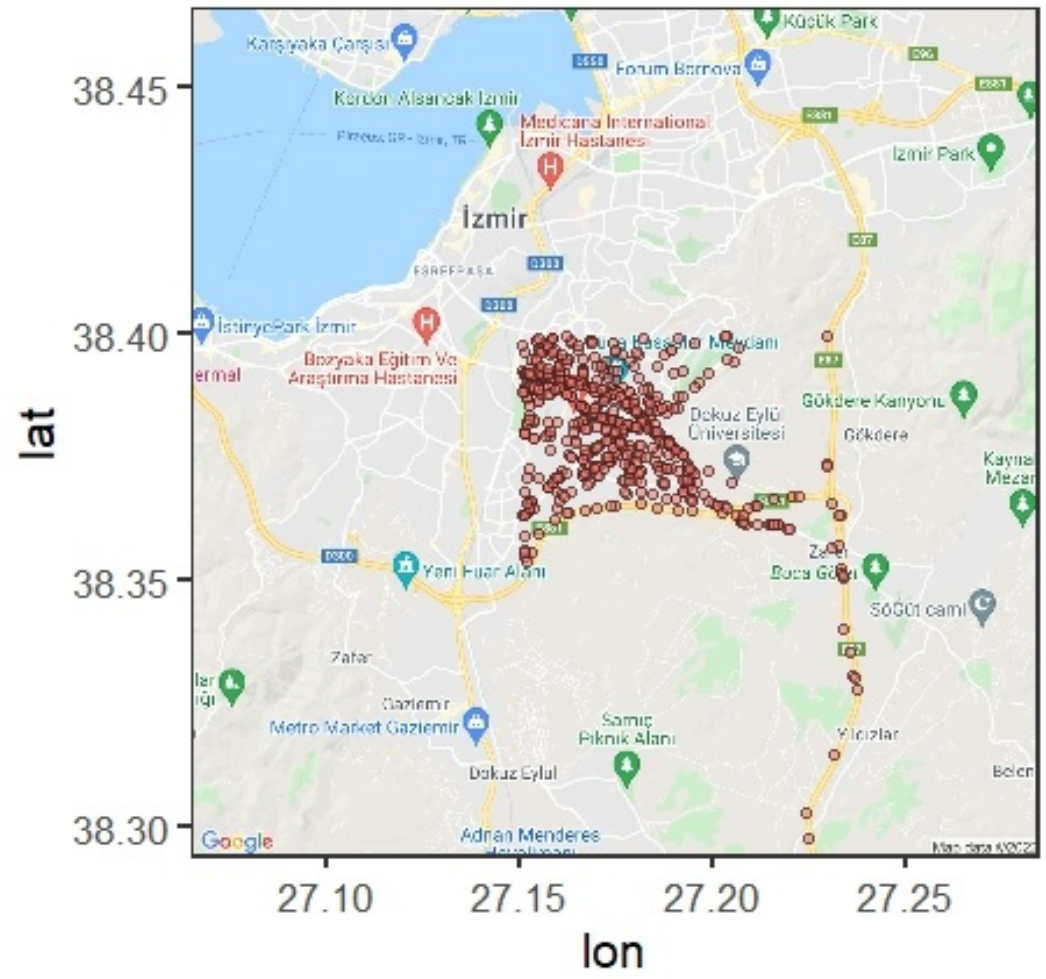


Figure 5.16 Accident locations in Buca in 2019

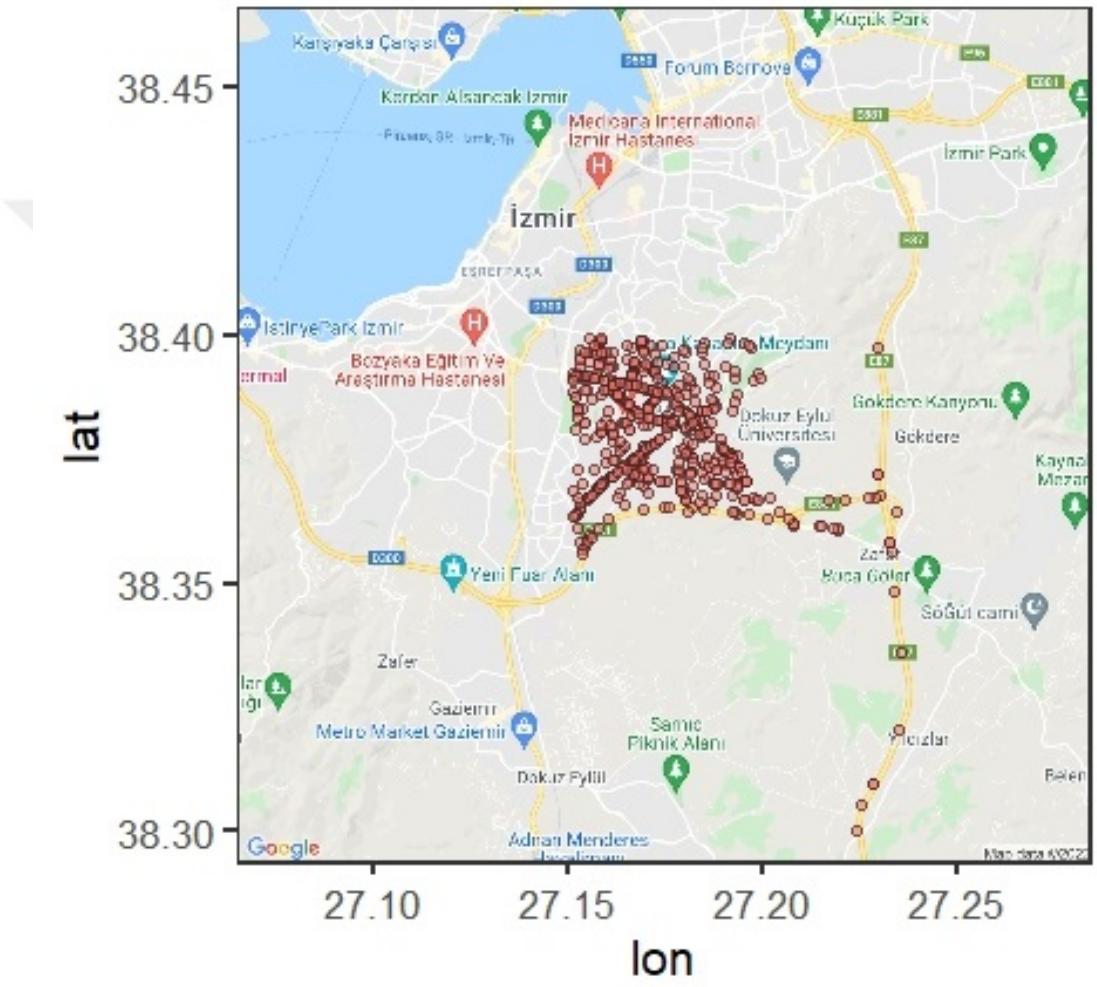


Figure 5.17 Accident locations in Buca in 2020

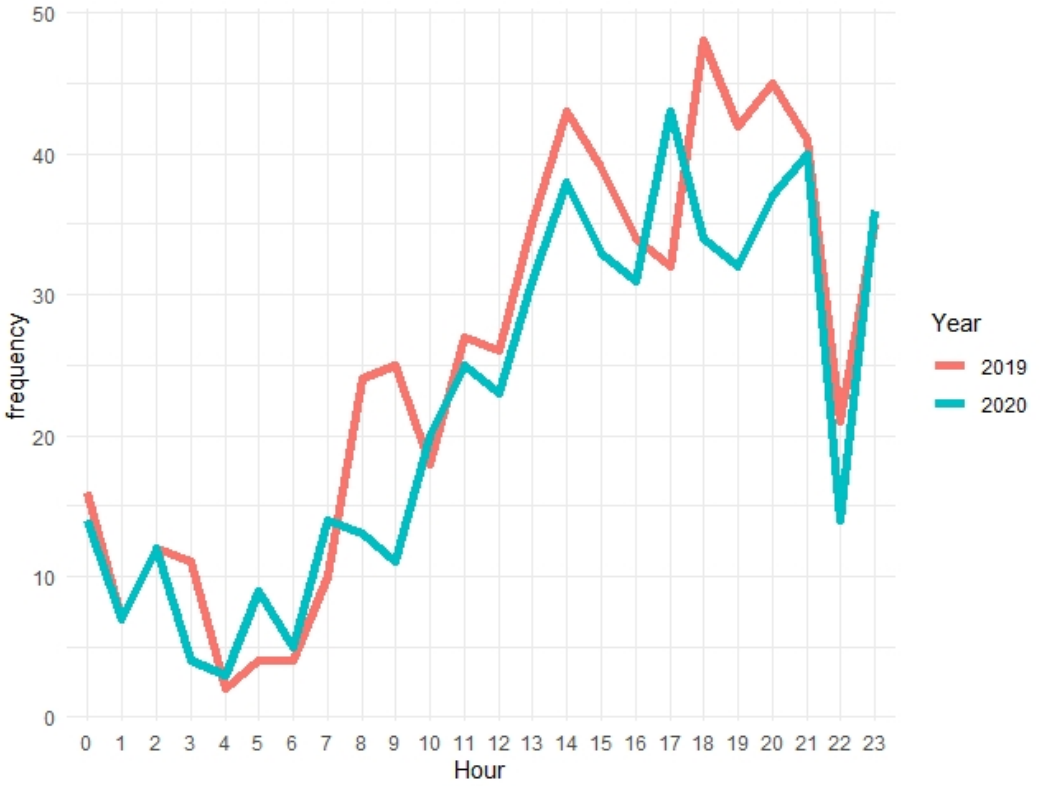


Figure 5.18 Hourly accident frequencies in Buca for 2019 and 2020

The locations of the accidents that occurred at night were examined according to the lighting conditions. There are 30 accidents in 2019 and 45 accidents in 2020 in locations where there is no lighting at night time or where the lighting is broken. The locations where the accidents occur at night and where there are lighting problems are shown on the map as Figure 5.19 and Figure 5.20, respectively. The locations marked in orange on the map show the accidents occurred in 2019, while the locations marked in blue represent the accidents occurred in 2020. From Figure 5.19, it is seen that there is an increase in the number of accidents occurring at night in places where there is no lighting in 2020. In addition, another common feature of these accidents is the absence or broken traffic light. These accidents led to serious losses with 1 dead and 91 injured. Lighting these areas and controlling them with traffic lights can reduce the number of accidents.

