

**HAREKETLİ NESNELER ÜZERİNDE  
DURMALARIN VE HAREKETLERİN YÖRÜNGELERİ  
İÇİN ALGORİTMALAR GELİŞTİRMEK**

**Mete YAĞANOĞLU**

**Yüksek Lisans Tezi  
Bilgisayar Mühendisliği Anabilim Dalı  
Doç. Dr. Abdulsamet HAŞILOĞLU  
2012  
Her hakkı saklıdır**

**ATATÜRK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

**HAREKETLİ NESNELER ÜZERİNDE  
DURMALARIN VE HAREKETLERİN YÖRÜNGELERİ İÇİN  
ALGORİTMALAR GELİŞTİRMEK**

**Mete YAĞANOĞLU**

**BİLGİSAYAR MÜHENDİLİĞİ ANABİLİM DALI**

**ERZURUM  
2012**

**Her hakkı saklıdır**



T.C.  
ATATÜRK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ



TEZ ONAY FORMU

Hareketli Nesnelere Üzerinde Durmaların ve Hareketlerin Yörüngeleri için Algoritmalar Geliştirmek

Doç. Dr. Abdulsamet HAŞILOĞLU danışmanlığında, Mete YAĞANOĞLU tarafından hazırlanan bu çalışma 25/12/2012 tarihinde aşağıdaki jüri tarafından **Bilgisayar Mühendisliği Anabilim Dalı**'nda Yüksek Lisans tezi olarak **oybirliği** ile kabul edilmiştir.

Başkan : Doç. Dr. Abdulsamet HAŞILOĞLU

İmza

Üye : Doç. Dr. Köksal ERENTÜRK

İmza

Üye : Doç. Dr. Tevhit KARACALI

İmza

Yukarıdaki sonucu onaylıyorum

Prof. Dr. İhsan EFEOĞLU  
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

## ÖZET

Yüksek Lisans Tezi

### HAREKETLİ NESNELER ÜZERİNDE DURMALARIN ve HAREKETLERİN YÖRÜNGELERİ İÇİN ALGORİTMALAR GELİŞTİRMEK

Mete YAĞANOĞLU

Atatürk Üniversitesi  
Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Abdulsamet HAŞILOĞLU

Hastaların zamana ve mekâna göre hasta izleme tarihçesi sonucunda terabayt hatta petabayt boyutunda veri üretimi oluşmaktadır. Bu boyuttaki mekânsal-zamansal verilerden durmaların ve hareketlerin olduğu yörüngeler bulunabilir. Hasta takibi ile elde edilen bu yörüngelerden anlamsal yörüngeler elde edilir. Sağlık hizmetlerinden elde ettiğimiz hareketli nesne verilerine algoritmanın uygulanmasından sonra, kullanıcılara elde ettiğimiz sonuçları grafiksel ara yüzler, raporlamalar ile sunulabilecek ve anlamsal yörüngeler yorumlanabilecektir. Bu görsel arayüzler sayesinde geçmiş ve günümüz verisinden gelecek ile ilgili çıkarımlar yapılabilecektir. Bu çalışmanın amacı, büyük miktardaki hareketli nesne verisinin etkili yönetim performansını artıracak kabul edilebilir en iyi algoritma tekniğini ve sonuçlarını bulmaya çalışmaktır. Geliştirdiğimiz algoritmalarda yörünge kümelemesi adımlarında bulanık kümeleme yöntemleri kullanarak uygulamalarda fark yaratacak teknikler geliştirilmiştir. Oluşturduğumuz grafiksel ara yüzler sayesinde her türlü hareketli nesne verisi için sonuçlar göstermek amaçlanmıştır. Yön tabanlı kümeleme algoritması olan DB-SMoT algoritmasının kümeleme aşamasında bulanık kümeleme yöntemleri kullanılmıştır. Geliştirilen algoritmalar arasında en iyi performansa bulanık k ortalamalar tabanlı DB-SMoT elde etmiştir. En hızlı çalışan algoritma ise bulanık c ortalamalar tabanlı DB-SMoT algoritmasıdır.

**2012, 102 sayfa**

**Anahtar Kelimeler:** Durmaların ve hareketlerin yörüngesi, mekânsal-zamansal veri madenciliği, anlamsal yörüngeler, yörünge kümeleme, bulanık kümeleme

## ABSTRACT

MS Thesis

### DEVELOPING STOPS AND MOVES OF TRAJECTORY ALGORITHMS ON MOVING OBJECTS

Mete YAĞANOĞLU

Atatürk University  
Graduate School of Natural and Applied Sciences  
Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Abdulsamet HAŞILOĞLU

As a result of patient tracking history according to time and space, data production occurs size in terabyte or even petabyte. Of this size, spatio-temporal data can be found stops and moves of trajectories. Semantic trajectories are obtained by moving data which is taken with patient tracking. After implementation of algorithm from the moving object data which is obtained from healthcare services, we are able to present results to end-users by graphical interfaces, reports, and comment semantic trajectories. With these visual interfaces, we are able to do inferences about future from past and today's data. In this study, we try to find best algorithm techniques and results that can be adopted to improve the performance for effective management of large amounts of moving object data and spatio-temporal data mining. We have developed algorithms by using the fuzzy clustering techniques in trajectory clustering step. So, this techniques that will make a difference. Through created graphical interfaces, we aim to show all of the moving object data results. In clustering process of direction based clustering algorithm DB-SMoT, fuzzy clustering methods have been used. Between developed algorithms, fuzzy k means based DB-SMoT has acquired the best performance. The fast running algorithm is fuzzy c means based DB-SMoT.

**2012, 102 pages**

**Keywords:** Stops and moves of trajectory, spatio-temporal data mining, semantic trajectory, trajectory clustering, fuzzy clustering.

## TEŞEKKÜR

Çalışmalarım boyunca bana her türlü maddi manevi destek sağlayan çok değerli danışman hocam Sayın Doç. Dr. A.Samet HAŞILOĞLU'na en içten teşekkürlerimi sunuyorum.

Çalışmaların sırasında her türlü desteği veren ve programın analizi, yazılımı ve test aşamalarında yardımcı olan mesai arkadaşlarımdan Sayın Arş. Gör. Ferhat BOZKURT'a ve Sayın Arş. Gör. Şeyma YÜCEL'e en içten teşekkürlerimi sunuyorum.

Tezim için gerekli verilerin temini konusunda bize yardımcı olan Erzurum Yakutiye Araştırma Hastanesi Bilgi İşlem Dairesine, tecrübeleri ve yorumlarıyla Erzurum Bölge Eğitim ve Araştırma Hastanesi Başhekimi Sayın Prof.Dr. Fazlı ERDOĞAN 'a teşekkürlerimi sunarım.

Ayrıca bütün çalışmalarımda olduğu gibi yine maddi ve manevi hiçbir yardımdan kaçınmayan, minnettar olduğum kıymetli aileme şükranlarımı sunuyorum.

Mete YAĞANOĞLU

Aralık 2012

## İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
SİMGELER ve KISALTMALAR DİZİNİ.....	vii
ŞEKİLLER DİZİNİ.....	viii
ÇİZELGELER DİZİNİ.....	x
<b>1. GİRİŞ.....</b>	<b>1</b>
1.1. Durmaların ve Hareketlerin Yörüngesi.....	6
1.1.1. Hareketli nesne.....	6
1.1.2. Durmalar.....	10
1.1.3. Hareketler.....	10
1.1.4. Yörünge.....	11
1.1.5. Anlamsal yörünge.....	13
1.1.6. Sorgulama.....	15
1.1.7. Durmaların ve hareketlerin yörüngeleri.....	17
<b>2. KAYNAK ÖZETLERİ.....</b>	<b>22</b>
2.1. Kümeleme.....	22
2.2. Algoritmalar.....	24
2.2.1. ID3.....	24
2.2.2. C4.5.....	24
2.2.3. Sınıflandırma ve Regresyon Ağaçları (CART).....	25
2.2.4. Araştırmada Denetimli Öğrenme (SLIQ).....	25
2.2.5. AIS.....	25
2.2.6. Geniş Uygulamaların Kümelenmesi (CLARA).....	26
2.2.7. Rasgele Aramaya Dayalı Geniş Uygulamaların Kümelenmesi (CLARANS).....	26
2.2.8. Medoid Etrafının Bölünmesi (PAM).....	26
2.2.9. CURE.....	27
2.2.10. CHALEMON.....	27
2.2.11. BIRCH.....	28

2.2.12. Kümeleme Yapısını Belirleyecek Noktaların Sıralanması (OPTICS) .....	28
2.2.13. Yoğunluk Tabanlı Kümelenme (DENCLUE) .....	28
2.2.14. Aramada Kümeleme (CLIQUE) .....	28
2.2.15. WEKA .....	29
2.2.16. WEKA-STPM .....	29
2.2.17. WEKA-GDPM .....	29
2.2.18. Yoğunluk Tabanlı Mekânsal Kümeleme (DBSCAN) .....	30
2.2.19. Mekansal-Zamansal DBSCAN (ST- DBSCAN) .....	31
2.2.20. STMPE .....	32
2.2.21. DJ-Cluster .....	33
2.3. Diğer Kaynak Özetleri .....	33
2.4. Durmalar ve Hareketlerin Yörüngeleri Algoritmaları .....	40
2.4.1. Durmaların ve hareketlerin yörüngesi (SMoT) .....	40
2.4.2. Kümeleme-tabanlı durmaların ve hareketlerin yörüngesi (CB-SMoT) .....	40
2.4.3. Kesişme-tabanlı durmaların ve hareketlerin yörüngesi (IB-SMoT) .....	41
2.5. Durmalar ve Hareketlerin Yörüngeleri için Sorgulama Dili .....	42
<b>3. MATERYAL ve YÖNTEM .....</b>	<b>45</b>
3.1. Yön-tabanlı Durmaların ve Hareketlerin Yörüngesi (DB-SMoT) .....	45
3.2. Geliştirilen Algoritmalar .....	48
3.2.1. Bulanık C-Ortalamlar DB-SMoT .....	48
3.2.2. Bulanık K-Ortalamlar DB-SMoT .....	50
3.2.3. Bulanık Medoid DB-SMoT .....	54
3.2.4. Bulanık Adaftif DB-SMoT .....	56
3.3. Sistem Altyapısı ve Mimarisi .....	59
3.4. Program: Petek Model .....	64
<b>4. ARAŞTIRMA BULGULARI ve DENEYSEL SONUÇLAR .....</b>	<b>73</b>
4.1. DB-SMoT Deneysel Sonuçlar .....	73
4.2. Bulanık C-Ortalamlar DB-SMoT Deneysel Sonuçlar .....	75
4.3. Bulanık K-Ortalamlar DB-SMoT Deneysel Sonuçlar .....	77
4.4. Bulanık Medoid DB-SMoT Deneysel Sonuçlar .....	80
4.5. Bulanık Adaftif DB-SMoT Deneysel Sonuçlar .....	82
<b>5. SONUÇ ve ÖNERİLER .....</b>	<b>86</b>



KAYNAKLAR .....	88
EKLER.....	97
EK 1. ....	97
EK 2. ....	98
EK 3. ....	99
EK 4. ....	100
EK 5. ....	101
EK 6. ....	102
ÖZGEÇMİŞ .....	103

## SİMGELER ve KISALTMALAR DİZİNİ

### Kısaltmalar

GPS	Global Konumlandırma Sitemi
RFID	Radyo Frekansı ile Tanımlama
SMoT	Durmaların ve Hareketlerin Yörüngesi
DB-SMoT	Yön-tabanlı Durmaların ve Hareketlerin Yörüngesi
FCM	Bulanık c ortalama
FCM-DBSMoT	Bulanık c ortalamalar tabanlı DB-SMoT
FKM	Bulanık k ortalama
FKM-DBSMoT	Bulanık k ortalamalar tabanlı DB-SMoT
FM	Bulanık medoid
FM-DBSMoT	Bulanık medoid tabanlı DB-SMoT
FA	Bulanık adaftif
FA-DBSMoT	Bulanık adaftif tabanlı DB-SMoT
MinTime	Minimum Zaman
maksTol	Maksimum Tolerans
minDirChange	Minimum Yön Değişimi
CURE	Temsilciler kullanarak kümeleme
BIRCH	Dengeli tekrarlayan azalma ve hiyerarşi kullanarak kümeleme
CART	Sınıflandırma ve Regrasyon Ağaçları
CLARA	Geniş Uygulamaların Kümelenmesi
CLARANS	Rasgele Aramaya Dayalı Geniş Uygulamaları Kümeleme
PAM	Medoid Etrafının Bölünmesi
DENCLUE	Yoğunluk Tabanlı Kümeleme
CLIQUE	Aramada Kümeleme
OPTICS	Kümeleme Yapısını Belirleyecek Noktaların Sıralanması
SLIQ	Araştırmada Denetimli Öğrenme

## ŞEKİLLER DİZİNİ

Şekil 1.1. Durmalar ve hareketler .....	11
Şekil 1.2. Hastane Petek Model’de yörünge örneği.....	13
Şekil 1.3. Anlamsal yörünge örneği.....	14
Şekil 1.4. Anlamsal hücre örneği .....	15
Şekil 1.5. Anlamsal yörünge Petek Modeli.....	15
Şekil 1.6. Mekansal-zamansal veri madenciliği sorgulama dili taslağı .....	16
Şekil 1.7. Durmaların ve hareketlerin olduğu yörünge .....	19
Şekil 3.1. Sistem mimarisi .....	60
Şekil 3.2. Veri tabanı varlık-ilişki (ER) diyagramı .....	62
Şekil 3.3. Durmaların olduğu yörünge .....	63
Şekil 3.4. Programa giriş.....	64
Şekil 3.5. Petek Model grafiksel arayüz.....	65
Şekil 3.6. Program veri seçimi .....	65
Şekil 3.7. Algoritma ve parametreler .....	66
Şekil 3.8. Program sonuçları.....	66
Şekil 3.9. Hastalık yörüngesi .....	67
Şekil 3.10. Hasta takip .....	68
Şekil 3.11. Harita üzerinde hastalık kümelemesi .....	68
Şekil 3.12. Erzurum ili hastalık kümelemesi .....	69
Şekil 3.13. Doğu Anadolu Bölgesi hastalık kümelemesi.....	69
Şekil 3.14. Yörünge kümeleme.....	70
Şekil 3.15. Laboratuvarın yörünge kümelemesi .....	71
Şekil 3.16. 3-Boyutlu yörünge kümeleme .....	71
Şekil 4.1. MinTime parametresine göre Durmalar.....	73
Şekil 4.2. MinTime parametresine göre BilinmeyenDurmalar .....	74
Şekil 4.3. MinTime parametresine göre Yörünge Noktası .....	74
Şekil 4.4. MinTime parametresine göre Yön Değişimi .....	75
Şekil 4.5. Durma sayısının DB-SMoT/FCM-DBSMoT karşılaştırılması.....	76
Şekil 4.6. BilinmeyenDurma sayısının DB-SMoT/FCM-DBSMoT karşılaştırılması ...	76

Şekil 4.7. Yörünge Noktası sayısının DB-SMoT/FCM-DBSMoT karşılaştırılması .....	77
Şekil 4.8. Yön Değişiminin DB-SMoT/FCM-DBSMoT karşılaştırılması .....	77
Şekil 4.9. Durma sayısının DB-SMoT/FKM-DBSMoT karşılaştırılması.....	78
Şekil 4.10. BilinmeyenDurma sayısının DB-SMoT/FKM-DBSMoT karşılaştırılması..	79
Şekil 4.11. Yörünge Noktası sayısının DB-SMoT/FKM-DBSMoT karşılaştırılması ....	79
Şekil 4.12. Yön Değişiminin DB-SMoT/FKM-DBSMoT karşılaştırılması .....	80
Şekil 4.13. Durma sayısının DB-SMoT/FM-DBSMoT karşılaştırılması .....	81
Şekil 4.14. BilinmeyenDurma sayısının DB-SMoT/FM-DBSMoT karşılaştırılması.....	81
Şekil 4.15. Yörünge Noktası sayısının DB-SMoT/FM-DBSMoT karşılaştırılması .....	82
Şekil 4.16. Yön Değişiminin DB-SMoT/FM-DBSMoT karşılaştırılması .....	82
Şekil 4.17. Durma sayısının DB-SMoT/FA-DBSMoT karşılaştırılması .....	83
Şekil 4.18. BilinmeyenDurma sayısının DB-SMoT/FA-DBSMoT karşılaştırılması .....	84
Şekil 4.19. Yörünge Noktası sayısının DB-SMoT/FA-DBSMoT karşılaştırılması.....	84
Şekil 4.20. Yön Değişiminin DB-SMoT/FA-DBSMoT karşılaştırılması.....	85

## ÇİZELGELER DİZİNİ

Çizelge 1.1. Mekansal-zamansal veri madenciliği sınıflandırılması .....	4
Çizelge 1.2. Hareketli nesnelere için uygulamalar .....	9
Çizelge 1.3. Durmaların verileri.....	20
Çizelge 1.4. Hareketlerin verileri .....	20
Çizelge 2.1. Sorgu örneği.....	42
Çizelge 2.2. DB-SMoT algoritması kullanarak sorgulama.....	43
Çizelge 2.3. DB-SMoT algoritması için genelleştirilmiş sorgulama .....	44
Çizelge 4.1. MinTime değeri 500 iken DB-SMoT algoritması sonuçları.....	73
Çizelge 4.2. MinTime değeri 500 iken DB-SMoT/FCM-DBSMoT karşılaştırılması ....	75
Çizelge 4.3. MinTime değeri 500 iken DB-SMoT/FKM-DBSMoT karşılaştırılması ....	78
Çizelge 4.4. MinTime değeri 500 iken DB-SMoT/FM-DBSMoT karşılaştırılması.....	80
Çizelge 4.5. MinTime değeri 500 iken DB-SMoT/FA-DBSMoT karşılaştırılması .....	83

## 1. GİRİŞ

Son yıllarda yaşanan bilişim teknolojilerindeki hızlı gelişmeler ile birlikte, dünyadaki veri ve gizli bilgi miktarının artışı akıl almaz boyutlara ulaşmıştır. Günümüz insanların yaptığı her alışveriş, her bankacılık işlemi, her telefon görüşmesi, gittiği her doktor muayenesi, kısacası her elektronik işlem kayıt altına alınmaktadır. Dolayısıyla bu şekilde uzaktan algılayıcılardan, uydulardan, sensörlerden, optik okuyuculardan toplanan, saklanan veriler her an inanılmaz boyutta artmaktadır. Bu çok farklı alanlardan oldukça artan bir hızda elde edilen veriler toplanılmakta ve biriktirilmektedir. Böylece büyük veri yığınları gün geçtikçe artmaktadır (Şentürk 2006).

Elektronik veri saklama ve analiz araçlarının gelişimi büyük miktarda veriyi işleme yeteneğine sahip teknolojilerin üretilmesini sağladı. Bu teknolojilerinin en yenileri veri ambarları ve veri madenciliğidir. Veri madenciliği 1980'lerin sonunda geliştirilen, 90'lı yıllarda büyük bir gelişme gösteren ve uygulama alanları artan, yeni binyılda da bu gelişimini sürdürmesi beklenen, veri temelli karar alma süreçlerinde önemli katkıları olan bir teknolojidir (Han and Kamber 2000; Beitel 2005;Özçınar 2006).

Günümüzde artan veri sayısı, bilgisayar ve iletişim teknolojilerindeki gelişmeler çok daha fazla verinin hızlı bir şekilde depolanmasına, işlenmesine ve bilgiye dönüştürülmesine imkan sağlamaktadır. Bu teknolojik gelişmeler ve verilerdeki hızlı artış toplanan bu verilerden nasıl faydalanılacağını ve bu verilerin nasıl anlamlı hale getirileceği sorununu ortaya çıkarmıştır. İşte bu sorunların çözümüne imkan sağlayan veri madenciliği son yıllarda yoğun olarak kullanılmaya başlanmıştır (Akın 2008).

Büyük ölçekli veriler arasından yararlı bilgiyi çıkarma işlemine veri madenciliği denir. Veri madenciliği sayesinde veriler arasındaki ilişkileri ortaya koyulur ve gerektiğinde ileriye yönelik tahminlerde bulunmak mümkün olabilecektir. Yani veri madenciliği üretilen tüm verilerin belirli yöntemler kullanarak var olan ya da gelecekte ortaya çıkabilecek gizli bilgiyi ortaya çıkarma süreci olarak değerlendirilebilir (Özkan 2008).

Veri madenciliđi, istatistik ve matematik tekniklerle birlikte örüntü tanıma teknolojilerini kullanarak, depolama ortamlarında saklanmış bulunan veri yığınlarının elenmesi ile anlamlı yeni korelasyon, örüntü ve eğilimlerin keşfedilmesi sürecidir (Akpınar 2000).

Büyük veriler arasından yararlı bilgiye ulaşma, gittikçe artan veri miktarının etkili bir biçimde kullanılmasının çözümü veri madenciliđidir. Bu durumdan dolayı veri madenciliđi diđer alanlarda olduđu gibi sađlık alanında da ilgi odađı haline gelmiştir. Özellikle sađlık alanındaki verinin büyük olması ve daha da önemlisi verinin hayati önem taşıması bu alandaki uygulamaları oldukça önemli kılmaktadır. Hastane otomasyonları sayesinde bu veriler düzenli olarak tutulmaktadır. Hayati öneme sahip olan bu verilerden daha fazla yararlanmak mümkündür. Hastane otomasyonlarından veya diđer tıbbi veri toplayan sistemlerden alınan veriler üzerinde yapılan veri madenciliđi çalışmaları hem uzmanlar için hem hastane yönetimi için hem de hastaların daha kaliteli bir hizmet almalarında etkin rol oynar (Kaya vd 1996).

Mekânsal boyut, ele alınan nesnelerin sabit bir konumla (örneğin zemine bağlanan sensörlerle toplanan bilgi) ilgili olup olmadığını ya da hareket edebilip edemediklerini yani konumlarının dinamik ve zamanla deđişebilip deđişemediklerini gösterir. (Kisilevich *et al.* 2010).

Mekânsal veri madenciliđi görevleri ve teknikleri bölümlene dahil beş kategoride; bağımlılık analizi, sapma ve aykırı deđer analizi, eğilim keşfi ve genelleştirme ve karakterizasyon şeklinde sınıflandırılabilir (Fayyad *et al.* 1996; Miller and Han 2001).

Zamansal boyut, nesnenin gelişiminin verilerle hangi ölçüde yakalandığını gösterir. Temel durum ise hiç gelişim göstermeyen nesnelere meydana gelir, bu durumda her nesnenin yalnızca durgun anlık görüntüsü mevcuttur. Biraz daha kompleks ortamlarda, her nesne konumunu deđiştirebilir, yalnız en yakın zamandaki deđer (yani güncellenmiş anlık görüntüsü), dolayısıyla da eski tarihi hakkında bir bilgi olmaksızın

bilinmektedir. Son olarak, nesnenin tüm tarihinin tutulduğu, böylece geçtiği konumun zaman serisinin oluşturulduğu ekstrem durumu ele alınabilir (Kisilevich *et al.* 2010).

Mekânsal-zamansal veri madenciliği hareketli nesnelere hareket desenlerini çıkaran bir çalışma alanıdır (Han *et al.* 1999). Mekânsal-zamansal veri madenciliğinin amaçlarından biri yer ve zaman bağıntılarını ortaya çıkarmaktır. Hem metrik (uzaklık gibi) hem de metrik olmayan yer bağıntıları ve zaman bağıntıları (daha önce ya da daha sonra gibi) tıbbi veri tabanlarında keşfedilerek anlamsal eğriler oluşturulabilir (Yao 2003).

Mekânsal-zamansal veri madenciliği, veri tabanlarının hem zaman hem de mekân açısından desteklendiği bir türüdür. Zamanla geometrisi değişen veri tiplerini yönetir (Rahim 2007). Telekomünikasyondaki gelişmelerden dolayı (GPS (Global konumlandırma sistemi), RFID (Radyo frekansı ile tanımlama) , cep telefonu vs.) mekânsal-zamansal veri tabanlarındaki veri miktarı hızla artmıştır. Büyük miktarda veri içeren bu veri tabanlarının analizi için alışılmışın dışındaki hesaplama tekniklerinin uygulanması ve geliştirilmesi amacıyla mekânsal-zamansal veri madenciliği adı altında bir araştırma alanı ortaya çıkmıştır (Gennady *et al.* 2006).

Mekânsal-zamansal verilerinden elde edilen bilgi olayların daha iyi öngörülmesi için bize yardımcı olur. Bu nedenle mekânsal-zamansal veri madenciliğinde araştırmayı gerçekleştirmek oldukça önemlidir.

Veri madenciliğinin tanımına paralel olarak, burada mekânsal-zamansal veri madenciliğinin dolaylı bilginin, yer ve zaman bağıntıları ya da mekânsal-zamansal veri tabanlarında açıkça yüklenmeyen diğer örneklerin elde edilmesiyle ilgilidir. Bu, veri tabanlarında veri madenciliğinin ve bilgi keşfinin bir alt alanıdır, son on yılda bilgisayar biliminde ve bilgi teknolojisinde başlayan bir araştırma alanıdır ve şimdi hemen hemen veri bakımından zengin tüm alanlarda kullanılmaktadır. Coğrafi enformasyon ve tıp biliminde, mekânsal-zamansal veri madenciliği çalışması yakın zamanda başlamıştır



(Roddick and Lees 2001; Yağanoğlu vd 2011). Hem mekân hem de zaman önekleri, veri madenciliği işlerine önemli ölçüde kolaylaştırmıştır (Koperski *et al.* 1996, 1998; Shekhar *et al.* 2001).

Çizelge 1.1, Mekânsal-zamansal veri madenciliği görevleri ve tekniklerinin sınıflandırılışını göstermektedir.

**Çizelge 1.1.** Mekânsal-zamansal veri madenciliği sınıflandırılışı

Mekânsal-zamansal veri madenciliği görevi	Tanımlar	Teknikler	
		Durgun zaman verisi	Mekânsal-zamansal Veri
Bölümleme	Kümelenme Sınıflandırma	<ul style="list-style-type: none"> <li>◆ Kümelenme analizi</li> <li>◆ Bayes sınıflandırması</li> <li>◆ Karar ağacı</li> <li>◆ Yapay sinir ağları</li> </ul>	<ul style="list-style-type: none"> <li>◆ Kümelenmenin zamansal genişletilmesi</li> <li>◆ Sınıflandırmanın zamansal genişletilmesi</li> </ul>
Bağımlılık analizi	Zamanla diğer özelliklerin değerini dayanan bir özelliğin değerini öngörecektir kuralları bulmak	<ul style="list-style-type: none"> <li>◆ İlişkilendirme kuralları</li> <li>◆ Bayes ağları</li> </ul>	<ul style="list-style-type: none"> <li>◆ Zamansal ilişkilendirme kuralları</li> <li>◆ Bayes ağlarının zamansal genişletilmesi</li> </ul>
Sapma ve aykırı değer analizi	Beklenenler kümesi ve diğer veri inceleme metodlarından alışılmamış sapmaları sergileyen veri birimlerini bulmak.	<ul style="list-style-type: none"> <li>◆ Kümelenme ve diğer veri madenciliği metodları</li> <li>◆ Aykırı değerlerin ortaya çıkarılması</li> </ul>	Sol sütundaki tekniklerin zamansal genişletilmesi
Eğilimin keşfi	Doğruların ve eğrilerin tahmini Çoğunlukla zaman üzerinden veri tabanının özetlenmesi Dizilerdeki olaylar arasında korelasyonların keşfi.	<ul style="list-style-type: none"> <li>Ortak eğilimlerin keşfi</li> <li>◆ Gerileme</li> </ul>	◆ Sıra madencilik
Genelleştirme ve karakterizasyon	Verilerin yoğun tanımları	<ul style="list-style-type: none"> <li>◆ Bayes ağları</li> <li>◆ Özellik yönlendirilmiş tümevarım</li> </ul>	Sol sütunda tekniklerin zamansal genişletilmesi

Çizelge 1.1, Mekânsal-zamansal veri madenciliği görevlerine uygular. Sağdaki son sütun en geçerli Mekânsal veri madenciliği tekniklerinin zamansal genişletişlerinin gerekliliğini gösterir.

Mekânsal-zamansal veri madenciliği, mekânsal-zamansal veri tabanları, otomatik öğrenme, istatistik, tıp, coğrafik görselleştirme ve enformasyon teorisi dahil birçok alanın birleşmesini gösterir. Mekânsal veri madenciliğinin ve zamansal veri madenciliğinin keşfi, çeşitli araştırma topluluğunda bağımsız şekilde dikkat çekmiştir. Yine de aynı zamanda hem yer hem de zaman bağıntılarını araştırmanın gereği veri madenciliği görevleri de epey önemlidir. Mekânsal-zamansal veri madenciliğinde dezavantajı ise, mekânsal-zamansal veri tiplerinin karmaşıklığı, bu verilerin temsili ve mekânsal verilerin yapısı nedeniyle etkili metotların keşfidir (Koperski *et al.* 1998; Miller and Han 2001; Roddick and Spiliopoulou 2002).

Mekânsal-zamansal veri madenciliği, daha önceden hızlı ve doğru şekilde toplanmış çok sayıda veriden, hareketli nesnelere hareket desenlerinin çıkarım çalışmasıdır (Bozkurt vd 2011; Yağanoğlu vd 2011).

Bir hareketli nesne verisini geleneksel verilerden farklı kılan özellik bu hareketi tanımlamakta ihtiyaç duyulan mekân ve zaman boyutlarının varlığıdır. Mekân ve zamana göre hareketli nesnenin izlenme tarihçesi sonucunda oluşan hareketli nesne verileri ile ilgilenmekteyiz. Hareket bilgileri somut (insanlar, araçlar, hayvanlar veya ürünler vb.) veya soyut (bir hastalığın, bilgisayar virüsünün veya terör faaliyetlerinin yayılımı) nesnelere ait olabilir. Bu ise hareketli nesnenin bıraktığı yörünge verisi üzerine geliştirilecek herhangi bir mekânsal-zamansal veri madenciliği yörünge algoritmalarının uygulanabilirlik alanının ne kadar geniş olabileceğini gösterir. Hareket desenlerinde yola çıkarak, hareketli nesnenin gelecekteki konumu hakkında tahminde bulunma birçok alan için önemli bir gelişmedir.

Günümüzde mekânsal-zamansal veri madenciliği yörünge durma ve hareket algoritmaları kümeleme tabanlı, yön tabanlı ve kesişme tabanlı durma ve hareket algoritmaları olarak ele alınır. Zamanla yarışılan bazı kritik problemlerde (örneğin ölümcül salgın bir hastalığın yayılış seyrinin tahmini) yörüngelerde belirsiz veri probleminin üstesinden gelmek adına bulanık kümeleme yöntemlerinin kullanılmasının fayda sağlayabileceği öngörülmektedir.

Kümeleme analizi, verileri gruplandırmada kullanışlı birçok değişkenli analiz tekniğidir. Kümeleme analizinin amacı, gruplanmamış verileri benzerliklerine göre sınıflandırmak ve araştırmacıya özetleyici bilgiler elde etmede yardımcı olmaktır. Kümeleme analizi farklı sorunların giderilmesinde kullanılan önemli bir analizdir. Bu analiz, doğru grupların bulunmasında yararlıdır (Akın 2008).

Kümeleme analizi mekânsal-zamansal veri madenciliğinin en önemli alanlarından birisidir; amacı, nesnelere birbirlerine olan benzerliklerine göre gruplara ayırmaktır. Elde bulunan veriler incelenerek birbirine benzeyenler bir kümeye, benzemeyenler ise başka bir kümeye toplanırlar.

