
GAZİ ÜNİVERSİTESİ
Gazi
1926

AYKIRI VERİ YÖNELİMLİ FAYDA TEMELLİ BÜYÜK VERİ ANONİMLEŞTİRME MODELİ

Yavuz CANBAY

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AĞUSTOS 2019

Yavuz CANBAY tarafından hazırlanan "AYKIRI VERİ YÖNELİMLİ FAYDA TEMELLİ BÜYÜK VERİ ANONİMLEŞTİRME MODELİ" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Ana Bilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Prof. Dr. Şeref SAĞIROĞLU

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum .....

İkinci Danışman: Dr. Yılmaz VURAL

Bilgisayar Mühendisliği Ana Bilim Dalı, KVKK

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum .....

Başkan: Prof. Dr. Kemal BIÇAKCI

Bilgisayar Mühendisliği Ana Bilim Dalı, TOBB ETÜ

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum .....

Üye: Prof. Dr. Suat ÖZDEMİR

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum .....

Üye: Dr. Öğr. Üyesi Murat AYDOS

Bilgisayar Mühendisliği Ana Bilim Dalı, Hacettepe Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum .....

Üye: Dr. Öğr. Üyesi Bülent TUĞRUL

Bilgisayar Mühendisliği Ana Bilim Dalı, Ankara Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum .....

Üye: Dr. Öğr. Üyesi Uraz YAVANOĞLU

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum .....

Tez Savunma Tarihi: 28/08/2019

Jüri tarafından kabul edilen bu tezin Doktora Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Prof. Dr. Sena YAŞYERLİ

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Yavuz CANBAY

28/08/2019

AYKIRI VERİ YÖNELİMLİ FAYDA TEMELLİ BÜYÜK VERİ ANONİMLEŞTİRME MODELİ

(Doktora Tezi)

Yavuz CANBAY

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Ağustos 2019

ÖZET

Veri mahremiyeti, mahremiyet seviyesi ile veri faydası arasındaki en iyi dengeyi bulmaya çalışan, zor ve güncel bir problemdir. Her ne kadar ilk bakışta veri sahiplerinin mahremiyetini korumak olarak anlaşılsa da, sadece bununla sınırlı olmayıp verinin fayda boyutunu da veri mahremiyeti koruma sürecine dâhil eder. Veri faydası, veri mahremiyeti sürecindeki en önemli unsurlardan biri olup, mahremiyeti korunmuş veri üzerinde yapılacak analizlerin ve geliştirilen modellerin doğruluğunu doğrudan etkiler. Veri mahremiyeti kapsamında, toplam veri faydasını düşüren veri grubu olarak tanımlanan aykırı verilerin mahremiyet koruma sürecinde yönetilmesi gerekir. Literatürde veri mahremiyeti kapsamında aykırı verileri dikkate alan ve bunları yöneten çeşitli çalışmalar mevcuttur. Ancak bu çalışmalar, aykırı verileri kısmen veya tamamen veri kümesinden çıkardığı veya aykırı verilerin değerini değiştirdiği için hem veri faydası hem de veri güvenilirliği açısından yeterli çözüm sunamamaktadır. Bu tezde, aykırı verileri yöneterek toplam veri faydasını arttıran geleneksel mimari tabanlı iki yeni anonimleştirme modeli (u-Mondrian ve u-Canon), Mondrian modelinden daha üstün yeni bir anonimleştirme modeli (Canon) ve büyük veri mimarisinde SMondrian modeline aykırı veri konsepti uygulayarak daha yüksek veri faydası sunan yeni bir anonimleştirme modeli (Su-Mondrian) ilk defa önerilmiş, geliştirilmiş, uygulanmış ve test edilmiştir. Elde edilen test sonuçlarına göre; DM, GCP ve AECS metrikleri için u-Mondrian modelinin Mondrian modeline göre sırasıyla %15,30-%49,75, %16,02-%44,50 ve %13,76-%48,98 aralıklarında daha yüksek veri faydası sunduğu; u-Canon modelinin Canon modeline göre ise sırasıyla %15,30-%49,08, %5,18-%32,43 ve %13,76-%48,99 aralıklarında daha yüksek veri faydası sunduğu, Canon modelinin Mondrian modeline göre GCP metriği için %43,01-%45,47 aralığında daha yüksek veri faydası sunduğu ve son olarak Su-Mondrian modelinin SMondrian modeline göre DM, GCP ve AECS metrikleri için sırasıyla %25,55-%33,12, %22,83-%29,16 ve %9,29-%17,29 aralıklarında daha yüksek veri faydası sunduğu görülmüştür.

Bilim Kodu	:	92414
Anahtar Kelimeler	:	Veri mahremiyeti, fayda temelli model, büyük veri, anonimleştirme, mondrian, u-mondrian, canon, u-canon, su-mondrian
Sayfa Adedi	:	182
Danışman	:	Prof. Dr. Şeref SAĞIROĞLU
İkinci Danışman	:	Dr. Yılmaz VURAL

OUTLIER ORIENTED UTILITY BASED BIG DATA ANONYMIZATION MODEL

(Ph. D. Thesis)

Yavuz CANBAY

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

August 2019

ABSTRACT

Data privacy is a difficult tradeoff problem between privacy and utility. Although it is understood as preserving the privacy of data owners at first glance, it has the utility dimension of data in privacy preserving processes. Data utility directly affects the accuracy of the analysis and models which are made and developed on the privacy preserved data. In the context of data privacy, outliers are defined as the data group that reduces total data utility and they need to be managed in the privacy preserving processes. In the literature, there exist various studies which focus on outliers and outlier management. Because these studies remove outliers partially or completely from the dataset or change the real values of outliers, they do not present sufficient solutions in terms of data utility and data reliability. In this thesis, two traditional architecture based anonymization models (u-Mondrian and u-Canon) which propose to increase total data utility by managing outliers, a new anonymization model (Canon) which is better than Mondrian and a new big data based anonymization model (Su-Mondrian) which manages outliers and presents higher data utility than SMondrian were proposed, developed, applied and tested. According to the experimental results, for DM, GCP and AECS metrics, it was seen that u-Mondrian presents higher data utility than Mondrian in the ranges of %15.30-%49.75, %16.02-%44.50 and %13.76-%48.98; u-Canon presents higher data utility than Canon in the ranges of %15.30-%49.08, %5.18-%32.43 and %13.76-%48.99 respectively; Canon presents higher data utility than Mondrian in the range of %43.01-%45.47 for GCP metric and finally Su-Mondrian presents higher data utility than SMondrian in the ranges of %25.55-%33.12, %22.83-%29.16 and %9.29-%17.29 for DM, GCP and AECS metrics respectively.

Science Code	:	92414
Key Words	:	Data privacy, utility based model, big data, anonymization, mondrian, u-mondrian, canon, u-canon, su-mondrian
Page Number	:	182
Supervisor	:	Prof. Dr. Şeref SAĞIROĞLU
Co-Supervisor	:	Dr. Yılmaz VURAL

TEŞEKKÜR

Bu tez çalışmamda büyük emekleri olan kıymetli danışmanlarım Prof. Dr. Şeref SAĞIROĞLU ve Dr. Yılmaz VURAL'a, tez izleme komitemde bulunan Prof. Dr. Kemal BIÇAKCI ve Prof. Dr. Suat ÖZDEMİR hocalarıma, hayatımın her anında benden desteğini esirgemeyen sevgili eşim Pelin CANBAY'a, mutluluk kaynağım olan oğlum Cihan Mert CANBAY'a ve bu günlere gelmemde büyük emeği olan değerli aileme çok teşekkür ederim. Ayrıca, Gazi Üniversitesi Büyük Veri ve Bilgi Güvenliği Laboratuvarı (BIDISEC) ve Gazi Üniversitesi Bilimsel Araştırmalar Projeleri birimine çalışmalarımı desteklediği için şükranlarımı sunarım.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge **Sayfa**

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

Şekil **Sayfa**

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
AECS	Average Equivalence Class Size
BUG	Bottuom-Up Generalization
CANON	Cihan Anonimleştirme
DM	Discernibility Metric
DM_T	Total Discernibility Metric
ES	Eşlenik Sınıf Sayısı
ES_T	Toplam Eşlenik Sınıf Sayısı
GCP	Generalized Certainity Penalty
GCP_T	Total Generalized Certainity Penalty
KD	K-Dimensional
LRD	Local Reachability Distance
LOF	Local Outlier Factor
N	Normal Veri
NCP	Normalized Certainity Penalty
NDS	Normal Data Set
O	Aykırı Veri
ODS	Outlier Data Set
U-CANON	Utility-aware Canon
U-MONDRIAN	Utility-aware Mondrian
SMONDRIAN	Spark-based Mondrian
SU-MONDRIAN	Spark-based utility-aware Mondrian
TDS	Top-Down Specification
VP	Vantage Point

1. GİRİŞ

Bu bölümde tezde ele alınan mevcut problem hakkında bilgi verilmiş, tezin amacı, kapsamı, literatüre katkısı, önemi ve yapısı alt başlıklarda sunulmuştur.

Elektronik bilgi toplumunun hızla gelişmesiyle beraber, sağlık, eğitim, finans, nüfus, yönetim gibi pek çok alanda üretilen verinin miktarı ve boyutu artmaktadır. Bu duruma paralel olarak veriden değer üretmek için işlenmesiyle beraber paylaşılması da gün geçtikçe artan bir ihtiyaç haline gelmektedir. Paylaşılan veriler içerisinde; sağlık verileri, ticari veriler, demografik veriler, konum bilgileri, maaş bilgileri, epostalar, fotoğraflar ve videolar gibi kişisel ve hassas bilgileri içerebilir [1].

Günümüzde pek çok kurum (veri sorumlusu) hizmet verdiği muhataplarına (müşteri, hasta, kullanıcı, firma vb.) ait çeşitli verileri toplamakta ve depolamaktadır. Veri sorumluları, görevlerini yerine getirmek ve muhataplarına daha iyi hizmet sunmak (model ve örüntü çıkarmak, planlamalar yapmak, politikalar oluşturmak, karar verme mekanizmaları geliştirmek vb.) amacıyla topladıkları bu verileri işlemektedir. İşlenen verilerden yüksek seviyede değer üretilmesi için verinin çeşitli kişi, kurum ve kuruluşlarla paylaşılması gerekir. Ancak bu veriler içerisinde bireyi doğrudan veya dolaylı olarak tanımlayabilecek çeşitli kişisel veriler de yer alabileceği için, böylesi bir durumda bu tür verilerin yeterli düzeyde korunması şarttır.

Veri paylaşımındaki en yaygın yöntemlerin başında yayınlama gelmektedir [2]. Yayınlanacak veri kümesi herhangi bir mahremiyet koruyucu önlem alınmadan paylaşılırsa, saldırganlar çeşitli saldırı yöntemlerini kullanarak veri sahiplerinin kimliklerini ifşa edebilir. Mahremiyet koruyucu önlemlerin yüksek seviyede alınması durumunda ise veriden elde edilecek fayda düşük olur [3]. Bu gerekçelerden dolayı veri mahremiyeti ile faydası arasında bir dengenin sağlanması önemlidir.

Veri mahremiyeti, mahremiyet seviyesi ile veri faydası arasındaki en iyi dengeyi bulmaya çalışan zor bir problemdir [4,5]. Veri mahremiyeti her ne kadar veri sahiplerinin mahremiyetinin korunması olarak anlaşılsa da, sadece bununla sınırlı olmayıp verinin fayda sağlama özgürlüğünü de veri mahremiyeti koruma sürecine dâhil eder.

Veri faydası, veri mahremiyeti koruma sürecindeki en önemli unsurlardan biri olup, mahremiyeti korunmuş veri üzerinde yapılacak analizlerin ve geliştirilecek modellerin kalitesini ve doğruluğunu doğrudan etkiler. Veri faydasını etkileyen faktörlerden en önemlileri aşağıda listelenmiştir [6-8];

- Mahremiyet seviyesi,
- Kullanılan algoritmanın optimallik durumu ve
- Aykırı verilerdir.

Mahremiyet seviyesi ile veri faydası arasındaki ilişki ters yönlüdür. Yani mahremiyet seviyesinin yüksek olduğu durumda veri faydası düşük, mahremiyet seviyesinin düşük olduğu durumda ise veri faydası yüksektir. Veri mahremiyetini sağlamak için optimal bir algoritmanın kullanılması her ne kadar yüksek veri faydası sunsa da, veri mahremiyetinin zor bir problem olmasından dolayı verinin büyümesi halinde kabul edilebilir bir zamanda bu probleme çözüm bulunamayacağı bir gerçektir. Aykırı veriler ise fayda odaklı bir yönetim anlayışı olması halinde veri faydasını arttıran önemli bir unsurdur.

Orijinal bir veri kümesinin %100 veri faydası sunduğu kabul edilirken, bu veri üzerinde uygulanacak anonimleştirme işlemi sonrası veride bilgi kaybı meydana gelir. Bu kaybın olabildiği kadar düşük seviyede tutulması halinde ise, anonim veri üzerinde oluşturulacak modellerin ve yapılacak analizlerin doğruluğu da yüksek seviyede sağlamış olur [9].

Veri mahremiyetinde, veri faydasını olumsuz etkileyen veriler aykırı veri olarak tanımlanır [10-12]. Aykırı veriler, yüksek seviyede anonimleştirmeye sebep oldukları için bilgi kaybını arttırmakta ve böylece veri faydasını olumsuz etkilemektedir. Bu sorunu çözmek için aykırı verilerin anonimleşme sürecinde yönetilmesi gerekir. Literatürde aykırı veri yönetiminde kullanılan çeşitli yaklaşımlar mevcuttur. Ancak bu yaklaşımlar, aykırı verileri kısmen veya tamamen veri kümesinden çıkardığı için veri faydası açısından yeterli çözüm sunamamaktadır [7,13,14].

Ayrıca, büyük veri mahremiyetinde de aykırı veriler önemli bir sorun olmasına rağmen, büyük veri alanında aykırı verileri yöneterek veri faydasını arttırmayı amaçlayan herhangi bir yaklaşım ve modelin literatürde geliştirilmediği görülmekte ve bundan dolayı da anonimleştirilmiş büyük verilerin kısıtlı veri faydası sunduğu değerlendirilmektedir.

Bu tezde, aykırı verileri yöneterek tamamından fayda elde edilmesini sağlayan, geleneksel mimari tabanlı iki yeni anonimleştirme modeli (u-Mondrian ve u-Canon), literatürde sıklıkla kullanılan bir anonimleştirme modeli olan Mondrian'dan daha üstün yeni bir anonimleştirme modeli (Canon) ve büyük veride aykırı verileri yöneterek tamamından fayda elde edilmesini sağlayan yeni bir anonimleştirme modeli (Su-Mondrian) olmak üzere toplam 4 farklı model ilk defa önerilmiş, geliştirilmiş, uygulanmış ve test edilmiştir.

Veri mahremiyetinde, mahremiyet seviyesini koruyarak aykırı veri yönetimi ile veri faydasını mümkün olabildiği kadar yüksek seviyeye çıkarmak bu tezin ana amacıdır. Bu kapsamda, tezin diğer amaçları ise aşağıda listelenmiştir;

- Geleneksel mimari üzerinde, aykırı verileri yöneterek bu verilerin tamamından fayda elde edilmesini sağlayan Mondrian tabanlı yeni bir anonimleştirme modeli (u-Mondrian) geliştirmek, uygulamak ve test etmek,
- Geleneksel mimari üzerinde, Mondrian anonimleştirme modeline alternatif ve ondan daha üstün bir model (Canon) geliştirmek, uygulamak ve test etmek,
- Canon anonimleştirme modeline aykırı veri konsepti uygulayarak Canon modelinden daha yüksek veri faydası sunan yeni bir model (u-Canon) geliştirmek, uygulamak ve test etmek ve
- Büyük veri mimarisi üzerinde, aykırı verileri yöneterek tamamından fayda elde etmeyi sağlayan yeni bir anonimleştirme modeli (Su-Mondrian) geliştirmek, uygulamak ve test etmektir.

Bu tez kapsamında, veri mahremiyetinde veri faydasını arttırmak için aykırı veri konsepti [8,13,15] uygulanmıştır. Bu tezin detaylı kapsamı ise aşağıda liste halinde sunulmuştur;

- Mahremiyet koruma modellerinden biri olan k-Anonimlik modeli tercih edilmiştir,
- Yarı-tanımlayıcı olarak sadece sayısal öznitelikler dikkate alınmıştır,
- Mondrian modeli genişletilmek üzere kullanılmıştır,
- Veri faydasını ölçmede sıklıkla tercih edilen DM, GCP ve AECS metrikleri kullanılmıştır ve
- Büyük veri işleme ana çatısı olarak Apache Spark tercih edilmiştir.

Bu tez kapsamında yapılan incelemeler ve gerçekleştirilen çalışmalar doğrultusunda, literatüre sunulan katkılar maddeler halinde aşağıda verilmiştir;

- Mondrian modeline ilk defa aykırı veri konsepti uygulanarak, daha yüksek veri faydası sunan u-Mondrian modeli geliştirilmiş, uygulanmış ve test edilmiştir,
- Veri uzayını parçalamada VP-tree yapısını kullanan ve Mondrian modelinden daha yüksek veri faydası sunan Canon modeli önerilmiş, uygulanmış ve test edilmiştir,
- Canon modeline aykırı veri konsepti uygulanarak, daha yüksek veri faydası sunan u-Canon modeli önerilmiş, uygulanmış ve test edilmiştir,
- Büyük veri mimarisinde aykırı veri konseptini uygulayan ilk model olan Su-Mondrian modeli önerilmiş, uygulanmış ve test edilmiştir,
- Aykırı veri yönetimi ile veri faydası arasındaki ilişkiyi göstermek adına ilk defa kapsamlı karar kuralları oluşturulmuştur,
- Anonimleştirmede, aykırı verilerin ve normal verilerin belirlenmesinde kullanılmak üzere ilk defa hem yoğunluk hem de yakınlık tabanlı yeni hibrit bir yaklaşım önerilmiş, geliştirilmiş ve uygulanmıştır.

Bu tezin yapısı maddeler halinde aşağıda verilmiştir;

- Birinci bölümde, tezde ele alınan problem, tezin amacı, kapsamı, önemi ve literatüre katkısı hakkında bilgiler verilmiştir,
- İkinci bölümde, veri mahremiyeti, mahremiyet koruma süreci ve mahremiyet problemi geleneksel veri ve büyük veri için ayrı ayrı değerlendirilmiş, büyük veri mimarisi ve teknolojileri sunulmuş, veri mahremiyetine yönelik tehditler ve koruma modelleri hakkında bilgi verilmiştir,
- Üçüncü bölümde literatürde sıklıkla kullanılan anonimleştirme modelleri ve algoritmaları verilmiş, bilgi metrikleri açıklanmış, aykırı veri, normal veri ve veri faydası için literatürde yapılan genel tanımlar sunulmuş, veri mahremiyeti kapsamında aykırı verilerin belirlenmesi ve yönetimi açıklanmış, genel bir literatür taraması sunulmuş ve kullanılan veri kümeleri tanıtılmıştır,
- Dördüncü bölümde, tezde önerilen u-Mondrian modeli geliştirilmiş, uygulanmış ve başarı ile test edilmiştir. Bu model için kabul edilen aykırı veri ve normal veri tanımları sunulmuş, veri faydasını değerlendirmede kullanılan karar kuralları ve önerilen modelin içerdiği katmanlar açıklanmıştır. Adult ve Diyabet veri kümeleri kullanılarak önerilen model test edilmiş ve elde edilen sonuçlar sunulmuştur,

- Beşinci bölümde, Mondrian modelinin dezavantajlarını gidermek adına Canon isimli yeni bir anonimleştirme modeli geliştirilmiş, uygulanmış ve başarı ile test edilmiştir. Adult veri kümesi kullanılarak önerilen model için elde edilen sonuçlar sunulmuştur,
- Altıncı bölümde, Canon modeline aykırı veri konsepti uygulayarak veri faydasını arttırmayı amaçlayan u-Canon isimli yeni bir model önerilmiştir. Bu model için kabul edilen aykırı veri ve normal veri tanımları sunulmuş, veri faydasını değerlendirmede kullanılan karar kuralları ve önerilen modelin içerdiği katmanlar açıklanmıştır. Adult veri kümesi kullanılarak önerilen model başarı ile test edilmiş ve elde edilen sonuçlar sunulmuştur,
- Yedinci bölümde, büyük veri mimarisi tabanlı aykırı veri yönelimli fayda temelli anonimleştirme modeli (Su-Mondrian) önerilmiş, uygulanmış ve test edilmiştir. Bu modelde kabul edilen aykırı veri ve normal veri tanımları sunulmuş ve içerdiği katmanlar açıklanmıştır. Adult veri kümesi kullanılarak yapılan deneyde önerilen model başarı ile test edilmiş ve elde edilen sonuçlar sunulmuştur.
- Sekizinci bölümde ise tez çalışmasının sonuçları değerlendirilmiştir.

2. VERİ MAHREMİYETİ

Bu bölümde geleneksel veride ve büyük veride mahremiyet kavramı, mahremiyet koruma süreci ve mahremiyet problemi açıklanmış, mahremiyeti hedef alan tehditler ve mahremiyet koruma modelleri hakkında detaylı bilgi verilmiştir.

2.1. Geleneksel Veri

Geleneksel veri, bilinen klasik sistemlerle işlenebilen ve literatürde normal veri ve küçük veri gibi isimlerle de anılan veri sınıfıdır. Klasik veri tabanlarında tutulabilen ve geleneksel programlama paradigmaları ile işlenebilen her türlü veri bu kapsamda değerlendirilmektedir [16].

2.2. Geleneksel Veride Mahremiyet

İlk olarak "Mahremiyet Hakkı" başlıklı makale [17] ile literatüre giren ve "yalnız bırakılma hakkı" olarak tanımlanan mahremiyet kavramı, günümüzde yasalarla korunan, temel ve zorunlu bir ihtiyaç haline gelmiştir. Mahremiyet; toplumdan topluma, kültürden kültüre, ülkeden ülkeye hatta bireyden bireye değişiklik gösterebilen çok yönlü ve değişken bir kavramdır. Bilişim alanında insanların mahremiyetinin korunması, bireyin kendi sınırlarını belirlediği gerçek ve sanal ortamlarda birey olabilme hakkı olarak tanımlanabilir.

Veri mahremiyeti literatürde, "bilgisel seçici kontrol" [18] ve "muhatapların bilgilerinin doğru kullanımı ve muhatabın hangi bilgisinin kiminle ve ne derecede paylaşılmasına karar verme mekanizması" [19] olarak tanımlanmıştır. Bu tanımlara ek olarak aşağıda sunulan tanımlar da konuyu daha iyi anlamaya yardımcı olacaktır;

- Veri üzerinde uygulanacak herhangi bir metot, teknik veya arka plan bilgileri ile veri sahiplerinin kimliklerinin mümkün olduğu ölçüde ifşa olmasının engellenmesi,
- Veriden bir ya da daha fazla kişiye doğrudan veya dolaylı olarak mümkün olduğu ölçüde erişilememesi,
- Verinin kiminle, hangi seviyede ve ne amaçla paylaşılacağına dair sınırların belirlenmesinde veri sahibinin seçici kontrolü veya

- Veriden veri sahibine ulaşmayı kolaylaştıran herhangi bir ilişkinin mümkün olduğu ölçüde ortadan kaldırılmasıdır.

Veri mahremiyeti günümüzde araştırmacılar tarafından çalışılan önemli konulardan biridir. Hassas bilgiler içeren verilerin yayınlanması ile ortaya çıkacak mahremiyet ihlali endişeleri, bu endişelerin olabildiğince giderilmesi adına araştırmacıları bu alana sevk etmiş ve çözüm aranan önemli problemlerden biri haline getirmiştir. Her ne kadar bu endişelerin giderilmesi adına çeşitli çözümler geliştirilse de bu endişelerin tamamen giderilmesi mümkün olamayacağı gibi; gelişen ve ilerleyen teknolojilerle birlikte arka plan bilgilerinin artması, büyük verilerin oluşturulması, zeki sistemlerin geliştirilmesi, veriden farklı çıkarılmaların yapılmasının sağlanması, saldırı şekillerinin değişmesi vb. gibi pek çok sebepten dolayı da mahremiyet ihlali endişelerinin arttığı gözlemlenmektedir [20]. Ancak bu tür endişelerin artması verinin ekonomiye kazandırılması, veriden değer üretilmesi, verinin araştırmacılara ve bilim insanlarına açılması ve veri güdümlü ekonomilerin oluşturulması konusunda bir engel değildir. Çünkü veriden değer üretilmesi toplumların gelişmesine yardımcı bir faktör olurken, aynı zamanda üretilen bu değerler dijitalleşmiş toplumların en önemli çıktısı olarak görülebilir [21].

Veri mahremiyeti ilk bakışta her ne kadar veri sahiplerinin mahremiyetini korumak olarak değerlendirilse de sadece bununla sınırlı olmayıp verinin fayda sağlama özgürlüğünün de veri mahremiyeti koruma sürecine dâhil olduğu bilinmelidir. Veri tabanı sistemlerinde mahremiyet endişelerine bağlı olarak kurumun veya kişinin kendi kullanımına açık olan verinin toplumlara fayda sağlama özgürlüğü kısıtlıdır. Mahremiyeti korunan ve fayda potansiyeline sahip verilerin özgürce paylaşılması, toplumların her alanda ilerlemesine yardımcı olacak önemli bir faktördür.

Veri paylaşımındaki en yaygın yöntemlerin başında veri yayınlama gelmektedir [2,22]. Veri sorumlularının paylaşım amacıyla yayınladıkları veriler içerisinde yer alabilecek hassas bilgiler, bu verilerin yeterli düzeyde korunmasını gerektirir. Yayınlanacak veri kümesi herhangi bir mahremiyet koruyucu önlem alınmadan paylaşılırsa; saldırganlar tarafından farklı saldırı yöntemleri kullanılarak bu verilerin sahipleri ifşa edilebileceği gibi, kanuni yaptırımlara maruz kalınabilir, kişiler bundan zarar görebilir veya istenmeyen olumsuzluklarla da karşılaşılabilir. Gerek veriden elde edilecek potansiyel faydanın ortaya

çıkarılması gerekse de mahremiyet ifşalarının olabildiği kadar engellenmesi amacıyla mahremiyet koruyucu yaklaşımlara her zaman ihtiyaç vardır.

Veri mahremiyetini korumada kullanılan ve en basit yöntem olarak kabul edilen kimliksizleştirme; tekil-tanımlayıcı olarak nitelendirilen ve bireyi doğrudan tanımlayan özniteliklerin (kimlik numarası, pasaport numarası, adı soyadı vb.) yayınlanan veriden çıkarılması olarak tanımlanır. Kimliksizleştirme, her ne kadar mahremiyet koruyucu bir yaklaşım olsa da yeterli seviyede bir koruma sağlayamadığı Sweeney tarafından yapılan araştırmada [23] gösterilmiştir. Sweeney yaptığı bu çalışmada, kimliksiz sağlık verilerini oy verileri ile cinsiyet-posta kodu-doğum tarihi öznitelikleri üzerinde eşleştirerek Amerika nüfusunun %87'sinin kimlik bilgilerinin ifşa edilebileceği göstermiştir. Benzer başka bir olayda ise 2006 yılında AOL firması 650 bin kullanıcısına ait 20 milyon arama sorgusu verisini kullanıcı kimliği ve IP numarası bilgilerini silip kimliksizleştirerek yayınlamış, ancak birkaç gün içerisinde bu sorguların kimlere ait olduğu araştırmacılar tarafından tespit edilmiştir [24,25].

Yukarıda belirtilen örnek durumlar dikkate alındığında, kimliksizleştirme yönteminin veri mahremiyetini tam olarak koruyamadığı ve bundan dolayı mahremiyet koruyucu yeni yaklaşımlara ihtiyaç duyulduğu araştırmacılar tarafından tespit edilmiştir. Bu kapsamda literatürde *k*-Anonimlik [6], *l*-Çeşitlilik [24] ve *t*-Yakınlık [26] gibi daha güçlü mahremiyet koruma modellerinin geliştirildiği görülmektedir.

Veri mahremiyeti, mahremiyet seviyesi ile veri faydası arasındaki bir denge problemidir. Yani veri mahremiyetinin tamamen karşılandığı bir durumda veri faydasından, veri faydasının tamamen karşılandığı bir durumda ise veri mahremiyetinden söz edilemeyeceği için, verinin mahremiyeti ile faydası arasında bir dengenin kurulması önemlidir. Ancak böylesi bir dengeyi sağlamak literatürde zor bir problem olarak karşımıza çıkmaktadır [5,27].

Veri mahremiyetini sağlama, verinin alınmasından mahremiyetinin korunarak yayınlanmasına kadar geçen süreç içerisindeki aşamalardan biridir. Bu kapsamda, veri mahremiyeti koruma sürecini iyi anlamak ve bu süreçteki tarafları ve bu tarafların rollerini iyi değerlendirmek gerekir. Devam eden bölümde geleneksel veride mahremiyet koruma süreci detaylandırılmıştır.

2.2.1. Geleneksel veride mahremiyet koruma süreci

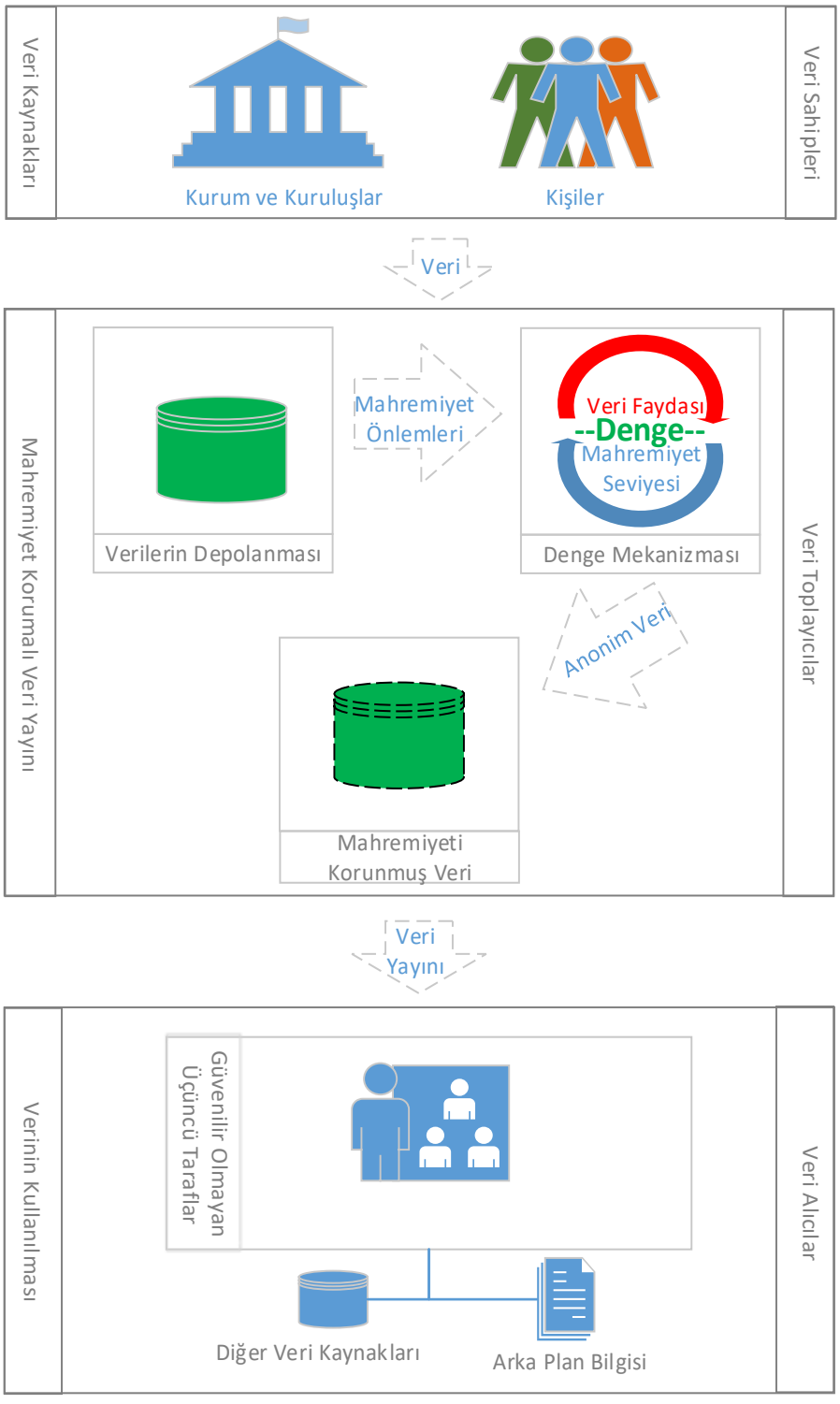
Verinin alınmasından mahremiyetinin korunarak yayınlanmasına kadar geçen süreç, bir veri mahremiyeti koruma sürecini temsil eder. Bu süreç, gerek ilgili varlıkları gerekse de bu varlıkların rollerini ve sorumlulukları altında gerçekleşen işlemleri barındırmaktadır. Veri mahremiyeti koruma sürecindeki ilgili varlıklar; veri sahipleri, veri toplayıcılar (yayıncılar) ve veri alıcılar olmak üzere üçe ayrılmakta bunlar aşağıda açıklanmaktadır [8].

Veri sahipleri; paylaşılan verideki mahremiyeti korunması gereken kişi, kurum ve kuruluşlardır. Gerek çeşitli yasal zorunluluklardan gerekse de kendi açık rızaları ile verilerini güvenilir kabul ettikleri veri toplayıcılara iletir. Bu veri iletimi için veri sahipleri ile veri toplayıcılar arasında gerek yasal gerekse de farklı teknik güven alt yapılarının oluşturulması gerekir.

Veri toplayıcılar; veri sahiplerinden çeşitli yöntemlerle topladıkları verileri depolayan, kullanan ve anonimleştirerek veri alıcıları ile paylaşan kişi, kurum veya kuruluşlardır. Anonimleştirme aşamasında mahremiyet seviyesi ile veri faydası arasındaki dengeyi sağlamakla yükümlüdür.

Veri alıcılar ise; paylaşılan anonim verileri kendi amaçları doğrultusunda kullanan ve güvenilir olmadığı kabul edilen üçüncü taraflardır. Saldırganların veri alıcısı gibi davranıp veri üzerinde ifşa saldırıları yapabilme ihtimali olduğu için, veri alıcıları güvenilir olmayan varlıklar olarak kabul edilir.

Genel bir veri mahremiyet koruma süreci, bu süreçteki varlıklar ve üstlendikleri roller Şekil 2.1’de gösterilmiştir.



Şekil 2.1. Geleneksel veride mahremiyet koruma süreci [8]

2.2.2. Geleneksel veride mahremiyet problemi

Veri mahremiyeti problemi, verinin mahremiyet seviyesi ile faydası arasındaki dengeyi sağlama problemi olarak tanımlanır. Bu problem, İstatistiksel İfşa Kontrolü (Statistical

Disclosure Control - SDC) [28], Mahremiyet Korumalı Veri Madenciliği (Privacy Preserving Data Mining - PPDM) [15,29] ve Mahremiyet Korumalı Veri Yayını (Privacy Preserving Data Publishing - PPDP) [2,9] gibi farklı araştırma gruplarınca çalışılan geniş bir konudur. Bu problemin çözümünde arzu edilen sonuç her ne kadar veri mahremiyetinin ve faydasının yüksek olması olsa da, böylesi bir durum pratikte mümkün olmadığı gibi; mahremiyetin en düşük olduğu yerde fayda en yüksek iken mahremiyetin en yüksek olduğu yerde fayda en düşüktür. Dolayısı ile veri mahremiyeti ile faydası arasında bir denge bulmak önemlidir.

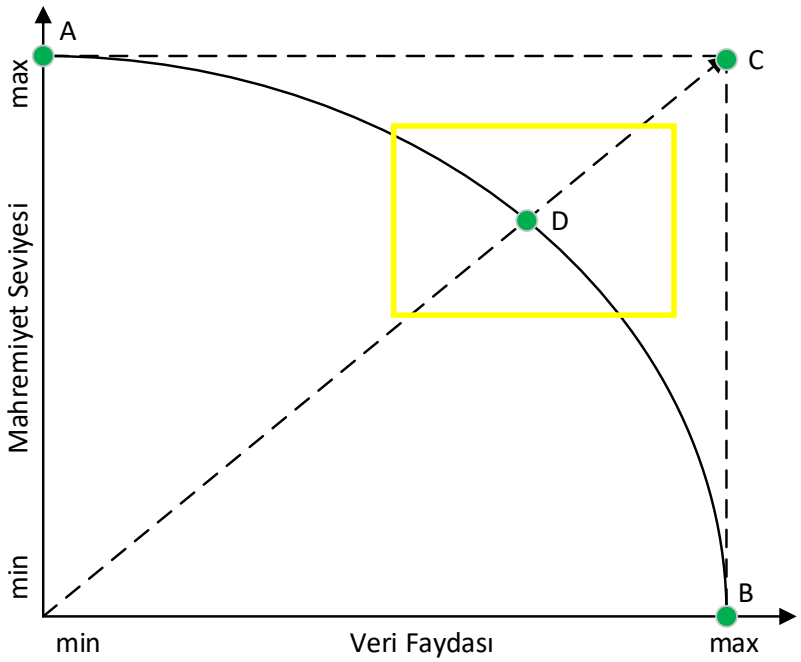
Veri mahremiyeti probleminin NP-Zor bir problem olduğu [5,30-39] numaralı çalışmalarda ortaya konulmuştur. Mahremiyeti korunacak veri sayısındaki artış arama uzayındaki ihtimallerin sayısını üssel olarak arttıracağı için, anonimleştirme karmaşıklık sınıfı açısından NP-Zor sınıfına girmektedir. Veri sayısının yanı sıra verinin boyutunun artması ile de veri mahremiyeti probleminin NP-Zor bir problem olduğu [27,35,40] çalışmalarında gösterilmiştir. Böylesi bir problemi çözmek için çeşitli kısıtlar altında optimale yakın ve kabul edilebilir çözümler aranır.

Veri mahremiyeti ile veri faydası arasındaki ilişki bir dengeyi temsil etmekte olup bu denge matematiksel olarak Eş. 2.1'de ve bu dengeyi gösteren bir denge eğrisi de Şekil 2.2'de sunulmuştur. Min ve Max ifadeleri sırasıyla %0 ve %100'ü temsil etmek üzere; verilen şekilde gösterilen A noktası $(Mahremiyet_{max}, Fayda_{min})$ noktasını, B noktası $(Mahremiyet_{min}, Fayda_{max})$ noktasını, C noktası $(Mahremiyet_{max}, Fayda_{max})$ noktasını ve D noktası ise $(Denge_i, Denge_j)$ noktasını temsil etmektedir. Burada A ve B birbirine zıt iki noktayı, C idealde hedeflenen ancak pratikte mümkün olmayan noktayı ve D ise denge eğrisi üzerinde aranacak denge noktasını temsil etmektedir. D noktası bu eğri üzerinde aşağı-yukarı hareket ettirilerek hedeflenen mahremiyet seviyesine ve veri faydasına ulaşılabilir.

$$\begin{cases} (Mahremiyet_{min}, Fayda_{max}) < (Denge_i, Denge_j) \\ (Denge_i, Denge_j) < (Mahremiyet_{max}, Fayda_{min}) \end{cases}, (min \leq i, j \leq max) \qquad (2.1)$$

Veri mahremiyeti problemini çözmek için literatürde çeşitli yaklaşımlara rastlamak mümkündür [35,41-43]. Bu yaklaşımlar optimale yakın çözüm ürettikleri için kabul

edilebilir çözümler olarak karşımıza çıkmakta ve her bir çözümün sunduğu veri faydasının değerlendirilmesi ise yine literatürde mevcut bulunan çeşitli bilgi metrikleri ile yapılmaktadır.



Şekil 2.2. Veri mahremiyeti - veri faydası denge eğrisi [3]

2.3. Büyük Veri

2012 yılında Gartner tarafından literatüre sunulan büyük veri kavramı, "daha iyi sezgi, karar verme ve süreç otomasyonunu mümkün kılan maliyet etkin, yenilikçi bilgi işleme biçimleri talep eden yüksek hacimli, yüksek hızlı ve/veya çok çeşitli bilgi varlıkları" [44] olarak tanımlanmıştır. Büyük veri kavramı için yapılan diğer tanımlar ise aşağıda sunulmuştur;

- Depolama, yönetim ve analiz için sıradan veri tabanı yazılım araçlarının yeteneklerinin gerisinde kaldığı veri kümesi [45],
- Yüksek hızda toplama, keşif ve analiz sayesinde çok çeşitli verinin çok büyük hacminden ekonomik olarak değer elde etmek için tasarlanan yeni nesil teknolojiler ve mimariler [46],
- Geleneksel veri işleme uygulamaları veya genel veri tabanı yönetim araçları kullanılarak yönetilmesi zor olan büyük ve karmaşık veri koleksiyonu [47],
- Farklı formatlarda ve farklı kaynaklardan çok yüksek hızlarda üretilen büyük miktardaki verilerdir [48].

Büyük veri, günümüzde gerek akademik alanlarda gerekse çeşitli sektör ve kurumlarda önemli ve popüler konulardan biri haline gelmiştir. Yeni teknolojilerden sistemlere, yazılımlardan donanımlara, programlama modellerinden tasarımlara kadar pek çok farklılığı içeren büyük veri kavramı, ülkelerin artık en çok yatırım yaptıkları alanlardan biridir. Sağlıktan eğitime, ülke yönetiminden istihbarata kadar pek çok alanda ele alınan büyük veri kavramı; politika belirleme, strateji geliştirme, seçim yönetimi, gelecek planlama, tahmin ve analiz yapma gibi çeşitli alanlarda da büyük fırsatlar sunmaktadır.

Büyük veri, geleneksel sistemlerin sınırlarını aşan ve kabul edilebilir bir sürede işlenemeyen karmaşık verilerin yönetimi, analizi, depolanması, anlamlandırılması ve görselleştirilmesi gibi konularda karşılaşılan zorlukların aşılması amacıyla ortaya çıkan bir kavramdır [49-51]. Hacim (volume), hız (velocity) ve çeşitlilik (variety) bileşenleriyle ortaya çıkan büyük veri kavramı, zamanla değişen problemlerin çözümü için değer (value), doğruluk (veracity), değişkenlik (variability) ve zafiyet (vulnerability) gibi pek çok bileşene de sahip olmuştur [51-57]. Bu bileşenler bazıları aşağıda kısaca açıklanmıştır;

- Hacim: verinin depolama biriminde kapladığı alanı temsil eder. Terabaytlarca sensör verisi, petabaytlarca sosyal medya resmi, milyarlarca satır ve binlerce sütundan oluşan gigabaytlarca arama verisi hacim bileşeni için örnek olarak verilebilir. Dijitalleşen dünyada, farklı teknolojilerin geliştirilmesi, yeni platformların oluşturulması ve ayrıca kurum ve kuruluşların verinin değerli bir varlık olduğunu anlamalarından dolayı üretilen ve toplanan verinin hacmi her geçen gün artmaktadır.
- Hız: verinin belirli periyotlardaki üretim frekansına karşılık gelir. Her 60 saniyede; 100 binden fazla tweet'in atılması, 700 binden fazla facebook durum güncellenmesinin yapılması, 170 milyondan fazla e-postanın atılması hız bileşeni için örnek olarak verilebilir. Yüksek hızda üretilen büyük veriler için geliştirilen veri işleme teknikleri toplu (batch), akan (streaming) ve gerçek zamanlı (real time) olarak üç farklı grupta ele alınmaktadır. Toplu işleme, belirli periyotlar arasında toplanan verinin toplu olarak işlendiği; akan veri işleme, çeşitli kaynaklar tarafından sürekli olarak üretilen verinin herhangi bir zaman kısıtlaması olmadan ve zaman gecikmesine töleranslı olarak işlendiği; gerçek zamanlı işleme ise gerçek zamanlı olarak gelen verinin anlık olarak işlenip bir aksiyonun ortaya konulduğu veri işleme teknikleridir.
- Çeşitlilik: veri kaynaklarında üretilen verinin formu olarak tanımlanır. Yapısal, yapısal olmayan ve yarı-yapısal olmak üzere üç kategoride sınıflandırılır. Bir veri tabanı tablosu veya excel dokümanı yapısal; resim, müzik, metin gibi dokümanlar yapısal olmayan;

xml, rdf ve html gibi yapılar ise yarı-yapısal olarak örneklendirilebilir.

- Değer: büyük veriden elde edilecek faydayı temsil eder. Verinin işlenmesinden sonra kuruma veya organizasyona bir değer katması beklenir. Örneğin; Sağlık Bakanlığı'nın kendi bünyesindeki büyük sağlık verilerini anlık olarak analiz etmesi ile bir salgın hastalığı daha ortaya çıkmadan tespit edebilmesi kuruma ve ülkemize katabilecek bir değerdir. Bu şekilde, verinin işlenmesi sonucu insanların veya kurumların faydalarına olabilecek çıktıların yani değerin üretilmesi beklenir.
- Doğruluk: verinin doğruluğunu ve güvenilirliğini belirtir. Veriye yetkisiz kişilerce yapılabilecek müdahaleler verinin doğruluğunu ve bütünlüğünü bozar. Dolayısıyla böylesi veriler üzerinde yapılacak analizlerin sonuçları da güvenilir olmayacaktır. Ancak, verinin doğruluğunu koruyarak o veriden üretilecek değerin de güvenilir olacağı kabul edilebilir.
- Değişkenlik: verinin içerisindeki çeşitli tutarsızlıklar veri üzerinde yapılacak analizlerin başarısını doğrudan etkiler. Örneğin; veri kümesinde gürültü veya aykırı verilerin bulunması durumunda yapılacak analizlerin başarısı düşük olacaktır. Dolayısıyla gürültülerin temizlenmesi veya aykırı verilerin onarılması gibi ön işlemler sayesinde veri üzerinde yapılacak analizlerin daha doğru olması sağlanabilir.
- Zafiyet: büyük veriler hassas bilgiler içerebileceğinden dolayı böylesi verilerin doğrudan yayınlanması veriyi ifşa saldırılarına karşı korumasız bırakır. Bu durumu önlemek amacıyla bu tür verilerin mahremiyetinin korunması ve yeterli seviyede mahremiyet önlemlerinin alınması önemlidir.

Büyük veri, literatürde çeşitli kriterlere göre sınıflandırılmıştır. Bu sınıflandırmanın temel amacı, birden fazla kriterin bir araya gelerek oluşturduğu büyük veri analitik sürecinde araştırmacılara yol göstermektir. Bu sınıflandırmalardan bazıları Çizelge 2.1'de belirtilmiş olup, verilen çizelge incelendiğinde büyük veri analitiği sürecinde işlenecek verilerin boyutu ve kapsamı daha iyi anlaşılacaktır. Büyük verinin üretilmesinden tüketilmesine kadar geçen süreçte, verinin kaynağının ve üretilme frekansının ne olabileceği, hangi formatta oluşacağı, tipinin ne olacağı, hangi türde analiz edileceği, işleme metodolojisinin ne olacağı ve bu veriyi kimlerin tüketeceği gibi sorular için verilebilecek muhtelif cevaplar verilen bu çizelgede belirtilmiştir. Burada dikkat edilmesi gereken en önemli husus, büyük veri analitik sürecini doğru olarak planlamak ve yönetmektir. Büyük veri ile geleneksel veri aradaki farkı tam olarak ortaya koymak adına yapılan karşılaştırma ise Çizelge 2.2'de verilmiştir.