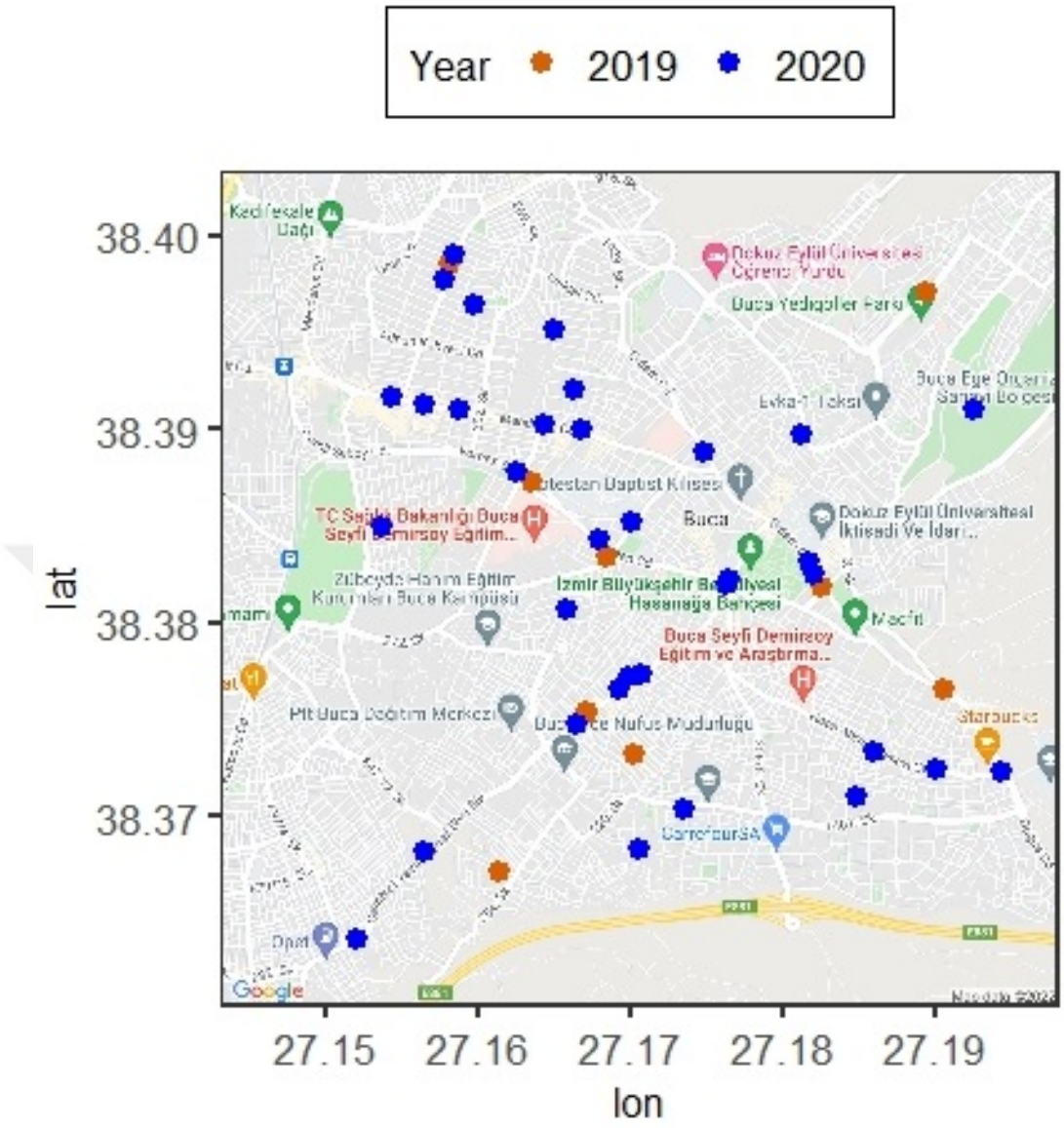


Figure 5.19 Accident points at night without lighting in Buca for 2019 and 2020

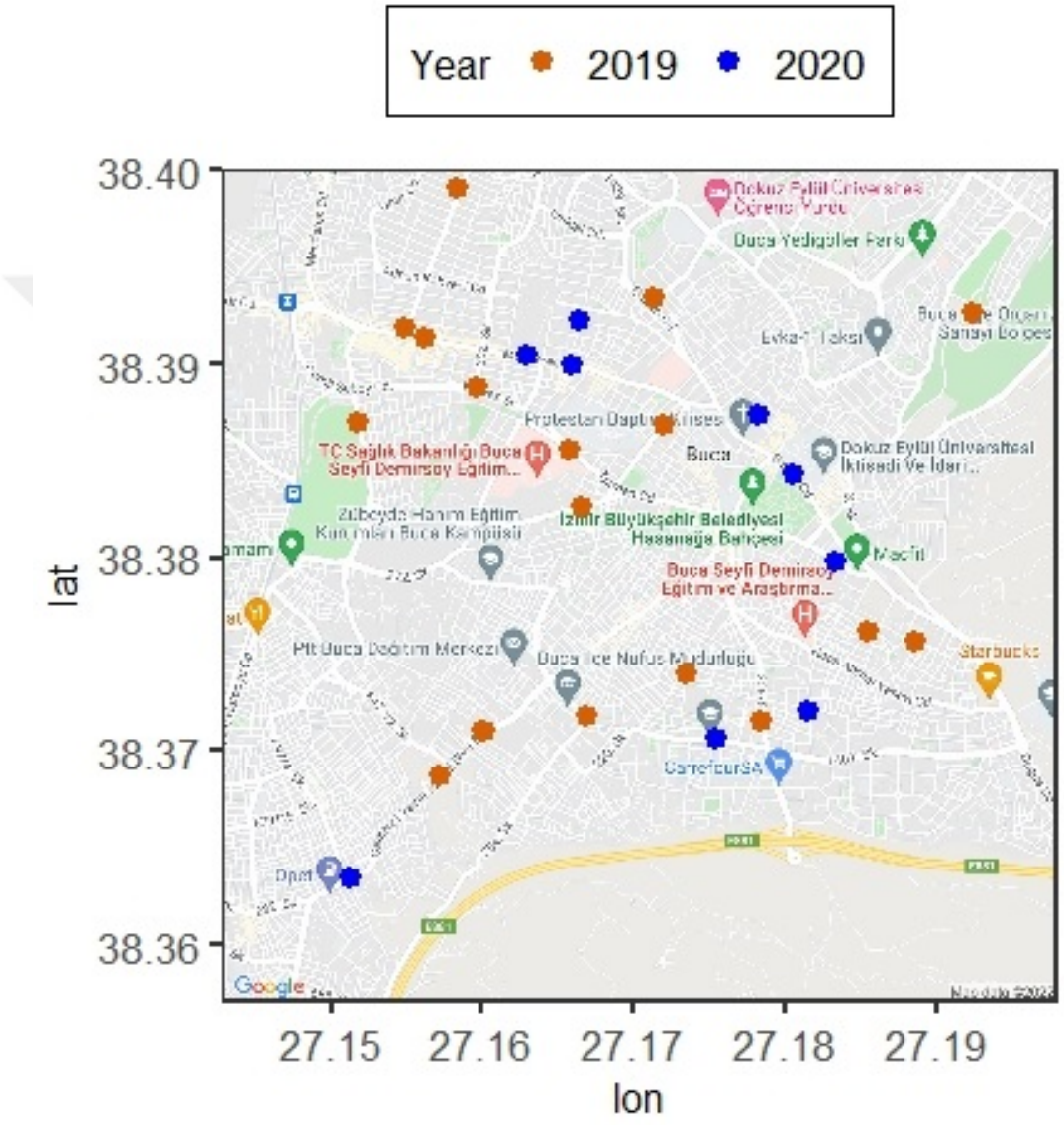


Figure 5.20 Accident points at night when the lighting is broken in Buca for 2019 and 2020

When the accident locations are examined according to the road junction types as in Figure 5.21, it is seen that the accidents mostly occured in the locations where there are no intersections. In addition, the number of accidents increased for three-way (T) and four-way junction locations in 2020. Also, we examined the Three Way (T) junction accidents according to the traffic light conditions as shown in Figure 5.22. It is seen that there is a serious increase in the number of accidents in locations where there is no traffic light in 2020 compared to 2019. It can be suggested that placing traffic lights in Three-Way (T) areas where it is determined that there are no traffic lights can reduce the number of accidents.

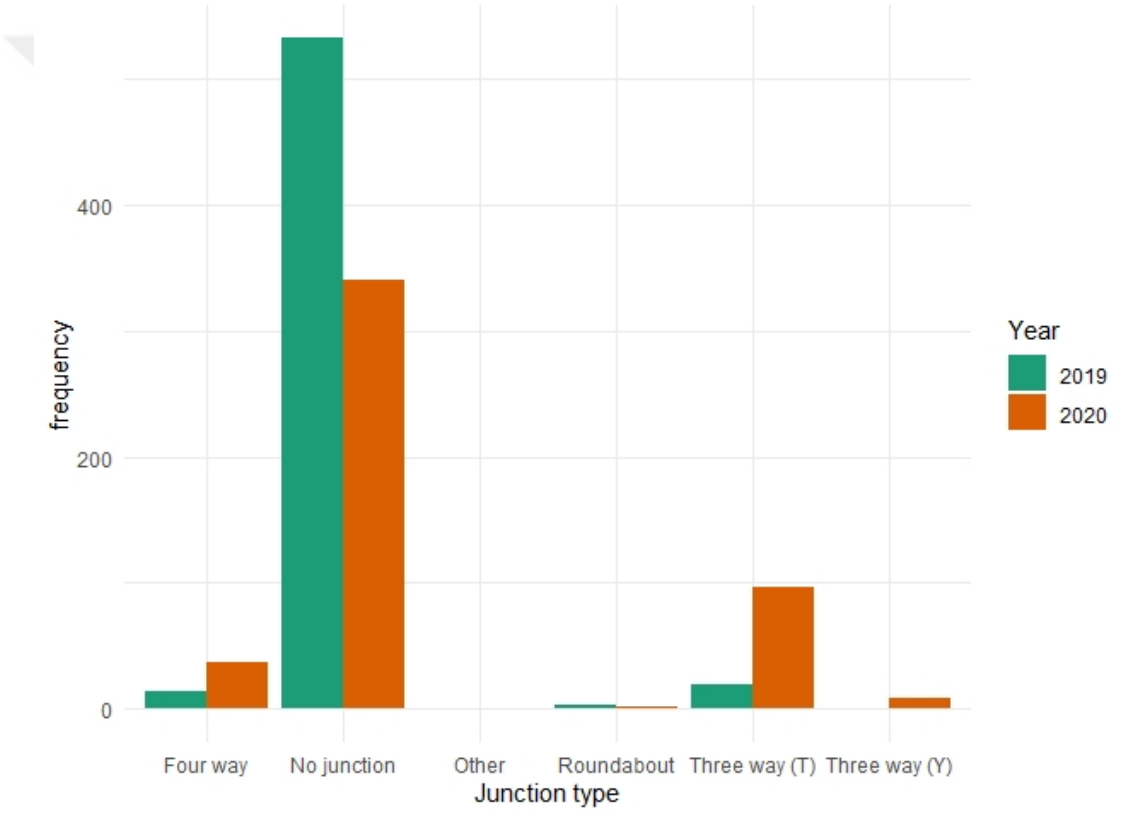


Figure 5.21 Accidents based on junction type

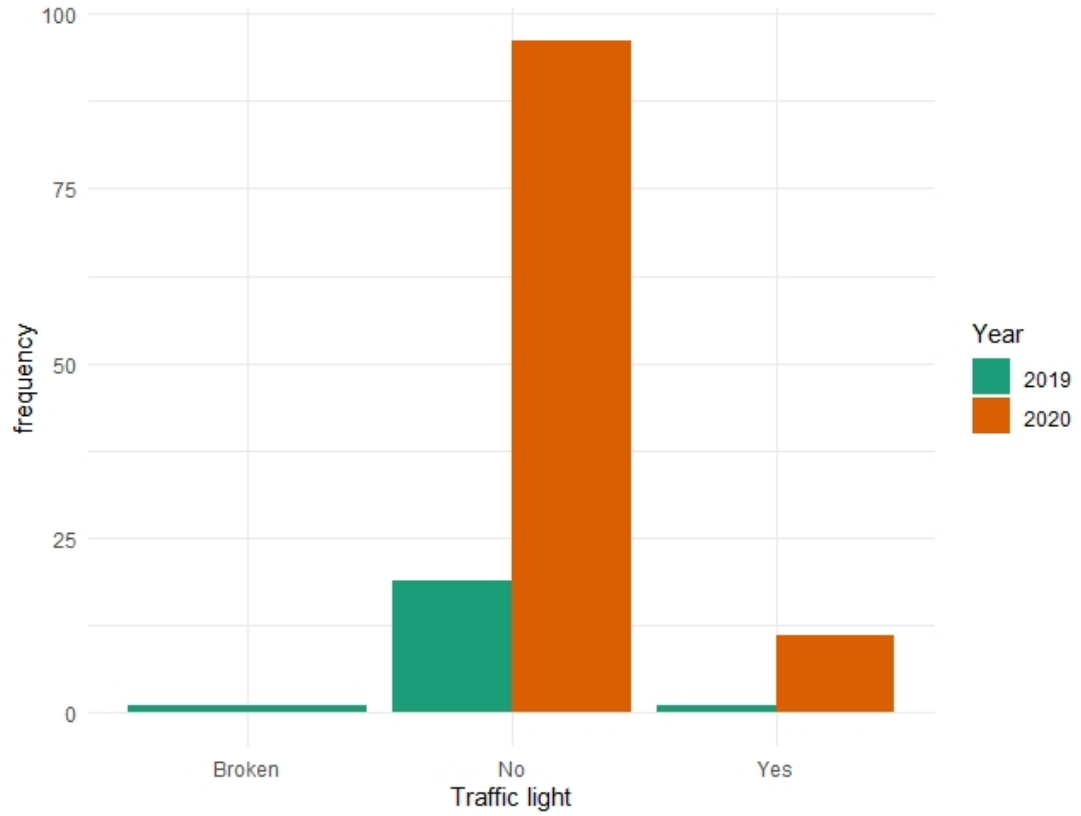


Figure 5.22 Three Way (T) junction accidents for traffic light condition

5.2.3 Hotspot detection for Buca/Izmir dataset

In this study, the accident locations are clustered using SAC method based on the number of accident in order to find road segments and intersections where a systemic safety problem may be at fault. Furthermore, the reasons behind the accident hotspots are detected using other features in Table 5.2. Accident data are clustered according to latitude and longitude features and accident frequencies in the locations.