## **1.1. Durmaların ve Hareketlerin Yörüngesi**

### **1.1.1. Hareketli nesne**

Durmaların ve hareketlerin olduğu her uygulamada hareketli nesneden bahsedebiliriz. Örneğin; insan bir mekandan başka bir mekana hareket ediyorsa bu bir hareketli nesne verisidir. Burada hareket ederken durmalara uğruyorsa ve bu durmalar yörünge verisi için önemli noktalar ise, bu hareketli nesne verisi bir anlamsal yörüngede bizim için önemlidir. Bunun dışında örnek vermek gerekirse; hayvanların göçü, uydu (uzay aracı) yörüngesi, araç trafiği (taksilerin yörüngesi), uçak, gemi, roket v.b. hareketli nesne verisi olarak düşünülebilir. Taksinin durağı yörüngenin başlangıcı olsun, taksinin

müşterilerini götürdüğü her an yeni bir anlamsal yörünge örneği oluşturur. Burada taksinin beklediği kavşaklar, dönemeçler ve buralardaki bekleme sürelerine göre trafik akışı düzeltilebilir, ek yollar ve güzergâhlar belirlenebilir. Hareketli nesne verisi sektör olarak ise; turizm, ulaşım ve sağlık alanlarında sıklıkla görülmektedir.

Bu örneklerin dışında; bir hastalığın yayılımı da hareketli bir nesnedir. Kanserin başlangıcı, yayılışı, görüldüğü yer ve zaman; hareketli nesne verisini oluşturur. Bunun dışında buzullardaki buzların çözülmesi de hareketli nesne verisidir. Belli bir zamanda küresel ısınmanın da etkisiyle buzullardaki buz kütlelerinde çözülme olur ve bu yörünge verisi incelenerek çeşitli çıkarımlar yapılabilir.

Ülkelerin ekonomik gelişmesi de yörünge verisine örnek olarak verilebilir ve bu aynı zamanda bir hareketli nesnedir. Bu yörüngede ülkelerin ekonomik durumu hangi durumlarda düştüğü ya da hangi durumlarda yükselişe olduğu tespit edilerek anlamsal yörünge eğrisi çıkartılabilir. Bu çıkarımlarla ülkelerin sosyal, ekonomik ve refah düzeyi artırılabilir.

İnsanların ve hayvanların göçü en belirgin hareketli nesne verisidir. Örneğin bir karganın hareketleri incelenerek yörüngesi çıkartılarak karganın uzun yaşamda izlediği yollar ve hareketleri incelenebilir. Bunun için kargaya öncelikle bir çip takılmalı ve bu çip sayesinde durmaları ve hareketleri bulunmalıdır. Daha sonra bu anlamsal yörüngeye algoritmalar uygulanarak kargaların doğadaki hareketleri incelenebilir ve faydalı sonuçlar çıkartılabilir.

Suyun sıcaklığının değişimi, 0'ın altına düşmesi, 0'ın üstüne çıkması ve bu durumların hangi koşullarda olduğu bilgisi bir hareketli nesne verisidir. Kısaca özetlemek gerekirse hareketin olduğu her veriden anlamsal yörüngeler çıkarılabilir. Bu yörüngelere geliştirdiğimiz algoritmalar sayesinde görsel ara yüzler ile sonuçlandırılabilir.

Hareketli nesne kümelerini bulma problemi doğrudan birkaç yakın zamanlı çalışmada dile getirilmiştir. Kalnis *et al.* (2005) zamanı ayırık zaman adımlar kümesi olarak ele alır. Her bir adımda nesnelerin konumları bağımsız statik veri takımları olarak ele alınır. Kümeler, hareketli bir kümeleme oluşturmak için bitişik zaman adımlarında oluşturulan kümelerle ilişkilidir. Bu teknik birbirleriyle çakıştıkları yerde ya da gürültü nesnelere mevcut iken periyotlarla hareketli kümeleri takip edememektedir.

Giannotti *et al.* (2010), hareketli nesnelerin birikmiş davranışını veren olağan hareket örneklerini bulmak üzere bir algoritma önerdiler burada T-örneği denilen bir örnek ardışık noktalar arasında zaman bağlantılarıyla verilen bir noktalar dizisi olarak tanımlanmıştır. Bir T-örneğin zaman ve mekan bileşenleri, yaklaşık olarak girdi yörüngelerine karşılık gelirse bulgulanır. Bu örneklerin anlamı, farklı nesnelerin benzer zaman aralıklarıyla aynı yerlere uğramasıdır. Örnekler bulunur bulunmaz, algoritmaları inceleyen klasik dizi olağan örneklerin bulunması için uygulanabilir. T-örneklerinin belirlenmesi uğranan bölgelerin tanımı için önemlidir (Kisilevish *et al.* 2010).

Kang and Yong (2009) çalışmasında, hücre uzunlukları çok büyükse çalışma zamanının gridlere ayrışımına dayanan metotlar bazı örneklerin kaybolmasına neden olabilir. Ayrıca aşırı teşekküle yol açabilen bazı metotların kayıtlı zaman-bilgilerine ve zaman görünümünün dizisel sıralı bölge kimliklerinde bulunduğu tekrarlı dizilere göre, bu metotlar yörünge ayrıklaştırmasını gerektirir. Yazarlar, bu sorunlara incelikli bir çözüm olarak iki saflaştırma önermiştir: (1) yörüngeleri ayırık parçalara ayrıştırmak ki bu nesnenin hareketinin anlamlı zamansal-mekânsal değişikliklerini temsil eder. Parça, alandaki devam süresi gibi başlangıç ve bitim uçları olan noktalar şeklinde tanımlanır. (2) Benzer parçaları gruplayacak kümeleme algoritmaları uygulamak. Bir zamansal-mekansal örnek devam süresi 3-boyutlu küp yüksekliği şeklinde tanımlı parça (alan ) dizileri olarak tanımlanmıştır. Böylece zamansal-mekansal örnek dizileri benzer küpler kümelenecek oluşturulur.

Çizelge 1.2’de hareketli nesnelere için kullanılan metot ve uygulamalar gösterilmiştir.

**Çizelge 1.2.** Hareketli nesnelere için uygulamalar

Kategori	Problem	Uygulama	...e dayanan metot	Seçilen literatür
Hareket	Yörünge kümelenmesi, Yörünge yığılması, Yörünge genelleştirme si	Otomobiller, tahliye izleri, göçmen vapur çıkarmaları ve önlemleri	OPTICS	Nanni and Pedreschi (2006) Rinzivillo et al.(2008) Andrienko and Andrienko(2008) Andrienko and Andrienko(2009) Andrienko <i>et al.</i> (2009)
	Hareketli kümeler	Göç eden hayvanlar, sürüler, araç konvoyları	DBSCAN Gridleme, DBSCAN	Kalnis <i>et al.</i> (2005) Jeung <i>et al.</i> (2008) Vieira <i>et al.</i> (2009)
	Yörüngelerden önemli yerler çıkaran	İnsanların oluşturdukları yörüngeler	DBSCAN Artan kümelenme	Palma <i>et al.</i> (2008) Kang <i>et al.</i> (2004)
	Yörünge örnekleri	Yük vagonu filoları	Uzam bölgelerinin yoğunluğu	Gianotti <i>et al.</i> (2007)
		Sentetik veriler	BIRCH	Kang and Yong (2009)
Hüresel şebekler	Kent aktivitesi	Telefon aramaları	k-means	Reades <i>et al.</i> (2007)
Ortam	Oşinografi	Deniz suyu dağılımı	DBSCAN	Birant and Kut (2006,2007)
	Sismoloji	Sismik aktivite	Gridleme, DBSCAN	Wang <i>et al.</i> (2006)

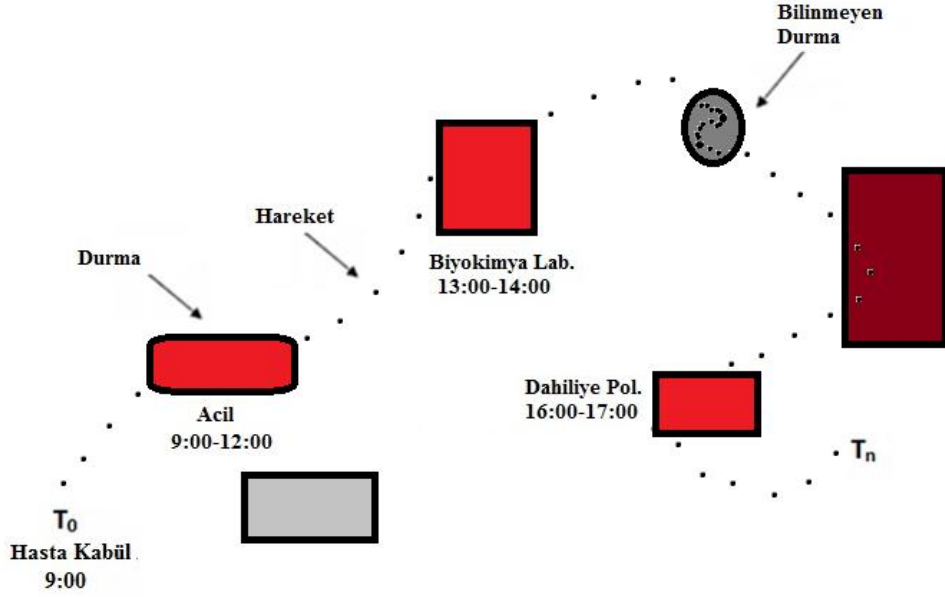
Büyük veritabanlarında hareketli nesnelere için mekânsal-zamansal veri madenciliği kullanarak yapılan çalışmada, mekânsal-zamansal veri tiplerine dayanan hareketli nesne noktaları için bir örüntü belirleme metodu sunulmuştur. Mekânsal-zamansal veri tipleri geliştirilmiştir ve bu veri tipleri veritabanıyla bütünleştirilmiştir (Rahim 2007).

### 1.1.2. Durmalar

Durmalar, yörünge'nin bir parçasıdır. Bir yörüngede durmalar özel bir yeri, bir zaman aralığında temsil ediyorsa anlam kazanmaktadır. Bu anlamsal yörüngelerde durmaların olduğu yerler, o yörünge için önemli noktalar'dır. Örneğin bir hasta  $t$  süresinde, bir lokasyonda (klinik, kan merkezi v.b.) duruyorsa, bu durma o zamanda anlamsal eğrinin durması'dır. Bir anlamsal yörüngedeki durmalar, o veri için bizim çıkarımlar yapmamızı sağlar. Örneğin hastanede kan merkezinde olan bir durma; burada geçirdiği süre ve yoğunluğuna bakılarak bu lokasyonda sürecin yavaş ilerlediği bilgisini çıkartabiliriz. Bu lokasyonda eleman ihtiyacının karşılanmasıyla sürecin hızlanabileceği sonucunu çıkartabiliriz. Buradaki yoğunluğun bu şekilde önlenmesiyle hem hasta güvenliği hem de zaman kazanılmış olunur. Durmalar, bir yörüngedeki en önemli noktalar'dır.

### 1.1.3. Hareketler

Bir hareket, yörüngedeki bir durmadan başka bir durmaya giderken geçirdiği konum değişikli'dir. Yani bir zamanda bir mekândan başka bir mekâna hareket söz konusudur. Anlamsal yörüngelerde durmaların, yörünge için en önemli noktalar olduğu daha önce bahsedilmişti. İşte bu önemli noktaya gelirken, bu noktada beklerken ve bu noktadan başka bir noktaya giderken geçirdiği zaman çok önemlidir. Örneğin biyokimya laboratuvarı anlamsal bir yörüngedeki bir durma olsun. Bu anlamsal yörüngede diğer durmamız ise kan merkezi olsun. Bir hastanın (ya da hastaların) biyokimya laboratuvarından, kan merkezine giderken izlediği yola hareket denir. Bu hareket uygulamalar için çok önemlidir. Bu hareketin başlangıç ve bitiş zamanı veritabanında tutulmaktadır. Bu sayede bir kişinin yaptığı hareket ile oluşan yörüngede; durmalar çok rahatlıkla görülebilmekte ve bu hareket ile ilgili çıkarımlar yapılabilir.



**Şekil 1.1.** Durmalar ve hareketler

Şekil 1.1 'de görüldüğü gibi durmalar ve hareketler gösterilmiştir. Durmalar, bu örnek için hastanın beklediği önemli yerlerdir. Hasta önce hasta kabüle gelir, burada yörüngesi oluşmaya başlar. Yaptığı hareketler ise onu yeni bir durmaya götürür. Hareketin bittiği en son yer ise yörüngemizin son noktasıdır.

#### 1.1.4. Yörünge

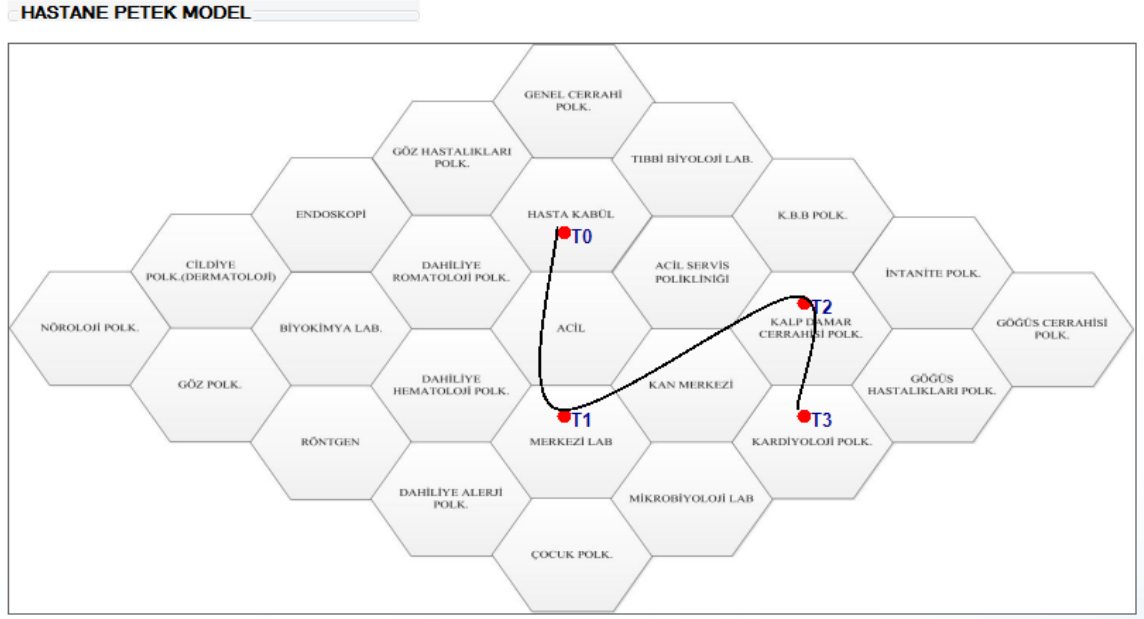
Bir hareketli nesnenin bir noktadan başka bir noktaya hareketi boyunca izlediği yola yörünge denir. Yörüngelerde bulunan örneklere yörünge örnekleri denir ve tek bir nesnenin ya da hareketli nesnelere grubunun ilginç davranışlarını karakterize eder. Farklı yaklaşımlar yörünge örneklerini inceleme konusunda mevcuttur. İki örnek verilmiştir; İlki grid-bazlı kümelenmeye ve yoğun bölgeler bulmaya dayanır, ikincisi yörüngelerin parçalanışının ve yörüngeler parçalarının kümelenmesine dayanır (Giannotti *et al.* 2007; Fosca and Dino 2008; Kang and Yong 2009).

Yörünge veri madenciliği teknikleri, zaman boyutunu ayrık aralıklara böler ve her zaman aralığında hareketli bir nesnenin konum listesi şeklinde temsil eder. Kümelenme



teknikleri benzer yörüngeli grupları bulmak için kullanılır (Gaffney and Smyth 1999; Vlachos *et al.* 2002; Nanni and Pedreschi 2006; Elnekave *et al.* 2008). Bu teknikler her hareketli nesnenin geçmiş konumlarını yüklemelidir. İki yörüngeyi karşılaştırmak için kullanılan benzerlik fonksiyonu hesap açısından kompleks bir prosesi kümelenemeyi sağlayan tüm geçmiş konumu da ele almalıdır.

Kang *et al.* (2004) çalışmasında yazarlar tek bir yörüngede önemli yerlerin belirlenmesi için artımlı bir kümelene önerdiler. Algoritma için birçok faktör tanımlanmıştır: keyfi sayıda küme, mümkün olduğu kadar pek önemli olmayan yerlerin çıkarılması ve mobil cihazlarda çalışmaya izin verecek kadar geniş hesaplanabilirlik. Algoritma önemli yerler bulmaya dayanır burada birçok konum ölçümü birlikte kümeleneir. İki parametre, kümenin oluşturulmasını kümedeki konumlar arasındaki mesafeyi ve harcanan zamanı kontrol etmiştir. Temel tez şöyledir: Konum-bazlı cihazla elde edilen her yeni konum ölçümü önceki konumla karşılaştırılır. Önceki konumla arasındaki uzaklık bir eşikten daha küçük ise, yeni konum daha önce oluşturulan kümeye eklenir. Aksi durumda yeni aday küme yeni konumla oluşturulur. Bir kümedeki ilk nokta ile son nokta arasındaki zaman farkı eşikten daha büyük iken, aday küme önemli yerler kümesi olur. Yörüngelerde ilginç yerler bulunarak, uygulamalarda kolaylık sağlanabilir (Palma *et. al* 2007; Zheng *et al.* 2009; Kisilevish *et al.* 2010).



**Şekil 1.2.** Hastane Petek Model’de yörünge örneği

Şekil 1.2’de görüldüğü gibi Hastane Petek modelimizde bir yörünge izlediği yol gözükmektedir. Bu örnekte bir hastanın yörüngesi gösterilmiştir.

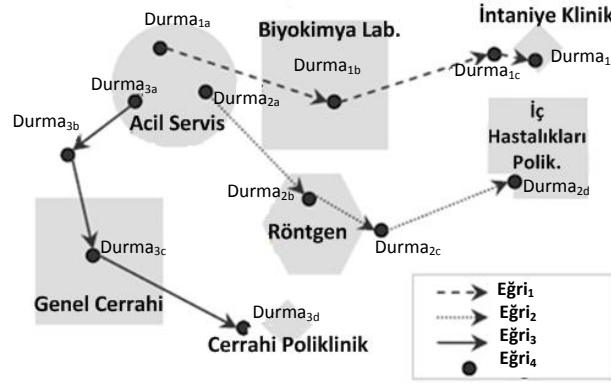
### 1.1.5. Anlamsal yörünge

Anlamsal yörüngelerde her durma özel bir yeri temsil eder. Örneğin hastanelerde acil servis bölümü bir durma lokasyonudur. Bu durmalardan, başka durmalara bir hareket söz konusu ise yörüngeler anlam kazanır. Bu yörüngelere de anlamsal yörünge denir.

Anlamsal yörünge durma ve hareket kümesinden oluşur. Durmanın bir yeri vardır, aynı zamanda başlangıç ve bitiş zamanına sahiptir. Hareketler de ardışık iki durma ile karakterize edilir. Anlamsal yörünge, coğrafi veri ve yörünge verinin birleşmesi ile ifade edilir (Bogorny 2010).

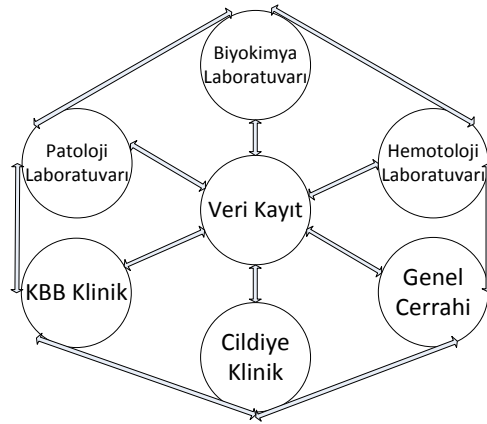
Sistemde bulunan hareketli nesne verisinin işlenmesiyle, sürekli durmaların ve hareketlerin olduğu, hız değişiminin yaşandığı yörünge verisine geliştirdiğimiz algoritmalar uygulanır. Bir yörüngesel hareket boyunca düşük hız değerlerinin

yaşandığı yerleri durmalar, hastane mekânları (acil servis, laboratuvar, klinik vs.) olarak belirlenir. Algoritma ile durma ve hareketlenme noktaları belirlenir. Bu sayedeki elimizdeki ham veriden, kliniklerde oluşan kümelenmeler, yoğunluklar, hasta hareketleri izlenmesi gibi bilgilere rahatlıkla ulaşılır (Bozkurt vd 2011).



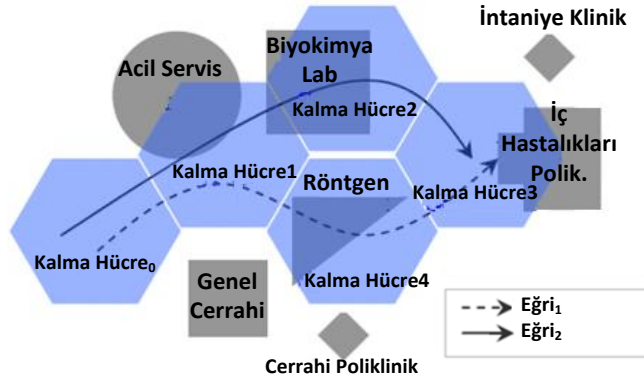
**Şekil 1.3.** Anlamsal yörünge örneği

Şekil 1.3’de olduğu gibi hareketli nesne olarak kabul ettiğimiz, hasta verilerinde bir hastanın sağlık kurumu içerisindeki durma ve hareketlerinden alınan veriler ile anlamsal yörüngeler oluşturulur. Bu yörüngeler sayesinde hareketli nesne verisinden yararlı bilgi elde edimi gerçekleştirilir.



**Şekil 1.4.** Anlamsal hücre örneği

Şekil 1.4'deki bir hücre yapısında olduğu gibi, her hücrede bölgeler ve bu bölgelerde oluşan yörünge eğrileri sonucunda anlamsal yörünge örneklerinden petek modeli oluşturulabilir.



**Şekil 1.5.** Anlamsal yörünge Petek Modeli

Şekil 1.5'deki modelin oluşturulmasıyla hasta takip sistemindeki sağlık hizmetlerinden alınan, hasta kayıt tarihçe bilgisinden, hastalığın görüldüğü yer kayıtlarından çıkarılan anlamsal yörünge petek modeli ile görsel olarak bilgiden çıkarım yapılır. Bir hastalığın hangi bölgede kümelendiği, klinik yoğunlukları, hastalığın yayılım eğrisi gibi birçok anlamsal bilgiye erişilebilir.

### 1.1.6. Sorgulama

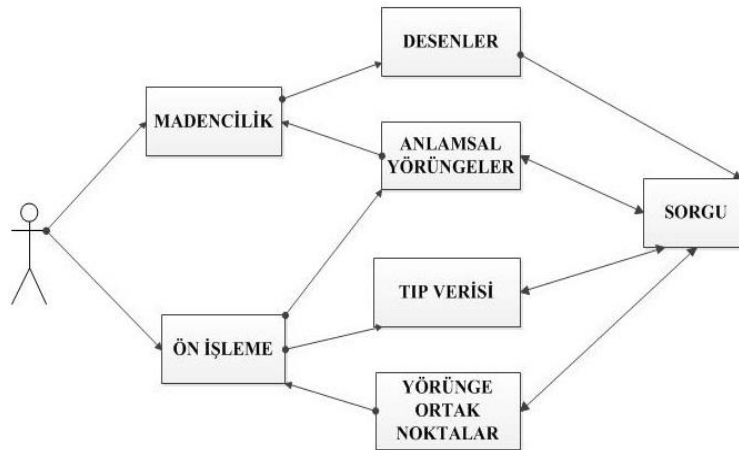
Mekânsal-zamansal veri tiplerinin birkaç farklı formu reel uygulamalarda mevcuttur. Tümü bir tür zaman ve mekan görünümü mevcudiyetini paylaşmasına rağmen böyle bir bilginin genişletiş ve aralarında kurulan bağlantı yolu birkaç farklı veri nesne tipiyle birleştirilebilir (Kisilevich *et al.* 2010).

Mekânsal-zamansal verinin bir örneği; sensörlerce yakalanan yer sarsıntıları ya da bir salgının yer-referanslı kayıtları gibi mekânsal-zamansal olaylardır. Her olay genellikle kayıt altına alındığı mekan ve karşılık gelen zaman bilgisiyle bağıntılıdır. Mekânsal-zamansal kümelenme, mekan ve zaman benzerliklerine dayanan nesnelere gruplama

işlemdir. Bu işlem, gerçek zamanlı nesnelere kümesini ya da bir nesnenin konumunu, zamanını veya çevresel özelliklerini kayıt altına alan tüm konum tabanlı ya da ortam cihazları türlerinin yaygınlaşması nedeniyle özellikle tıp bilimlerinde yüksek popülarite kazanan veri analizinin ilgili yeni alt alanıdır (Kisilevich *et al.* 2010; Yaganoglu vd 2011).

Sağlık sektöründe mekânsal-zamansal veri madenciliği ile sorgulama; bir sağlık kuruluşundan aldığımız veri içerisinde, bir hastanın, belirli bir zaman aralığında, hastane ziyareti sırasında hangi klinik ya da departmanlarda işlem gerçekleştirdiğini bulmaya çalışmaktır. Tıp verisinden oluşturulan anlamsal yörüngeler ve desenlerden mekânsal-zamansal veri madenciliği sorgulama yaparak, çeşitli çıkarımlar ve kurullar oluşturulabilir.

Şekil 1.6'daki taslakta kullanıcılar hem sorgu hem de veri madenciliği yapabilir. Mekânsal-zamansal veri madenciliği sorgulama dili, tıp verisini kullanarak kolaylıkla anlamsal eğriler üretebilir (Bogorny *et al.* 2008).



**Şekil 1.6.** Mekânsal-zamansal veri madenciliği sorgulama dili taslağı

### 1.1.7. Yörüngelerin durmaları ve hareketleri

Tıp bilgilerinden anlamsal yörüngeler oluşturmak için durmalar ve hareketler kullanılır. Tıp veritabanlarındaki durmaların ve hareketlerin yörüngesinin uygulanması, diğer alanlardan farklı değildir. Ancak tıp alanındaki verilerde belirli bir standardın olmayışı ve mevcut standartlar arasında tam bir uyumun olmaması nedeniyle, bu alanda bir veritabanı oluşturmak ve bu veritabanını işlemek zor bir işlemdir. Tıbbi verilerin kendine has bazı özellikleri bulunmaktadır. Temel veri yapıları, diğer birçok alanla karşılaştırıldığında, matematiksel olarak karakterize edilmeye pek uygun değildir. Bilginin düzenlenebilmesi için kümeleme veya dizi çözümlenmeleri gibi karşılaştırılabilir yapılar yoktur. Hekimler, görüntü, sinyal veya diğer klinik bilgilerle ilgili yorumlarını, standartlaştırılması çok güç olan serbest metinler olarak yazmaktadır. Örneğin aynı hastalığın açıklamasında farklı isimler kullanılmaktadır. Tıbbi kavramlar arasındaki ilişkileri açıklamak için de farklı dilbilgisi yapıları kullanılmaktadır (Dinçer 2006).

**Tanımlama 1:** Yörünge'nin mekân-zaman noktaları şu şekilde listelenir:  $\{P_0 = (x_0, y_0, t_0), P_1 = (x_1, y_1, t_1), \dots, P_N = (x_N, y_N, t_N)\}$ , burada  $x_i, y_i$  pozitif reel sayıdır,  $i$  değeri 0'dan  $N$ 'e kadar doğal sayılardır ve  $t_0 < t_1 < t_2 < \dots < t_N$  dir (Alveras *et al.* 2007).

Durmalar yörüngedeki hareketli nesnenin bir miktar süre kaldığı önemli yerleri temsil eder. Bir miktar süre aynı tip mekansal kesişmenin gözlemlendiği noktalar aday nokta çiftleri olarak belirlenir.

**Tanımlama 2:** Aday Durma: Aday Durma  $C$  bir  $(R_c, \Delta_c)$  kayıt,  $R_c$  kümesi aday durma geometrisini,  $\Delta_c$  ise minimum süreyi temsil eder. Aday durmalar özel uygulamalara bağlıdır. Bir uygulama, aday noktaları sonlu bir kümesi  $\{C_1 = (R_{c1}, \Delta_{c1}), C_2 = (R_{c2}, \Delta_{c2}), \dots, C_N = (R_{cN}, \Delta_{cN})\}$  olarak temsil edilir.

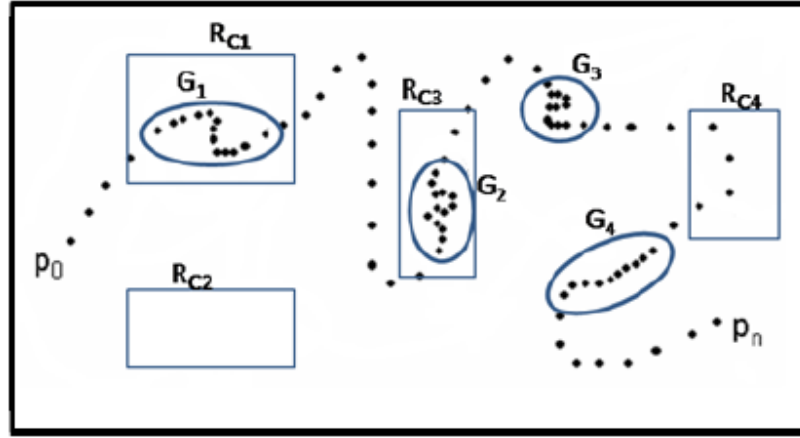
**Tanımlama 3:** Durma: T yörüngesindeki bir durma kayıt;  $(R_k, t_j, t_{j+n})$  alt eğri T;  $T\{(x_i, y_i, t_i) \mid (x_i, y_i) \text{ kesişim } R_k\} = \{(x_i, y_i, t_j), (x_{i+1}, y_{i+1}, t_{i+1}), \dots, (x_{i+n}, y_{i+n}, t_{i+n})\}$  ile temsil edilir, burada,  $R_k$ ;  $C_k$  aday noktalar geometrisi,  $|t_{i+n} - t_i| \geq \Delta_k$  dır (Alveras *et al.* 2007; Palma *et al.* 2008).

**Tanımlama 4:** Hareket: T yörüngesindeki bir hareket, zamansal olarak iki ardışık duraklar arasındaki T'nin en yüksek bitişik alt eğrisi olarak ifade edilir (Palma *et al.* 2008).

**Tanımlama 5:** Yörünge kümesi: Eps, bir veri nesnesinin komşularını belirlemek için gerekli olan yakınlık mesafesidir. MinTime zaman değişim değeridir. Eps ve MinTime parametresine göre T yörüngesindeki G kümesi; boş olmayan T alt eğrisindeki bitişik mekânsal-zamansal noktalar kümesinden oluşur.

**Tanımlama 6:** Bilinmeyen durma: T yörüngesi sırasında Eps, MinTime parametrelerine göre, T yörüngesindeki  $G_k$  kümesinde en az  $\Delta_c$  minimum süresinde hiçbir  $R_i$  kesişme bölgesinin olmamasıdır.  $C_i = (R_{ci}, \Delta_{ci})$  sadece aday noktadır.

Şekil 1.7'de görüldüğü gibi, dört durma kümesi  $G_1, G_2, G_3$  ve  $G_4$  mevcuttur. Aday durmalar  $R_{c1}, R_{c2}, R_{c3}$  ve  $R_{c4}$  ile temsil edilir. Örneğin  $G_1$  kümesi,  $\Delta_{c1}$  zamanından daha fazla  $R_{c1}$  aday durma ile kesiştiği için bu  $R_{c1}$  bölgesi eğride bir durma kabul edilir.  $G_3, G_4$  ise bilinmeyen durmalardır.