Çizelge 2.1. Büyük verinin çeşitli kriterlere göre sınıflandırılması [58]

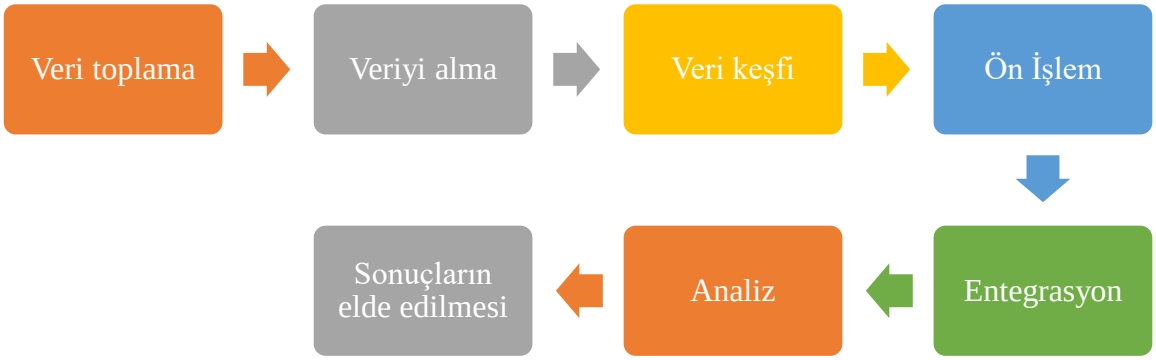
Kriter	Sınıflar
Veri Kaynağı	Web ve sosyal medya, makine üretimi veriler, insan üretimi veriler, dâhili kaynaklar, işlemsel veriler, biyometrik veriler, veri sağlayıcı kaynaklı veriler, veri üretici kaynaklı veriler
Veri Frekansı	İsteğe bağlı, sürekli, gerçek zamanlı, zaman serileri
Veri Formatı	Yapısal, yapısal olmayan, yarı yapısal
Veri Tipi	Meta, ana, geçmiş, işlemsel
Analiz Türü	Gerçek zamanlı, toplu işlem, akan veri
İşleme Metodolojisi	Tahminsel analiz, analitik (sosyal ağ analizi, konum tabanlı analiz, öznitelik tanıma, metin analizi, istatistiksel algoritmalar, uyarlama, konuşma analizi), sorgu ve raporlama
Veri Tüketiciler	İnsan, iş süreçleri, diğer kurumsal uygulamalar, diğer veri depoları

Çizelge 2.2. Geleneksel veri ile büyük verinin karşılaştırılması [16]

Kriter	Geleneksel Veri	Büyük Veri
Hacim	KB, MB, GB	TB, PB
Veri üretim oranı	saatlik, günlük	anlık, saniyelik, dakikalık
Veri yapısı	yapısal	yapısal, yapısal olmayan veya yarı yapısal
Veri kaynağı	merkezi	tamamen dağıtık
Veri entegrasyonu	kolay	zor
Veri depolama	ilişkisel veri tabanı	HDFS, NoSQL
Veri erişim	interaktif	toplu veya yakın gerçek zamanlı

Büyük veri analitiği bir süreç olduğu için, verinin üretilmesinden tüketilmesine kadar geçen bu süreçte bir iş akışı takip edilir. Büyük veri işlemede izlenebilecek örnek bir iş akışı Şekil 2.3'de gösterilmiştir [59]. Verilen şekle göre, büyük veri analitiğinde yapılacak ilk iş verinin ilgili kaynaktan veya kaynaklardan planlanan işe göre toplanmasıdır. Sonraki iş adımlarında ise sırasıyla, toplanan verinin tek bir veri deposuna alınması, veri deposuna tutulan verinin içeriğinin keşfedilmesi, işlenmeye hazır bir formata getirilmesi, farklı kaynaklardan veri

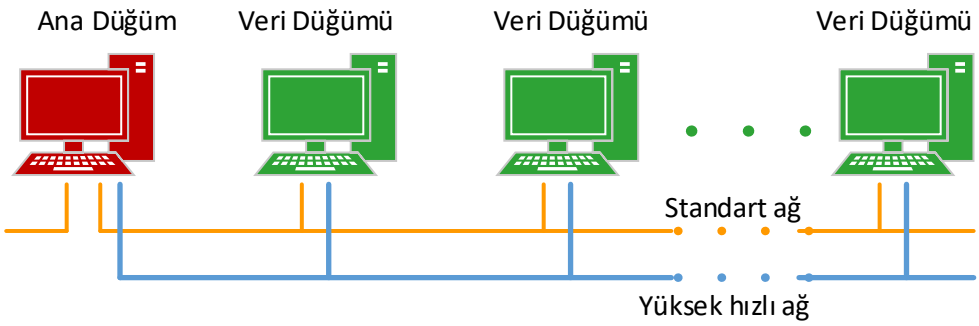
kümeleri toplanmışsa bu verilerin birleştirilmesi ve üzerinde hedeflenen analiz ve analitik işlemlerinin yapılarak sonuçların elde edilmesi ve yorumlanmasıdır.



Şekil 2.3. Büyük veri işlemede iş akışı [59]

Büyük veri mimarisi dağıtık bir yapıdan oluşur. Böylesi bir yapıda temel işleri yerine getiren bir ana düğüm ve hesaplama ve dağıtık depolama işlerinden sorumlu birden fazla alt düğüm bulunur. Efendi ve köle mimarisine dayalı olan bu mimaride, ana düğüm veri kümesini parçalara ayırıp veri düğümlerine iletir. Veri düğümleri ise kendi üzerindeki veri parçalarında paralel olarak verilen işleri yapar. Bu mimaride yer alan sistemler, tüm düğümlerde aynı yazılım, donanım ve ağ teknolojisi yapısına sahip olan homojen yapılardır. Bu sistemlerde yüksek hızlı ağ üzerinden veri transferi yapılırken, standart ağ üzerinden ilgili haberleşmeler sağlanır [60]. Örnek bir dağıtık sistem mimarisi Şekil 2.4'de gösterilmiştir.

Büyük veri analitiği yapmak için çeşitli teknik ve teknolojilere ihtiyaç duyulur. Bu ihtiyaçlar kapsamında hangi teknik ve teknolojilerin kullanılacağına doğru karar vermek gerekir. Aksi halde, analitik sürecinin doğru bir şekilde yönetilememesi ve beklenen çıktıların elde edilememesi gibi sonuçlar ortaya çıkabilir. Büyük veri teknolojileri, hizmet ettiği alana özgü çeşitli sınıflara ayrılmaktadır. Çizelge 2.3'de büyük veri analitiğinde kullanılan çeşitli teknolojiler ve bunların kullanıldığı çeşitli alanlar gösterilmektedir.



Şekil 2.4. Örnek bir dağıtık sistem mimarisi [60]

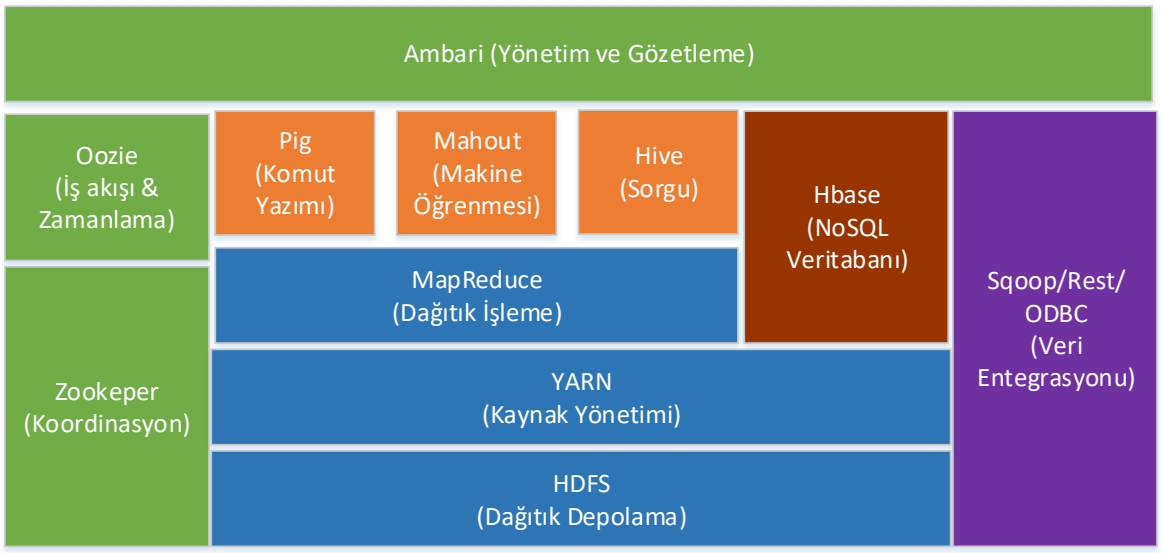
Çizelge 2.3. Farklı alanlar için kullanılan büyük veri teknolojileri [61,62]

Alanlar	Teknolojiler
İş zekâsı	Microsoft, Aws, Domo, Looker, GoodData, Information Builders, MicroStrategy, Atscale, Birst
Görselleştirme	Tableau, Sap, Google Cloud, Celonis, Qlik, Plotly, Chartio
Makine Öğrenmesi	Azure, Aws, Goole Cloud, Gamalon, DataRobot, Element, Versive, Bonsai, Weka, Keras, Mahout, Mllib
Platform-Lokal	Cloudera, Hortonworks, MapR, IBM Infosphere, BlueData
Platform-Bulut	Aws, Azure, Google Cloud, IBM Infosphere, Treasure Data, Altiscale, Cazena, CenturyLink, Databricks
Veri Tabanı-NoSQL	Google Cloud, Aws, Oracle, Azure, MongoDB, DataStax, RedisLabs, Cassandra, Apache Hbase, Nifi, MongoDB, Flume
Veri Transferi	Talend, Pentaho, Alteryx, Trifacta, Tamr, Paxata, StreamSets
Veri Entegrasyonu	Sap Data Services, Informatica, Mulesoft, Enigma, Segment
Veri Yönetimi	Informatica, SailPoint, Collibra, Alation, Okera
Ana çatı	Hadoop HDFS, Hadoop MapReduce, Spark
Sorgu	Spark SQL, Apache Drill, Flink, Hive, Pig, SlamData
Akan Veri Yönetimi	Spark, Flink, Apex, Kafka, Storm, Beam, Druid

Büyük veri işlemede yapılacak en önemli tercih hangi veri işleme anaçatısının kullanılacağına karar vermektir. Hadoop ve Spark, büyük veri analitiğinde en çok kullanılan anaçatı teknolojileridir. Bu anaçatılardan, hangisinin neye göre tercih edilmesi gerektiği ele

alınan probleme bağlıdır. Örneğin; akan veri üzerinde bir analitik uygulaması yapılması durumunda Spark Hadoop'a göre daha avantajlı iken, diskte tutulan bir veriyi işlemede Hadoop Spark'a göre daha avantajlı olabilir [63,64]. Hadoop ve Spark anaçatıları aşağıda kısaca açıklanmıştır.

Hadoop: büyük veri analitiğinde kullanılan, güvenilir, hızlı, hata toleranslı, ölçeklenebilir ve dağıtık hesaplamaya imkân sağlayan bir açık kaynak anaçatı sistemidir. Hadoop Dosya Sistemi (Hadoop File System - HDFS) olarak adlandırılan bir dosya sistemi ile dağıtık bir programlama modeli olan MapReduce yapılarını kullanır. Büyük veri kümelerini küçük parçalara bölerek her bir parçanın bulunduğu düğüm üzerinde işlenmesini sağlar. Bilgisayar kümeleri arasında büyük verilerin dağıtık ve paralel işlenmesine imkan tanıyarak bir tek makineden yerel işlemleri yürüten binlerce makineye işleri ölçeklendiren bir yapıdır [65-67]. Hadoop anaçatısının sahip olduğu temel bileşenler Şekil 2.5'de gösterilmiştir.

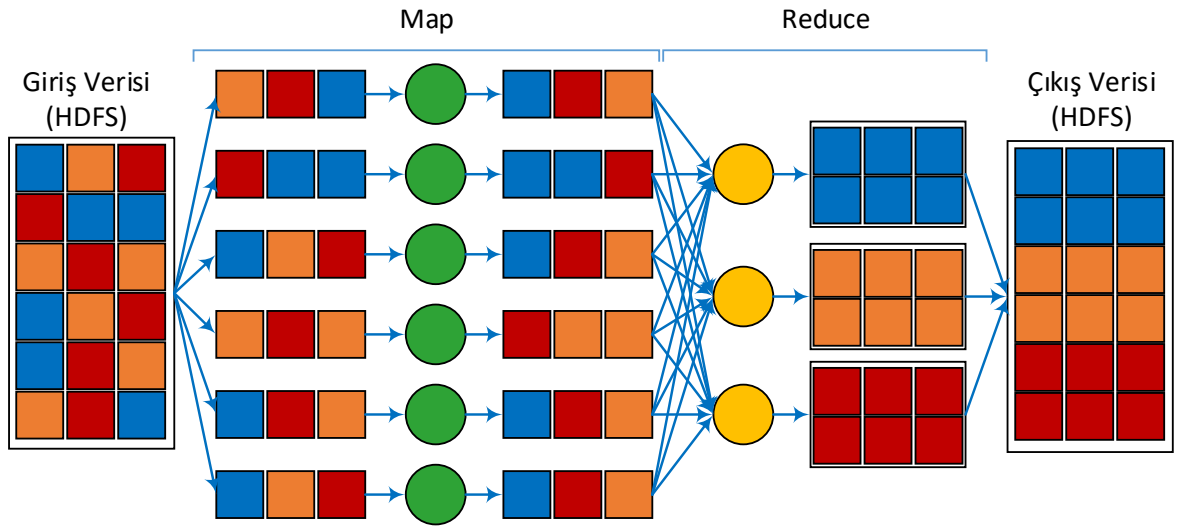


Şekil 2.5. Temel Hadoop bileşenleri [68]

HDFS: dağıtık dosya sistemi olup, uygulamalarda kullanılan tüm giriş ve çıkış verilerini depolayan yapıdır. Temel mimarisi efendi-köle örüntüsüne dayanmakla beraber, bir ana düğüm (NameNode) ve çok sayıda köle düğümden (DataNode) oluşur. Ana düğüm, paylaşılan dosya sistemi için üst bilgi yapısını saklayarak giriş ve çıkış işlemleri için köle düğümü yönlendirir. Sistemdeki düğümlerin çökmesi halinde sistemin devamlılığını sağlamak amacıyla ana düğüm kullanılarak çöken düğümler tekrardan aktif hale getirilir. Veri çok sayıda parçalara bölünerek farklı düğümlere dağıtılır ve kopyalar halinde tutulur.

Herhangi bir düğümde tutulan veride sorun oluşması halinde otomatik olarak mevcut kopyalardan alınarak işlemlerin devamlılığı sağlanır [66].

MapReduce: Hadoop için geliştirilen ve dağıtık mimariye uygun bir programlama modelidir. Bu modelde, veriyi işlemek için kullanıcı tanımlı Map ve Reduce olarak adlandırılan iki fonksiyon mevcuttur. Map fonksiyonu HDFS'te tutulan verileri anahtar/değer ikililerine dönüştürürken bu anahtar/değer ikilileri Reduce fonksiyonuna ileterek yapılacak işlem için ilgili süreç işletilir [67,69]. Genel bir MapReduce paradigması Şekil 2.6'da sunulmuştur. Verilen şekilde, renkler anahtarları kutular ise veri değerlerini göstermekte olup, HDFS'de dağıtık olarak tutulan veriler Map fonksiyonu ile anahtar/değer ikililerine dönüştürülerek sıralanmış ve Reduce fonksiyonunda ise aynı anahtarlara sahip veriler gruplandırılmıştır.



Şekil 2.6. MapReduce tabanlı örnek bir sistem mimarisi [67,68]

Diğer Hadoop bileşenleri ise aşağıda kısaca açıklanmıştır [70,71];

- HBase: sütun tabanlı dağıtık bir veri tabanıdır. Büyük veri kümelerine gerçek zamanlı olarak rasgele erişime, okuma ve yazmaya olanak sağlayan bir uygulamadır.
- Hive: bir veri ambarı olup, araştırmacılara büyük veri kümeleri üzerinde sorgu yapabilme imkânı tanır.
- Mahout: ölçeklenebilir makine öğrenmesi ve veri madenciliği algoritmalarını sağlayan kütüphanedir.
- Pig: büyük veriyi keşfetmede kullanılan komut yazım dilidir.
- Sqoop/Rest/ODBC: yapısal bir veri deposundan Hadoop'a veri aktarmak için kullanılır.

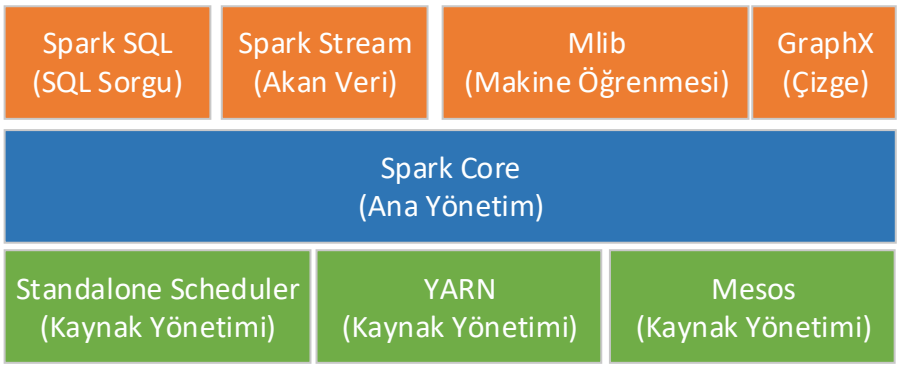
- Oozie: bağımsız Hadoop işleri için iş akışı zamanlayıcısıdır.
- Zookeper: dağıtık uygulamalar için koordinasyon hizmeti sunar.
- Ambari: Hadoop kümelerinin yönetimi ve izlenmesini sağlar.
- YARN: kaynak yönetimini sağlayan yapıdır.

Spark: bilgisayar kümeleri üzerinde büyük boyutlu veri analitiği yapılmasına imkân sağlayan, hızlı, etkili, hata toleranslı, ölçeklenebilir ve genel amaçlı bir büyük veri işleme anaçatısıdır [72]. Genel olarak toplu işlemlerin, iteratif algoritmaların, interaktif sorguların ve akan verilerin işlenmesi için tasarlanmıştır. Veriyi geçici bellekte tutarak işlediği için yüksek hızlarda işlem kapasitesi sunar. Geçici bellekte tutulan, Esnek Dağıtılmış Veri Kümesi (Resillient Distributed Dataset – RDD) adı verilen ve pek çok hesaplama düğümü arasında dağıtılmış parçaların bir koleksiyonu olan kayıt dosyaları üzerinde işlem yapılmasını sağlar [73]. Büyük veri kümelerini parçalara bölerek veri düğümlerinde işler ve kopya tabanlı çalışarak mevcut işlenen veride herhangi bir bozulmasa olması halinde kopyasını kullanarak işlemlerin devam etmesini sağlar. Klasik MapReduce yapısını kendi mimarisi için genişleterek dağıtık işlem yapılmasına imkân tanır. Genişletme işlemini üç şekilde yapar. İlki; katı bir Map-Reduce işleminden ziyade çevrimsiz yönlü çizge yapısını kullanarak ara sonuçların diske yazılması gerektiği durumlarda bu işlemi işlem hattının sonuna ekler. İkincisi; çeşitli dönüşüm işlemlerini kullanıcıya sunarak işlemlerin daha kolay yapılmasını sağlar. Üçüncü ise, düğümler arası bellekte işlem yapılmasını desteklediği için ara sonuçların da bellekte tutulmasına imkân tanır [74].

Şekil 2.7'de sunulan temel Spark bileşenleri aşağıda açıklanmıştır [72-75];

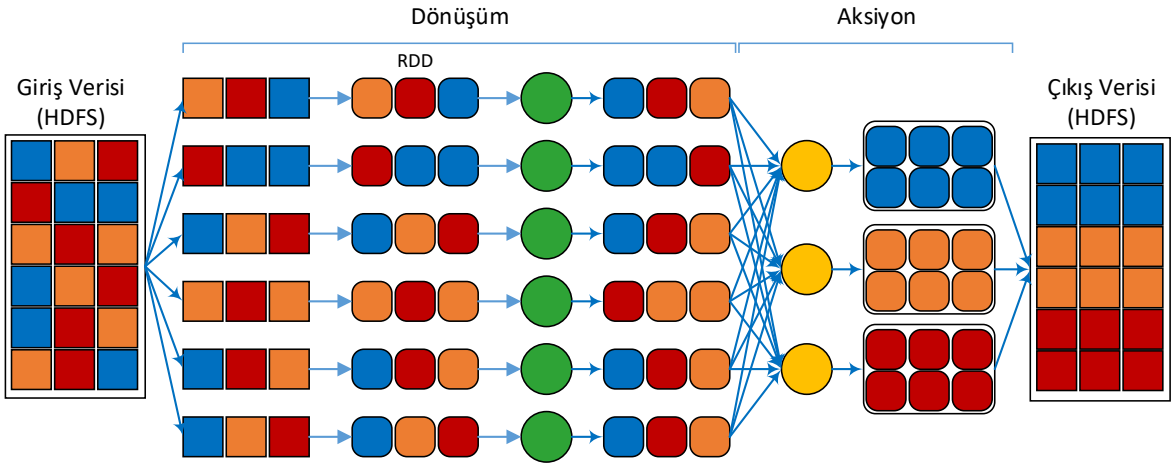
- Spark Core: görev zamanlayıcısı, hafıza yönetimi, hata onarımı, depolama sistemleri ile etkileşim gibi pek çok temel fonksiyonları içerir. Aynı zamanda RDD'lerin tanımlandığı, değiştirildiği ve oluşturulduğu pek çok uygulamaları barındırır.
- SparkSQL: yapısal veri üzerinde sorgu yapmayı sağlayan, Hive tabloları, Parquet ve JSon gibi veri kaynaklarını destekleyen bir araçtır.
- Spark Streaming: akan veriler üzerinde analitik işlemlerinin yapılmasına imkân sağlar.
- MLib: Makine öğrenmesi uygulamaları için sunulan bir kütüphanedir. Sınıflandırma, regresyon analizi, kümeleme gibi çeşitli makine öğrenmesi algoritmalarını içerir.
- GraphX: çizge tabanlı paralel hesaplamaların yapılmasına, çeşitli çizgelerin oluşturulmasına ve bu çizgeler üzerinde dönüşümlerin uygulanmasına imkân sağlar.

- Küme Yöneticileri: küme içerisindeki düğümleri yöneten yapılardır. Basit uygulamalar için Standalone Scheduler yöneticisi yeterli olabilirken; büyük çaplı uygulamalarda kaynak ve işlemci kullanımı gibi durumları optimize etmesinin getirdiği avantajlardan dolayı YARN veya Mesos tercih edilebilir.



Şekil 2.7. Temel Spark bileşenleri [72]

Spark tabanlı bir sistem mimarisi Şekil 2.8'de gösterilmiştir. Verilen şekle göre, HDFS'de dağıtık halde tutulan veri kümesi RDD yapısına çevrilerek geçici belleğe alınır. Burada çeşitli dönüşüm işlemleri uygulanarak veri istenilen formata getirilir. Sonrasında herhangi bir aksiyon işlemi uygulanarak istenilen sonuçlar elde edilir ve gerek duyulması halinde bellekteki RDD yapıları HDFS'e geri yazılır.



Şekil 2.8. Spark tabanlı örnek bir sistem mimarisi [76]

2.4. Büyük Veride Mahremiyet

Büyük veri mahremiyeti, büyük verinin mahremiyet seviyesi ile faydası arasındaki dengeyi bulmaya yönelik zor bir problemdir [8,13]. Büyük verinin işlenmesi, paylaşılması ve anlamlandırılması sırasında ortaya çıkabilecek mahremiyet ihlallerinin engellenmesine yönelik çözümler bulmaya çalışmak, oldukça kapsamlı ve bütünsel bir bakış açısı gerektirir. Büyük veride mahremiyet problemi, sadece büyük veri sahiplerinin mahremiyetini korumayı değil aynı zamanda büyük verinin sunacağı faydayı da içerir. Geleneksel veride olduğu gibi, büyük veride de mahremiyet seviyesi ile veri faydası arasında bir dengenin bulunması önemlidir. Çünkü büyük veride de mahremiyet önlemlerinin gereğinden fazla alınması durumunda veri faydası en düşük seviyesine ulaşırken, veri faydasının en üst seviyede sağlanması durumunda ise mahremiyet ihlallerinin yaşanma ihtimali en üst seviyeye çıkar. Diğer zorluklar da dikkate alındığında, büyük veri mahremiyeti probleminin ideal çözümü için probleme özgü varsayımlar altında optimale yakın çözümler üzerinde çalışılması gerektiği görülebilir.

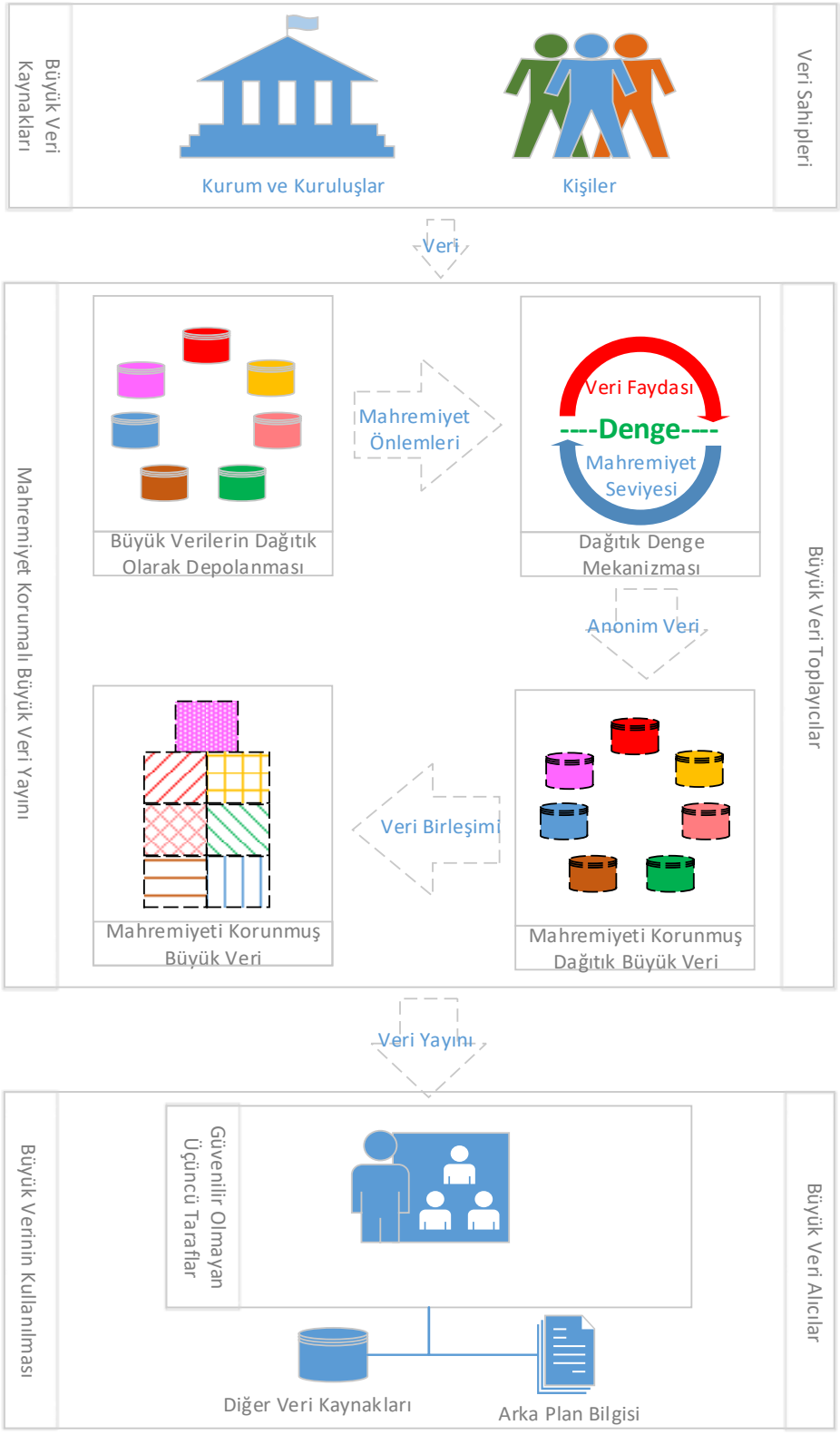
2.4.1. Büyük veride mahremiyet koruma süreci

Büyük veride mahremiyet koruma süreci, büyük verinin elde edilmesinden mahremiyetinin korunarak yayınlanmasına kadar geçen süreci temsil etmekte olup, gerek çeşitli varlıkları gerekse de bu varlıkların rollerini ve sorumluluğu altında gerçekleşen işlemleri barındırmaktadır. Genel bir büyük veri mahremiyet koruma süreci Şekil 2.9'da gösterilmiştir. Bu süreçteki varlıklar; büyük veri sahipleri, büyük veri toplayıcılar (yayıncılar) ve büyük veri alıcılar olmak üzere üçe ayrılmakta olup aşağıda açıklanmıştır.

Büyük veri sahipleri, paylaşılan büyük verideki mahremiyeti korunması gereken kişi, kurum ve kuruluşlardır. Büyük veri sahipleri, gerek yasal bildirimler gerekse de çeşitli gerekliliklerden dolayı sahip oldukları verilerini büyük veri toplayıcılara iletir. Bu veri iletimi için çeşitli güven ve teknik mekanizmalarının oluşturulması gerekir.

Büyük veri toplayıcılar, büyük veri sahiplerinden çeşitli yöntemlerle topladıkları verileri dağıtık olarak depolayan, dağıtık olarak kullanan, dağıtık olarak anonimleştirip büyük veri alıcıları ile paylaşan ve büyük veri işleme alt yapısına sahip kişi, kurum veya kuruluşlardır.

Dağıtık veriler üzerinde dağıtık mimariye uygun olarak geliştirilen anonimleştirme tekniklerini uygulayarak veriyi anonim hale getirirler.



Şekil 2.9. Büyük veride mahremiyet koruma süreci

Büyük veri alıcıları ise, paylaşılan büyük anonim veriler üzerinde belirledikleri amaç doğrultusunda analitik işlemlerini gerçekleştiren, büyük veri işleme alt yapısına sahip ve güvenilir olmadığı kabul edilen üçüncü taraflardır. Saldırganların da büyük veri alıcısı olarak davranıp veri üzerinde çeşitli saldırılar yapabilme ihtimali olduğu için, bu varlıklar güvenilir olarak kabul edilmez.

2.4.2. Büyük veride mahremiyet problemi

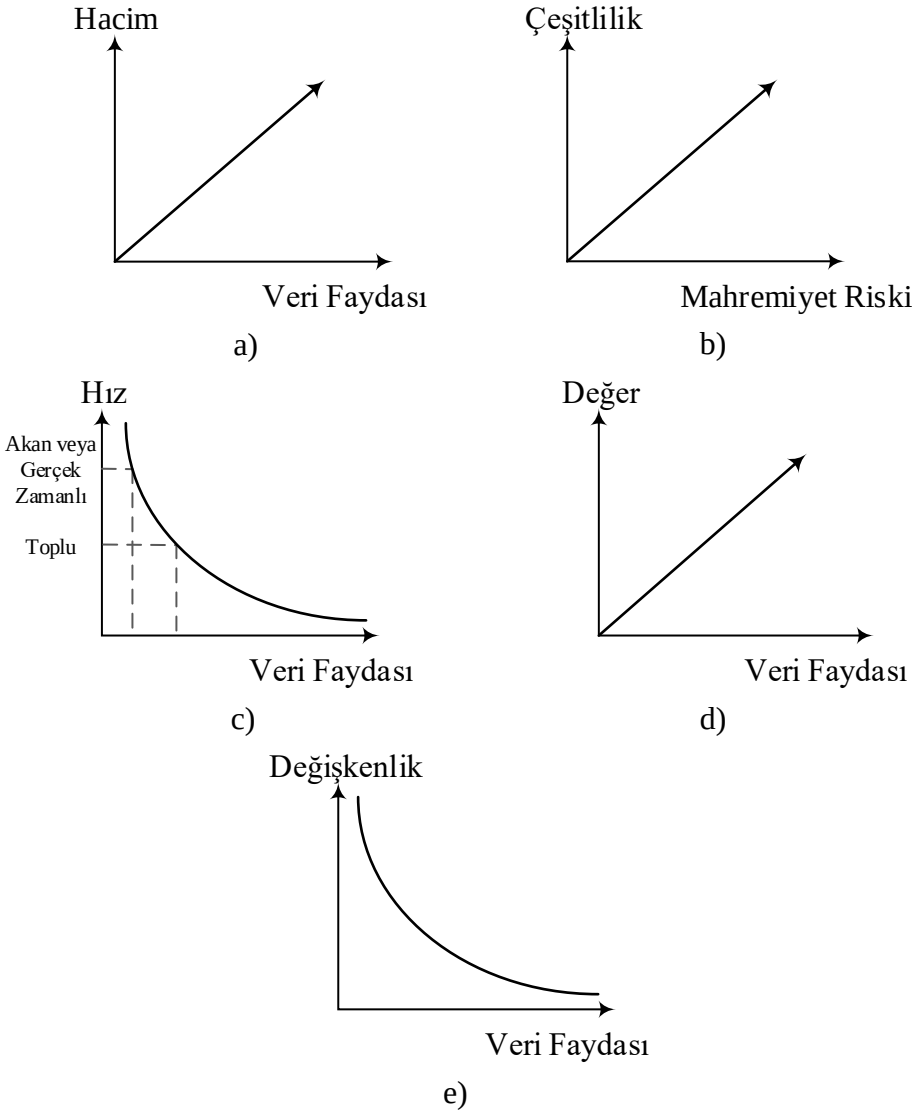
Bu bölümde [77] numaralı kaynak ışığında; büyük veride mahremiyet-fayda problemine büyük veri bileşenleri dikkate alınarak yeni bir bakış açısı kazandırılmaya çalışılmış ve aşağıda sunulmuştur.

Yapısal formdaki büyük veriler için, bu bileşenlerden bazılarının veri mahremiyeti kapsamında ilişkilendirilmesi Şekil 2.10'da gösterilmiş olup aşağıda açıklanmıştır;

- Hacim: üretilen veri sayısının artması daha yüksek hacimli veriler üzerinde analiz yapılmasına olanak sağlar. Bundan dolayı, mahremiyet seviyesi sabit olarak kabul edildiğinde, veri hacminin artması ile beraber veri faydası da artar.
- Hız: veri faydasının ömrünü doğrudan etkiler. Veriler, belirlenen periyotlarla toplanarak toplu halde işlenebileceği gibi, akan veya gerçek zamanlı olarak da işlenebilmektedir. Bu durumda, üç farklı yaklaşımdan toplu veri işlemede veri ömrü diğerlerine göre daha uzun olacağı için en yüksek veri faydası sunan grup olarak değerlendirilebilir.
- Çeşitlilik: farklı kaynaklardan toplanan verilerin bir araya getirilmesi ile daha anlamlı analitikler yapılarak daha iyi sonuçlar elde edilebilir. Ancak, kişiyi tanımlayıcı yeni bilgilerin mevcut veri kümesine eklenmesi kişi hakkında daha detaylı çıkarımlar yapılmasına imkân sağlayacağı için bu durumda mahremiyet riski de artacaktır.
- Doğruluk: bütünlüğü korunmuş ve güvenilir kabul edilen büyük veri kümesinin anonimleştirilmesi sonrası elde edilecek veri faydası da geçerli ve kabul edilebilirdir. Ancak yetkisiz kişilerin veriye erişerek veriyi değiştirmesi, anonimleştirilen veriden anlamlı sonuçların elde edilmesini doğrudan etkileyeceği için bu durumda verinin sunacağı faydanın güvenilirliği de sorgulanacaktır.
- Değer: yüksek veri faydası sunan anonim bir veri kümesinden yüksek seviyede değer üretildiği kabul edilir. Bundan dolayı, verinin mahremiyetini korurken aynı zamanda

yüksek fayda sunmasını da sağlamak, büyük anonim veriden elde edilecek değerin de yüksek seviyede olmasına imkân tanır.

- Değişkenlik: veride bulunması muhtemel gürültü ve aykırı veri gibi çeşitli tutarsızlıklar anonim verinin sunacağı faydayı düşürür. Veri faydasını arttırmak adına bu tür tutarsızlıkların üstesinden gelebilecek çeşitli yaklaşımlara ihtiyaç duyulur.
- Zafiyet: büyük veri kümeleri hassas bilgiler içerebileceğinden dolayı, ifşa saldırılarını engellemek için çeşitli mahremiyet koruyucu önlemlere başvurulması gerekir.



Şekil 2.10. a) Hacim-veri faydası ilişkisi, b) Çeşitlilik-mahremiyet riski ilişkisi, c) Hız-veri faydası ilişkisi, d) Değer-veri faydası ilişkisi, e) Değişkenlik-veri faydası ilişkisi

2.5. Veri Mahremiyetini Hedef Alan Tehditler

Arka plan bilgileri ile veri bağlama (eşleştirme) yöntemleri, veri mahremiyetine yönelik tehditlerin başında gelir [2]. Yayınlanan veriler ile halka açık veya önceden edinilmiş arka plan bilgilerinin bağlanmasıyla yapılan veri eşleştirmeleri sonucunda istenmeyen ifşalar meydana gelebilir. Arka plan bilgisine sahip bir saldırgan, sahip olduğu bilgiler ile yayınlanan veriler arasında kayıt, hassas öznitelik veya tablo düzeyinde bağlantı kurarak çeşitli saldırılar düzenleyebilir [78,79]. Bu saldırılar sonucunda kimlik [80-82], hassas öznitelik [36,83] ve üyelik ifşalarının [84] yaşandığı rapor edilmiştir. Mahremiyeti hedef alan tehditlerden bazılarına aşağıdaki paragraflarda yer verilmiştir.

Arka plan bilgileri: mahremiyet saldırılarının ve ihlallerinin yaşanmasında önemli rol oynayan arka plan bilgileri, çeşitli kurum ve kuruluşlarca yayınlanan verilerden, sosyal ve sanal ortamlardan, basılı ve görsel yayınlardan, gerçek dünyadaki ilişkilerden ve diğer pek çok yollardan elde edilebilen bilgilerdir. Bu bilgiler yayınlanan veri kümeleriyle eşleştirildiğinde çeşitli mahremiyet ihlalleri yaşanabilir [85].

Kimlik ifşası: arka plan bilgisine sahip bir saldırgan, kimlik bilgileri içeren veri kümeleri ile yayınlanan kimliksiz verileri yarı tanımlayıcılar düzeyinde eşleştirerek mahremiyet ifşası gerçekleştirebilir. Böylelikle saldırgan, kimliksiz yayınlanan veri içerisindeki kurbana ait hassas bilgileri öğrenerek kurbanın kimliğini hassas bilgileriyle birlikte ifşa edebilir [4].

Hassas öznitelik ifşası: arka plan bilgisine sahip bir saldırgan, yayınlanan verideki hassas özniteliklerin homojen dağılımına bağlı olarak kurbanın hassas bilgilerini ifşa edebilir [24].

Üyelik ifşası: bir saldırgan, paylaşılan veri kümesinde kurbana ait verilerin olduğunu öğrendiğinde yayınlanan veriye göre çıkarımlar yapabilir. Yayınlanan veride kurbanın bulunduğunu bilen bir saldırgan kurbanın bu veriyi yayınlayanla ilişkisini ortaya koyarak üyelik ifşası gerçekleştirebilir [84].

2.6. Veri Mahremiyeti Koruma Yaklaşım ve Modelleri

Şifreleme ve anonimleştirme, veri mahremiyetini korumada kullanılan iki temel yaklaşım olup aşağıda açıklanmıştır [9,86].

Şifreleme; hassas veriyi gizleme adına kullanılan, verinin gizlilik ve bütünlük ilkelerini korumasını sağlayan, yüksek seviyede veri mahremiyeti sunan bir yaklaşımdır. Maliyetli bir işlem olması ve veri faydası sunmamasından dolayı mahremiyet korumalı veri yayınlama yaklaşımlarında kullanılmaz [86].

Anonimleştirme; sıklıkla kullanılan fayda temelli bir mahremiyet koruma yaklaşımıdır. Veri sahibinin kimliğinin tespit edilmesini zorlaştırmak amacıyla, veri üzerinde genelleştirme ve baskılama gibi teknikler kullanarak paylaşılan verileri ifşa saldırılarına karşı korur [87]. Paylaşılan veriler genel olarak tablo formunda verilerdir. Anonimleştirilecek tablolarda yer alan öznitelikler tam-tanımlayıcı, yarı-tanımlayıcı, hassas ve hassas-olmayan olmak üzere dört farklı sınıfta incelenir [9]. Tam-tanımlayıcılar; kimlik numarası, ad-soyad, pasaport numarası vb. gibi kişiyi doğrudan tanımlayan bilgilerdir. Doğrudan tanımlama özelliği olmayan özniteliklerin bir araya gelmesiyle veri sahibinin kimliğini tanımlayabilme potansiyeline sahip olan öznitelik grubuna ise yarı-tanımlayıcı denilmektedir. Yaş, cinsiyet ve posta kodu özniteliklerinin oluşturduğu öznitelik grubu bu sınıfa örnek olarak verilebilir. Kişinin mahremiyetini ifşa etme potansiyeline sahip olan veriler ise hassas öznitelik sınıfına girmektedir. Hastalık, etnik köken, maaş vb. bilgiler hassas özniteliklere örnek olarak verilebilir. Yukarıdaki sınıflandırmalar dışında kalan diğer kamuya açık öznitelikler ise hassas olmayan öznitelikler olarak sınıflandırılabilir.

Çizelge 2.4'de anonimleştirilecek örnek bir veri kümesi, Çizelge 2.5'de bu veri kümesi için özniteliklerin sınıflandırılması, Çizelge 2.6'da ise anonimleştirme sonrası elde edilen 2-anonim veri kümesi ve bu veri kümesinde oluşan eşlenik sınıfları gösterilmektedir.

Çizelge 2.4. Anonimleştirilecek örnek bir veri kümesi

TC No	Yaş	Boy	Gelir	Hobi	Hastalık
1111111111	30	185	3000	Futbol	Grip
2222222222	25	190	3500	Tenis	Diyabet
3333333333	65	175	4000	Yüzme	Kanser
4444444444	70	180	4500	Müzik	Baş ağrısı

Çizelge 2.5. Anonimleştirme için özniteliklerin sınıflandırılması

Tam Tanımlayıcı	Yarı Tanımlayıcı			Hassas Olmayan	Hassas
TC No	Yaş	Boy	Gelir	Hobi	Hastalık
1111111111	30	185	3000	Futbol	Grip
2222222222	25	190	3500	Tenis	Diyabet
3333333333	65	175	4000	Yüzme	Kanser
4444444444	70	180	4500	Müzik	Baş ağrısı

Çizelge 2.6. Eşlenik sınıfların ve anonimleştirme operatörlerinin gösterilmesi

Yaş	Boy	Gelir	Hobi	Hastalık	
25-30	185-190	3***	Futbol	Grip	Eşlenik Sınıf 1
25-30	185-190	3***	Tenis	Diyabet	
65-70	175-180	4***	Yüzme	Kanser	Eşlenik Sınıf 2
65-70	175-180	4***	Müzik	Baş ağrısı	

Çizelge 2.6'da gösterilen 2-anonim veri kümesi için, anonimleştirme sonrası iki tane eşlenik sınıf elde edilmiş, yaş ve boy öznitelikleri için genelleştirme operatörleri uygulanırken, gelir özniteliği için ise baskılama işlemi uygulanmıştır. Anonimleştirmede sıklıkla kullanılan genelleştirme ve baskılama operatörleri aşağıda daha detaylı açıklanmıştır.

Genelleştirme: verinin anlamsal bütünlüğünü koruyarak daha az detay sunacak şekilde yeniden ifade edilmesini sağlayan yapıdır [88,89]. Bu detay azaltma işlemi sayısal ve kategorik değerler için farklı uygulanır. Örneğin sayısal bir değer bir üst seviyedeki değer aralıkları ile genelleştirilirken, kategorik bir değer ise bir üst seviyedeki başka bir kategorik değer ile ifade edilebilir. Çizelge 2.6'da yaş ve boy öznitelikleri için bir genelleştirme işlemi uygulanmıştır.

Baskılama: karakter maskeleme işlemi olarak bilinen bu operatör, kayıt, değer veya öznitelik seviyesinde yapılabilir [90,91]. Çizelge 2.6'da gelir özniteliği için bir baskılama işlemi uygulanmıştır.

Çizelge 2.7. Örnek veri tablosu A [26]

Posta Kodu	Yaş	Maaş	Hastalık
47677	29	3000	Mide Ülseri
47602	22	4000	Gastrit
47678	27	5000	Mide Kanseri
47905	43	6000	Gastrit
47909	52	11000	Grip
47906	47	8000	Bronşit
47605	30	7000	Bronşit
47673	36	9000	Zatürre
47607	32	10000	Mide Kanseri

Anonimleştirme kapsamında belirtilen öznitelik sınıflarını dikkate alan ve anonimleştirme operatörlerini kullanarak mahremiyet koruma gereksinimlerini karşılayan çeşitli mahremiyet koruma modellerine ihtiyaç vardır. *k*-Anonimlik, *l*-Çeşitlilik, *t*-Yakınlık ve *δ*-Mevcudiyet literatürde bilinen ve yaygın olarak kullanılan temel mahremiyet koruma modelleri olup aşağıda kısaca açıklanmıştır.

k-Anonimlik: bir verinin en az *k-1* tane veriden ayırt edilememesini sağlayan mahremiyet koruma modelidir. Kimliği ifşa edilmek istenen kişinin yani kurbanın yarı tanımlayıcılarının değeri bilinse bile, o kişinin kaydı diğer *k-1* tane kayıttan ayırt edilemez [6]. Bu şekilde kayıt bağlama saldırısı olarak da bilinen kimlik saldırısı engellenmiş olur. İlk bakışta basit bir problem olarak görünmesine rağmen optimum *k*-Anonimliği sağlamanın NP-Zor bir problem olduğu çeşitli çalışmalarda [5,30,34,92] ispatlanmış ve gerek optimal gerekse de yakın optimal çözümler geliştirilmeye çalışılmıştır. Çizelge 2.7'de sunulan A tablosunun 3-anonim versiyonu Çizelge 2.8'de gösterilmektedir.