The HR value is used in order to determine appropriate parameters among different settings of parameters in SAC method. HR values based on different parameter settings of SAC are shown in Figure 5.23 and Figure 5.24 for 2019 and 2020, respectively. The parameters of the method with the maximum HR value is m=130 and c=3 which used for further analysis for both years. By this way, the maximum distance between accidents is 80 meters and the number of accident criteria is 3.

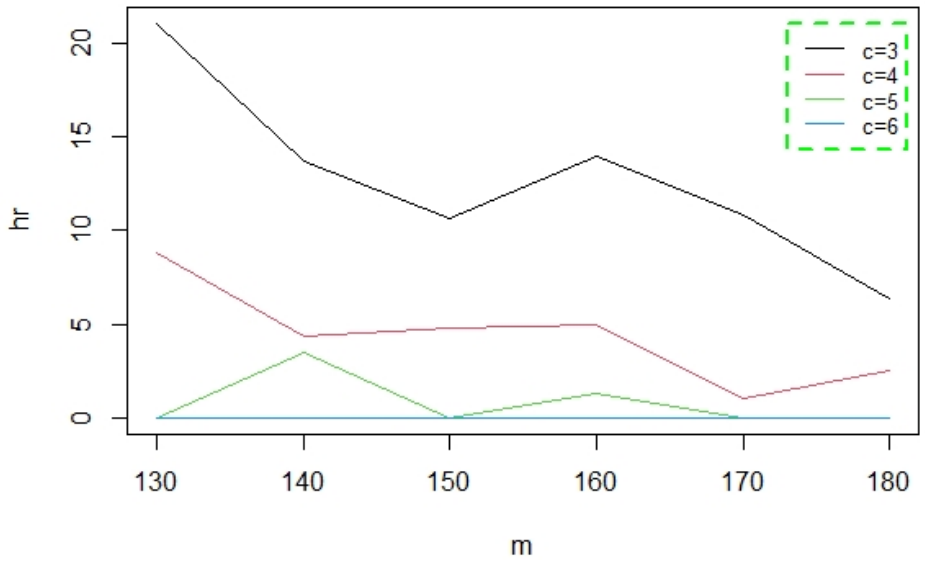


Figure 5.23 Hit rates for different setting of parameters in SAC for 2019

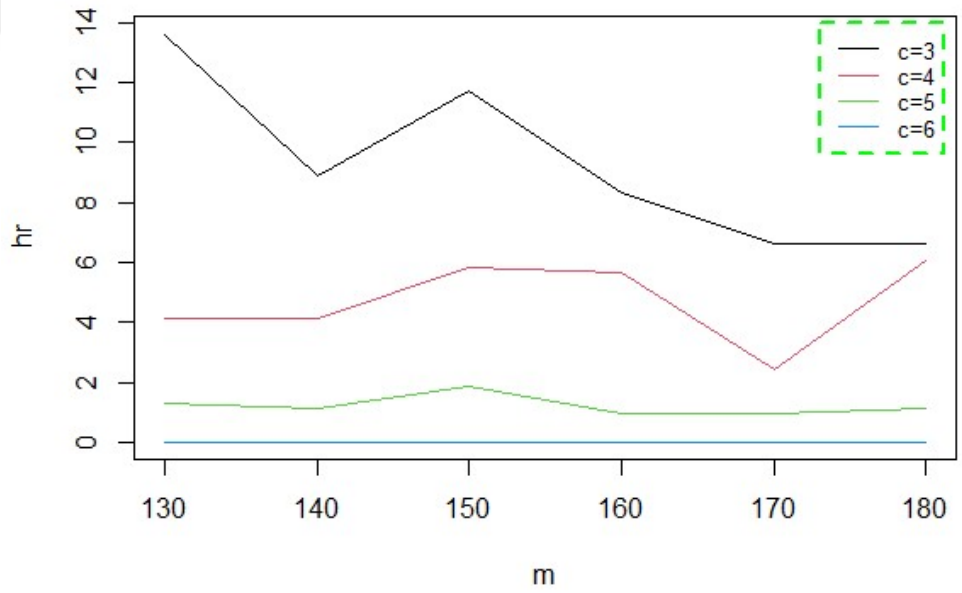


Figure 5.24 Hit rates for different setting of parameters in SAC for 2020

The accident locations in 2019 divided into 21 clusters based on their accident frequency and spatial features as shown in Figure 5.25. There are 475 accidents marked as noise. The accident locations in 2020 are divided into 13 clusters as shown in Figure 5.26. For 2020, there are 457 accidents marked as noise. In Figure 5.27, the accident locations are marked on the map using orange and blue colors for 2019 and 2020, respectively. It is seen that, the number of hotspot locations decreased in 2020 and the location of the problematic zones are changed.