**Şekil 1.7.** Durmaların ve hareketlerin olduğu yörünge

Durmaların ve hareketlerin yörüngelerine örnek vermek gerekirse, Yağanoğlu vd (2011) çalışmasında Hastane verisinde durmaların ve hareketlerin olduğu hareket verisi için bir yaklaşım gösterilmiştir. Çizelge 1.3 ve Çizelge 1.4, durmaların ve hareketlerin olduğu yörünge verisini göstermektedir.

Çizelge 1.3, durma verilerinin örneğini gösterir. DurmaNumarası bu verilerden durmaların saklanması temsil eder. EğriNumarası yörüngeyi tanımlayıcısını, DurmaTipi ise durmaların ismini göstermektedir. Giriş zamanı ve çıkış zamanı ise bulunduğu mekanlardaki zaman unsurlarını temsil etmektedir. Tabloda görüldüğü gibi aynı anlamsal yörünge üzerindeki durmaları ve bu durmaların zamanlarını göstermektedir.

Çizelge 1.4, hareket verilerinin örneğini gösterir. HareketNumarası bu verilerden hareketlerin saklanması temsil eder. DurmaTipi1 ve DurmaTipi2 durmaların ismini göstermektedir. DurmaTipi1’de bulunan durmadan DurmaTipi2’ye doğru bir hareket vardır. Başlangıç Zamanı bu hareketin başlangıç zamanını, bitiş zamanı ise bu hareketin bitişini temsil eder. Tabloda aynı anlamsal verideki bir durmadan başka bir durmaya hareket olduğu görülmektedir.



**Çizelge 1.3.** Durmaların verileri

<b>DurmaNumarası</b>	<b>EğriNumarası</b>	<b>Mesafe</b>	<b>DurmaTipi</b>	<b>Giriş Zamanı</b>	<b>Çıkış Zamanı</b>
1	1	1	Hastane	07:10	22:05
2	1	2	Acil Servis	07:13	07:50
3	1	1	Radyoloji	08:40	09:25
4	1	3	Ameliyathane	10:00	11:00
...	...	...	...	...	...

**Çizelge 1.4.** Hareketlerin verileri

<b>HareketNumarası</b>	<b>EğriNumarası</b>	<b>Mesafe1</b>	<b>durmaTipi1</b>	<b>Mesafe2</b>	<b>durmaTipi2</b>	<b>Başlangıç Zamanı</b>	<b>Bitiş Zamanı</b>
1	1	1	Aile Hekimi	2	Hastane	09:00	09:30
2	1	2	Hastane	1	Radyoloji	10:15	11:25
3	1	1	Radyoloji	3	Ameliyathane	11:25	12:00
4	1	3	Ameliyathane	1	Yoğun bakım	12:03	13:00
...	...	...	...	...	...	...	...

Bu tezin ikinci bölümünde ilgili çalışma konum hakkında literatür özeti verilecektir. Tezin üçüncü bölümünde materyal ve yöntem ile birlikte durmaların ve hareketlerin olduğu yörünge verisi için yazdığımız program anlatılacaktır. Takip eden bölümlerde de çalışma ile ilgili deneysel sonuçlar, yorum ve değerlendirmelerden bahsedilecektir.

## 2. KAYNAK ÖZETLERİ

### 2.1. Kümeleme

Kümeleme, bir veritabanında oluşturulan, benzer nesnelerin gruplanması olarak ifade edilir (Han and Kamber 2006). Kümeleme, küme içine veri bölünmesi amacıyla geliştirilmiştir denetimsiz bir tekniktir. Her küme benzer nesnelere temel alınarak oluşturulur. Böylece aynı kümedeki nesnelere yüksek benzerlik, farklı kümelerdeki nesnelere önemli bir farklılık oluşur (Anil *et al.* 1999). Verilerin kümeleme analizine göre modellenmesinde matematik, istatistik, makine öğrenmesi, yapay zekâ ve yapay sinir ağları gibi birçok alandan yararlanır. Makine öğrenimi açısından, her bir küme gizli bir örüntüyü temsil eder ve uygulanan öğrenme ise bir denetimsiz öğrenmedir. Bu açıdan bakıldığında ise kümelemeyi, “gizli örüntülerin ortaya çıkarılması için uygulanan bir denetimsiz öğrenme yaklaşımı” olarak tanımlanabilir (Berkhin 2004; Silahtaroglu 2008).

Kümeleme analizi birçok disiplin tarafından kullanılmaktadır. İstatistik, bilgisayar ve matematik bilimi bunların en önde gelenleridir. Bilgisayar biliminde ses, karakter ve resim tanınması ve makine öğrenmesi kümeleme analizine dayanan çalışmalarda sağlanır. İstatistikte çok değişkenli istatistiksel tahmin ve örüntü tanıma konuları kümeleme analizinden yararlanır (Dempster *et al.* 1977; Silahtaroglu 2008).

Bunların dışında kümeleme analizi İnternet üzerinden Web sayfalarının aranması, DNA analizi, coğrafi bilişim sistemleri ve bunlarla ilgili alanlarda kullanılmaktadır (Xiaowei *et al.* 1998; Ben and Yakhini 1999).

Kümeleme analizi, sınıflandırmada olduğu gibi sahip olunan verileri gruplara ayırma işlemidir. Sınıflandırma işleminde, sınıflar önceden belirli iken kümelemede sınıflar önceden belli değildir. Verilerin hangi gruplara/kümelere, hatta kaç değişik gruba ayrılacağı eldeki verilerin birbirlerine olan benzerliğine göre belirlenir. Belirlenen her

bir gruba da küme ismi verilir. Kümeleme analizi biyoloji, tıp, antropoloji, pazarlama, ekonomi ve telekomünikasyon gibi birçok ve birbirinden çok farklı alanlarda kullanılmaktadır (Dunham 2003).

Kümelenme algoritmaları beş ana tipte kategorize edilebilir: parçalı, sıralı, grid-tabanlı, model-bazlı ve yoğunluk-bazlı kümelenme algoritmaları. Parçalı algoritmalarda, küme benzerliği bir kümede, çekim merkezindeki nesnelere ortalama değeri göz önüne alınarak ölçülen  $k$  ortalamalar ya da her küme merkezine yakın yerleştirilmiş kümenin nesnelere birisiyle yani  $k$  medoid ile temsil edilir.  $k$  bu algoritmalar için bir girdi parametresidir, bu parametre birçok uygulama için geçerlidir. Temsilciler kullanarak kümeleme (CURE), Dengeli tekrarlayan azalma ve hiyerarşi kullanarak kümeleme (BIRCH) gibi sıralı algoritmalar, sıralı bir ağaç gibi düzenlenen iç içe kümeler takımını oluşturur. Ağacın her bir düğümü, bir  $D$  veri tabanı kümesini temsil eder. STING, DalgaKümesi gibi grid-tabanlı algoritmalar, tüm kümelenme işlemlerinin yapıldığı çok-seviyeli grid yapısına dayanır. Model-tabanlı algoritmalarda (COB-WEB gibi) her bir küme için bir model varsayılır ve tez o modelin birbirine en iyi şekilde uyanını bulmaktır. Bu algoritmalar, verilerin çoğunlukla başlıca olasılık dağılımları bileşimiyle oluşturulduğu varsayımına dayanır (MacQueen 1967; Vinod 1969; Fisher 1987; Zhang *et al.* 1996; Wang *et al.* 1997; Guha *et al.* 1998; Han and Kamber 2001; Birant and Kut 2007).

Mekânsal-zamansal yörüngelerin benzerlik kavramları, ele alınan uygulama senaryosuna bağlı olarak değişebilir. Örneğin iki nesne verilen bir aralıkta aynı mekânsal-zamansal yörüngeyi takip etmişlerse yani aynı zamanlarda aynı yerlerde bulunmuşlarsa benzer olarak düşünülebilirler. Bununla birlikte gözlenen hareketlerin ortalama öge boyu (yani her yörünge için örneklenmiş mekânsal-zamansal noktaların sayısı), ölçülmüş noktalardaki belirsizlik ve genel olarak karşılaştırılmış iki nesnenin konumlarının mevcudiyetinin diğer değişimleri mekânsal-zamansal yörüngelerin birkaç benzerlik ölçümünün tanımını gerektirmişti. Bu ölçümlerin tanımı yalnızca kümelenme analiz işine uyarlanmamıştır ancak benzerlik araştırma problemi için hareketli nesne veritabanları alanında fazlasıyla kullanılmıştır ve zaman serisi analizi üzerine olan

çalışmadan ve en uzun ortak alt dizi (LCSS) modelinde etkilenmiştir. Çalışmalarında tanımlanan uzaklık fonksiyonları, yörünge bölgesinde açıkça tanımlanır ve zamanda hız ve ortak yerleşim gibi yörüngelerin birkaç mekânsal-zamansal özelliklerini dikkate alır (Agrawal *et al.* 1993; Berndt and Clifford 1996; Chan and Chee 1999; Vlachos *et al.* 2002, 2003; Theodoridis 2003; ; Chen *et al.* 2005; Nanni and Pedreschi 2006; Pelekis *et al.* 2007; Kisilevich *et al.* 2010).

## **2.2. Algoritmalar**

### **2.2.1. ID3**

ID3 algoritması diğer değişkenler içerisinde sınıflamada en ayırıcı özelliğe sahip değişkeni bulurken entropi kavramından yararlanır. Entropi kavramı, eldeki bilgilerin sayısallaştırılmasıdır (Dunham 2003). Entropi beklentisizliğin maksimumlaşmasıdır (Fiske 1998). Dunham entropinin bir veri kümesi içindeki belirsizlik, sapma ve rastgeleliği ölçmek için kullanıldığını söyler. Eldeki bütün veriler tek bir sınıfa ait olsaydı, örneğin herkes aynı futbol takımını tutsaydı, herhangi bir kişiye tuttuğu takımı sorduğumuzda alacağımız yanıt bizi şaşırtmayacaktı; bu durumda entropi sıfır(0) olacaktı. Entropi 0-1 arasında bir değer alır. Bütün olasılıklar eşit olduğunda entropi maksimum değerine ulaşacaktır (Silahtaroglu 2008).

### **2.2.2. C4.5**

C4.5 algoritması ID3 algoritmasına şu konular açısından üstünlük sağlamaktadır: Karar ağacı oluştururken kayıp veriler hesaba katılmaz. Yani, kazanım oranı hesaplanırken, sadece verileri eksik olmayan diğer kayıtlar kullanılır. C4.5 algoritması, kayıp verileri diğer veri ve değişkenler yardımıyla öngörerek kazanım oranının hesaplanmasında kullanır (Dunham 2003). Böylece daha duyarlı ve daha anlamlı kurallar çıkartabilen bir ağaç üretilebilir.

### 2.2.3. Sınıflandırma ve Regresyon Ağaçları (CART)

CART tekniği ID3 algoritmasında olduğu gibi en iyi dallara ayırma kriterini seçmek için entropiden yararlanır. En iyi ayırma kriterini belirlemek için ise ID3 ve C4.5'ten farklı bir formül kullanır (Yohannes and Webb 1999; Dunham 2003).

### 2.2.4. Araştırmada Denetimli Öğrenme (SLIQ)

SLIQ algoritması hem sayısal hem de kategorik verilerin sınıflandırılmasında kullanılabilir. Sayısal verilerin değerlendirilmesindeki maliyeti azaltmak için ağacın oluşturulması sırasında önceden-sıralama tekniği kullanılır. Sayısal verilerle işlem yapılırken en iyi dallara ayırma kriterini bulmak için verileri sıraya dizmek önemlidir. SLIQ algoritmasında kullanılan teknik ise verileri sıralama işlemini her düğümde yapmak yerine öğrenme verileri sadece bir kere, o da ağacın büyüme aşamasının başlangıcında yapılarak gerçekleştirilir. ID3 ve C4.5 gibi algoritmalar “önce derinlik” ilkesine göre çalışırken, SLIQ algoritması “önce genişlik” düşüncesiyle hareket ederek aynı anda birçok yaprağı oluşturur. Bu durumda mevcut ağacın yapraklara ayrılma işlemi verinin üzerinden bir kere geçilmesiyle tamamlanmış olur. Dallara ayrılma işleminde gini indeksi kullanılır (Manish 1996). SLIQ ayrıca kategorik verileri alt kümelerle ayırmada hızlı bir algoritma kullanır. Ağacın budanması işlemi ise en küçük uzunluk tanımlaması ilkesine dayanan bir strateji izler (Rissanen 1989; Silahtaroglu 2008).

### 2.2.5. AIS

AIS algoritması, geniş nesne kümeleri üretmek için geliştirilmiş bir algoritmadır. AIS algoritması veri tabanını birçok kez tarar ve her tarama esnasında tüm işlemleri okur. İlk tarama esnasında veritabanındaki nesnelere, teker teker sayarak hangilerinin geniş nesnelere olduğunu belirler. Bunlardan geniş olanlar aday nesne kümeleri olarak işaretlenirler. Bir işlem tarandıktan sonra, bir önceki taramada geniş oldukları belirlenen nesne kümeleriyle, o işlemin nesnelere arasındaki ortak nesne kümeleri belirlenir.

Belirlenen bu ortak nesne kümeleri işlemde mevcut olan diğer nesnelere birleştirilerek yeni aday küme oluşturulur. Herhangi bir nesne kümesi, bir işlemdeki nesnelere birleşip aday kümelere birini oluşturabilmesi için, birleşeceği nesnenin hem geniş olması hem de harf sırası açısından nesne kümesi içindeki tüm nesnelere sonra geliyor olması gerekir.

### **2.2.6. Geniş Uygulamaların Kümelenmesi (CLARA)**

CLARA, büyük veritabanlarının daha kısa süre içerisinde kümelenmesi amacıyla geliştirilmiş bir algoritmadır (Kaufman and Rousseeuw 1990). Kümeleme işleminin yapım süresi azaldığı halde gerçekte algoritmanın zaman karmaşıklığında herhangi bir iyileşme söz konusu değildir. CLARA algoritması bütün veritabanını tarayarak temsilci noktalar seçmek yerine, veritabanından rastgele bir örnek küme alarak, PAM algoritmasını bu örnek küme üzerine uygular. PAM algoritmasıyla kıyaslandığında CLARA algoritmasının daha geniş veritabanlarında güvenli bir şekilde çalışabildiği belirlenmiştir.

### **2.2.7. Rasgele Aramaya Dayalı Geniş Uygulamaları Kümeleme (CLARANS)**

CLARANS, yer veri tabanlarında incelenmesi için k medoid algoritmasının geliştirilmiş bir versiyonudur (Ng and Han 1994). CLARANS algoritması PAM ve CLARA algoritmalarının geliştirilmiş halidir. CLARANS algoritmasının iki parametresi vardır. Bunlar maksimum-komşu ve yerel miktardır. Maksimum-komşu parametresi incelenerek komşu sayısının üst limitini ifade ederken, yerel-miktarı ise elde edilecek yerel minimum nokta sayısının alt sınırını ifade etmektedir.

### **2.2.8. Medoid Etrafının Bölünmesi (PAM)**

PAM, ilk ortaya atılan k medoid algoritmasıdır. PAM, öncelikle k-ortalamlar algoritmasında olduğu gibi rasgele seçtiği k adet sayıyı küme merkezi olarak alır. Kümeye her yeni eleman katıldığında kümenin elemanlarını deneyerek kümenin

gelişmesine en fazla katkıda bulunabilecek noktayı tespit edince bulduğu noktayı yeni merkez, eski merkezi ise sıradan küme elamanı olacak şekilde yer değiştirme işlemi yapar (Dinçer 2006). PAM, 1990 yılında geliştirilmiş bir algoritmadır (Kaufman and Rousseeuw 1990). PAM algoritması k adet kümeyi bulmak için seçilen temsilcilerin etrafına ana kümedeki tüm elemanları toplayarak ve her defasında bu temsilcileri değiştirerek kümeleme işlemi tamamlar. PAM küçük veritabanlarında çok iyi sonuçlar vermesine rağmen hesaplanabilir karmaşıklığı yüksek olduğu için çok büyük veritabanlarında iyi performans göstermez. Bu bakımdan büyük veritabanları için CLARA ve CLARANS algoritmaları geliştirilmiştir.

### **2.2.9. CURE**

CURE algoritması, kümeleme işlemi yapılırken, oluşturulan kümelerin kalitesini en çok etkileyen faktör, ana veri topluluğu içinde diğer verilerden uzakta bulunan ve sayıları az olup aslında hiçbir kümeye ait olmaması gereken uç verilerdir. CURE algoritması bu uç verilerin oluşturulan kümelerin kalitesini etkilememesi düşüncesiyle 1998 yılında geliştirilmiş bir algoritmadır. Küresel bir geometrik şekil taşımayan veri gruplarının kümelenebilmesi oldukça elverişli bir algoritmadır.

### **2.2.10. CHAMELEON**

Chameleon algoritması ilk olarak 1999 yılında geliştirilmiş bir algoritmadır (Karypis 1999). Chameleon algoritması, iki küme arasındaki benzerliği dinamik bir model kullanarak belirler. Diğer algoritmalarından farklı olarak iki alt kümenin birbirine olan benzerliği ve yakınlığı bu iki kümeden her birinin kendi iç benzerlikleri ve yakınlıkları ile kıyaslanarak belirlenir ve bu karşılaştırma sonucunda bu iki alt küme birbirine yakınsa birleştirilir. Bu sayede daha kaliteli ve homojen kümeler yaratılmış olunur. Benzerlik / mesafe matrisinin oluşturulabildiği tüm veri türleri ve veri kümeleri için uygulanabilecek bir algoritmadır.



### **2.2.11. BIRCH**

BIRCH, çok büyük veritabanlarının kümelenmesi için geliştirilmiş bir algoritmadır; ayrıca gürültülü verilerin kontrol edilmesi için bu alanda öne sürülen ilk algoritmadır (Zhang 1996).

### **2.2.12. Kümeleme Yapısını Belirleyecek Noktaların Sıralanması (OPTICS)**

OPTICS algoritması DBSCAN algoritmasının geliştirilmiş halidir. Algoritmanın kendisi bir Eps değeri üretir ve bir kümeleme sıralaması geliştirir. Bu işlemlerin yapılması için algoritma iki ayrı parametreden yararlanır (Ankerst 1999). OPTICS algoritması veritabanındaki tüm noktalar için ulaşabilirlik ve iç-mesafe değerlerini tutarken, aynı zamanda veritabanını da bir sıralamaya tabi tutar. Eldeki bu bilgilerin yardımıyla, veritabanındaki sıralamayı oluştururken, yoğunluğa dayalı kümeleme işlemleri yapılır. OPTICS algoritması DBSCAN algoritmasıyla aynı zaman karmaşıklığına sahiptir (Han and Kamber 2003).

### **2.2.13. Yoğunluk Tabanlı Kümeleme (DENCLUE)**

DENCLUE algoritması temel olarak 3 adımdan oluşur: 1. adımda, her bir noktanın kendi çevresi kapsamındaki etki fonksiyonu hesaplanır; bu fonksiyon veritabanındaki tüm noktalara uygulanır. 2. adımda, bu fonksiyonların toplamı bulunur; hesaplanan bu etki fonksiyonları toplamı veritabanının genel yoğunluğunu vermiş olacaktır. 3. adımda kümeler matematiksel olarak yoğunluk-çekicileri yardımıyla tespit edilir (Hinneburg and Keim 1998).

### **2.2.14. Aramada Kümeleme (CLIQUE)**

CLIQUE algoritması yoğunluğa dayalı ve grid temelli yöntemleri birleştiren bir algoritmadır. CLIQUE algoritması genel olarak 3 adımda tamamlanır. Bunlar: (Agrawal 1998)

1. Kümeleri kapsayan alt uzayların tanımlanması
2. Kümelerin tanımlanması
3. Kümeler için minimum tanımın geliştirilmesi.

### **2.2.15. WEKA**

WEKA, Yeni Zelanda'daki Waikato Üniversitesi tarafından geliştirilmiş, makine öğrenmesi algoritmalarının bir arada bulunduran, işlevsel bir grafik ara yüze sahip, açık kaynak kodlu bir veri madenciliği programıdır (Witten 2011). WEKA çeşitli veri ön işleme, sınıflandırma, regresyon, kümeleme, ilişkilendirme kuralları ve görselleştirme araçları içerir. Algoritmalar veri kümesine doğrudan çağrılarak uygulanabilir (Pettersson 2008, Hall 2009; İşler ve Narin 2012).

### **2.2.16. WEKA-STPM**

WEKA-STPM, anlamsal yörünge ön işleme modülü olarak WEKA'nın geliştirilmesiyle oluşturulmuştur. Geliştirilen bu modül mekânsal-zamansal veri madenciliğini desteklemektedir. Sadece ön işleme değil aynı zamanda anlamsal yörünge bulma, veri madenciliği, zamana ve mekâna göre değişen hareket desenlerini keşfetmeyi de yapar (Bogorny *et al.* 2011).

### **2.2.17. WEKA-GDPM**

WEKA-GDPM, coğrafik bilgi sistemleri için entegre edilmiş veri madenciliği aracıdır. WEKA-GDPM birlikte işlenebilir bir modüldür ve Mekânsal veri madenciliği için otomatik coğrafik ön işlemeyi destekler. Bu modül WEKA'ya uygulanmıştır, WEKA'nın görsel ara yüzüne çeşitli butonlar ve araçlar eklenerek geliştirilmiştir (Bogorny *et al.* 2006).

### 2.2.18. Yoğunluk Tabanlı Mekânsal Kümeleme (DBSCAN)

DBSCAN, nesnelerin komşuları ile olan mesafelerini hesaplayarak belirli bir bölgede önceden belirlenmiş eşik değerden daha fazla nesne bulunan alanları gruplandırarak kümeleme işlemini gerçekleştirir (Rocha *et al.* 2010). DBSCAN isteğe bağlı şekilli kümeleri bulmak için tasarlanan yoğunluk tabanlı bir kümeleme algoritmasıdır. DBSCAN tek parametre gerektirir ve kullanıcıyı bunun uygun değerleri için destekler. Sınıf tanımlama amacı kümeleme algoritmaları için önemlidir. Ancak büyük zamansal veri tabanlarına uygulanması kümeleme algoritmaları için birçok gereksinime yol açar (Avcı 2006).

DBSCAN'da, bir noktaya bağlı yoğunluk, nokta civarında belirlenmiş bir yarıçapa sahip bölgede nokta sayısı sayılarak elde edilir. Belirlenen bir eşğin üzerindeki yoğunluktaki noktalar küme olarak oluşturulur. Mevcut kümeleme algoritmaları arasında, DBSCAN algoritması lineer, dışbükey, oval gibi keyfi şekli olan kümeleri ortaya çıkarma yeteneğine sahiptir. Öte yandan bazı kümeleme algoritmalarının tersine küme sayısının önceden belirlenmesini gerektirmez. DBSCAN'ın çok büyük veri tabanlarını işleme yeteneği ispatlanmıştır (Ester *et al.* 1996, 1998; Aoying and Shuigeng 2000; Birant and Kut 2007).

DBSCAN algoritması birçok çalışmada kullanılmıştır. Örneğin diğer popüler yoğunluk-bazlı algoritma olan OPTICS, DBSCAN algoritmasının kavramlarına dayanır ve iç içe kümeleri ve kümelerin yapısını belirler (Ankerst *et al.* 1999). Artımlı DBSCAN algoritması, DBSCAN kümeleme algoritmasına da dayanır ve veri tabanına yeni bir nesnenin eklenmesinden ve veri tabanından mevcut nesnenin silinmesinden sonra kümelemenin artımlı güncelleştirmeleri için kullanılmıştır (Ester *et al.* 1998). Biçimsel küme kavramına bağlı olarak, artımlı algoritma artımlı olmayan DBSCAN kümeleme ile aynı sonucu verir. SDBDC (Ölçeklenebilir Yoğunluk-Bazlı Dağılımlı Kümeleme) metodu dağılımlı nesnelere kümelemek için hem yerel bölgelerde hem de global bölgede DBSCAN algoritmasını kullanır. Bu metotta, DBSCAN algoritması önce her yerel bölgede yürütülür. Daha sonra bu yerel kümeleme sonuçlarına dayanarak, küme

temsilleri belirlenir. Daha sonra da bu yerel kümelenme sonuçlarına dayanarak küme temsilleri belirlenir. Şu halde bu yerel temsillere dayanarak, standart DBSCAN algoritması dağılımlı kümelenmeyi oluşturmak üzere global bölgede yürütülür. Bu çalışma her yerel temsil farklı Eps-değerlerinin kullanılmasını önerir (Hinneburg and Keim 1998). Wen *et al.* (2002), sorgulamalı kümelenme araçlarının ana algoritmaları olarak DBSCAN'ı ve Artımlı DBSCAN'ı uyguladılar. Sıkça sorulan soruları ve bir arama motorunda en popüler konuları kümelemek için DBSCAN'ı kullandılar. Spieth *et al.* (2005) düzenleyici şebekelerin öngöründe bulunma çözümlerini belirlemek için DBSCAN'ı uyguladılar. SNN yoğunluk-bazlı kümelenme algoritması da DBSCAN'a dayanır ve yeryüzünün farklı noktalarında atmosferik basıncın zaman serisi verilerinden ibaret olan yüksek-boyutlu verilerine uygulanabilir (Birant and Kut 2007).

### **2.2.19. Mekansal- Zamansal DBSCAN (ST-DBSCAN)**

ST-DBSCAN, temeli DBSCAN'a dayanan yoğunluk tabanlı kümeleme algoritmasıdır. Hem zaman hem de mekân boyutu ile uğraşan yoğunluk tabanlı kümeleme algoritmasıdır. Bu çalışmada, DBSCAN'ın (i) çekirdek nesnelere, (ii) gürültü nesnelere ve (iii) bitişik kümelerin belirlenmesine yönelik üç marjinal genişletişini önerilmiştir. Mevcut yoğunluk-bazlı kümelenme algoritmalarının aksine, algoritma nesnelere mekânsal olmayan, mekânsal ve zamansal değerlerine göre kümeleri ortaya çıkarma yeteneğindedir. Bu çalışmada, aynı zamanda çok sayıda zamansal- mekânsal verinin kümelenmesi ve depolanması için tasarlanan Mekânsal-Zamansal veri depolama sistemi sunulmuştur DBSCAN algoritmasını üç önemli yönde geliştirilerek bu yeni algoritma oluşturulmuştur. İlk olarak, mevcut yoğunluk-bazlı kümelenme algoritmalarına benzemeksizin, algoritma Mekânsal olmayan, Mekânsal ve zamansal özelliklerine göre yer-zaman verilerini kümeleyebilir. İkinci olarak, farklı yoğunluktaki kümeler mevcut olduğunda DBSCAN farklı yoğunluklarda bazı gürültü noktalarını belirleyemez. ST-DBSCAN, her küme için bir yoğunluk faktörü vererek bu problemi çözer. Üçüncü olarak, komşu nesnelere Mekânsal olmayan değerlerinin farklılıkları az ve kümeler birbirlerine bitişik ise, bir kümedeki sınır nesnelere değerleri karşı taraftaki sınır nesnelere değerlerinden epey farklı olabilir. Algoritma bu problemi

yeni gelen deęerle kümenin ortalama deęerini karşılaştırarak çözer (Birant and Kut 2007).

### **2.2.20. Mekansal-Zamansal Veri Madencilięi için Etkili bir Hareket Deseni Keşfi Algoritması (STMPE)**

STMPE, mekânsal-zamansal veri madencilięi için etkili bir hareket deseni keşfi algoritmasıdır. Bu algoritma, yüksek miktardaki mekânsal-zamansal veriden hareket desenlerinin etkin bir şekilde çıkarılmasını sağlar. STMPE algoritması mekânsal-zamansal veriyi genelleştirir ve bellek kullanımını düşürür. Çünkü kısa zamanlı hareket desenlerini üretir, saklar ve veri tabanı taraması minimize edilebilir. STMPE algoritması benzer algoritmalara göre avantaj sağlar (Kim *et al.* 2006). STMPE algoritması hareketli nesnelerin geçmişini analiz eder ve Mekânsal bilgileri gösteren genelleştirilmiş hareket desenlerini çıkarır. STMPE algoritması ayrıca mekânsal-zamansal veriyi genelleştirir ve yüksek miktarda mekânsal-zamansal verinin saklanmasından dolayı oluşan bellek kullanımını azaltır. Kısa zamanlı hareket desenlerinin üretilmesi ve saklanması ile veritabanı tarama frekansı düşürülebilir.

STPMine1 algoritması hareket desenlerini mekânsal-zamansal hareketli nesnelerin zamanına göre çıkarır (Mamoulis *et al.* 2004). STPMine1 algoritması hareketli nesnelerin geçmiş bilgisi üzerine her zaman bölgesi ünitesi için önceden tanımlanmış mekansal bölgelerin sık tekrar eden desenlerini bulur ve sonuçları girdi olarak kullanıp daha uzun desenleri bulur. STPMine1 algoritması ayrıca AprioriTID algoritmasını değiştirerek sık hareket desenlerini çıkarır. Çünkü AprioriTID algoritması tekrarlayan şekilde bir veritabanını desenin uzunluğu kadar tarar ve bu performans süresi ve belleğin kullanımı zaman alanının sayısı arttığında daha büyük olur (Agrawal and Srikant 1994).

STPMine2 algoritması da mekan ve zamana baęlı olarak hareketli nesnelerin hareket desenini çıkarabilir (Mamoulis *et al.* 2004). Hareket desenlerini veritabanını sadece 2 kez tarayarak bulabilir ki bu yönüyle STPMine1 algoritmasından farklıdır. Bu şekilde

performans zamanını da düşürmüş olur. Hareket desenlerini yüksek minimum destek ve gizli kısmi desenler ile çıkarır (Kim *et al.* 2006).

### 2.2.21. DJ-Cluster

DJ-Cluster, yörüngelerde kişi adlarını bulmak için geliştirilen yoğunluk-bazlı algoritmadır. Yalnızca yörüngelerin yer özelliklerini ele alır. Bu DB-SMoT'a en yakın metod olmasına rağmen, önemli yerleri kullanıcı tarafından açık olarak belirtilir ve bu yerler DB-SMoT metoduna benzer yörüngelerde ortaya çıktığında DJ-Cluster gözden geçirilir (Zhou *et al.* 2007; Rocha *et al.* 2010).

## 2.3. Diğer Kaynak Özetleri

Çevresel uygulamalar için yapılan çok yönlü Mekânsal-Zamansal veri madenciliği çalışmasında, yazarlar çok-değişkenli küme analizini çevreyle ilgili yöreselleştirme; ortam görüntüleme dizaynı; uydu analizi, havadan taşıma ve karada yerleşik uzaktan algılama ve iklim model-model ve model-döngülü model karşılaştırması dahil çeşitli ortam bilim bölgelerine uygulamıştır. Kümelenme metodolojisi, oldukça iyi ölçeklenebilir, paralel yüksek performanslı hesaplama (HPC) uygulamasında yerine getirilen bir  $k$ -araç istatistik kümelenme algoritmasını kullanır. Yeterliği ve HPC platformlarının kullanılması nedeniyle, kümelenme kodu çok uzun ya da yüksek frekanslı, çözünürlüklü zaman serisi ölçümleri ya da model çıktısı gibi yüksek boyutlu çok büyük veri kümelerini analiz etmek ve karşılaştıracak bir veri madenciliği aracı olarak uygulanabilir (Hoffman *et al.* 2008).

Mekânsal-zamansal veri için hareketli kümelerin keşfedilmesi çalışmasında, yörünge nesnelere veritabanlarında hareketli kümeler incelenmiştir. Hareketli bir küme, uzun bir zaman aralığında birbirlerine yakın hareket eden nesne kümesi ile tanımlanır. Reel-yaşam örnekleri, göç eden hayvanlar grubu, bir şehirde hareket eden araba konvoyu vb. şeklindedir. Yörüngelerin kümelenmesi ve hareketli nesnelere incelenmesi ile karşılaştırılan bu problemin farkı, konumu ve içeriği zamanla değişebilirken hareketli

küme biriminin değişmez kalmasıdır. Örneğin, bir hayvan grubu göç ederken, bazı hayvan grupları ayrılabilir ya da yeni hayvanlar gruba katılabilir. Hareketli nesnelere için biçimsel bir tanım sağlanmıştır ve otomatik keşif için üç algoritma açıklanmıştır: (i) tanıma dayalı doğrudan metot, (ii) gereksiz kontrollerden kaçınan daha etkili metot ve (iii) MPEG-2 video şifrelemesinden tezler alarak hızın kesinliğini değiştiren yaklaşık bir algoritma (Kalnis *et al.* 2005).