Çizelge 2.8. A tablosunun 3-anonim versiyonu [26]

Posta Kodu	Yaş	Maaş	Hastalık
476**	2*	[3000-5000]	Mide Ülseri
476**	2*	[3000-5000]	Gastrit
476**	2*	[3000-5000]	Mide Kanseri
4790*	≥40	[6000-11000]	Gastrit
4790*	≥40	[6000-11000]	Grip
4790*	≥40	[6000-11000]	Bronşit
476**	3*	[7000-10000]	Bronşit
476**	3*	[7000-10000]	Zatürre
476**	3*	[7000-10000]	Mide Kanseri

Çizelge 2.9. A tablosunun 3-Çeşit versiyonu [26]

Posta Kodu	Yaş	Maaş	Hastalık
476**	2*	[3000-5000]	Mide Ülseri
476**	2*	[3000-5000]	Gastrit
476**	2*	[3000-5000]	Mide Kanseri
4790*	≥40	[6000-11000]	Gastrit
4790*	≥40	[6000-11000]	Grip
4790*	≥40	[6000-11000]	Bronşit
476**	3*	[7000-10000]	Bronşit
476**	3*	[7000-10000]	Zatürre
476**	3*	[7000-10000]	Mide Kanseri

l-Çeşitlilik: *k*-Anonimlik, kimlik ifşası saldırılarına karşı koruma sağlarken hassas verilerin ifşasına karşı bir koruma sağlayamaz. Machaanavajjhala ve ark. [24], *k*-Anonimlik modelinin bu sorununu vurgulayarak hassas özniteliklerin de mahremiyetini koruyan *l*-Çeşitlilik modelini önermiştir. *l*-Çeşitlilik modeli, hassas verilerin ifşa edilmesini mümkün

olabildiği kadar engellemek amacıyla hassas verilerin çeşitliliğinin en az *l* sayıda olmasını garanti eder. Çizelge 2.7'de sunulan A tablosunun 3-Çeşit versiyonu Çizelge 2.9'da gösterilmektedir.

t-Yakınlık: *l*-Çeşitlilik, güçlü bir mahremiyet modeli olmasına rağmen, Li ve ark. [93] çarpık veri dağılımına sahip veri kümelerinde *l*-Çeşitlilik modelinin yetersiz olduğunu göstermiş ve *t*-Yakınlık modelini önermiştir. *l*-Çeşitlilik, hassas değerler arasındaki anlamsal yakınlıklara ve hassas değerlerin dağılımının genel dağılımdan önemli ölçüde farklı olmasına bağlı olarak yapılabilecek muhtemel çarpıklık saldırılarına karşı mahremiyet korumasında yetersiz kalır. Örneğin, bir hassas verinin tüm tablodaki oranı %5 iken, bir eşlenik sınıfı içerisindeki oranı %50 ise bu durumda ciddi bir mahremiyet ihlali ortaya çıkabilir. *t*-Yakınlık yöntemi, yarı-tanımlayıcılar üzerindeki herhangi bir gruptaki bir hassas özniteliğin dağılımını tüm tablodaki özniteliklerin dağılımına yakın olmasını gerektirir [9,93,94]. Çizelge 2.7'de sunulan A tablosuna, *t*-Yakınlık modeli uygulanması sonucu elde edilen muhtemel bir tablo Çizelge 2.10'da gösterilmektedir.

Çizelge 2.10. A tablosunun 0,278-Yakınlık versiyonu [26]

Posta Kodu	Yaş	Maaş	Hastalık
4767*	≤40	[3000-10000]	Mide Ülseri
4767*	≤40	[3000-10000]	Mide Kanseri
4767*	≤40	[3000-10000]	Zatürre
4790*	≥40	[6000-11000]	Gastrit
4790*	≥40	[6000-11000]	Grip
4790*	≥40	[6000-11000]	Bronşit
4760*	≤40	[4000-10000]	Bronşit
4760*	≤40	[4000-10000]	Gastrit
4760*	≤40	[4000-10000]	Mide Kanseri

δ-Mevcudiyet: arka plan bilgisine sahip bir saldırgan, yayınlanan anonim veride kurbana ait verilerin de olduğunu bilmesi durumunda önemli bir mahremiyet ihlali gerçekleşebilir. Arka

plan bilgisine sahip bir saldırgan üyelik bilgisini de kullanarak veri bağlama yöntemleriyle yapacağı saldırılar ile kimlik ifşası gerçekleştirebilir. *k*-Anonimlik, *l*-Çeşitlilik ve *t*-Yakınlık modelleri kimlik ve öznitelik ifşalarına karşı koruma sağlarken üyelik ifşalarına karşı koruma sağlayamaz. Bu problemi çözmek adına Nergiz ve ark. [84] δ-Mevcudiyet modelini önermiştir. Önerilen bu modeldeki temel yaklaşım, yayınlanan veri kümesinin saldırganın arka plan bilgisini temsil eden genel veri kümesinin alt kümesi olarak modellenebilmesidir. Örneğin, Çizelge 2.8'de gösterilen 3-Anonim A tablosu Çizelge 2.11'de belirtilen R tablosunun bir alt kümesi olsun. Her ikisi de iki farklı anonim veri kümeleri olmalarına rağmen, R tablosunda yer alan C kişisinin A tablosunda yer alma ihtimali %75 olarak hesaplanabilir. Dolayısıyla burada $\delta=(\delta_{min}, \delta_{max})$ olacak şekilde bir aralık belirlenerek olasılığın bu aralıkta tutulması sağlanır.

Çizelge 2.11. 4-Anonim harici R tablosu

	Posta Kodu	Yaş	Maaş	Cinsiyet
A	476**	[29-43]	[3000-6000]	Erkek
B	476**	[29-43]	[3000-6000]	Erkek
C	476**	[29-43]	[3000-6000]	Erkek
D	476**	[29-43]	[3000-6000]	Erkek
E	47***	[30-40]	[7000-11000]	Kadın
F	47***	[30-40]	[7000-11000]	Kadın
G	47***	[30-40]	[7000-11000]	Kadın
H	47***	[30-40]	[7000-11000]	Kadın
I	478**	[32-55]	[3000-5000]	Erkek
J	478**	[32-55]	[3000-5000]	Erkek
K	478**	[32-55]	[3000-5000]	Erkek
L	478**	[32-55]	[3000-5000]	Erkek

Yukarıda verilen bilgiler ışığında, bu çalışmada mahremiyet koruma modeli olarak *k*-Anonimlik seçilmiştir. Bu modelin uygulanması diğer mahremiyet koruma modellerine göre daha kolay ve işlem maliyeti düşüktür. Ayrıca, bu tezde önerilen modeller literatürdeki ilk

modeller olduğu için, *k*-Anonimlik modelini uygulamanın daha uygun olacağı değerlendirilmektedir.

2.7. Bölümde Sunulan Katkılar

Bu bölümde sunulan katkılar aşağıda listelenmiştir;

- Geleneksel veride; mahremiyet kavramı, mahremiyet koruma süreci ve mahremiyet problemi hakkında bilgi verilmiştir,
- Büyük veride; genel mimari ve teknolojiler, mahremiyet kavramı, mahremiyet koruma süreci ve mahremiyet problemi hakkında bilgi verilmiştir,
- Büyük veri bileşenleri mahremiyet açısından değerlendirilmiştir,
- Mahremiyet tehditleri ve korunma yaklaşımları özetlenmiştir.

3. MATERYAL VE METOT

Bu bölümde, literatürde sıklıkla kullanılan çeşitli anonimleştirme modelleri ve algoritmaları tanıtılmış, bu tez çalışmasında kullanılan Mondrian anonimleştirme modeli sunularak bu modelin üst sınır problemi açıklanmış, sıklıkla kullanılan bilgi metrikleri sunulmuş, veri faydası ile aykırı veri ve normal veri tanımları verilmiş, aykırı ve normal verilerin belirlenmesinde kullanılan yaklaşımlar sunulmuş ve yapılan literatür taramasına yer verilmiştir.

3.1. Anonimleştirme Model ve Algoritmaları

Literatürde pek çok anonimleştirme modelleri ve algoritmaları mevcuttur [9]. Bunlardan sıklıkla kullanılanlar aşağıda kısaca tanıtılmış ve karşılaştırılarak bu çalışmada hangi modelin neden seçildiği belirtilmiştir.

MinGen [88]: optimal genelleştirmeyi belirlemek için muhtemel tüm genelleştirmeleri sorgulayan bir anonimleştirme algoritmasıdır. Verinin satır veya sütun seviyesinde büyümesi halinde algoritmanın efektif olmayacağı belirtilmiştir. *k*-Anonimliği sağlamak amacıyla genelleştirme ve baskılama operatörlerini kullanır.

Incognito [95]: *k*-Anonimliği sağlamak için muhtemel tüm genelleştirmeleri sorgulayan bir anonimleştirme algoritmasıdır. Verinin satır veya sütun seviyesinde büyümesi halinde arama uzayı da üssel olarak artacağı için zaman karmaşıklığı açısından etkin bir yaklaşım olmadığı raporlanmıştır.

Flash [96]: *k*-Anonimliği sağlamada kullanılan optimal bir algoritmadır. Tam genelleştirme kafesinde aşağıdan yukarıya dolaşarak bu kafeste yollar oluşturur ve bu yolları k-Anonimliği sağlama adına kontrol eder. Optimal çözümü bulmaya çalıştığı için yüksek boyutlu ve geniş veri kümleri için uygun bir çözüm olmadığı belirtilmiştir.

Datafly [41]: geniş veri kümeleri üzerinde *k*-Anonimliği sağlamak için geliştirilen ilk yakın optimal algoritmadır. Yarı-tanımlayıcı grubunun büyüklüğü kadar bir dizi üreterek bu dizideki elemanların kombinasyonlarının *k* tekrarlamadan daha az olacak şekilde aç-gözlü olarak genelleştirilmesini sağlar. Genelleştirme ve baskılama operatörlerini uygular.

Bottom-up Generalization (BUG) [42]: minimal *k*-Anonimliği sağlamak için geliştirilen bir algoritmadır. Algoritma ilk olarak orijinal veride *k*-Anonimliği ihlal eden veriden başlar ve her bir aşamada arama metriğine dayanarak bir genelleştirme işlemi uygular. Bir seviyedeki veri eğer bir üst seviyeye genelleştirilecek ise aynı ağaçta ve seviyedeki diğer veriler de bir üst seviyeye genelleştirilir. Genelleştirmesi ve baskılama operatörlerini uygular.

Top-down Specialization (TDS) [43]: bir tablodaki değerleri yukarıdan aşağıya doğru özelleştirerek anonimleştirir. Tüm orijinal veri kümesini alt seviyelerde küçük parçalar haline getirerek anonim gruplar oluşturur. *k*-Anonimliği ihlal eden herhangi bir özelleştirme kalmayana kadar süreç devam eder. Genelleştirme ve baskılama operatörlerini kullanır.

Mondrian [35]: minimal *k*-Anonimliği bulmaya çalışan ve çok boyutlu genelleştirme yapan bir algoritmadır. Yukarıdan aşağıya doğru işlem yaparak tüm veriyi küçük parçalar haline getirir ve sonrasında genelleştirir. Veri parçalama için KD-tree yapısını kullanır. Gerek anonimleştirme için bir hiyerarşi yapısı kullanmaması gerekse de veri uzayını parçalayarak birbirine olabildiği kadar yakın veri grupları oluşturmasından dolayı diğer algoritmalara göre daha çok tercih edilir. Yüksek veri faydası sunması ve zaman karmaşıklığının düşük olması en önemli avantajlarıdır.

Çizelge 3.1. Anonimleştirme model ve algoritmalarının karşılaştırılması

Model/Algoritma	Optimallik	Veri Boyutu	İşlem Yönü	Bölme Stratejisi
MinGen [88]	Optimal	Tekli	Aşağıdan-yukarıya	Hiyerarşik
Incognito [95]	Optimal	Tekli	Aşağıdan-yukarıya	Hiyerarşik
Flash [31]	Optimal	Tekli	Aşağıdan-yukarıya	Hiyerarşik
Datafly [41]	Optimale Yakın	Tekli	Aşağıdan-yukarıya	Hiyerarşik
BUG [42]	Optimale Yakın	Tekli	Aşağıdan-yukarıya	Hiyerarşik
TDS [43]	Optimale Yakın	Tekli	Yukarıdan-aşağıya	Hiyerarşik
Mondrian [35]	Optimale Yakın	Çoklu	Yukarıdan-aşağıya	Parçalı

Çizelge 3.1'de 7 farklı anonimleştirme model ve algoritmasının karşılaştırılmasına yer verilmiştir. Bu tez kapsamında, Mondrian modeli tercih edilmiş, gerekçeleri ise aşağıda sunulmuştur;

- Optimale yakın bir çözüm sunmasından dolayı kabul edilebilir bir zamanda çözüm üretmesi,
- Çok boyutlu anonimleştirmeyi desteklediği için yüksek veri faydası sunması,
- Veri uzayını KD-tree yapısı ile parçalı olarak bölmesinden dolayı genelleştirme için taksonomi ağacına gerek duymaması,
- Taksonomi ağacı kullanmamasından dolayı taksonomi ağacı tabanlı yaklaşımlara göre daha yüksek veri faydası sunması ve
- Yakın optimal bir çözüm olmasından dolayı büyük veri kümeleri için uygulanabilir olmasıdır.

3.2. Mondrian Anonimleştirme Modeli ve Üst Sınır Problemi

Mondrian, literatürde sıklıkla kullanılan ve yüksek veri faydası sunan bir anonimleştirme modelidir. Bu tez çalışmasında, Mondrian modelinden faydalanıldığı için bu model aşağıda detaylı olarak açıklanmış ve bu modelin üst sınır problemi sunulmuştur.

3.2.1. Mondrian Anonimleştirme Modeli

Mondrian modeli, öznitelik uzayında temsil edilen veri kümesini yatayda ve dikeyde yinelemeli bir şekilde bölerek alt veri grupları oluşturur ve bu alt veri gruplarına genelleştirme operatörünü uygulayarak anonimleştirir. Mondrian modelinin algoritması Şekil 3.1'de sunulmuş olup çalışma adımları ise sırasıyla aşağıda listelenmiştir;

1. Öznitelik uzayında temsil edilen veri kümesi için bir boyut seç,
2. Seçilen boyuttaki verilerin frekansını hesapla,
3. Frekansların medyanı bulunarak bölme değerini belirle,
4. Bölme değerine göre veriyi sol ve sağ alt grup olarak iki parçaya böl,
5. Sol veya sağ alt gruplarda bölünecek eleman varsa Adım 1'e git ve
6. Bölünecek eleman yoksa oluşturulan tüm alt grupları genelleştirerek anonim veriyi elde et.

```
1: function Mondrian(Veri, k)
2:     if k ≤ |Veri| < l then
3:         return Genellestir(Veri)
4:     else
5:         dim = Boyut_Sec(veri)
6:         fs = FrekansKumesi(veri, dim)
7:         splitVal = Ortancayi_Bul(fs)
8:         lhs = parca ∈ Veri : parca.dim ≤ splitVal
9:         rhs = parca ∈ Veri : parca.dim > splitVal
10:        return Mondrian(lhs, k) ∪ Mondrian(rhs, k)
11:    end if
12: end function
```

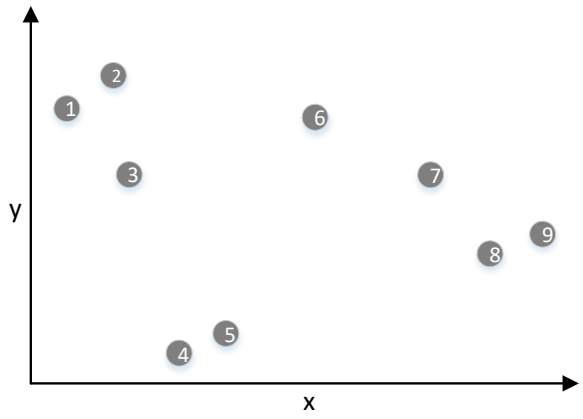
Şekil 3.1. Mondrian algoritması

O(nlogn) zaman karmaşıklığına sahip olan Mondrian algoritması, iki ana aşamadan oluşur. İlk aşamada öznitelik uzayında temsil edilen veri kümesi KD-tree ağacı ile alt gruplara bölünürken, ikinci aşamada ise bu alt gruplardan her birine genelleştirme işlemi uygulanarak anonim veri kümesi elde edilir.

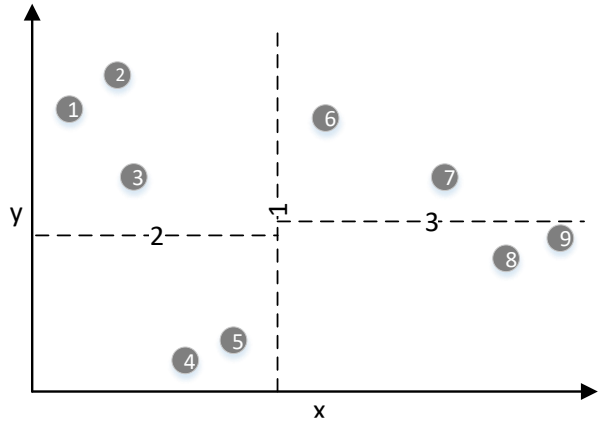
KD-tree ağacı [97], veri uzayını alt gruplara ayırmada kullanılan ve özellikle en yakın komşuluk bulma problemlerinde arama maliyetini düşürmek için tercih edilen ikili bir ağaç yapısıdır. Veri uzayını yinelemeli olarak iki parçaya böler ve irdelenecek herhangi bir veri noktası kalmayana kadar bu işlemi tekrarlar. Boyut (eksen veya öznitelik) tabanlı çalışan bir yapı olup, seçilen herhangi bir boyuttaki verilerin frekanslarının medyanını bularak bu değere göre veriyi yatayda veya dikeyde böler. Bir KD-tree yapısında düğümlerde tutulan bilgiler aşağıdaki gibidir;

- Bölünecek boyut,
- Bölme değeri,
- Bölme değerinden küçük ve eşit verileri içeren sol alt ağaç işaretçisi ve
- Bölme değerinden büyük verileri içeren sağ alt işaretçisidir.

Mondrian modelinde kullanılan KD-tree yapısının çalışma prensibi *k*=2 için aşağıda örnek bir durum üzerinde gösterilmiştir. Şekil 3.2'de gösterilen iki boyutlu örnek bir veri kümesi ele alınsın. Şekil 3.3'de ise, örnek veri kümesi için seçilen boyutlara ve belirlenen bölme değerlerine göre sırasıyla 1, 2 ve 3 numaralı çizgilerle verinin alt gruplara bölünmesi gösterilmiştir.



Şekil 3.2. İki boyutlu örnek veri kümesi

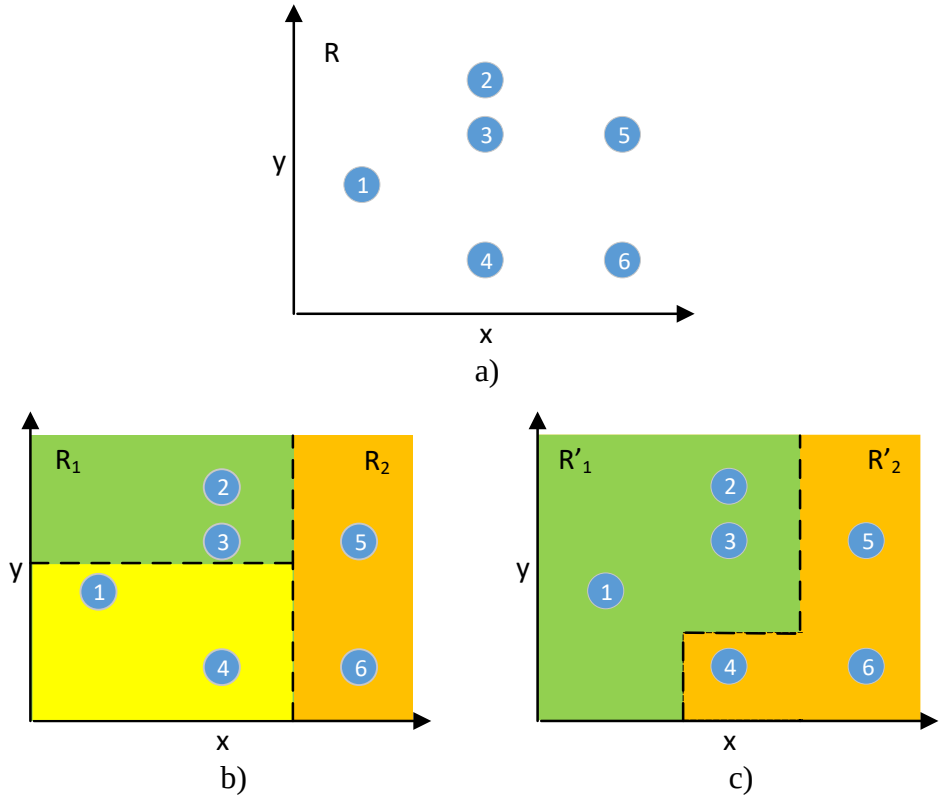


Şekil 3.3. Veri kümesinin KD-tree ile ilk bölünmesi

Mondrian modeli, veri uzayını parçalamak ve araştırmacılara esneklik sağlamak için Strict ve Relaxed olmak üzere iki farklı bölme stratejisi sunar. Bu stratejiler k=2 için, Şekil 3.4'de örnek bir veri üzerinde gösterilmiş olup aşağıda açıklanmıştır;

- Strict bölme: veri uzayının birbiriyle kesişmeyen alt uzaylara bölünmesidir. Şekil 3.4.b'de *x* ekseni dikkate alınarak yapılan bölme işleminde; aynı *x* değerine sahip 2, 3 ve 4 numaralı veriler ile 1 numaralı veri sol gruba, 5 ve 6 numaralı veriler ise sağ gruba atanır. Ancak *y* ekseni için halen bir bölme olduğundan dolayı ikinci bir böle işlemini de gerçekleştirerek 2 ve 3 numaralı verileri sol gruba, 1 ve 4 numaralı verileri ise sağ gruba atar. Bu şekilde üç alt veri grubu elde edilmiş olur.
- Relaxed bölme: veri uzayının birbiriyle kesişen alt uzaylara bölünmesidir. Şekil 3.4.c'de *x* ekseni dikkate alınarak yapılan bölme işleminde; aynı *x* değerine sahip 2, 3 ve 4 numaralı verilerden 2 ve 3 ile 1 numaralı veriler sol gruba, 4, 5 ve 6 numaralı veriler ise

sağ gruba atanmıştır. Bölünecek daha fazla alan kalmadığı için ikici bir bölme yapılamamakta ve dolayısıyla iki alt veri grubu elde edilmektedir.



Şekil 3.4. a) İki boyutlu örnek veri kümesi, b) Strict bölme stratejisi, c) Relaxed bölme stratejisi

Relaxed stratejisi genellikle tek bir yarı-tanımlayıcı olması halinde tercih edilirken, birden fazla yarı tanımlayıcı olması halinde ise Strict stratejisi tercih edilir. Tüm bu stratejiler, seçilen mevcut boyutlar üzerinde işlem yaparak verileri her biri minimum mahremiyet seviyesini sağlayacak şekilde farklı sayıda eleman içeren alt gruplara parçalamaktadır. Böylesi bir durumda, belirlenen mahremiyet seviyesinden daha fazla veri aynı gruba atandığı için veriden elde edilecek muhtemel fayda da düşük olacaktır. Üst sınır problemi olarak bilinen bu sorun sonraki başlıkta detaylı olarak açıklanmıştır.

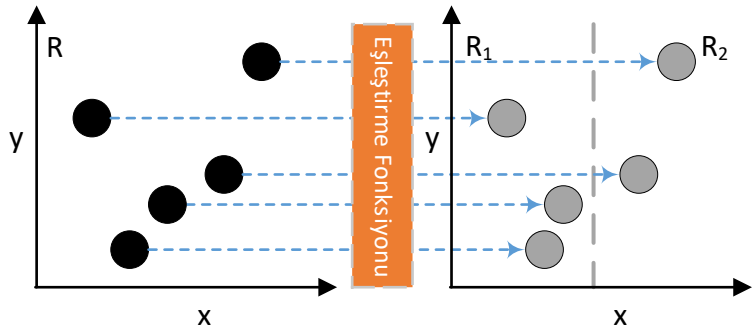
3.2.2. Üst sınır problemi

Mondrian modeli veri uzayını Strict ve Relaxed olmak üzere iki farklı stratejiye uygun olarak böler. Strict bölme stratejisinde birbiriyle kesişmeyen çok boyutlu alanlar oluşurken,

Relaxed bölme stratejisinde ise birbiriyle kesişen çok boyutlu alanlar oluşur. Bu bölmeler sonucu oluşan alt gruplar için; d boyutu ve m ise bir noktanın ilgili alandaki kopya sayısını göstermek üzere, bu grupların maksimum üst sınırı Strict bölme stratejisinde $2d(k-1)+m$ iken Relaxed bölme stratejisinde ise $2k-1$ olmaktadır. Ancak bu sınırlar eşlenik sınıfların büyüklüğünü belirlerken, en önemli dezavantajı ise potansiyel veri faydasını düşürmesidir. Bu problem literatürde üst sınır problemi [98] olarak bilinmektedir.

Üst sınır problemini matematiksel olarak daha iyi açıklayabilmek amacıyla aşağıda çeşitli tanımlar ve şekiller sunulmuştur.

Eşleştirme fonksiyonu: V bir veri uzayı, v ise bir veri noktası ve $v \in V$ olmak üzere; V üzerinde birbirleriyle kesişmeyen $\{R_1, R_2, \dots, R_N\}$ alt uzayları oluşturarak v'yi ilgili alt uzaya atayan fonksiyon bir eşleştirme fonksiyonudur. Şekil 3.5'de, R uzayında bulunan veriler bir eşleştirme fonksiyonu ile R_1 ve R_2 alt uzaylarına eşleştirilmiştir.

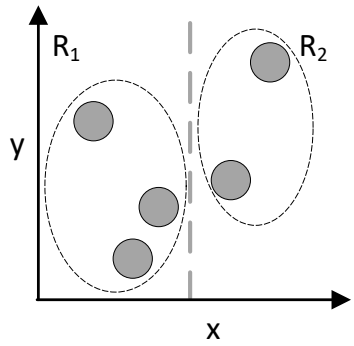


Şekil 3.5. Eşleştirme fonksiyonu için örnek bir gösterim

Sınır: V bir veri uzayı ve m ise V üzerindeki veri noktalarını ilgili $\{R_1, R_2, \dots, R_N\}$ alt uzaylarına eşleştiren bir eşleştirme fonksiyonu olsun. $\{R_1, R_2, \dots, R_N\}$ içerisindeki her bir R_i için; R_i'nin eleman sayısı R_i'nin sınırını temsil eder. Örneğin; Şekil 3.6'da R_1 ve R_2 alt uzaylarının sınırları sırasıyla 3 ve 2 olarak belirlenir.

Üst Sınır: R_i'nin sınırı t_i ve $T = \{t_1, t_2, \dots, t_N\}$ ise $\{R_1, R_2, \dots, R_N\}$'nin sınırlar kümesi olsun. T içerisindeki maksimum t_i değeri, R için üst sınır olarak kabul edilir. Örneğin; Şekil 3.6'da gösterilen alt uzaylar için, R_1'nin sınırı R_2'den daha büyük olduğundan dolayı R'nin üst sınırı 3 olur.

Üst Sınır Problemi: V bir veri uzayı, V'nin üst sınırı t ve k-Anonimlik parametresi k olmak üzere, eğer $k < t$ ise bu durum üst sınır problemi olarak tanımlanır. Örneğin; Şekil 3.6'da gösterilen durum dikkate alındığında, k=2 için, üst sınır 3 olduğundan dolayı $2 < 3$ olduğu görülür ve bu durumda üst sınır problemi ortaya çıkar.

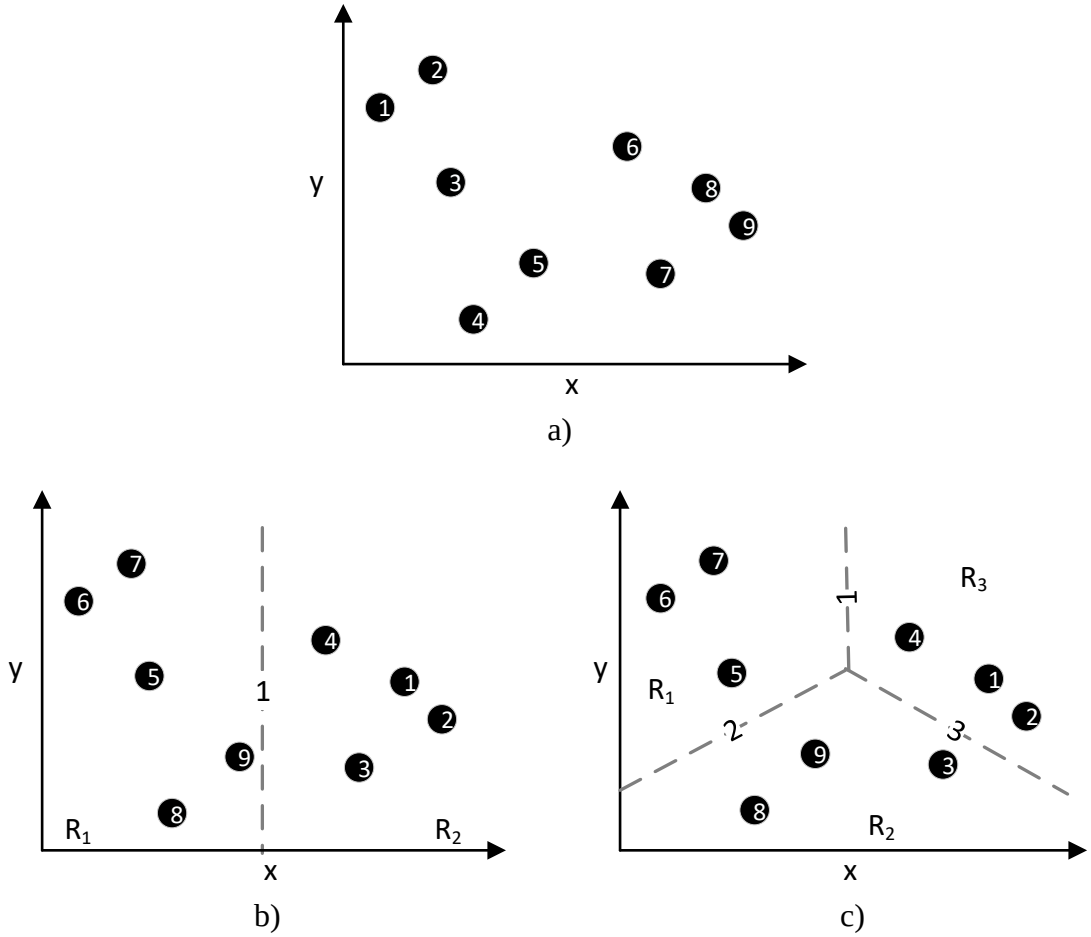


Şekil 3.6. Örnek bir sınır gösterimi

Üst sınır problemi için örnek bir senaryo şu şekilde gösterilebilir. Şekil 3.7.a'da iki boyutlu örnek bir veri kümesi sunulmuştur. Bu veri kümesine Relaxed bölme stratejisi uygulanması durumunda, k=3 için muhtemel bir bölme sonucu oluşacak yapı Şekil 3.7.b'de gösterilmiştir. Verilen bu şekilde R_1 ve R_2 oluşan alt veri uzaylarını temsil etmekte ve bu alt uzayların her biri için alt ve üst limitler mevcut stratejiye göre 3 ve 5 olmaktadır. k=3 için, R_1 uzayının 5 ve R_2 uzayının ise 4 veri noktasına sahip olduğu görülmektedir. Bu durumda, R_1 uzayındaki herhangi 2 nokta ile R_2 uzayındaki herhangi 1 noktanın veri faydası sunmadığı ve dolayısıyla toplam veri faydasını düşürdüğü görülmektedir.

Yukarıda örneklendirilen üst sınır problemi için; Şekil 3.7.b'de gösterilen R_1 ve R_2 alanlarının sınırları detaylı bir şekilde incelenirse, veri kümesinden daha yüksek veri faydası sağlamak için bu alanların sınırlarının optimize edilebileceği açıkça görülebilmektedir. Şekil 3.7.c'de sunulan fayda duyarlı bir bölme yaklaşımında ise daha uygun bir bölme işlemi gerçekleştirilmiş ve veriden elde edilebilecek fayda arttırılmıştır.

Verilen şekilde sunulan her iki bölme yaklaşımı karşılaştırıldığında; Şekil 3.7.b'deki parçalama için veri faydası DM metriği cinsinden ölçülürse; DM = $5^2+4^2=41$ olarak elde edilirken; Şekil 3.7.c'deki parçalama için ise; DM = $3^2+3^2+3^2=27$ olarak bulunur. Daha düşük DM değerinin daha yüksek veri faydası sunduğu bilindiği için, Şekil 3.7.c'deki parçalama diğer parçalamaya kıyasla veri faydası açısından daha uygundur.



Şekil 3.7. a) İki boyutlu örnek veri kümesi, b) Veri kümesine relaxed bölme yaklaşımının uygulanması, c) Veri kümesine fayda duyarlı bir bölme yaklaşımının uygulanması

Yukarıda verilen örneklendirmeye bakıldığında, Mondrian modelinin üst sınır problemi veri faydası açısından üzerinde durulması gereken önemli bir problem olarak karşımıza çıkmaktadır.

Mondrian modelini problem özelinde veri faydasını arttırmak için kullanan ve Mondrian'ın varyasyonlarını öneren çeşitli çalışmalar [98-103] Çizelge 3.2'de gösterilmiştir. Bu çalışmalardan sadece Tang ve ark. [98] Mondrian modelinin üst sınır problemine değinmiş ve bu problemi Relaxed bölme stratejisi kapsamında çözmek amacıyla Mondrian algoritmasından esinlenen yeni bir algoritma önermiş ve test etmiştir. Adult veri kümesinin ve DM metriğinin kullanıldığı çalışmada, önerilen algoritmanın Strict Mondrian, Relaxed Mondrian ve Bottom-up algoritmalarına göre daha yüksek veri faydası sunduğu raporlanmıştır.

Çizelge 3.2. Mondrian modelini iyileştiren çalışmalar

Çalışma	Kapsam	İyileştirme Hedefi	İyileştirme Yaklaşımı	Bölme Stratejisi	k Değerini Garantileme	Veri Dağılımını Dikkate Alma	Üst Sınır Problemini Dikkate Alma
[98]	PPDP	Üst Sınır	Esnek bölme	Relaxed	Hayır	Hayır	Evet
[99]	PPDP	Öznitelikler	Öznitelik ağırlıklandırma	Mevcut değil	Hayır	Hayır	Hayır
[100]	PPDP	Öznitelikler	Eksik veri yönetimi	Mevcut değil	Hayır	Hayır	Hayır
[102]	PPDP	Öznitelikler	Veri değişimi	Mevcut değil	Hayır	Hayır	Hayır
[103]	PPDM	Bölme yaklaşımı	Entropi	Mevcut değil	Hayır	Evet	Hayır
[103]	PPDM	Bölme yaklaşımı	En az kare sapması	Mevcut değil	Hayır	Hayır	Hayır
[103]	PPDM	Bölme yaklaşımı	Belgisizlik	Mevcut değil	Hayır	Hayır	Hayır
Bu tez çalışması	PPDP	Üst Sınır	Yoğunluk ve Yakınlık Tabanlı Bölme	Relaxed ve Strict	Evet	Evet	Evet

Her ne kadar üst sınır problemi sadece Tang ve ark. [98] tarafından ele alınsa da, önerdikleri model ile ilgili değerlendirmelerimiz aşağıda maddeler halinde sunulmuştur;

1. Önerdikleri model sadece Relaxed bölme stratejisi için uygulanabilir olup, Strict bölme stratejisi dikkate alınmamıştır,
2. Önerdikleri model sadece eşlenik sınıfların büyüklüğünü iyileştirmeyi dikkate almış, veri dağılımını dikkate alan herhangi bir yaklaşım sunulmamıştır,
3. Önerdikleri model için üst sınırının $k+1$ olduğu raporlanmıştır.

Bu tez kapsamında üst sınır problemi için sunulan çözüm önerileri ise ilerleyen bölümlerde verilmiştir.

3.3. Bilgi Metrikleri

Anonimleştirme algoritmaları mahremiyet problemine belirli seviyelerde çözüm sunarken, ürettikleri anonim verinin kalitesini ölçmek için çeşitli ölçeklere ihtiyaç duyulur. Bilgi metrikleri, veri mahremiyetinde anonim verinin kalitesini veya faydasını ölçmede kullanılan temel ölçeklerdir. Literatürde çok sayıda bilgi metriği mevcut olup bunlardan bazıları aşağıda özetlenmiş ve bu çalışmada hangi metriğin neden seçildiğine dair gerekçeler verilmiştir.

Bilgi kaybı metriği (ILoss) [104]: bir v değerinin genel bir v_g değerine genelleştirilmesi sonrası oluşan bilgi kaybını ölçen metriktir. Tüm orijinal veri değerlerinin taksonomi ağacında tutulmasını gerektirir. v_g'nin alt ağacındaki elemanların sayısı $|v_g|$, $|D_A|$ ise v_g'nin A özniteliği içerisindeki domain değerlerinin sayısı olmak üzere, v_g için oluşan bilgi kaybı Eş. 3.1'de verilmiştir;

$$ILoss(v_g) = \frac{|v_g| - 1}{|D_A|} \tag{3.1}$$

Genelleştirilmiş bir r verisi için, A özniteliğinin ceza değerini gösteren bir ağırlık w_i olmak üzere, r verisinin bilgi kaybı Eş. 3.2'deki gibi hesaplanır;

$$ILoss(r) = \sum_{v_g \in r} (w_i * ILoss(v_g)) \tag{3.2}$$

Tüm veri kümesi için toplam bilgi kaybı ise Eş. 3.3 kullanılarak hesaplanır;

$$ILoss(T) = \sum_{r \in T} ILoss(r) \tag{3.3}$$

En az bozulma metriği (MD) [31]: genelleştirilen veya baskılanan her bir veriye bir ceza değeri atayan metriktir. Örneğin; 10 adet veri için, herhangi bir özniteliği bir üst seviyeye genelleştirmenin cezası 10 birim olarak hesaplanır.

Eşlenik sınıflar ortalaması (AECS) [35]: eşlenik sınıflarının ortalama büyüklüğünü ölçen bilgi metriğidir. T tablosundaki veri sayısı $|T|$, EC eşlenik sınıf, $|EC|$ eşlenik sınıfların sayısını ve k ise k-Anonimlik parametresini göstermek üzere; AECS metriği Eş. 3.4'de gösterildiği gibi hesaplanır;

$$AECS(T) = \frac{|T|}{|EC| * k} \tag{3.4}$$

Ayırt edilebilirlik metriği (DM) [105]: veri kümesi içerisindeki her bir veriye diğer verilerden ayırt edilememe durumu üzerine bir ceza puanı atayan metriktir. Eğer bir veri, $|T[qid]|$ büyüklüğünde bir qid grubuna ait ise, bu veri için ceza değeri $|T[qid]|$ olurken, ilgili gruba ait ceza ise $|T[qid]|^2$ olarak hesaplanır. Genelleştirilmiş bir T tablosuna ait tüm ceza değeri ise Eş. 3.5'deki gibi hesaplanır;

$$DM(T) = \sum_{qid_i} |T[qid_i]|^2 \tag{3.5}$$

Global kesinlik cezası (GCP) [106]: eşlenik sınıflar içerisindeki elemanların birbirine olan yakınlıklarını ölçen bir metriktir. G bir eşlenik sınıf, A eşlenik sınıftaki sayısal bir yarı-tanımlayıcı öznitelik olmak üzere G'nin NCP (Normalized Certainity Penalty) değeri Eş. 3.6'daki gibi hesaplanır;

$$NCP_A(G) = \frac{max_A^G - min_A^G}{max_A - min_A} \tag{3.6}$$

Bir eşlenik sınıf içerisindeki tüm yarı-tanımlayıcılar için NCP değeri aşağıdaki gibi hesaplanır;

$$NCP(G) = \sum_{i=1}^{d} w_i * NCP_{A_i}(G) \tag{3.7}$$

T anonim tablosu içerisindeki tüm eşlenik sınıflar için, N toplam kayıt sayısı olmak üzere, bu tablonun GCP değeri Eş. 3.8'de gösterildiği gibi hesaplanır;

$$GCP(T) = \frac{\sum_{G \in T} |T| * NCP(G)}{d * N} \quad (3.8)$$

Bu tez kapsamında kullanılan Mondrian modeli eşlenik sınıfları dikkate alan bir model olduğu için, eşlenik sınıfların çeşitli özelliklerini ölçen DM, GCP ve AECS metrikleri bu tez çalışmasında kullanılmıştır.

3.4. Veri Mahremiyeti Kapsamında Aykırı Veri ve Normal Veri Tanımları

Aykırı veri ve normal veri tanımını veri mahremiyeti kapsamında değerlendirmeden önce veri madenciliği kapsamında değerlendirmek, ilgili tanımın her iki alanda da kolay ayırt edilmesini sağlayacaktır.

Aykırı veri ve normal veri kavramları veri madenciliği kapsamında değerlendirildiğinde; aykırı veri, bulunduğu küme içerisinde diğer verilere göre farklı davranışlar sergileyen veri grubu olarak nitelendirilirken; normal veri ise normal davranış sergilediği kabul edilen veri grubu olarak nitelendirilir [107-109].

Aykırı veri ve normal veri kavramları veri mahremiyeti kapsamında değerlendirildiğinde ise; aykırı veri tüm veriden elde edilecek toplam veri faydasını düşürdüğü için veri kümesinden çıkarılan veri grubu iken; normal veri ise tamamından fayda elde edilen veri grubu olarak literatürde tanımlanmıştır [7,8,13,14,110-112].

Veri mahremiyeti kapsamında aykırı veri ve normal veri için yapılan ve yukarıda sunulan genel tanımlar aşağıdaki gibi ifade edilebilir;

Aykırı veri; V veri kümesi, $v \in V$ ve $Fayda()$ ise mahremiyet açısından veri faydasını ölçen bir fonksiyon olmak üzere; eğer $Fayda(V) \leq Fayda(V - v)$ ise v aykırı veridir.

Normal veri; V anonim veri kümesi ve $v \in V$ olmak üzere; eğer v aykırı veri değil ise normal veridir.

Her ne kadar yukarıda belirtilen tanımlar aykırı veri ve normal veri için genel tanımlar olsa da, bu tanımları sağlamak şartı ile uygulamaya özgü güncellenmiş yeni tanımların da

oluşturulması gerekebilir. Çünkü uygulama özelinde kullanılacak olan algoritmalar, bu algoritmaların kullandığı anonimleştirme operatörleri ve aykırı veri tespitinin hangi aşamada yapıldığı uygulamadan uygulamaya farklılık gösterebileceği için, bu gerekçelerden ötürü aykırı veri ve normal veri kavramları da farklı olarak ele alınabilir.

Literatürde veri mahremiyeti kapsamında aykırı veri için yapılan çeşitli tanımlar aşağıda verilmiştir;

- Belirli bir eşik değerinden daha yüksek değere sahip veri grubu [113],
- Diğer verilere uzaklığı bir eşik değerinden daha yüksek olan veri grubu [7,14,112],
- Tamamen baskılanan veri grubu [8,13,102,110,114],
- Mahremiyet kriterini ihlal eden veri grubu [115] ve
- Diğer verilere göre yoğunluğu bir eşik değerinden daha düşük olan veri grubudur [116].

3.5. Veri Mahremiyeti Kapsamında Veri Faydası Tanımı

Anonimleştirmede veri faydası, anonim verinin orijinal veriye olan yakınlığı olarak ifade edilebilir [9]. Bu yakınlık ne kadar fazla olursa, anonim veri üzerinde geliştirilecek olan modelin doğruluğu ve başarısı, orijinal veri üzerinde geliştirilecek olan modelin doğruluğu ve başarısına o kadar yakın olur. Bir veri kümesinin orijinal halinin %100 veri faydası sunduğu kabul edilirse, anonimleştirme sonrası orijinal veriye kıyasla kayıplı bir veri kümesi elde edileceği için anonim veri orijinal veriye göre daha düşük veri faydası sunacaktır.

Anonimleştirme kapsamında veri faydası aşağıdaki şekilde tanımlanabilir;

Veri faydası: V orijinal veri kümesi, V_A anonim veri kümesi, $Yakınlık(x, y)$ ise x ile y'nin birbirine yakınlığını ölçen bir fonksiyon ve $u \geq 0$ olmak üzere; $u = Yakınlık(V, V_A)$ için, V_A'nın sunduğu veri faydası u olarak kabul edilir.

Literatürde yer alan genel bilgi metrikleri yukarıda verilen $Yakınlık()$ fonksiyonu olarak kullanılabilir. Bu tezde kapsamında, tercih edilen Mondrian modeli doğrudan eşlenik sınıfları dikkate aldığından dolayı, veri faydasını ölçmek için eşlenik sınıfların çeşitli özelliklerini değerlendiren DM, GCP ve AECS metrikleri bu tez kapsamında kullanılmıştır.

DM metriği eşlenik sınıfların büyüklüğünü, AECS metriği eşlenik sınıflar içerisindeki ortalama eleman sayısını ve GCP metriği ise eşlenik sınıflardaki elemanların birbirine olan yakınlıklarını ölçer. Çeşitli uygulamalar için DM ve AECS ile ölçülen veri faydası yeterli olabilirken, bu tez kapsamında GCP metriği de dikkate alınmıştır.

3.6. Veri Mahremiyeti Kapsamında Aykırı Verilerin Belirlenmesi ve Yönetimi

Literatürde, veri mahremiyeti kapsamında aykırı verilerin belirlenmesi için uzaklık tabanlı [7,14,112], taksonomi ağacı tabanlı [8,13,102,110,111,114] ve yoğunluk tabanlı [116] çeşitli yaklaşımlar mevcuttur. Uzaklık tabanlı yaklaşımlarda, öznitelik uzayında temsil edilen veriler üzerinde bir uzaklık fonksiyonu kullanılarak; taksonomi ağacı tabanlı yaklaşımlarda en üst seviyede genelleştirilen veya baskılanan veriler bulunarak; yoğunluk tabanlı yaklaşımlarda ise yoğunluğu belirli bir eşik değerinden düşük olan veriler bulunarak aykırı veriler belirlenir. Her bir yaklaşım kendi içerisinde detaylı yöntem ve teknikleri barındırabilir.

Taksonomi ağacı, herhangi bir öznitelik değerinin alt-üst ilişkisini gösterdiği için özellikle genelleştirme ve özelleştirme gibi anonimleştirmenin alt aşamalarında sıklıkla kullanılan bir yapıdır. Taksonomi ağacının kullanıcı tanımlı olması, aykırı verilerin oluşmasında en belirgin faktör olması ve veri dağılımını dikkate almaması en önemli dezavantajlarındandır. Uygulamadan bağımsız, veriyi ve dağılımını dikkate almayan taksonomi ağaçlarının kullanımı veriden elde edilecek faydayı da düşürecektir. Klasik yaklaşımlarda aykırı veriler veri kümesinden çıkarılırken, bazı yaklaşımlarda bunlardan fayda elde etme yoluna gidilmiştir.