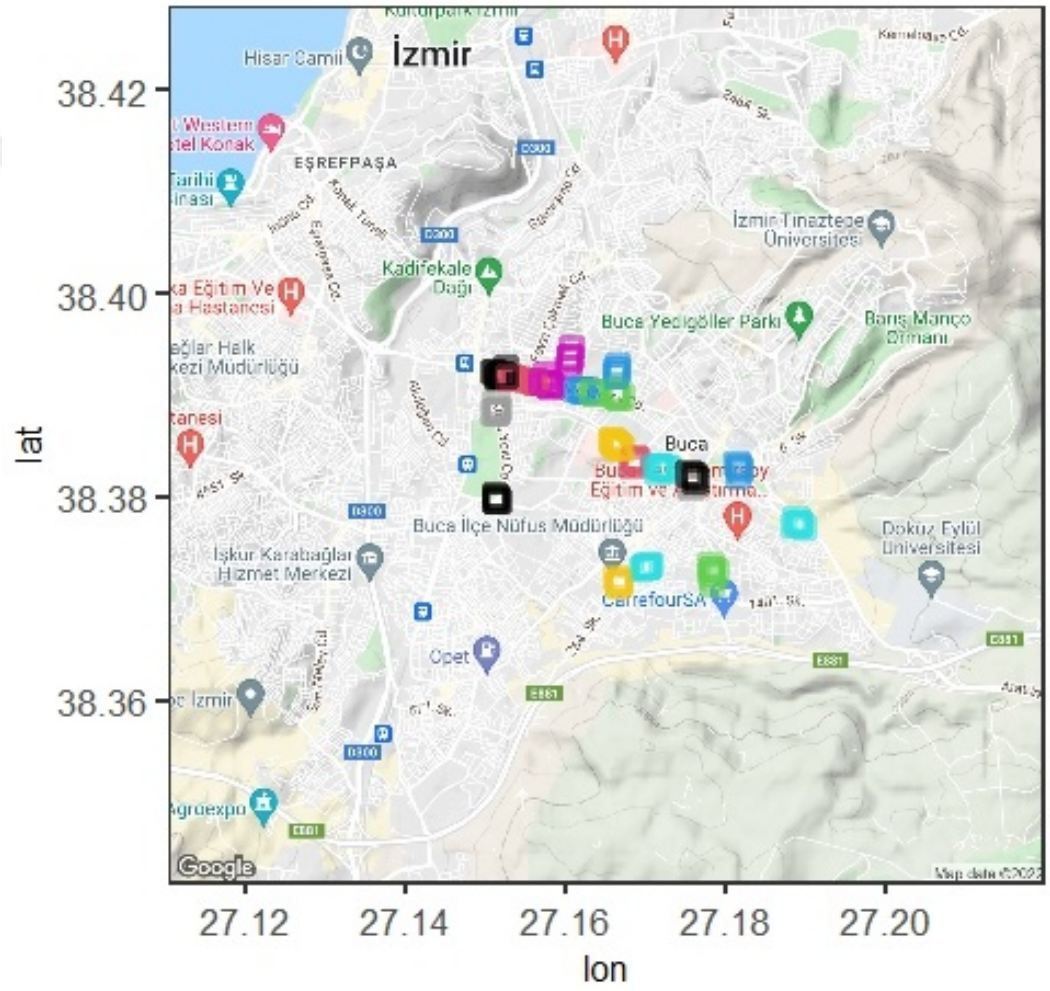


Figure 5.25 Detected hotspots using SAC for 2019

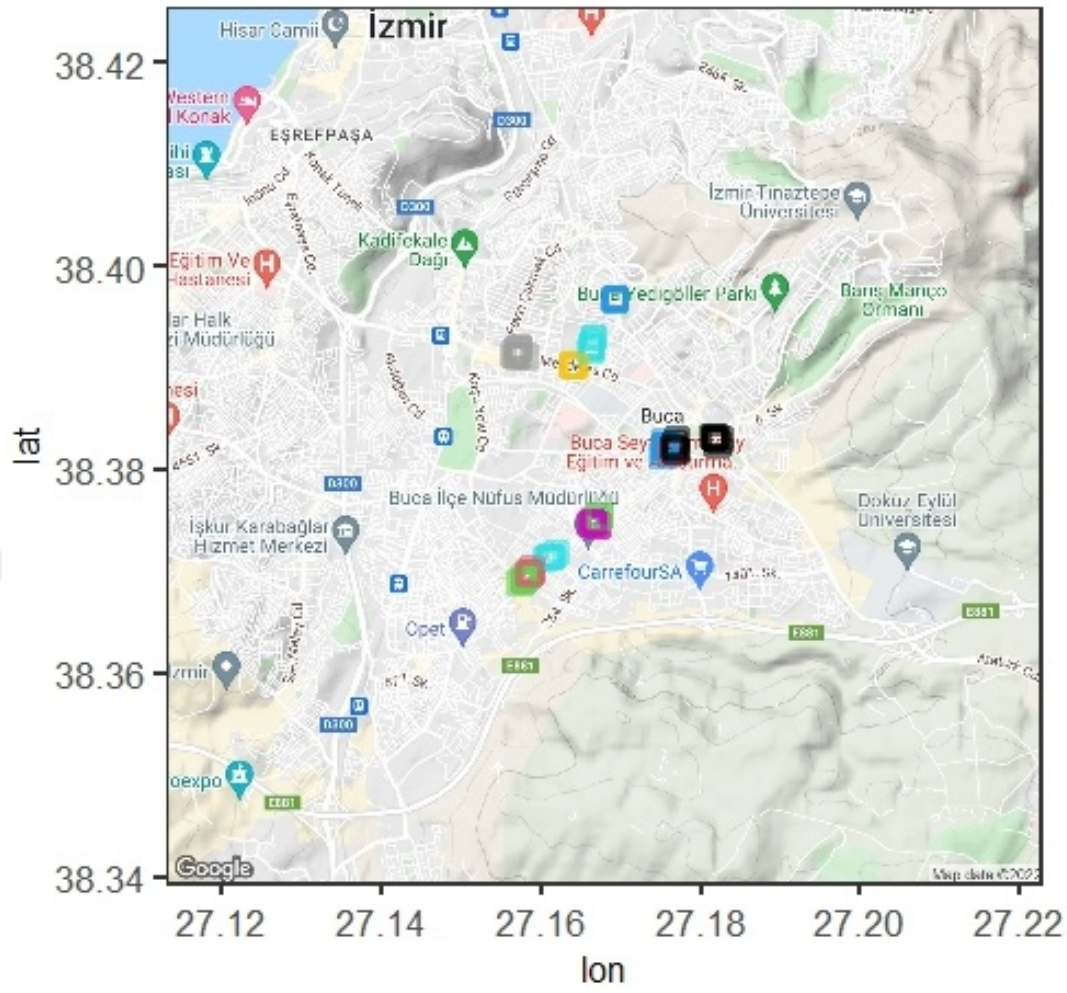
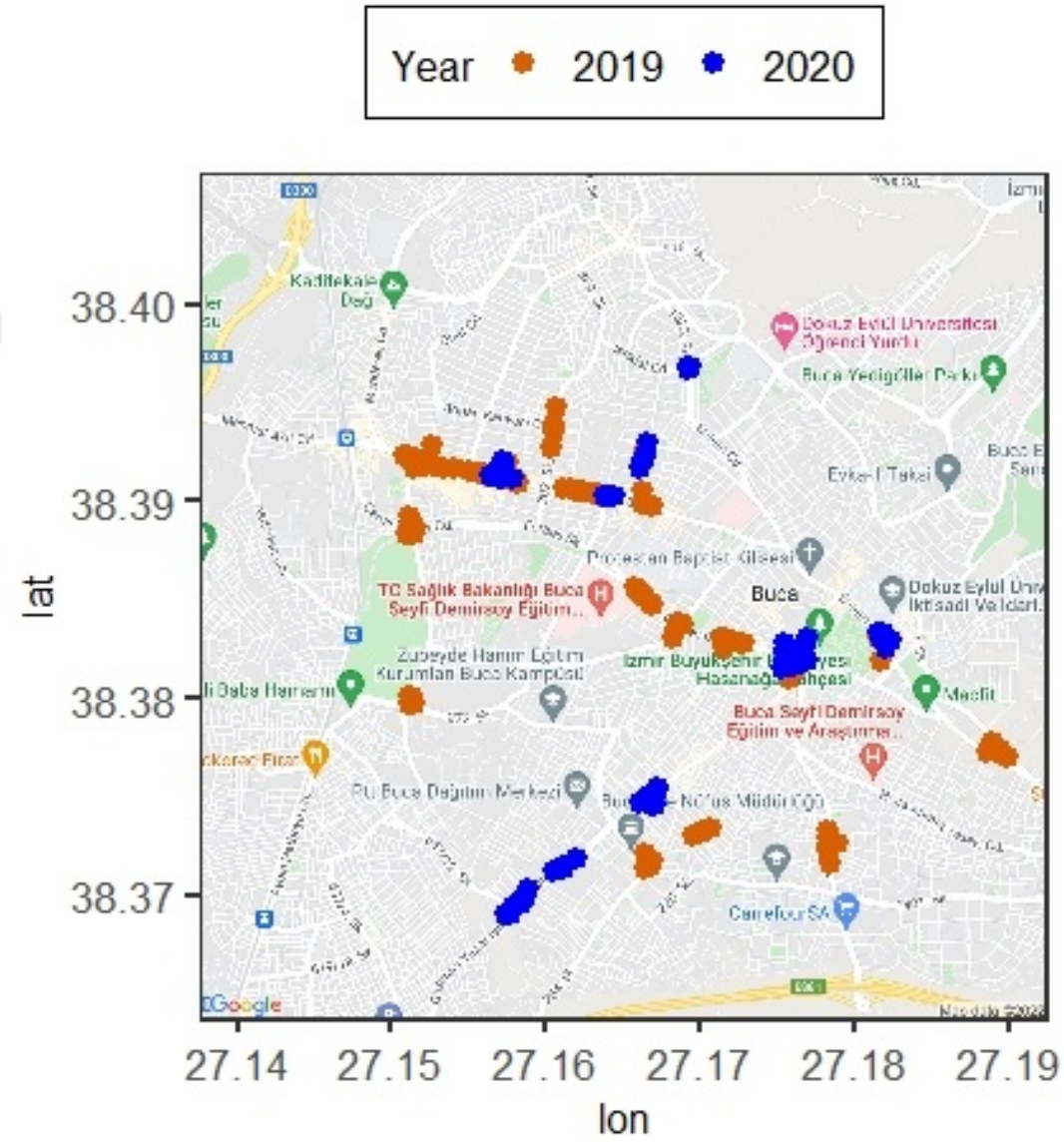


Figure 5.26 Detected hotspots using SAC for 2020



(a) 2019

Figure 5.27 Hotspot locations for 2019 and 2020

Some of the hotspots detected in the 2020 accident data were further examined. Üçkuyular is one of the hot spots obtained using SAC, and the accident sites that occurred here are shown in Figure 5.28. All of the accidents occurred during the daytime and when the weather conditions were good. There are four separate car accidents in that hotspot and 4 people injured. Moreover, the accidents in this hotspot caused the pedestrian injuries. Also, we investigate this locations in terms of junction controlls and it is observed that there is no traffic lights on 3 of them. Therefore, in that hotspot the number of accidents can be decrased by controlling the locations with traffic lights.

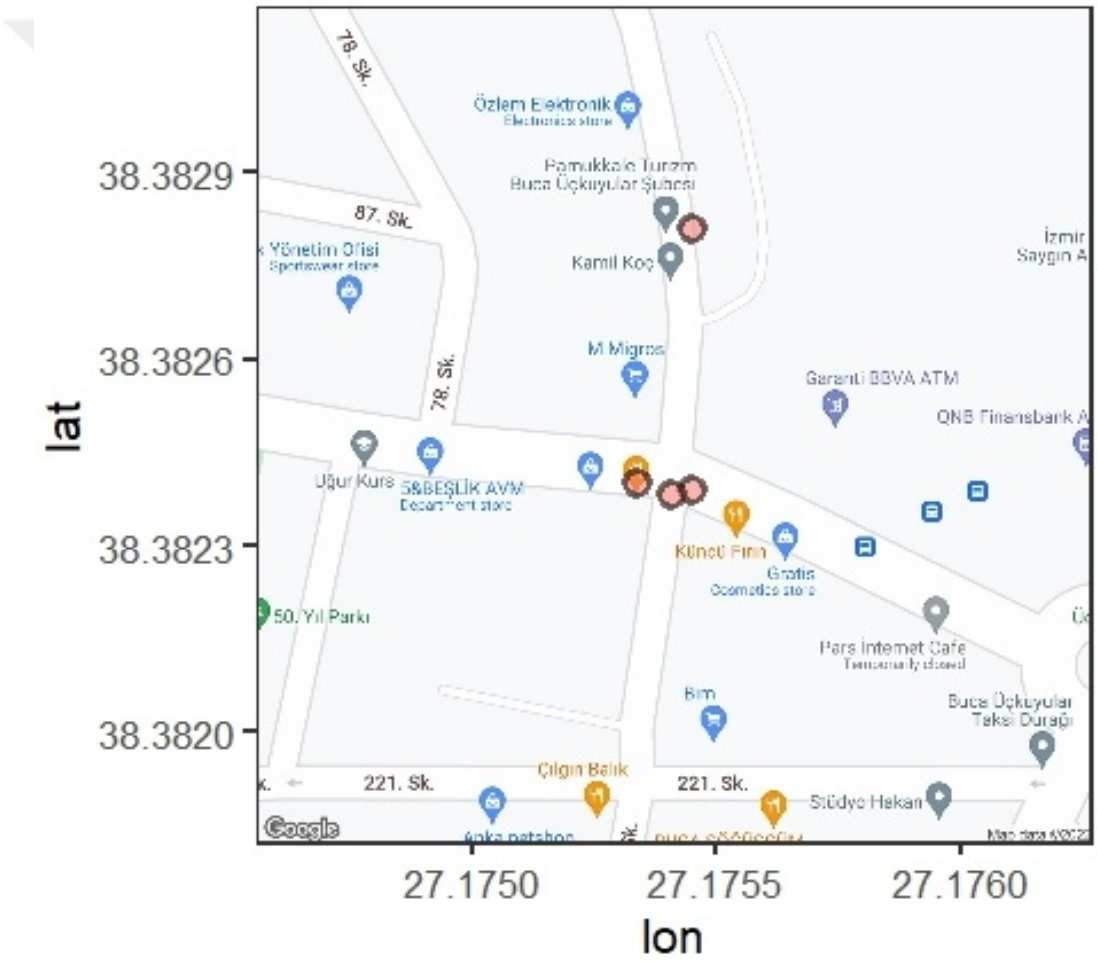


Figure 5.28 A hotspot in Üçkuyular obtained by SAC

Another hotspot shown in Figure 5.29 is a junction in Üçkuyular. There were 8 accidents in this region and it was determined that 6 of these accidents were pedestrian crashes. When we examined the region in more detail, we found that the intersection was uncontrolled, which seems to be the factor behind the accidents in that hotspot. Also, there is no pedestrian crossing at this location. Turning this intersection into a controlled intersection and providing a pedestrian crossing will reduce the number of accidents at this hot spot.

Figure 5.29 A junction in Üçkuyular obtained by SAC

Another junction determined as a hotspot in the study is located at Evka-1/Buca, which is shown in Figure 5.30. There were five accidents with injuries and 80% of the accidents occurred in day time. It is seen from the data that this hotspot is an uncontrolled roundabout.

Figure 5.30 A hotspot in Evka-1 obtained by SAC

Another hot spot was detected in the Dokuzçeşmeler region shown in Figure 5.31. There were 7 accidents with 9 injuries. The main feature of the location is that there are no traffic signs or traffic lights.

Figure 5.31 A hotspot in Dokuzçeşmeler obtained by SAC

CHAPTER 6
MODELING HEADWAY DATA USING EXPONENTIATED WEIBULL DISTRIBUTION BASED ON RANKED SET SAMPLING

Modeling the vehicle headway data is fundemental for intelligent transportation applications in traffic engineering. It is useful for the traffic signal optimization and flow modelling. Exponentiated Weibull (EW) is one of the best flexible model for characterizing uncertainty in various fields of data. In this chapter, we provide the results for the estimation of the unknown parameters for EW distribution using ranked set sampling (RSS) method. We present the results of the Monte Carlo simulation study in order to compare the performance of the simple random sampling (SRS) and RSS methods. We analyse the data given in the study of Badhrudeen et al. (2016) for modeling a vehicle headway data using EW distribution.

6.1 Exponentiated Weibull (EW) Distribution

Mudholkar & Srivastava (1993) introduced the exponentiated Weibull (EW) distribution as a generalization of the Weibull distribution. EW distribution has a second shape parameter, and it is commonly used in modeling of data in various fields, such as reliability, finance, medicine and environmental studies. The detailed information can be found in Mudholkar & Srivastava (1993), Pal et al. (2006), Shittu & Adepoju (2014) and additional results can be found in Elshahhat (2017), Ghnimi & Gasmi (2014) for modelling with EW distribution.

The probability density function (pdf) and cumulative distribution function (cdf) of the EW family is given Mudholkar & Srivastava (1993);

$$g(y;\alpha,\beta,\lambda) = \alpha\beta\lambda^{\beta}y^{\beta-1}e^{-(\lambda y)^{\beta}}(1-e^{-(\lambda y)^{\beta}})^{\alpha-1}, \tag{6.1}$$

$$G(y;\alpha,\beta,\lambda) = (1-e^{-(\lambda y)^{\beta}})^{\alpha}, \;\; 0 < y < \infty, \tag{6.2}$$

where $\lambda > 0$ is the scale parameter, and, $\alpha > 0$ and $\beta > 0$ are referred to as the shape parameters. The corresponding hazard function is

$$r(y;\alpha,\beta,\lambda) = \frac{\alpha\beta\lambda^{\beta}y^{\beta-1}e^{-(\lambda y)^{\beta}}(1-e^{-(\lambda y)^{\beta}})^{\alpha-1}}{(1-e^{-(\lambda y)^{\beta}})^{\alpha}}. \tag{6.3}$$

The hazard function has different shapes depending on parameter values, such as monotone increasing ($\beta \geq 1$, $\alpha\beta \geq 1$), monotone decreasing ($\beta \leq 1$, $\alpha\beta \leq 1$), bath-tube shaped ($\beta > 1$, $\alpha\beta < 1$) or unimodal ($\beta < 1$, $\alpha\beta > 1$)(see, Mudholkar et al. (1995)). Figure 6.1 and Figure 6.2 illustrate the shapes of the EW with different values of α, β, and λ for pdf and the hazard function, respectively. If α is equal to 1, the distribution is said to be Weibull distribution, if β is equal to 1, it is equal to the exponentiated exponential model. If α and β are both 1, the distribution has a constant hazard function, that is exponential. For more results and references, one can see Achcar et al. (2009), Barrios & Dios (2013), Cancho & Bolfarine (2001), Jiang (2010), Mudholkar & Hutson (1996), Nassar & Eissa (2003). Also, comprehensive review about EW and other modified Weibull distributions can be found in Almalki & Nadarajah (2014), Nadarajah et al. (2013).