Mekânsal-zamansal veri madenciliği için yoğunluk bazlı kümeleme tekniği çalışmasında, gürültülü bir ortamda hareketli nesne kümelerinin belirlenmesi ve takibi problemini dile getirilmiştir. Bu problemi dile getirmede öncelikli teşebbüsler ya diğer kümelerle kesişen kümelerin takibinin mümkün olmaması ya da yüksek hesaplama zamanı ve depolama gereksinimlerinin olmasıdır. Kümeleme algoritmasının kesinliğini artırmak için hareketli nesnelerin konum tarihçesinden yararlanan Dinamik Yoğunluk Bazlı Kümeleme algoritması önerilmiştir.

Li *et al.* (2004) hareketli mikro-küme fikrine dayanan hareketli kümeyi belirleyecek bir metot önerir. Hareketli mikro-kümeleme, BIRCH kümeleme algoritmasında kullanılan mikro-kümeleme fikrinin genişletilmesidir. İlk mikro kümeleme, hareketli nesne kümesinde geleneksel statik kümeleme algoritması kullanılarak meydana getirilir. Her nesnenin mevcut konumu ve hızı, bu ilk kümeleme boyunca kullanılır. Kümeleme nesnelerin hızlarına dayanarak ayrılacağı ve birleşeceğini öngörürler. Nesnelerin hızları değiştiğinde, öngörüler güncellenmiştir. Bu tekniğin birkaç sınırı vardır. Mikro kümeleme tanımı algoritmayı yalnızca küresel kümelerin bulunmasına sınırlar. Algoritma, kümeleri diğer kümelerle çakışma periyotları boyunca ve kümeleme ait olmayan nesnelere de ayıramaz. Ayrılan ve birleşen öngörülerini güncelleştirme, hareketli nesnelerin hızlarının sıkça değişmesi durumunda algoritmanın işlem zamanı tayin eder (Zhang *et al.* 1996).

Nehme and Rundensteiner (2006) SCUBA denilen yer-zaman sorgulamaları çözme tekniğini önerir. SCUBA ile kullanılan kümeleme tekniği hareketli mikro kümelemeye dayanır ve aynı sınırlandırma kümeleri paylaşır. Chen *et al.* (2007) BIRCH

kümelenme stratejisini yollar gibi bir şebekeye sınırlanmış hareketli nesnelere üzerine çalışmaya uyarlamıştır. Jensen *et al.* (2007) geçmiş harekete dayanan hareketli nesnelere gelecek konumlarını öngörerek ve bu nedenle nesnelere hızlarındaki değişikliklere algoritma duyarlılığını indirgeyerek Hareketli Mikro-Küme algoritmasını geliştirir. Olayları ayıran kümeyi daha kesin belirleyecek yeni bir teknik, de temsil edilir. Bu teknik yalnızca küresel teknikleri bulabilir ve diğer kümeler ve gürültü ile çakışan periyot ile kümeleri takip edebilmektedir.

Bernkert *et al.* (2007) hareketli nesne kümelerinde sürüleri belirlemek için birkaç teknik verir. Bir sürü, ardışık zaman adımlarının belirli bir sayısı için birbirlerinin bir yarıçapı içinde bir nesne kümesi şeklinde tanımlanır. Bu sürü tanımı, algoritmayı yalnız küresel sürülere sınırlar.

Philips Adası'nda turistlerin mekânsal-zamansal hareket örnekleri incelenerek çeşitli çıkarımlar yapılmıştır. Bu çıkarımlar mekânsal-zamansal veri madenciliği yardımıyla kolaylaşmıştır. Bu günlerde turistlerin mekân-zaman hareket davranışını anlamak, turizm pazarlama başarısı açısından gittikçe önemli bir etken olmaktadır. Bu çalışma, turist profilleri arasındaki turist yer-zaman hareket örnekleri ile çapraz örneklerini ve zaman-mekân hareket örneklerini belirleyecek genel amaçlı bir veri madenciliğinden yararlanır. Büyük zaman hareket dizileri ve zamansal hareket dizileri keşfedilmiş ve karşılaştırılmıştır. Örneğin, turistler gününbirlik gezileri için yalnız akşam bir yeri ziyaret etmişler, bu Penguen gezisidir. Kümelenme metodu, her yer için turistlerin pazar dilimini belirlemek amacıyla kullanılır. Farklı yerler için turistleri gruplayacak çeşitli anahtar simgeler vardır. Ziyaretçilerin tipi (uluslararası ya da yerli), Phillip Adası'ndaki turistleri kümeleyecek en önemli niteliklerdir. Her yer ve hareket örneği için turistlerin farklılıkları da sınıflandırma algoritmaları kullanılarak karşılaştırılmıştır. Bu çalışmanın hedefleri; turistlerin zaman hareketinin anlamlı örneklerini belirlemek ve mevcut veri madenciliği tekniklerini kullanarak turist profilleri ile mekânsal-zamansal hareket örneklerini keşfetmektir (Xia *et al.* 2005).



Hwang *et al.* (2005) çalışmasında, farklı bir yaklaşım önerilmiştir burada yörüngeler muhtemelen eksik aralıklarla parçalı kısımlar halinde temsil edilmiştir. Önerilen metot, yakın bir zaman aralığını yani tüm yörüngelerin birbirine yakın çiftler olduğu en yüksek zaman aralığını belirlemeye çalışır. Yörüngelerin benzerliği, yörüngelerin yakın olduğu zaman miktarına dayalıdır ve problemin incelenmesi verilen bir eşikte yakın olan tüm yörünge gruplarını bulmaktır.

Yağanoğlu vd (2011) çalışmasında, sağlık sektöründe RFID kullanılarak Mekânsal-zamansal veri madenciliği veri tiplerine bir yaklaşım önerilmiştir. Bu çalışmada, RFID'nin sağlık sektöründe kullanılmasının doktor ve hemşirelerin hastalara daha hızlı ve sağlıklı bir şekilde müdahale etmesinde getireceği yararlar gösterilmeye amaçlanmıştır. Farklı yöntem ve teknik aracılığı ile veri kaynakları analiz edilerek anlamsal bilgi keşfedilmeye çalışılmıştır. Hastanelerde bu teknoloji ile birlikte Mekânsal-zamansal veri madenciliği kullanarak çeşitli konularda kolaylık ve hız katılabilir. Hasta, doktor ve hemşirelerin eş zamanlı takibi sağlanabilir. Hastaların bekleme süresi azaltılarak memnuniyetleri arttırılabilir. RFID ile elde edilen veriyi Mekânsal-zamansal veri madenciliği uygulayarak hastalık yayılımları çok hızlı bir şekilde görülebilir. Hastalığın yayılımı izlenebilir, hastalığın nerede başladığı ve yoğunlaştığı görülebilir.

Bozkurt vd (2011) çalışmasında, sağlık hizmetlerinde Mekânsal-zamansal veri madenciliği üzerine algoritma yaklaşımı yapılmıştır. Sağlık hizmetlerinde elde edilen hareketli nesne verilerine algoritmanın uygulanmasından sonra, son kullanıcılara sunuşu grafiksel ara yüzler, raporlamalar ile elde ettiğimiz sonuçları sunabilecek ve anlamsal yörüngeler yorumlanabilecektir.

Mikro kümelenmenin genişletişine dayalı benzer bir yaklaşım Li *et al.* (2004) çalışmasında önerilmiştir. Bu durumda verilen bir dikdörtgende farklı yörüngelerin parçaları benzer zaman aralıklarında meydana geldiğinde birlikte gruplanır. Metodun amacı, eşik dikdörtgeninde maksimal grup boyutunu ve zaman boyutunu belirlemektir

Lee *et al.* (2007) çalışmasında, yörüngeler açık zaman bilgisi olmaksızın noktalar dizisi olarak temsil edilir ve yarı-lineer parça kümesine ayrıştırılır. Tüm parçalar kümelenme metoduna dayanan bir yoğunluk vasıtasıyla gruplanır ve her kümenin temsili yörüngesi belirlenir.

Kang *et al.* (2004) çalışmasında, yazarlar tek bir yörüngede önemli yerlerin belirlenmesi için artımlı bir kümelenme önerdiler. Algoritma için birçok faktör tanımlanmıştır: keyfi sayıda küme, mümkün olduğu kadar pek önemli olmayan yerlerin çıkarılması ve mobil cihazlarda çalışmaya izin verecek kadar geniş hesaplanabilir değildir. Algoritma önemli yerler bulmaya dayanır burada birçok konum ölçümü birlikte kümelenir. İki parametre, kümenin oluşturulmasını-kümedeki konumlar arasındaki mesafeyi ve harcanan zamanı kontrol etmiştir.

Coğrafik verilerin zaman modellemesi son on yılda özellikle önemli dikkat çekmiştir. İlk model, hala çoğu ticari coğrafik bilgi sistem paketlerinde etkili olan yaklaşımını ele alır. Sonraki modeller, mekânsal-zamansal küpünü, mekânsal-zamansal kompoziti ve değişiklikleriyle birlikte temel durum üzerinedir. Bu dizideki çoğu yakın zamanlı çalışma, olayları, faaliyetleri, prosesleri ya da coğrafik özelliklerin nesnelere değişimi vasıtasıyla mekânsal-zamansal örnekleri modellemeyi destekler (Hagerstrand 1970; Langran 1992; Claramunt and Theriault 1995; Peuquet and Duan 1995; Usery 1996; Yuan 1997; Chen and Jiang 1998; Wachowicz 1999; Hornsby and Egenhofer 2000; Wang and Cheng 2001).

Mekânsal-zamansal verilerinin incelenmesine ait metotların geliştirilmesi ve yer verileri alt yapısının geliştirilmesi, etkin ve etkili mekânsal-zamansal verileri incelenişiyle birlikte gerçekleşmeli. Her mekânsal-zamansal temsili yaklaşımı ve karşılık gelen veri yapıları veri inceleme algoritmalarının bazı zorluklarını dayatabilir. Ayrıca farklı yaklaşımlar durgundan tam zamana sıralanan farklı zaman seviyelerine yerleştirilebilir. Uygun veri inceleme metotları verildiğinde, mekânsal-zamansal veri takımından elde edilen bilgi mekânsal-zamansal veri tiplerine, temsiline ve veri takımında kullanılan veri yapısına bağlıdır.

Mekânsal-zamansal veri madenciliği hareketli nesnelere hareket desenlerini çıkaran bir çalışma alanıdır. Fakat hareket desenlerinin mekânsal-zamansal veri madenciliği ile çıkarılmasında zaman gözardı edilmektedir ve hareket desenleri mekâna göre çıkarılmaktadır. Bu nedenle zamanın değişimine göre hareket desenlerinin çıkarılması ile ilgili çalışma yapılması gereklidir (Han *et al.* 1999; Yang *et al.* 2000; Tsoukatos and Gunopoulos 2001; Peng and Chen 2003).

Hareket desenlerinin çıkarılması için genel desen bulma metotlarının geliştirilmesi gerekir. Varolan STPMine1 ve STPMine2 sık görülen hareket desenleri ve kısmi hareket desenlerinin çıkarılması için etkin bir yol sağlayabilir (Mamoulis *et al.* 2004). Buna rağmen minimum destek düşük olduğunda birçok aday hareket deseni üretilir bu nedenle zaman ve mekan performansı keskin bir şekilde artarak zayıf bir nokta oluşturur (Kim *et al.* 2006).

Patlama olaylarının yapay ağlarda senkronize olarak Mekânsal-Zamansal sınıflandırması çalışmasında sapmalar tespit edilmiştir (Barkan and Horn 2005).

Olay tabanlı 3 boyutlu görüntüleme gerçek zamanlı hareket analizi için mekânsal-zamansal kümeleme yöntemi kullanılmıştır. Bu çalışma, dinamik 3 boyutlu görüntü sistemi ile görünüm aktivitelerinde oluşturulan eş zamansız olayların kümelenmesine dair bir metot önerir. Dinamik görüntü sensör çiftinden oluşan dinamik stereo görüntü sistemi ile sağlanan hareketli nesnelere esas belirlenişi, reel zamanda olay-tabanlı stereo görüntüyü ve hareketli nesnelere bir 3D temsilini sağlar. Kümelenme metodu, reel zaman belirlenim için sensörlerle elde edilen verilerin mekânsal-zamansal temsilinden ve hareketli nesnelere arasındaki ayrımından yararlanır. Metot, görünüm dinamiği ile oluşturulan eş zamansız durumların kümelenmesi için yoğunluk ve uzaklık metriği kullanılır. Görünüş sensör alanı boyunca hareketli insanların başından geçen vakaların kümelenmesi üzerine değerlendirilmiştir (Schraml and Belbachir 2010).

Hastalık salgınının erken belirlenmesi için Bir Z Skoru Tabanlı Çok seviyeli Mekânsal Kümelenmesi (ZMSC) algoritması önerilmiştir. Algoritmanın değerlendirilmesi için yarı-sentetik verileri kullanılarak, ZMSC'yi Dalgacık Anomali Detektörü, bir zaman algoritması ve iki yer kümelenme algoritması ile karşılaştırılmıştır. Bu karşılaştırılan algoritmalar ise Kulldorff'un ve Bayes'in yer tarama istatistiği algoritmalarıdır (Kulldorff 1997; Zhang *et al.* 2003; Neill *et al.* 2005; Que *et al.* 2008)

Yatay olarak bölümlenmiş veriler için gizlilik koruyucu mekânsal-zamansal kümeleme metodu önerilmiştir. Metot, üçüncü bir partiye yapılabilen güvenli çok-partili yörünge karşılaştırmaları serisi vasıtasıyla benzersizlik matrisini oluşturmaya dayanır. Yörünge karşılaştırma protokolü çoğu yörünge karşılaştırma fonksiyonuna uyar ve metot karmaşıklık analizi merkezileşmiş yaklaşıma nazaran benzersizlik matrisini oluşturduğunda protokolün ekstra işletim yükü vermediğini gösterir. Metot, herhangi bir hiyerarşik kümelenme algoritmasına girdi olabilen gizlilik koruyucu biçimindeki nesne yörüngelerinin benzersizlik matrisini oluşturmaya dayanır. Esas katkılar, yörünge uzaklıklarının güvenli çok-partili hesabına giriştir ve yer-zaman verilerinin gizlilik koruyan kümelenmesine uygulamasıdır. Aynı zamanda önerilen metodun karmaşıklık ve gizlilik analizini sağlar. Bu protokol yörüngelerin gizlilik koruyan karşılaştırılması ve yatay bölümlenmiş yer-zaman verilerinin kümelenmesine uygulanması için önerilmiştir. Protokolümüzün esas avantajı, çoğu yörünge karşılaştırma fonksiyonlarına ve hiyerarşik kümelenme gibi farklı kümelenme metotlarına uygulanmasıdır (Inan ve Saygin 2006).

Agrawal and Srikant (2000) gizliliği korurken, sınıflandırma modellerini oluşturmada biçimleyici çalışmalarıyla gizlilik koruyan veri incelemesi üzerine araştırma başlatmıştır Saygin *et al.* (2001) verileri açıklamadan önce hassas işbirliği kurallarını gizlemek için metotlar önermektedir. Gizlilik koruyan veri inceleme metotları iki başlıkta sınıflandırılabilir: veri temizleme ve güvenli çok-partili hesaplama. Veri temizleme yaklaşımları arttırılmış gizliliğin kesinliğini gözden çıkarır oysa güvenli çok-partili hesaplama yaklaşımları yüksek hesaplama ve hesaplama maliyetlerinin giderlerinde hem kesinliği hem de gizliliği elde etmeye çalışır. Merugu and Ghosh (2003) , girdi

verilerinden veri inceleme modellerini oluşturmak için metotlar önermektedir. Bu modeller özel bilgi olarak ele alınmaz. Tüm kümelenme şeması, dikey ya da dağılmış veri kaynaklarından gelen bu modeller birleştirilerek oluşturulur. Oliveira and Zaiane (2004), verilerin boyutunu azaltarak gizliliğini korumak için metotlar önerir. Metotlar yatay bölünmüş verilere uygulanabilir değildir ve öte yandan kesinlik kaybına yol açarlar. Vaidya and Clifton (2003), dikey bölünmüş verilerin  $k$ -araç kümelenmesi için güvenli çok-partili bir hesaplama protokolü önerir. Jha *et al.* (2005), küme araçlarının güvenli çok-partili hesaplamayla yatay bölünmüş verileri üzerinde  $k$ -araç protokolünü veren bir gizlilik koruması önerir. Inan *et al.* (2006) sayısal, kategorik ve alfa-nümerik verileri ele alabilen yatay bölünmüş veriler üzerine başka bir gizlilik koruyucu kümelenme algoritması önerir.

## **2.4. Durmaların ve hareketlerin yörüngeleri algoritmaları**

### **2.4.1. Durmaların ve Hareketlerin Yörüngesi (SMoT)**

SMoT algoritmaları aday noktalardan durmaların bulunması fikrine dayanmaktadır (Alvares *et al.* 2007). Durmalar hareketli nesnenin bir miktar zaman durmasıdır. Hareketler durmalar olmadığı bütün eylemlerdir (Spaccapietra *et al.* 2007). Örneğin trafik ışıkları, kavşaklar, hastane klinikleri, kayıt kabul bölümünde bir zaman dilimi bekleme durma olarak ele alınır. Yörüngelere anlamsal bilgi eklemek için farklı uygulama alanlarında yörünge veri analizini kolaylaştırmak için, veri ön işleme modülü önerilmiştir. Uygulama ile ilgili yörünge parçalarını temsil etmek bu model için önemlidir. Önemli yörünge parçalarını hesaplamak için önerilen bu algoritmada, anlamsal yörünge analiz sorgulama karmaşıklığını azaltır (Alvares *et al.* 2007).

### **2.4.2. Kümeleme-tabanlı Durmaların ve Hareketlerin Yörüngesi (CB-SMoT)**

CB-SMoT, yörünge kümeleme tabanlı durma ve hareketleri, hız değişimine dayanan yörünge kümeleme algoritmasıdır. Yörüngeler ile çalışan hız dayanan Mekânsal-

Zamansal kümeleme algoritmasıdır. Düşük hız verisini önemli bir yer olarak baz almaktadır. Bir yörüngesel hareket boyunca düşük hız değerlerinin yaşandığı yerleri ilginç mekânlar olarak belirleme yöntemine dayanmaktadır. Girdi olarak yörünge verisi, hız değişimi ve en küçük zaman (minTime) parametreleri alınır, zaman varyasyonu ile düşük hızların görüldüğü yerler işaretlenir. Sonuç olarak durma ve hareketlenme noktaları belirlenir (Palma *et al.* 2008). Bu metot, hızın önemli olduğu uygulamalarda trafik yönetimi gibi çalışmalarda tercih edilir.

### **2.4.3. Kesişme-tabanlı Durmaların ve Hareketlerin Yörüngesi (IB-SMoT)**

IB-SMoT, yörünge tabanlı durma ve hareketleri; özellikle uygulama açısından ilginç olan coğrafik nesne tipleriyle ilgili yörünge örnek noktalarının kesişimine dayalı durakları oluşturur. Bu kesişim, durak olarak ele alınacak alt yörünge için minimum zaman eşğine uymalıdır. Bu metot uygulamalarda ilginçtir burada hareketli nesnelerin örneğin bir parkta, şehir planlamada v.b. turizm, eğlence aktiviteleri olarak minimum zaman miktarında kaldığı yerleri bulmak önemlidir. Bu metot önce yörünge örnek noktalarını hesaplar ve yörünge hızının minimal bir zaman miktarı için verilen eşikten daha düşük olduğu yerlerde küme oluşturur. İkinci adımda, metot, uygulamada önemli olan kullanıcı tanımlı ilgili coğrafik yer takımının kümeleriyle eşleşir. Bu metot, hızın trafik yönetimi olarak esas rol oynadığı uygulamalarda ilginçtir. Verilen bir zaman aralığında nesnenin geçtiği önemli yerler hangileridir şeklindeki basit bir sorgu için yörüngeler ve coğrafik özellik tipleri arasında bir çok Mekânsal birleşmenin olduğu kompleks bir sorgu gerekir. Anlamsal yörüngeleri veri kümesinde ise durmaların tablosu üzerindeki tek bir sorgu böyle bir sorguyu cevaplandırabilir. Bir durma bir hareket başladığı zaman, o alt yörünge tablosu üzerindeki durmasıyla kesişen son noktasında başlar. Benzer olarak bir hareket bir durma ile biterse, o durmayı kesen alt yörüngelerin ilk noktasında sonlanır (Rocha *et al.* 2010).

## 2.5. Durmaların ve Hareketlerin Yörüngeleri için Sorgulama Dili

Yörünge verini yönetmek ve sorgulamak için özel bir sorgulama dili geliştirilmemiştir. Geleneksel sorgulama SQL diline, mekansal özellik entegre edilerek coğrafik bilgi sistemlerinde kullanılmıştır. Coğrafik madencilik sorgulama dili (GMQL) ve mekânsal veri madenciliği nesnesi sorgulama dili (SDMOQL) gibi mekânsal veriler için birkaç veri madenciliği sorgu dilleri ileri sürülmüştür (Han *et al.* 1997; Malerba *et al.* 2004). GMQL mekânsal veri madenciliğini destekler, SDMOQL ise mekânsal veri madenciliği nesne sorgulama dilidir. Mekânsal-zamansal veri madenciliği sorgulama dili STDMQL, mekânsal SQL dilinden türetilmiştir (Egenhofer 1994). STDMQL, anlamsal yörüngeleri üretmek için coğrafi bilgi ile yörüngeleri kolayca entegre işlemi yapabilmek için önişleme operatörleri sağlamaktadır (Bogorny *et al.* 2008).

Yörünge sorguları, yörüngelerin entegrasyonu ve anlamsal coğrafik bilgi ile hesaplamalı ve formülasyon karmaşıklığından basitleştirilmiş ve optimize edilmiştir (Alvares *et al.* 2007). İlgili veriye erişmek için bazı sorgulara ihtiyaç duyarız. Bu basitleştirmeyi anlamak için aşağıdaki sorgu örneğinde olduğu gibi bir sağlık kurumunda, belirli bir zaman aralığında hastanın hareket yörüngesi sırasında işlem ve kayıtlarını öğrenmek istiyorsak, Çizelge 2.1 deki gibi sorgu oluştururuz:

**Çizelge 2.1.** Sorgu örneği

```
SELECT 'Clinics as place
      FROM trajectory t, Clinic C
WHERE t.tid='1234'
      AND
      intersects (t.movingpoint.geometry,C.geometry)
      AND
      t.time between '08:00' AND '12:00'
```

STDMQL, daha yüksek soyutlama seviyesinde yörünge elde edimi için veri önişleme fonksiyonları ilgili coğrafik bilgi ile yörünge örneklerini entegre eder. Bu fonksiyonlar, veritabanında ikili ilişki olarak saklanan durmalar ve hareketleri tespit eder. Anlamsal

yörüngeleri elde edebilmek için kullanılan fonksiyonun aldığı giriş parametreleri şunlardır:

Metot, durmaları ve hareketleri hesaplar. Aday durmalar ve nokta-çizgiler ile temsil edilen aday durmalar çevresi tampon alanından oluşur. Bu fonksiyonundan dönen değer durmaların ve hareketlerin ilişkileridir. Yörüngeden durma ve hareketlerin çıkarımı için genel sorgu yapısı aşağıdaki gibidir:

```
SELECT generateSM (method, candidateStops, buffer)
FROM trajectory
```

Parametre metot, durmaları ve hareketleri nasıl hesaplanacağını belirtir. Geliştirilen metotlara örnek SMoT, CB-SMoT, DB-SMoT, IB-SMoT verilebilir. CandidateStops minimum durma süresine göre ile ilgili mekânsal özellik türlerinin kümesidir. Buffer ise geometrik doğruluk sorunları aşmak için noktalar ya da çizgiler ile temsil özellikler çevresinde bölgenin büyüklüğüdür (Alvares *et al.* 2007; Palma *et al.* 2008). Çizelge 2.2’ de durmaların ve hareketlerin algoritmalarından DB-SMoT kullanılarak sorgulama yapılmıştır.

#### Çizelge 2.2. DB-SMoT algoritması kullanarak sorgulama

```
SELECT generateSM (DB-SMOT,[Kan Merkezi,60,Röntgen,15,Biyokimya
Lab,30], 5, 20)
FROM trajectory t, district d
WHERE d.name='Yakutiye Hastanesi' and
intersects(t.movingpoint.geometry, d.geometry)
```

Örneğin, herhangi bir bölgedeki belirli bir zaman diliminde gerçekleşen durma ve hareketlerin tespit edilmesi işlemi Çizelge 3.3’deki gibi olmaktadır. Erzurum ili, Yakutiye ilçesinde bulunan Erzurum Yakutiye Araştırma Hastanesi’nde “20.08.2011” ve “20.08.2012” tarihleri arasında yaş aralığı 30-50 arası ve cinsiyeti kadın olan hastaların durma ve hareketlerin getirilmesi. Bu işlem daha da özele inerek, hastane içerisinde herhangi bir lokasyona da uygulayabiliriz. Örneğin, Kan Merkezi,



Biyokimya, Röntgen, Laboratuvar gibi yerlerde belirli bir zaman aralığı için de benzer yörünge sorguları alınabilmektedir.

**Çizelge 2.3.** DB-SMoT algoritması için geliştirilmiş sorgulama

```

select generateSM (DB-SMoT(Erzurum Yakutiye Araştırma Hastanesi, 30),5,20)
IL.il_Adi as IL, ILCE.ilce_Adi as ILCE,HS.Adi as Hastane,
    P.Adi as Poliklinik,BRY.Ad as Ad,BRY.Soyad as Soyad,
    cast((year(GETDATE()))-year(Bry.DogumTar)) as int) as Yas,
    Sbt.Deger as Cinsiyet,
    H.Adi as Hastalik, isnull(HK.Sikayet,0) as Sikayet,
isnull(HK.Tarih,M.Tarih) as GelisTarihi, M.Tarih as CikisTarihi,
    M.Id as MuayeneId, M.Birey_Id as BireyId
from Muayene M
    inner join Poliklinik P on P.Id = M.Poliklinik_Id
    inner join Hastalik H on H.Id = M.Hastalik_Id
    inner join Hastane HS on HS.Id = P.Hastane_Id
    inner join Birey BRY on BRY.Id = M.Birey_Id
    inner join il IL on IL.il_Id=P.il_Id
    inner join ilce ILCE on ILCE.ilce_Id = P.ilce_Id
    left join HastaKayit HK on HK.Id = M.HastaKayit_Id
    inner join Sabitler Sbt on Sbt.Sabit= BRY.Cinsiyet And Sbt.Parameter
    ='Cinsiyet'
Where BRY.Cinsiyet='Kadın' And cast((year(GETDATE()))-
year(BRY.DogumTar)) as int) between 30 And 50 P.il,ilce( ERZURUM-
YAKUTİYE, ) order by il,ilce,Hastane,Poliklinik
And M.Tarih between '20.08.2011' and '20.08.2012'

```

### 3. MATERYAL ve YÖNTEM

#### 3.1. Yön-tabanlı Durmaların ve Hareketlerin Yörüngesi (DB-SMoT)

Hareketli nesnelere oluşturulan veriler örnek noktalar olarak  $\langle d, x, y, t \rangle$  formunda normal olarak mevcuttur burada  $tid$  nesne tanımlayıcısıdır ve  $x, y; t$  zamanında hareketli nesnenin coğrafik konumudur. Yörünge örnek noktalarının çok az semantiği vardır ya da hiç yoktur ki bu uygulama açısından analizlerini çok kompleksleştirir.

Yakın zamanda, Spaccapietra *et al.* (2008) durmalar ve hareketler denilen yörüngeleri ele alacak yeni bir model sunmuştur. Bu model, yörüngeleri işlemek üzere özellikle semantik bilgiyi ekleyerek uygulamalar için önemli bir yer edinmiştir. Genellikle, hareketler duraklar arasında hareketli nesnenin hareketleri iken, duraklar uygulama açısından bir yörünge için en önemli kısımlarıdır. Bu model, kuş göçü gibi çok sayıda uygulamayı destekler burada duraklar kuşların beslendiği ya da dinlendiği ülkeler olabilir; trafik yönetimi, burada duraklar trafik ışıkları, dolambaçlı, hız kontrolcülerini, trafik sıkışıklıkları vb. olabilir (Rocha *et al.* 2010).

DB-SMoT, yörünge için yön-tabanlı durma ve hareketlerini bulan bir algoritmadır. Yön değişimlerine göre durma ve hareket noktaları belirlenip kümelemeler tespit edilir. Girdi olarak yörünge örnekleri, minimal yön varyasyonu, minimum zaman, maksimum tolerans belirlenir. Her yörünge örneği için, yön varyasyonu minimal yön varyasyonundan yüksek kümelerin bulunması gerçekleştirilir. Sonuç olarak anlamsal yörüngeler elde edilir. Bu metod, yön varyasyonun önemli olduğu uygulamalarda tercih edilir. Anlamsal yörünge veri analizine dair mevcut çalışmalar, yörüngelerin önemli coğrafi bilginin uygulamasıyla kesişimine ve ilginç yerler bulmak üzere hızın kullanılmasına odaklanmıştır. Bu çalışmada, yörüngelerde ilginç yerler bulmak için çarpıcı bir yaklaşım gösterilmiştir, esas bakış açısıyla yön değişimini ele alınmıştır. Önerilen yaklaşım, teknelerin balık tutma aktivitelerini geliştirdikleri gerçek yerleri otomatik olarak bulmak amacıyla okyanus balıkçı tekneleriyle ilgili reel yörünge

verileriyle geçerli kılınır. Sonuçlar, yön değişiminin esas rolü oynadığı uygulamalar için metodun çok uygun olduğunu göstermiştir (Hazin *et al.* 2006; Rocha *et al.* 2010)

Bu bölümde Spaccapietra'nın genel tanımına göre yörünge, durmaların ve hareketlerin tanımlarını sunacağız. Bu tanım kullanıcının ilgilendiği belirli uygulamaya bağlıdır.

**Tanım 1:** Bir örnek yörünge  $\langle p_0, p_1, \dots, p_N \rangle$  uzay-zaman noktaları listesidir burada  $i = 0, \dots, N$  ve  $t_0 < t_1 < \dots < t_N$  için  $p_i = \langle x_i, y_i, t_i \rangle$  ve  $x_i, y_i, t_i \in \mathbb{R}$ 'dir.

Yörüngeyle toplanan tam bilgi uygulama bağlamıyla ilgilidir ancak yörünge esas olarak duraklar ve hareketler denilen daha küçük parçalara ayrılır ve ilave bilgi yörüngelerin daha anlamlı bir kavrayışını sağlar.

Spaccapietra *et al.* (2008), haklarında bazı özellikler tanımlamışsa da, hangi bilginin bir durak/ hareket şeklinde toplanabileceğini açık olarak belirtmemiştir:

**Durma:** Durma, bir yörünge ilgili, zaman aralığıdır. Öyle ki;

1. Kullanıcı bir durmayı temsil etmek üzere yörünge bu kısmını açıkça tanımlamıştır;
2. Zaman genişletisi boş olmayan bir zaman aralığıdır;
3. Bu yörünge uygulama amacına göre seyahat eden nesne hareket etmez;
4. Aynı yörüngedeki tüm duraklar zaman olarak ayrıktır yani iki durağın zaman genişletişleri daima ayrıktır.