Mahremiyet risklerini arttıran ve veri faydasını olumsuz etkileyen aykırı verilerin anonimleşme sürecinde yönetilmesi gerekir. Aykırı verilerin yönetilmesi amacıyla literatürde kullanılan mevcut yaklaşımlar aşağıda maddeler halinde verilmiştir;

1. Aykırı verilerin anonimleştirme öncesi tespit edilerek tamamının yayınlanacak veri kümesinden çıkarılması [113],
2. Aykırı verilerin anonimleştirme sonrası tespit edilerek tamamının yayınlanacak veri kümesinden çıkarılması [115],

3. Aykırı verilerin anonimleştirme sonrası tespit edilerek global aykırı verilerin yayınlanacak veri kümesinden çıkarılması ve yerel aykırı verilerin en yakın normal veri gruplarına eklenmesi [7,14,112],
4. Aykırı verilerin anonimleştirme sonrası tespit edilerek fayda elde edilebilecek olanlarının yeniden kullanımı ve fayda elde edilemeyecek olanların ise yayınlanacak veri kümesinden çıkarılması [8,13,110],
5. Aykırı verilerin anonimleştirme öncesi tespit edilerek tamamından fayda elde edilmesi [116] ve
6. Aykırı verilerin anonimleştirme sonrası tespit edilerek değerlerinin değiştirilip yeniden kullanılmasıdır [114].

Yukarıda belirtilen 1, 2 ve 3 numaralı yaklaşımlarda sadece mahremiyetin korunması; 4, 5 ve 6 numaralı yaklaşımlarda ise mahremiyetin korunarak veri faydasının arttırılması amaçlanmıştır. Ayrıca, 6 numaralı yaklaşımda aykırı veri değerlerinde veri değişimi yapıldığı için verinin güvenilirliği konusunda da çeşitli sorunların ön plana çıktığı değerlendirilmektedir.

3.7. Yerel Aykırılık Faktörü Algoritması

Yerel Aykırılık Faktörü (Local Outlier Factor - LOF) algoritması [117], literatürde aykırı veri tespitinde sıklıkla kullanılan bir algoritmadır. Verilere yoğunluklarına göre aykırılık skoru atayarak, belirlenen bir eşik değerinden yüksek skora sahip olanların aykırı, düşük olanların ise normal veri olarak kabul edilmesini sağlar. Hesaplama maliyeti en iyi durumda O(nlogn) en kötü durumda ise $O(n^2)$'dir.

Şekil 3.8'de sunulan LOF algoritmasının genel adımları sırasıyla şu şekildedir;

1. Tüm veriler arasındaki uzaklığın hesaplanması,
2. k. en yakın komşuya olan uzaklığın hesaplanması,
3. k en yakın komşuluk kümesinin oluşturulması,
4. k en yakın komşuluktaki verilere toplam erişim uzaklığının hesaplanması,
5. yerel erişilebilirlik uzaklıklarının (LRD) hesaplanması ve
6. LOF değerlerinin hesaplanmasıdır.

```
function LOF(Veri, k)
    for all veri ∈ Veri do
        Uzaklik(veri, Veri) = Uzaklik_Hesapla(veri, Veri)
        k_Uzaklik(veri) = k_EnYakinKomsuyaUzaklik_Bul(veri, Veri, k)
        k_EnYakinKume(veri) = EnYakinVerileri_Getir(Uzaklik(veri, Veri), k_Uzaklik(veri))
        for all eleman ∈ k_EnYakinKume(veri) do
            topla(veri) = topla(veri) + ErisimUzakligi(eleman, veri)
        end for
        LRD(veri) = |k_EnYakinKume(veri)|/topla(veri)
    end for
    for all veri ∈ Veri do
        LOF(veri) = topla(LRD(k_EnYakinKume(veri))/LRD(veri)/|k_EnYakinKume(veri)|
    end for
end function
```

Şekil 3.8 LOF algoritması

Bu tez kapsamında LOF algoritmasının tercih edilmesinin sebepleri aşağıda listelenmiştir [117];

1. Tüm verilere bir aykırılık skoru atayarak kullanıcıya esneklik sağlaması,
2. Hesaplama maliyetinin uygun olması,
3. Tek bir parametre gerektirdiği için parametre karmaşıklığının az olması,
4. Doğrudan aykırı veri tespitine yönelik olarak geliştirilmiş olması,
5. Hem global hem lokal aykırılıkları tespit edebilmesi ve
6. Veri yoğunluğu hakkında bir ön kabul yapmamasıdır.

3.8. Literatür Taraması

Bu bölümde, büyük veride mahremiyet koruma ile aykırı veri tespiti ve yönetimi üzerine yapılan çalışmalar incelenmiş ve özetlenmiştir.

3.8.1. Büyük veri mahremiyeti ile ilgili yapılan güncel çalışmalar

Literatürde, büyük veri mahremiyeti konusunu ele alan çeşitli çalışmalar mevcuttur. Bu çalışmalardan güncel olanlar; önerdikleri yöntemler, kullandıkları teknik ve yaklaşımlar, kullandıkları veri kümeleri ve performans değerlendirme kriterleri açısından incelenmiş ve özetleri ise aşağıdaki paragraflarda sunulmuştur.

Büyük veri kümelerinin anonimleştirmesi için hibrit bir yaklaşımın önerildiği [118] numaralı çalışmada, TDS ve BUG algoritmaları MapReduce yapısına uygun olarak geliştirilmiştir.

Önerilen yaklaşımın mevcut yaklaşımlara kıyasla alt-ağaç anonimleştirme tekniğinin ölçeklenebilirliğini ve etkinliğini önemli derecede arttırdığı gözlemlenmiştir. Adult veri kümesi kullanılarak Hadoop üzerinde yapılan deneylerde, çalışma zamanı ve *k* parametresine göre elde edilen sonuçlar için önerilen metodun merkezi TDS ve BUG algoritmalarına göre daha performanslı olduğu raporlanmıştır.

Büyük ölçekli veri kümelerini anonimleştirmek için ölçeklenebilir iki aşamalı TDS yaklaşımının önerildiği [87] numaralı çalışmada, TDS yaklaşımı MapReduce yapısına uygun olarak Hadoop üzerinde geliştirilmiş ve Adult veri kümesi ile test edilmiştir. Anonimleştirme işlemi iki aşamaya bölünmüş olup, ilk aşamada orijinal veri kümeleri küçük gruplara bölünerek paralel olarak anonimleştirilmiş ve ara sonuçlar elde edilmiştir. İkinci aşamada ise, bu ara sonuçlar birleştirilerek tutarlı *k*-anonim veri kümeleri elde etmek için çözümler üretilmiştir. Önerilen yaklaşım, merkezi TDS algoritması ile karşılaştırılmış ve daha iyi sonuçlar elde edildiği belirtilmiştir. Performans kıyaslamalarında çalışma zamanı, veri büyüklüğü, *k* parametresi, bilgi kaybı ve bölünen alt grup sayısı gibi parametreler dikkate alınmıştır.

Büyük veri mahremiyetini sağlamak için ölçeklenebilir çok boyutlu bir anonimleştirme yaklaşımı [52] numaralı çalışmada sunulmuştur. MapReduce yapısını kullanan ölçeklenebilir bir algoritma önerilmiştir. Birden fazla MapReduce işi tasarlanmış ve veri kümeleri üzerinde çok boyutlu anonimleştirme için koordine edilmiştir. Hadoop üzerinde Adult veri kümesi ile yapılan deneylerde, her öznitelik için değişim katsayısı hesaplanmıştır. Çalışma zamanı ve veri büyüklüğüne göre yapılan performans değerlendirmesinde, literatürdeki çeşitli tekniklere göre yüksek performans elde edildiği raporlanmıştır.

Büyük veri anonimleştirme için hücre genelleştirme problemi [119] numaralı çalışmada ele alınmıştır. Hassas değerlerin ve çoklu hassas özniteliklerin anlamsal yakınlığını kullanarak bir yakınlık mahremiyet modelinin önerildiği bu çalışmada, hücre genelleştirme problemi yakınlık-duyarlı kümeleme problemi olarak modellenmiştir. Belirtilen problemi çözmek adına iki aşamalı bir kümeleme yaklaşımı önerilmiştir. Ölçeklenebilirliği sağlamak adına MapReduce yapısına uygun bir model Hadoop üzerinde geliştirilmiştir. Adult veri kümesinin kullanıldığı çalışmada, çalışma süresi, veri sayısı ve düğüm sayısı parametreleri dikkate alınarak bir performans değerlendirmesi yapılmıştır.

Büyük veride çevrimdışı anonimleştirmenin değerlendirildiği [120] numaralı çalışmada, anonimleştirme sistemlerinin güvenlik ve hız gereksinimleri belirtilerek anonimleştirmede kullanılan popüler teknikler açıklanmıştır. Telekomünikasyon verilerinin kullanıldığı bu çalışmada, geliştirilen anonimleştirme yaklaşımı Apache Storm üzerinde uygulanmıştır.

Büyük veri anonimleştirme için *k*-anonimlik mahremiyet modelinin kullanıldığı [121] numaralı çalışmada, büyük verinin ölçeklenebilirlik problemini çözmek için MapReduce yapısına uygun iki aşamalı bir TDS modeli geliştirilmiştir. Hadoop üzerinde Adult verisi kullanılarak yapılan deneysel çalışmalarda, parça sayısı ve çalışma süresi performans kriteri olarak belirlenmiştir.

k-Anonimlik mahremiyet koruma modeli için Hadoop üzerinde TDS algoritmasının MapReduce yapısı ile geliştirildiği [49] numaralı çalışmada, büyük veri mahremiyeti ve bunun ölçeklenebilirliği tartışılmıştır. Önerilen yaklaşımda ilk olarak büyük veri kümesi küçük gruplara bölünerek anonimleştirilmiş, sonrasında ise bu anonim veri kümeleri birleştirilerek tek bir veri kümesi haline getirilmiştir.

Hassasiyet tabanlı büyük veri anonimleştirme işleminin yapıldığı [122] numaralı çalışmada, çok boyutlu hassasiyet tabanlı anonimleştirme adı verilen bir metot Hadoop üzerinde geliştirilmiştir. Adult veri kümesinin kullanıldığı bu çalışmada, *k*-Anonimlik modeli uygulanarak eşlenik sınıfı oranı ve veri sayısı parametrelerine göre bir analiz yapılmıştır.

Büyük veri anonimleştirme işlemi için genelleştirme yaklaşımının temel alındığı [53] çalışmasında, yerel kayıt tabanlı mekanizma adı verilen bir mahremiyet koruma modeli önerilmiştir. Önerilen yapıyı test etmek adına büyük veri için bir mahremiyet metriği geliştirilmiştir. Bir ağ simülatöründen toplanan veri üzerinde yapılan işlemlerde veri sayısı, çalışma süresi, kullanıcı sayısı ve önerilen metrik dikkate alınarak analiz işlemleri yapılmıştır.

Büyük veride anonimleştirme için Hadoop üzerinde *k*-Anonimlik ve *l*-Çeşitlilik modellerini uygulayan MapReduce tabanlı algoritmaların geliştirildiği [123] numaralı çalışmada, PokerHand ve sentetik veri kümeleri analiz edilerek anonimleştirme işlemi gerçekleştirilmiştir. Bilgi kaybı, *k* ve *l* parametreleri dikkate alınarak gerçekleştirilen

deneylerde başarılı sonuçlar elde edildiği belirtilmiş, literatürdeki iki farklı algoritmaya kıyasla önerilen yaklaşımın daha performanslı olduğu raporlanmıştır.

MapReduce tabanlı yerel hassas özetleme metoduna dayalı yeni bir yaklaşım [124] numaralı çalışmada önerilmiştir. İki veri kaydının arasındaki anlamsal uzaklığı elde etmek için yeni bir anlamsal uzaklık metriği geliştirilmiş ve özyinelemeli yığınsal k-üyeli kümeleme algoritması kullanılmıştır. Hadoop üzerinde Adult verisi ile yapılan deneylerde, çalışma süresi, veri sayısı, bilgi kaybı, düğüm sayısı gibi parametreler dikkate alınarak performans değerlendirmeleri yapılmıştır.

Büyük veride anonimleştirme işlemlerinin Apache Spark teknolojisi tabanlı yapılabileceği ilk defa [125] numaralı çalışmada ele alınmış ve bu teknoloji üzerine gerçekleştirilecek anonimleştirme işlemlerinde nasıl bir yol izleneceği rapor edilmiştir.

Büyük veri bileşenlerinin mahremiyet kapsamında değerlendirilmesi ve mahremiyet-fayda probleminin büyük veriyi de kapsayacak şekilde genişletilmesi ilk defa [77] numaralı çalışmada ele alınmıştır. Ayrıca bu çalışmada büyük veride mahremiyet tehditleri ve mahremiyet koruma modelleri de irdelenmiş olup mahremiyet korumalı veri yayınlama geleneksel veri ve büyük veri kapsamında karşılaştırılmıştır.

Mahremiyet korumalı büyük veri yayınlama için ölçeklenebilir *k*-Anonimlik modelinin uygulandığı [126] numaralı çalışmada, veri kümesi öncelikle küçük eşlenik sınıflara bölünmekte sonrasında ise mahremiyet seviyesini sağlamak için birleştirilmektedir. Apache Pig platformunun kullanıldığı bu çalışmada, Adult ve Pokerhand veri kümesi ile test yapılmış ve GCP metriği ile çalışma zamanına göre performans değerlendirmesi yapılmıştır.

Büyük veride anonimleştirme için ölçeklenebilir *l*-Çeşitlilik modelinin uygulandığı [127] numaralı çalışmada, PokerHand veri kümesi kullanılarak testler yapılmış ve önerilen model literatürdeki diğer modellerle karşılaştırılarak daha yüksek performans sergilediği raporlanmıştır. GCP metriği ve çalışma zamanı performans değerlendirmelerinde kullanılmıştır.

Mahremiyet korumalı büyük veri yayınlama kapsamında yukarıda verilen literatür incelendiğinde;

1. Testlerde genel olarak Adult veri kümesinin kullanıldığı,
2. Büyük veri işleme teknolojisi olarak Hadoop anaçatısının tercih edildiği ve bu platform üzerinde modellerin geliştirildiği,
3. Mahremiyet seviyesi, bilgi kaybı, çalışma zamanı, düğüm sayısı ve veri boyutu gibi kriterlere göre performans değerlendirmelerinin yapıldığı,
4. *k*-Anonimlik ve *l*-Çeşitlilik mahremiyet koruma modellerinin uygulandığı,

görülmüş olsa da;

1. Büyük veri mahremiyetinde aykırı verilerin etkisini inceleyen bir çalışmanın olmadığı ve
2. Büyük veri mahremiyetinde aykırı verileri yöneterek veri faydasını arttıran bir modelin henüz önerilmediği tespit edilmiştir.

3.8.2. Veri mahremiyeti kapsamında aykırı verilerin belirlenmesi ve yönetimi ile ilgili yapılan çalışmalar

Aykırı verilerin anonimleştirilmiş kategorik veriler üzerindeki etkisini inceleyen [14] numaralı çalışmada, aykırı veriler belirli uzaklık ölçütlerine göre yerel ve global olarak ikiye ayrılmıştır. Veri anonim hale getirildikten sonra tüm aykırı veriler belirlenmiş ve bu aykırı veriler yerel ve global olmak üzere iki sınıfa ayrılmıştır. Bu işlemden sonra veri kümesi global aykırı verilerden temizlenmiş, yerel aykırı veriler ise kendilerine en yakın normal verilerin olduğu gruplara atanmıştır. Yapılan testlerde Adult veri kümesinden faydalanılmıştır.

Aykırı verilerin gizlenebilir ve gizlenemez olarak ikiye ayrıldığı [7] numaralı çalışmada, mevcut veriler içerisine gizlenemeyen aykırı veriler veri kümesinden çıkarılmış, gizlenebilir olan aykırı veriler ise güncellenen veri havuzundan çeşitli verilerin alınmasıyla *l*-Çeşitlilik şartını sağlamada kullanılmıştır. Önerilen model sayesinde aykırı verileri içeren veri kümelerinin düşük bilgi kayıplarıyla anonim hale getirilebildiği raporlanmıştır. Gerçekleştirilen çalışmada nüfus istatistik verileri kullanılmış, uzaklık tabanlı aykırı veri tespiti yapılmış ve bilgi kaybı metriğinden faydalanılmıştır.

Veri kümesindeki aykırı verilerin fark edilebilirlik saldırısı ile mahremiyetinin ifşa edilebileceğinin belirtildiği [112] numaralı çalışmada, aykırı veriler yerel ve global olarak iki sınıfa ayrılmıştır. Global aykırı veriler veri kümesinden çıkarılmış, yerel aykırı veriler ise normal veri içerisine aktarılarak yeniden düzenlenen gruplar oluşturulmuştur. *k*-Anonimlik mahremiyet koruma modelinin uygulandığı bu çalışmada, nüfus istatistik verileri kullanılmış ve bilgi kaybı metriği ile kıyaslamalar yapılmıştır.

Veri kümesindeki aykırı verilerin çıkarılması yerine bu verilerden fayda elde edilmesinin sağlanması [8,13,110] numaralı çalışmalarda açıklanmış, aykırı verilerin kendi içerisinde anonimleştirilerek elde edilecek veri faydasının arttırılabileceği belirtilmiştir. Adult ve Demografik veri kümelerinin kullanıldığı çalışmalarda, önerilen model sayesinde aykırı verilerden mümkün olabildiğince fayda elde edilmesi üzerinde durulmuştur. *k*-Anonimlik ve *l*-Çeşitlilik modelleri uygulanarak başarılı sonuçlar elde edilmiştir. Ortalama eşlenik sınıf büyüklüğü, ortalama aykırı veri eşlenik sınıf büyüklüğü, veri sayısı ve sınıf sayısı gibi kriterlere göre değerlendirmeler yapılmıştır.

k-Anonimlik standardını bozan kayıtların aykırı veri olarak tanımlandığı [102] numaralı çalışmada, bu kayıtların anonimleştirilebilmesi için başka kayıtlarla yer değiştirilmesini öneren yeni bir anonimleştirme modeli geliştirilmiştir. Census veri kümesinin kullanıldığı bu çalışmada, kayıp metriği kullanılarak önerilen modelin sunduğu veri faydası değerlendirilmiştir. Önerilen model, çeşitli varyasyonları ile karşılaştırılmış ve test sonuçları raporlanmıştır.

Aykırı verilerin anonimleştirme standardını ihlal eden veri grubu olarak nitelendirildiği [113,115] çalışmalarında ise, aykırı veriler veri faydası sunmadığı gerekçesi ile veri kümesinden çıkarılmıştır.

[116] numaralı çalışmada, aykırı veriler yoğunluk tabanlı bir yaklaşımla belirlenerek bunlardan fayda elde edilmeye çalışılmıştır. LOF algoritması kullanılarak veriler yoğunluklarına göre skorlanmış ve bir eşik değerinden düşük yoğunluğa sahip veriler aykırı veri olarak kabul edilmiştir. Aykırı veriler kendi içlerinde iteratif bir şekilde ayrıştırılarak veri faydası mümkün olabildiği kadar arttırılmaya çalışılmıştır. Anonimleştirme için Mondrian algoritmasının kullanıldığı bu çalışmada testlerde Adult veri kümesinden faydalanılmıştır. DM ve AECS metrikleri performans değerlendirmesinde kullanılmıştır.

Aykırı verileri irdeleyen yukarıdaki mahremiyet koruma yaklaşımlarından elde edilen çıkarımlar aşağıda listelenmiştir;

1. Önerilen modellerin test edilmesi için genel olarak Adult veri kümesinin kullanıldığı,
2. *k*-Anonimlik ve *l*-Çeşitlilik modellerinin tercih edildiği,
3. Bilgi kaybı, kayıp metriği, DM ve AECS gibi metriklerinin kullanıldığı,
4. Aykırı verilerin tamamınından fayda üreten çeşitli çalışmalar olsa da bu çalışmaların farklı yaklaşımlarla güncellenmesi gerektiği ve
5. Veri yer değişimi yapılarak veri güvenilirliğinin azaltıldığı değerlendirilmektedir.

3.9. Kullanılan Veri Kümelerinin Tanıtılması

Tez kapsamında yapılan testlerde, literatürdeki anonimleştirme çalışmalarında defakto-standart olarak kullanılan ve UCI Makine Öğrenmesi Deposunda bulunan Adult veri kümesi [128] tercih edilmiştir. Eksik veri yönetimi kapsamında Adult veri kümesi içerisinde bulunan 18680 adet eksik veri, veri kümesinden çıkarılmıştır. Bu veri kümesi hakkında özet bir bilgi Çizelge 3.3'de sunulmuştur.

Çizelge 3.3. Adult veri kümesi özet bilgisi

Özellik	Açıklama
Veri sayısı	48842
Öznitelik sayısı	14
Öznitelik karakteristiği	Kategorik ve nümerik
Sınıf sayısı	2
Eksik veri içeren veri sayısı	18680
Eksik veri içermeyen veri sayısı	30162

İkinci olarak tercih edilen veri kümesi ise UCI Makine Öğrenmesi Deposunda bulunan ve diyabet hastalarına ait olan Diyabet veri kümesidir [128]. Eksik veri içeren 6 özniteliğe ait veriler, veri kümesinden çıkarılmıştır. Bu veri kümesi hakkında özet bir bilgi Çizelge 3.4'de sunulmuştur.

Çizelge 3.4. Diyabet veri kümesi özet bilgisi

Özellik	Açıklama
Veri sayısı	101766
Öznitelik sayısı	55
Öznitelik karakteristiği	Kategorik ve nümerik
Sınıf Sayısı	2
Eksik veri içeren öznitelik sayısı	6
Eksik veri içermeyen öznitelik sayısı	49

3.10. Bölümde Sunulan Katkılar

Bu bölümde sunulan katkılar aşağıda listelenmiştir;

- Literatürdeki anonimleştirme modelleri ve algoritmaları karşılaştırılmış ve Mondrian anonimleştirme modeli ile bu modelin üst sınır problemi sunulmuştur,
- Literatürde sıklıkla kullanılan çeşitli bilgi metrikleri özetlenmiştir,
- Veri mahremiyeti kapsamında literatürdeki aykırı veri ve normal veri tanımları verilmiştir,
- Literatürdeki aykırı veri belirleme ve yönetim yaklaşımları özetlenmiştir,
- Yoğunluk tabanlı aykırı veri tespit algoritması olan LOF açıklanmıştır,
- Literatürdeki büyük veri anonimleştirme çalışmaları ile veri mahremiyeti kapsamında aykırı verileri dikkate alınan çalışmalar özetlenmiş ve
- Son olarak bu tezde kullanılan veri kümeleri tanıtılmıştır.

4. ÖNERİLEN AYKIRI VERİ YÖNELİMLİ FAYDA TEMELLİ ANONİMLEŞTİRME MODELİ: U-MONDRIAN

Bir önceki bölümde, Mondrian anonimleştirme modeli ve bu modelin üst sınır problemi tanıtılmış, literatürde veri mahremiyeti kapsamında kabul edilen aykırı veri tanımlarına ve bu verilerin nasıl yönetildiğine dair bilgiler verilmiştir.

Bu bilgiler ışığında, bu bölümde ise Mondrian anonimleştirme modeline aykırı veri konsepti uygulayarak veri faydasını arttırmayı amaçlayan ve u-Mondrian (utility-aware Mondrian) olarak isimlendirilen yeni bir anonimleştirme modeli önerilmiş, geliştirilmiş ve test edilmiştir. Önerilen model için aykırı veri konsepti kapsamında oluşturulan ve kabul edilen aykırı veri ve normal veri tanımları sunulmuştur. Aykırı veri yönetiminin veri faydası üzerindeki etkisini değerlendirmek için karar kuralları yapısından faydalanılarak veri faydası karar kuralları oluşturulmuştur. Önerilen modelde aykırı verilerin belirlenmesi için yoğunluk ve yakınlık tabanlı hibrit yeni bir yaklaşım önerilmiş, geliştirilmiş ve uygulanmıştır. Ayrıca aykırı verilerin yönetilmesi için kullanılan aykırı veri yönetim yaklaşımı açıklanmıştır. Son olarak, yapılan deneysel çalışmalarda, çeşitli veri kümeleri, bölme stratejileri, *k* değerleri ve fayda metrikleri dikkate alınarak önerilen model test edilmiş ve doğrulanmıştır.

4.1. Önerilen Model İçin Kabul Edilen Aykırı Veri ve Normal Veri Tanımları

u-Mondrian modeli Mondrian modeline aykırı veri konsepti uygulayarak veri faydası odaklı genişlettiğinden dolayı, u-Mondrian modeli kapsamında aykırı veri ve normal veri tanımları için yapılan kabuller aşağıda verilmiştir;

Normal veri; V bir veri kümesi, $v_T \in V$ bir alt küme, $k \leq |v_T| \leq p$ olmak üzere, v_T kümesi içerisinde birbirine en yakın *k* adet veri normal veridir.

Aykırı veri; V bir veri kümesi, $v_T \in V$ bir alt küme, $k \leq |v_T| \leq p$ olmak üzere, v_T kümesi içerisinde normal veri haricinde kalan veri grubu aykırı veridir.

4.2. Aykırı Veri Yönetiminin Veri Faydasına Etkisini Ölçmek İçin Veri Faydası Karar Kurallarının Oluşturulması

Bu tezde Mondrian modeline aykırı veri konsepti uygulanarak veri faydası odaklı iyileştirilmeye çalışıldığı için, aykırı veri yönetimi ile veri faydası arasındaki ilişkinin gösterilmesi gerekmektedir. Bu kapsamda, DM, AECS ve GCP metrikleri dikkate alınarak oluşturulan karar kuralları aşağıda tanıtılmıştır.

V orijinal veri kümesi, N normal veri kümesi, O aykırı veri kümesi ve $V = N \cup O$ olmak üzere; U aykırı veri yönetimi yapılmadan ölçülen veri faydasını, U' aykırı veri yönetimi yapıldıktan sonra ölçülen veri faydasını, N_i $i.$ iterasyondaki normal veri kümesini, n durdurma kriteri sağlandığı andaki iterasyon sayısını, O_n ise $n.$ iterasyondaki aykırı veri kümesini ve ES ise eşlenik sınıf sayısını temsil etsin. Bu durumda, aykırı veri yönetimi ile veri faydası arasındaki ilişki, karar kuralı yapısı kullanılarak gösterilebilir. DM, AECS ve GCP metrikleri dikkate alınarak oluşturulan A, B ve C karar kuralları aşağıda tanıtılmıştır.

DM metriğine göre oluşturulan A karar kuralı aşağıda verilmiştir;

$U = DM(V)$,

$U' = \sum_{i=1}^{n} DM(N_i) + DM(O_n)$,

olmak üzere, oluşturulan A karar kuralı Eş. 4.1'de gösterilmektedir;

$$A = \begin{cases} 1, eğer\ U' < U \\ 0, diğer\ durumlarda \end{cases} \tag{4.1}$$

AECS metriğine göre oluşturulan B karar kuralı aşağıda verilmiştir;

$U = \frac{|V|}{ES(V)}$,

$U' = \frac{|V|}{\sum_{i=1}^{n} ES(N_i) + ES(O_n)}$,

olmak üzere, oluşturulan B karar kuralı Eş. 4.2'de gösterilmektedir;

$$B = \begin{cases} 1, eğer\ U' < U \\ 0, diğer\ durumlarda \end{cases} \tag{4.2}$$

GCP metriğine göre oluşturulan C karar kuralı ise aşağıda verilmiştir;

$U = GCP(V)$,

$U' = \sum_{i=1}^{n} GCP(N_i) + GCP(O_n)$,

olmak üzere, oluşturulan C karar kuralı ise Eş. 4.3'de sunulmuştur;

$$C = \begin{cases} 1, eğer\ U' < U \\ 0, diğer\ durumlarda \end{cases} \quad (4.3)$$

A, B ve C karar kurallarının her biri için; kural çıktısının 1 olması veri faydasının arttığını, 0 olması ise veri faydasının aynı kaldığı veya azaldığını göstermektedir. Dolayısıyla bu karar kuralları kullanılarak önerilen modellerin sunduğu veri faydası değerlendirilebilir.

4.3. Önerilen Modelde Aykırı Verilerin Belirlenmesi ve Yönetilmesi

u-Mondrian modelinde aykırı veri ve normal verilerin belirlenmesi için, bu verilerin tanımları ile ilgili yapılan kabullere uygun bir yaklaşımın kullanılması gerekir. Bu bölümde, yapılan kabullere uygun olarak geliştirilen aykırı veri belirleme yaklaşımı arkasındaki metodolojilerle beraber tanıtılmış ve açıklanmış, ve ayrıca önerilen modelde aykırı verilerin yönetilmesinde kullanılan aykırı veri yönetim yaklaşımı sunulmuştur.

4.3.1. Aykırı verilerin belirlenmesinde yeni bir hibrit yaklaşım önerisi

u-Mondrian modelinde, öznitelik uzayında temsil edilen veri kümesinin KD-tree ile alt uzaylara bölünmesi sonucu elde edilen ve eleman sayıları minimum *k* ve maksimum *p* olan alt veri gruplarının her birinde; birbirine en yakın *k* adet veri normal veri, arta kalan veriler ise aykırı veri olarak kabul edilmektedir. Bir veri grubu içerisinde birbirine en yakın *k* adet veriyi içeren grubun tespit edilmesi kombinasyonel bir arama uzayı gerektirdiği için, girdi sayısının artması ile beraber arama uzayı da üssel olarak artacaktır. Bu tür problemlere polinomsal zamanda kabul edilebilir çözümler üretmek için yakın optimal çözümlere ihtiyaç duyulur.

Anonimleştirme kapsamında, birbirine en yakın *k* adet veriyi içeren alt veri grubunun tespit edilmesi için ilk defa yoğunluk ve yakınlık tabanlı hibrit bir yaklaşım bu bölümde önerilmiş, geliştirilmiş ve uygulanmıştır. Yoğun bir bölgenin seyrek bir bölgeye göre birbirine daha

yakın veri noktalarını içerdiği bilinmektedir. Bundan dolayı, yoğunluk tabanlı bir yaklaşım kullanılarak birbirine daha yakın veri noktaları belirlenebilir. Önerilen bu hibrit yaklaşımda kullanılan bazı tanımlar aşağıda verilmiştir.

Yoğunluk tabanlı referans noktası: V bir veri kümesi, $yoğunluk\_skorlama()$ verileri yoğunluklarına göre skorlayan bir fonksiyon olmak üzere; $max(yoğunluk\_skorlama(V))$ fonksiyonuyla elde edilen maksimum yoğunluğa sahip veri noktası yoğunluk tabanlı referans noktası olarak kabul edilir. Bu tanım Eş. 4.4'de matematiksel olarak ifade edilmiştir;

$$ref = \begin{cases} max(yoğunluk\_skorlama(V)), & eğer\ üst\_sınır > k \\ yok, & diğer\ durumlarda. \end{cases} \quad (4.4)$$

k-1 en yakın komşu (NN^{k-1}): V veri kümesi içerisinde yoğunluk tabanlı bir referans noktası ref, $v \in V$ ve x ile y noktaları arasındaki uzaklığı ölçen bir fonksiyon $uzaklık(x, y)$ olmak üzere; $min(uzaklık(ref, v))$ fonksiyonu kullanılarak tespit edilen minimum uzaklığa sahip $k-1$ adet veri noktası, ref için $k-1$ en yakın komşu olarak kabul edilmektedir. Bu tanıma ait matematiksel ifade Eş. 4.5'de sunulmuştur.

$$NN^{k-1} = \begin{cases} min(uzaklık(ref, (V - ref)))^{k-1}, & eğer\ üst\_sınır > k \\ yok, & diğer\ durumlarda. \end{cases} \quad (4.5)$$

Önerilen hibrit yaklaşımda, öncelikle LOF algoritması kullanılarak veri kümesindeki noktalar yoğunluklarına göre skorlanır. LOF algoritması ile elde edilen bu skorlar verinin yoğunluğu ile ters orantılıdır. Yani LOF skoru büyük olan veri noktasının yoğunluğu az, LOF skoru küçük olan veri noktasının yoğunluğu ise fazladır. Bu şekilde LOF algoritması $yoğunluk\_skorlama()$ fonksiyonu olarak kullanılabilir.

Her bir noktanın yoğunluk skoru belirlendikten sonra, birbirine en yakın noktaların belirlenmesi için maksimum yoğunluğa (minimum LOF skorlarına) sahip k adet verinin tespit edilmesi gerekir. Ancak, veri kümesinde farklı yoğunlukta veri grupları olması halinde, LOF algoritması lokal yoğunlukları dikkate aldığından dolayı benzer yoğunluk skorlarını farklı gruplardaki elemanlara atayabilir. Bu durumda sadece yoğunluk tabanlı bir yaklaşımı uygulamanın yeterli olmadığı görülmektedir.

Bu durum için örnek bir senaryo şu şekilde verilebilir. Bir veri kümesinde X ve Y olmak üzere farklı yoğunluklarda iki veri grubu bulunsun. Bu veri kümesine LOF algoritmasının uygulanması sonrası X ve Y kümelerindeki verilere benzer yoğunluk skorları atanabilir. Bu durumda, birbirine en yakın *k* adet veriyi içeren grubu belirlerken sadece LOF skorlarının dikkate alınması halinde hem X kümesinden hem de Y kümesinden elemanlar seçilmiş olabileceği için, gerçekte birbirine uzak veri noktaları bu gruba dâhil olabilir. Ancak birbirine en yakın *k* adet verinin ya X kümesinden ya da Y kümesinden seçilmesi daha tutarlı bir yaklaşım olacaktır.

Bu sorunu çözmek için önerilen yaklaşımda ilk olarak, LOF skoru minimum yani yoğunluğu maksimum olan veri noktası referans noktası olarak kabul edilir. Sonra, bu nokta ile alt gruptaki diğer veriler arasındaki uzaklık bir uzaklık fonksiyonu yardımı ile hesaplanır. Referans noktasına uzaklığı minimum olan *k-1* adet veri noktası tespit edilerek en yakın *k-1* komşu bulunmuş olur. Bu şekilde referans noktası da dâhil olmak üzere, birbirine en yakın *k* adet veriyi içeren grup belirlenir.

Alt veri uzaylarında aykırı veri ve normal verilerin belirlenmesi matematiksel olarak şu şekilde ifade edilebilir;

V veri uzayının birbiriyle kesişmeyen alt uzayları $\{R_1, R_2, \dots, R_N\}$ olsun. $\{R_1, R_2, \dots, R_N\}$ içerisindeki her bir R_i için, R_i içerisindeki aykırı veriler ve normal veriler Eş. 4.6 ve Eş. 4.7'de gösterildiği gibi belirlenir;

$$Normal_i = \{ref_i \cup NN_i^{k-1}\} \tag{4.6}$$

$$Aykırı_i = \{R_i - Normal_i\} \tag{4.7}$$

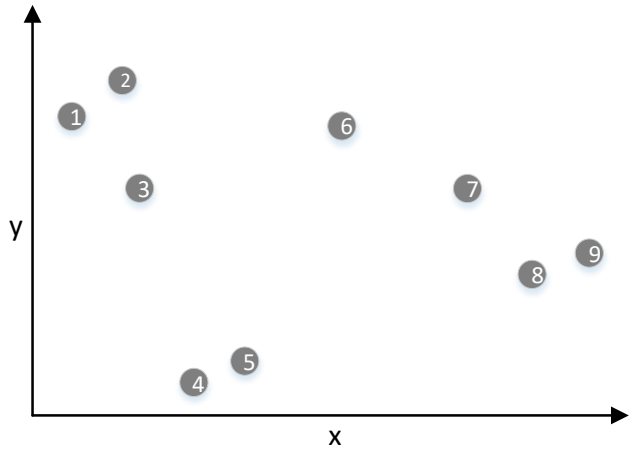
Her bir R_i alt uzayında normal veri ve aykırı veriler belirlendikten sonra tüm V veri kümesi içerisindeki normal veri kümeleri ve aykırı veri kümeleri Eş. 4.8 ve Eş. 4.9'da gösterildiği gibi oluşturulur;

$$Normal = \bigcup_{i=1}^{N} Normal_i \tag{4.8}$$

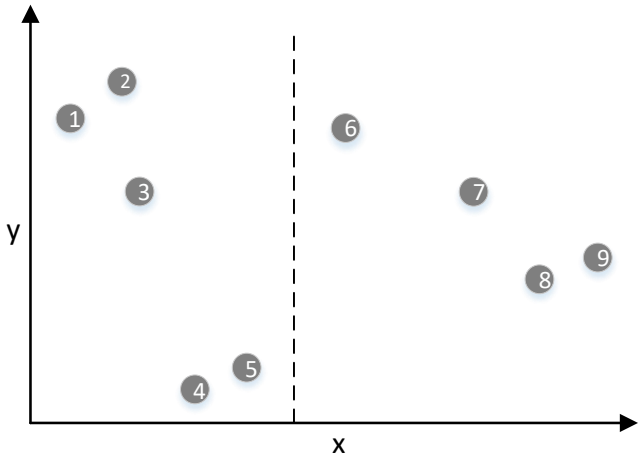
$$Aykırı = \bigcup_{i=1}^{N} Aykırı_i \tag{4.9}$$

Yukarıda önerilen hibrit yaklaşımı daha iyi açıklayabilmek için aşağıda örnek bir senaryo gösterilmiştir.

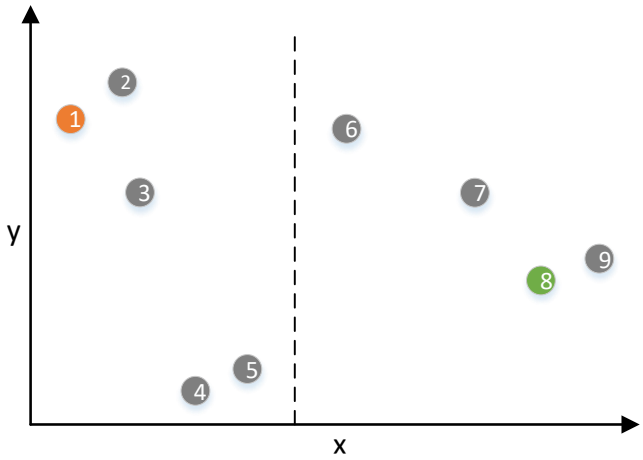
İki boyutlu düzlemde temsil edilen örnek bir veri kümesi Şekil 4.1'de gösterilmektedir. Bu veri kümesine k=3 için KD-tree yapısının uygulanması durumunda elde edilen alt veri grupları Şekil 4.2'de verilmiştir. Verilen şekilde her bir alt veri grubuna LOF algoritmasının uygulanmasıyla elde edilen LOF skorları kullanılarak belirlenen referans noktaları, Şekil 4.3'de 1 ve 8 numara ile gösterilen veriler olarak kabul edilsin. Şekil 4.4'de, 1 numaralı referans noktasına en yakın veriler 2 ve 3; 8 numaralı referans noktasına en yakın veriler ise 7 ve 9 numaralı veriler olur. Şekil 4.5'de, 1, 2 ve 3 numaralı verileri içeren grup ile 7, 8 ve 9 numaralı verileri içeren grup normal veri olarak kabul edilirken, arta kalan 4, 5 ve 6 numaralı veriler ise aykırı veri olarak kabul edilir.



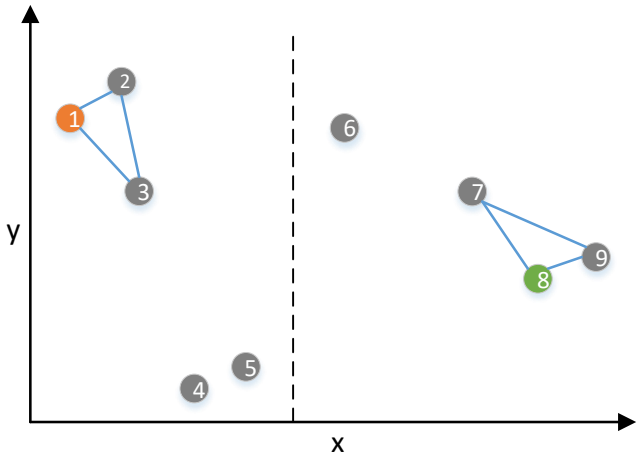
Şekil 4.1. İki boyutlu örnek veri kümesi



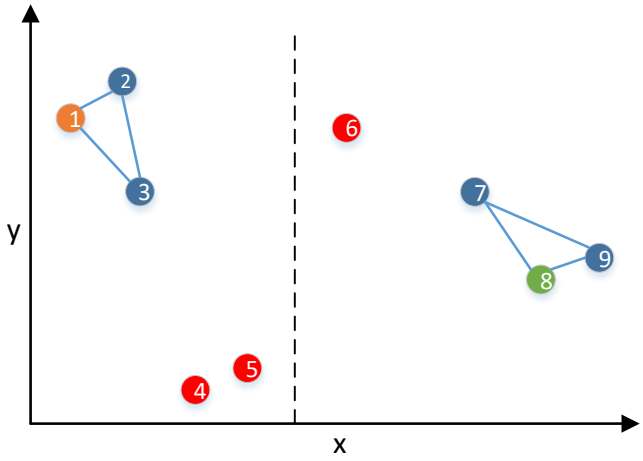
Şekil 4.2. *k*=3 için veri uzayının KD-tree yapısı ile parçalanması



Şekil 4.3. Alt veri uzaylarında LOF skorlarına göre referans noktalarının belirlenmesi



Şekil 4.4. *k*=3 için referans noktalarına en yakın *k-1* komşulukların bulunması



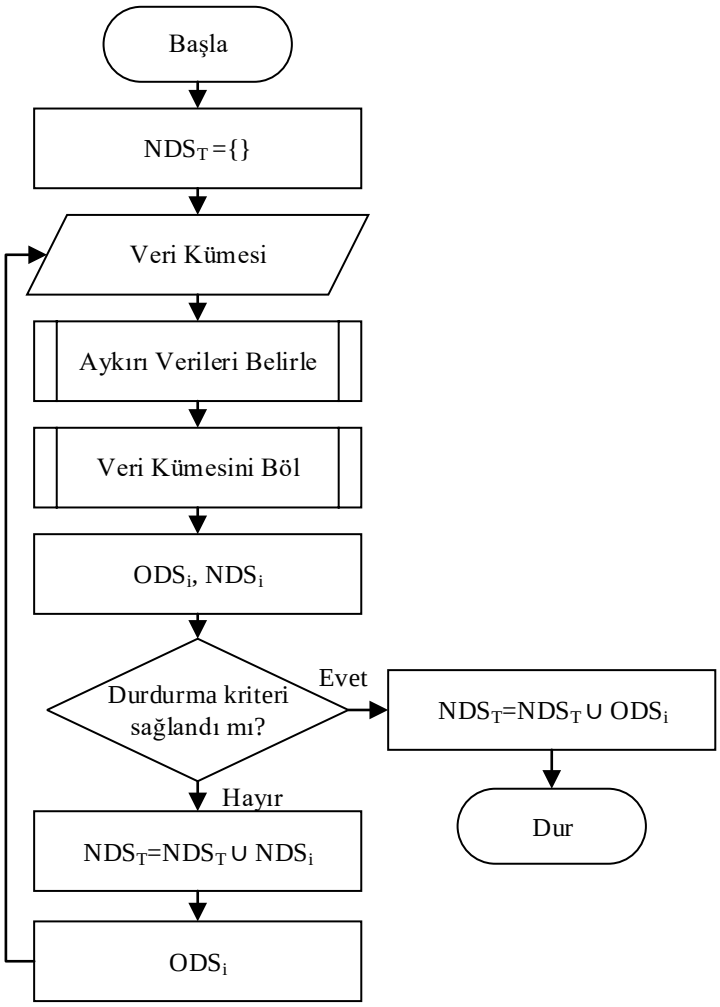
Şekil 4.5. Aykırı veri ve normal verilerin belirlenmesi

4.3.2. Aykırı veri yönetimi

u-Mondrian modelinde aykırı veri yönetimi, aykırı veri kümesinin yinelemeli olarak yeni alt kümelere bölünmesi ve herhangi bir alt kümenin çıkarılmadan tamamının kullanılmasıyla gerçekleştirilir. Aykırı veri kümesinin yinelemeli olarak bölünmesi, bu verilerin kendi içerisinde daha iyi ayrıştırılmasını sağlayarak veri faydasına olumlu katkı sağlar.

Önerilen modelde kullanılan aykırı veri yönetim yaklaşımı Şekil 4.6'da gösterilmektedir. Bu yaklaşım [110] numaralı çalışmadan alınmış ve uyarlanarak önerilen modelde kullanılmıştır. Uyarlama işlemi elde edilen son aykırı veri kümesinin silinmeden tekrar kullanılması ile sağlanmıştır.

Şekil 4.6'da gösterilen aykırı veri yönetim yaklaşımında, ilk olarak veri kümesindeki aykırı veriler ve normal veriler belirlenir. Bu veriler dikkate alınarak mevcut veri kümesi; normal veri kümesi (normal data set - NDS) ve aykırı veri kümesi (outlier data set - ODS) olmak üzere ikiye bölünür. Mevcut iterasyondaki NDS_i kümesi NDS_T kümesine eklenirken, ODS_i kümesi ise sorgulanacak yeni veri kümesi olarak kabul edilir. Devamında, ODS_i kümesi için yukarıda açıklanan süreç devam ettirilerek yeni NDS_{i+1} ve ODS_{i+1} alt kümeleri oluşturulur. Bu süreç bir durdurma kriteri sağlanana kadar devam eder. Bu şekilde ODS kümesinde herhangi bir eleman kalmadan aykırı verilerin tamamı NDS_T kümesine aktarılmış olur.



Şekil 4.6. u-Mondrian modelindeki aykırı veri yönetim yaklaşımının akış diyagramı

4.4. Önerilen Anonimleştirme Modeli: u-Mondrian

Mondrian modeli, KD-tree ve Genelleştirme olmak üzere iki aşamadan oluşur. KD-tree aşamasında, veri kümesi çeşitli kısıtlar altında alt gruplara bölünürken, oluşturulan bu veri grupları ise Genelleştirme aşamasında anonimleştirilir.

Şekil 4.7'de matematiksel ifadesi verilen Mondrian modeli şu şekilde çalışır; X, D boyutlu bir veri kümesi olsun. İlk aşamada, X'e KD-tree yapısı uygulanarak $\{P_1, P_2, \dots, P_T\} \in X$ alt veri grupları elde edilir. İkinci aşamada ise $\{P_1, P_2, \dots, P_T\} \in X$ içerisindeki her bir P_i alt veri grubu $gen()$ fonksiyonu ile genelleştirilerek X^G anonim veri kümesi oluşturulur.

u-Mondrian ise, klasik Mondrian modeline aykırı veri konsepti uygulayarak genişleten ve toplam veri faydasını arttırmayı amaçlayan bir anonimleştirme modelidir. Bu kapsamda u-

Mondrian modeli, Mondrian modelindeki KD-tree ve Genelleştirme aşamaları arasına "Aykırı Veri Belirleme" aşamasını ekleyerek Mondrian'ı genişletir. Bu aşamanın amacı Mondrian modeline aykırı veri konsepti uygulayarak birbirine daha yakın ve k adet veri içeren eşlenik sınıflar oluşturup veri faydasını arttırmaktır. u-Mondrian modelinde, KD-tree ve Aykırı Veri Belirleme aşamaları iteratif bir şekilde çalıştırılarak veri faydası mümkün olabildiği kadar arttırılır.