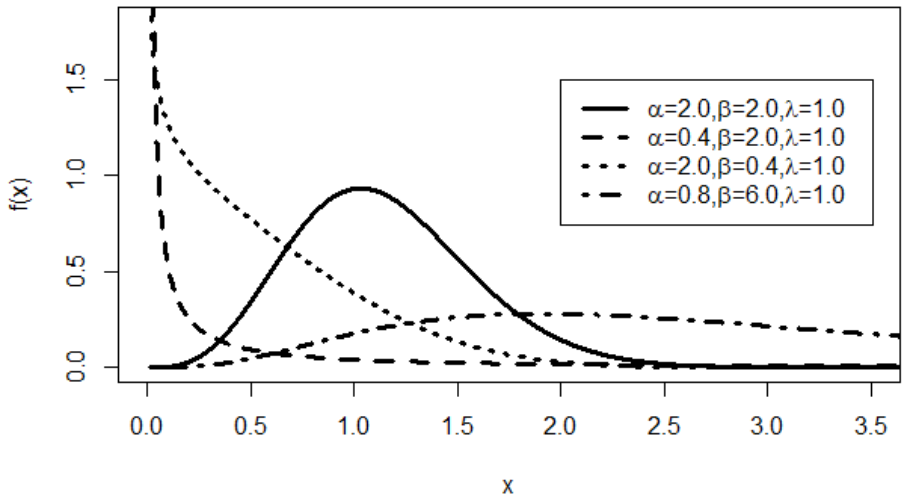


Figure 6.1 Pdf of EW for different values of parameters.

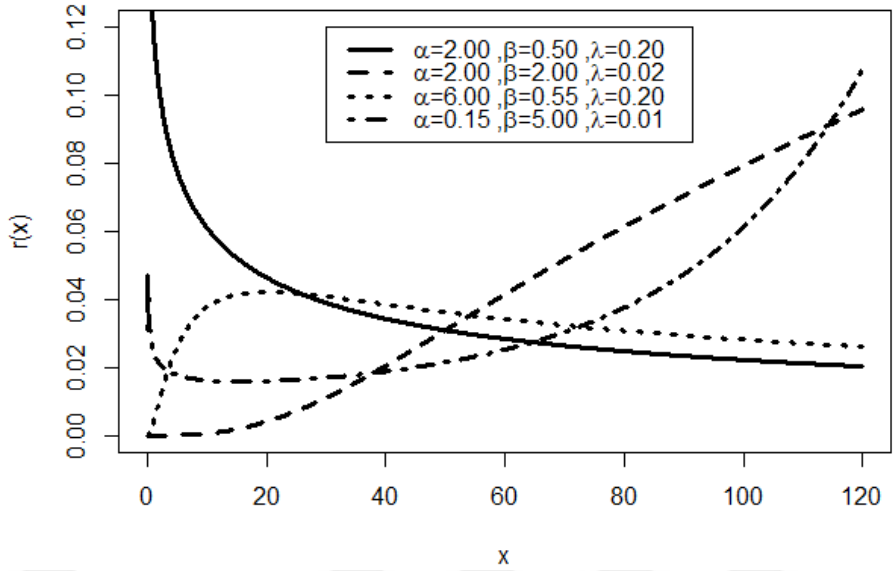


Figure 6.2 Hazard function of EW for different values of parameters.

6.2 Parameter estimation with SRS

Consider that $Y_1, Y_2, ..., Y_m$ is a random sample of size m from $EW(\alpha, \beta, \lambda)$, where α, β and λ are unknown. We aim to apply the procedure of maximum likelihood method using loglikelihood function, $l(\alpha, \beta, \lambda)$, which is given in Mudholkar & Srivastava (1993) as below.

$$\begin{aligned} l(\alpha,\beta,\lambda) &= m\log\alpha + m\log\beta + m\beta\log\lambda + (\beta-1)\sum_{s=1}^{m}\log(y_s) \\ &\quad -\lambda^{\beta}\sum_{s=1}^{m}y_s^{\beta} + (\alpha-1)\sum_{s=1}^{m}\log(1-\exp(-(\lambda y_s)^{\beta})) \end{aligned} \tag{6.4}$$

Therefore, maximum likelihood equations for the parameters of EW distribution become,

$$\frac{\partial l}{\partial \alpha} = \frac{m}{\alpha} + \sum_{s=1}^{m}\log(1-\exp(-(\lambda y_s)^{\beta})) = 0, \tag{6.5}$$

$$\begin{aligned}\frac{\partial l}{\partial \beta} &= \frac{m}{\beta} + m\log\lambda + \sum_{s=1}^{m}\log(y_s) \\ &\quad +(\alpha-1)\lambda^{\beta}\sum_{s=1}^{m}\frac{y_s^{\beta}\log(\lambda y_s)\exp(-(\lambda y_s)^{\beta})}{1-\exp(-(\lambda y_s)^{\beta})} \\ &\quad -\lambda^{\beta}\sum_{s=1}^{m}y_s^{\beta}\log(\lambda y_s) = 0, \end{aligned} \tag{6.6}$$

$$\begin{aligned}\frac{\partial l}{\partial \lambda} &= \frac{m\beta}{\lambda} + (\alpha-1)\beta\lambda^{\beta-1}\sum_{s=1}^{m}\frac{\exp(-(\lambda y_s)^{\beta})y_s^{\beta}}{1-\exp(-(\lambda y_s)^{\beta})} \\ &\quad -\beta\lambda^{\beta-1}\sum_{s=1}^{m}y_s^{\beta} = 0. \end{aligned} \tag{6.7}$$

Since functions given in Eq. (6.5), (6.6), (6.7) have nonlinear forms, it is necessary to solve these by using numerical methods to obtain maximum likelihood estimators of the parameters, namely, $\hat{\alpha}_{srs}$, $\hat{\beta}_{srs}$, $\hat{\lambda}_{srs}$.

6.3 Parameter estimation using RSS

Ranked set sampling (RSS) method considers the ranking information of sample units and it obtaines more representative units than commonly used simple random sampling (SRS) method. It is a useful method of data collection, especially when measuring units is time consuming or expensive. McIntyre (1952) is introduced RSS as a sampling method in order to estimate the mean of yields. Takahasi & Wakimoto (1968) presented the theoretical background of the RSS method and proved that the RSS give smaller variance than the SRS for the mean estimation when the ranking of the units are perfect. There are several papers which deal with estimation of parameters of a model using RSS method and its modificated versions. Lam et al. (1994) obtained the estimator for the parameters of two-parameter exponential

distribution under RSS. Hassan (2013) used maximum likelihood estimation method and Bayesian method for parameter estimation in exponentiated exponential distribution and compared the SRS and RSS methods. Esemen & Gürler (2018) estimated the parameters of generalized Rayleigh distribution by using maximum likelihood method based on the sample obtained from RSS and its modified versions. For some additional study, see Al-Omari & Bouza (2014), Barabesi & El-Sharaawi (2001), Qian et al. (2019), Shaibu & Muttlak (2004), and references therein.

When the our sampling method is RSS, we should consider the following steps for obtaining a sample of size k (Al-Omari & Bouza (2014))

Step 1: Select randomly k sets with k size of sample units from the target population.

Step 2: Rank the units in each set without measuring the variable of interest. Ranking process can be done by using any inexpensive method, such as using an auxiliary variable, visual inspection or expert judgment etc.

Step 3: Choose from the first set the smallest ranked unit, from the second set the second smallest ranked unit, and so forth until from the last set the largest ranked unit.

The three steps above provides a k-sized ranked set sample and this whole procedure is referred as a cycle. The cycle can be repeated c times to obtain a sample of size kc.

Suppose $Y_{s(s:k)t},\ s = 1,2,...,k,\ t = 1,2,...,c$ is a ranked set sample of size $m = kc$ drawn from EW distribution where k and c are the set size and the cycle size, respectively. We represent $Y_{s(s:k)t}$ by X_{st}, which is the sth order statistic, in order to

simplify the following formulations. The pdf of X_{st} is

$$h_s(x_{st}) = \frac{k!}{(s-1)!(k-s)!} g(x_{st})[G(x_{st})]^{s-1}[1-G(x_{st})]^{k-s}, \tag{6.8}$$

where $g(x_{st})$ denotes the pdf and $G(x_{st})$ denotes the cdf. The likelihood function for EW distribution using RSS can be written as

$$\begin{aligned} L^*(\alpha,\beta,\lambda) &= \prod_{t=1}^{c}\prod_{s=1}^{k} h_s(x_{st};\alpha,\beta,\lambda) \\ &= A^{kc}\alpha^{kc}\beta^{kc}\lambda^{kc\beta}\prod_{t=1}^{c}\prod_{s=1}^{k} x_{st}^{\beta-1}(1-\exp(-(\lambda x_{st})^{\beta}))^{\alpha s-1} \\ &\times \exp(-(\lambda x_{st})^{\beta})(1-(1-\exp(-(\lambda x_{st})^{\beta}))^{\alpha})^{k-s}, \end{aligned} \tag{6.9}$$

where $A = \frac{k!}{(k-s)!(s-1)!}$.

Then, the log-likelihood function is

$$\begin{aligned} l^*(\alpha,\beta,\lambda) &= kc\log A + kc\log\alpha + kc\log\beta + kc\beta\log\lambda \\ &+(\beta-1)\sum_{t=1}^{c}\sum_{s=1}^{k}\log(x_{st}) - \lambda^{\beta}\sum_{t=1}^{c}\sum_{s=1}^{k} x_{st}^{\beta} \\ &+\sum_{t=1}^{c}\sum_{s=1}^{k}(\alpha s-1)\log(1-\exp(-(\lambda x_{st})^{\beta})) \\ &+\sum_{t=1}^{c}\sum_{s=1}^{k}(k-s)\log(1-(1-\exp(-(\lambda x_{st})^{\beta}))^{\alpha}). \end{aligned} \tag{6.10}$$

Maximum likelihood equations for each parameter are in the followings:

$$\frac{\partial l^*}{\partial \alpha} = \frac{kc}{\alpha} + \sum_{t=1}^{c}\sum_{s=1}^{k} s\log(1-\exp(-(\lambda x_{st})^{\beta})) - \sum_{t=1}^{c}\sum_{s=1}^{k}(k-s) \times\left[\frac{(1-\exp(-(\lambda x_{st})^{\beta}))^{\alpha}\log(1-\exp(-(\lambda x_{st})^{\beta}))}{1-(1-\exp(-(\lambda x_{st})^{\beta}))^{\alpha}}\right] = 0, \quad (6.11)$$

$$\frac{\partial l^*}{\partial \beta} = \frac{kc}{\beta} + kc\log\lambda + \sum_{t=1}^{c}\sum_{s=1}^{k}\log(x_{st}) - \lambda^{\beta}\log\lambda\sum_{t=1}^{c}\sum_{s=1}^{k}x_{st}^{\beta} + \sum_{t=1}^{c}\sum_{s=1}^{k}(\alpha s-1)\frac{(\lambda x_{st})^{\beta}\exp(-(\lambda x_{st})^{\beta})\log(\lambda x_{st})}{1-\exp(-(\lambda x_{st})^{\beta})} -\lambda^{\beta}\sum_{s=1}^{k}x_{st}^{\beta}\log(x_{st}) - \sum_{t=1}^{c}\sum_{s=1}^{k}(k-s)\alpha\log(\lambda x_{st})(\lambda x_{st})^{\beta} \times\left[\frac{(1-\exp(-(\lambda x_{st})^{\beta}))^{\alpha-1}\exp(-(\lambda x_{st})^{\beta})}{1-(1-\exp(-(\lambda x_{st})^{\beta}))^{\alpha}}\right] = 0, \quad (6.12)$$

$$\frac{\partial l^*}{\partial \lambda} = \frac{kc\beta}{\lambda} - \beta\lambda^{\beta-1}\sum_{t=1}^{c}\sum_{s=1}^{k}x_{st}^{\beta} - \sum_{t=1}^{c}\sum_{s=1}^{k}(k-s)\alpha\beta x_{st}(\lambda x_{st})^{\beta-1} \times\left[\frac{\exp(-(\lambda x_{st})^{\beta})(1-\exp(-(\lambda x_{st})^{\beta}))^{\alpha-1}}{1-(1-\exp(-(\lambda x_{st})^{\beta}))^{\alpha}}\right] + \sum_{t=1}^{c}\sum_{s=1}^{k}(\alpha s-1)\frac{\beta x_{st}(\lambda x_{st})^{\beta-1}\exp(-(\lambda x_{st})^{\beta})}{1-\exp(-(\lambda x_{st})^{\beta})} = 0. \quad (6.13)$$

Since the maximum likelihood equations obtained for estimation of parameters are in nonlinear form, they are solved by numerical methods to obtain maximum likelihood estimates namely, $\hat{\alpha}_{rss}$, $\hat{\beta}_{rss}$, $\hat{\lambda}_{rss}$.