**Hareket:** Hareket, yörünge bir parçasıdır. Öyle ki;

1. Parça, iki ardışık durma ile temsil edilen iki ekstremite ile ayrılır.  $t_{begin}$  ilk durma,  $t_{end}$  son durma ya da yörünge durmalar yok ise  $[t_{begin}, t_{end}]$  ile ifade edilir.

2. Zamansal ölçüde  $[t_{begin}, t_{end}]$  boş olmayan bir zaman aralığıdır.
3. Yörünge'nin  $[t_{begin}, t_{end}]$  zaman dilimindeki mekânsal aralığı, yörünge fonksiyonu ile tanımlanan mekânsal-zamansal hat (nokta değildir) dır.

### Tanım 2: Yönün Değişimi

$\langle p_{i-1}, p_i, p_{i+1} \rangle$  bir alt yörünge olsun.  $p_i$  'de yön değişimi  $DC \phi_i$  ile gösterilen  $\overline{p_{i-1}, p_i}$  ile  $\overline{p_i, p_{i+1}}$  yönleri arasında açıdır.

### Tanım 3: Aday-küme-noktası

$\langle p_{i-1}, p_i, p_{i+1} \rangle$  bir alt yörünge olsun.  $DC \phi_i \geq \min DC$  ise  $\min DC$ 'e göre  $p_i$  noktası aday-küme-noktasıdır. Bir  $p_i$  noktasının aday-küme-noktası olarak ele alınabilmesi için  $\min DC$  eşiği bu noktadaki minimum yön değişimini açıklar.

### Tanım 4: Bağlantılı-aday-nokta

$\langle p_i, p_{i+1}, p_{i+2}, \dots, p_{i+n+1} \rangle$  bir alt yörünge olsun.  $p_i$  ve  $p_{i+n+1}$  aday-nokta noktası ve  $n \leq maksTol$  ise,  $p_i$  noktası  $p_{i+n+1}$  ile bağlantılı-aday-noktasıdır.  $maksTol$  maksimal tolerans eşiği, bir kümede sonuç olarak bulunabilen  $\min DC$  eşiğinden daha az yön değişimiyle maksimum sayıda yörünge noktasını belirtir.

### Tanım 5: Yörünge kümesi

$\min DC$ ,  $maksTol$  ve  $\min Zaman$ 'e göre bir  $T$  yörünge'sinin  $C = \langle p_1, p_2, \dots, p_n \rangle$  kümesi bitişik zaman-uzay noktaları takımıyla oluşturulan şöyle boştan farklı alt yörünge'sidir:

- 1)  $\forall p, q \in T : p \in C$  ve  $\min DC$  ve  $\max Tol$ 'e göre  $p$   $q$  ile bağlantılı-aday-noktası ise,  $q \in C$ 'dir.
- 2)  $\forall p, q \in C : \min DC$  ve  $\max Tol$ 'e göre  $p$   $q$  ile bağlantılı-aday-noktasıdır.
- 3)  $t_n - t_1 \geq \min Zaman$  burada  $p_i = \langle x_i, y_i, t_i \rangle$ 'dir.

DB-SMoT metodu, tanımlandığı gibi yön değişimine dayanan tek yörüngelerde kümeler bulur. **EK 1**'de, girdi olarak örnek noktalar şeklinde temsil edilen yörünge kümesi, minimal yön değişimi eşiği, bir küme oluşturacak minimal zaman miktarı ve yön değişimini hesaplayacak maksimal toleranslı algoritmanın yalancı kodunu gösterir. Yörünge'nin (10.satır) nokta sayısını kontrol ederek başlar ve her iki nokta arasında yön-değişimini hesaplar (12-14.satır). Yön-değişimini hesaplarken, *findClusters* (16.satır) metodu kümelerini bulur. Kümelerde (duraklarda) olmayan tüm alt yörüngeler için, hareketler oluşur (17-25.satır) (Rocha *et al.* 2010).

**EK 2**'de ayrıntılandırılan *findClusters* metodu, yörünge'nin (5.satır) her iki nokta arasında yön-değişimini kontrol ederek başlar. Değişim minimal yön değişimi eşiği aşarken (MinDirChange), noktalar küme'ye ilave edilir (6-7 satır). Bir nokta yönünü değiştirmedikçe (8. satır), yönü değiştirmeyen nokta gürültü olup olmadığını ya da yön değişiminin sonlanıp sonlanmadığını (9-17 satır) doğrulamak amacıyla maksimal yönü kontrol edilir. Kümenin yeterli yön değişimi olan noktaları ilave ettikten sonra, minimal zaman devamının sınırlanışını aşmış olmadığını kontrol edilir (19-23 satır).

## 3.2. Geliştirilen Algoritmalar

### 3.2.1. Bulanık C-Ortalamlar DB-SMoT (FCM-DBSMoT)

DB-SMoT algoritmasında yörünge kümelemesi adımında bulanık kümeleme algoritmalarından Bulanık C-Ortalamlar (FCM) kullanarak, FCM-DBSMoT algoritmasını geliştirdik. Geliştirdiğimiz algoritmayı aynı veri için geliştirdiğimiz diğer

algoritmalarla karşılaştırarak durmaların ve hareketlerin olduğu algoritmalarından en iyisini bulmaya çalıştık.

FCM algoritması, bulanık bölünmeli kümeleme tekniklerinden en iyi bilinen ve yaygın kullanılan yöntemdir. FCM metodu nesnelere iki veya daha fazla kümeye ait olabilmesine izin verir. Bulanık mantık prensibi gereği her veri, kümelerin her birine  $[0,1]$  arasında değişen birer üyelik değeri ile aittir. Bir verinin tüm sınıflara olan üyelik değerleri toplamı "1" olmalıdır. Nesne hangi küme merkezine yakın ise o kümeye ait olma üyeliği diğer kümelere ait olma üyeliğinden daha büyük olacaktır (Yıldız *et al.* 2010).

FCM algoritmasının en önemli özelliği olan üyelik matrisinin kümeleme üzerinde olumlu etkileri vardır. Bu matris belirsiz durumların tanımlanmasını kolaylaştırır (Azem 2003). Ayrıca üyelik dereceleri düşük olduğundan sıra dışı verilerin etkisi azdır. Esnek bir yapıya sahiptir. Örtüşen kümeleri bulma kabiliyeti diğer bölünmeli algoritmalara göre daha fazladır.

Yukarıda bahsedilen avantajların yanında bulanık c-ortalama algoritmasının bazı dezavantajları da vardır. Üyelik fonksiyonu işlemsel karmaşıklığı arttırdığı için zaman açısından maliyetli bir bölünmeli kümeleme algoritmasıdır.

FCM algoritması da amaç fonksiyonu temelli bir metottur. Algoritma, en küçük kareler yönteminin genellemesi olan aşağıdaki amaç fonksiyonunu öteleyerek minimize etmek için çalışır (Jain *et al.* 1999; Moertini 2002).

$$Jm = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, 1 \leq m < \infty \quad (3.1)$$

$u$  üyelik matrisi rastgele atanarak algoritma başlatılır. İkinci adımda ise merkez vektörleri hesaplanır (Davidson 2002; Moertini 2002; Azem 2003).

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m x_i}{\sum_{i=1}^N u_{ij}^m} \quad (3.2)$$

Hesaplanan küme merkezlerine göre  $u$  matrisi aşağıdaki formül kullanılarak yeniden hesaplanır. Eski  $u$  matrisi ile yeni  $u$  matrisi karşılaştırılır ve fark  $\varepsilon$ 'dan küçük olana kadar işlemler devam eder (Davidson 2002; Moertini 2002; Azem 2003).

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_i - c_i\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad (3.3)$$

Kümeleme işlemi sonucunda bulanık değerler içeren  $u$  üyelik matrisi kümelemenin sonucunu yansıtır. İstenirse, berraklaştırma yapılarak bu değerler yuvarlanıp 0 ve 1'lere dönüştürülebilir (Yıldız *et al.* 2010).

Bulanık c ortalamalar metodunun pseudo-kodu **EK 3**'de gösterildiği gibidir. DB-SMoT algoritmasında yörünge kümeleme yaptığı adımda, bulanık c ortalamalar kullanarak uygulamamızı geliştirdik.

### 3.2.2. Bulanık K-Ortalamlar DB-SMoT (FKM-DBSMoT)

DB-SMoT algoritmasında yörünge kümelemesi adımımda bulanık kümeleme algoritmalarından Bulanık k ortalamalar (FKM) kullanarak, FKM-DBSMoT algoritmasını geliştirdik. Yörünge kümesi tanımlamalarında FKM kullanarak, uygulamalar için yeni bir algoritma geliştirdik.

Kümeleme birçok uygulamada sık sık kullanılmaktadır, Bu uygulamalara; vektör niceleme, bilgi keşfi, örüntü tanıma, arıza tespiti, benzerlik ve web madenciliği örnek olarak verilebilir. Kümeleme fonksiyonları arasında amaç fonksiyonunu en aza indiren FKM algoritması yaygın olarak kullanılmaktadır ve çalışılmaktadır (Liu and George 2003).

Yörünge kümeleme her yoğun yolun aralarındaki benzerliğe göre yörünge gruplarından oluşmaktadır. Yörünge kümeleme ayrı koleksiyonlarda, kümeler halinde bir yörünge bölümlenir, böylece her yörünge benzerlik tanımına göre benzer yörüngeleri içerir. Temel yörünge özellikleri ile ilgili yörüngeler arasındaki benzerliği ölçmek için çeşitli yaklaşımlar önerilmiştir (Pelekis *et al.* 2009).

Bulanık kümeleme klasik kümeleme tekniklerinin bir uzantısıdır ve örüntü tanımlama alanlarında ve bulanık modellemede birçok sorunu çözmek için kullanılmıştır (Forgy 1965). Çeşitli bulanık kümeleme yöntemleri öne sürülmüştür ve bunların birçoğu mesafe kriterlerine dayanır. FKM kümeleme desenleri, özellikle nerelerde kümeleme çakışması olduğunu anlayabilmek için yaygın olarak kullanılır. Bu arazi kullanımı/arazi örtüsü sınıflandırmada, gen kümeleme tanımlamasında ve su kimyasının sınıflandırılmasında uygulanmıştır (Liu *et al.* 2003).

DB-SMoT algoritmasına, bulanık yörünge kümeleme uygulamayı öneriyoruz. Biz DB-SMoT' daki yörünge kümeleme yerine Bulanık k ortalamalar kümeleme metodunu uyguladık. Gerçek uygulamalarda genellikle kümeler arasında keskin bir sınır yoktur bu yüzden veriler için bulanık kümeleme kullanmak daha uygundur.

FKM kümeleme algoritmasının parçaları; veri noktaları için  $k$ , kümeler  $S_i (i = 1, 2, \dots, k)$  ve kümeler  $S_i$  ile ilişkili küme merkezi kümelemesi  $C_1$ . Küme temsilcisi ve bir veri noktası arasındaki ilişki bulanıktır. Yani, bir üyelik  $u_{i,j} \in [0,1]$  temsil etmek için veri noktası üyelik derecesi  $X_i$  ve kümeleme merkezi  $C_j$  kullanılır. Veri noktalarının



kümesi  $S = X_i$  gibi gösterilir. FKM algoritması aşağıdaki fonksiyonun  $J_{FKM}$ 'nin minimize edilmesine dayanır (Chang *et al.* 2011).

$$J_{FKM} = \sum_{j=1}^k \sum_{i=1}^N \mu_{i,j}^m d_{ij} \quad (3.4)$$

Kümeleme ile ilgili temsili  $C_j$  ve üyelik  $u_{i,j}$ ;  $N$  yörünge veri noktası sayısı;  $m$  bulanıklaştırma parametresi;  $k$  kümelerin sayısı ve  $d_{ij}$  veri noktası  $X_i$  arasındaki Öklid mesafesinin karesi ve kümeleme temsili  $C_j$ 'dir. Aşağıdaki kısıtlamada  $u_{i,j}$  tahmin edilmektedir:

$$\sum_{j=1}^k \mu_{i,j} = 1, \text{ for } i = 1 \text{ to } N. \quad (3.5)$$

FKM'nin ana süreci, geliştirilmiş veri noktalarının bölümlenmesini verilen bir grubu vektörlerinin temsilini haritalamaktır. FKM, başlangıç küme merkezlerinin kurulması ile başlar ve bu durma kriteri istenilen duruma gelene kadar, bu eşleştirme süreci tekrarlanır. Aynı küme temsiline sahip iki küme olmaması gerekir. Bu durumda, iki küme merkezi bir küme merkezinin yinelemeli süreci önlemek için tedirgin olmasına denk gelir. Eğer  $d_{ij} < \eta$ ,  $l \neq j$  için  $u_{i,j} = 1$  ve  $u_{i,l} = 0$  olur, burada  $\eta$  çok küçük bir pozitif sayıdır. Fuzzy K-means kümeleme algoritması aşağıdaki gibi gösterilir (Chang *et al.* 2011):

- (1) Başlangıç küme merkezlerinin girişi  $SC_0 = \{C_j(0)\}$  ve  $\varepsilon$  nin değeri  $p = 1$  ayarlanır.
- (2)  $SC_p$  Kümeleme merkezini ayarla,  $d_{ij}$ 'yi  $i = 1$  to  $N$  ve  $j = 1$  to  $k$  için hesapla.  $u_{i,j}$  üyelikleri aşağıdaki denklemi kullanarak güncelle:

$$u_{ij} = \left( (d_{ij})^{1/m-1} \sum_{l=1}^k \left( \frac{1}{d_{il}} \right)^{1/m-1} \right)^{-1} \quad (3.6)$$

Eğer  $d_{ij} < \eta$ ,  $u_{i,j} = 1$  ayarla, burada  $\eta$  çok küçük bir pozitif sayıdır

(3) yeni bir kümeleme merkezi  $SC_{p+1}$  elde etmek için, her bir küme için 4 numaralı eşitlik kullanılarak merkezleri hesaplanır.

$$C_j(p) = \frac{\sum_{i=1}^N u_{ij}^m X_i}{\sum_{i=1}^N u_{ij}^m} \quad (3.7)$$

(4) Eğer  $j = 1$  to  $k$  için  $\|C_j(p) - C_j(p-1)\| < \epsilon$  ise,  $\epsilon > 0$  olduğu yerde çok küçük bir pozitif sayı vardır. Aksi halde,  $p + 1 \rightarrow p$  ayarla ve 2.ci adıma git.

FKM-DBSMoT algoritması yörüngeler için tekrarlayıcı bir süreçtir. Bu algoritmanın adımları aşağıdaki gibidir:

**Adım 1:** Yörünge (T) ayarla ve kümeleme için minimum zamanı belirle (minTime).

**Adım 2:** Her bir alt yörünge için (t) yörüngeleri (T) ayarla.

**Adım 3:** Kümeleri hesapla.

**Adım 4:** Kümeleme sayısını, bulanık parametresi (a sabiti  $>1$ ) ve durma koşulunu ayarla.

**Adım 5:** Bulanık bölümlenme matrisini sıfırla.

**Adım 6:** Döngü sayacı  $k=0$  ayarla.

**Adım 7:** Küme merkezlerini ve nesnel değer J'yi hesapla.

**Adım 8:** Her bir nokta için ve her bir küme için matris üyelik değerlerini hesapla.

**Adım 9:** Ardışık tekrarlamalarda Eğer J değeri durma koşulundan daha küçükse, dur. Aksi takdirde  $k=k+1$  ayarla ve 7. adıma git.

**Adım 10:** Netleştirme ve segmentasyon.

**Adım 11:** Kümelerdeki durmaları ve hareketleri bul. MinTime için yörünge kalma zamanını hesapla.

**Adım 12:** Adım 2'ye git.

Bulanık k ortalamalar metodunun pseudo-kodu **EK 4**'de gösterildiği gibidir. DB-SMoT algoritmasında yörünge kümeleme yaptığı adımda, FKM kullanarak uygulamamızı geliştirdik.

### 3.2.3. Bulanık Medoid DB-SMoT (FM-DBSMoT)

DB-SMoT algoritmasında yörünge kümelemesi adımında bulanık kümeleme algoritmalarından Bulanık medoid (FM) kullanarak, FM-DBSMoT algoritmasını geliştirdik. Yörünge kümesi tanımlamalarında FM kullanarak, uygulamalar için yeni bir algoritma geliştirdik.

Veri kümelemede bulanık mantık fikri Bezdek (1984) tarafından ileri sürüldü. Bulanık kümelemede, her veri noktası kısmen ayrı ayrı kümelere atanmıştır. Bu kısmen atama her küme için her nesnenin üyelik ilişkisi derecesini gösterir ve 0 ile 1 arasında bir ondalık sayı ile temsil edilmektedir (Sabzi *et al.* 2011).

K medoids algoritmasının temeli, verinin çeşitli yapısal özelliklerini temsil eden k tane temsilci nesneyi bulma esasına dayanır. Temsilci nesne medoid olarak adlandırılır ve kümenin merkezine en yakın noktadır. Bir grup nesneyi k tane kümeye bölerken asıl amaç, birbirine çok benzeyen nesnelere bir arada bulunduğu ve farklı kümelerdeki nesnelere birbirinden benzersiz olduğu kümeleri bulmaktır. En yaygın kullanılan k-medoids algoritması, Kaufman and Rousseeuw (1990) tarafından geliştirilmiştir. Temsilci nesne, diğer nesnelere olan ortalama uzaklığı minimum yapan kümenin en merkezi nesnesidir. Bu nedenle, bu bölünme metodu her bir nesne ve onun referans noktası arasındaki benzersizliklerin toplamını küçültme mantığı esas alınarak uygulanır. Kümeleme literatüründe temsilci nesnelere çoğunlukla merkez tipler denilmektedir. PAM (Medoid Etrafının Bölünmesi) algoritmasında temsilci nesnelere medoid olarak adlandırılmaktadır. Amacın k tane nesneyi bulmak olmasından dolayı, k-medoids metodu olarak adlandırılmaktadır. k adet temsilci nesne tespit edildikten sonra her bir nesne en yakın olduğu temsilciye atanarak k tane küme oluşturulur. Sonraki adımlarda her bir temsilci nesne temsilci olmayan nesne ile değiştirilerek kümelemenin kalitesi

yükseltinceye kadar ötelenir. Bu kalite nesne ile ait olduğu kümenin temsilci nesnesi arasındaki ortalama benzersizlik maliyet fonksiyonu kullanılarak değerlendirilir.

Son on yılda, literatürde yörüngeler üzerinde çeşitli yaklaşımlar ve algoritmaları çalışmaları öne sürülmüştür. Bu yaklaşımların çoğu zaman serisi analizinden esinlenmiş olup, cisimin yörüngedeki hareket şekli ile yörüngelerin benzerliklerinin ölçümü yapılmıştır. En iyi kümeleme yöntemini bulma problemi zor bir işittir. Burada, kümeleme bulanık mantık mantığını esas alır, örneğin FCM gibi, her bir veri ögesini belli bir üyelik ölçütü ile farklı kümeye aitliği sağlanarak belirsizliği düşünlür (Zadeh 1965; Bezdek 1984). Yörünge kümeleme, her bir yoğun yoldaki yörüngelerin benzerliğine göre yörüngelerin gruplandırma işlemidir. Yörüngeleri ayrı koleksiyonlar halinde kümeleme ile yörüngeler kümelere bölümlenir. Bu sayede her küme benzerlik tanımına göre benzer yörüngeleri içerir. Diğer yaklaşımlar da, basit yörünge özellikleri ile ilgilenecek yörüngeler arasındaki benzerlikleri ölçmeye çalışır (Pelekis *et al.* 2009).

Bulanık kümeleme klasik kümeleme tekniğinin bir uzantısı olup, örüntü tanıma ve bulanık model kimlik alanları gibi birçok sorunları çözmek için kullanılır olmuştur (Forgy 1965). Bulanık kümelemede çoğu mesafe kriterine dayalı çeşitli yöntemler önerilmiştir.

Gerçek uygulamalarda, kümeler arasında keskin çizilerin olmadığı durumlarda veri için bulanık kümeleme daima daha iyi sonuç vermektedir. Biz DB-SMoT algoritmasına bulanık yörünge kümelemeyi uygulamayı öneriyoruz. DB-SMoT'un kullandığı yörünge kümelemesi yerine FM kümeleme üzerine çalışmaktayız.

FKM kümeleme algoritma kavramı medoids oluşturmak için uygulanmıştır. FM algoritmasının değerlendirme fonksiyonu, minimize etmek için eşitlik 3.8 kullanılır (Chu 2004).

$$J_m(O, X) = \sum_{j=1}^T \sum_{i=1}^k \mu_{ij}^m r(x_j, o_i) \quad (3.8)$$

Buradaki  $r(x_j, o_i)$  , nesne  $x_j$  ve medoid  $o_i$  arasındaki benzeşmezliği gösterir.

FM algoritması aşağıdaki gibi tanımlanır:

**Adım 1:** Medoid'in başlangıç kümesinin seçilmesi.

$O = \{o_1, o_2, \dots, o_k\}$  nesne kümesinden

$X = \{x_1, x_2, \dots, x_T\}$  rasgele, k küme sayısı ve T nesne sayısı olmak üzere

**Adım 2:** Üyeliğin hesaplanması

$$\mu_{ij} = \frac{\left(\frac{1}{r(x_j, o_i)}\right)^{\frac{1}{m-1}}}{\sum_{q=1}^k \left(\frac{1}{r(x_j, o_q)}\right)^{\frac{1}{m-1}}}, \quad i=1, \dots, k \quad j=1, \dots, T. \quad (3.9)$$

burada  $r(x_j, o_i)$  ;  $x_j$  ve medoid  $o_i$ , arasındaki benzeşmezliği gösterir, bulanıklaştırma  $m \in [1, \infty]$

**Adım 3:** Yeni medoid 'in aşağıdaki gibi güncellenmesi yapılır.

$$p = \arg \min_{1 \leq v \leq T} \sum_{j=1}^T u_{ij}^m r(x_v, x_j),$$

$$o_i = x_p, i = 1, \dots, k$$

**Adım 4:** Medoid değişmez veya maksimum iterasyon sayısına ulaşıldığında program sonlandırılır.

Bulanık medoid metodunun pseudo-kodu **EK 5**'de gösterildiği gibidir. DB-SMoT algoritmasında yörünge kümeleme yaptığı adımda, FM kullanarak uygulamamızı geliştirdik.

### 3.2.4. Bulanık Adaftif DB-SMoT (FA-DBSMoT)

DB-SMoT algoritmasında yörünge kümelemesi adımı bulanık kümeleme algoritmalarından Bulanık Adaftif (FA) kullanarak, FA-DBSMoT algoritmasını

geliřtirdik. Yörünge kümesi tanımlamalarında FA kullanarak, uygulamalar için yeni bir algoritma geliřtirdik.

Biz DB-SMoT algoritmasına bulanık yörünge kümelemeyi uygulamayı öneriyoruz. DB-SMoT'un kullandığı yörünge kümelemesi yerine FA kümeleme üzerine çalışmaktayız. Gerçek uygulamalarda, kümeler arasında keskin çizilerin olmadığı durumlarda veri için bulanık kümeleme daima daha iyi sonuç vermektedir.

FA kümeleme, veri noktası üyeliklerinde kısıtlama aşağıdaki eşitlik 3.10 ile empoze edilir (Cox 2004).

$$\sum_{j=1}^p \sum_{i=1}^n \mu_j(x_i) = n \quad (3.10)$$

$\mu_j(x_i)$  j. kümedeki,  $x_i$  nin üyelik fonksiyonu

$p$  belirlenen küme sayısı

$n$  yörünge veri nokta sayısı

Elimizdeki bu daha rahat kısıtlama ile, tüm numune noktaları için toplam üyelik niceleyicileri  $n$ 'e eşittir. Bu anlamlı küme sağlamlığını artırmak için bir yol sağlayarak, kümeleme optimizasyon problemi için çok daha esnek bir yaklaşımdır. Bu algoritma üyelik fonksiyonunda rejim olarak keyfi bir üst sınır esas almadan ziyade örneklem büyüklüğü esas aldığından bu anlamada uyarlamalı(adaptif) adını alır. Doğal olarak, bireysel üyelik değerleri artık geleneksel bulanık üyelik  $[0,1]$  aralığında değer almaz. Ancak üyeliklerin basit bir renormalizasyonu ile üyelikler bu aralığın içine eşleştirir.

Adaptif algoritma, FCM algoritması gibi oldukça basittir ve temel parametreleri içerir (parametrelerin çoğu, bulanık c ortalamalar yaklaşımı ile aynıdır). Adaptif algoritmanın hedefi, tüm bulanık kümeleme görevleri gibi, üyelik dereceleri değişen veri noktalarının kümelere atamasıdır. Bu üyelik noktasının küme temsili için derecesini yansıtmaktadır.

FA ve FCM arasındaki bu temel fark, eşitlik 3.11 deki çok küçük bir kısıtlama ile ifade edilir. Yeni bir küme merkezleri hesaplamak için kullanılan ifade aşağıdaki gibi bulanık c-means ile aynıdır (Cox 2004; Chen *et al.* 2007).

$$c_j = \frac{\sum_i \mu_j(x_i)^m x_i}{\sum_i \mu_j(x_i)^m} \quad (3.11)$$

$c_j$  j.inci kümenin merkezi

$\mu_j(\cdot)$  j.inci kümedeki  $x_i$  nin üyelik fonksiyonunu hesaplar

$x_i$  I . inci veri noktası

$m$  bulanıklaştırma parametresi

İkinci adım, bulanık üyelik fonksiyonun belirlenmesidir. Aynı tip uzaklık ölçütü kullanır. Bu eşitlik 3.12'deki gibi ifade edilir.

$$\mu_j x_i = \frac{n \frac{1}{d_{ji}^{\frac{1}{m-1}}}}{\sum_{k=1}^p \sum_{z=1}^n \frac{1}{d_{kz}^{\frac{1}{m-1}}}} \quad (3.12)$$

$\mu_j x_i$  j.inci kümedeki  $x_i$  nin üyelik fonksiyonu

$d_{ji}$   $c_j$  kümesindeki  $x_i$  için uzaklık ölçütü

$m$  bulanıklaştırma parametresi

$p$  belirtilen küme sayısı

$n$  veri noktası sayısı

$d_{kz}$   $c_k$  kümesindeki  $x_z$  için uzaklık ölçütü

FA yöntem ile geliştirilen küme üyeliği değerleri (atama için) sıfırdan büyüktür, fakat maksimumum 1'e sınırlı değildir. Konvansiyonel bulanık üyelik dağılımları gerekli olduğunda, iki süreçli normalleşme ve hedge seyreltme işlemi oluşabilir. Normalleştirme tüm kümeleri arasından en fazla üyelik değerini bulur ve bu maksimum değerden üyelikleri yeniden ölçekler.

$$\mu_{ik}^{norm} x_i = \frac{\mu_{ik}^{old}(x_i)}{\max(\mu_k^{old})}; \quad i = 1 \text{ to } n; k = 1 \text{ to } p \quad (3.13)$$

$m_{ik}^{norm} x_i$	k.kümedeki $x_i$ nin normalizasyon olmuş üyelik fonk. değeri
$m_{ik}^{old}(x_i)$	eski (orjinal) üyelik fon değeri
$p$	belirtilen küme sayısı
$n$	veri noktası sayısı
$\max()$	k.kümesindeki maksimumu üyelik fonk. değerini geri döndürür

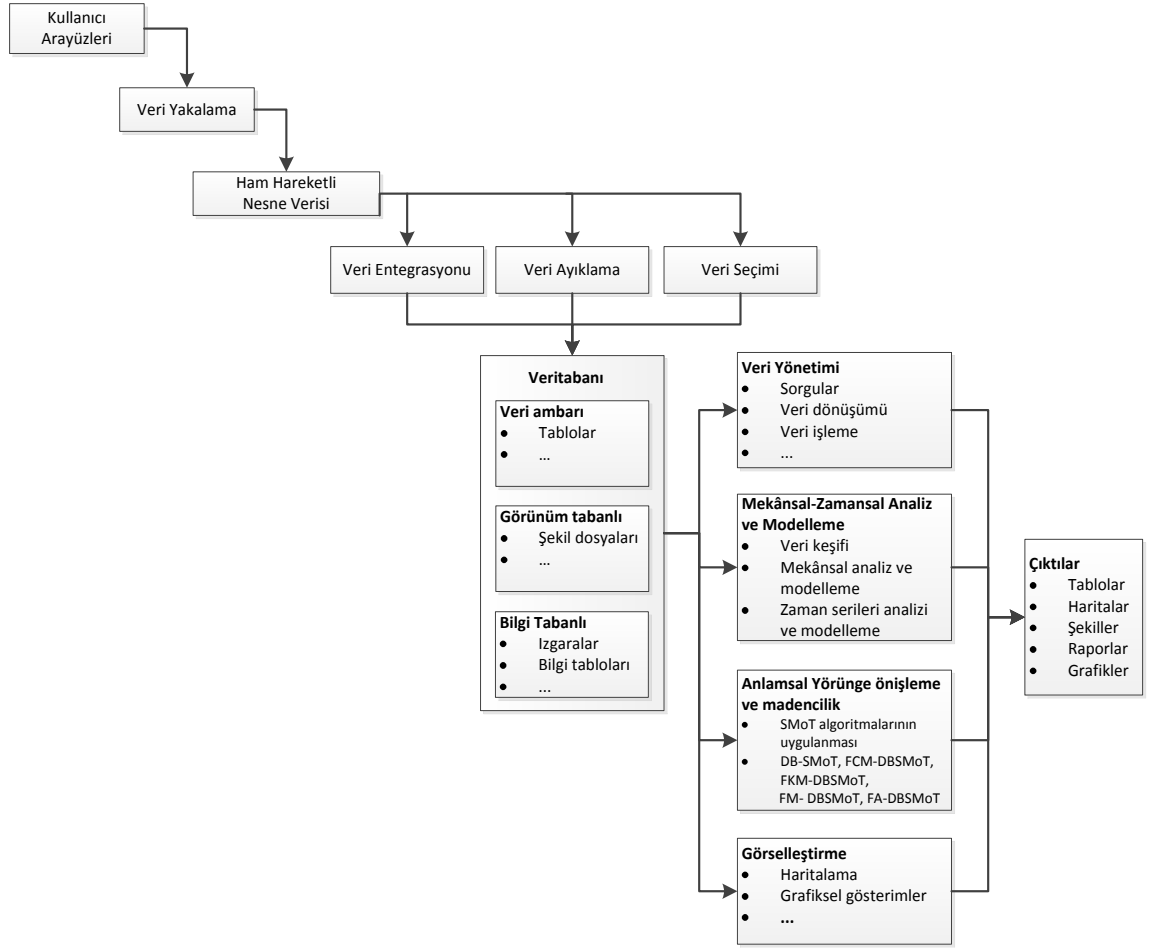
Normalizasyon, maksimum üyelik tarafından bir örnek nokta için, her bulanık üye değerlerini böler. Bu maksimum 1'e ayarlanır ve orantısal olarak üyelik değerinin geri kalanları yeniden ayarlar. FA üyelik normalizasyonu, büyük bir veri dosyası için hesaplaması yoğun bir süreç olabilir. Çünkü biz maksimum değeri bulabilmek için her küme merkez kolonunu taramamız gerekir, daha sonra üyelikler boyunca tekrar geriye dönüp bu küme için maksimum üyelik değerine bölme işlemi gerçekleştirilir.

Bulanık adaftif metodunun pseudo-kodu **EK 6**'da gösterildiği gibidir. DB-SMoT algoritmasında yörünge kümeleme yaptığı adımda, FA kullanarak uygulamamızı geliştirdik.

### 3.3. Sistem Altyapısı ve Mimarisi

Sistem birçok uygulama alanına uygulanabilir şekilde düzenlenebilmektedir. Mimarimiz, durmanın ve hareketin olduğu tüm uygulamalar için uygulanabilir. Şekil 3.1 programımızın sistem mimarisi göstermektedir. Görüldüğü üzere hareketli nesne verisi alındıktan sonra, verinin sisteme entegrasyonu, ilgili verinin ayıklanması ve veri seçimi aşamaları uygulanmaktadır. Bu hareketli nesne verilerinden anlamsal bilgiler, anlamsal yörüngeler çıkararak bilgi keşfine katkıda bulunmaktayız.