Girdi: veri kümesi $X = \{x_1, x_2, \dots, x_N\}$; $x_N \in R^D$, k-Anonimliğin k parametresi
Başlangıç değer ataması: $veri = X$
1) KD-tree uygula:
Yinele:
- Boyut seç; $boyut = \{d \in D,\ d = max_{aralık}(veri, D)\}$
- Frekans hesapla; $f_{boyut} = \left\{\{f_1, f_2, \dots, f_M\} = calc_f(veri, boyut)\right\}$
- Medyan bul; $\mu = \{m \in veri_{boyut}, m = med(f_{boyut})\}$
- Veriyi böl; $lhs = \{l \in veri,\ l_{boyut} \leq \mu\}$
$rhs = \{l \in veri,\ l_{boyut} > \mu\}$
- Durdurma kriteri sağlanana kadar; $veri = lhs$ ve $veri = rhs$ için tekrarla

Dönder:
- $\{P_1, P_2, \dots, P_T\} \in X$ parçalarını dönder

2) Genelleştir: $\{P_1, P_2, \dots, P_T\} \in X$ içerisindeki her bir P_i için;
Yinele:
- Genelleştir; $P_i^G = gen(P_i)$

Dönder:
- $\{P_1^G, P_2^G, \dots, P_T^G, \} \in X^G$ anonimleştirilmiş veri parçalarını dönder

Çıktı: X^G anonim veri kümesi

Şekil 4.7. Mondrian modelinin matematiksel gösterimi

Şekil 4.8'de matematiksel ifadesi verilen u-Mondrian modeli şu şekilde çalışır; X, D boyutlu bir veri kümesi olsun. İlk aşamada, X'e KD-tree yapısının uygulanması ile $\{P_1, P_2, \dots, P_T\} \in X$ alt veri grupları elde edilir. İkinci aşamada, $\{P_1, P_2, \dots, P_T\} \in X$ içerisindeki her bir P_i için, aykırı veriler ve normal veriler belirlenir. Aykırı veriler $AykırıKümesi$'nde tutulurken, normal veriler ise $NormalKümesi$'nde tutulur. $AykırıKümesi$ iteratif bir şekilde işlenerek elde edilen normal veriler $NormalKümesi$'ne eklenir ve böylece fayda sağlayan küme oluşturulur. Üçüncü aşamada ise, $\{P_1, P_2, \dots, P_T\} \in NormalKümesi$ içerisinde her bir P_i alt veri grubu $gen()$ fonksiyonu ile genelleştirilerek X^G anonim veri kümesi oluşturulur.

Girdi: veri kümesi $X = \{x_1, x_2, \dots, x_N\}$; $x_N \in R^D$, *k*-Anonimliğin *k* parametresi
Başlangıç değer ataması: $veri = X$
1) KD-tree uygula:
Yinele:
- Boyut seç; $boyut = \{d \in D,\ d = max_{aralık}(veri, D)\}$
- Frekans hesapla; $f_{boyut} = \left\{\{f_1, f_2, \dots, f_M\} = calc_f(veri, boyut)\right\}$
- Medyanı bul; $\mu = \{m \in veri_{boyut}, m = med(f_{boyut})\}$
- Veriyi böl; $lhs = \{l \in veri,\ l_{boyut} \leq \mu\}$
$rhs = \{l \in veri,\ l_{boyut} > \mu\}$
- Durdurma kriteri sağlanana kadar; $veri = lhs$ ve $veri = rhs$ için tekrarla

Dönder:
- $\{P_1, P_2, \dots, P_T\} \in X$ parçalarını dönder

2) Aykırı verileri belirle: $\{P_1, P_2, \dots, P_T\} \in X$ için
Yinele:
- Her bir $v \in P_i$ noktasını skorla; $skor = \{\{s_1, s_2, \dots, s_R\} = LOF(P_i)\}$
- $ref_i \in P_i$ referans noktasını belirle; $ref_i = \{g = P_i\{min(skor)\}\}$
- ref_i'nin $k-1$ en yakın komşuluğunu bul; $NN_i^{k-1} = \{x: (k-1) = argmin\|x - ref_i\|\}$
- Normal veriyi belirle; $normal_i = ref_i \cup NN_i^{k-1}$
- Aykırı veriyi belirle; $aykırı_i = P_i - normal_i$
- $normal_i$'i $NormalKümesi$'ne ekle
- $aykırı_i$'yı $AykırıKümesi$'ne ekle

Değerlendir:
- Eğer $AykırıKüme = \emptyset$ ise Adım 3'e git
değilse $veri = AykırıKüme$ ve Adım 1'e git

3) Genelleştir: $\{P_1, P_2, \dots, P_T\} \in NormalKümesi$ içerisindeki her bir P_i için
Yinele:
- Genelleştir; $P_i^G = gen(P_i)$

Dönder:
- $\{P_1^G, P_2^G, \dots, P_T^G, \} \in X^G$ anonimleştirilmiş veri parçalarını dönder

Çıktı: X^G anonim veri kümesi

Şekil 4.8. u-Mondrian modelinin matematiksel gösterimi

Şekil 4.9'da algoritması verilen u-Mondrian modelinin zaman karmaşıklığı aşağıda belirtilmiştir. M en dıştaki döngünün iterasyon sayısı, N veri kümesindeki eleman sayısı ve *k* ise *k*-Anonimlik parametresi olmak üzere;

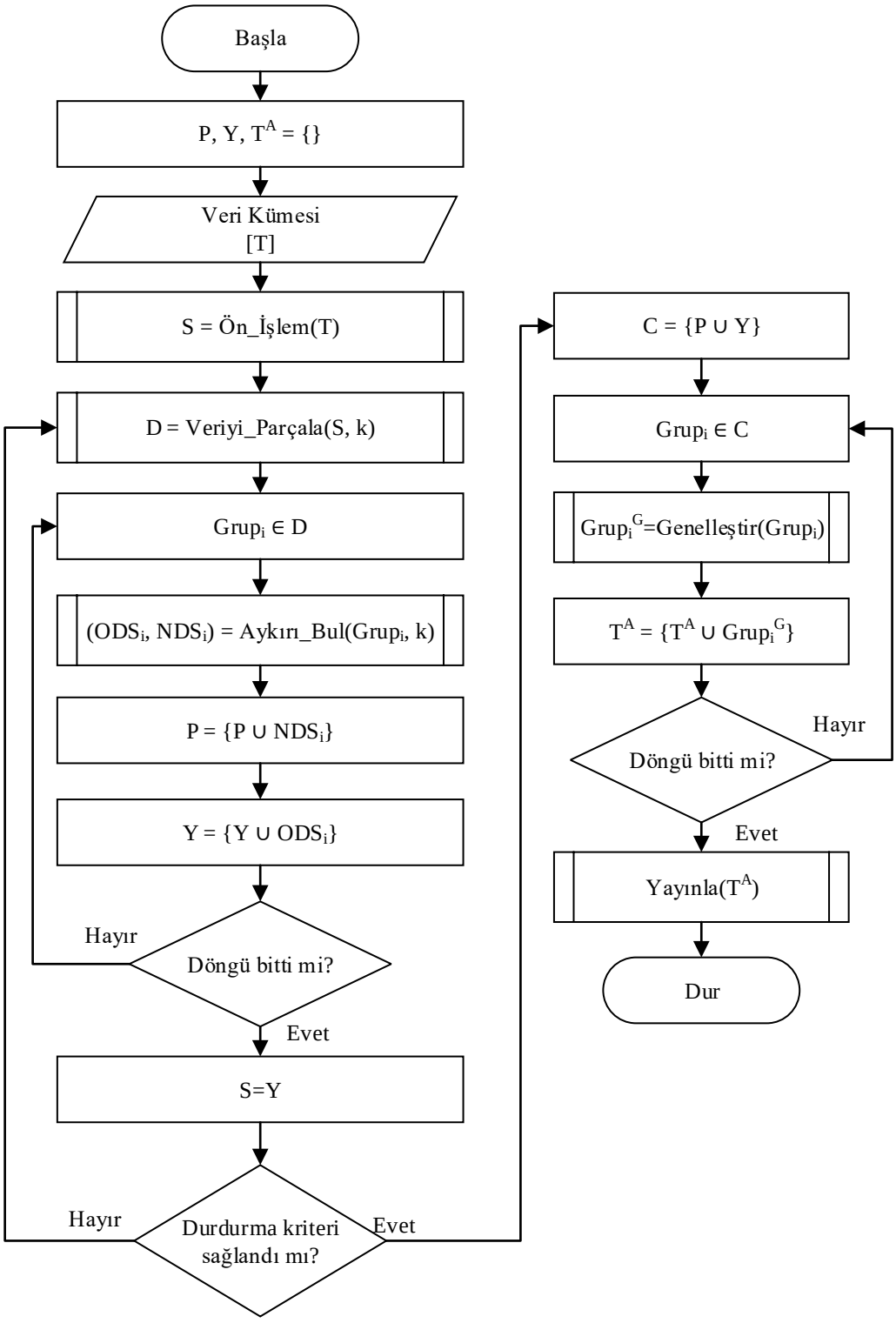
En kötü durum analizi:$O(M * \left(NlogN + \frac{N}{k}N^2\right))$ olarak tespit edilmiştir. KD-tree algoritmasının zaman karmaşıklığı $NlogN$, elde edilen parça sayısı $\frac{N}{k}$ ve LOF algoritması için en kötü durumda zaman karmaşıklığı N^2 olmak üzere; M, N'den çok küçük olduğu için ihmal edilebilir durumda olup, elde edilen karmaşıklık $O(\frac{N^3}{k})$ olur.

En iyi durum analizi: $O(M * \left(NlogN + \frac{N}{k}NlogN\right))$ olarak tespit edilmiştir. KD-tree algoritmasının zaman karmaşıklığı $NlogN$, elde edilen parça sayısı $\frac{N}{k}$ ve LOF algoritması için en iyi durumda zaman karmaşıklığı $NlogN$ olmak üzere; M, N'den çok küçük olduğu için ihmal edilebilir durumda olup, elde edilen karmaşıklık $O(\frac{N^2logN}{k})$ olur.

Mondrian modelinin zaman karmaşıklığı $O(NlogN)$'dir. Her ne kadar önerilen u-Mondrian modeli Mondrian modeline göre daha yüksek zaman karmaşıklığına sahip olsa da, veri faydası açısından Mondrian'a üstünlük sağlar. Önerilen modelde, veri faydasının iyileştirilmesi öncelikli olduğu için zaman karmaşıklığı ikinci planda tutulmaktadır.

```
1: P, Y, T^A ← ∅
2: S ← On_Islem(Veri)
3: function u_Mondrian(S)
4:     while durdurma_kriteri == false do
5:         D ← Veriyi_Parcala(S, k)
6:         for all grup ∈ D do
7:             (ODS_i, NDS_i) ← Aykiri_Verileri_Bul(grup, k)
8:             P ← P ∪ NDS_i
9:             Y ← Y ∪ ODS_i
10:        end for
11:        S ← Y
12:    end while
13:    C ← P ∪ Y
14:    for all grup ∈ C do
15:        grup^G ← Genellestir(grup)
16:        T^A ← T^A ∪ grup^G
17:    end for
18:    Yayinla(T^A)
19: end function
```

Şekil 4.9. u-Mondrian modelinin algoritması



Şekil 4.10. u-Mondrian modelinin akış diyagramı

u-Mondrian modeline ait akış diyagramı Şekil 4.10'da gösterilmekte olup, akıştaki her adım aşağıda sırasıyla açıklanmıştır;

1. T veri kümesi modele girdi olarak verilir,

2. Ön işlem aşamasında özniteliklerin sınıfları belirlenir, normalizasyon işlemi yapılır, durdurma kriteri belirlenir ve S veri kümesi elde edilir,
3. S veri kümesi KD-tree yapısı ile her biri minimum *k* maksimum *p* adet veri içeren alt gruplara bölünerek bu grupları içeren D kümesi elde edilir,
4. D kümesi içerisindeki her bir grup için;
    i. Aykırı veriler ve normal veriler belirlenerek NDS_i ve ODS_i kümeleri oluşturulur,
    ii. NDS_i, P kümesine eklenir,
    iii. ODS_i, Y kümesine eklenir,
5. Eğer durdurma kriteri sağlanmadıysa irdelenecek yeni veri kümesi S = Y olur ve Adım 3'e gidilir,
6. Eğer durdurma kriteri sağlandıysa, P kümesi Y ile birleştirilerek C kümesi elde edilir,
7. C kümesindeki her bir grup için;
    i. Mevcut grup genelleştirilir,
    ii. Genelleştirilen grup T^A kümesine eklenir,
8. T^A anonim veri kümesi yayınlanır.

Mondrian ve u-Mondrian modellerinin çalışma süreçlerindeki farkını daha iyi kavrayabilmek amacıyla oluşturulan blok diyagramlar sırasıyla Şekil 4.11 ve Şekil 4.12'de gösterilmekte olup, bu şekiller aşağıdaki kabuller çerçevesinde oluşturulmuştur;

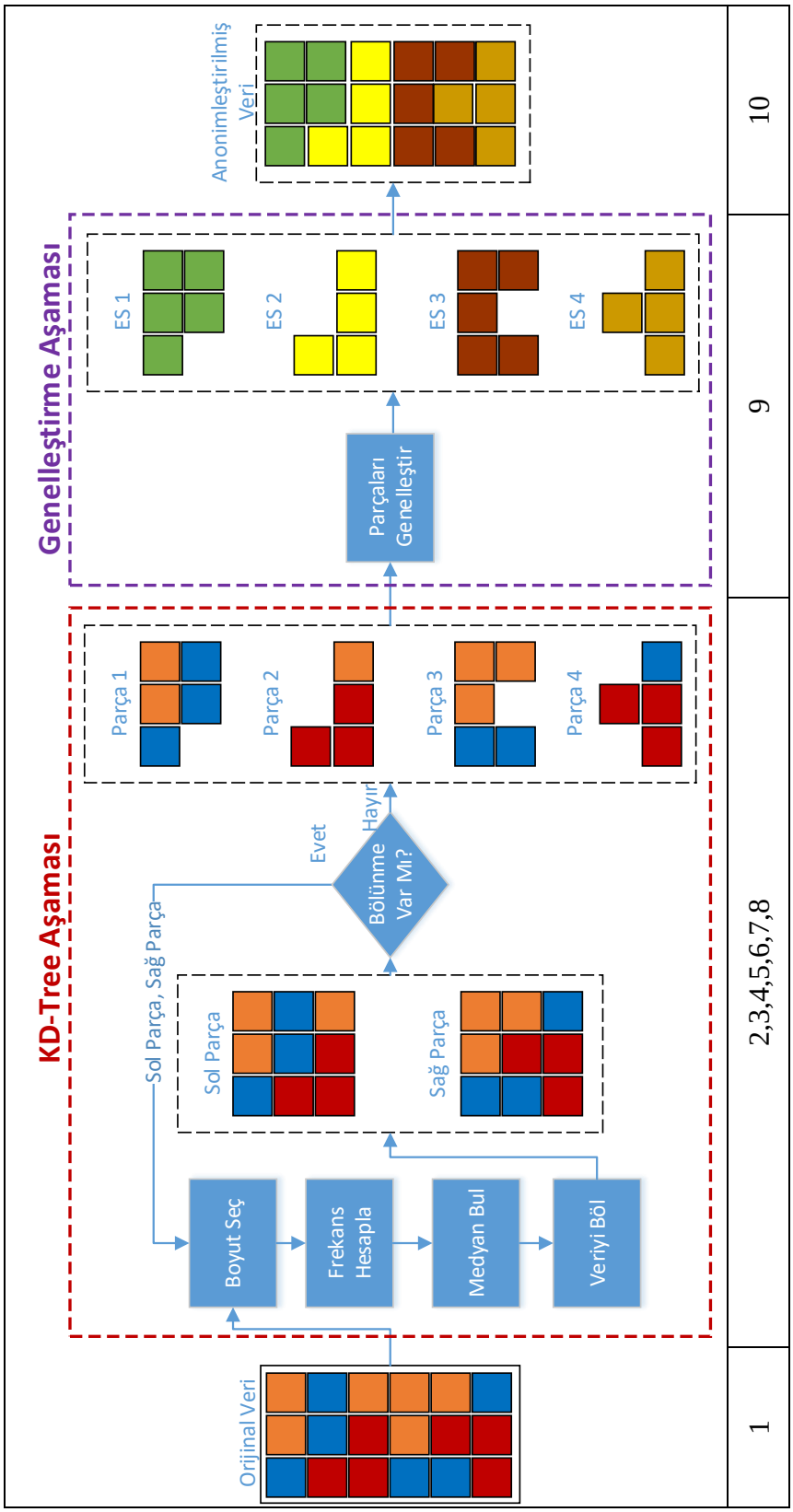
- *k*-Anonimlik için *k*=3 olarak belirlenmiştir,
- Her bir kutu bir veri noktasını temsil eder,
- Siyah hariç tüm renkler normal veri noktalarını temsil eder,
- Siyah renkli kutular aykırı veri noktalarını temsil eder,
- Her renk veri noktaları arasındaki yakınlığı temsil eder, yani aynı renklere sahip olan veriler birbirine diğer verilerden daha yakındır,
- Genelleştirme sonrası aynı renkli kutuları içeren her bir grup eşlenik sınıfları temsil eder,
- Bir renkten başka bir renge sert geçiş yüksek seviyede genelleştirmeyi gösterir (örneğin; Şekil 4.11'de Parça 1'deki mavi ve turuncu renkler yüksek genelleştirme sonrası sarıya dönüşmektedir),
- Bir renkten başka bir renge yumuşak bir geçiş ise düşük genelleştirmeyi gösterir (örneğin; Şekil 4.12'de normal veri içerisindeki ilk mavi parça ES1'de olduğu gibi açık maviye dönüşmektedir).

Şekil 4.11'de blok diyagramı sunulan Mondrian modelinin işlem adımları aşağıda sırasıyla verilmiştir;

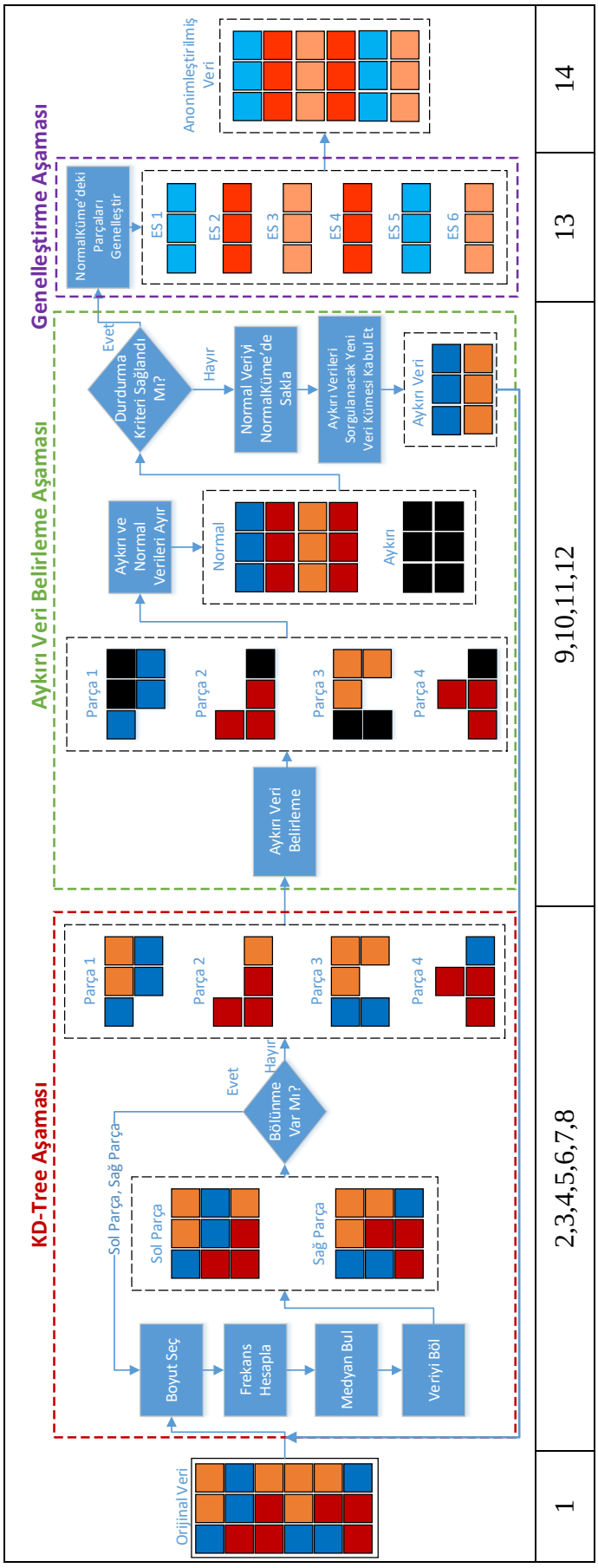
1. Anonimleştirilecek veri kümesini al,
2. Veri kümesi üzerinde bir boyut seç,
3. Seçilen boyut üzerindeki verilerin frekansını hesapla,
4. Bu frekansların medyanı bul,
5. Medyan değerini kullanarak veri uzayını Sol Parça ve Sağ Parça olmak üzere iki alt gruba ayır,
6. Bölünebilen veri uzayı kalıp kalmadığı kontrol et,
7. Eğer bölünebilen veri uzayı kalmış ise Adım 2'ye git,
8. Eğer kalmamış ise veri parçalarını elde et,
9. Her bir parçayı genelleştir ve eşlenik sınıflarını oluştur ve
10. Tüm eşlenik sınıfları birleştir ve anonim veri kümesini oluştur.

Şekil 4.12'de blok diyagramı gösterilen u-Mondrian modelinin çalışma prensibi ise aşağıda sırasıyla sunulmuştur;

1. Anonimleştirilecek veri kümesini al,
2. Veri kümesi üzerinde bir boyut seç,
3. Seçilen boyut üzerindeki verilerin frekansını hesapla,
4. Bu frekansların medyanı bul,
5. Medyan değerini kullanarak veri uzayını Sol Parça ve Sağ Parça olmak üzere iki alt gruba ayır,
6. Bölünebilen veri uzayı kalıp kalmadığı kontrol et,
7. Eğer bölünebilen veri uzayı kalmış ise Adım 2'ye git,
8. Eğer kalmamış ise veri parçalarını elde et,
9. Her bir veri parçasında aykırı verileri ve normal verileri belirle,
10. Veri kümesini, normal ve aykırı olarak iki alt kümeye ayır,
11. Durdurma kriterinin sağlanıp sağlanmadığını kontrol et,
12. Eğer sağlanmamışsa, normal veriyi normal kümesine ekle, aykırı verileri sorgulanacak yeni veri kümesi olarak kabul et ve Adım 1'e git,
13. Normal veri kümesindeki her bir veri parçasını genelleştir ve eşlenik sınıfları oluştur ve
14. Tüm eşlenik sınıfları birleştir ve anonim veri kümesini oluştur.

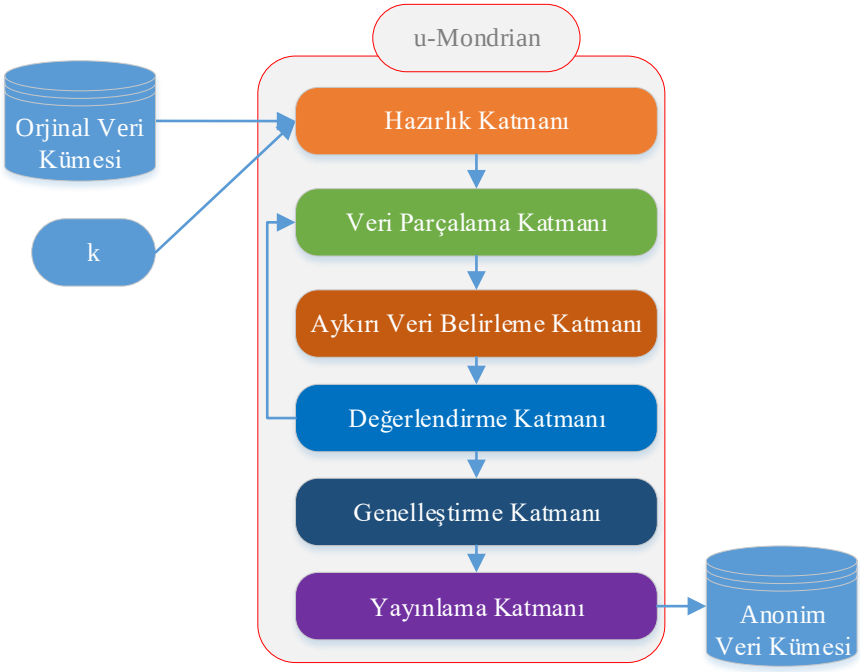


Şekil 4.11. Mondrian modelinin blok diyagramı



Şekil 4.12. u-Mondrian modelinin blok diyagramı

u-Mondrian modeli Şekil 4.13'de gösterildiği üzere, hazırlık, veri parçalama, aykırı veri belirleme, değerlendirme, genelleştirme ve yayınlama olmak üzere toplam 6 katmanlı bir yapıya sahiptir. Hazırlık katmanında anonimleştirilecek veri kümesi işlenmeye hazır hale getirilir. Veri parçalama katmanında veri kümesi KD-tree yapısı ile alt veri gruplarına parçalanır. Aykırı veri belirleme katmanında aykırı veriler ve normal veriler belirlenir. Değerlendirme katmanında durdurma kriterinin sağlanıp sağlanmadığı kontrol edilir. Genelleştirme katmanında, normal veri kümesi içerisindeki her alt grup kendi içerisinde genelleştirilerek anonim hale getirilir. Yayınlanma katmanında ise anonim veri grupları birleştirilerek yayınlanır. Bu katmanlar aşağıda detaylı olarak açıklanmıştır.



Şekil 4.13. u-Mondrian modelinin katmanlı yapısı

4.4.1. Hazırlık katmanı

Veri kümesi içerisindeki özniteliklerin sınıflandırılması, verinin normalize edilmesi ve durdurma kriterinin belirlenmesi bu katmanda gerçekleştirilir. Bu katmanına ait algoritma Şekil 4.14'de gösterilmekte olup, yapılan işlemler aşağıda açıklanmıştır;

- Özniteliklerin sınıflandırılması: veri içerisindeki özniteliklerin; tam-tanımlayıcı, yarı-tanımlayıcı, hassas ve hassas olmayan olmak üzere sınıflandırıldığı aşamadır.
- Veri normalizasyonu: her biri farklı değer aralıklarına sahip özniteliklerin normalize edildiği aşamadır.

- Durdurma kriterinin belirlenmesi: iterasyon sayısı, çeşitli metrik değerleri veya farklı parametreler durdurma kriteri olarak belirlenebilir.

```
1: function Hazirlik(Veri)
2:     T ← Oznitelik_Siniflandir(Veri)
3:     S ← Normalize_Et(T)
4:     Durdurma_Kriteri_Belirle()
5:     return S
6: end function
```

Şekil 4.14. Hazırlık katmanı algoritması

4.4.2. Veri parçalama katmanı

Bu katmanda, Mondrian modelinde kullanılan KD-tree yapısı ile veri kümesi her biri minimum *k* maksimum *p* adet veri içerecek şekilde alt gruplara parçalanır. Bu şekilde veri uzayı alt uzaylara bölünerek birbirine mümkün olabildiği kadar yakın elemanları içeren gruplar oluşturulur. Şekil 4.15'de veri parçalama katmanı algoritması gösterilmektedir.

```
1: function Veriyi_parcala(S, k)
2:     D ← KD_Tree(S, k)
3:     return (D)
4: end function
```

Şekil 4.15. Veri parçalama katmanı algoritması

4.4.3. Aykırı veri belirleme katmanı

Aykırı verilerin ve normal verilerin belirlendiği katmandır. Şekil 4.16'da algoritması sunulan bu katman için, veri parçalama katmanı sonrası elde edilen her bir alt grupta sırasıyla şu işlemler yapılır;

1. alt gruptaki verilere LOF algoritması uygulanarak LOF skorları elde edilir,
2. minimum LOF skoruna sahip veri, referans noktası olarak kabul edilir,
3. referans noktasına en yakın *k-1* adet veri tespit edilir,
4. referans noktası da dahil olmak üzere birbirine en yakın *k* adet veri alınarak NDS normal veri kümesine, arta kalan veriler ise ODS aykırı veri kümesine eklenir ve
5. tüm alt gruplar için bu işlemler yapıldıktan sonra NDS ve ODS geri gönderilir.

```
1: function Aykiri_Verileri_Bul(D_i, k)
2:     for all grup ∈ D_i do
3:         t ← LOF(grup)
4:         referans ← grup(min(t))
5:         N ← kNN_Bul(referans, grup)
6:         NDS_i ← NDS_i ∪ U
7:         ODS_i ← grup − U
8:     end for
9:     return (NDS, ODS)
10: end function
```

Şekil 4.16. Aykırı veri belirleme katmanı algoritması

4.4.4. Değerlendirme katmanı

Bir durdurma kriterine bağlı olarak veri parçalama, aykırı veri belirleme, genelleştirme ve yayınlama katmanlarının çalışmasını kontrol eden katmandır. Belirlenen durdurma kriteri sağlanmazsa veri parçalama ve aykırı veri belirleme katmanları çalıştırılırken, bu kriter sağlanırsa genelleştirme ve yayınlama katmanına geçilir. Şekil 4.17'de bu katmana ait algoritma gösterilmektedir.

```
1: while durdurma_kriteri == false do
2:     devam
3:     ....
4: end while
```

Şekil 4.17. Değerlendirme katmanı algoritması

4.4.5. Genelleştirme katmanı

Veri kümesinin anonimleştirildiği katmandır. Aykırı veri belirleme katmanı sonrası oluşturulan NDS kümesi içerisindeki tüm alt veri gruplarının anonimleştirilmesi bu aşamada gerçekleşir. Anonimleştirme için genelleştirme operatörünü kullanır. Bu katmana ait algoritma Şekil 4.18'de gösterilmektedir.

```
1: function Genellestir(grup)
2:     grup^A ← Genellestir(grup)
3:     return grup^A
4: end function
```

Şekil 4.18. Genelleştirme katmanı algoritması

4.4.6. Yayınlama katmanı

Faydası arttırılmış olan anonim verinin yayınlandığı son katmandır. En son aşamada elde edilen T^A anonim veri kümesi bu katmanda yayınlanır.

4.5. Deneysel Çalışmalar

Bu bölümde, önerilen u-Mondrian modelini test etmek ve doğrulamak için gerçekleştirilen deneysel çalışmalar açıklanmış ve elde edilen sonuçlar sunulmuştur.

Deneysel çalışmalar, Intel Core i3 2.3 GHz işlemci ve 4 GB ram özelliğine sahip bilgisayar üzerinde, aşağıda belirtilen durumlar çerçevesinde gerçekleştirilmiştir;

- Strict ve Relaxed bölme stratejileri uygulanmıştır,
- Adult ve Diyabet veri kümeleri kullanılmıştır,
- *k* değerleri literatürde sıklıkla tercih edilen değerlerden seçilmiştir,
- DM, GCP ve AECS metrikleri ve önceki bölümde verilen karar kuralları kullanılmıştır,
- İterasyon sayısı durdurma kriteri olarak belirlenmiş ve modelleri eşit bir şekilde değerlendirebilmek amacıyla 5'e sabitlenmiştir.

4.5.1. Deney 1: u-Mondrian ve Mondrian modellerinin adult veri kümesi ile test edilmesi

Bu deneyde, u-Mondrian ve Mondrian modelleri Adult veri kümesi ile test edilmiş ve elde edilen sonuçlar karşılaştırılmıştır. Her *k* değeri ve bu değerlerdeki her iterasyon için elde edilen metrik sonuçları tablolar halinde sunulmuş ve karşılaştırmaları grafiksel olarak gösterilmiştir.

Deneylerde DM, GCP ve AECS metriklerine göre elde edilen sonuçlar gösterilmiş ve bu sonuçlardaki değişimler için karar kurallarının çıktıları yorumlanmıştır. Bu metriklere ek olarak, eşlenik sınıf sayılarını değerlendirmek için bu sınıfların sayılarını toplayan bir ES fonksiyonu da kullanılmıştır. DM, GCP ve AECS metrikleri veri faydası ile ters orantılı iken ES veri faydası ile doğru orantılıdır. Bu deney kapsamında yoğunluk ve yakınlık tabanlı hibrit bir aykırı veri belirleme yaklaşımı kullanıldığı için GCP metriği ile ölçülen değerlerin

diğer metriklerle ölçülen değerlere göre daha kıymetli olduğu değerlendirilmektedir. Çünkü GCP metriği eşlenik sınıflar içerisindeki elemanların birbirine yakınlığını ölçerken, diğer metrikler eşlenik sınıfların boyutu ve ortalama eleman sayıları gibi özelliklerini ölçmektedir. Bu kapsamda eşlenik sınıflar içerisindeki verilerin birbirine olan yakınlıklarının değerlendirilmesi veri faydası açısından daha anlamlı olacaktır.

Yapılan bu deneyde elde edilen sonuçlar, önerilen modelin Mondrian modeline göre yüksek veri faydası sunduğunu açıkça göstermektedir.

Strict Bölme Stratejisine Göre u-Mondrian ve Mondrian Modelleri için Elde Edilen Sonuçlar

Bu deneyde, Strict bölme stratejisine göre u-Mondrian ve Mondrian modelleri test edilmiş ve sonuçları sunulmuştur. Tüm çizelgelerde, u-Mondrian modeli u-M ile temsil edilirken Mondrian modeli ise M ile gösterilmiştir.

Çizelge 4.1'de gösterilen sonuçlar incelendiğinde; *k*=5 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 207996, 3952,62 ve 6,62 değerlerini sunarken, u-Mondrian sırasıyla 150725, 2636,09 ve 5,02 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.1. *k*=5 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	4553	4553	207996	207996	3952,61	3952,61	6,62	M
1	22765	7397	4553	4553	113825	113825	1522,65	1522,65	5,00	u-M
2	5625	1772	1125	5678	28125	141950	642,64	2165,29	5,00	
3	1390	382	278	5956	6950	148900	303,44	2468,73	5,00	
4	295	87	59	6015	1475	150375	116,64	2585,37	5,00	
5	87	0	17	6032	350	150725	50,72	2636,09	5,11	

Çizelge 4.2'de gösterilen sonuçlar incelendiğinde; *k*=10 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 425890, 6141,80 ve 13,57 değerlerini sunarken, u-Mondrian sırasıyla 301200, 4173,54 ve 10,02 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.2. *k*=10 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	2222	2222	425890	425890	6141,80	6141,80	13,57	M
1	22220	7942	2222	2222	222200	222200	2275,42	2275,42	10,00	u-M
2	5840	2102	584	2806	58400	280600	1078,84	3354,26	10,00	
3	1530	572	153	2959	15300	295900	506,97	3861,23	10,00	
4	430	142	43	3002	4300	300200	214,26	4075,49	10,00	
5	142	0	14	3016	1000	301200	98,05	4173,54	10,14	

Çizelge 4.3'de gösterilen sonuçlar incelendiğinde; *k*=20 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 856562, 8582,57 ve 27,29 değerlerini sunarken, u-Mondrian sırasıyla 602800, 6096,82 ve 20,08 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.3. *k*=20 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	1105	1105	856562	856562	8582,57	8582,57	27,29	M
1	22100	8062	1105	1105	442000	442000	3189,90	3189,90	20,00	u-M
2	5960	2102	298	1403	119200	561200	1573,70	4763,60	20,00	
3	1480	622	74	1477	29600	590800	709,29	5472,89	20,00	
4	520	102	26	1503	10400	601200	457,62	5930,51	20,00	
5	102	0	5	1508	1600	602800	166,31	6096,82	20,40	

Çizelge 4.4'de gösterilen sonuçlar incelendiğinde; *k*=30 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 1267990, 10367,27 ve 40,26 değerlerini sunarken, u-Mondrian sırasıyla 904500, 7732,21 ve 32,40 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.4. *k*=30 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	749	749	1267990	1267990	10367,27	10367,27	40,26	M
1	22470	7692	749	749	674100	674100	3854,61	3854,61	30,00	u-M
2	5730	1962	191	940	171900	846000	2079,47	5934,08	30,00	
3	1440	522	48	988	43200	889200	960,31	6894,39	30,00	
4	480	42	16	1004	14400	903600	720,97	7615,36	30,00	
5	42	0	1	1005	900	904500	116,85	7732,21	42,00	

Çizelge 4.5'de gösterilen sonuçlar incelendiğinde; *k*=40 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 1725228, 11586,60 ve 54,84 değerlerini sunarken, u-Mondrian sırasıyla 1206400, 8885,98 ve 40,01 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.5. *k*=40 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	550	550	1725228	1725228	11586,60	11586,60	54,84	M
1	22000	8162	550	550	880000	880000	4479,64	4479,64	40,00	u-M
2	5600	2562	140	690	224000	1104000	1943,77	6423,41	40,00	
3	1760	802	44	734	70400	1174400	1122,82	7546,23	40,00	
4	640	162	16	750	25600	1200000	733,84	8280,07	40,00	
5	162	0	4	754	6400	1206400	604,91	8884,98	40,5	

Çizelge 4.6'da gösterilen sonuçlar incelendiğinde; *k*=50 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 2153108, 12664,65 ve 68,86 değerlerini sunarken, u-Mondrian sırasıyla 1505000, 9247,80 ve 50,80 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.6. *k*=50 için u-Mondrian ve Mondrian modellerinin test sonuçları

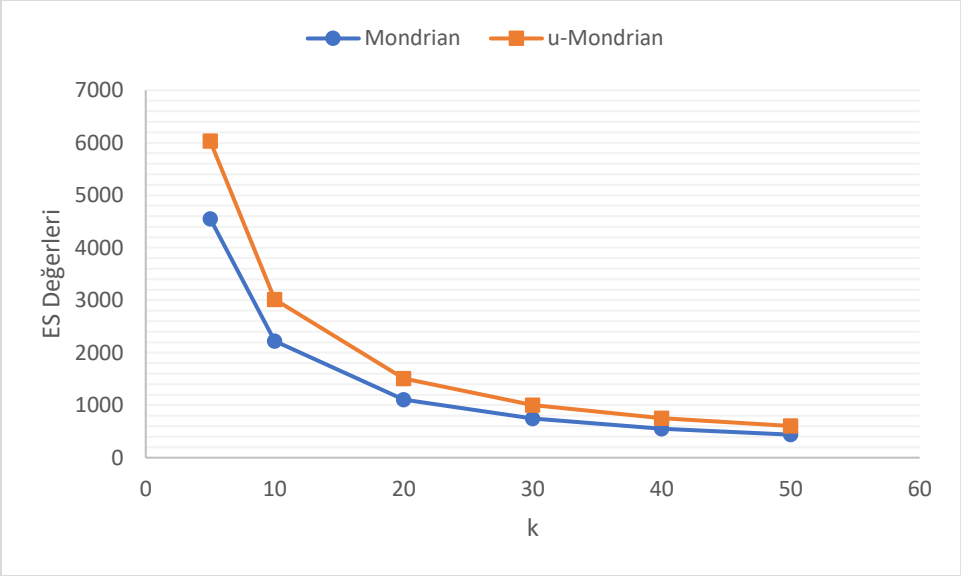
i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	438	438	2153108	2153108	12664,65	12664,65	68,86	M
1	21900	8262	438	438	1095000	1095000	4495,13	4495,13	50,00	u-M
2	5800	2462	116	554	290000	1385000	2427,90	6923,03	50,00	
3	1750	712	35	589	87500	1472500	1323,75	8246,78	50,00	
4	550	162	11	600	27500	1500000	672,09	8918,87	50,00	
5	162	0	3	603	5000	1505000	328,93	9247,80	54,00	

Çizelge 4.7'de u-Mondrian ile Mondrian modellerinin ES, DM, GCP ve AECS değerleri genel olarak gösterilmektedir. Elde edilen bu sonuçlara ve karar kurallarına göre, önerilen modelin Mondrian modeline göre tüm *k* değerleri için veri faydasını büyük oranda iyileştirdiği görülmektedir.

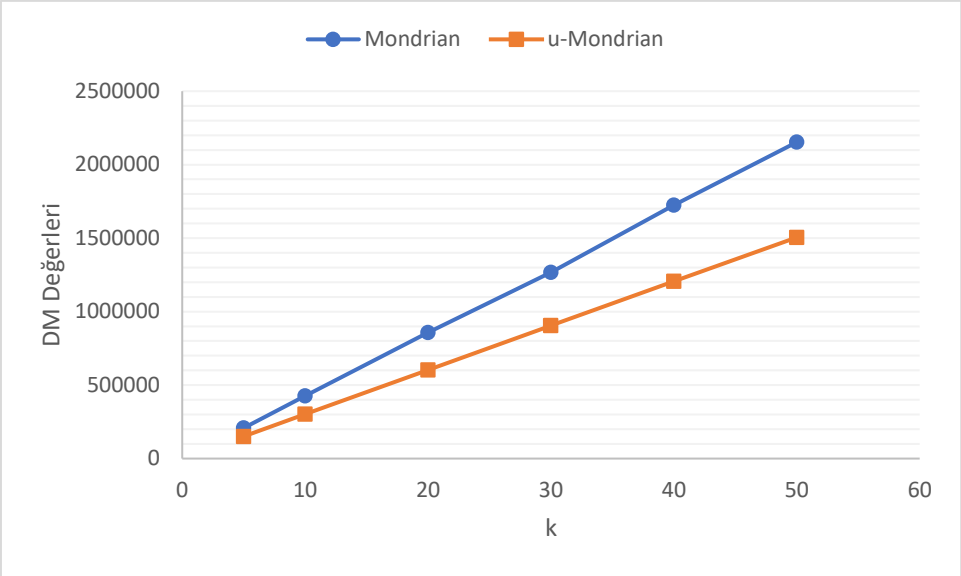
Şekil 4.19, Şekil 4.20, Şekil 4.21 ve Şekil 4.22'de her iki model için farklı *k* değerlerine göre ölçülen ES, DM, GCP ve AECS değerleri sırasıyla gösterilmiştir. Sunulan tüm bu şekillerde, u-Mondrian modelinin Mondrian modeline göre daha yüksek veri faydası sunduğu açıkça görülmektedir.

Çizelge 4.7. u-Mondrian ve Mondrian modellerinin genel test sonuçları

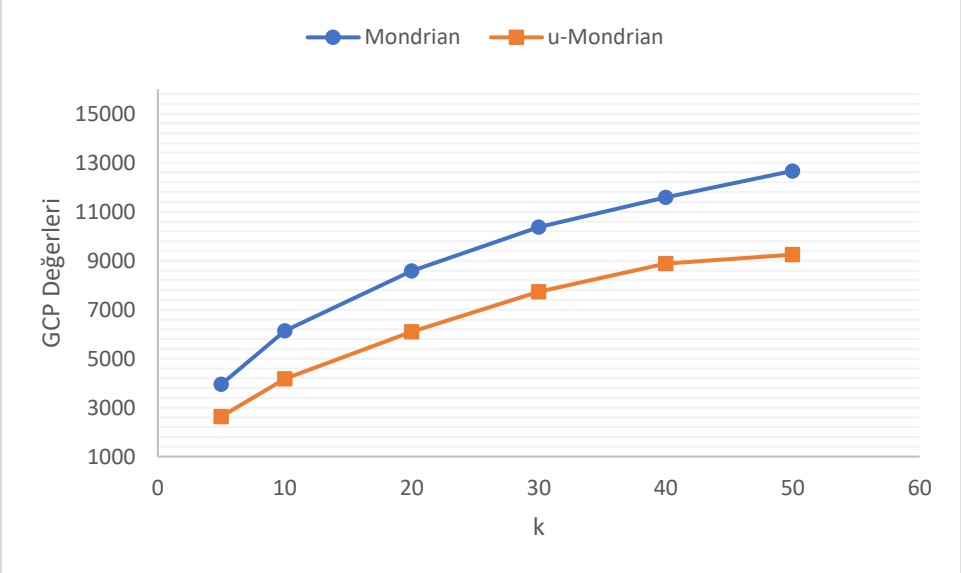
k	ES	DM	GCP	AECS	Model
5	4553	207996	3952,61	6,62	Mondrian
	6032	150725	2636,09	5,02	u-Mondrian
10	2222	425890	6141,80	13,57	Mondrian
	3016	301200	4173,54	10,02	u-Mondrian
20	1105	856562	8582,57	27,29	Mondrian
	1508	602800	6096,82	20,08	u-Mondrian
30	749	1267990	10367,27	40,26	Mondrian
	1005	904500	7732,21	32,4	u-Mondrian
40	550	1725228	11586,60	54,84	Mondrian
	754	1206400	8884,98	40,10	u-Mondrian
50	438	2153108	12664,65	68,86	Mondrian
	603	1505000	9247,80	50,80	u-Mondrian



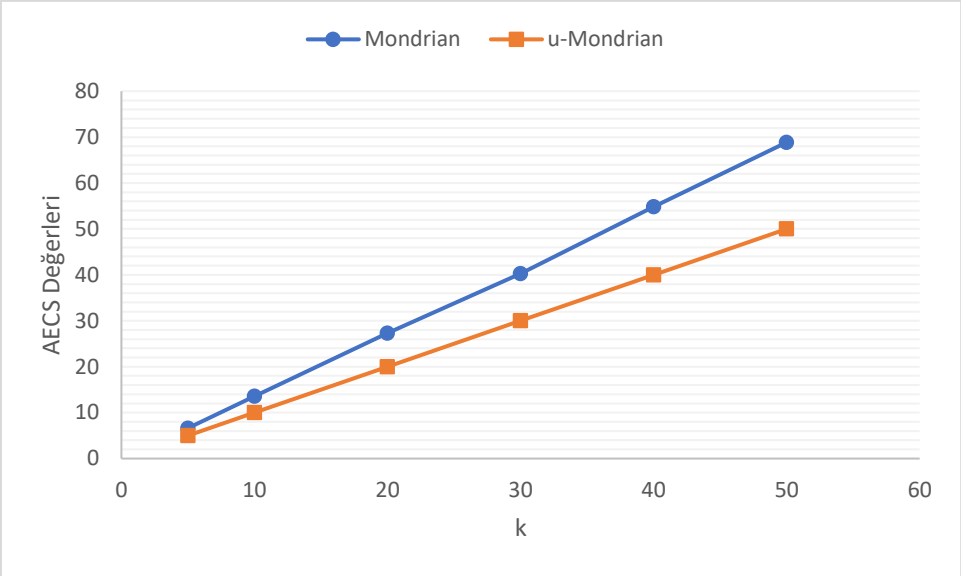
Şekil 4.19. Farklı *k* değerleri için elde edilen ES değerleri



Şekil 4.20. Farklı *k* değerleri için elde edilen DM değerleri



Şekil 4.21. Farklı *k* değerleri için elde edilen GCP değerleri



Şekil 4.22. Farklı *k* değerleri için elde edilen AECS değerleri

Çizelge 4.7'de sunulan sonuçlara göre u-Mondrian modelinin Mondrian modeline göre veri faydasında yaptığı iyileştirmeler oransal olarak Çizelge 4.8'de gösterilmiştir. Bu sonuçlara göre, u-Mondrian modeli Mondrian modeline göre DM, GCP ve AECS değerlerinde sırasıyla %27,53-%30,10, %23,31-%33,30 ve %19,54-%26,87 aralıklarında daha yüksek veri faydası sunduğu görülmektedir.