6.4 Simulation study

In this section, we conduct the Monte Carlo simulation study for numerical comparison of the SRS and RSS methods for estimating the unknown parameters (α, β, and λ) of EW distribution by using the maximum likelihood estimation method. In numerical analysis, a quasi-Newton type algorithm is used to approximate solutions of the maximum likelihood equations. The study is designed using R-software with $N = 10,000$ repetitions for different values of sample size (m). In addition, different values for α, β, and λ parameters are considered to obtain different shapes of EW distribution. The numerical results of bias and mean squared error (MSE) are obtained by the following formulations, respectively,

$$bias(\hat{\theta}) = \frac{1}{N}\sum_{s=1}^{N}(\hat{\theta}_s - \theta), \tag{6.14}$$

$$MSE(\hat{\theta}) = \frac{1}{N}\sum_{s=1}^{N}(\hat{\theta}_s - \theta)^2, \tag{6.15}$$

where $\hat{\theta}$ is the estimator of θ parameter and $\hat{\theta}_s$ represents the estimation of parameter in s^{th} repetition for $s = 1, \ldots, N$.

Table 6.1 and Table 6.2 show the estimated MSE and bias values of the parameters, respectively. We numerically observe that MSE values based on RSS are smaller than the values based on SRS for all cases. One can conclude that the MSE values are high for small sample sizes for the parameter λ in the case of $(0.5; 1.5; 2.5)$. In addition, MSE values based on RSS method decrease as the set size increases for all parameters.

Table 6.1 The estimated MSE values for parameters of the EW distiribution based on SRS and RSS.

$(\alpha;\beta;\lambda)$	m	$\hat{\alpha}_{srs}$	$\hat{\beta}_{srs}$	$\hat{\lambda}_{srs}$	$(k;c)$	$\hat{\alpha}_{rss}$	$\hat{\beta}_{rss}$	$\hat{\lambda}_{rss}$
(2; 0.8; 2)	12	1.8605	0.6490	2.2361	(3; 4)	1.5435	0.5288	1.6833
					(4; 3)	1.5065	0.5320	1.6814
					(6; 2)	1.3241	0.4842	1.3158
	24	1.7822	0.3867	2.2826	(3; 8)	1.4777	0.3313	1.7156
					(4; 6)	1.3970	0.3276	1.5903
					(6; 4)	1.1829	0.3042	1.2787
	48	1.5893	0.1847	2.2706	(3; 16)	1.3428	0.1505	1.7203
					(4; 12)	1.2170	0.1445	1.5080
					(6; 8)	1.0094	0.1279	1.2038
	96	1.2852	0.0667	1.9634	(3; 32)	1.0330	0.0565	1.4225
					(4; 24)	0.9892	0.0513	1.3432
					(6; 16)	0.7666	0.0432	0.9909
(2; 1.5; 2)	12	3.8382	1.3380	1.0787	(3; 4)	3.0376	0.7630	0.8230
					(4; 3)	2.6299	0.8299	0.7334
					(6; 2)	1.8949	1.1963	0.5511
	24	3.4557	0.8576	1.0135	(3; 8)	2.7100	0.5032	0.7645
					(4; 6)	2.3642	0.5343	0.6478
					(6; 4)	1.8448	0.7671	0.5465
	48	3.0878	0.5005	0.8976	(3; 16)	2.2815	0.3374	0.6500
					(4; 12)	2.0997	0.3257	0.6062
					(6; 8)	1.5208	0.3840	0.4337
	96	2.2837	0.2594	0.6784	(3; 32)	1.7379	0.1885	0.5000
					(4; 24)	1.5303	0.1727	0.4401
					(6; 16)	1.1161	0.1688	0.3259
(0.5; 1.5; 2.5)	12	0.7459	0.6505	10.1066	(3; 4)	0.5544	0.6325	7.6216
					(4; 3)	0.4868	0.5849	6.6253
					(6; 2)	0.4152	0.5397	5.3953
	24	0.4620	0.4318	7.6939	(3; 8)	0.3330	0.4302	4.7245
					(4; 6)	0.3081	0.4287	4.3945
					(6; 4)	0.2475	0.4056	3.4146
	48	0.2184	0.3186	3.6440	(3; 16)	0.1668	0.3014	2.4107
					(4; 12)	0.1504	0.2907	2.1035
					(6; 8)	0.1156	0.2819	1.4939
	96	0.0833	0.1859	1.2315	(3; 32)	0.0636	0.1767	0.8009
					(4; 24)	0.0531	0.1628	0.6296
					(6; 16)	0.0473	0.1530	0.5415
(0.8; 0.8; 1)	12	0.8692	0.2417	4.5121	(3; 4)	0.5877	0.1925	2.8218
					(4; 3)	0.5827	0.1836	2.4713
					(6; 2)	0.5088	0.1535	1.9837
	24	0.6621	0.1494	3.6248	(3; 8)	0.4840	0.1300	2.3875
					(4; 6)	0.4226	0.1291	1.8962
					(6; 4)	0.3530	0.1143	1.4309
	48	0.4276	0.0940	2.4840	(3; 16)	0.3334	0.0915	1.5563
					(4; 12)	0.2988	0.0878	1.2995
					(6; 8)	0.2497	0.0795	0.9663
	96	0.2360	0.0578	1.2053	(3; 32)	0.1917	0.0541	0.7589
					(4; 24)	0.1651	0.0530	0.5904
					(6; 16)	0.1457	0.0474	0.4772

Table 6.2 The estimated bias values for parameters of the EW distribution based on SRS and RSS.

$(\alpha;\beta;\lambda)$	m	$\hat{\alpha}_{srs}$	$\hat{\beta}_{srs}$	$\hat{\lambda}_{srs}$	$(k;c)$	$\hat{\alpha}_{rss}$	$\hat{\beta}_{rss}$	$\hat{\lambda}_{rss}$
(2; 0.8; 2)	12	-0.1975	0.4590	-0.0689	(3; 4)	-0.2618	0.4150	-0.1381
					(4; 3)	-0.2911	0.4176	-0.1551
					(6; 2)	-0.3306	0.3957	-0.2241
	24	-0.1009	0.3245	0.0501	(3; 8)	-0.1602	0.2984	-0.0332
					(4; 6)	-0.1816	0.2968	-0.0577
					(6; 4)	-0.2554	0.2882	-0.1412
	48	0.0668	0.1781	0.2265	(3; 16)	0.0388	0.1563	0.1627
					(4; 12)	0.0294	0.1480	0.1438
					(6; 8)	-0.0466	0.1471	0.0528
	96	0.2158	0.0656	0.3578	(3; 32)	0.1911	0.0532	0.3021
					(4; 24)	0.1837	0.0496	0.2870
					(6; 16)	0.1045	0.0525	0.1866
(2; 1.5; 2)	12	0.3510	0.5520	0.1531	(3; 4)	0.3691	0.3625	0.1494
					(4; 3)	0.2898	0.3786	0.1231
					(6; 2)	0.0494	0.5564	0.0491
	24	0.3426	0.3962	0.1575	(3; 8)	0.3348	0.2719	0.1475
					(4; 6)	0.2662	0.2793	0.1143
					(6; 4)	0.0427	0.3958	0.0063
	48	0.3814	0.2450	0.1866	(3; 16)	0.3193	0.1834	0.1509
					(4; 12)	0.3121	0.1655	0.1555
					(6; 8)	0.1296	0.2241	0.0537
	96	0.3781	0.1191	0.1963	(3; 32)	0.3215	0.0905	0.1608
					(4; 24)	0.2816	0.0865	0.1417
					(6; 16)	0.1771	0.0981	0.0877
(0.5; 1.5; 2.5)	12	-0.4064	-0.0916	-1.6464	(3; 4)	-0.3356	-0.1072	-1.3677
					(4; 3)	-0.3048	-0.0977	-1.2323
					(6; 2)	-0.2746	-0.0891	-1.0904
	24	-0.3028	-0.0221	-1.2932	(3; 8)	-0.2393	-0.0523	-0.9724
					(4; 6)	-0.2217	-0.0681	-0.9101
					(6; 4)	-0.1898	-0.0673	-0.7670
	48	-0.1757	-0.0363	-0.7556	(3; 16)	-0.1343	-0.0610	-0.5627
					(4; 12)	-0.1271	-0.0594	-0.5230
					(6; 8)	-0.0985	-0.0778	-0.4074
	96	-0.0850	-0.0358	-0.3595	(3; 32)	-0.0632	-0.0501	-0.2622
					(4; 24)	-0.0529	-0.0520	-0.2157
					(6; 16)	-0.0452	-0.0566	-0.1879
(0.8; 0.8; 1)	12	0.3502	0.1609	0.8925	(3; 4)	0.2316	0.1469	0.6418
					(4; 3)	0.2542	0.1273	0.6379
					(6; 2)	0.2339	0.1074	0.5723
	24	0.2993	0.0874	0.7921	(3; 8)	0.2293	0.0847	0.5891
					(4; 6)	0.2046	0.0853	0.5291
					(6; 4)	0.1698	0.0824	0.4373
	48	0.2224	0.0486	0.6076	(3; 16)	0.1813	0.0522	0.4632
					(4; 12)	0.1565	0.0569	0.4054
					(6; 8)	0.1296	0.0570	0.3325
	96	0.1357	0.0325	0.3622	(3; 32)	0.1093	0.0348	0.2744
					(4; 24)	0.0988	0.0348	0.2439
					(6; 16)	0.0892	0.0311	0.2153

6.5 Vehicle headway data modeling using EW distribution

Vehicle headway modeling is a very important concept for intelligent transportation in traffic engineering. It is useful for the traffic signal optimization and flow modelling by characterizing the distribution of vehicles on a road. Vehicle headway is defined as the time interval between the successive arrival of vehicles in a lane and it is a fundamental topic for many traffic flow optimization study and vehicle simulation issues. Statistical modeling of vehicle headway data has received great attention along the last two decades, and there are a variety of distributions that can be used for different traffic conditions. Some of the statistical distributions found suitable for modelling vehicle headway are inverse Weibull (Riccardo & Massimiliano (2012)), shifted lognormal (Abtahi et al. (2011)), gamma (Al-Ghamdi (2001), Abtahi et al. (2011)), negative exponential (Al-Ghamdi (2001)), lognormal (Greenberg (1966), Yin et al. (2009), Dey & Chandra (2009)), log-logistic (Yin et al. (2009), Jang (2012)), and generalized extreme value (Panichpapiboon (2014)). Recently, Li & Chen (2017) presented a comprehensive review on the vehicle headway modeling and provide the genealogical tree of headway models used in past studies.