**Şekil 3.1.** Sistem Mimarisi

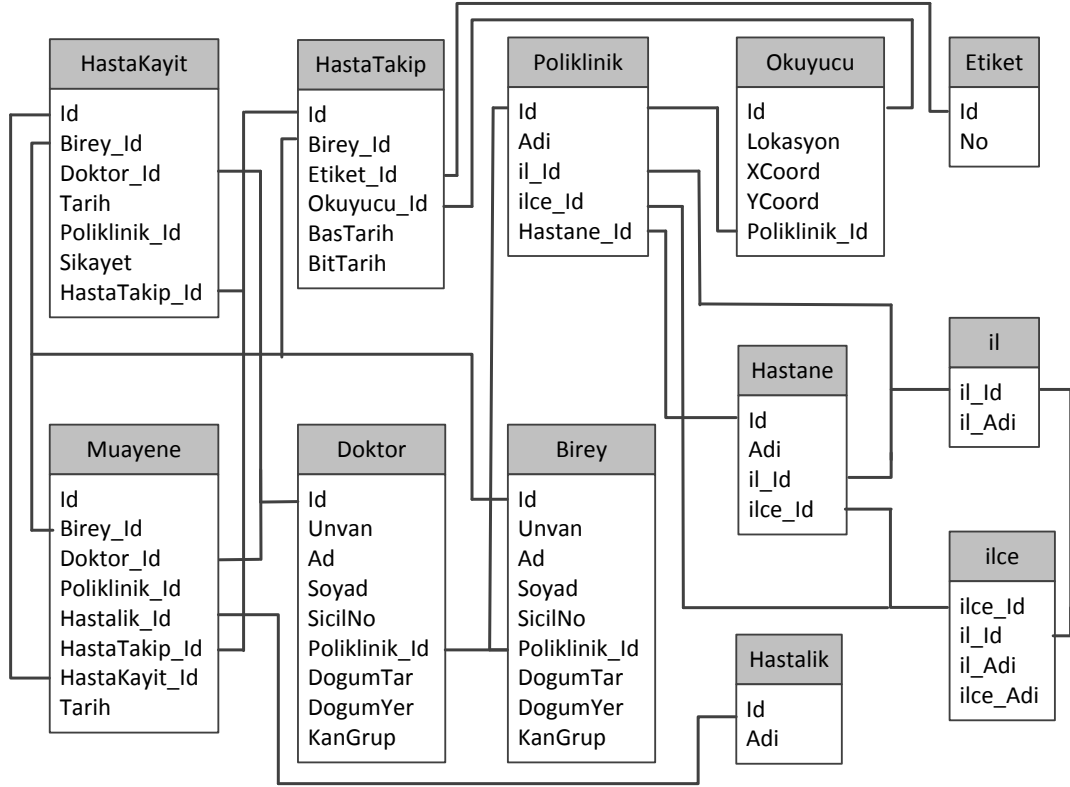
Üzerinde çalışacağımız verinin tam olarak belirlenmesinde sonra, veri tabanımızda bu verinin saklanması için veri aktarımını yapmaktayız. Veri tabanında tablolarımız, şekil dosyaları, fonksiyonlarımız bulunmaktadır. Veri yönetimi aşamasında sorgulama, veri dönüşümü, verinin işlenmesi gibi işlemler yapılmaktadır. Mekânsal-zamansal analiz ve modelleme aşamasında, verinin mekânsal ve zamansal özelliğinin ele alınarak analizi ve modellenmesi yapılmaktadır. Anlamsal yörünge önişleme aşamasında, SmoT algoritmalarının uygulanarak hareketli nesneden anlamsal yörünge çıkarımında kullanılacak uygun algoritmanın seçimi yapılmaktadır. Algoritmaların işlenmesinden sonra alınan sonuçları kullanıcılara metinsel olarak verdiğimiz gibi, görsel olarak haritalar, grafikler, raporlar tablolar ile de sunmaktayız.

Bir hastane sisteminden alınan sađlık hizmetleri hareketli nesne verisini saklayabilmek için, Şekil 3.2’de olduđu gibi bir veritabanı tasarlanmıştır. Zamansal ve mekânsal özelliđe sahip olan hareketli nesne verisinin yönetimini yapabilmek için, tablolarımızda koordinat bilgileri içeren mekânsal kavram ve zaman anını kaydeden zamansal kavramını esas alarak verilerin sistemimize uygun olarak toplanması gerçekleştirilmiştir.

Bir sađlık kuruluşunda RFID gibi kablosuz teknolojiler ile hasta takibinin yapıldığı bir alt yapıdan alınan verilerin yönetimini řu şekilde yapmaktayız. Hastanın sađlık kuruluşuna geldiğinde kayıdı alındıktan sonra, kendisine takılan RFID etiketi ile bundan sonraki işlemleri takip edilmektedir. Hastane tablomuzda, sađlık kuruluşunun adı ve il, ilçe gibi bulunduđu yer hakkında bilgiler bulunmaktadır. Poliklinik tablomuzda hangi hastanenin işlem görülen poliklinikleri olduđu bilgileri bulunmaktadır. Sađlık kuruluşunun çeşitli lokasyonlardaki polikliniklere yerleştirilen okuyucu bilgileri saklayabilmek için, Okuyucu tablosuna ihtiyaç duymaktayız. Bu tabloda koordinatları ile beraber lokasyon bilgileri bulunmaktadır. Etiket tablosunda her birey için bir tane verilen ve tekil olan numara saklanmaktadır. Hasta kayıt işlemine başlar başlamaz hasta takip tablosu içerisinde saklanan zaman ve mekân bilgileri ile hastanın tüm işlemleri takip edilmektedir. Hastanın kayıt sırasında hangi zamanda ne şikâyet ile geldiği, nereye yönlendirildiği ve muayene işlemleri yapılanaya kadar hangi lokasyonlara hangi zaman diliminde gittiğın kayıtları tablolarımızda saklanmaktadır.

Şekil 3.2’de bir veritabanı tasarımı ile haritasını aldığımız sađlık kuruluşu üzerinde, bir hastanın hareketli yörünge verisini görsel olarak çizebilmekteyiz. Dolayısıyla, incelemesi yapılacak bir hastanın tedavi süreci boyunca yapmış olduđu hareketleri zaman ve mekâna göre kullanıcılara sunmaktayız. Bu işlemi bir birey için yapabileceğimiz gibi, belirli bir şikâyet ile gelen hastaların tedavi süresince yapmış oldukları hareketliliğide görebilmekteyiz. Benzer şekilde, bir hastalığın görüldüğü mekân-zaman bilgisine erişebildiğimiz için, hastalığın görüldüğü il, ilçe, hastane, poliklinik ve zaman bilgilerini ele almaktayız. Bu sayede özel bir hastalık türünün görüldüğü mekân bilgilerini (il, ilçe, hastane gibi) alarak kümeleme yöntemleri ile

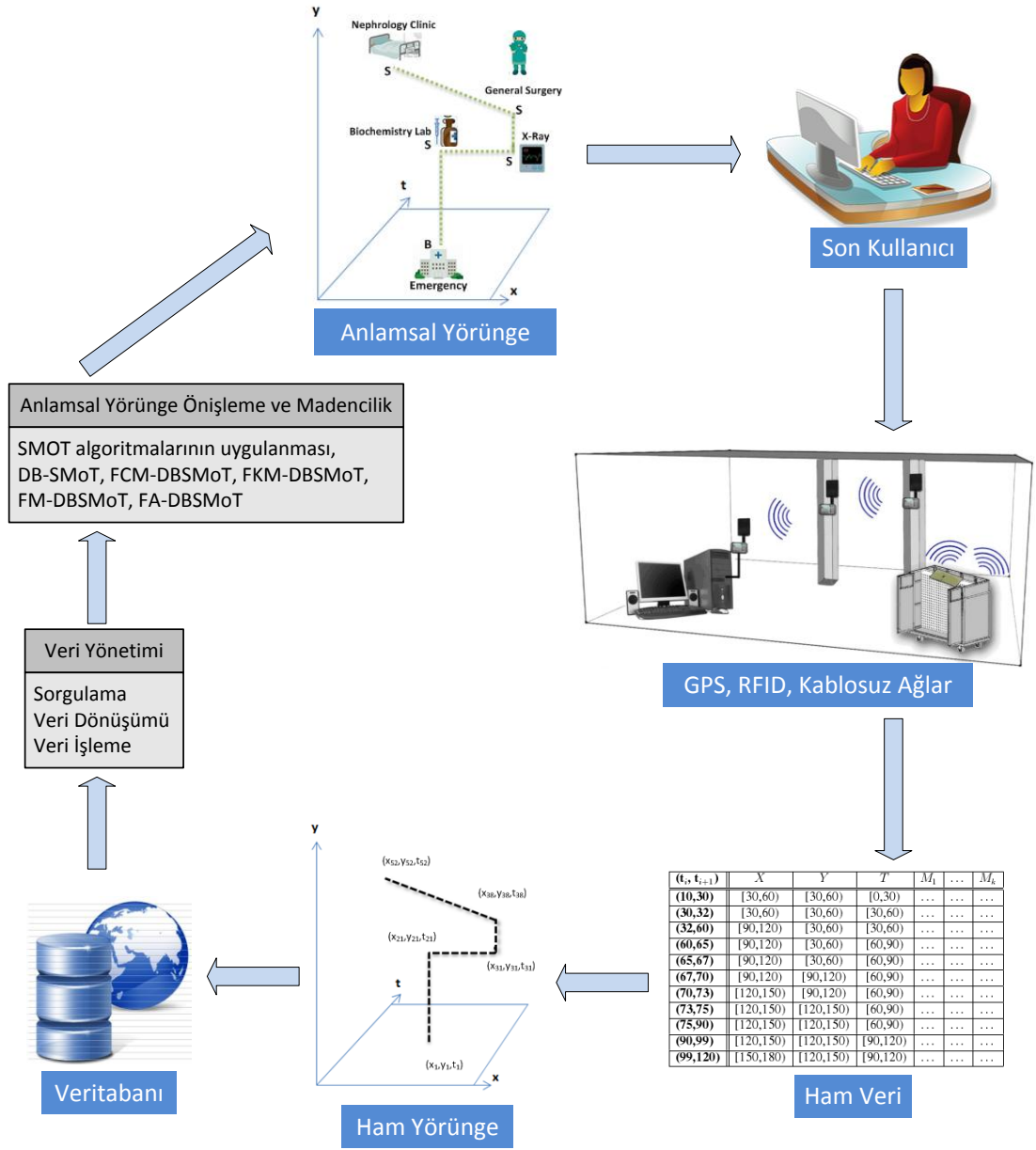
kullanıcılara bölge, il, ilçe, şehir merkezi gibi görsel harita üzerinde sonuçları sunmaktayız.



**Şekil 3.2.** Veri tabanı varlık-ilişki (ER) diyagramı

Şekil 3.3' de görüldüğü gibi, sistemde süreklilik teşkil etmektedir. Son kullanıcılardan alınan ham verilerin veritabanına alınarak yönetilmesi, işlenmesi ve uygun algoritmaların seçilerek durma ve hareketlerin yörüngesi algoritmaları yöntemlerinin uygulanması ve yine son kullanıcılara bu elde edilen bilgileri gösterilmesi şeklinde döngüsel olarak sistemin sürekliliği bulunmaktadır.


Bu yapı sayesinde kullanıcıların günlük olarak işlemlerini yaptığı ancak herhangi bir bilgi çıkarılmayan bu hareketli nesne ham verisinden kullanıcıların ve uygulama alanındaki yönetimin yorumlar ve çıkarımlar yapabileceği bilgiler görsel olarak sunulmaktadır.



Şekil 3.3. Bilgi keşfi süreci

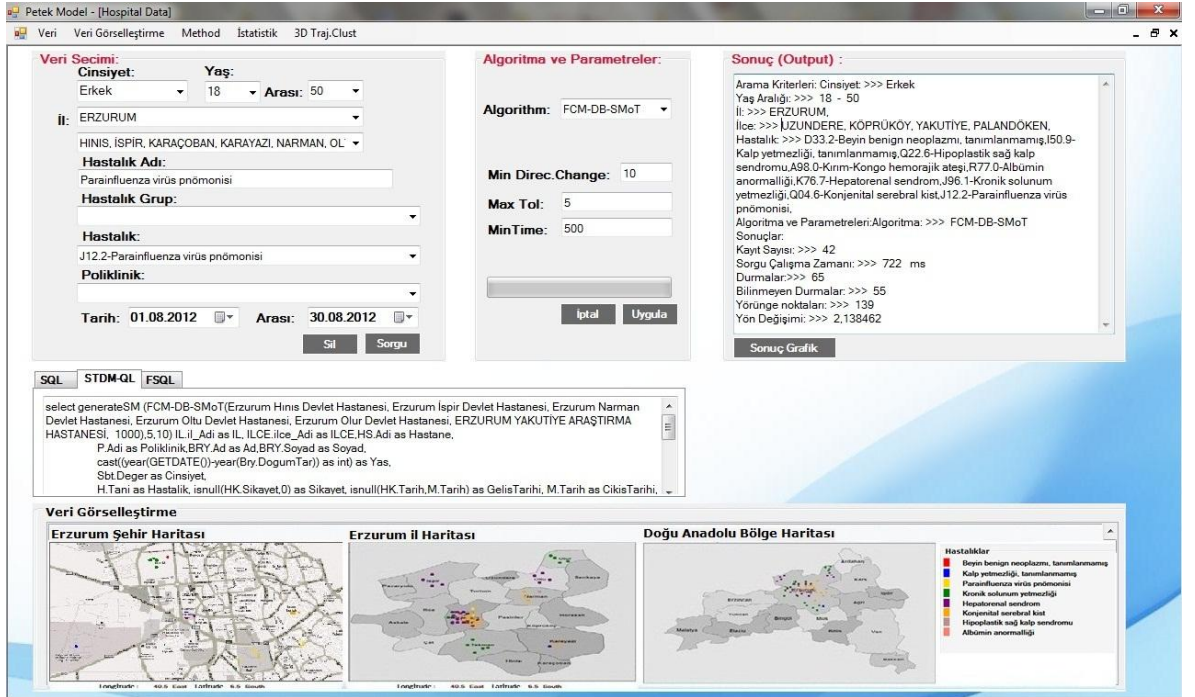
### 3.4. Program: Petek Model

Şekil 3.4’de görüldüğü gibi programımıza girişte kullanıcı adı ve parola sormaktadır. Hastane yönetimine, doktorlara, hemşirelere ve hastalara programa girişte kullanıcı adı ve parola vererek yetkisi olmayan alanlara erişimi engellenmiştir. Örneğin hastane yönetimi daha geniş bir yetki alanına sahiptir.



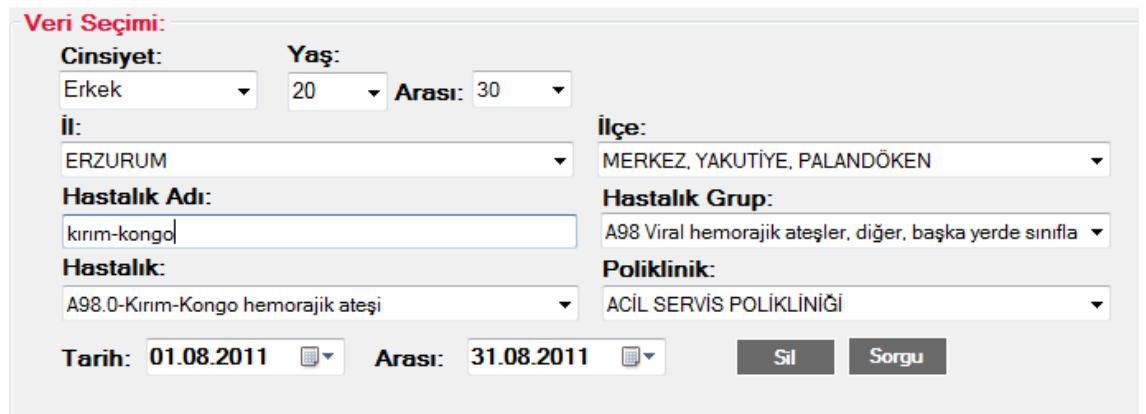
**Şekil 3.4.** Programa giriş

Şekil 3.5’de programımızın genel görünümü gözükmektedir. Programımızda 3 temel adım vardır: 1. adımda verinin seçimi, 2. adımda algoritmanın ve parametrelerin seçimi, 3. adımda ise sonuçların gösterilmesidir. Sonuçlar matematiksel, istatistiksel, haritalama ve grafiksel olarak gösterilmiştir.



Şekil 3.5. Petek Model grafiksel arayüz

Şekil 3.6'da veri seçimi adımı gösterilmiştir. Bu adımda hangi veri ile ilgili işlem yapmak istiyorsak seçilir. Cinsiyeti, yaş aralığı, ülke, şehir, hastalık ve tarih seçilerek istenilen veri için programımız çalıştırılabilir. Örnek belli hastalıklarda sadece erkek olan hastalar gözlenebilir ya da yaş aralığı 20 ile 30 arasındaki belli hastalıklar incelenebilir. Veri seçiminde hem mekan hem de zamansal özellikler seçilerek, veri tabanımızdan Mekânsal-Zamansal veri çıkartılmış olur.



Şekil 3.6. Program veri seçimi

Şekil 3.7’de algoritma ve parametre seçimi adımı gösterilmiştir. Veri seçimi yapıldıktan sonra bu veri için hangi algoritma seçileceği belirlenir ve programımıza uygulanır. Programda kullandığımız algoritmalarımız; DB-SMoT, FCM-DBSMoT, FKM-DBSMoT, FM-DBSMoT, FA-DBSMoT’dır.

**Algoritma ve Parametreler**

Algorithm: FCM-DB-SMoT

Min Direc.Change: 10

Max Tol: 5

Min Time: 500

İptal Uygula

**Şekil 3.7.** Algoritma ve parametreler

Veri seçimi yapıldıktan sonra istenilen algoritma uygulanarak programımızda sonuçlar gözükür. Şekil 3.8’de bu aşama sonucunda elde edilen çıkış gözükmektedir. Bu aşamada şekilde görüldüğü gibi matematiksel çıktılar elde edildiği gibi, programımızda anlık olarak bu değerlerin grafiksel sonuçları da gözükmektedir. Sonuç aşamasında yaptığımız en önemli yenilik ise görselleştirme ve haritalama ile sonuçları göstermemizdir.

**Sonuç:**

Arama Kriterleri:

Cinsiyet >>> Erkek  
Yaş Aralığı: >>> 20 - 65  
İl: >>> ERZURUM,  
İlçe: >>>  
Hastalık: >>> D33.2-Beyin benign neoplazmi, tanımlanmamış,Q22.6-  
Hipoplastik sağ kalp sendromu,R77.0-Albümin anormalliği,

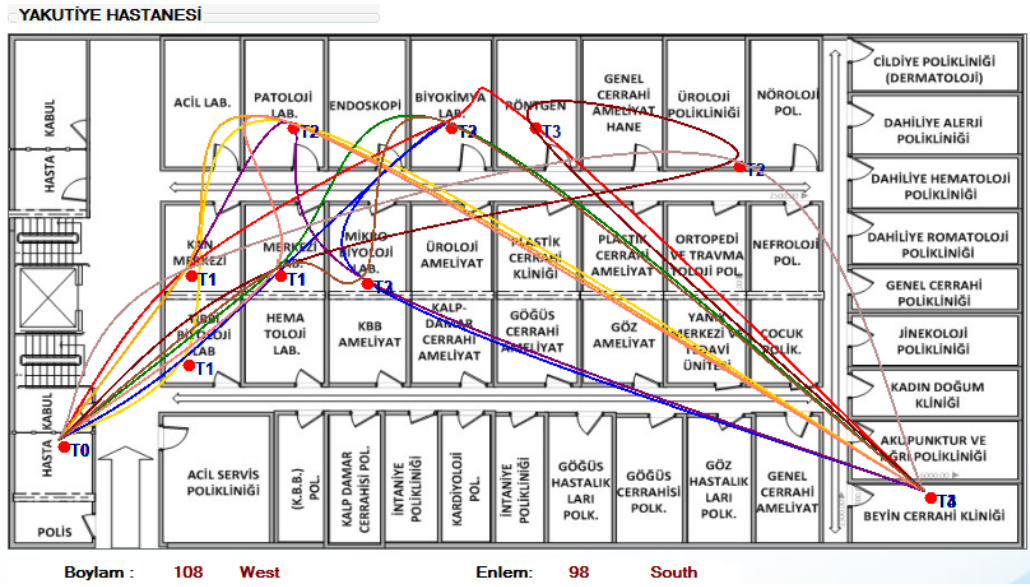
Algoritma: >>> FCM-DBSMoT

Sonuçlar:

Kayıt Sayısı: >>> 10  
Sorgu Çalışma Zamanı: >>> 273 ms  
Stops: >>> 11  
Unknown Stops: >>> 5  
Trajectory Points: >>> 31  
Direction Variation: >>> 2.818182

**Şekil 3.8.** Program sonuçları

Bir hastanın hastane içerisinde yapmış olduğu hareketler bir yörünge olacağı gibi, bir hastalığın da zaman içerisindeki yayılımı da bir eğri olarak kabul edilir. Şekil 3.9'da Albümin Anormalliği hastalığı teşhisi konulan hastaların yörünge eğrileri gözükmemektedir. Buradan yola çıkarak kan merkezi servisinde yoğunluk gözükmemektedir. Bu yoğunluğa bakılarak ek hizmet desteği verilebilir ve hizmet kalitesi artırılabilir.



Şekil 3.9. Hastalık yörüngesi

Herhangi bir hastanın, hastane içerisindeki takibini yapabileceğimiz gibi, servis bölgelerindeki işlem süresini izleyebiliriz. Bu sayede lokasyonlardaki hizmet süresini, yoğunluğu takip edebiliriz. Şekil 3.10'da bir hastanın gittiği lokasyonlar (klinik, kan merkezi v.b.) görülebilir ve bu lokasyonlardaki işlem zamanları incelenebilir. Eğer lokasyonda harcadığı işlem zamanı fazla ise o lokasyonda yoğunluğun oluşabileceği çıkarımını yapabiliriz.



HastaTakip

**Hasta Bilgileri :**

**Ad:** ██████████ **Soyad:** ██████████

**Yaş:** 59 **Cinsiyet:** Erkek **Kan Grup:** 0 Rh(+)

**Doktor:** Dr. ██████████ **Poliklinik:** KARDİYOLOJİ POLİKLİNİĞİ

**Şikayet:** Kalp çarpıntısı, nefes darlığı. **Doktor Tanı:** HİPOPLASTİK SAĞ KALP SENDROMU

**Ort. İşlem Zamanı :**

Ortalama Saniye: > 825

Ortalama İşlem Zamanı: > 0 Gün 0 Saat 13 Dakika 45 Saniye

**Lokasyon İşlem Zamanı:**

> BİYOKİMYA LABORATUVARI

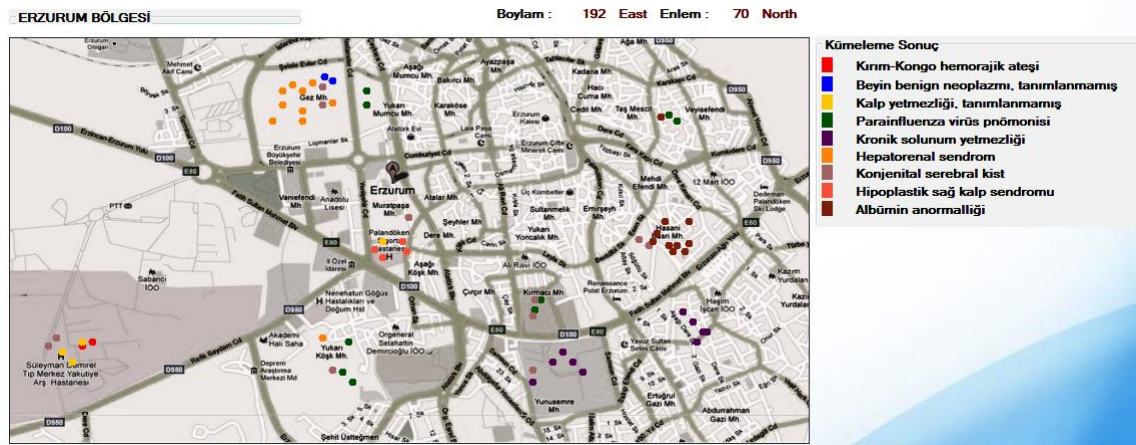
Ortalama İşlem Zamanı: > 0 Gün 0 Saat 5 Dakika 0 Saniye

**Hasta İşlemleri:**

Hastane Adı	Poliklinik	Giriş Tarihi	Çıkış Tarihi
ERZURUM YAKÜTİYE ARAŞTIRMA HASTANESİ	HASTA KABUL	01.08.2011 10:10	01.08.2011 10:15
ERZURUM YAKÜTİYE ARAŞTIRMA HASTANESİ	MERKEZİ LAB.	01.08.2011 10:25	01.08.2011 10:45
ERZURUM YAKÜTİYE ARAŞTIRMA HASTANESİ	BIYOKİMYA LABORATUVARI	01.08.2011 10:50	01.08.2011 10:55
ERZURUM YAKÜTİYE ARAŞTIRMA HASTANESİ	KARDİYOLOJİ POLİKLİNİĞİ	01.08.2011 11:10	01.08.2011 11:35

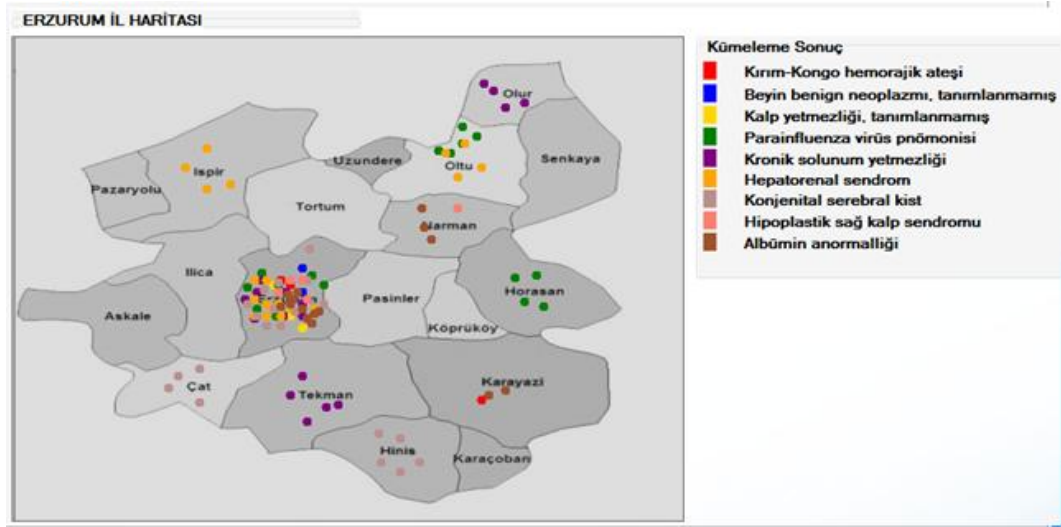
Şekil 3.10. Hasta takip

Şekil 3.11, Erzurum'un Google Maps' den elde edilen görüntüsünde seçilen hastalıkların kümelenmesini göstermektedir. Seçilen hastalıkların hangi bölgelerde gözüktüğü bilgisini de erişebiliriz.



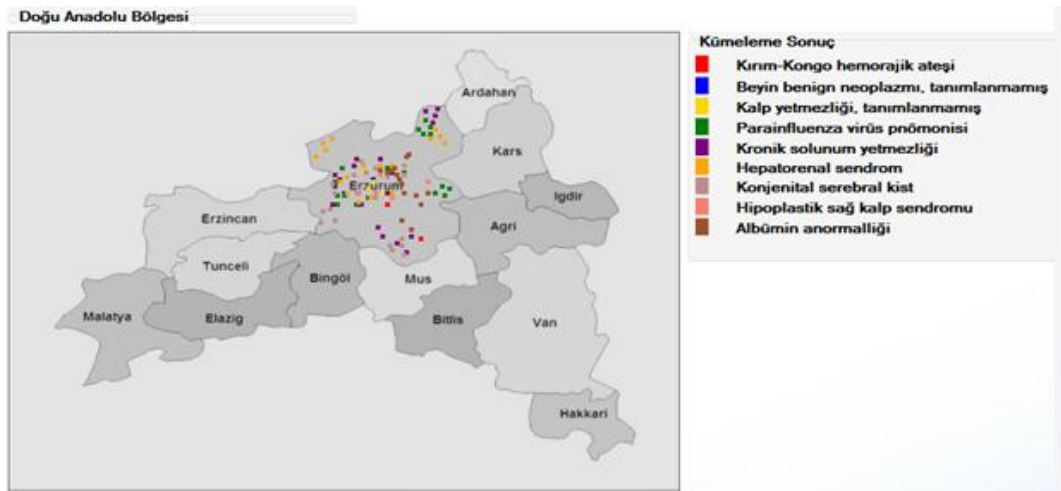
Şekil 3.11. Harita üzerinde hastalık kümelenmesi

Şekil 3.12’de Erzurum haritasında seçilen hastalıkların nerelerde gözüktüğünü görebiliriz. Seçilen hastalıkların kümelenmesi ilçe haritası üzerinde zaman ve mekan boyutuna göre gösterilmiştir.



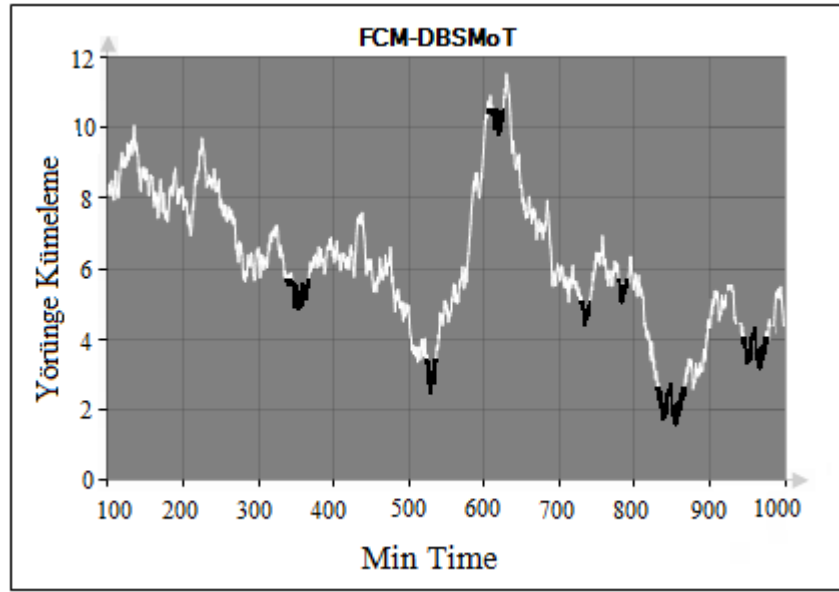
Şekil 3.12. Erzurum ili hastalık kümelemesi

Şekil 3.13’da Doğu Anadolu bölgesindeki hastalıkların kümelenmesini göstermektedir. Bu sayede bölgesel olarak hastalıkların takibi de sağlanabilir. Hangi hastalık hangi bölgede ve bölgede başka nerelerde görüldüğü bilgilerine de ulaşabiliriz.