Çizelge 4.8. u-Mondrian modelinin Mondrian'a göre veri faydası iyileştirme oranları

k	DM (%)	GCP (%)	AECS (%)
5	27,53	33,30	24,16
10	29,27	32,04	26,12
20	29,62	28,96	26,43
30	28,66	25,41	19,54
40	30,07	23,31	26,87
50	30,10	26,97	26,23

Relaxed Bölme Stratejisine Göre u-Mondrian ve Mondrian Modelleri için Elde Edilen Sonuçlar

Bu deneyde, Relaxed bölme stratejisine göre u-Mondrian ve Mondrian modelleri test edilmiş ve sonuçları sunulmuştur. Tüm çizelgelerde, u-Mondrian modeli u-M ile Mondrian modeli ise M ile kısaltılarak temsil edilmiştir.

Çizelge 4.9'da gösterilen sonuçlar incelendiğinde; *k*=5 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 223054, 6266,90 ve 7,36 değerlerini sunarken, u-Mondrian sırasıyla 150650, 4575,40 ve 5,00 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.9. *k*=5 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	4096	4096	223054	223054	6266,90	6266,90	7,36	M
1	20480	9682	4096	4096	102400	102400	2396,58	2396,58	5,00	u-M
2	7450	2232	1490	5586	37250	139650	1491,73	3888,31	5,00	
3	1280	952	256	5842	6400	146050	255,81	4144,12	5,00	
4	640	312	128	5970	3200	149250	213,22	4357,34	5,00	
5	312	0	62	6032	1400	150650	218,06	4575,40	5,03	

Çizelge 4.10'da gösterilen sonuçlar incelendiğinde; *k*=10 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 444618, 9561,73 ve 14,72 değerlerini sunarken, u-Mondrian sırasıyla 300800, 7276,44 ve 10,00 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.10. *k*=10 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	2048	2048	444618	444618	9561,73	9561,73	14,72	M
1	20480	9682	2048	2048	204800	204800	3855,32	3855,32	10,00	u-M
2	5120	4562	512	2560	51200	256000	1243,87	5099,19	10,00	
3	2560	2002	256	2816	25600	281600	831,28	5930,47	10,00	
4	1280	722	128	2944	12800	294400	690,87	6621,34	10,00	
5	722	0	72	3016	6400	300800	655,10	7276,44	10,02	

Çizelge 4.11'de gösterilen sonuçlar incelendiğinde; *k*=20 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 888678, 13938,85 ve 29,45 değerlerini sunarken, u-Mondrian sırasıyla 601600, 10413,81 ve 20,00 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.11. *k*=20 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	1024	1024	888678	888678	13938,85	13938,85	29,45	M
1	20480	9682	1024	1024	409600	409600	5366,97	5366,97	20,00	u-M
2	5120	4562	256	1280	102400	512000	1793,01	7159,98	20,00	
3	2560	2002	128	1408	51200	563200	1360,24	8520,22	20,00	
4	1280	722	64	1472	25600	588800	1035,55	9555,77	20,00	
5	722	0	36	1508	12800	601600	858,04	10413,81	20,05	

Çizelge 4.12'de gösterilen sonuçlar incelendiğinde; *k*=30 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 1776890, 19792,51 ve 58,91 değerlerini sunarken, u-Mondrian sırasıyla 892800, 10984,54 ve 30,05 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.12. *k*=30 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	512	512	1776890	1776890	19792,51	19792,51	58,91	M
1	15360	14802	512	512	460800	460800	4032,28	4032,28	30,00	u-M
2	7680	7122	256	768	230400	691200	2617,72	6650,00	30,00	
3	3840	3282	128	896	115200	806400	1846,89	8496,89	30,00	
4	1920	1362	64	960	57600	864000	1369,63	9866,52	30,00	
5	1362	0	45	1005	28800	892800	1118,02	10984,54	30,26	

Çizelge 4.13'de gösterilen sonuçlar incelendiğinde; *k*=40 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 1776890, 19792,52 ve 58,91 değerlerini sunarken, u-Mondrian sırasıyla 1203200, 14076,81 ve 40,02 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.13. *k*=40 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	512	512	1776890	1776890	19792,52	19792,52	58,91	M
1	20480	9682	512	512	819200	819200	6974,07	6974,07	40,00	u-M
2	5120	4562	128	640	204800	1024000	2475,28	9449,35	40,00	
3	2560	2002	64	704	102400	1126400	1811,61	11260,96	40,00	
4	1280	722	32	736	51200	1177600	1425,78	12686,74	40,00	
5	722	0	18	754	25600	1203200	1390,07	14076,81	40,11	

Çizelge 4.14'de gösterilen sonuçlar incelendiğinde; *k*=50 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 1776890, 19792,52 ve 58,91 değerlerini sunarken, u-Mondrian sırasıyla 1505000, 16620,55 ve 50,08 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.14. *k*=50 için u-Mondrian ve Mondrian modellerinin test sonuçları

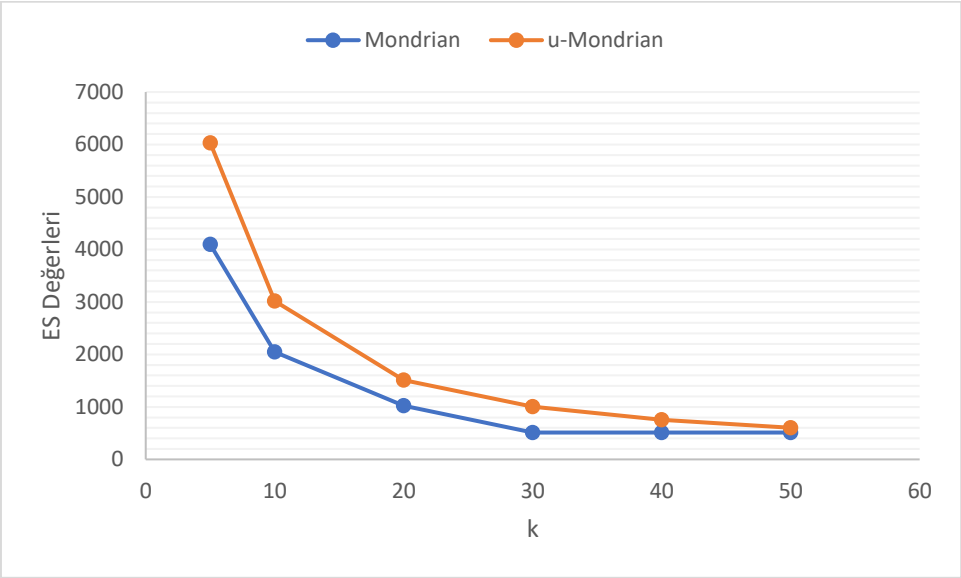
i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	512	512	1776890	1776890	19792,52	19792,52	58,91	M
1	25600	4562	512	512	1280000	1280000	11345,24	11345,24	50,00	u-M
2	3200	1362	64	576	160000	1440000	2874,47	14219,71	50,00	
3	800	562	16	592	40000	1480000	1107.73	15327,44	50,00	
4	400	162	8	600	20000	1500000	976,20	16303,64	50,00	
5	162	0	3	603	5000	1505000	316,91	16620,55	54,00	

Çizelge 4.15'de, u-Mondrian ile Mondrian modellerinin ES, DM, GCP ve AECS değerleri gösterilmektedir. Elde edilen bu sonuçlara ve karar kurallarına göre, önerilen modelin Mondrian modeline göre tüm *k* değerleri için veri faydasını büyük oranda iyileştirdiği görülmektedir.

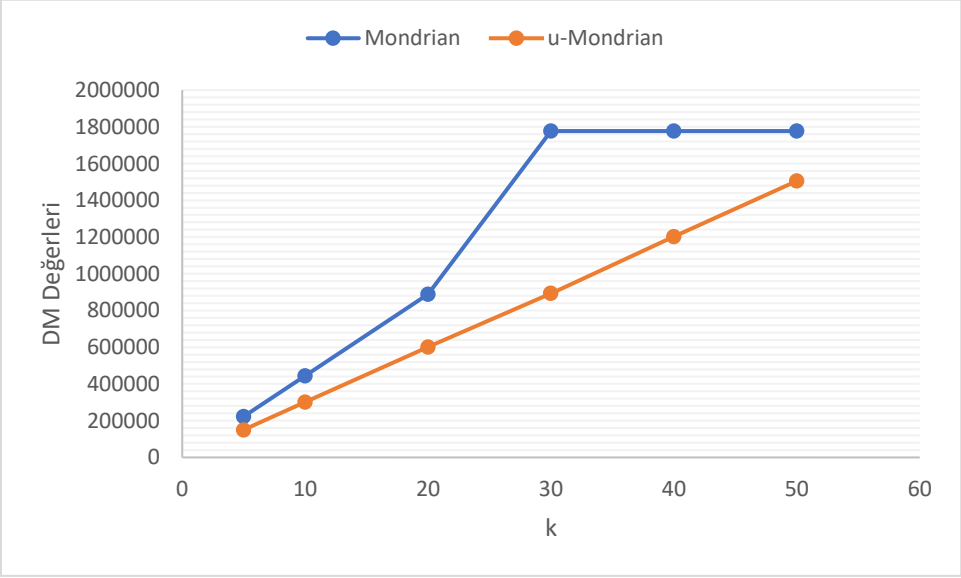
Çizelge 4.15. u-Mondrian ve Mondrian modellerinin genel sonuçları

k	ES	DM	GCP	AECS	Model
5	4096	223054	6266,90	7,36	Mondrian
	6032	150650	4575,40	5,00	u-Mondrian
10	2048	444618	9561,73	14,72	Mondrian
	3016	300800	7276,44	10,00	u-Mondrian
20	1024	888678	13938,85	29,45	Mondrian
	1508	601600	10413,81	20,00	u-Mondrian
30	512	1776890	19792,51	58,91	Mondrian
	1005	892800	10984,54	30,05	u-Mondrian
40	512	1776890	19792,52	58,91	Mondrian
	754	1203200	14076,81	40,02	u-Mondrian
50	512	1776890	19792,52	58,91	Mondrian
	603	1505000	16620,55	50,08	u-Mondrian

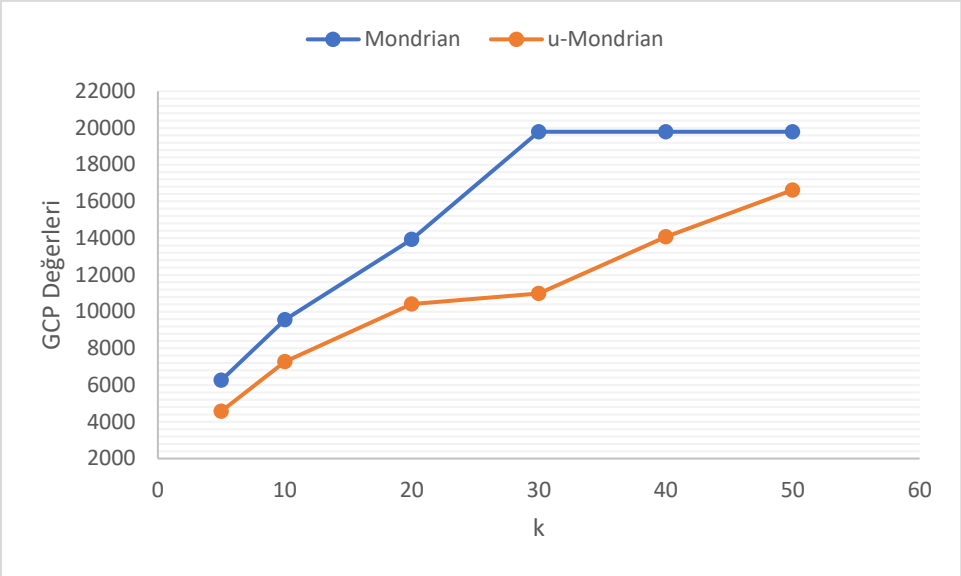
Şekil 4.23, Şekil 4.24, Şekil 4.25 ve Şekil 4.26'da iki model için farklı *k* değerlerine göre ölçülen ES, DM, GCP ve AECS değerleri gösterilmiştir. u-Mondrian modelinin Mondrian modeline göre daha yüksek veri faydası sunduğu açıkça görülmektedir.



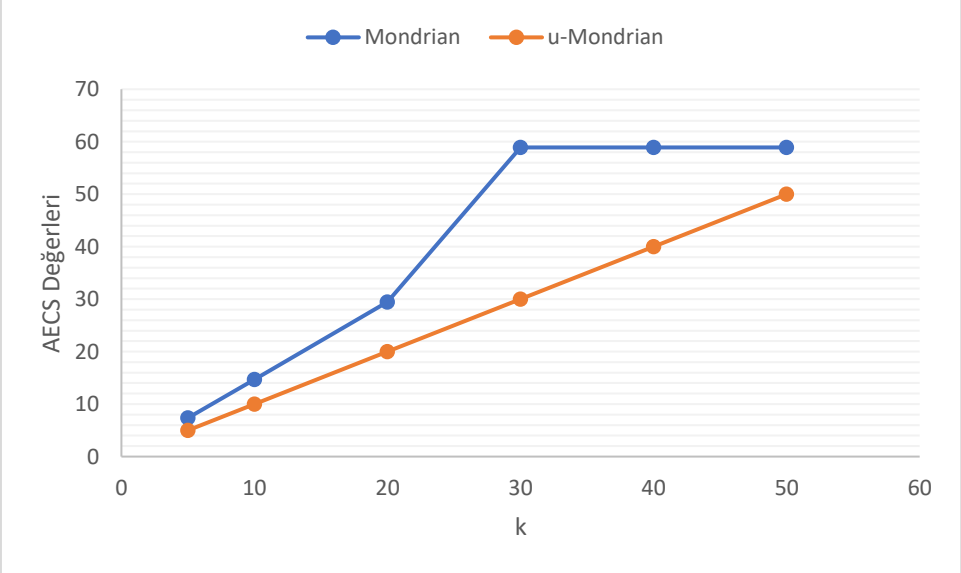
Şekil 4.23. Farklı *k* değerleri için elde edilen ES değerleri



Şekil 4.24. Farklı *k* değerleri için elde edilen DM değerleri



Şekil 4.25. Farklı *k* değerleri için elde edilen GCP değerleri



Şekil 4.26. Farklı *k* değerleri için elde edilen AECS değerleri

Çizelge 4.15'de sunulan sonuçlara göre u-Mondrian modelinin Mondrian modeline kıyasla veri faydasında yaptığı iyileştirmeler oransal olarak Çizelge 4.16'da gösterilmiştir. Bu sonuçlara göre, u-Mondrian modeli Mondrian modeline göre DM, GCP ve AECS değerlerinde sırasıyla %15,30-%49,75, %16,02-%44,50 ve %13,76-%48,98 aralıklarında daha yüksek veri faydası sunduğu görülmektedir.

Çizelge 4.16. u-Mondrian modelinin Mondrian'a göre veri faydası iyileştirme oranları

k	DM (%)	GCP (%)	AECS (%)
5	32,46	26,99	32,01
10	32,34	23,90	32,06
20	32,30	25,28	32,06
30	49,75	44,50	48,98
40	32,28	28,87	32,06
50	15,30	16,02	13,76

4.5.2. Deney 2: u-Mondrian ve Mondrian modellerinin diyabet veri kümesi ile test edilmesi

Bu deneyde ise, u-Mondrian ve Mondrian modelleri Diyabet veri kümesi ile test edilmiş ve elde edilen sonuçlar karşılaştırılmıştır. Her *k* değeri ve bu değerlerdeki her iterasyon için elde edilen metrik sonuçları tablolar halinde sunulmuş ve karşılaştırmaları grafiksel olarak gösterilmiştir.

Deneylerde DM, GCP ve AECS metriklerine göre elde edilen sonuçlar gösterilmiş, bu sonuçlardaki değişimler için karar kurallarının çıktıları yorumlanmış ve ES fonksiyonu ile de eşlenik sınıf sayıları değerlendirilmiştir

Yapılan bu deneyde elde edilen sonuçlar, önerilen modelin Mondrian modeline göre yüksek veri faydası sunduğunu açıkça göstermektedir.

Strict Bölme Stratejisine Göre u-Mondrian ve Mondrian Modelleri için Elde Edilen Sonuçlar

Bu deneyde, Strict bölme stratejisine göre u-Mondrian ve Mondrian modelleri test edilmiş ve sonuçları sunulmuştur. Tüm çizelgelerde, u-Mondrian modeli u-M ile temsil edilirken Mondrian modeli ise M ile gösterilmiştir.

Çizelge 4.17'de gösterilen sonuçlar incelendiğinde; *k*=5 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 1068636, 48485,08 ve 7,95 değerlerini sunarken, u-Mondrian sırasıyla 502625, 29901,50 ve 5,00 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.17. *k*=5 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	101766	0	12791	12791	1068636	1068636	48485,08	48485,08	7,95	M
1	63955	37811	12791	12791	319775	319775	15108,64	15108,64	5,00	u-M
2	23760	14051	4752	17543	118800	438575	8319,69	23428,33	5,00	
3	8525	5526	1705	19248	42625	481200	3818,39	27246,72	5,00	
4	3075	2451	615	19863	15375	496575	1826,28	29073,00	5,00	
5	2451	0	490	20353	6050	502625	828,50	29901,50	5,00	

Çizelge 4.18'de gösterilen sonuçlar incelendiğinde; *k*=10 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 1752930, 64090,00 ve 15,22 değerlerini sunarken, u-Mondrian sırasıyla 1015500, 44138,28 ve 10,00 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.18. *k*=10 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	101766	0	6683	6683	1752930	1752930	64090,00	64090,00	15,22	M
1	66830	34936	6683	6683	668300	668300	23438,55	23438,55	10,00	u-M
2	24680	10256	2468	9151	246800	915100	12274,51	35713,06	10,00	
3	7340	2916	734	9885	73400	988500	5236,97	40950,03	10,00	
4	2130	786	213	10098	21300	1009800	2344,11	43294,14	10,00	
5	786	0	78	10176	5700	1015500	844,14	44138,28	10,07	

Çizelge 4.19'da gösterilen sonuçlar incelendiğinde; *k*=20 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 3254622, 83088,26 ve 29,63 değerlerini sunarken, u-Mondrian sırasıyla 2030400, 59553,39 ve 20,02 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.19. *k*=20 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	101766	0	3434	3434	3254622	3254622	83088,26	83088,26	29,63	M
1	68680	33086	3434	3434	1373600	1373600	32972,18	32972,18	20,00	u-M
2	23280	9806	1164	4598	465600	1839200	15494,78	48466,96	20,00	
3	6860	2946	343	4941	137200	1976400	6750,56	55217,52	20,00	
4	2020	926	101	5042	40400	2016800	2844,09	58061,62	20,00	
5	926	0	46	5088	13600	2030400	1491,76	59553,39	20,13	

Çizelge 4.20'de gösterilen sonuçlar incelendiğinde; *k*=30 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 4778192, 95088,58 ve 43,97 değerlerini sunarken, u-Mondrian sırasıyla 3046500, 69425,36 ve 30,05 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.20. *k*=30 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	101766	0	2314	2314	4778192	4778192	95088,58	95088,58	43,97	M
1	69420	32346	2314	2314	2082600	2082600	38677,27	38677,28	30,00	u-M
2	23160	9186	772	3086	694800	2777400	18222,38	56899,66	30,00	
3	6690	2496	223	3309	200700	2978100	7968,63	64868,30	30,00	
4	1860	636	62	3371	55800	3033900	3428,60	68296,90	30,00	
5	636	0	21	3392	12600	3046500	1128,46	69425,36	30,28	

Çizelge 4.21'de gösterilen sonuçlar incelendiğinde; *k*=40 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 6393690, 104847,97 ve 58,99 değerlerini sunarken, u-Mondrian sırasıyla 4067200, 76710,43 ve 40,07 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.21. *k*=40 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	101766	0	1725	1725	6393690	6393690	104847,97	104847,97	58,99	M
1	69000	32766	1725	1725	2760000	2760000	41968,22	41968,22	40,00	u-M
2	23280	9486	582	2307	931200	3691200	19710,62	61678,84	40,00	
3	6960	2526	174	2481	278400	3969600	8985,26	70664,10	40,00	
4	1880	646	47	2528	75200	4044800	4072,89	74736,99	40,00	
5	646	0	16	2544	22400	4067200	1973,44	76710,43	40,37	

Çizelge 4.22'de gösterilen sonuçlar incelendiğinde; *k*=50 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 8037076, 113145,87 ve 74,33 değerlerini sunarken, u-Mondrian sırasıyla 5082500, 83005,60 ve 50,26 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.22. *k*=50 için u-Mondrian ve Mondrian modellerinin test sonuçları

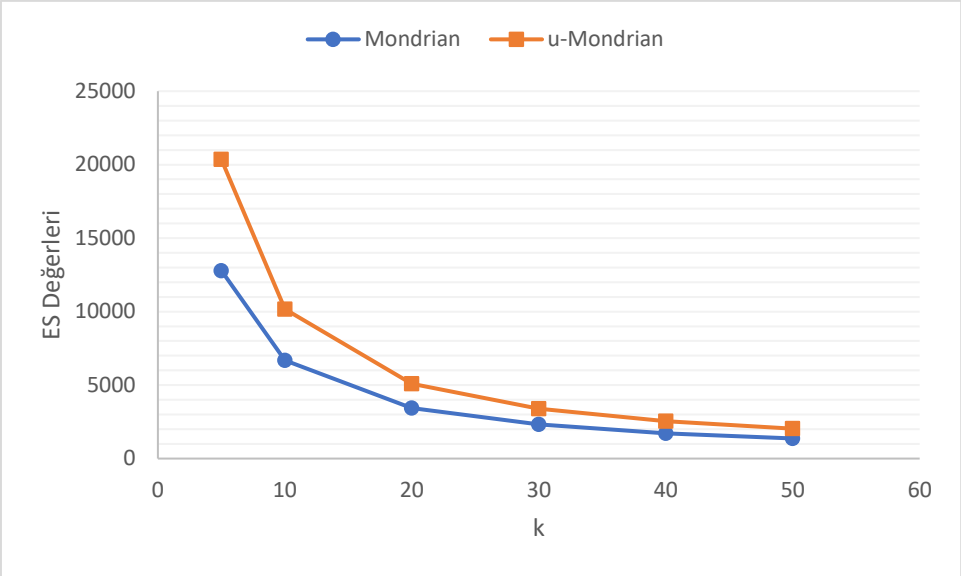
i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	101766	0	1369	1369	8037076	8037076	113145,88	113145,88	74,33	KM
1	68450	33316	1369	1369	3422500	3422500	44790,48	44790,49	50,00	u-M
2	23750	9566	475	1844	1187500	4610000	21947,84	66738,33	50,00	
3	7200	2366	144	1988	360000	4970000	10313,93	77052,27	50,00	
4	1750	616	35	2023	87500	5057500	4224,95	81277,23	50,00	
5	616	0	12	2035	25000	5082500	1728,37	83005,60	51,33	

Çizelge 4.23'de u-Mondrian ile Mondrian modellerinin ES, DM, GCP ve AECS sonuçları gösterilmektedir. Bu sonuçlara göre, önerilen modelin Mondrian'a göre tüm *k* değerleri için veri faydasını büyük oranda iyileştirdiği görülmektedir.

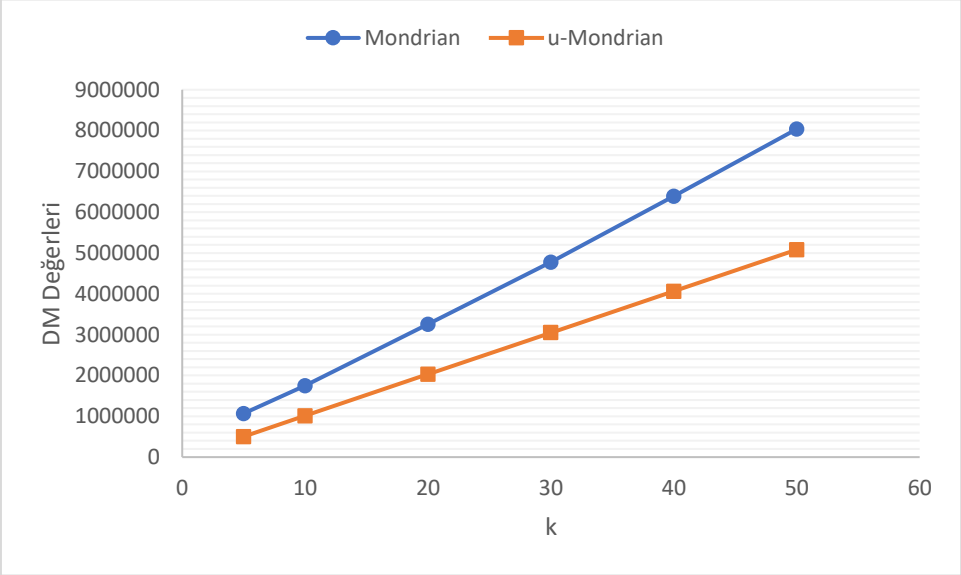
Çizelge 4.23. u-Mondrian ile Mondrian modellerinin genel test sonuçları

k	ES	DM	GCP	AECS	Model
5	12791	1068636	48485,08	7,95	Mondrian
	20353	502625	29901,50	5,00	u-Mondrian
10	6683	1752930	64090,00	15,22	Mondrian
	10176	1015500	44138,28	10,00	u-Mondrian
20	3434	3254622	83088,26	29,63	Mondrian
	5088	2030400	59553,39	20,02	u-Mondrian
30	2314	4778192	95088,58	43,97	Mondrian
	3392	3046500	69425,36	30,05	u-Mondrian
40	1725	6393690	104847,97	58,99	Mondrian
	2544	4067200	76710,43	40,07	u-Mondrian
50	1369	8037076	113145,88	74,33	Mondrian
	2035	5082500	83005,60	50,26	u-Mondrian

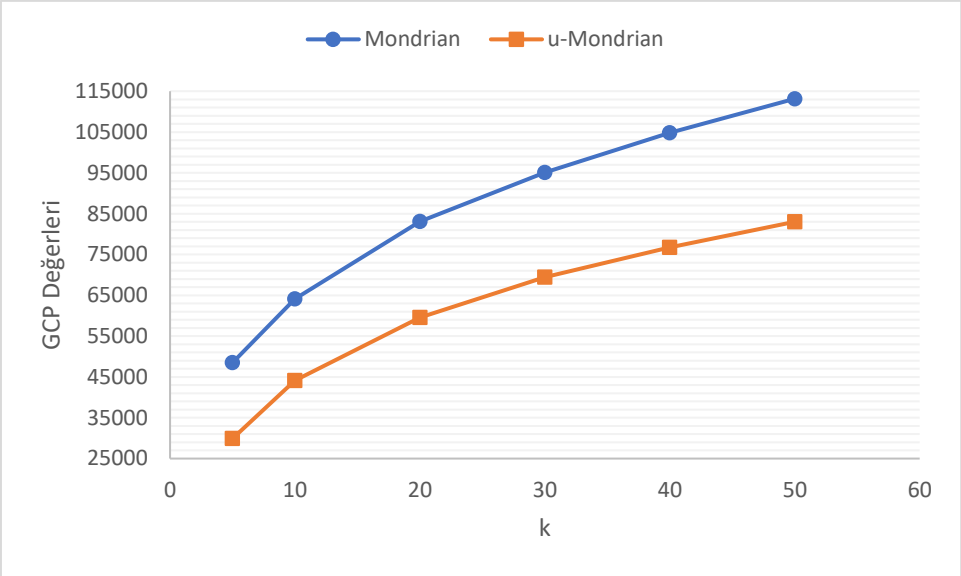
Şekil 4.27, Şekil 4.28, Şekil 4.29 ve Şekil 4.30'da farklı *k* değerleri için ölçülen ES, DM, GCP ve AECS değerleri gösterilmiştir. u-Mondrian modelinin klasik Mondrian modeline göre daha yüksek veri faydası sunduğu açıkça görülmektedir.



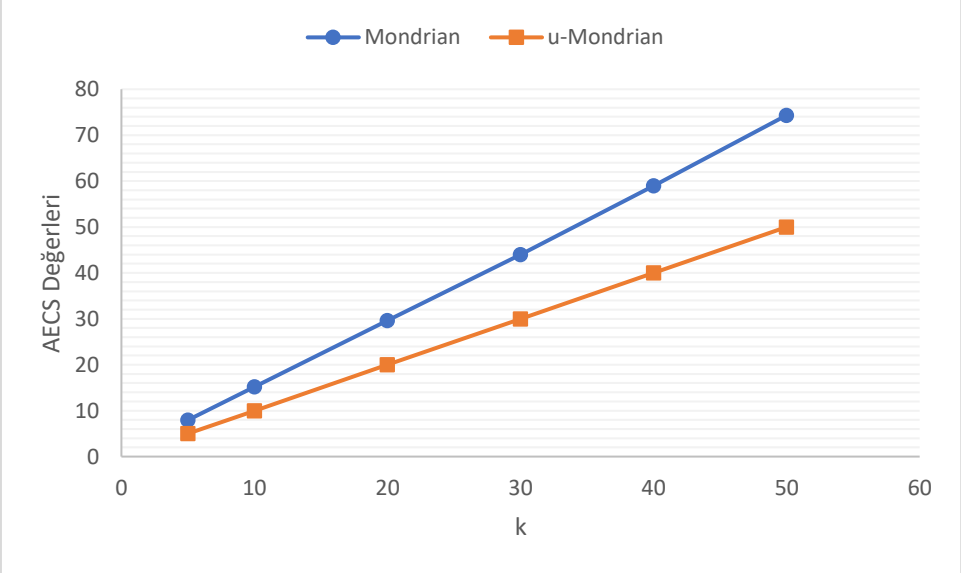
Şekil 4.27. Farklı *k* değerleri için elde edilen ES değerleri



Şekil 4.28. Farklı *k* değerleri için elde edilen DM değerleri



Şekil 4.29. Farklı *k* değerleri için elde edilen GCP değerleri



Şekil 4.30. Farklı *k* değerleri için elde edilen AECS değerleri

Çizelge 4.23'de sunulan sonuçlara göre u-Mondrian modelinin Mondrian modeline kıyasla veri faydasında yaptığı iyileştirmeler oransal olarak Çizelge 4.24'de gösterilmiştir. Bu sonuçlara göre, u-Mondrian modeli Mondrian modeline göre DM, GCP ve AECS değerlerinde sırasıyla %36,38-%52,96, %26,63-%38,32 ve %31,65-%37,14 aralıklarında daha yüksek veri faydası sunduğu görülmektedir.

Çizelge 4.24. u-Mondrian modelinin Mondrian'a göre veri faydası iyileştirme oranları

k	DM (%)	GCP (%)	AECS (%)
5	52,96	38,32	37,14
10	42,06	31,13	34,22
20	37,61	28,32	32,42
30	36,50	28,17	31,65
40	36,38	26,83	32,07
50	36,76	26,63	32,37

Relaxed Bölme Stratejisine Göre u-Mondrian ve Mondrian Modelleri İçin Elde Edilen Sonuçlar

Bu deneyde, Relaxed bölme stratejisine göre u-Mondrian ile Mondrian modelleri test edilmiş ve sonuçları sunulmuştur. Verilen tüm çizelgelerde, u-Mondrian modeli uM ile Mondrian modeli ise M ile kısaltılarak temsil edilmiştir.

Çizelge 4.25'de gösterilen sonuçlar incelendiğinde; k=5 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 634830, 44251,69 ve 6,21 değerlerini sunarken, u-Mondrian sırasıyla 508830, 39704,08 ve 5,01 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.25. k=5 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	101766	0	16384	16384	634830	634830	44251,69	44251,69	6,21	M
1	81920	19846	16384	16384	409600	409600	26291,89	26291,89	5,00	u-M
2	17310	2536	3462	19846	86550	496150	10682,33	36974,22	5,00	
3	2440	96	488	20334	12200	508350	2419,53	39393,75	5,00	
4	96	0	19	20353	480	508830	310,33	39704,08	5,05	
5	96	0	19	20353	480	508830	310,33	39704,08	5,05	

Çizelge 4.26'da gösterilen sonuçlar incelendiğinde; *k*=10 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 1266198, 63888,38 ve 12,42 değerlerini sunarken, u-Mondrian sırasıyla 1016600, 59528,64 ve 10,04 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.26. *k*=10 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	101766	0	8192	8192	1266198	1266198	63888,38	63888,38	12,42	M
1	81920	19846	8192	8192	819200	819200	39774,84	39774,84	10,00	u-M
2	14140	5706	1414	9606	141400	960600	12420,84	52195,68	10,00	
3	5120	586	512	10118	51200	1011800	6277,55	58473,23	10,00	
4	320	266	32	10150	3200	1015000	613,94	59087,17	10,00	
5	266	0	26	10176	1600	1016600	441,47	59528,64	10,23	

Çizelge 4.27'de gösterilen sonuçlar incelendiğinde; *k*=20 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 2528934, 85786,45 ve 24,84 değerlerini sunarken, u-Mondrian sırasıyla 2022400, 77304,72 ve 20,01 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.27. *k*=20 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	101766	0	4096	4096	2528934	2528934	85786,45	85786,45	24,84	M
1	81920	19846	4096	4096	1638400	1638400	53933,34	53933,34	20,00	u-M
2	10240	9606	512	4608	204800	1843200	9891,05	63824,39	20,00	
3	5120	4486	256	4864	102400	1945600	6518,79	70343,18	20,00	
4	2560	1926	128	4992	51200	1996800	4229,91	74573,09	20,00	
5	1926	0	96	5088	25600	2022400	2731,63	77304,72	20,06	

Çizelge 4.28'de gösterilen sonuçlar incelendiğinde; *k*=30 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 5057234, 111330,67 ve 49,69 değerlerini sunarken, u-Mondrian sırasıyla 3052800, 91045,87 ve 30,01 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.28. *k*=30 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	101766	0	2048	2048	5057234	5057234	111330,67	111330,67	49,69	M
1	61440	40326	2048	2048	1843200	1843200	41796,08	41796,08	30,00	u-M
2	30720	9606	1024	3072	921600	2764800	30966,40	72762,48	30,00	
3	7680	1926	256	3328	230400	2995200	12618,39	85380,87	30,00	
4	1926	0	64	3392	57600	3052800	5665,00	91045,87	30,09	
5	1926	0	64	3392	57600	3052800	5665,00	91045,87	30,09	

Çizelge 4.29'da gösterilen sonuçlar incelendiğinde; *k*=40 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 5057234, 111330,67 ve 49,69 değerlerini sunarken, u-Mondrian sırasıyla 4044800, 98504,34 ve 40,02 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.29. *k*=40 için u-Mondrian ve Mondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	101766	0	2048	2048	5057234	5057234	111330,67	111330,67	49,69	M
1	81920	19846	2048	2048	3276800	3276800	69093,40	69093,41	40,00	u-M
2	10240	9606	256	2304	409600	3686400	12558,20	81651,62	40,00	
3	5120	4486	128	2432	204800	3891200	8472,53	90124,15	40,00	
4	2560	1926	64	2496	102400	3993600	5141,08	95265,23	40,00	
5	1926	0	48	2544	51200	4044800	3239,11	98504,34	40,12	

Çizelge 4.30'da gösterilen sonuçlar incelendiğinde; *k*=50 için DM, GCP ve AECS metriklerine göre Mondrian sırasıyla 8163834, 128781,89 ve 71,97 değerlerini sunarken, u-Mondrian sırasıyla 5075000, 111447,67 ve 50,24 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Mondrian modelinin Mondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 4.30. *k*=50 için u-Mondrian ve Mondrian modellerinin test sonuçları

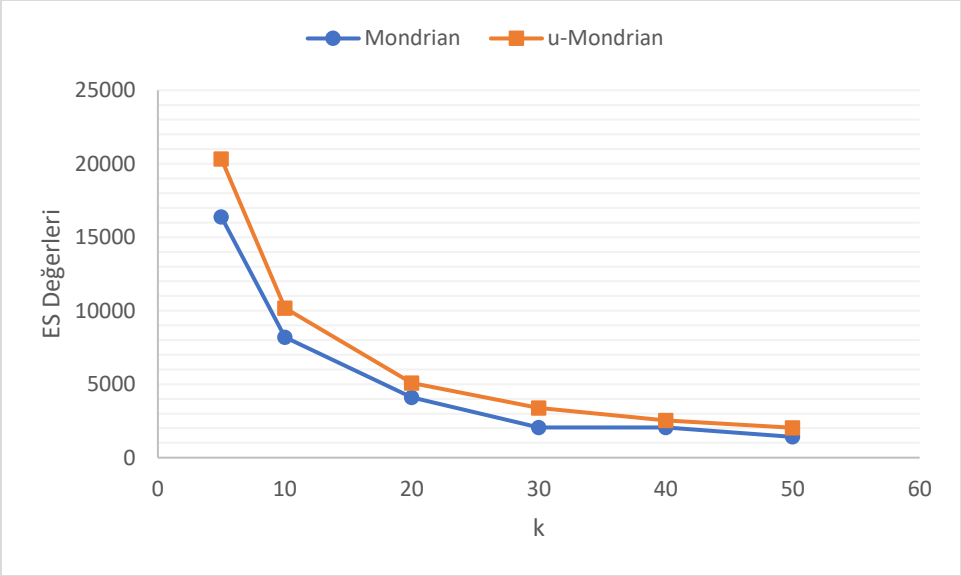
i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	101766	0	1414	1414	8163834	8163834	128781,89	128781,89	71,97	M
1	70700	31066	1414	1414	3535000	3535000	68835,18	68835,19	50,00	u-M
2	25600	5466	512	1926	1280000	4815000	31310,25	100145,44	50,00	
3	3200	2266	64	1990	160000	4975000	6116,56	106262,01	50,00	
4	1600	666	32	2022	80000	5055000	3919,89	110181,90	50,00	
5	666	0	13	2035	20000	5075000	1265,77	111447,67	51,23	

Çizelge 4.31'de u-Mondrian ile Mondrian modellerinin ES, DM, GCP ve AECS sonuçları gösterilmektedir. Bu sonuçlara göre, önerilen modelin Mondrian'a göre tüm *k* değerleri için veri faydasını büyük oranda iyileştirdiği görülmektedir.

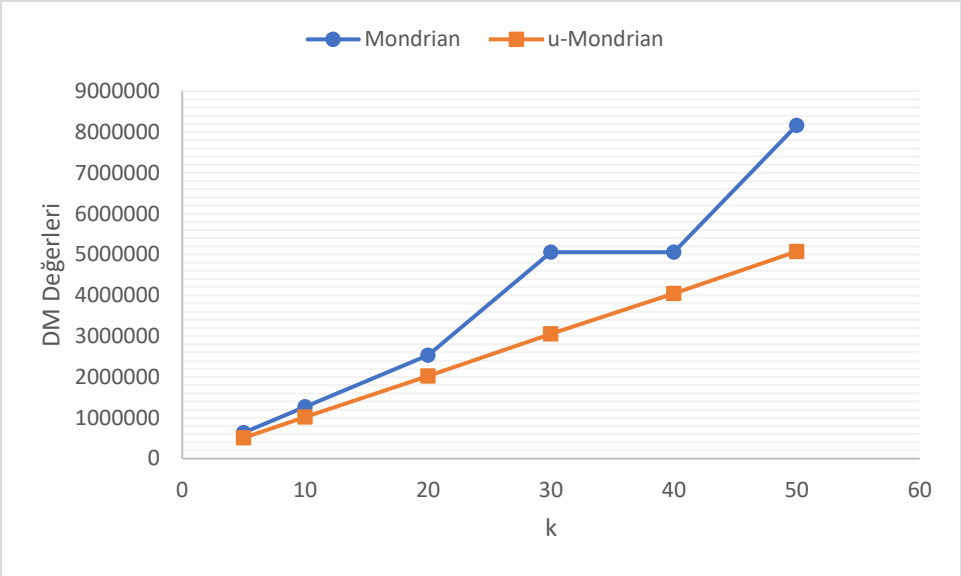
Şekil 4.31, Şekil 4.32, Şekil 4.33 ve Şekil 4.34'de farklı *k* değerleri için ölçülen ES, DM, GCP ve AECS değerleri gösterilmiştir. u-Mondrian modelinin klasik Mondrian modeline göre daha yüksek veri faydası sunduğu açıkça görülmektedir.

Çizelge 4.31. u-Mondrian ile Mondrian modellerinin genel test sonuçları

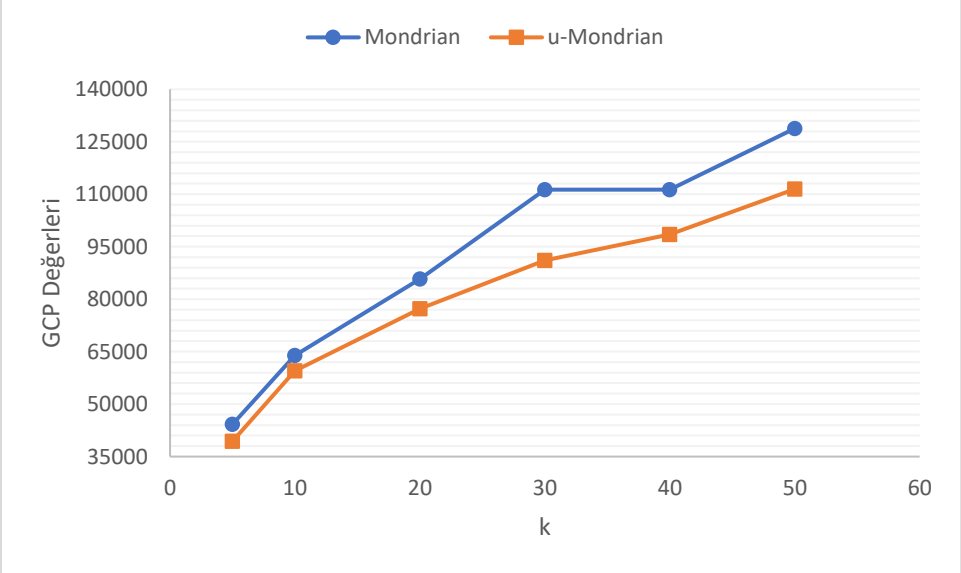
k	ES	DM	GCP	AECS	Model
5	16384	634830	44251,69	6,21	Mondrian
	20353	508830	39704,08	5,01	u-Mondrian
10	8192	1266198	63888,38	12,42	Mondrian
	10176	1016600	59528,64	10,04	u-Mondrian
20	4096	2528934	85786,45	24,84	Mondrian
	5088	2022400	77304,72	20,01	u-Mondrian
30	2048	5057234	111330,67	49,69	Mondrian
	3392	3052800	91045,87	30,01	u-Mondrian
40	2048	5057234	111330,67	49,69	Mondrian
	2544	4044800	98504,34	40,02	u-Mondrian
50	1414	8163834	128781,89	71,97	Mondrian
	2035	5075000	111447,67	50,24	u-Mondrian



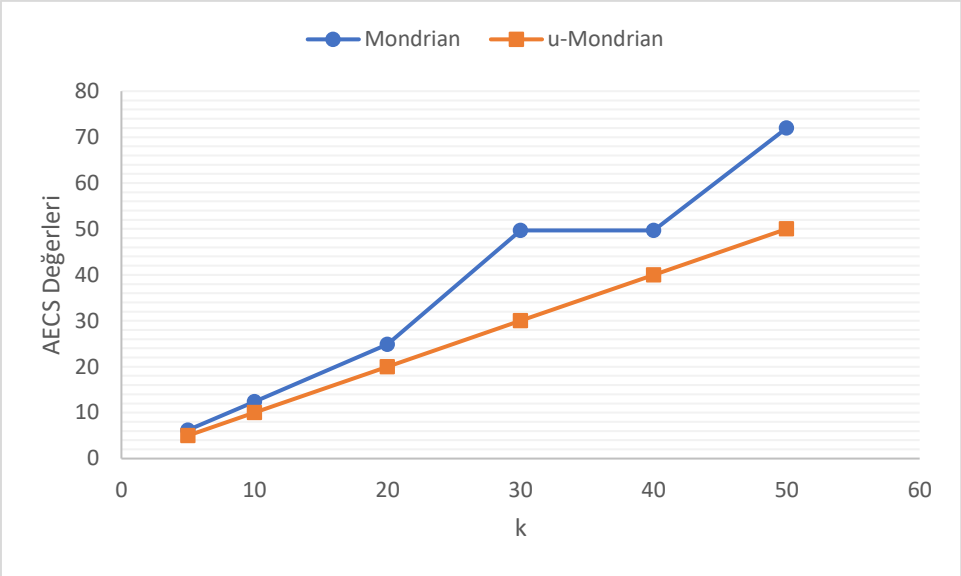
Şekil 4.31. Farklı *k* değerleri için elde edilen ES değerleri



Şekil 4.32. Farklı *k* değerleri için elde edilen DM değerleri



Şekil 4.33. Farklı *k* değerleri için elde edilen GCP değerleri



Şekil 4.34. Farklı *k* değerleri için elde edilen AECS değerleri

Çizelge 4.31'de sunulan sonuçlara göre u-Mondrian modelinin Mondrian modeline kıyasla veri faydasında yaptığı iyileştirmeler oransal olarak Çizelge 4.32'de gösterilmiştir. Bu sonuçlara göre, u-Mondrian modeli Mondrian modeline göre DM, GCP ve AECS değerlerinde sırasıyla %19,71-%39,63, %6,82-%18,22 ve %19,16-%39,60 aralıklarında daha yüksek veri faydası sunduğu görülmektedir.

Çizelge 4.32. u-Mondrian modelinin Mondrian'a göre veri faydası iyileştirme oranları

k	DM (%)	GCP (%)	AECS (%)
5	19,84	10,27	19,33
10	19,71	6,82	19,16
20	20,02	9,88	19,44
30	39,63	18,22	39,60
40	20,01	11,52	19,46
50	37,83	13,46	30,19

4.6. Bölümde Sunulan Katkılar

Bu bölümde sunulan katkılar aşağıda listelenmiştir;

- Aykırı veri yönetimi yaparak tüm aykırı verilerden fayda üreten bir model ilk defa bu tezde önerilmiş, uygulanmış ve test edilerek başarılı sonuçlar elde edilmiştir.
- Anonimleştirmede aykırı verilerin belirlenmesi için yoğunluk ve yakınlık tabanlı hibrit bir yaklaşım ilk defa bu tezde önerilmiştir.
- DM, GCP ve AECS metrikleri için aykırı veri yönetimine dayalı kapsamlı veri faydası karar kuralları ilk defa bu çalışmada oluşturulmuştur.