Badhrudeen et al. (2016) proposed to use the Weibull distribution as the best model fitted for vehicle headway data in India. In their study, the headway observations are collected from an automated sensor located on a urban road consisting of six lanes and analysed seperately based on the leader-follower pairs and also for the specific time of a day. They found that the best model for the headway data between the hours 07:30 am and 10:45 am was the Weibull distribution and they estimated the shape and the scale parameters as 1.9718 and 2.6372, respectively. In this section, we use a simulated data set consisting of 1000 observations which is generated from the established Weibull model in the study of Badhrudeen et al. (2016) for the vehicle headway between the hours 07:30 and 10:45. We analyse the simulated headway data set using the EW distribution. We used a quasi-Newton type algorithm in R-software to maximize the likelihood function and obtain that the maximum likelihood

estimates as $\hat{\alpha} = 1.3475$, $\hat{\beta} = 1.7686$, and $\hat{\lambda} = 0.4233$. We found that the EW distribution provides an acceptable fit for the simulated headway data in terms of Anderson-Darling goodness of fit test (p-value$= 0.8293$).

In order to estimate the parameters of EW distribution by using SRS and RSS methods, we analyse two sets of sample data of size $m = 40$. For the RSS method, the sample is drawn with replacement and the set and cycle sizes are determined as $k = 5$ and $c = 8$. The maximum likelihood estimates of the parameters of EW distribution based on SRS and RSS are given in Table 6.3. Figure 6.3 shows the histogram of the simulated data and the shape of the model using the maximum likelihood estimates of parameters given in Table 6.3 for each sampling method.

Table 6.3 Maximum likelihood estimates for the parameters of EW distribution using SRS and RSS.

	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\lambda}$
SRS	0.6135	2.3532	0.3103
RSS	1.7320	1.4549	0.5000

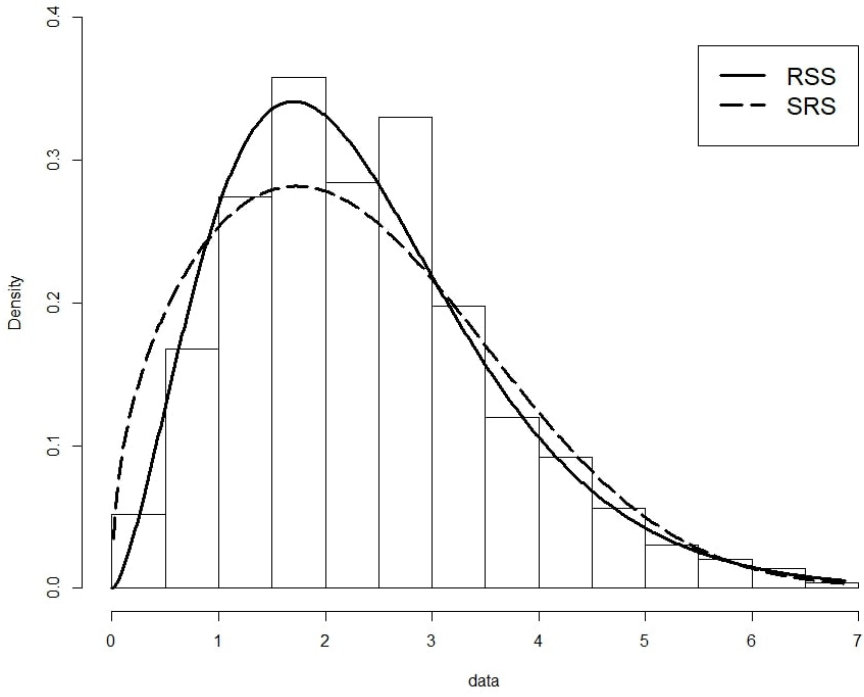


Figure 6.3 Histogram of the simulated data and the pdfs of the fitted EW distribution using SRS and RSS.

CHAPTER 7
CONCLUSION

In this thesis, we propose the SAC method as a new clustering approach that has the advantage of being highly fast. The main idea under the algorithm is based on a sampling method which is called Adaptive Cluster Sampling (ACS). The nature of the ACS design has the ability to detect clustered units in an easy way. Therefore, the ACS method is adapted to a new form to be used as an efficient clustering method. The SAC algorithm consists of four main parts; constructing the grid structure, calculating the density of each grid, forming the clusters using neighborhood search and marking the noise points. It is easy to understand and implement. Also, it does not require a similarity or a distance measure. Additionally, the performance of discovering clusters is efficient for convex-shaped data and arbitrary shaped data with noise. The algorithm compared with popular density-based clustering methods, namely, DBSCAN, OPTICS and HDBSCAN in two different aspects. First, the algorithms compared in terms of external validation measures in artificial data sets and real data sets from different fields with various sample sizes and dimensions. The results of the study show that the SAC algorithm gives better results than the DBSCAN, OPTICS and HDBSCAN algorithms in most artificial data sets and in almost all real data sets. Second, the algorithms compared in terms of their runtimes in artificial data sets for different values of sample sizes. The SAC algorithm outperforms DBSCAN and OPTICS in terms of runtimes since the runtime of the SAC is not affected by increasing the number of data points. In addition, the algorithm was applied to accident data sets from Birmingham and Buca to detect the hotspot locations. After detecting the accident hotspot locations, they were examined to investigate the factors behind the accidents. In this way, the number of accidents may be decreased by taking some actions to change the factors that may cause the accident. Besides the superiority of the method in terms of other indices, the SAC method can be recommended, especially in cases where the number of data points is large. Moreover, the noise points do not affect the performance of the SAC method since it has the ability to detect noise points. As a result, the SAC algorithm can be an alternative approach to clustering methods. Additionally, we dealt with the

vehicle headway data modeling. Our study shows that EW distribution may be used as an alternative model for the vehicle time headway data. We used maximum likelihood equations for parameters of EW distribution based on RSS when all parameters are unknown. The Monte Carlo simulation study is conducted in R-software in order to compare the performances of the RSS and SRS methods in the parameter estimation of EW distribution. The results support that RSS gives more efficient results than the SRS method based on their MSE values for all cases of sample and set sizes. Also, a numerical study based on a real data application for vehicle headway is presented to show the flexibility of EW distribution. The results indicate that the estimates of the parameters based on RSS are closer to the fitted values than the estimates based on SRS.

REFERENCES

Abtahi, S. M., Tamannaei, M., & Haghshenash, H. (2011). Analysis and modeling time headway distributions under heavy traffic flow conditions in the urban highways: Case of Isfahan. *Transport*, *26*(4), 375–382.

Achcar, J. A., Ortiz-Rodríguez, G., & Rodrigues, E. R. (2009). Estimating the number of ozone peaks in Mexico City using a non-homogeneous Poisson model and a Metropolis–Hastings algorithm. *International Journal of Pure and Applied Mathematics*, *53*(1), 1–20.

Al-Ghamdi, A. S. (2001). Analysis of time headways on urban roads: Case study from Riyadh. *Journal of Transportation Engineering*, *127*(4), 289–294.

Al-Omari, A. I., & Bouza, C. N. (2014). Review of ranked set sampling: Modifications and applications. *Investigación Operacional*, *35*(3), 215–241.

Almalki, S. J., & Nadarajah, S. (2014). Modifications of the Weibull distribution: A review. *Reliability Engineering & System Safety*, *124*, 32–55.

Alotaibi, A. S. (2018). Density-based clustering for road accident data analysis. *International Journal of Advanced and Applied Sciences*, *5*(8), 113–121.

Anderson, T. K. (2009). Kernel density estimation and k-means clustering to profile road accident hotspots. *Accident Analysis & Prevention*, *41*(3), 359–364.

Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record*, *28*(2), 49–60.

Ayech, M. W., & Ziou, D. (2015). Segmentation of Terahertz imaging using k-means clustering based on ranked set sampling. *Expert Systems with Applications*, *42*(6), 2959–2974.

Badhrudeen, M., Ramesh, V., & Vanajakshi, L. (2016). Headway analysis using automated sensor data under Indian traffic conditions. *Transportation Research Procedia*, *17*, 331–339.

Barabesi, L., & El-Sharaawi, A. (2001). The efficiency of ranked set sampling for parameter estimation. *Statistics & probability letters*, *53*(2), 189–199.

Barrios, R., & Dios, F. (2013). Exponentiated Weibull model for the irradiance probability density function of a laser beam propagating through atmospheric turbulence. *Optics & Laser Technology*, *45*, 13–20.

Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, *60*(1), 208–221.

Boonchoo, T., Ao, X., Liu, Y., Zhao, W., Zhuang, F., & He, Q. (2019). Grid-based DBSCAN: Indexing and inference. *Pattern Recognition*, *90*, 271–284.

Borah, B., & Bhattacharyya, D. K. (2004). An improved sampling-based DBSCAN for large spatial databases. In *International Conference on Intelligent Sensing and Information Processing, 2004*, (92–96). IEEE.

Bryant, A., & Cios, K. (2017). RNN-DBSCAN: A density-based clustering algorithm using reverse nearest neighbor density estimates. *IEEE Transactions on Knowledge and Data Engineering*, *30*(6), 1109–1121.

Caliskanelli, S. P., & Tanyel, S. (2010). Investigation of vehicle bunching at signalized arterials in turkey. *Canadian Journal of Civil Engineering*, *37*(3), 380–388.

Campello, R. J., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, (160–172). Springer.

Cancho, V. G., & Bolfarine, H. (2001). Modeling the presence of immunes by using the exponentiated-Weibull model. *Journal of Applied Statistics*, *28*(6), 659–671.

Chandra, G., Tiwari, N., & Chandra, H. (2011). Adaptive cluster sampling based on ranked sets. *Metodoloski zvezki*, *8*(1), 39–55.

Chen, Y., Tang, S., Bouguila, N., Wang, C., Du, J., & Li, H. (2018). A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data. *Pattern Recognition*, *83*, 375–387.

Davis, J. G., Cook, S. B., & Smith, D. D. (2011). Testing the utility of an adaptive cluster sampling method for monitoring a rare and imperiled darter. *North American Journal of Fisheries Management*, *31*(6), 1123–1132.

Department for Transport (2019). Road safety data. Available at https://data.gov.uk/dataset/ road-accidents-safety-data.

Dey, P. P., & Chandra, S. (2009). Desired time gap and time headway in steady-state car-following on two-lane roads. *Journal of transportation engineering*, *135*(10), 687–693.

Dua, D., & Graff, C. (2019). UCI machine learning repository. Irvine, CA: University of California, school of information and computer science.

Elshahhat, A. (2017). Parameters estimation for the exponentiated Weibull distribution based on generalized progressive hybrid censoring schemes. *American Journal of Applied Mathematics and Statistics*, *5*(2), 33–48.

Esemen, M., & Gürler, S. (2018). Parameter estimation of generalized Rayleigh distribution based on ranked set sample. *Journal of Statistical Computation and Simulation*, *88*(4), 615–628.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *2nd Int. Conf. Knowledge Discovery and Data Mining (KDD'96)*, volume 96, (226–231).

Gan, G., Ma, C., & Wu, J. (2020). *Data clustering: theory, algorithms, and applications*. SIAM.

Gan, J., & Tao, Y. (2015). DBSCAN revisited: Mis-claim, un-fixability, and approximation. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, (519–530).

Ganjali Khosrowshahi, A., Aghayan, I., Kunt, M. M., & Choupani, A.-A. (2021). Detecting crash hotspots using grid and density-based spatial clustering. In

Proceedings of the Institution of Civil Engineers-Transport, (1–13). Thomas Telford Ltd.

Ghnimi, S., & Gasmi, S. (2014). Parameter estimations for some modifications of the Weibull distribution. *Open Journal of Statistics*, *4*(8), 597–610.

Greenberg, I. (1966). The log normal distribution of headways. *Australian Road Research*, *2*(7), 14–18.

Hahsler, M., Piekenbrock, M., & Doran, D. (2019). dbscan: fast density based clustering with R. *Journal of Statistical Software*, *91*(1), 1-30.

Hassan, A. S. (2013). Maximum likelihood and Bayes estimators of the unknown parameters for exponentiated exponential distribution using ranked set sampling. *International Journal of Engineering Research and Applications*, *3*, 720–725.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of classification*, *2*(1), 193–218.