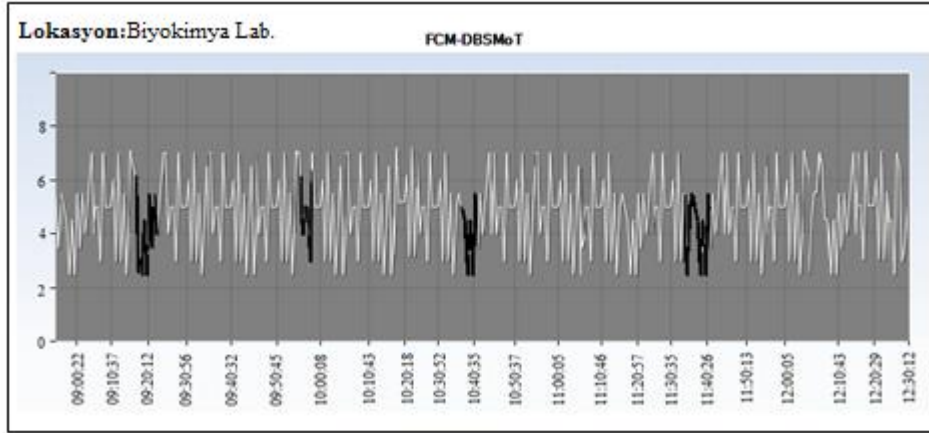
Şekil 3.13. Doğu Anadolu Bölgesi hastalık kümelemesi

Yörünge kümeleme durmaların ve hareketlerin yoğun olduğu benzer noktaları sınıflandırmak için önemli bir süreçtir. Yörünge kümeleme verinin analizi için çok önemli bir role sahiptir ve hareketli nesnelerin temel eğilimlerini ortaya çıkarmaktadır. Biz zaman ve mekana göre benzer hareket kalıpları gösteren yörünge kümelerini tespit etmeye çalıştık. Şekil 3.14, FCM-DBSMoT algoritması için minTime parametresine göre değişen Yörünge kümeleme grafiğini göstermektedir.



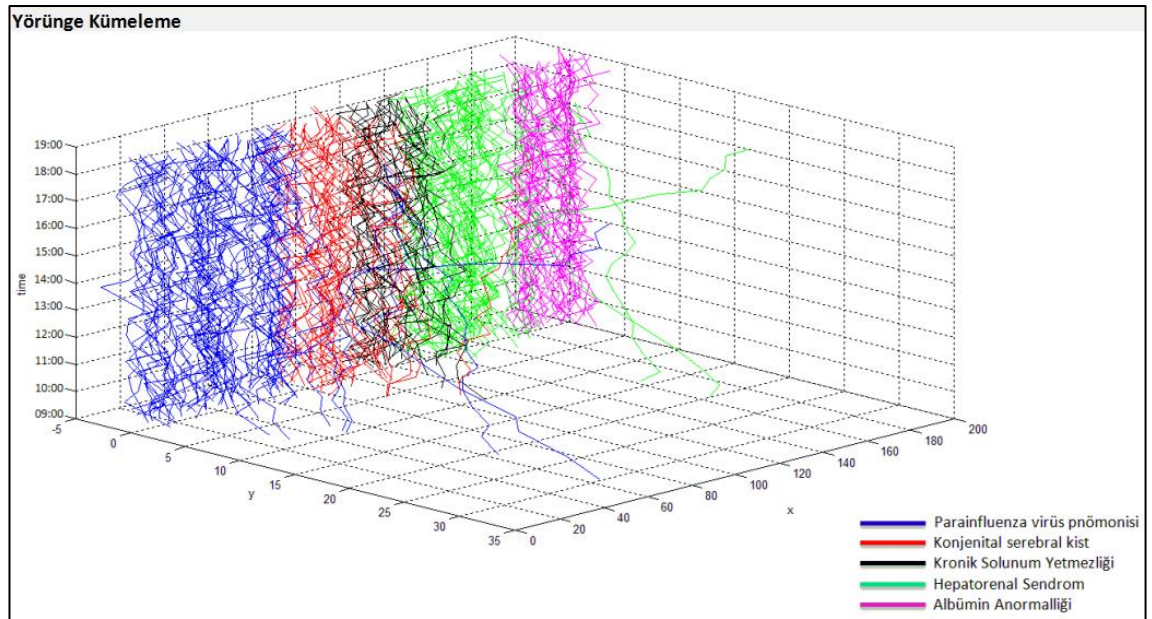
**Şekil 3.14.** Yörünge kümeleme

Benzer olarak, farklı lokasyonların da zamana göre yörünge kümelemesi çıkartılabilir. Şekil 3.15’de Biyokimya laboratuvarının zamana göre FCM-DBSMoT algoritması kullanılarak yörünge kümesi gösterilmiştir. Biz zamana göre lokasyon yoğunluğunu tespit etmeye çalıştık. Bu hastane yönetimi için belirli bir konumda hizmet kalitesini artırmak için faydalı olabilir.



**Şekil 3.15.** Laboratuvarın yörünge kümelemesi

Bir lokasyonda zaman ve mekana göre 3 boyutlu olarak görselleştirme şekil 3.16'da gösterildiği gibidir. Burada hastalıkların yörünge kümelemesini görebiliriz.



**Şekil 3.16.** 3-Boyutlu yörünge kümeleme

Algoritma ve parametre seçimi adımı, seçilen veri için ilgili algoritma ve parametreleri seçilerek programımıza uygulanabilir. DB-SMoT algoritması temel alınarak FCM-DBSMoT, FKM-DBSMoT, FM-DBSMoT ve FA-DBSMoT

algoritmaları geliştirilmiştir. Verimize bu 5 algoritma uygulanarak çeşitli çıkarımlar alınmıştır.

Sonuç adımında ise; ilgili veri için seçilen algoritma uygulandıktan sonra algoritmanın sonuçları gözükmemektedir. Bu aşamada algoritmanın çalışma zamanı, kayıt sayısı gibi sonuçlar da gözükmemektedir. Aynı zamanda bu aşamada elde edilen değerler için grafiksel sonuçlar da üretilmiştir

## 4. ARAŞTIRMA BULGULARI VE DENEYSEL SONUÇLAR

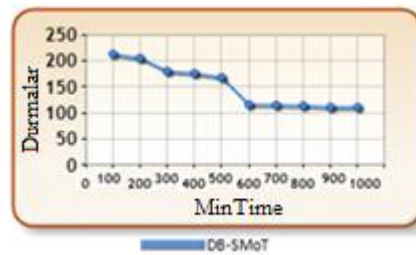
### 4.1. DB-SMoT Deneysel Sonuçlar

Hastane verisine DB-SMoT algoritması uygulanarak çeşitli sonuçlar çıkardık. Öncelikle cinsiyeti ve yaş aralığı seçilir. Daha sonra hangi hastalıkları incelemek istiyorsak, hastalıklar seçilir. Daha sonra tarih aralığı seçilerek, Algoritmamız ve parametreleri yazılarak uygulanır. Çizelge 4.1’ de görüldüğü gibi seçilen hastalıklara MinTime parametresi 500 girildiği zaman durma ve bilinmeyenDurma değerleri gözükmemektedir. Aynı zamanda Algoritmanın çıkışları olan Çalışma zamanı, Yörünge noktası ve yön değişimi değerleri gözükmemektedir. MinTime aralığı 100-300 arasında ve 700-1000 arasında benzer sonuçlar üretmiştir. En iyi sonuçlar minTime 500 ve 600 seçildiği zaman elde edilmiştir.

**Çizelge 4.1.** MinTime değeri 500 iken DB-SMoT algoritması sonuçları

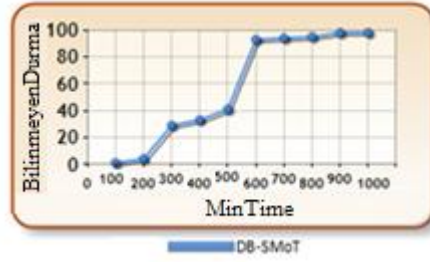
Algoritma	minTime	Durma	BilinmeyenDurma	Yörünge noktası	Yön değişimi	Çalışma zamanı
DB-SMoT	500	168	41	224	1,3333	590

Şekil 4.1, MinTime parametresine göre Durma sayısını göstermektedir. MinTime değeri 600 olana kadar, algoritma iyi sonuç üretememiştir. En iyi sonuç minTime parametresi 600 iken elde edilmiştir. MinTime değeri 600 değerine ulaştıktan sonra benzer sonuçlar üretmiştir. Ne kadar az sürede ne kadar çok Durma bulursa algoritmamız o kadar iyi çalışır.



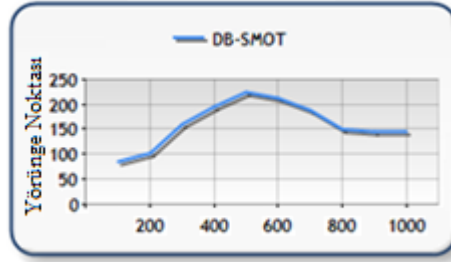
**Şekil 4.1.** MinTime parametresine göre Durmalar

Şekil 4.2, MinTime parametresine göre BilinmeyenDurma sayısını göstermektedir. MinTime değeri 500 den 600'e çıkarken daha çok BilinmeyenDurma bulmuştur. MinTime değeri 600 olduktan sonra ise benzer sonuçlar üretmiştir.



**Şekil 4.2.** MinTime parametresine göre BilinmeyenDurmalar

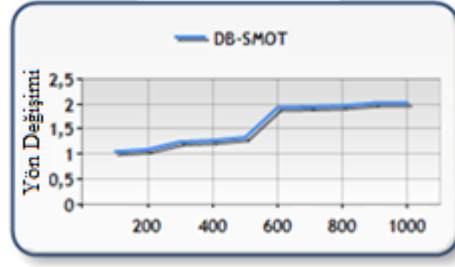
Şekil 4.3, MinTime parametresine göre Yörünge Noktası sayısını göstermektedir. MinTime değeri 500 olana kadar Yörünge Noktası sayısı artarken, bu değerden sonraki MinTime değerleri için ise azalmıştır. MinTime 800 değerinden sonra ise benzer sonuç üretmiştir.



**Şekil 4.3.** MinTime parametresine göre Yörünge Noktası

Şekil 4.4, MinTime parametresine göre Yön Değişimini göstermektedir. En iyi sonucu minTime 600 iken elde etmiştir. MinTime değeri 600 olduktan sonra ise benzer sonuç üretmiştir.





Şekil 4.4. MinTime parametresine göre Yön Değişimi

#### 4.2. FCM-DBSMoT Deneysel Sonuçlar

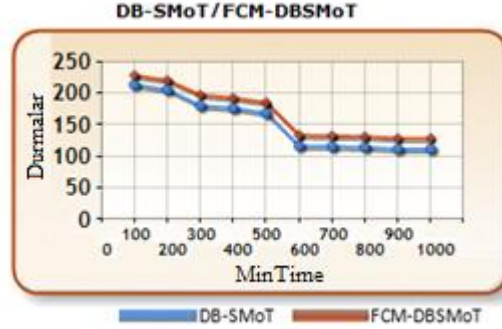
Çizelge 4.2’de görüldüğü gibi DB-SMoT ve FCM-DBSMoT algoritmaları karşılaştırılmıştır. Aynı veri için her iki algoritma da çalıştırılmıştır, tablodan da rahatlıkla görüldüğü gibi FCM-DBSMoT algoritması DB-SMoT algoritmasından daha çok Durma bulmuştur. Algoritma ne kadar çok Durma bulursa o kadar iyi performans sağlar. Aynı zamanda FCM-DBSMoT algoritması daha az BilinmeyenDurma bulmuştur. Algoritma ne kadar az BilinmeyenDurma bulursa o kadar iyi performans sağlar. FCM-DBSMoT algoritması daha çok yörünge sayısına sahipti ve daha kısa sürede çalışmıştır. Yukarıda yazılan nedenlerden dolayı FCM-DBSMoT algoritması DBSMoT algoritmasından daha iyi sonuç üretmiştir.

Çizelge 4.2. MinTime değeri 500 iken DB-SMoT/FCM-DBSMoT algoritmalarının karşılaştırması

Algoritma	minTime	Durma	BilinmeyenDurma	Yörünge Noktası	Yön Değişimi	Çalışma Zamanı
DB-SMoT	500	168	41	224	1,3333	590
FCM-DBSMoT	500	170	40	229	1,3471	510

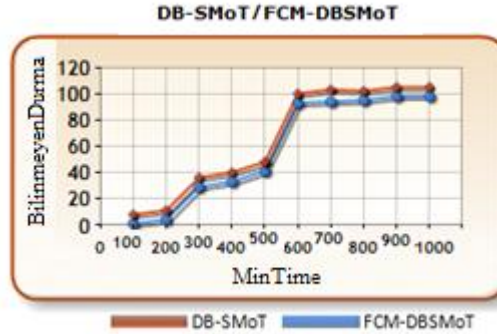
Şekil 4.5, MinTime parametresine göre Durma sayısının DB-SMoT/FCM-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Şekilde görüldüğü gibi FCM-DBSMoT algoritması daha çok Durma bulmuştur. FCM-DBSMoT algoritması bu sebepten dolayı daha iyi bir algoritmadır. MinTime 600’den sonra ise her iki algoritmada minTime 600 de aldıkları değerlere benzer sonuçlar üretmiştir.





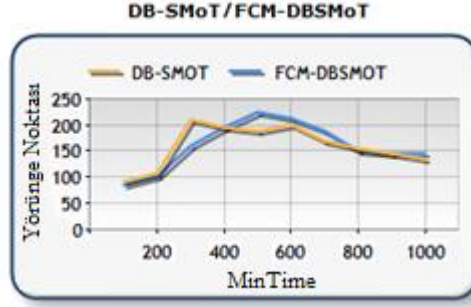
**Şekil 4.5.** MinTime parametresine göre Durma sayısının DB-SMoT/FCM-DBSMoT algoritmalarının karşılaştırması

Şekil 4.6, MinTime parametresine göre BilinmeyenDurma sayısının DB-SMoT/FCM-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Şekilde görüldüğü gibi FCM-DBSMoT algoritması daha az BilinmeyenDurma bulmuştur. FCM-DBSMoT algoritması bu sebepten dolayı daha iyi bir algoritmadır.



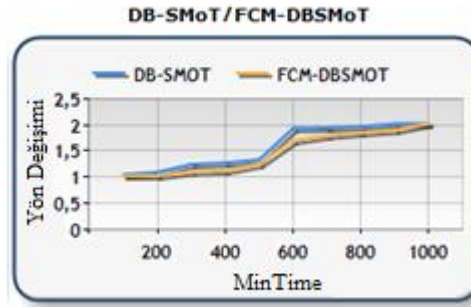
**Şekil 4.6.** MinTime parametresine göre BilinmeyenDurma sayısının DB-SMoT/FCM-DBSMoT algoritmalarının karşılaştırması

Şekil 4.7, MinTime parametresine göre Yörünge Noktası sayısının DB-SMoT/FCM-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Şekilde görüldüğü gibi bazen DB-SMoT, bazen de FCM-DBSMoT algoritması daha iyi sonuç üretmiştir. Örneğin minTime 500 için FCM-DBSMoT daha iyi sonuç üretmiştir, minTime 300 için ise DB-SMoT daha iyi sonuç üretmiştir.



**Şekil 4.7.** MinTime parametresine göre Yörünge Noktası sayısının DB-SMoT/FCM-DBSMoT algoritmalarının karşılaştırması

Şekil 4.8, MinTime parametresine göre Yön Değişiminin DB-SMoT/FCM-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Şekilde görüldüğü gibi FCM-DBSMoT algoritması daha az yön değişimi yapmıştır. FCM-DBSMoT algoritması bu sebepten dolayı daha iyi bir algoritmadır.



**Şekil 4.8.** MinTime parametresine göre Yön Değişiminin DB-SMoT/FCM-DBSMoT algoritmalarının karşılaştırması

### 4.3. FKM-DBSMoT Deneysel Sonuçlar

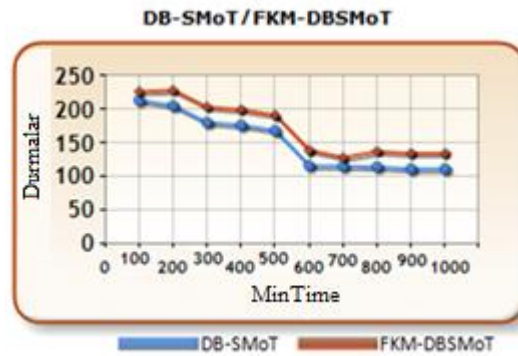
Çizelge 4.3'de görüldüğü gibi DB-SMoT ve FKM-DBSMoT algoritmaları karşılaştırılmıştır. Aynı veri için her iki algoritma da çalıştırılmıştır, tablodan da rahatlıkla görüldüğü gibi FKM-DBSMoT algoritması DB-SMoT algoritmasından daha çok Durma bulmuştur. Aynı zamanda FKM-DBSMoT algoritması DB-SMoT algoritmasından daha çok yörünge noktası bulmuştur. Çalışma zamanlarına bakıldığı zaman ise FKM-DBSMoT algoritması DB-SMoT algoritmasından daha kısa sürede sonuç üretmiştir. FKM-DBSMoT algoritması DB-SMoT algoritmasından daha az

BilinmeyenDurma üretmiştir. Görüldüğü üzere her özellik incelendiği zaman minTime 500 için FKM-DBSMoT algoritması daha iyi bir algoritmadır.

**Çizelge 4.3.** MinTime değeri 500 iken DB-SMoT/FKM-DBSMoT algoritmalarının karşılaştırması

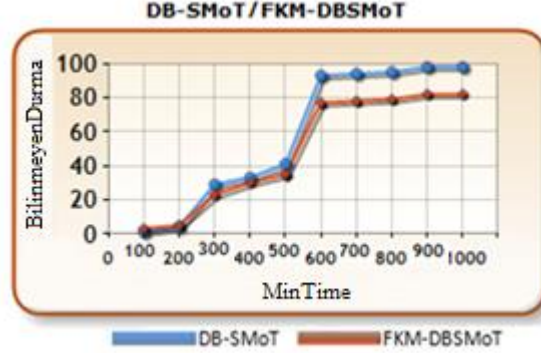
Algoritma	minTime	Durma	BilinmeyenDurma	Yörünge noktası	Yön değişimi	Çalışma zamanı
DB-SMoT	500	168	41	224	1,3333	590
FKM-DBSMoT	500	171	39	227	1,3274	570

Şekil 4.9, MinTime parametresine göre Durma sayısının DB-SMoT/FKM-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Her iki algoritma minTime değeri arttıkça daha az Durma bulmuştur. Şekilde görüldüğü gibi FCM-DBSMoT algoritması daha çok Durma bulmuştur. FKM-DBSMoT algoritması bu sebepten dolayı daha iyi bir algoritmadır. MinTime 600'den sonra ise her iki algortmada minTime 600 de aldıkları değerlere benzer sonuçlar üretmiştir.



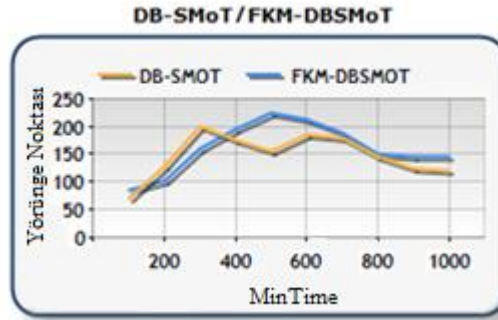
**Şekil 4.9.** MinTime parametresine göre Durma sayısının DB-SMoT/FKM-DBSMoT algoritmalarının karşılaştırması

Şekil 4.10, MinTime parametresine göre BilinmeyenDurma sayısının DB-SMoT/FKM-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Şekilde görüldüğü gibi FKM-DBSMoT algoritması daha az BilinmeyenDurma bulmuştur. FKM-DBSMoT algoritması bu sebepten dolayı daha iyi bir algoritmadır.



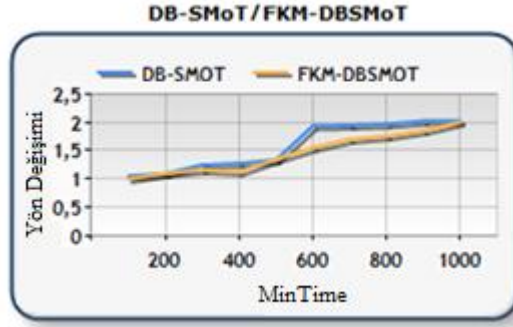
**Şekil 4.10.** MinTime parametresine göre BilinmeyenDurma sayısının DB-SMoT/FKM-DBSMoT algoritmalarının karşılaştırması

Şekil 4.11, MinTime parametresine göre Yörünge Noktası sayısının DB-SMoT/FKM-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Şekilde görüldüğü gibi bazen DB-SMoT, bazen de FKM-DBSMoT algoritması daha iyi sonuç üretmiştir. Örneğin minTime 500 için FKM-DBSMoT daha iyi sonuç üretmiştir, minTime 300 için ise DB-SMoT daha iyi sonuç üretmiştir.



**Şekil 4.11.** MinTime parametresine göre Yörünge Noktası sayısının DB-SMoT/FKM-DBSMoT algoritmalarının karşılaştırması

Şekil 4.12, MinTime parametresine göre Yön Değişiminin DB-SMoT/FKM-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Şekilde görüldüğü gibi FKM-DBSMoT algoritması çoğu zaman daha az yön değişimi yapmıştır.



**Şekil 4.12.** MinTime parametresine göre Yön Değişiminin DB-SMoT/FKM-DBSMoT algoritmalarının karşılaştırması

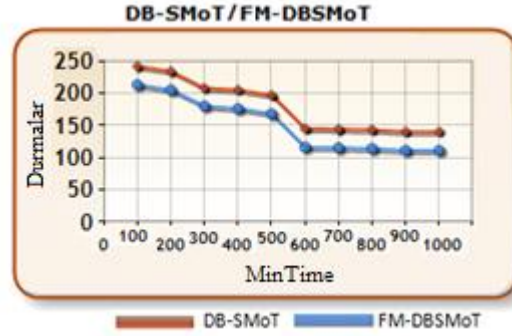
#### 4.4. FM-DBSMoT Deneysel Sonuçlar

Çizelge 4.4’de görüldüğü gibi DB-SMoT ve FM-DBSMoT algoritmaları karşılaştırılmıştır. Aynı veri için her iki algoritma da çalıştırılmıştır, DB-SMoT algoritması FM-DBSMoT algoritmasından daha çok Durma bulmuştur. DB-SMoT algoritması FM-DBSMoT algoritmasından daha az BilinmeyenDurma bulmuştur. Aynı zamanda DB-SMoT algoritması daha az yön değişimine sahiptir. Bu özelliklerden dolayı DB-SmoT algoritması daha iyi bir algoritmadır.

**Çizelge 4.4.** MinTime değeri 500 iken DB-SMoT/FM-DBSMoT algoritmalarının karşılaştırması

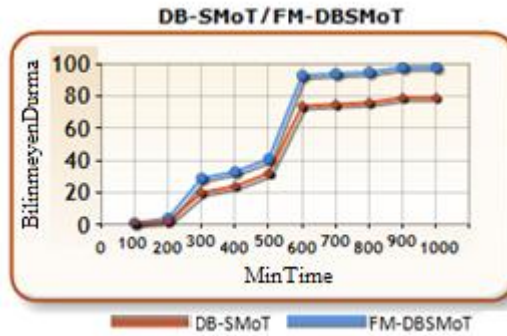
Algoritma	minTime	Durma	BilinmeyenDurma	Yörünge noktası	Yön değişimi	Çalışma zamanı
DB-SMoT	500	168	41	224	1,3333	590
FM-DBSMoT	500	167	42	226	1,3532	584

Şekil 4.13, MinTime parametresine göre Durma sayısının DB-SMoT/FM-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Her iki algoritma minTime değeri arttıkça daha az Durma bulmuştur. Şekilde görüldüğü gibi DBSMoT algoritması daha çok Durma bulmuştur. DBSMoT algoritması bu sebepten dolayı daha iyi bir algoritmadır. MinTime 600’den sonra ise her iki algortmada minTime 600 de aldıkları değerlere benzer sonuçlar üretmiştir.



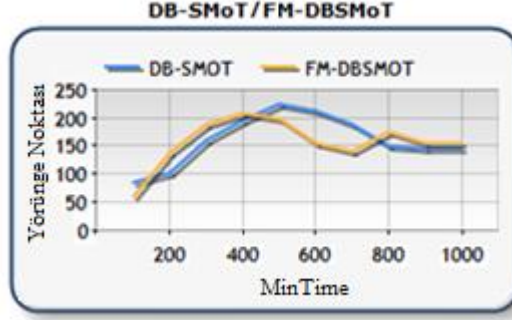
**Şekil 4.13.** MinTime parametresine göre Durma sayısının DB-SMoT/FM-DBSMoT algoritmalarının karşılaştırması

Şekil 4.14, MinTime parametresine göre BilinmeyenDurma sayısının DB-SMoT/FM-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Şekilde görüldüğü gibi DBSMoT algoritması daha az BilinmeyenDurma bulmuştur. Bu sebepten dolayı DBSMoT algoritması daha iyi bir algoritmadır.



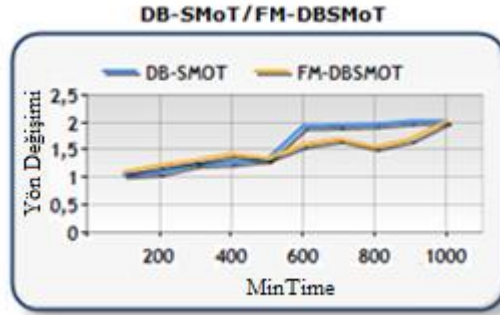
**Şekil 4.14.** MinTime parametresine göre BilinmeyenDurma sayısının DB-SMoT/FM-DBSMoT algoritmalarının karşılaştırması

Şekil 4.15, MinTime parametresine göre Yörünge Noktası sayısının DB-SMoT/FM-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Şekilde görüldüğü gibi bazen DB-SMoT, bazen de FM-DBSMoT algoritması daha iyi sonuç üretmiştir. Örneğin minTime 400 ile 800 arasında DB-SMoT daha iyi sonuç üretmiştir.



**Şekil 4.15.** MinTime parametresine göre Yörünge Noktası sayısının DB-SMoT/FM-DBSMoT algoritmalarının karşılaştırması

Şekil 4.16, MinTime parametresine göre Yön Değişiminin DB-SMoT/FM-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Yön değişimleri genelde benzer sonuç üretmiştir. FM-DBSMoT algoritması minTime 500den sonra daha az yön değişimine sahiptir.



**Şekil 4.16.** MinTime parametresine göre Yön Değişiminin DB-SMoT/FM-DBSMoT algoritmalarının karşılaştırması

#### 4.5. FA-DBSMoT Deneysel Sonuçlar

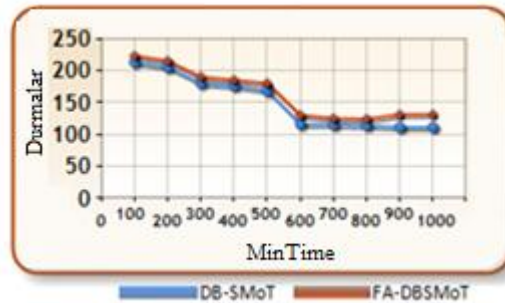
Çizelge 4.5' de görüldüğü gibi DB-SMoT ve FA-DBSMoT algoritmaları karşılaştırılmıştır. Aynı veri için her iki algoritma da çalıştırılmıştır, tablodan da rahatlıkla görüldüğü gibi FA-DBSMoT algoritması DB-SMoT algoritmasından daha çok Durma bulmuştur. Aynı zamanda FA-DBSMoT algoritması DB-SMoT algoritmasından daha çok yörünge noktası bulmuştur. Çalışma zamanlarına bakıldığı zaman ise FA-DBSMoT algoritması DB-SMoT algoritmasından daha kısa sürede sonuç

üretmiştir. Görüldüğü üzere özellikler incelendiği zaman FA-DBSMoT algoritması daha iyi bir algoritmadır.

**Çizelge 4.5.** MinTime değeri 500 iken DB-SMoT/ FA-DBSMoT algoritmalarının karşılaştırması

Algoritma	minTime	Durma	BilinmeyenDurma	Yörünge noktası	Yön değişimi	Çalışma zamanı
DB-SMoT	500	168	41	224	1,3333	590
FA-DBSMoT	500	169	41	228	1,3491	514

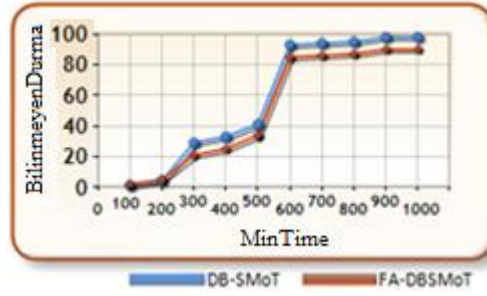
Şekil 4.17, MinTime parametresine göre Durma sayısının DB-SMoT/FA-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Her iki algoritma minTime değeri arttıkça daha az Durma bulmuştur. Şekilde görüldüğü gibi FA-DBSMoT algoritması çok az bir farkla da olsa daha çok Durma bulmuştur. FA-DBSMoT algoritması bu sebepten dolayı biraz daha iyi bir algoritmadır. MinTime 600'den sonra ise her iki algoritmada minTime 600 de aldıkları değerlere benzer sonuçlar üretmiştir.



**Şekil 4.17.** MinTime parametresine göre Durma sayısının DB-SMoT/FA-DBSMoT algoritmalarının karşılaştırması

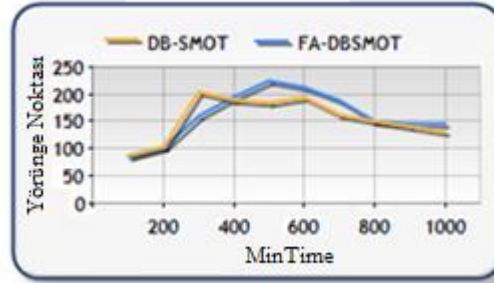
Şekil 4.18, MinTime parametresine göre BilinmeyenDurma sayısının DB-SMoT/FA-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Şekilde görüldüğü gibi FA-DBSMoT algoritması daha az BilinmeyenDurma bulmuştur. FA-DBSMoT algoritması bu sebepten dolayı daha iyi bir algoritmadır.





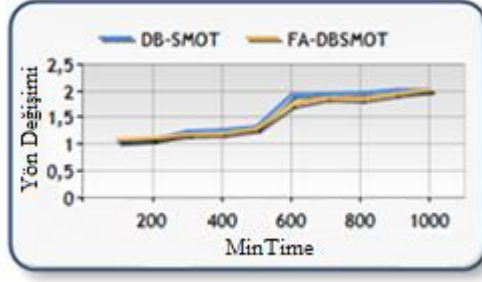
**Şekil 4.18.** MinTime parametresine göre BilinmeyenDurma sayısının DB-SMoT/FA-DBSMoT algoritmalarının karşılaştırması

Şekil 4.19, MinTime parametresine göre Yörünge Noktası sayısının DB-SMoT/FA-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Şekilde görüldüğü gibi bazen DB-SMoT, bazen de FA-DBSMoT algoritması daha iyi sonuç üretmiştir. Örneğin minTime 500 için FA-DBSMoT daha iyi sonuç üretmiştir, minTime 300 için ise DB-SMoT daha iyi sonuç üretmiştir.



**Şekil 4.19.** MinTime parametresine göre Yörünge Noktası sayısının DB-SMoT/FA-DBSMoT algoritmalarının karşılaştırması

Şekil 4.20, MinTime parametresine göre Yön Değişiminin DB-SMoT/FA-DBSMoT algoritmalarının karşılaştırmasını göstermektedir. Şekilde görüldüğü gibi FA-DBSMoT algoritması daha az yön değişimi yapmıştır



**Şekil 4.20.** MinTime parametresine göre Yön Değişiminin DB-SMoT/FA-DBSMoT algoritmalarının karşılaştırması

## 5. SONUÇ ve ÖNERİLER

Algoritmaların zaman performansını iyileştirmek adına yapılan sorgu veri taramasında indekisleme, genel sistemin yanıt süresini azaltma, kullanılan bellek boyutunu azaltma, veri türlerini doğru belirleme, ham veriden veri temizliği yapma ve donanım iyileştirilmesiyle hızlı cevap veren bir mimari tasarlanmıştır.