5. ÖNERİLEN ANONİMLEŞTİRME MODELİ: CANON

Mondrian anonimleştirme modeli [35] önceki bölümlerde de bahsedildiği gibi veri mahremiyeti çalışmalarında sıklıkla kullanılan, hiyerarşi bağımsız ve yakın optimal bir çözümdür. Veri uzayını alt uzaylara bölmek için KD-tree yapısını ve anonimleştirme için de genelleştirme operatörünü kullanır. Hesaplama maliyetinin düşük olması, hiyerarşi yapısından bağımsız olarak çalışması ve çok boyutlu genelleştirmeyi desteklemesi en önemli avantajlarıdır.

Mondrian modeli, yukarıda belirtilen avantajlara sahip olmasına rağmen, içerisinde kullandığı KD-tree yapısının sahip olduğu çeşitli dezavantajlardan da doğrudan etkilenir. KD-tree yapısı için literatürde belirtilen dezavantajlar aşağıda listelenmiştir [107,129-131];

1. Esnek olmayan veri uzayı parçalama yaklaşımı kullanması,
2. Frekans tabanlı bir bölme değeri kullanması,
3. Yüksek boyutlu verilerde etkin olmaması,
4. Çarpık dağılım sergileyen verilerde dengesiz ağaç üretmesi ve
5. Veri dağılımından bağımsız bir bölme bileşeni kullanmasıdır.

Yukarıda belirtilenden dezavantajlardan sadece 1 ve 2 numaralı olanlar Mondrian modelini etkiler. Çünkü Mondrian, KD-tree yapısını sadece veriyi parçalamak için kullanmaktadır. Belirtilen bu iki dezavantajın Mondrian modeline etkisi ise, veriden elde edilecek potansiyel faydayı düşürmeleri olarak değerlendirilmektedir. Bu sorunları mümkün olduğu ölçüde giderip potansiyel veri faydasını arttıran yeni çözümlere ihtiyaç vardır.

Bu bölümde, öznitelik uzayında temsil edilen veri kümesini birbirine yakın alt veri gruplarına parçalamak için VP-tree yapısını kullanan yeni bir anonimleştirme modeli önerilmiş, uygulanmış ve test edilmiştir. Önerilen model, her ne kadar yaklaşım olarak Mondrian modeline benzese de veri uzayını parçalama aşaması açısından farklılık göstermektedir. Ayrıca bölme stratejisi olarak Mondrian modelindeki Relaxed stratejisinden esinlenmekte olup, aynı parçalama mantığı ve üst sınır değerine sahiptir.

5.1. VP-tree Ağaç Yapısı

Vantage-point Tree (VP-tree) [132], metrik uzayda bir veri uzayını yinelemeli olarak iki alt uzaya bölen ikili bir metrik ağaçtır. Literatürde sıklıkla kullanıldığı rapor edilen [133] bu ağaç yapısı, veri uzayındaki noktalar üzerinde tanımlanmış bir uzaklık fonksiyonu aracılığıyla veri noktaları arasında bir ilişki kurulmasını sağlar. Belirlenen bir referans noktasından (vantage-point) tüm veri noktalarına olan uzaklık bir uzaklık fonksiyonu aracılığıyla hesaplanarak bu değerler üzerinden işlem yapar.

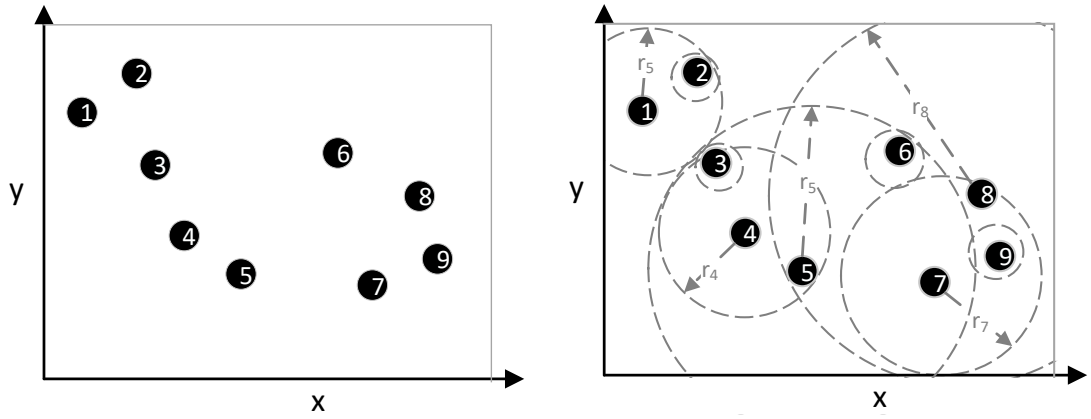
VP-tree yapısında bir düğüm aşağıdaki dört bilgiyi içerir;

- Referans noktası: bölme değerini belirlemede kullanılan veri noktası,
- Yarıçap: referans noktasının sınırını belirleyen uzaklık değeri,
- Sol alt ağaç işaretçisi: referans noktasının yarıçapı içerisinde ve üzerinde kalan verileri içeren sol alt ağaç işaretçisi ve
- Sağ alt ağaç işaretçisi: referans noktasının yarıçapı dışarısında kalan verileri içeren sağ alt ağaç işaretçisidir.

VP-tree ağacının oluşturulması işlemleri aşağıda sırasıyla belirtilmiştir [129,132,134-138];

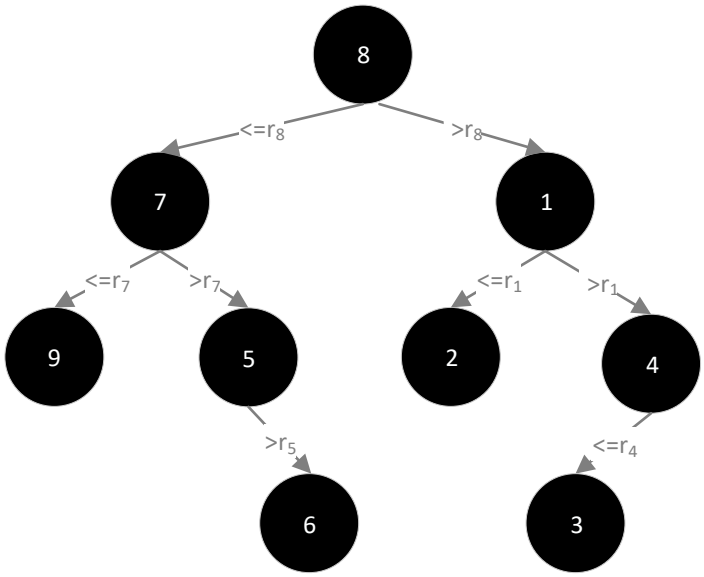
1. Öznitelik uzayında temsil edilen veri kümesinde bir referans noktası seç,
2. Seçilen referans noktasından tüm noktalara olan uzaklıkları hesapla,
3. Bu uzaklıkların medyanını bul ve bunu bölme değeri olarak kabul et,
4. Bölme değerinden küçük ve eşit uzaklığa sahip verileri sol alt ağaca, büyük uzaklığa sahip verileri ise sağ alt ağaca atayarak veri kümesini iki parçaya böl,
5. Bölünecek herhangi bir veri uzayı kalmayana kadar bu işlemleri ilk adımdan itibaren tekrar et.

Aşağıda VP-tree yapısı kullanılarak veri kümesinin alt gruplara parçalanması örnek bir senaryo üzerinde gösterilmiştir. Şekil 5.1.a'da gösterilen iki boyutlu örnek bir veri kümesinin VP-tree ile bölünmesi sonucu oluşan alt veri grupları Şekil 5.1.b'de sunulmuştur. Verilen şekilde her bir referans noktası rasgele olarak belirlenmiş ve bu değere göre veri kümesi alt gruplara parçalanmıştır. Bu parçalamalar sırasıyla; 8, 7, 9, 5, 6, 1, 2, 3 ve 4 numaralı noktalar üzerinde yapılmıştır. Elde edilen bu gruplar için oluşan ağaç yapısı Şekil 5.2'de gösterilmiş olup, siyah renkli noktalar referans noktalarını temsil etmektedir.



a) b)

Şekil 5.1. a) Örnek iki boyutlu veri kümesi, b) Veri kümesinin VP-tree ile alt veri gruplarına parçalanması



Şekil 5.2. Veri kümesinin VP-tree ile parçalanmasıyla oluşan ağaç yapısı

VP-tree yapısında uzaklık fonksiyonu olarak Öklid, Manhattan, Minkowski, Hamming vb. gibi farklı yapılar kullanılabileceği için, bu sayede veri uzayını bölmede bir esneklik sağlanabilir. Gerek veri uzayını esnek bir şekilde parçalamaya imkân tanıması gerekse de uzaklık tabanlı bir bölme değeri kullanmasından dolayı, KD-tree için yukarıda belirtilen iki olumsuzluğun VP-tree yapısıyla mümkün olduğu ölçüde giderilebileceği değerlendirilmektedir. VP-tree ile KD-tree yapılarının karşılaştırılması sonraki başlıkta verilmiştir.

5.2. VP-tree ile KD-tree Yapılarının Karşılaştırılması

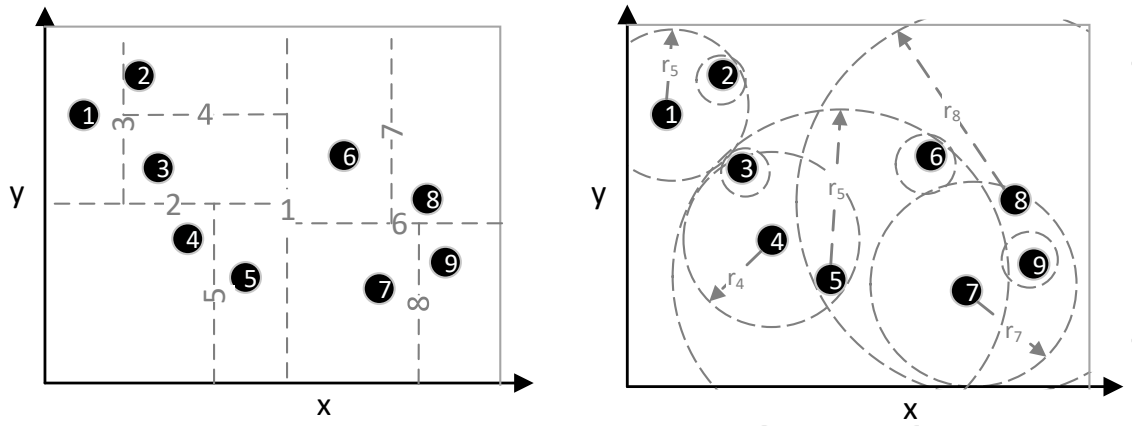
KD-tree özellikle en yakın komşuluk arama çalışmalarında hesaplama maliyetini düşürmek için sıklıkla kullanılan ikili bir ağaç yapısıdır. Literatürde, KD-tree ile VP-tree yapılarını karşılaştıran çeşitli çalışmalar [129,131,139,140] mevcuttur. Bu çalışmalarda, VP-tree'nin KD-tree'ye göre daha yüksek performans sağladığı belirtilmiştir. Ancak önerilen modelde VP-tree yapısının kullanılmasının temel nedeni, veriler arasında uzaklığa dayalı bir ilişki kurarak veri kümesini mümkün olabildiği kadar birbirine yakın veriler içerebilecek alt gruplara parçalamaktır.

Mondrian modelini etkileyen dezavantajlar için VP-tree yapısının sunduğu çözümler aşağıda açıklanmıştır;

1. Esnek olmayan veri uzayı parçalama stratejisi:
    a. KD-tree; veri uzayını eksen tabanlı bir yaklaşımla alt uzaylara böler. Seçilen bir eksen için veri uzayını, o eksende belirlediği bir bölme değerine göre sol ve sağ parça olarak iki alt kümeye ayırır. Veri uzayı yatay veya dikey eksende bölündüğü için eksen bağımlılığı mevcuttur.
    b. VP-tree; veri uzayını bir yarıçap değerine göre alt uzaylara böler. Belirlenen bir referans noktasının tüm veri noktalarına olan uzaklıklarının medyanını alarak bir yarıçap değeri belirler. Bu şekilde veri uzayını bölmede eksen bağımlılığını ortadan kaldırır ve uzaklık hesaplamada Öklid, Manhattan, Minkowski. Hamming vb. gibi farklı yapılar da kullanılabileceği için daha esnek bir yapı sunar.
2. Frekans tabanlı veri uzayı bölme değeri hesaplama stratejisi:
    a. KD-tree; belirlenen bir eksendeki verilerin frekansını hesaplayarak bu frekansların medyanını bölme değeri olarak kabul eder. Ancak böylesi bir yaklaşımın veri dağılımı göz ardı ettiği aşikârdır.
    b. VP-tree; belirlenen bir referans noktası ile diğer veri noktaları arasındaki uzaklıkları hesaplayarak bu uzaklıkların medyanını bölme değeri olarak kabul eder. Bu şekilde veri dağılımını mümkün olduğu ölçüde dikkate alır.

KD-tree ve VP-tree yapılarının bölme yaklaşımlarının karşılaştırılması Şekil 5.3'de örneklendirilmiştir. Şekil 5.1'de sunulan örnek veri kümesi için; Şekil 5.3.a'da KD-tree yapısıyla ve Şekil 5.3.b'de ise VP-tree yapısıyla parçalanması gösterilmektedir. KD-tree için verilen şekilde parçalama işlemi sırası ile 1, 2, 3, 4, 5, 6, 7 ve 8 numaralı çizgiler ile temsil

edilirken, VP-tree ile yapılan parçalamada ise sırasıyla 8, 7, 9, 5, 6, 1, 2, 4 ve 3 numaralı veri noktalarının ilgili yarıçapları için bölmeler yapılmıştır.



a) b)

Şekil 5.3. a) Veri kümesinin KD-tree ile parçalanması, b) Veri kümesinin VP-tree ile parçalanması

5.3. VP-tree Yapısının Anonimleştirmeye Adaptasyonu

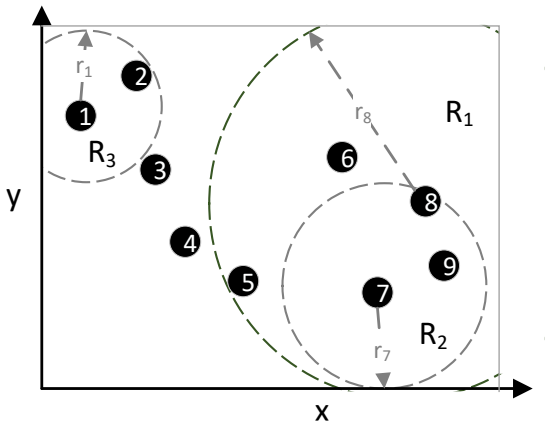
Canon modeli, veri uzayını parçalamak için VP-tree yapısını kullanmaktadır. Şekil 5.3.b'de gösterildiği üzere bir veri kümesi VP-tree yapısı ile başarılı bir şekilde alt veri gruplarına parçalanabilir. Böylesi bir yapıyı anonimleştirme kapsamında kullanabilmek için, anonimleştirmenin gereksinimlerini karşılayacak ve uygulayacak şekilde güncellemek ve adapte etmek gerekir. Bu kapsamda, Mondrian modelindeki çeşitli tanımlar alınarak bu çalışma kapsamında tekrardan tanımlanmıştır. Ayrıca, önerilen modelde bölme stratejisi Bölüm 4'de sunulan Relaxed stratejisi ile benzer olduğundan dolayı, çok boyutlu parçalama için alt sınır limiti k üst sınır limiti ise $2k-1$ olarak kabul edilmiştir.

Çok boyutlu parçalama; d boyutlu bir P veri uzayının, referans noktası (vantage-point) merkez alınarak ilgili $R_1, \dots, R_n$ alanları içerisinde bulunan çok boyutlu alt veri uzaylarına $P_1, \dots, P_n$ bölünmesidir.

Geçerli çok boyutlu parçalama; d boyutlu P veri uzayında r_i yarıçaplı R_i alanı için çok boyutlu parçalama; eğer $ElemanSayısı(R_i.P_i) \geq k$ ve $ElemanSayısı(\overline{R_i.P_i}) \geq k$ ise geçerlidir.

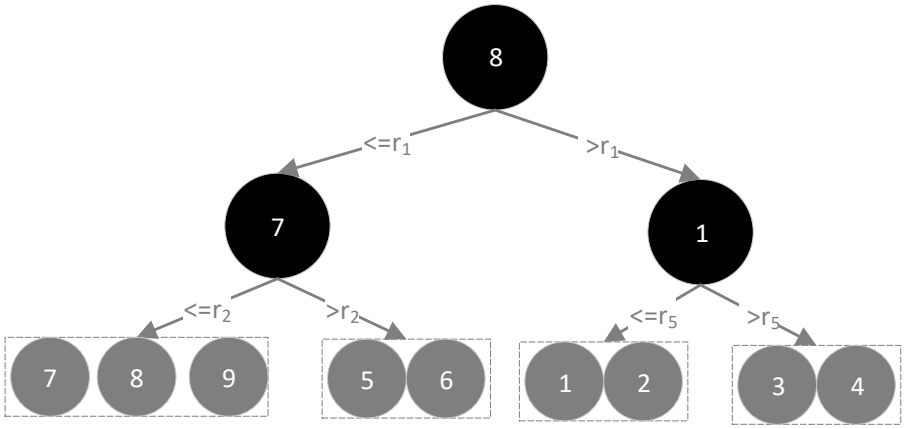
Gerekli adaptasyonu sağlayacak ilgili tanımlar verildikten sonra, yukarıda sunulan geçerli çok boyutlu parçalama işleminin VP-tree yapısına uyarlanması gerekir. Çünkü VP-tree yapısı ile alt uzaylar oluşturulurken, herhangi bir alt uzay için muhtemel bir parçalamanın gerçekleşip gerçekleşmeyeceği kontrol edilmelidir.

Geçerli bir parçalama, önceden belirlenen şartların sağlanması durumunda gerçekleşebilir. Şekil 5.4'de $k \geq 2$ için örnek bir geçerli bölme yapısı gösterilmektedir. Bu örnekte R_1, R_2 ve R_3 alanları için, $ElemanSayısı(R_i.P_i) \geq k$ ve $ElemanSayısı(\overline{R_i.P_i}) \geq k$ şartının sağlandığı açıkça görülmektedir.



Şekil 5.4. Geçerli parçalama örneği

Şekil 5.4'de gösterilen geçerli parçalama örneği için oluşan VP-tree ağacı Şekil 5.5'de sunulmuştur. Oluşturulan ağaçta siyah düğümler referans noktalarını temsil ederken, dikdörtgen kutular ile de işaretçilerin gösterdiği veri noktaları temsil edilmiştir.



Şekil 5.5. Geçerli parçalama örneği için oluşturulan VP-tree ağacı

5.4. Önerilen Anonimleştirme Modeli: Canon

Canon (Cihan Anonimleştirme), VP-tree ve Genelleştirme aşamalarını içeren ve *k*-anonimlik mahremiyet koruma modelini uygulayan çok boyutlu bir anonimleştirme modelidir. VP-tree aşamasında veri uzayını yinelemeli olarak alt parçalara bölerken, Genelleştirme aşamasında ise her bir veri parçasını kendi içerisinde genelleştirerek anonim hale getirir.

Şekil 5.6'da matematiksel ifadesi gösterilen Canon modeli şu şekilde çalışır. X, D boyutlu bir veri kümesi olsun. İlk aşamada, X'e VP-tree yapısı uygulanarak $\{P_1, P_2, \dots, P_T\} \in X$ veri parçaları elde edilir. İkinci aşamada ise, $\{P_1, P_2, \dots, P_T\} \in X$ içerisindeki her bir P_i, $gen()$ fonksiyonu kullanılarak genelleştirilir ve X^G anonim veri kümesi oluşturulur.

Girdi: veri kümesi $X = \{x_1, x_2, \dots, x_N\}$; $x_N \in R^D$, *k*-Anonimliğin *k* parametresi
Başlangıç değer ataması: $veri = X$
1) VP-tree uygula:
Yinele:
- Referans noktası seç; $vp = \{x \in X,\ x = vp\_sec(veri)\}$
- Uzaklık hesapla; $dist = \{\{d_1, d_2, \dots, d_N\} = uzaklık(vp, veri)\}$
- Medyanı bul; $\mu = \{m \in dist, m = med(dist)\}$
- Veriyi böl; $lhs = \{l \in veri,\ l_{dis} \leq \mu\}$
  $rhs = \{l \in veri,\ l_{dis} > \mu\}$
- Durdurma kriteri sağlanana kadar; $veri = lhs$ ve $veri = rhs$ için tekrarla

Dönder:
- $\{P_1, P_2, \dots, P_T\} \in X$ parçalarını dönder

2) Genelleştir: $\{P_1, P_2, \dots, P_T\} \in X$ içerisindeki her bir P_i için;
Yinele:
- Genelleştir; $P_i^G = gen(P_i)$

Dönder:
- $\{P_1^G, P_2^G, \dots, P_T^G, \} \in X^G$ anonimleştirilmiş veri parçalarını dönder

Çıktı: X^G anonim veri kümesi

Şekil 5.6. Canon modelinin matematiksel gösterimi

Önerilen modelin algoritması Şekil 5.7'de gösterilmiş olup, bu algoritmanın karmaşıklığı VP-tree yapısının kullanılmasından dolayı $O(nlogn)$ olarak elde edilir.

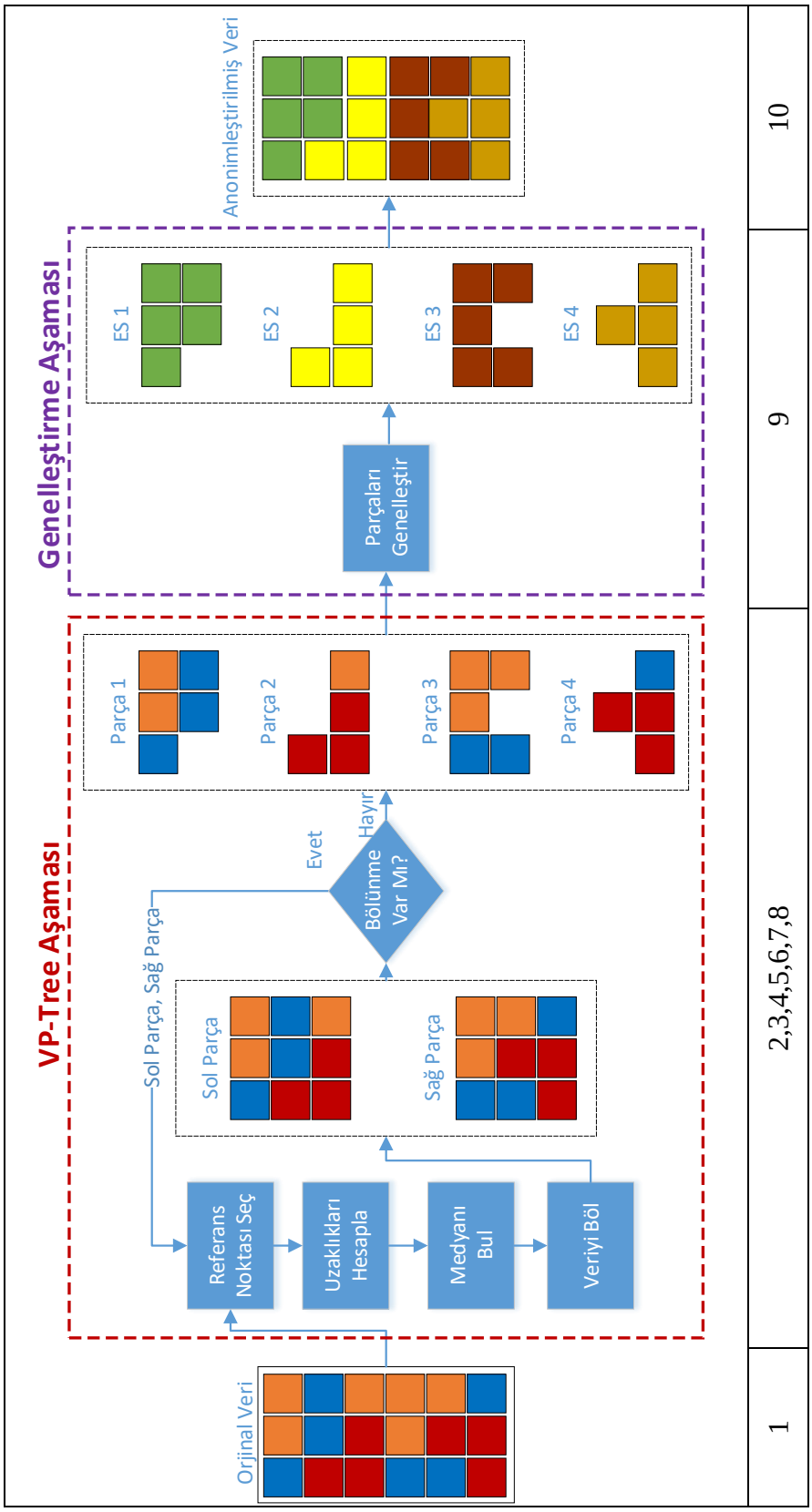
```
1: function Canon(Veri, k)
2:     if k ≤ |Veri| < 2k then
3:         return Genellestir(Veri)
4:     else
5:         vp ← VP_Sec(veri)
6:         dist = Uzaklilari_Hesapla(vp, veri)
7:         μ = Ortancayi_Bul(μ)
8:         lhs = parca ∈ Veri : parca_dist ≤ μ
9:         rhs = parca ∈ Veri : parca_dist > μ
10:        return Canon(lhs, k) ∪ Canon(rhs, k)
11:    end if
12: end function
```

Şekil 5.7. Canon modelinin algoritması

Önerilen Canon modelinin genel bir blok diyagramı Şekil 5.8'de gösterilmiş olup, bu şekil Bölüm 4'de verilen kabuller çerçevesinde oluşturulmuştur.

Şekil 5.8'de blok diyagramı gösterilen modelin çalışma adımları aşağıda sırasıyla verilmiştir;

1. Anonimleştirilecek veri kümesini al,
2. Veri kümesi içerisinde bir referans noktası seç,
3. Seçilen referans noktasından diğer tüm noktalara olan uzaklıkları hesapla,
4. Bu uzaklıkların medyanını bul ve bölme değeri olarak kabul et,
5. Bölme değerinden küçük ve eşit uzaklığa sahip verileri sol grupta, büyük uzaklığa sahip verileri ise sağ gruba atayarak veri kümesini iki parçaya böl,
6. Geçerli bölme olup olmadığı kontrol et,
7. Eğer geçerli bölme varsa Adım 2'ye git,
8. Eğer geçerli bölme yok ise veri parçalarını elde et,
9. Her bir veri parçasını genelleştirilerek eşlenik sınıflarını elde et,
10. Genelleştirilen eşlenik sınıfları birleştirerek anonim veri kümesini oluştur.



Şekil 5.8. Canon modelinin blok diyagramı

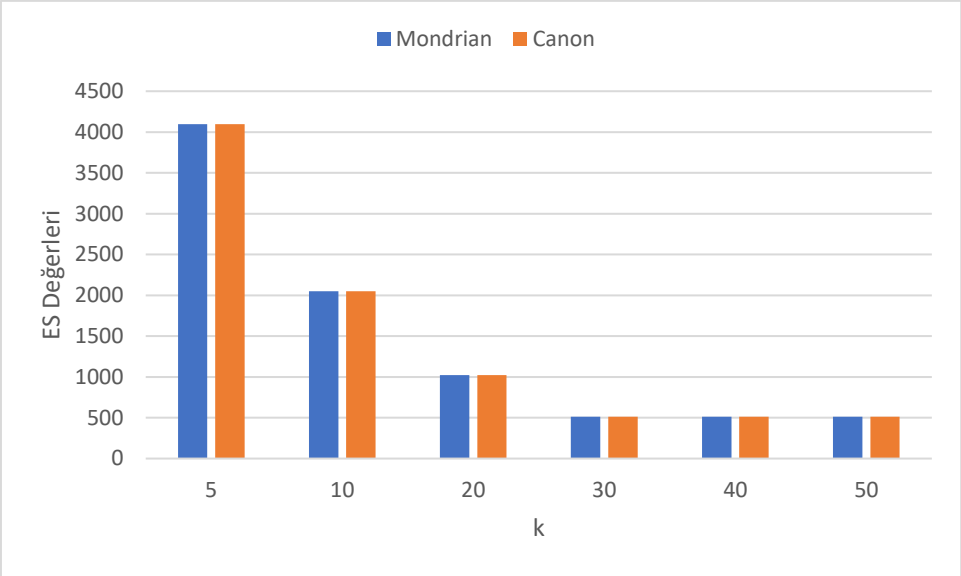
5.5 Deneysel Çalışmalar

Bu bölümde, önerilen Canon modeli test edilmiş, doğrulanmış ve Mondrian modeli ile veri faydası açısından karşılaştırılmıştır. Canon modeli bölme stratejisi olarak Relaxed stratejisinden esinlendiği için, test sonuçlarının Mondrian modeli ile karşılaştırılması bu strateji üzerinden yapılmıştır. Veri faydasını değerlendirmek için DM, GCP ve AECS metrikleri kullanılmış ve bu metriklere ek olarak ES fonksiyonu ile de eşlenik sınıfların sayısı ölçülmüştür. Adult veri kümesinin kullanıldığı bu deneyde, yaş, son ağırlık, gelir, gider ve haftalık çalışma saati öznitelikleri yarı-tanımlayıcı olarak seçilmiş ve bunlar üzerinde işlemler yapılmıştır. *k* değeri literatürde sıklıkla kullanılan değerlerden seçilmiştir.

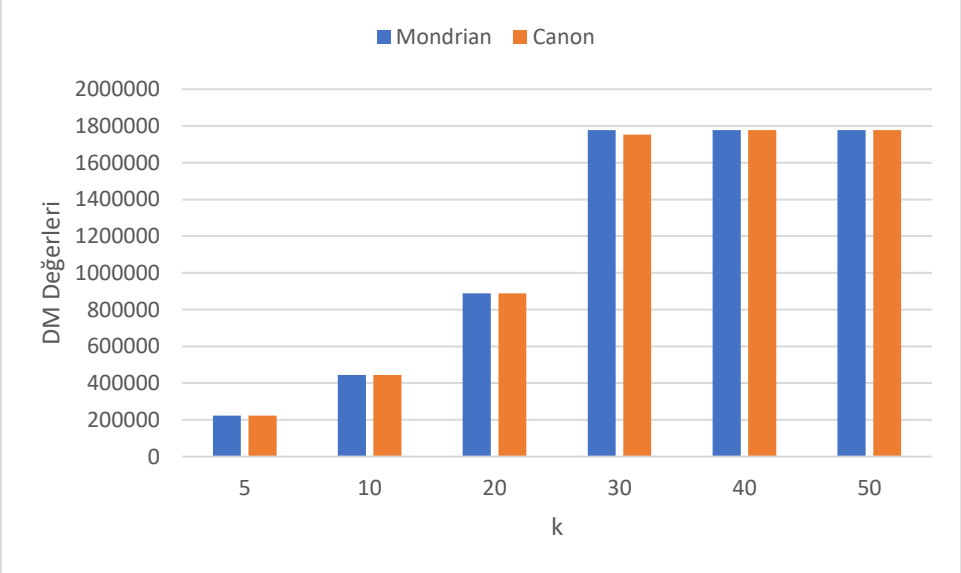
Canon ve Mondrian modelleri için elde edilen genel sonuçlar Çizelge 5.1'de gösterilmiş olup; ES, DM, GCP ve AECS için elde edilen değerler grafiksel olarak sırasıyla Şekil 5.9, Şekil 5.10, Şekil 5.11 ve Şekil 5.12'de gösterilmiştir.

Çizelge 5.1. Canon ve Mondrian modellerinin genel sonuçları

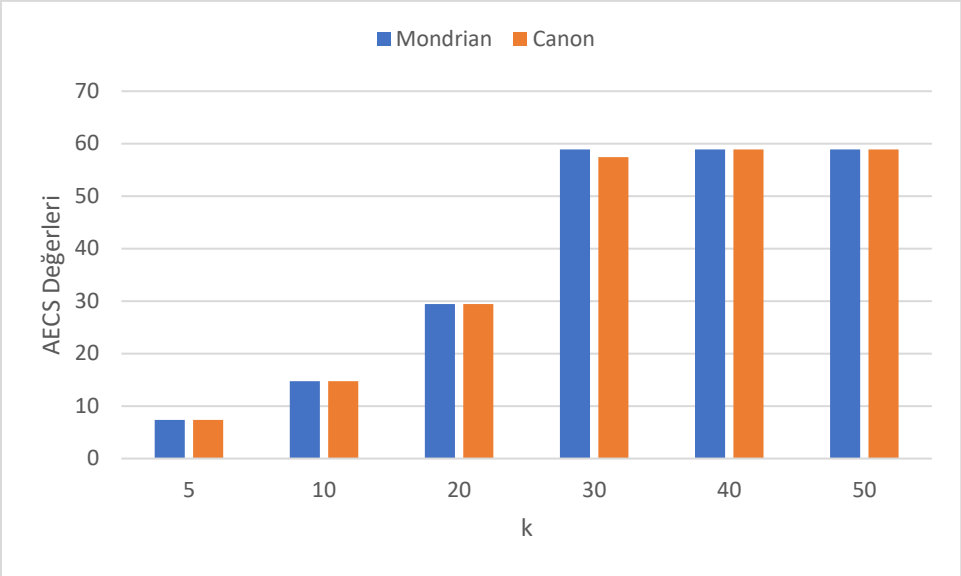
k	ES	DM	GCP	AECS	Model
5	4096	223054	6266,90	7,36	Mondrian
	4096	223160	3417,01	7,36	Canon
10	2048	444618	9561,73	14,72	Mondrian
	2048	444710	5365,65	14,72	Canon
20	1024	888678	13938,86	29,45	Mondrian
	1024	888742	7943,42	29,45	Canon
30	512	1776890	19792,52	58,91	Mondrian
	512	1753584	10961,02	57,45	Canon
40	512	1776890	19792,52	58,91	Mondrian
	512	1776920	10808,74	58,91	Canon
50	512	1776890	19792,52	58,91	Mondrian
	512	1776916	11079	58,91	Canon



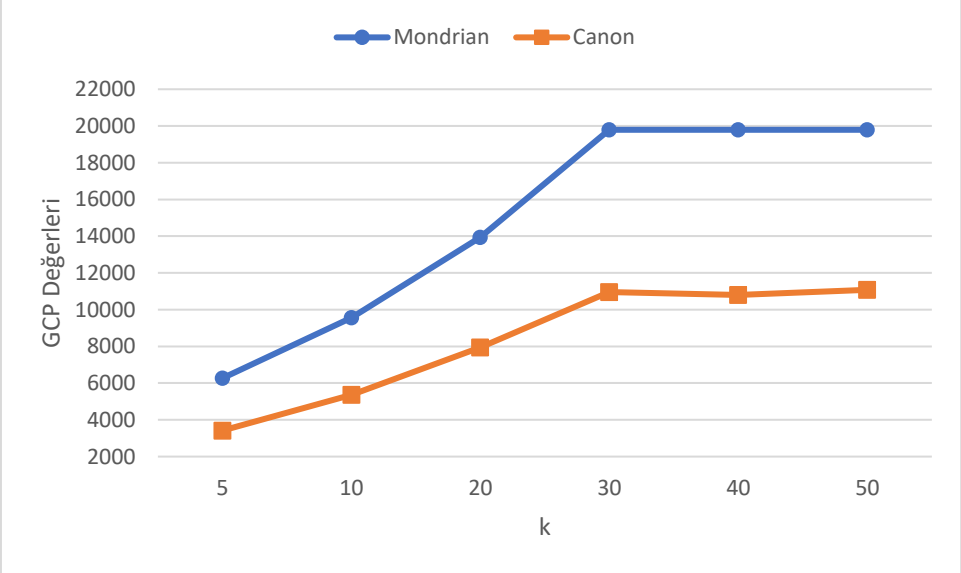
Şekil 5.9. Farklı *k* değerleri için elde edilen ES değerleri



Şekil 5.10. Farklı *k* değerleri için elde edilen DM değerleri



Şekil 5.11. Farklı *k* değerleri için elde edilen AECS değerleri



Şekil 5.12. Farklı *k* değerleri için elde edilen GCP değerleri

Şekil 5.9, Şekil 5.10 ve Şekil 5.11'de Canon ve Mondrian modellerinin ES, DM ve AECS değerlerinin neredeyse aynı olduğu görülmektedir. Bu durumun en önemli sebebi, bu metriklerinin eşlenik sınıfların sayısını, büyüklüğünü ve ortalama eleman sayısını ölçen metrikler olmasıdır. Yani, bu metrikler eşlenik sınıflardaki elemanlar arasında herhangi bir ilişkiyi gözetmekten ziyade sadece eşlenik sınıfları nicelik olarak değerlendirmektedir. Bundan dolayı, veriler arasındaki niteliği değerlendiren bir ölçek olan GCP metriğinin sonucu bu modelin başarısını değerlendirmede önemli bir kriter olacaktır. Şekil 5.12'de

gösterildiği üzere, Canon modelinin Mondrian modeline göre GCP metriği cinsinden daha yüksek veri faydası sunduğu görülmektedir. Bu sonuç, önerilen modelin doğruluğunu ortaya koymaktadır.

Çizelge 5.1'de sunulan sonuçlara göre Canon modelinin veri faydasında yaptığı iyileştirmeler oransal olarak Çizelge 5.2'de gösterilmiştir. Bu sonuçlara bakıldığında, Canon modelinin Mondrian modeline göre yüksek veri faydası sunduğu ve uygulanabilir bir çözüm olduğu görülebilir. Canon modeli Mondrian modeline göre GCP metriği için %43,01-%45,47 aralığında daha yüksek veri faydası sunmaktadır.

Çizelge 5.2. Canon modelinin Mondrian'a göre veri faydası iyileştirme oranları

k	GCP (%)
5	45,47
10	43,88
20	43,01
30	44,62
40	45,38
50	44,02

5.6. Bölümde Sunulan Katkılar

Bu bölümde sunulan katkılar aşağıda listelenmiştir;

- Mondrian modeline alternatif ve ondan daha üstün olduğu değerlendirilen yeni bir anonimleştirme modeli önerilmiş, uygulanmış ve test edilmiştir,
- Önerilen model, veri parçalama aşamasında veri dağılımını dikkate alan ilk modeldir,
- Önerilen model, esnek bir parçalama yapısı sunduğundan dolayı Mondrian modeline göre daha esnektir,
- Önerilen model, uzaklık tabanlı bir bölme gerçekleştirdiği için Relaxed stratejili Mondrian modeline göre birbirine daha yakın veri grupları oluşturarak daha yüksek veri faydası sunar.

6. ÖNERİLEN AYKIRI VERİ YÖNELİMLİ FAYDA TEMELLİ ANONİMLEŞTİRME MODELİ: U-CANON

Bu bölümde, Canon anonimleştirme modeline aykırı veri konsepti uygulayarak veri faydasını arttırmayı amaçlayan ve u-Canon (utility-aware Canon) olarak isimlendirilen yeni bir anonimleştirme modeli önerilmiş, geliştirilmiş ve test edilerek doğrulanmıştır.

Bölüm 4'de sunulan aykırı veri ve normal veri tanımları, veri faydası karar kuralları ve aykırı veri yönetim yaklaşımı bu bölümde de aynen kabul edilmekte, kullanılmakta ve uygulanmaktadır. Bundan dolayı benzer konuları tekrarlamama adına bu bölümde yukarıdaki hususlara ait detaylar verilmemiştir.

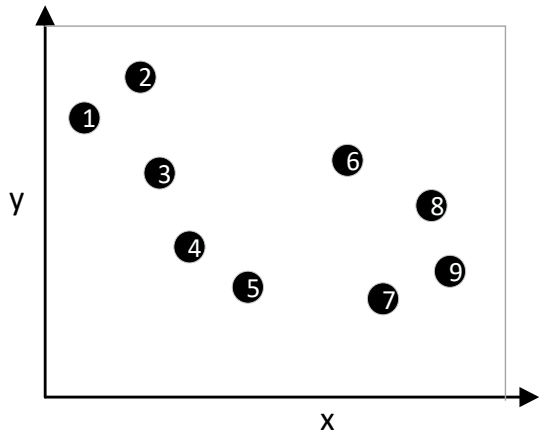
6.1. Önerilen Modelde Aykırı Verilerin Belirlenmesi ve Yönetilmesi

u-Canon, aykırı veri konseptini uygulayan bir model olduğundan dolayı yapılan kabullere uygun bir yaklaşımla aykırı verilerin belirlenmesi gerekir. Bu modelde, aykırı verilerin yönetilmesi için Bölüm 4'de sunulan aykırı veri yönetim yaklaşımı aynen uygulanmış ve tekrarı önlemek amacıyla burada verilmemiştir. Önerilen modelde, veri kümesinin VP-tree ile alt uzaylara bölünmesi sonucu elde edilen ve eleman sayıları minimum *k* ve maksimum *p* olan alt grupların her birinde birbirine en yakın *k* adet veri normal veri, arta kalan veriler ise aykırı veri olarak kabul edilmektedir.

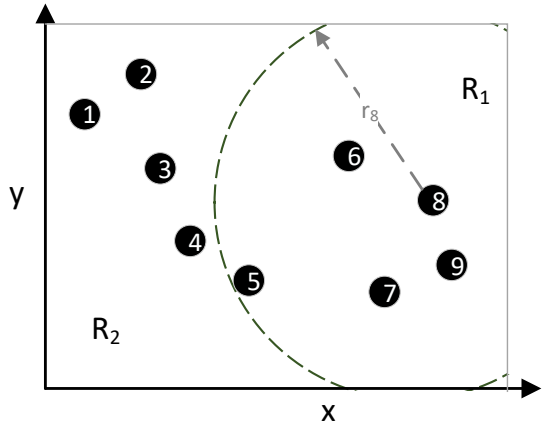
Önerilen modelde kabul edilen aykırı veri ve normal veri tanımları Bölüm 4'de sunulan tanımlarla aynı olup, aykırı verilerin belirlenmesi için yine aynı bölümde sunulan hibrit çözümden burada da faydalanılmıştır. Bu çözüm iki aşamadan oluşmakta olup, ilk aşamada LOF algoritması kullanılarak her bir alt grup içerisinde maksimum yoğunluğa sahip veri noktası bulunmakta ve bu veri noktası referans noktası olarak kabul edilmektedir. İkinci aşamada ise bu referans noktasına en yakın *k-1* adet veriyi içeren grup tespit edilerek birbirine yakın toplam *k* adet veri belirlenmektedir.

u-Canon modelinde aykırı veri belirleme için örnek bir durum aşağıda sunulmuştur. Şekil 6.1'de gösterilen iki boyutlu örnek veri kümesi ele alınsın. Bu veri kümesine *k*=3 için, r_8

yarıçapına göre VP-tree bölünmesi sonrası veri uzayı parçalanmış ve Şekil 6.2'de gösterildiği gibi R_1 ve R_2 olmak üzere iki alt veri grubu elde edilmiştir.



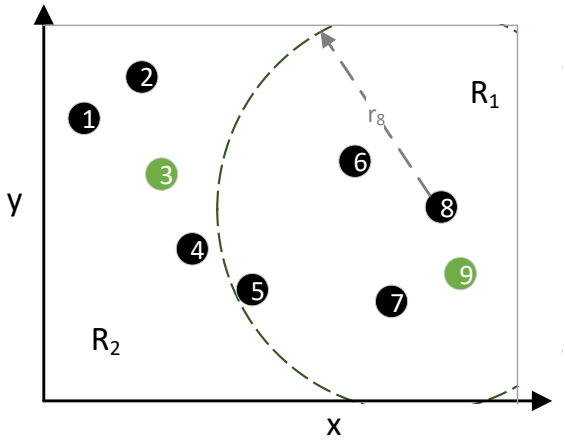
Şekil 6.1. İki boyutlu örnek veri kümesi



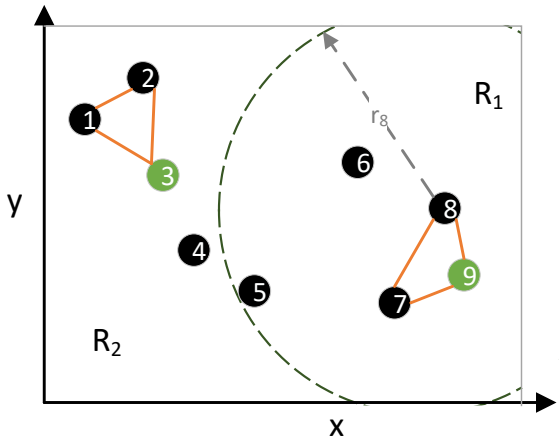
Şekil 6.2. *k*=3 için VP-tree yapısı ile veri uzayının parçalaması

Şekil 6.2'de elde edilen her bir alt veri grubuna, LOF algoritmasının uygulanmasıyla elde edilen LOF skorlarından minimum skora yani maksimum yoğunluğa sahip noktalar referans noktaları olarak kabul edilsin ve bu noktalar Şekil 6.3'de yeşil renk ile temsil edilen 3 ve 9 numaralı veri noktaları olsun.

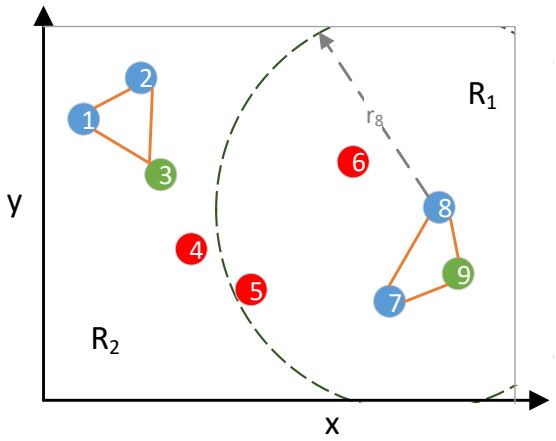
Şekil 6.4'de gösterildiği üzere referans noktalarına en yakın *k-1* adet veri noktası bulunarak birbirine en yakın *k* adet veriyi içeren veri grupları oluşturulmuştur. Şekil 6.5'de ise 1, 2 ve 3 numaralı verileri içeren grup ile 7, 8 ve 9 numaralı verileri içeren grup normal veri grupları olarak; 4, 5 ve 6 numaralı veriler ise aykırı veri grubu olarak kabul edilmiştir. Aykırı veri kümesi sonraki iterasyonlarda fayda üretilmek üzere tekrardan değerlendirilmektedir.



Şekil 6.3. Alt veri gruplarında referans noktalarının belirlenmesi



Şekil 6.4. Referans noktalarına en yakın *k-1* komşulukların bulunması



Şekil 6.5. Aykırı verilerin ve normal verilerin belirlenmesi

6.2. Önerilen Anonimleştirme Modeli: u-Canon

Bu bölümde, Canon modeline aykırı veri konsepti uygulanmasıyla elde edilen u-Canon modeli önerilmiş, modelin algoritması, akış şeması ve blok diyagramı sunulmuştur.