Iliopoulou, C. A., Milioti, C. P., Vlahogianni, E. I., & Kepaptsoglou, K. L. (2020). Identifying spatio-temporal patterns of bus bunching in urban networks. *Journal of Intelligent Transportation Systems*, *24*(4), 365–382.

Jang, J. (2012). Analysis of time headway distribution on suburban arterial. *KSCE Journal of Civil Engineering*, *16*(4), 644–649.

Jang, J., Park, C., Kim, B., & Choi, N. (2011). Modeling of time headway distribution on suburban arterial: Case study from south korea. *Procedia-social and behavioral sciences*, *16*, 240–247.

Jiang, R. (2010). Discrete competing risk model with application to modeling bus-motor failure data. *Reliability Engineering & System Safety*, *95*(9), 981–988.

Kahle, D. J., & Wickham, H. (2013). ggmap: spatial visualization with ggplot2. *R J.*, *5*(1), 144.

Khan, M. M. R., Siddique, M. A. B., Arif, R. B., & Oishe, M. R. (2018). ADBSCAN: Adaptive density-based spatial clustering of applications with noise for identifying

clusters with varying densities. In *4th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, (107–111). IEEE.

Kim, K., & Yamashita, E. Y. (2007). Using ak-means clustering algorithm to examine patterns of pedestrian involved crashes in honolulu, hawaii. *Journal of advanced transportation*, *41*(1), 69–89.

Kong, D., & Guo, X. (2016). Analysis of vehicle headway distribution on multi-lane freeway considering car–truck interaction. *Advances in Mechanical Engineering*, *8*(4), 1687814016646673.

Kumar, K. M., & Reddy, A. R. M. (2016). A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method. *Pattern Recognition*, *58*, 39–48.

Kumar, S., & Toshniwal, D. (2015). A data mining framework to analyze road accident data. *Journal of Big Data*, *2*(1), 1–18.

Kumar, S., & Toshniwal, D. (2016). A data mining approach to characterize road accident locations. *Journal of Modern Transportation*, *24*(1), 62–72.

Lam, K., Sinha, B. K., & Wu, Z. (1994). Estimation of parameters in a two-parameter exponential distribution using ranked set sample. *Annals of the Institute of Statistical Mathematics*, *46*(4), 723–736.

Le, K. G., Liu, P., & Lin, L.-T. (2020). Determining the road traffic accident hotspots using gis-based temporal-spatial statistical analytic techniques in hanoi, vietnam. *Geo-spatial Information Science*, *23*(2), 153–164.

Li, L., & Chen, X. M. (2017). Vehicle headway modeling and its inferences in macroscopic/microscopic traffic flow theory: A survey. *Transportation Research Part C: Emerging Technologies*, *76*, 170–188.

Liao, S. H., Chu, P. H., & Hsiao, P. Y. (2012). Data mining techniques and applications–A decade review from 2000 to 2011. *Expert Systems with Applications*, *39*(12), 11303–11311.

Luttinen, R. (1992). Statistical properties of vehicle time headways. *Transportation research record*, 92–92.

Lyon, R. J., Stappers, B., Cooper, S., Brooke, J. M., & Knowles, J. D. (2016). Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach. *Monthly Notices of the Royal Astronomical Society*, *459*(1), 1104–1123.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, (281–297). Oakland, CA, USA.

Maltaş, A., Özen, H., Saraçoğlu, A. et al. (2019). Determination of accident black spots by using network screening and k-means clustering methods: a case study on d100 highway in istanbul. *Pamukkale University Journal of Engineering Sciences*, *25*(6), 672–682.

McIntyre, G. (1952). A method for unbiased selective sampling, using ranked sets. *Australian Journal of Agricultural Research*, *3*(4), 385–390.

Mudholkar, G. S., & Hutson, A. D. (1996). The exponentiated Weibull family: Some properties and a flood data application. *Communications in Statistics–Theory and Methods*, *25*(12), 3059–3083.

Mudholkar, G. S., & Srivastava, D. K. (1993). Exponentiated Weibull family for analyzing bathtub failure-rate data. *IEEE Transactions on Reliability*, *42*(2), 299–302.

Mudholkar, G. S., Srivastava, D. K., & Freimer, M. (1995). The exponentiated Weibull family: A reanalysis of the bus-motor-failure data. *Technometrics*, *37*(4), 436–445.

Nadarajah, S., Cordeiro, G. M., & Ortega, E. M. (2013). The exponentiated Weibull distribution: A survey. *Statistical Papers*, *54*(3), 839–877.

Nassar, M. M., & Eissa, F. H. (2003). On the exponentiated Weibull distribution. *Communications in Statistics-Theory and Methods*, *32*(7), 1317–1336.

Nolau, I., Gonçalves, K., & Pereira, J. (2021). Model-based inference for rare and clustered populations from adaptive cluster sampling using auxiliary variables. *Journal of Survey Statistics and Methodology*.

Pal, M., Ali, M. M., & Woo, J. (2006). Exponentiated Weibull distribution. *Statistica*, *66*(2), 139–147.

Panichpapiboon, S. (2014). Time-headway distributions on an expressway: Case of Bangkok. *Journal of Transportation Engineering*, *141*(1), 1–8.

Park, H.-S., & Jun, C.-H. (2009). A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, *36*(2), 3336–3341.

Patummasut, M., & Borkowski, J. J. (2014). Adaptive cluster sampling with spatially clustered secondary units. *Journal of Applied Sciences*, *14*(20), 2516–2522.

Qian, W., Chen, W., & He, X. (2019). Parameter estimation for the Pareto distribution based on ranked set sampling. *Statistical Papers*, 1–23.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, *66*(336), 846–850.

Riccardo, R., & Massimiliano, G. (2012). An empirical analysis of vehicle time headways on rural two-lane two-way roads. *Procedia-Social and Behavioral Sciences*, *54*, 865–874.

Roesch, F. A. (1993). Adaptive cluster sampling for forest inventories. *Forest Science*, *39*(4), 655–669.

Rossi, R., Mantuano, A., Pascucci, F., & Rupi, F. (2017). Fitting time headway and speed distributions for bicycles on separate bicycle lanes. *Transportation research procedia*, *27*, 19–26.

Rousseeuw, P. J., & Kaufman, L. (1990). Finding groups in data: An introduction to cluster analysis. *Hoboken: Wiley Online Library*.

Roy, R., & Saha, P. (2018). Headway distribution models of two-lane roads under mixed traffic conditions: a case study from india. *European transport research review*, *10*(1), 1–12.

Ruspini, E. H., Bezdek, J. C., & Keller, J. M. (2019). Fuzzy clustering: A historical perspective. *IEEE Computational Intelligence Magazine*, *14*(1), 45–55.

Sakai, T., Tamura, K., & Kitakami, H. (2017). Cell-based DBSCAN algorithm using minimum bounding rectangle criteria. In *International Conference on Database Systems for Advanced Applications*, (133–144). Springer.

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A. et al. (2017). A review of clustering techniques and developments. *Neurocomputing*, *267*, 664–681.

Schikuta, E. (1996). Grid-clustering: An efficient hierarchical clustering method for very large data sets. In *Proceedings of 13th international conference on pattern recognition*, volume 2, (101–105). IEEE.

Schikuta, E., & Erhart, M. (1997). The bang-clustering system: Grid-based data analysis. In *International Symposium on Intelligent Data Analysis*, (513–524). Springer.

Shaibu, A.-B., & Muttlak, H. A. (2004). Estimating the parameters of the normal, exponential and gamma distributions using median and extreme ranked set samples. *Statistica*, *64*(1), 75–98.

Shen, L., Lu, J., Long, M., & Chen, T. (2019). Identification of accident blackspots on rural roads using grid clustering and principal component clustering. *Mathematical Problems in Engineering*, *2019*.

Shinde, P., Kudalkar, G., Pawar, S., Tiwari, H., & Khairnar, H. (2022). Accident hotspot detection by db-scan clustering. In *ICT Analysis and Applications* (413–418). Springer.

Shittu, O. I., & Adepoju, K. (2014). On the exponentiated Weibull distribution for modeling wind speed in south western Nigeria. *Journal of Modern Applied Statistical Methods*, *13*(1), 431–445.

Smith, D. R., Villella, R. F., & Lemarié, D. P. (2003). Application of adaptive cluster sampling to low-density populations of freshwater mussels. *Environmental and Ecological Statistics*, *10*(1), 7–15.

Steenberghen, T., Dufays, T., Thomas, I., & Flahaut, B. (2004). Intra-urban location and clustering of road accidents using gis: a belgian example. *International Journal of Geographical Information Science*, *18*(2), 169–181.

Takahasi, K., & Wakimoto, K. (1968). On unbiased estimates of the population mean based on the sample stratified by means of ordering. *Annals of the Institute of Statistical Mathematics*, *20*(1), 1–31.

Thompson, S., & Seber, G. (1995). Adaptive sampling. *The survey statistician*, *32*, 13–15.

Thompson, S. K. (1990). Adaptive cluster sampling. *Journal of the American Statistical Association*, *85*(412), 1050–1059.

Touhbi, S., Babram, M. A., Nguyen-Huu, T., Marilleau, N., Hbid, M. L., Cambier, C., & Stinckwich, S. (2018). Time headway analysis on urban roads of the city of marrakesh. *Procedia computer science*, *130*, 111–118.

Türkiye İstatistik Kurumu - TÜİK (22.12.2021). Karayolu trafik kaza istatistikleri, 2020, https://data.tuik.gov.tr/Bulten/Index?p=Karayolu-Trafik-Kaza-Istatistikleri-2020-37436.

Turk, P., & Borkowski, J. J. (2005). A review of adaptive cluster sampling: 1990–2003. *Environmental and Ecological Statistics*, *12*(1), 55–94.

Ulutagay, G. (2009). *Construction and analysis of clustering algorithms based on fuzzy relations and their applications to EEG data*. PhD thesis, DEÜ Fen Bilimleri Enstitüsü.

Ulutagay, G., & Nasibov, E. (2012). Fuzzy and crisp clustering methods based on the neighborhood concept: A comprehensive review. *Journal of Intelligent & Fuzzy Systems*, *23*(6), 271–281.

Uncu, O., Gruver, W. A., Kotak, D. B., Sabaz, D., Alibhai, Z., & Ng, C. (2006). Gridbscan: Grid density-based spatial clustering of applications with noise. In *2006 IEEE International Conference on Systems, Man and Cybernetics*, volume 4, (2976–2981). IEEE.

Wang, L., Li, M., Han, X., & Zheng, K. (2018). An improved density-based spatial clustering of application with noise. *International Journal of Computers and Applications*, *40*(3), 1–7.

Wang, W., Yang, J., Muntz, R. et al. (1997). Sting: A statistical information grid approach to spatial data mining. In *Vldb*, volume 97, (186–195). Citeseer.

Wickham, H. (2011). ggplot2. *Wiley interdisciplinary reviews: computational statistics*, *3*(2), 180–185.

Yanchang, Z., & Junde, S. (2001). GDILC: a grid-based density-isoline clustering algorithm. In *2001 International Conferences on Info-Tech and Info-Net. Proceedings (Cat. No. 01EX479)*, volume 3, (140–145). IEEE.

Yin, S., Li, Z., Zhang, Y., Yao, D., Su, Y., & Li, L. (2009). Headway distribution modeling with regard to traffic status. In *2009 IEEE Intelligent Vehicles Symposium*, (1057–1062). IEEE.

Zadegan, S. M. R., Mirzaie, M., & Sadoughi, F. (2013). Ranked k-medoids: A fast and accurate rank-based partitioning algorithm for clustering large datasets. *Knowledge-Based Systems*, *39*, 133–143.

Zhao, X., Liang, J., & Dang, C. (2019). A stratified sampling based clustering algorithm for large-scale data. *Knowledge-Based Systems*, *163*, 416–428.

Zhu, G., & Fu, L. (2018). k-step adaptive cluster sampling with horvitz–thompson estimator. *International Journal of Biomathematics*, *11*(02), 1850029.