Petek model programı sayesinde hasta takibi yapılabilir. Hastanın anlamsal yörüngesine bakılarak hastanın hangi lokasyonlara gittiği ve bu lokasyonlarda ne kadar kaldığı bilgisini çıkarabiliriz.

Bölgesel olarak benzer hastalıkların sınıflandırılması, kümelenmesi, benzerliğini görebiliriz. Hastalıkların anlamsal yörüngelerine bakılarak örneğin Erzurum ili Hasankale ilçesinde görülen hastalıklar bulunabilir. Bu sayede bir bölgede gözüken benzer hastalıklar bulunabilir.

Tehlikeli bir hastalığın ilçe ve ildeki yayılımı görülebilir. Örneğin kırım-kongo hemorajik ateşi hastalığının Erzurum ilinde hangi ilçelerinde gözüktüğü bulunabilir. Bu sayede gerekli önlemler alınarak hastalığın yayılımı önenebilir.

Salgın bir hastalık türünün ilk olarak hangi yerlerde görüldüğü bilgisini çıkarabiliriz. Örneğin kuş gribi hastalığının anlamsal yörüngesine bakılarak, bu hastalığın yayılımı görülebilir. Bu hastalığın ilk nerede görüldüğünü ve daha sonra nerelerde görüldüğünü görebiliriz. Yani hastanın yörüngesi çıkarılabildiği gibi hastalığın da anlamsal yörüngesinin çıkarımı yapılabilir. Hastalığın ilk görüldüğü yerden itibaren yayılımı görülebilir. Hastalığın mekânsal-zamansal olarak görsel sınıflandırılması ve görsel kümelenmesi yapılabilmektedir.

Sağlık hizmetlerinde klinik yoğunlukları ve sağlık hizmet verilerinin yönetimi ve anlamsal yörüngeler çıkarımı yapılabilir. Hastaneye gelen hastaların yörüngelerine bakılarak nerede yoğunluk olduğunu görebiliriz. Bu lokasyona eleman ihtiyacı

sağlanırsa yoğunluk azaltılabilir. Servis (klinik, laboratuvar vb.) yoğunluklarını tespit edilerek servis işlem süresi azaltılabilir.

Algoritmalar karşılaştırıldığında, yön değişiminin geliştirdiğimiz algoritmalarda çoğu zaman daha az sayıda bulunduğu gözlenmiştir. Ancak minTime parametresi değiştirildiğinde bazı değerlerde DB-SMoT algoritmasının geliştirdiğimiz bulanık kümeleme tabanlı algoritmalarından daha iyi sonuç üretmiştir. Yörünge noktası sayısının geliştirdiğimiz algoritmalarda çoğu zaman daha çok sayıda bulunduğu gözlenmiştir. Ancak minTime parametresi değiştirildiğinde bazı değerlerde DB-SMoT algoritmasının geliştirdiğimiz bulanık kümeleme tabanlı algoritmalarından daha iyi sonuç üretmiştir. Çalışma zamanı karşılaştırıldığında geliştirdiğimiz algoritmalarda daha az olduğu gözlenmiştir.

Geliştirdiğimiz algoritmalar daha çok Durma bulup, daha az BilinmeyenDurma bulmuştur. Bu anlamda bulanık kümeleme tabanlı algoritmalarımız daha iyi performans göstermiştir.

FKM-DBSMoT algoritması daha çok Durma ve daha az BilinmeyenDurma bulmuştur. Bu yüzden algoritmalar arasında en iyi performansı FKM-DBSMoT algoritması göstermiştir. FM-DBSMoT algoritması daha az Durma ve daha çok BilinmeyenDurma bulmuştur. Bu yüzden algoritmalar arasında en kötü performansı FM-DBSMoT algoritması göstermiştir. FCM-DBSMoT ile FA-DBSMoT algoritmaları benzer sonuçlar üretmiştir. Algoritmalarımız çalışma zamanları kıyaslandığında en hızlı çalışan algoritma ise FCM-DBSMoT algoritmasıdır.

## KAYNAKLAR

- Agrawal, R., Faloutsos, C., Swami, A.N., 1993. Efficient Similarity Search In sequence Databases. 4th International Conference of Foundations of Data Organization and Algorithms (FODO), Springer Verlag, Chicago, 69-84.
- Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P., 1998. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. ACM-SIGMOD Management of Data, 94-105.
- Agrawal, R., Srikant, R., 1994. Fast Algorithms for Mining Association Rules. VLDB Conf., 487-499
- Agrawal, R., Srikant, R., 2000. Privacy Preserving Data Mining. ACM-SIGMOD, 439-450.
- Akın, Y.K., 2008. Veri Madenciliğinde Kümeleme Algoritmaları ve Kümeleme Analizi, Doktora Tezi, Marmara Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul.
- Akpınar, H., 2000. Veritabanlarında Bilgi Keşfi ve Veri Madenciliği. İ.Ü. İşletme Fakültesi Dergisi,1-22.
- Alvares, L.O., Bogorny, V., Kuijpers, B., Macedo, J.A.F., Moelans, B., Vaisman, A., 2007. A model of Enriching Trajectories with Semantic Geographical Information. ACM 15th International Symposium on Advances in Geographic Information Systems (ACM-GIS'07) Washington, 162-169.
- Alvares, L.O., Bogorny, V., Palma, A., Kuijpers, B., Moelans, B., Macedo, J.A.F., 2007. Towards Semantic Trajectory Knowledge Discovery.
- Andrienko, G., Andrienko, N., 2008. Spatio-temporal aggregation for visual analysis of movements. VAST 2008, IEEE Computer Society Press, 51-58.
- Andrienko, G., Andrienko, N., 2009. Interactive cluster analysis of diverse types of spatiotemporal data. ACM SIGKDD Explorations, 11(2), 19-28.
- Andrienko, G., Andrienko, N., Rinzivillo, S., Nanni, M., Pedreschi, D., Giannotti, F., 2009. Interactive Visual Clustering of Large Collections of Trajectories. VAST 2009, IEEE Computer Society Press, 3-10.
- Anil, K., Jain, M., Murty, N., Flynn, P.J., 1999. Data Clustering: A Review. ACM Comput. Surv., 31(3), 264-323.
- Ankerst, M., Breunig, M.M., Kriegel, H.P., Santer, J., 1999. OPTICS: Ordering Points to Identify the Clustering Structure. ACM SIGMOD'99 Int. Conf. On Management of Data, Philadelphia, 49-60.
- Aoying, Z., Shuigeng, Z., 2000. Approaches for scaling DBSCAN algorithm to large spatial databases. Journal of Computer Science and Information Systems (JIS), Springer 15(1), 51-70.
- Avcı, U., 2006. Bulanık Kümeleme Algoritmalarının Karşılaştırmalı Analizi ve Bilgisayar Uygulamaları. Yüksek Lisans Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü, İzmir.
- Azem, Z., 2003. A Comprehensive Cluster Validity Framework For Clustering Algorithms. MSc Thesis, The University of Guelph, Canada, 15-19.
- Barkan, U., Horn, D., 2006. Spatiotemporal clustering of synchronized bursting events in neuronal networks. Neurocomputing, 69(10-12) : 1108-1111.

- Beitel, S.E., 2005. Applying Artificial Intelligence Data Mining Tools to the Challenges of Program Evaluation. Ph.D. Thesis, University of Connecticut, Connecticut, USA.
- Ben, A., Yakhini. Z. 1999. Clustering gene Expression Patterns. RECOMB, 33-42.
- Benkert, M., Gudmundsson, Hübner, F., Wolle, T., 2007. Reporting flock patterns. *Computational Geometry: Theory and Applications*, 41(3), 111-125.
- Berkhin, P., 2002. Survey of Clustering Data Mining Techniques, Accrue Software Inc., San Jose, California, USA.
- Berkhin, P., 2004. Survey of Clustering Data Mining Techniques, Citiseer.
- Berndt, D.J., Clifford, J., 1996. Finding patterns in time series: a dynamic programming approach. *Advances in knowledge discovery and data mining*, 229-248.
- Bezdek, R.E.J., 1984. FCM:The fuzzy c-means clustering algorithm., in *Computers & Geosciences*, 1984.
- Birant D., Kut, A., 2007. ST-DBSCAN: An algorithm for clustering spatialtemporal data. *Data Knowl. Eng.*, vol. 60, no. 1, 208–221.
- Bogorny, V., 2010. Spatial and Spatio-Temporal Data Mining, Trajectory Knowledge Discovery. Universidade Federal de Santa Catarina.
- Bogorny, V., Avancini, H., Cesar de Paula, B., Kuplich, C.R., Alvares, L.O., 2011. Weka- STPM: a Software Architecture and Prototype for Semantic Trajectory Data Mining and Visualization. *Transactions in GIS*, 15(2), 227-248.
- Bogorny, V., Palma, A.T., Engel, P.M., Alvares, L.O., 2006. Weka- GDPM: Integrating Classical Data Mining Toolkit to Geographic Information Systems. SBBB Workshop on Data Mining Algorithms and Applications (WAAMD'06) , Brazil, 9-16.
- Bogorny, V., Kuijpers, B., Alvares, L.O., 2008. A Spatio-temporal Data Mining Query Language for Moving Object Trajectories. UFRGS, Brazil Technical Report TR-357, 22.
- Bozkurt, F., Haşiloğlu, A.S., Yağanoğlu, M., Erdoğan, F., 2011. Sağlık hizmetlerinde Mekansal-Zamansal Veri Madenciliği Üzerine Algoritma Yaklaşımı. *Tıp Teknolojileri Ulusal Kongresi*, 67-70.
- Chan, K.P., Chee, F.A.W., 1999. Efficient time series matching by wavelets. *ICDE*, 126-133.
- Chang, C.T., Lai, J.Z.C, Jeng, M.D., 2011. A Fuzzy K-means Clustering Algorithm Using Cluster Center Displacement. *J. Inf. Sci. Eng.* , 27(3), 995-1009.
- Chen, J., Jiang, J., 1998. An event-based approach to spatio-temporal data modeling in land subdivision systems. *Geoinformatica*, 2, 387-402.
- Chen, J., Lai, C., Meng, X., Xu, J., Hu, H., 2007. Clustering Moving Objects in Spatial Networks. *Lecture Notes in Computer Science, DASFAA'07*, 611-623.
- Chen, J., Lai, C., Meng, X., Xu, J., Hu, H., 2007. Clustering Moving Objects in Spatial Networks. *Lecture Notes in Computer Science*, 4443.
- Chen, L., Özsu, M.T., Oria, V., 2005. Robust and fast similarity search for moving object trajectories. *SIGMOD '05, ACM SIGMOD international conference on Management of data*, New York, USA, 491-502.
- Chu, S.C., 2004. Improved Clustering and Soft Computing Algorithms. Thesis (Ph.D.) – Flinders University of South Australia, School of Informatics and Engineering.
- Claramunt, C., Theriault, M., 1995. Managing time in GIS: An event-oriented approach. *Recent advances in temporal databases*, Berlin:Springer-Verlag, 23-42.

- Cox, E., 2004. *Fuzzy Modeling and Genetic Algorithms for Data Mining and Explorations*. Morgan Kaufmann, San Francisco.
- Davidson, I, 2002. *Understanding K-means Non-hierarchical Clustering*. Technical Report, Computer Science Department of State University of New York (SUNY)–Albany.
- Dempster, A., Laird, N.M., Rubin, D.B., 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society Agglomerative, Series B*, 39(1), 1-38.
- Dinçer E., 2006. *Veri Madenciliginde K-means Algoritması ve Tıp alanında uygulanması*. Yüksek Lisans Tezi, Kocaeli Üniversitesi FenBilimleri Enstitüsü.
- Dunham, M.H., 2002. *Data Mining Introductory and Advanced Topics*. Prentice-Hall.
- Egenhofer, M., 1994. Spatial SQL:A query and presentation language. *IEEE Transactions on Knowledge and Data Engineering*, 6, 86-95.
- Elnekave, S., Last, M., Maimon, O., Ben-Shimol, Y., Einsiedler, H., Friedman, M., Siebert, M., 2008. Discovering Regular Groups of Mobile Objects Using Incremental Clustering. *5th Workshop on Positioning, Navigation and Communication WPNC 2008*.
- Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *Second International Conference on Knowledge Discovery and Data Mining*, 236-231.
- Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1998. Clustering for mining in large spatial databases. *KI-Journal*, 12(1), 18-24.
- Feyyad, U.M., Piatetsky S.G., Smyth, P., 1996. From data mining to knowledge discovery: an Overview. *Advances in knowledge Discovery and Data Mining*, Cambridge, 1-34.
- Fisher, D., 1987. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2 (2) , 139–172.
- Fiske, J., 1998. *Introduction to Communication Studies*. Routledge, 188.
- Forgy, E., 1965. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometry*, 21(785).
- Fosca, G., Dino, P. 2008. *Mobility, Data Mining and Privacy, Geographic Knowledge Discovery*, Springer.
- Gaffney, S., Smyth, P., 1999. Trajectory Clustering with mixtures of regression models. *ACM SIGKDD, International Conference on Knowledge discovery and data mining*, New York, 63-72.
- Gennady, A., Donato, M., Michael, M., Maguelonne, T., 2006. Mining spatio-temporal data. *Journal of Intelligent Information Systems*, 27(3), 187-190.
- Giannotti, F., Nanni, M., Pinelli, F., Pedreschi, D., 2007. Trajectory pattern mining. *13th ACM SIGKDD international conference on Knowledge discovery and mining*, ACM, 330-339.
- Guha, S., Rastogi, R., Shim, K., 1998. CURE: an efficient clustering algorithms for large databases, in: *Proceeding ACM SIGMOD International Conference on Management of Data*, Seattle, 73–84.
- Hagerstrand, T., 1970. What about people in regional science. *Paper of the Regional Science Association*, 14, 7-21.

- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1), 10-18.
- Han, J., Koperski, K., Stefanovic, N., 1997. Geominer: A System Prototype for Spatial Data Mining. *SIGMOD*, 553-556.
- Han, J., Dong, G., Yin, Y., 1999. Efficient Mining of Partial Periodic Patterns in Time Series Databases. *Conference on Data Engineering*, 106-115.
- Han, J., Kamber, M., 2000. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 800, San Francisco.
- Han, J., Kamber, M., 2003. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 800, San Francisco.
- Hazin, H.G., Havin, F.H.V., Travassos, P.E. P.F., Carvalho, F.C.D., Erzini, K., 2006. Fishing strategy and target species of the brazilian tuna longline fishery, from 1978 to 2005, inferred from cluster analysis. *Collective Volume of Scientific Papers, International Commission for the Conservation of Atlantic Tunas*, vol. 56, 1942-1951.
- Hinneburg, A., Keim D.A., 1998. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. *KDD'98, ABD*, 58-65.
- Hinneburg, A., Keim, D.A., 1998. An efficient approach to clustering in large multimedia database with noise. *4<sup>th</sup> International Conference on Knowledge Discovery and Data Mining, New York City*, 58-65.
- Hoffman, F.M., Hargrove, W.W., Mills, R.T., Mahajan, S., Erickson, D.J., Oglesby, R.J., 2008. Multivariate Spatio-Temporal Clustering (MSTC) as a Data Mining Tool for Environmental Applications. *International Congress on Environmental Modelling and Software(IEMSs 2008)*, 1039-1046.
- Hornsby, K., Egenhofer, M.J., 2000. Identity-based change: A foundation for spatio-temporal knowledge representation. *Geographical Information Science*, 14(3), 207-224.
- Hwang, S.Y., Liu, Y.H., Chiu, J.K., Lim, E.P., 2005. Mining mobile group patterns: A trajectory-based approach. In: *PAKDD*, 713-718.
- İnan, A., Saygın, Y., 2006. Privacy Preserving Spatio-Temporal Clustering on Horizontally Partitioned Data. *DAWAK 2006*, 459-468.
- İnan, A., Saygın, Y., Savaş, E., Hintoglu, A.A., Levi, A., 2006. Privacy Preserving Clustering on Horizontally Partitioned Data. *ICDE Workshops*, 95.
- İşler, Y., Narin, A., 2012. WEKA yazılımında k-Ortalama Algoritması Kullanılarak Kojestif Yetmezliği Hastalarının Teşhisi. *SDU Teknik Bilimler Dergisi*, 2(4), 21-29.
- Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data Clustering: A Review. *ACM Computing Surveys*, Vol. 31, No. 3.
- Jensen, C., Lin, D., Ooi, B., 2007. Continuous Clustering of Moving Objects. *IEEE Trans. On Knowledge and Data Engineering*, 19(9), 1161-1174.
- Jeung, H., Yiu, M.L., Zhou, X., Jensen, C.S., Shen H.T., 2008. Discovery of convoys in trajectory databases. *VLDB Endowment*, 1(1), 1068-1080.
- Jha, S., Kruger, L., McDaniel, P., 2005. Privacy Preserving Clustering. *European Symposium on Research in Computer Security*, 397-417.
- Kalnis, P., Mamoulis, N., Bakiras, S., 2005. On Discovering moving clusters in spatio-temporal data. *Advances in Spatial and Temporal Databases*, 364-381.



- Kalnis, P., Mamoulis, N., Bakiras, S., 2005. On Discovering Moving Clusters in Spatio-temporal Data. *SSTD 2005, LNSC 3633*, 364-381.
- Kang, J., Welbourne, W., Stewart, B., Borriello, G., 2004. Extracting places from traces of locations. *WMASH'04: 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, ACM, New York, 110-118.
- Kang, J., Yong, H.S., 2009. Mining Trajectory Patterns by Incorporating Temporal Properties. *1st International Conference on Emerging Databases (EDB2009)*.
- Kang, J.H., Welbourne, W., Stewart, B., Borriello, G., 2004. Extracting places from traces of locations. *2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, ACM, New York, NY, USA, 110-118.
- Karypis, G., Han, E., Kumar, V., 1999. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer: Special Issue on Data Analysis and Mining*, 32(8), 68-75.
- Kaufman, L., Rousseeuw P.J., 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons Inc., New York, USA.
- Kaya, E., Bulun, M., Arslan A., 1996. *Tıpta Veri Ambarları Oluşturma ve Veri Madenciliği Uygulamaları*.
- Kim, D.O., Kang, H.K., Hong, D.S., Yun, J.K., Han, K.J., 2006. STMPE: An Efficient Movement Pattern Extraction Algorithm for Spatio-temporal Data Mining. *ICCSA (2) 2006*, 259-269.
- Kisilevich, S., Mansmann, F., Nanni, M., Rinzivillo S., 2010. Spatio-Temporal Clustering: a Survey. *Data Mining and Knowledge Discovery Handbook*, 885-874.
- Koperski, K., Adhikary, J., Han, J., 1996. Spatial Data Mining: Progress and challenges. *ACM-SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, Canada, 55-70.
- Koperski, K., Han, J., Adhikary, J., 1998. Mining Knowledge in Geographical Data, *Communications of ACM*, 1998.
- Kulldorff, M., 1997. A Spatial Scan Statistic. *Communications in Statistics Theory and Methods*, 26(6), 1481-1496.
- Langran, G., 1992. *Time in geographic information systems*. London: Taylor&Francis.
- Lee, J.G., Han, J., Whang, K.Y., 2007. Trajectory clustering: a partition-and-group framework. In: *SIGMOD Conference*, 593-604.
- Li, Y., Han, J., Yang, J., 2004. Clustering Moving Objects. *10th ACM SIGKDD International Conference on Knowledge discovery and data mining*, 617-622.
- Liu, Z., George, R., 2003. Fuzzy Cluster Analysis of Spatio-Temporal Data. *ISCIS*, 984-991.
- MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations, in: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281-297.
- Malerba, D., Aprice, A., Ceci, M., 2004. A Data Mining Query Language for Knowledge Discovery in a Geographical Information System. *Database Support for Data Mining Applications*, 95-116.
- Mamoulis, N., Cao, H., Kollios, G., Hadjieleftheriou, M., Tao, Y., Cheung, D., 2004. Mining, Indexing, and Querying Historical Spatiotemporal Data. *10th Knowledge Discovery and Data Mining (KDD)*, 236-245.

- Manish, M., Rakesh, A., Rissanen, J., 1996. SLIQ: A Fast Scalable Classifier for Data Mining. EDBT, Fransa, 18-32.
- Merugu, S., Ghosh, J., 2003. Privacy Preserving Distributed Clustering using Generative Models. ICDM, 211-218.
- Miller, H.J., Han, J., 2001. Geographic data mining and knowledge discovery: an overview. Geographic data mining and knowledge discovery, London, Taylor&Francis, 3-32.
- Moertini, V.S., 2002. Introduction To Five Clustering Algorithms. Integral, Vol. 7, No. 2.
- Nanni, M., Pedreschi, D., 2006. Time-focused clustering of trajectories of moving objects. Journal of Intelligent Information Systems, 27(3), 267-289.
- Nehme, R., Rundensteiner, E., 2006. SCUBA: Scalable Cluster-Based Algorithm for Evaluating Continuous Spatio-temporal Queries on Moving Objects. 10th International Conference on Extending Databases Technology, Munich, Germany, 3896, 1001-1019.
- Neill, D.B., Moore, A.W., Cooper, G.F., 2005. A Bayesian Spatial Scan Statistic. Advances in Neural Information Processing Systems, 18, 1003-1010.
- Ng, R.T., Han, J., 1994. Efficient and effective clustering methods for spatial data mining, in: Proceedings of 20th International Conference on Very Large Data Bases, Santiago, Chile, 144-155.
- Oliveira, S.R.M., Zaiane, O.R., 2004. Privacy Preserving Clustering by Object Similarity-Based Representation. ICDM Workshop on Privacy and Security Aspects of Data Mining. 40-46.
- Özçınar, H., 2006. KPSS Sonuçlarının Veri Madenciliği Yöntemleriyle Tahmin Edilmesi. Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Denizli.
- Özkan, Y., 2008. Veri Madenciliği Yöntemleri. Papatya yayıncılık, İstanbul, 216.
- Palma, A.T., Bogorny, V., Kuijpers, B., Alvares, L.O., 2008. A Clustering-based Approach for Discovering Interesting Places in Trajectories. In: 23rd Annual Symposium on Applied Computing, (ACM-SAC'08), Fortaleza, Ceara, 16-20 March (2008) Brazil. 863-868.
- Patterson, D., Liu, F., Turner, D., Concepcion, A., Lynch, R., 2008. Performance Comparison of the Data Reduction System. SPIE Symposium on Defense and Security, Orlando.
- Pelekis, N., Kopanakis, I., Kotsifakos, E.E., Frenzos, E., Theodoridis, Y., 2009. Clustering Trajectories of Moving Objects in an Uncertain World. ICDEM'09, 417-427.
- Pelekis, N., Kopanakis, I., Marketos, G., Ntoutsi, I., Andrienko, G., Theodoridis, Y., 2007. Similarity search in trajectory databases. 14th International Symposium on Temporal Representation and Reasoning, IEEE Computer Society, Washington, USA, 129-140.
- Pelekis, N., Kopanakis, I., Kotsifakos, E.E., Frenzos, E., Theodoridis, Y., 2009. Clustering Trajectories of Moving Objects in an Uncertain World. ICDM , 417-427.
- Peng, W.C., Chen, M.S., 2003. Developing Data Allocation Schemes by Incremental Mining of User Moving patterns in a Mobile Computing System. IEEE Transactions on Knowledge and Data Engineering, 15(1), 70-85.

- Peuquet, D.J., Duan, N., 1995. An event-based spatiotemporal data model(ESTDM) for temporal analysis of geographical data. *International journal of Geographic Information systems*, 9, 7-24.
- Que, J., Tsui, F.C., Espino, J., 2008. A Z-Score Based Multi-level Spatial Clustering Algorithm for the Detection of Disease Outbreaks. *BioSecure 2008, LNCS 5354*, 106-118.
- Rahim, Y.B.A., 2007. *Spatio-Temporal Data Mining On Moving Objects in DBMS*. GIS Section OTB Research Institute for Housing, Urban and Mobility Studies, 1-33.
- Reades, J., Calabrese, F., Sevtsuk, A., Ratti, C., 2007. Cellular census: Explorations in urban data collection. *IEEE Pervasive Computing*, 6(3), 30-38.
- Rinzivillo, S., Pedreschi, D., Nanni, M., Giannotti, F., Andrienko, N., Andrienko, G., 2008. Visually driven analysis of movement data by progressive clustering. *Information Visualization*, 7(3), 225-239.
- Rissanen, J., 1989. *Stochastic Complexity in Statistical Inference*. World Scientific Publication, 250.
- Rocha, J.A., Times, V.C., Oliveira, G., Alvares, L.O., Bogorny, V., 2010. DB-SMoT: A direction-based spatio-temporal clustering method. *IEEE Conf. of Intelligent Systems*, 114-119.
- Roddick, J.F., Lees, B.G., 2001. *Paradigms for spatial and spatio-temporal data mining*. Geographic Data Mining and Knowledge Discovery, London: Taylor&Francis.
- Roddick, J.F., Spiliopoulou, M., 2002. A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and data engineering*, 14(4), 750-767.
- Sabzi, A., Farjami, Y., Zihayat, M., 2011. An improved fuzzy k-medoids clustering algorithm with optimized number of clusters. *HIS 2011*, 206-210.
- Saygın, Y., Verykios, V.S., Clifton, C., 2001. Using Unknowns to Prevent Discovery of Association Rules. *SIGMOD*, 30(4), 45-54.
- Schraml, S., Belbachir, A.N., 2010. *A Spatio-temporal Clustering Method Using Real-time Motion Analysis on Event-based 3D Vision*. *Computer Vision and Pattern Recognition*.
- Shekhar, S., Huang, Y., Wu, W., Lu, C.T., Chawia, S., 2001. What's spatial about spatial data mining: three case studies. *Data Mining for Scientific and Engineering Applications*.
- Silahtaroglu, G., 2008. *Kavram ve Algoritmalarıyla Temel Veri Madenciliği*. Papatya Yayıncılık, 172, İstanbul.
- Spaccapietra, S., Parent, C., Damiani, M.L., Macedo, A.D., Porto, F., Vangenot, C., 2008. A conceptual view on trajectories. *Data and Knowledge Engineering*, 65(1), 126-146.
- Spieth, C., Streichert, F., Speer, N., Zell, A., 2005. Clustering based approach to identify solutions for the inference of regulatory networks. *Congress on Evolutionary Computation 2005*, 660-667.
- Şentürk, A., 2006. *Giriş ve Temel Kavramlar*. Veri Madenciliği Kavram ve Teknikler, Ekin basım yayın dağıtım, Bursa, Türkiye, 1-10.
- Theodoridis, Y., 2003. Ten benchmark databases queries for location-based services. *The Computer Journal*, 46(6), 713-725.

- Tsoukatos, E., Gunopoulos, D., 2001. Efficient Mining of Spatio-Temporal Patterns. ACM Symposium on Spatial and Temporal Databases, 214-223.
- Usery, E.L., 1996. A feature-based geographic information system model. *Photogrammetric Engineering & Remote Sensing*, 62, 833-838.
- Vaidya, J., Clifton, C., 2003. Privacy Preserving K-Means Clustering over Vertically Partitioned Data. ACM SIGKDD, 206-215.
- Vieira, M.R., Bakalov, P., Tsotras, V.J., 2009. On-line discovery of flock patterns in spatio-temporal data. 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, New York, 286-295.
- Vinod, H., 1969. Integer programming and the theory of grouping, *Journal of the American Statistical Association* 64 (326) , 506-519.
- Vlachos, M., Kollios, G., Gunopoulos, D., 2002. Discovering similar multidimensional trajectories. International Conference on Data Engineering, 673-684.
- Vlachos, M., Kollios, G., Gunopoulos, D., 2002. Discovering similar multidimensional trajectories. ICDE '02 Proceedings of the 18th International Conference on Data Engineering, 673.
- Wachowicz, M., 1999. Object-oriented design for temporal GIS. London:Taylor&Francis, 136.
- Wang, D., Cheng, T., 2001. A spatio-temporal data model for activity-based transport demand modeling. *International Journal of Geographic Information Science*, 15, 561-585.
- Wang, M., Wang, A., Li, A., 2006. Mining Spatial-temporal Clusters from Geodatabases. *Lecture notes in Computer Science*, 4093(2006), 263-270.
- Wang, W., Yang, J., Muntz, R., 1997. STING: a statistical information grid approach to spatial data mining. 23rd International Conference on Very Large Data Bases (VLDB), 186-195.
- Wen, J.R., Nie, J.Y., Zhang, H.J., 2002. Query clustering using user logs. *ACM Transactions on Information Systems*, 20(1), 59-81.
- Witten, I.H., Frank, E., Hall, M.A., 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, London.
- Xia, J., Ciesielski V., Arrowsmith C., 2005. Data Mining of Tourists' Spatio-temporal Movement Patterns: A Case Study on Phillip Island, *GeoComputation*.
- Xiaowei, X., Ester, M., Kriegel, H. Sander, J., 1998. A Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases. ICDE, 14, 324-331.
- Yağanoğlu, M., Haşiloğlu, A.S., Bozkurt, F., Erdoğan, F., 2011. Sağlık Sektöründe Radyo Frekanslı Tanımlama (RFID): Mekansal-Zamansal Veri Madenciliği Veri Tiplerine Bir Yaklaşım. *Tıp Teknolojileri Ulusal Kongresi*, 292-295.
- Yang, J., Wang, W., Yu, P.S., 2000. Mining Asynchronous Periodic Patterns in Time Series Data. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 275-279.
- Yao, X., *Research Issues in Spatio-Temporal Data Mining*. 2003. *Geospatial Visualization and Knowledge Discovery*.18-20.
- Yıldız, K., Çamurcu, Y., Doğan, B., 2010. Veri Madenciliğinde Temel Bileşenler Analizi ve Negatifsiz Matris Çarpanlarına Ayırma Tekniklerinin Karşılaştırmalı Analizi. *Akademik Bilişim Konferansı, AB2010, Muğla*.

- Yohannes, Y., Webb, P., 1999. Classification and Regression Trees, CART: A user Manuel For Identifying Indicators of Vulnerability to Famine and Chronic Food Insecurity. International Food Policy Research Institute, U.S.A., 1-48.
- Yuan, M., 1997. Use of knowledge acquisition to build wildfire representation in geographical information systems. International Journal of Geographic Information Science, 11, 723-745.
- Zadeh, L.A. 1965. Fuzzy sets. Information Control, 8, 1965.
- Zhang, J., Tsui, F.C., Wagner, M.M., Hogan, W.R., 2003. Detection of Outbreaks from Time Series Data Using Wavelet Transform. In: AMIA Annual Symposium Proceeding, 748-752.
- Zhang, T., Ramakrishnan, R., Livny, M., 1996. BIRCH: An Efficient Data Clustering Method for Very Large Databases. ACM International Conference on Management of Data, 103-114.
- Zheng, Y., Zhang, L., Xie, X., Ma, WY., 2009. Mining interesting locations and travel sequences from gps trajectories. WWW'09: 18th international conference on World wide web, 791-800.
- Zhou, C., Frankowski, D., Ludford, P.J., Shekhar, S., Terveen, L. G., 2007. Discovering personally meaningful places: An interactive clustering approach. ACM Trans. Inf. Syst., 25(3), Article No. 12.

## ÖZGEÇMİŞ

1985 yılında Erzurum' da doğdu. İlk ve orta öğrenimini Erzurum' da tamamladı. 2009 yılında Erciyes Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünde lisans eğitimini tamamladı. 2009 yılında Atatürk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümün'de yüksek lisansa başladı. Halen Atatürk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünde araştırma görevlisi olarak görev yapmaktadır.