Canon modeli, VP-tree aşaması ve Genelleştirme aşaması olmak üzere iki aşamaya sahiptir. VP-tree aşamasında, veri kümesi alt veri gruplarına parçalanırken, oluşturulan bu veri grupları Genelleştirme aşamasında anonim hale getirilir. u-Canon modeli, klasik Canon modeline aykırı veri konsepti uygulayarak toplam veri faydasını arttırmayı amaçlayan bir anonimleştirme modelidir. u-Canon modelinde, Canon modelini genişletmek için VP-tree ile Genelleştirme aşamaları arasına "Aykırı Veri Belirleme" aşaması eklenmiştir.

Şekil 6.6'da matematiksel ifadesi verilen u-Canon modelinin çalışma prensibi şu şekildedir. X, D boyutlu bir veri kümesi olsun. İlk aşamada, X'e VP-tree yapısının uygulanması ile $\{P_1, P_2, \dots, P_T\} \in X$ alt veri grupları elde edilirken, ikinci aşamada ise $\{P_1, P_2, \dots, P_T\} \in X$ içerisindeki her bir P_i için aykırı veriler ve normal veriler belirlenir. Aykırı veriler $AykırıKümesi$'nde tutulurken, normal veriler ise $NormalKümesi$'nde tutulur. $AykırıKümesi$ iteratif bir şekilde işlenerek her bir iterasyonda elde edilen normal veriler $NormalKümesi$'ne eklenir ve böylece fayda sağlayan küme oluşturulur. Üçüncü aşamada ise, $\{P_1, P_2, \dots, P_T\} \in NormalKümesi$ içerisinde her bir P_i alt veri grubu $gen()$ fonksiyonu ile genelleştirilerek X^G anonim veri kümesi oluşturulur.

Şekil 6.7'de algoritması verilen u-Canon modelinin zaman karmaşıklığı aşağıda belirtilmiştir. M en dıştaki döngünün iterasyon sayısı, N veri kümesindeki eleman sayısı ve k ise k-Anonimlik parametresi olmak üzere;

En kötü durum analizi: $O(M * \left(NlogN + \frac{N}{k}N^2\right))$ olarak tespit edilmiştir. VP-tree algoritmasının zaman karmaşıklığı $NlogN$, elde edilen parça sayısı $\frac{N}{k}$ ve LOF algoritması için en kötü durumda zaman karmaşıklığı N^2 olmak üzere; M, N'den çok küçük olduğu için ihmal edilebilir durumda olup, elde edilen karmaşıklık $O(\frac{N^3}{k})$ olur.

En iyi durum analizi: $O(M * \left(NlogN + \frac{N}{k}NlogN\right))$ olarak tespit edilmiştir. VP-tree algoritmasının zaman karmaşıklığı $NlogN$, elde edilen parça sayısı $\frac{N}{k}$ ve LOF algoritması için en iyi durumda zaman karmaşıklığı $NlogN$ olmak üzere; M, N'den çok küçük olduğu için ihmal edilebilir durumda olup, elde edilen karmaşıklık $O(\frac{N^2 logN}{k})$ olur.

Girdi: veri kümesi $X = \{x_1, x_2, \dots, x_N\}$; $x_N \in R^D$, *k*-Anonimliğin *k* parametresi
Başlangıç değer ataması: $veri = X$
1) VP-tree uygula:
Yinele:
- Referans noktası seç; $vp = \{x \in X,\ x = rasgele(veri)\}$
- Uzaklıkları hesapla; $dist = \{\{d_1, d_2, \dots, d_N\} = uzaklık(vp, veri)\ \}$
- Medyanı bul; $\mu = \{m \in dist, m = med(dist)\}$
- Veriyi böl; $lhs = \{l \in veri,\ l_{dis} \leq \mu\}$
  $rhs = \{l \in veri,\ l_{dis} > \mu\}$
- Durdurma kriteri sağlanana kadar; $veri = lhs$, $veri = rhs$, için tekrarla

Dönder:
- $\{P_1, P_2, \dots, P_T\} \in X$ parçalarını dönder

2) Aykırı verileri belirle: $\{P_1, P_2, \dots, P_T\} \in X$ için
Yinele:
- Her bir $v \in P_i$ noktasını skorla; $skor = \{\{s_1, s_2, \dots, s_R\} = LOF(P_i)\}$
- $ref_i \in P_i$ referans noktasını belirle; $ref_i = \{g = P_i\{min(skor)\}\}$
- ref_i'nin $k-1$ en yakın komşuluğunu bul; $NN_i^{k-1} = \{x: (k-1) = argmin\|x - ref_i\|\}$
- Normal veriyi belirle; $normal_i = ref_i \cup NN_i^{k-1}$
- Aykırı veriyi belirle; $aykırı_i = P_i - normal_i$
- $normal_i$'i $NomalKümesi$'nde sakla
- $aykırı_i$'yı $AykırıKümesi$'nde sakla

Değerlendir:
- Eğer $AykırıKüme = \emptyset$; Adım 3'e git
  değilse $veri = AykırıKümesi$ ve Adım 1'e git

3) Genelleştir: $\{P_1, P_2, \dots, P_T\} \in NormalKümesi$ içerisindeki her bir P_i için
Yinele:
- Genelleştir; $P_i^G = gen(P_i)$

Dönder:
- $\{P_1^G, P_2^G, \dots, P_T^G, \} \in X^G$ anonimleştirilmiş veri parçalarını dönder

Çıktı: X^G anonim veri kümesi

Şekil 6.6. u-Canon modelinin matematiksel gösterimi

Canon modelinin zaman karmaşıklığı $O(NlogN)$'dir. Her ne kadar önerilen u-Canon modeli klasik Canon modelinden daha yüksek zaman karmaşıklığına sahip olsa da, veri faydası açısından u-Canon modeli klasik Canon'a göre daha üstündür. Önerilen modelde, zaman karmaşıklığının verinin sunacağı fayda karşısında ikinci planda olduğu değerlendirilmektedir.

```
1: P, Y, T^A ← ∅
2: S ← On_Islem(Veri)
3: function u_Canon(S)
4:     while durdurma_kriteri == false do
5:         D ← Veriyi_Parcala(S, k)
6:         for all grup ∈ D do
7:             (ODS_i, UDS_i) ← Aykiri_Verileri_Bul(grup, k)
8:             P ← P ∪ UDS_i
9:             Y ← Y ∪ ODS_i
10:        end for
11:        S ← Y
12:    end while
13:    if durdurma_kriteri == true then
14:        C ← P ∪ Y
15:        for all grup ∈ C do
16:            grup^G ← Genellestir(grup)
17:            T^A ← T^A ∪ grup^G
18:        end for
19:        Yayinla(T^A)
20:    end if
21: end function
```

Şekil 6.7. u-Canon modelinin algoritması

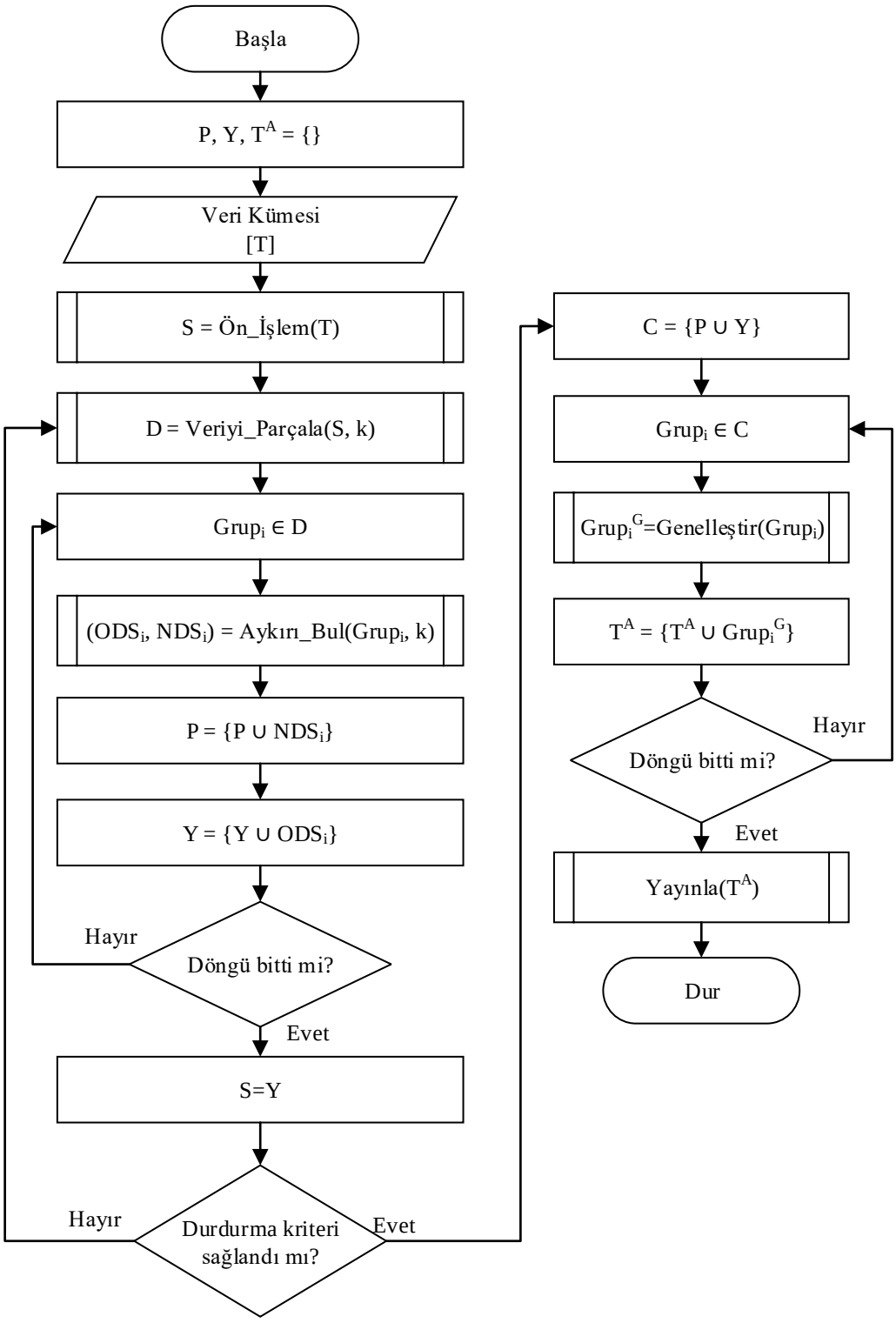
u-Canon modeline ait akış şeması Şekil 6.8'de gösterilmekte olup, akıştaki her adım aşağıda sırasıyla açıklanmıştır;

1. T veri kümesi modele girdi olarak verilir,
2. Ön işlem aşamasında özniteliklerin sınıfları belirlenir, normalizasyon işlemi yapılır, durdurma kriteri belirlenir ve S veri kümesi elde edilir,
3. S veri kümesi VP-tree yapısı ile her biri minimum *k* maksimum *p* adet veri içeren alt gruplara bölünerek bu grupları içeren D kümesi elde edilir,
4. D kümesi içerisindeki her bir grup için;
    i. Aykırı veriler ve normal veriler belirlenerek NDS_i ve ODS_i kümeleri oluşturulur,
    ii. NDS_i, P kümesine eklenir,
    iii. ODS_i, Y kümesine eklenir,

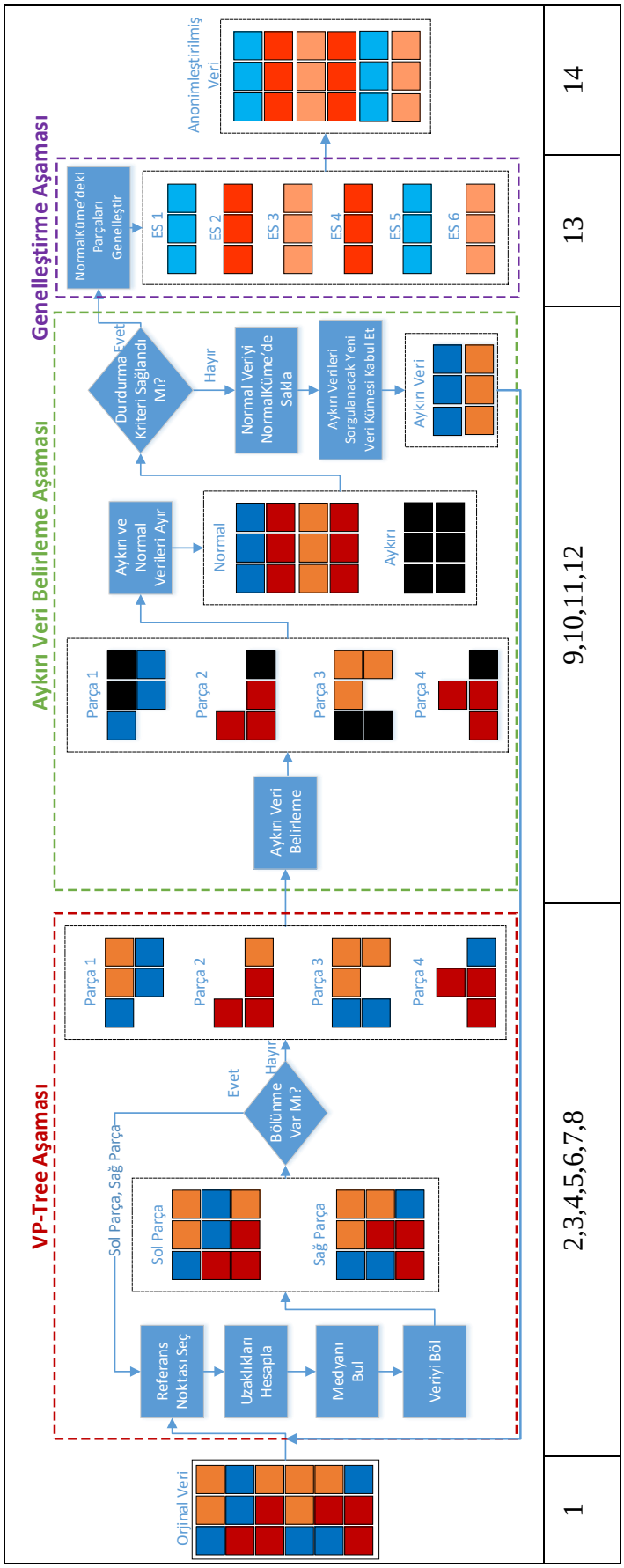
5. Eğer durdurma kriteri sağlanmadıysa irdelenecek yeni veri kümesi S = Y olur ve Adım 3'e gidilir,
6. Eğer durdurma kriteri sağlandıysa, P kümesi Y ile birleştirilerek C kümesi elde edilir,
7. C kümesindeki her bir grup için;
    i. Grup genelleştirilir,
    ii. Genelleştirilen grup T^A kümesine eklenir
8. T^A anonim veri kümesi yayınlanır.

Şekil 6.9'da sunulan u-Canon modelinin blok diyagramı Bölüm 4'de belirtilen kabuller çerçevesinde oluşturulmuş ve çalışma prensibi aşağıda sırasıyla verilmiştir;

1. Anonimleştirilecek veri kümesini al,
2. Bir referans noktası seç,
3. Referans noktasından tüm noktalara olan uzaklıkları hesapla,
4. Uzaklıkların medyanını bul,
5. Medyan değerini kullanarak veri uzayını Sol Parça ve Sağ Parça olmak üzere iki alt kümeye parçala,
6. Bölünebilen veri uzayı kalıp kalmadığını kontrol et,
7. Eğer kalmış ise Adım 2'ye git,
8. Eğer kalmamış ise alt ve üst sınırları sağlayan parçaları elde et,
9. Her bir parça için aykırı verileri belirle,
10. Veriyi normal ve aykırı veri olarak iki alt kümeye ayır,
11. Durdurma kriterinin sağlanıp sağlanmadığını kontrol et,
12. Eğer sağlanmamışsa, normal veriyi normal kümesinde sakla, aykırı verileri sorgulanacak yeni veri kümesi olarak kabul et ve Adım 1'e git,
13. Normal veri kümesindeki her bir veri parçasını genelleştir ve eşlenik sınıfları elde et,
14. Tüm eşlenik sınıfları birleştirerek anonim veri kümesini oluştur.

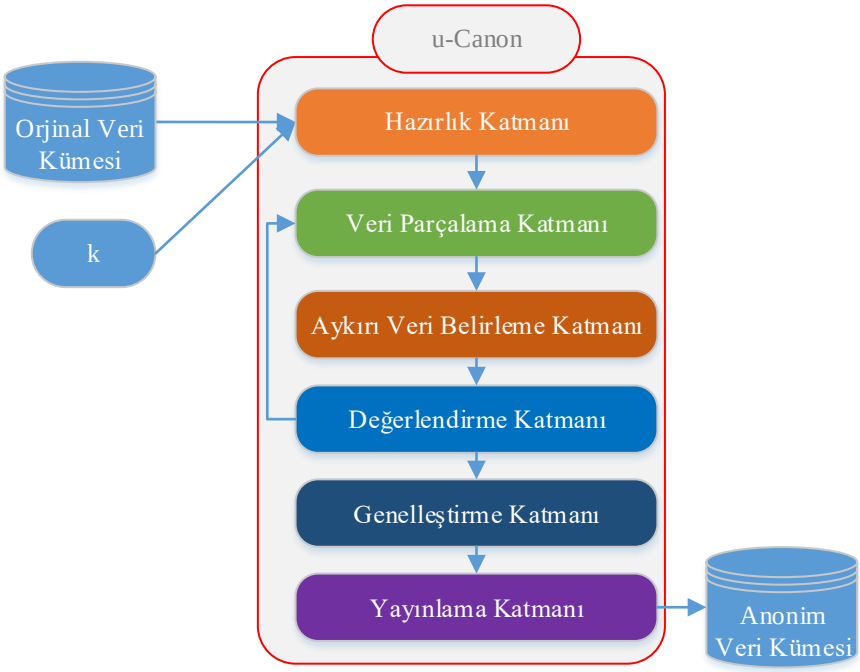


Şekil 6.8. u-Canon modelinin akış diyagramı



Şekil 6.9. u-Canon modelinin blok diyagramı

Şekil 6.10'da gösterildiği üzere, u-Canon modeli hazırlık, veri parçalama, aykırı veri belirleme, değerlendirme, genelleştirme ve yayınlama olmak üzere toplam 6 katmanlı bir yapıya sahiptir. Hazırlık katmanında anonimleştirilecek veri kümesi işlenmeye hazır hale getirilirken, veri parçalama katmanında veri kümesi VP-tree yapısı ile alt veri gruplarına ayrılır. Aykırı veri belirleme katmanında aykırı veriler ve normal veriler belirlenir. Değerlendirme katmanında durdurma kriterinin sağlanıp sağlanmadığı kontrol edilir. Genelleştirme katmanında, normal veri kümesi içerisindeki her alt grup kendi içerisinde genelleştirilerek anonim hale getirilir. Yayınlanma katmanında ise anonim veri grupları birleştirilerek yayınlanır. Bu katmanlardan hazırlık, aykırı veri belirleme, değerlendirme, genelleştirme ve yayınlama katmanları Bölüm 4'de sunulan u-Mondrian modeli ile aynı yapıya sahip olduğu için tekrarı engellemek amacıyla bu bölümde verilmemiş, sadece Veri Parçalama Katmanı aşağıda açıklanmıştır.



Şekil 6.10. u-Canon modelinin katmanlı yapısı

Veri parçalama katmanında, Canon modelinde kullanılan VP-tree ağacı ile veri kümesi her birinde minimum *k* maksimum *p* adet veri içerecek şekilde alt gruplara parçalanır. Bu şekilde veri uzayı alt uzaylara bölünerek birbirine olabildiği kadar yakın eleman grupları oluşturulur. Şekil 6.11'de veri parçalama katmanı için sunulan algoritma gösterilmektedir.

```
1: function Veriyi_Parcala(S, k)
2:     D_i ← VP_Tree(S, k)
3:     return (D_i)
4: end function
```

Şekil 6.11. Veri parçalama katmanı algoritması

6.3. Deneysel Çalışmalar

Bu bölümde, önerilen u-Canon modelini test etmek için gerçekleştirilen deneyler verilmiş ve elde edilen sonuçlar sunulmuştur.

Deneysel çalışmalar, Intel Core i3 2.3 GHz işlemci ve 4 GB ram özelliğine sahip bilgisayar üzerinde gerçekleştirilmiştir. Adult veri kümesi ile yapılan testlerde, farklı *k* değerleri için veri faydası DM, GCP ve AECS metrikleri ile ölçülmüştür. Elde edilen sonuçlar tablolar halinde sunulmuş, son olarak bu sonuçları daha kolay yorumlayabilmek amacıyla grafiklerle gösterilmiştir.

Bu deneyde, durdurma kriteri iterasyon sayısı olarak belirlenmiş ve bu sayı 5'e sabitlenmiştir. Tüm tablolarda klasik Canon modeli C ile u-Canon modeli ise u-C ile temsil edilmiştir. Yapılan tüm deneyler, önerilen modelin veri faydasında önemli artışlar sağladığını göstermektedir.

Çizelge 6.1'de gösterilen sonuçlar incelendiğinde; *k*=5 için DM, GCP ve AECS metriklerine göre Canon sırasıyla 223160, 3417,02 ve 7,36 değerlerini sunarken, u-Canon sırasıyla 150532, 2504,69 ve 5,00 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Canon modelinin Canon modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 6.1. *k*=5 için u-Canon ve Canon modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	4096	4096	223160	223160	3417,02	3417,02	7,36	C
1	20480	9683	4096	4096	102391	102391	1220,74	1220,74	5,00	u-C
2	5150	4533	1030	5126	25741	128132	448,16	1668,90	5,00	
3	2560	1973	512	5638	12800	140932	332,99	2001,89	5,00	
4	1280	693	256	5894	6400	147332	267,80	2269,69	5,00	
5	693	0	138	6032	3200	150532	235,00	2504,69	5,02	

Çizelge 6.2'de gösterilen sonuçlar incelendiğinde; *k*=10 için DM, GCP ve AECS metriklerine göre Canon sırasıyla 444710, 5365,66 ve 14,72 değerlerini sunarken, u-Canon sırasıyla 300800, 347,78 ve 10,00 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Canon modelinin Canon modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 6.2. *k*=10 için u-Canon ve Canon modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	2048	2048	444710	444710	5365,66	5365,66	14,72	C
1	20480	9682	2048	2048	204800	204800	2115,14	2115,15	10,00	u-C
2	5120	4562	512	2560	51200	256000	761,66	2876,81	10,00	
3	2560	2002	256	2816	25600	281600	546,44	3423,25	10,00	
4	1280	722	128	2944	12800	294400	421,63	3844,88	10,00	
5	722	0	72	3016	6400	300800	347,78	4192,67	10,02	

Çizelge 6.3'de gösterilen sonuçlar incelendiğinde; *k*=20 için DM, GCP ve AECS metriklerine göre Canon sırasıyla 888742, 7943,43 ve 29,45 değerlerini sunarken, u-Canon sırasıyla 601600, 6306,42 ve 20,00 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Canon modelinin Canon modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 6.3. *k*=20 için u-Canon ve Canon modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	1024	1024	888742	888742	7943,43	7943,43	29,45	C
1	20480	9682	1024	1024	409600	409600	3215,62	3215,62	20,00	u-C
2	5120	4562	256	1280	102400	512000	1066,99	4282,61	20,00	
3	2560	2002	128	1408	51200	563200	852,97	5135,58	20,00	
4	1280	722	64	1472	25600	588800	677,16	5812,74	20,00	
5	722	0	36	1508	12800	601600	493,68	6306,42	20,05	

Çizelge 6.4'de gösterilen sonuçlar incelendiğinde; *k*=30 için DM, GCP ve AECS metriklerine göre Canon sırasıyla 1753584, 10961,02 ve 58,91 değerlerini sunarken, u-Canon sırasıyla 892800, 7406,09 ve 30,05 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Canon modelinin Canon modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 6.4. *k*=30 için u-Canon ve Canon modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	512	512	1753584	1753584	10961,02	10961,02	58,91	C
1	15360	14802	512	512	460800	460800	2729,79	2729,79	30,00	u-C
2	7680	7122	256	768	230400	691200	1820,42	4550,21	30,00	
3	3840	3282	128	896	115200	806400	1331,40	5881,61	30,00	
4	1920	1362	64	960	57600	864000	850,92	6732,53	30,00	
5	1362	0	45	1005	28800	892800	673,56	7406,09	30,26	

Çizelge 6.5'de gösterilen sonuçlar incelendiğinde; *k*=40 için DM, GCP ve AECS metriklerine göre Canon sırasıyla 1776920, 10808,74 ve 58,91 değerlerini sunarken, u-Canon sırasıyla 1203200, 9071,57 ve 40,02 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Canon modelinin Canon modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 6.5. *k*=40 için u-Canon ve Canon modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	512	512	1776920	1776920	10808,74	10808,74	58,91	C
1	20480	9682	512	512	819200	819200	4595,49	4595,49	40,00	u-C
2	5120	4562	128	640	204800	1024000	1716,31	6311,80	40,00	
3	2560	2002	64	704	102400	1126400	1201,14	7512,94	40,00	
4	1280	722	32	736	51200	1177600	843,67	8356,61	40,00	
5	722	0	18	754	25600	1203200	714,96	9071,57	40,11	

Çizelge 6.6'da gösterilen sonuçlar incelendiğinde; *k*=50 için DM, GCP ve AECS metriklerine göre Canon sırasıyla 1776916, 11079 ve 58,91 değerlerini sunarken, u-Canon sırasıyla 1505000, 10505,04 ve 50,08 değerlerini sunmaktadır. İlgili karar kurallarına göre, u-Canon modelinin Canon modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 6.6. *k*=50 için u-Canon ve Canon modellerinin test sonuçları

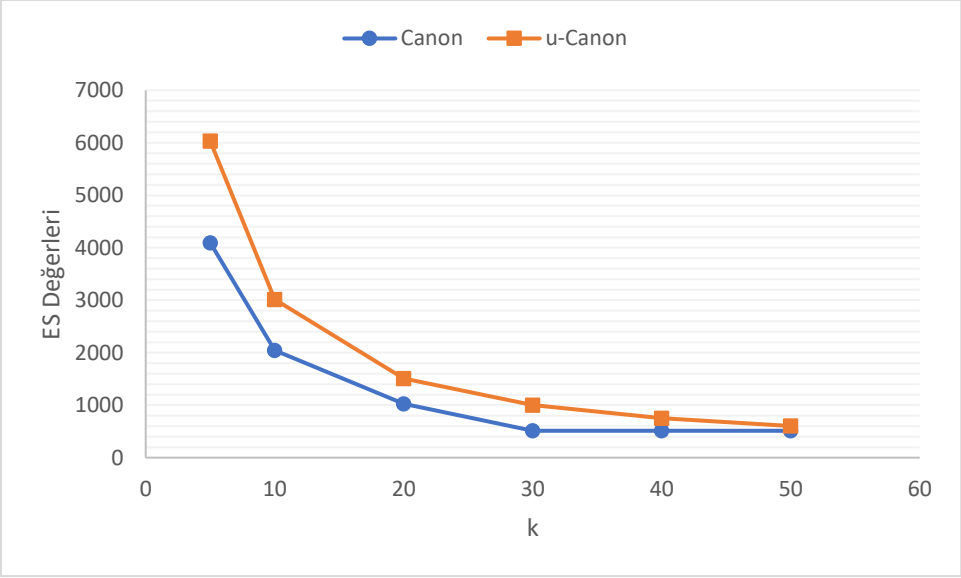
i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	30162	0	512	512	1776916	1776916	11079,00	11079,00	58,91	C
1	25600	4562	512	512	1280000	1280000	7568,57	7568,57	50,00	u-C
2	3200	1362	64	576	160000	1440000	1708,94	9277,52	50,00	
3	800	562	16	592	40000	1480000	621,58	9899,10	50,00	
4	400	162	8	600	20000	1500000	456,47	10355,58	50,00	
5	162	0	3	603	5000	1505000	149,46	10505,04	54,00	

Çizelge 6.7'de u-Canon ile Canon modellerinin ES, DM, GCP ve AECS sonuçları gösterilmektedir. Bu sonuçlara ve karar kurallarına göre, önerilen modelin Canon modeline göre tüm *k* değerleri için veri faydasını büyük oranda iyileştirdiği görülmektedir.

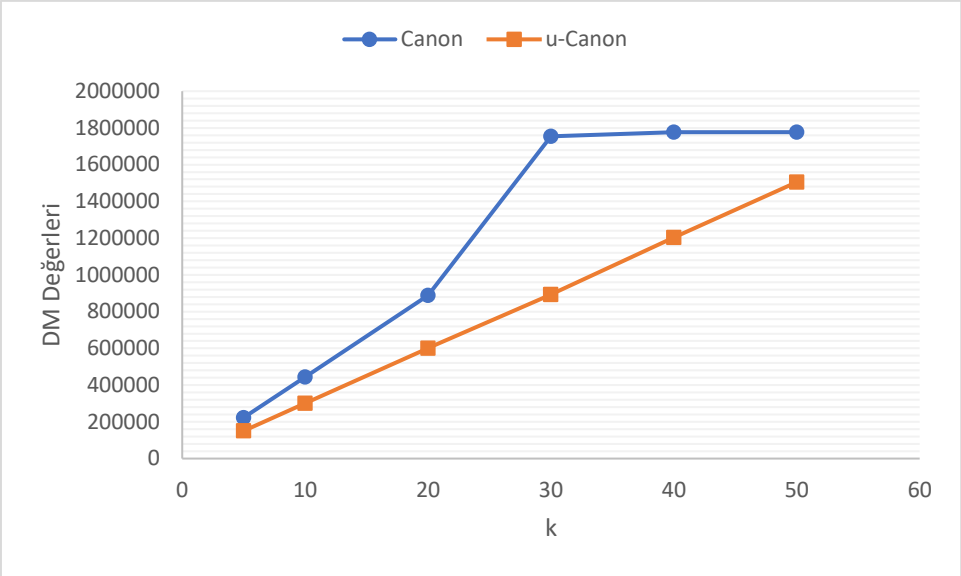
Şekil 6.12, Şekil 6.13, Şekil 6.14 ve Şekil 6.15'de her iki model için farklı *k* değerlerine göre ölçülen ES, DM, GCP ve AECS değerleri sırasıyla gösterilmiştir. Verilen tüm bu şekillerde, u-Canon modelinin Canon modeline göre daha yüksek veri faydası sunduğu açıkça görülmektedir.

Çizelge 6.7. u-Canon ve Canon modellerinin genel sonuçları

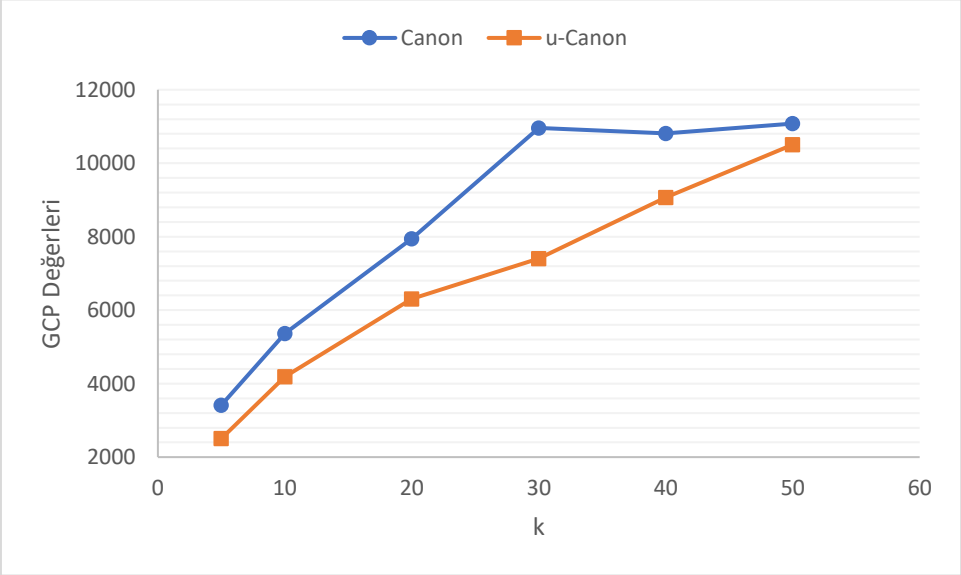
k	ES	DM	GCP	AECS	Model
5	4096	223160	3417,02	7,36	Canon
	6032	150532	2504,69	5,00	u-Canon
10	2048	444710	5365,66	14,72	Canon
	3016	300800	4192,67	10,00	u-Canon
20	1024	888742	7943,43	29,45	Canon
	1508	601600	6306,42	20,01	u-Canon
30	512	1753584	10961,02	58,91	Canon
	1005	892800	7406,09	30,05	u-Canon
40	512	1776920	10808,74	58,91	Canon
	754	1203200	9071,57	40,02	u-Canon
50	512	1776916	11079,00	58,91	Canon
	603	1505000	10505,04	50,8	u-Canon



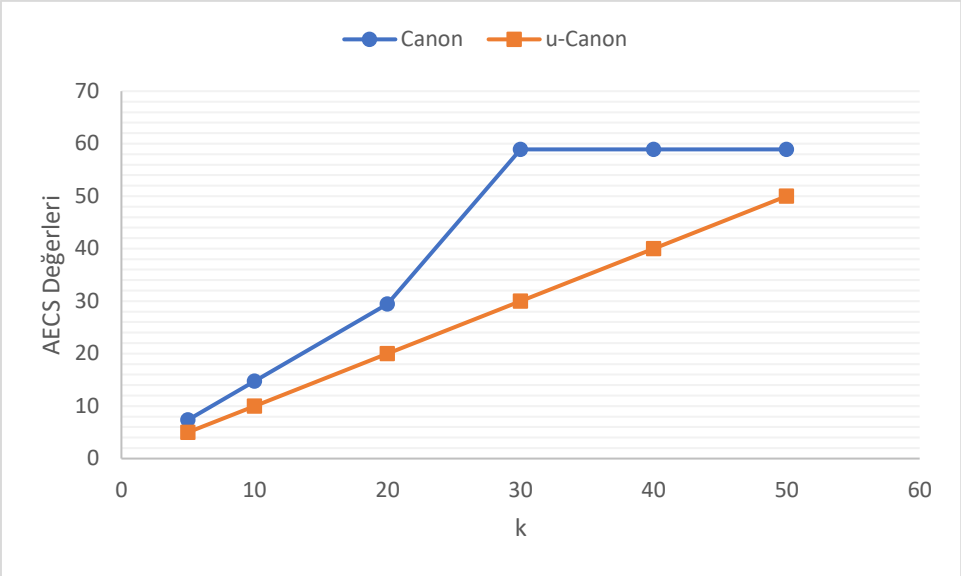
Şekil 6.12. Farklı *k* değerleri için elde edilen ES değerleri



Şekil 6.13. Farklı *k* değerleri için elde edilen DM değerleri



Şekil 6.14. Farklı *k* değerleri için elde edilen GCP değerleri



Şekil 6.15. Farklı *k* değerleri için elde edilen AECS değerleri

Çizelge 6.7'de sunulan sonuçlara göre u-Canon modelinin Canon modeline kıyasla veri faydasında yaptığı iyileştirmeler oransal olarak Çizelge 6.8'de gösterilmiştir. Bu sonuçlara göre, u-Canon modeli Canon modeline göre DM, GCP ve AECS değerlerinde sırasıyla %15,30-%49,08, %5,18-%32,43 ve %13,76-%48,99 aralıklarında daha yüksek veri faydası sunduğu görülmektedir.

Çizelge 6.8. u-Canon modelinin Canon'a göre veri faydası iyileştirme oranları

k	DM (%)	GCP (%)	AECS (%)
5	32,54	26,69	32,06
10	32,36	21,86	32,06
20	32,30	20,60	32,05
30	49,08	32,43	48,99
40	32,28	16,07	32,06
50	15,30	5,18	13,76

6.4. Bölümde Sunulan Katkılar

Bu bölümde sunulan katkılar aşağıda listelenmiştir;

- Canon modeline aykırı veri konsepti uygulayarak tüm aykırı verilerden fayda üreten ve veri dağılımının dikkate alan bir model ilk defa bu tezde önerilmiş, uygulanmış ve test edilerek başarılı sonuçlar elde edilmiştir.
- Canon modeline aykırı veri konsepti uygulayarak daha yüksek veri faydası sunan yeni bir anonimleştirme modeli geliştirilmiş ve başarı ile test edilmiştir.

7. ÖNERİLEN AYKIRI VERİ YÖNELİMLİ FAYDA TEMELLİ BÜYÜK VERİ ANONİMLEŞTİRME MODELİ: SU-MONDRIAN

Bu bölümde, Su-Mondrian (Spark-based utility-aware Mondrian) modeli olarak adlandırılan ve büyük veri mimarisi üzerinde geliştirilen aykırı veri yönelimli fayda temelli yeni bir anonimleştirme modeli ilk defa önerilmiş, uygulanmış ve test edilerek doğrulanmıştır. Önerilen model için kabul edilen aykırı veri ve normal veri tanımları verilmiş, Bölüm 4'de oluşturulan ve sunulan veri faydası karar kuralları bu bölümde de aynen kullanılmıştır. Veri anonimleştirme ve aykırı veri belirlemede kullanılan dağıtık yaklaşımlar sunulmuş, önerilen model detaylandırılarak akış şeması, algoritması ve alt katmanları verilmiştir. Son olarak önerilen model test edilerek doğrulanmıştır.

7.1. Önerilen Model İçin Kabul Edilen Aykırı Veri ve Normal Veri Tanımları

Önerilen modelde, Spark tabanlı Mondrian anonimleştirme modeli (SMondrian) aykırı veri konseptine uygun olarak veri faydası odaklı genişletildiğinden dolayı, Su-Mondrian modeli kapsamında aykırı veri ve normal veri tanımları için yapılan kabuller aşağıda sunulmuştur;

Normal veri; V dağıtık bir veri kümesi, $v_T \in V$ bir alt küme, $k \leq |v_T| \leq p$ olmak üzere, v_T kümesi içerisinde birbirine en yakın k adet veri normal veridir.

Aykırı veri; V dağıtık bir veri kümesi, $v_T \in V$ bir alt küme, $k \leq |v_T| \leq p$ olmak üzere, v_T kümesi içerisinde normal veri haricinde kalan veriler aykırı veridir.

7.2. Önerilen Model İçin Aykırı Verilerin Belirlenmesi ve Yönetilmesi

Su-Mondrian, aykırı veri konseptini uygulayan dağıtık bir model olduğundan dolayı, bu modelde aykırı veri ve normal veri tanımları için yapılan kabullere uygun bir şekilde aykırı verilerin belirlenmesi gerekir. Bu bölümde, yukarıda belirtilen kabullere uygun dağıtık bir aykırı veri belirleme yaklaşımı önerilmiş, geliştirilmiş ve tanıtılmıştır. Ayrıca, belirlenen aykırı verilerin büyük veri mimarisine uygun olarak nasıl yönetilmesi gerektiği de yine bu bölümde verilmiştir.

7.2.1. Büyük veri mimarisinde aykırı verilerin belirlenmesi

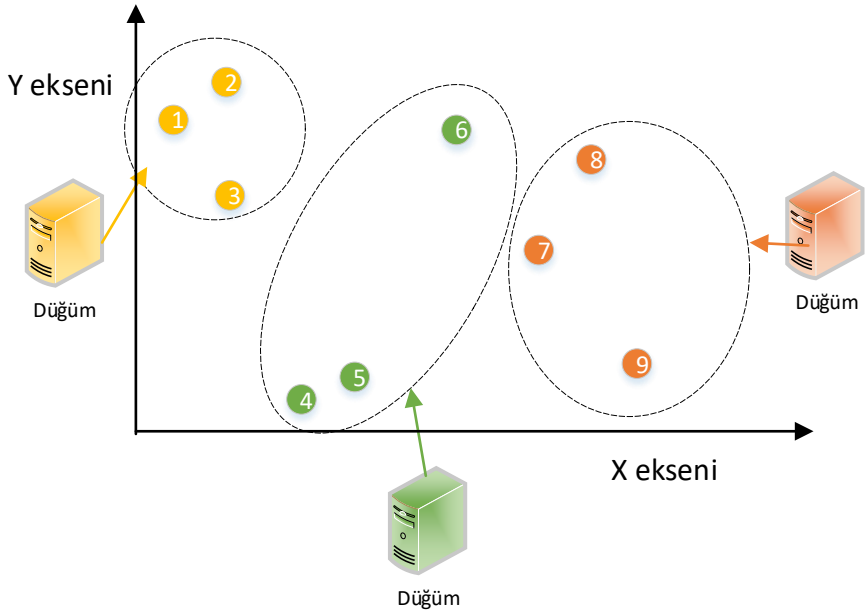
Önerilen Su-Mondrian modelinde, dağıtık halde tutulan veri kümesinin KD-tree yapısı ile parçalanması sonucu elde edilen, eleman sayıları minimum *k* ve maksimum *p* olan ve dağıtık olarak tutulan alt veri gruplarında, her bir grup içerisinde birbirine en yakın *k* adet veri normal veri, arta kalan veriler ise aykırı veri olarak kabul edilmektedir.

Önerilen modelde, dağıtık KD-tree yapısı ile oluşturulan her bir alt grup içerisinde birbirine yakın *k* adet veriyi tespit etmek için Bölüm 4'de sunulan hibrit yaklaşımın dağıtık bir versiyonu geliştirilmiştir. Geliştirilen bu yaklaşım iki aşamadan oluşur. İlk aşamada dağıtık LOF algoritması kullanılarak her bir alt grup içerisinde maksimum yoğunluğa sahip veriler belirlenerek referans noktası olarak kabul edilir ve ikinci aşamada ise bu referans noktalarına en yakın *k-1* adet veri tespit edilir. Bu şekilde birbirine yakın *k* adet veri belirlenerek bu veriler normal veri kümesine, arta kalan diğer verileri ise aykırı veri kümesine atanır.

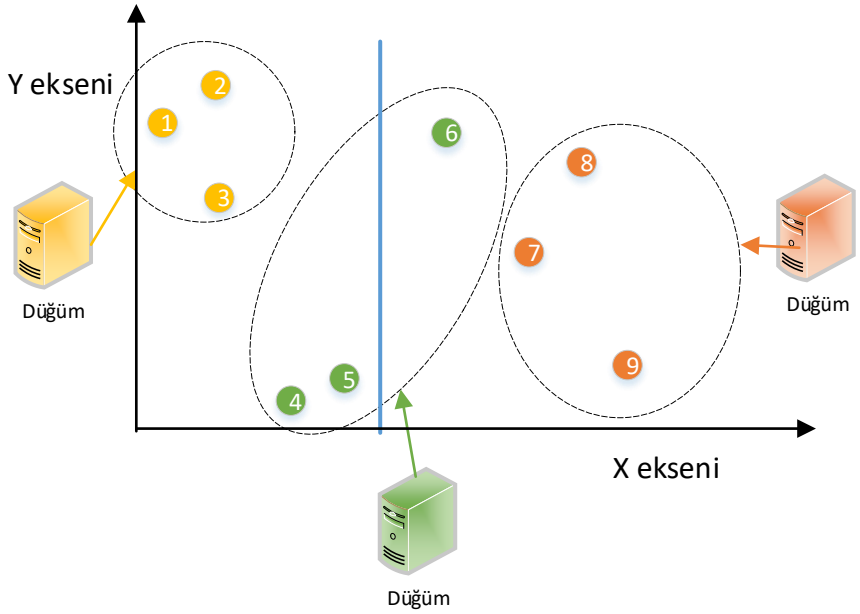
Su-Mondrian modelinde aykırı veri belirleme için örnek bir durum aşağıda sunulmuştur. İki boyutlu düzlemde temsil edilen örnek bir dağıtık veri kümesi Şekil 7.1'de gösterilmektedir. Bu veri kümesine k=3 için dağıtık KD-tree yapısının uygulanmasıyla veri uzayı parçalanmış ve iki alt veri grubu elde edilmiştir. Parçalanma sonrası elde edilen veri ayrımı Şekil 7.2'de gösterilmiş olup gerçekleştirilen bu parçalama mavi renkli çizgi ile temsil edilmiştir.

Şekil 7.2'de elde edilen her bir alt veri grubuna dağıtık LOF algoritmasının uygulanmasıyla elde edilen LOF skorlarına dayanarak, minimum LOF skoruna yani maksimum yoğunluğa sahip veri noktaları referans noktaları olarak kabul edilmiş ve bu noktalar Şekil 7.3'de 3 ve 7 numara ile gösterilmiştir.

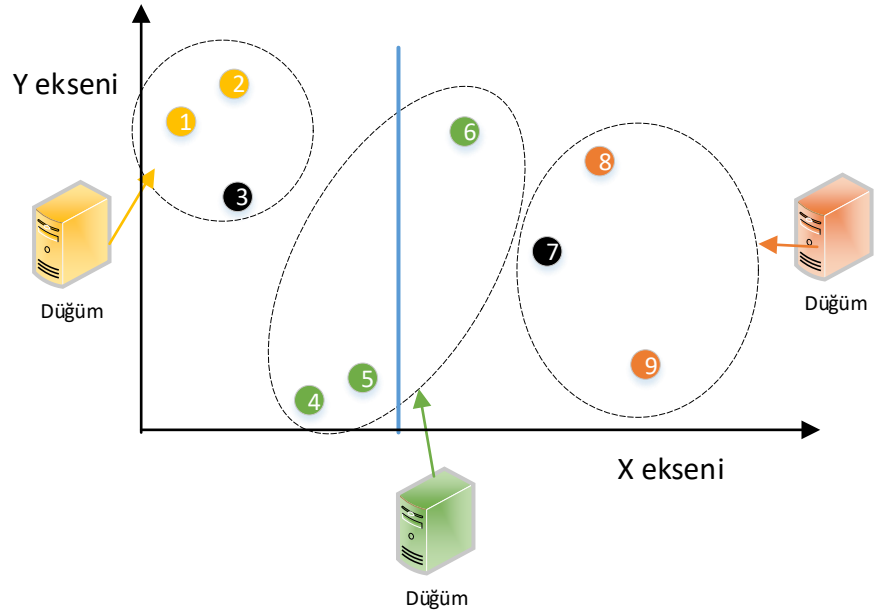
Şekil 7.4'de referans noktalarına en yakın k-1 adet veri noktası bulunarak birbirine en yakın verileri içeren gruplar oluşturulmaktadır. 3 numaralı referans noktasına en yakın veriler 1 ve 2 iken, 7 numaralı referans noktasına en yakın veriler 6 ve 8 olmaktadır. Bu veri grupları birleşerek normal veri kümesini, arta kalan 4, 5 ve 9 numaralı diğer veriler ise aykırı veri kümesini oluşturmaktadır. Şekil 7.5'de, belirlenen normal veriler ve aykırı veriler gösterilmiştir.



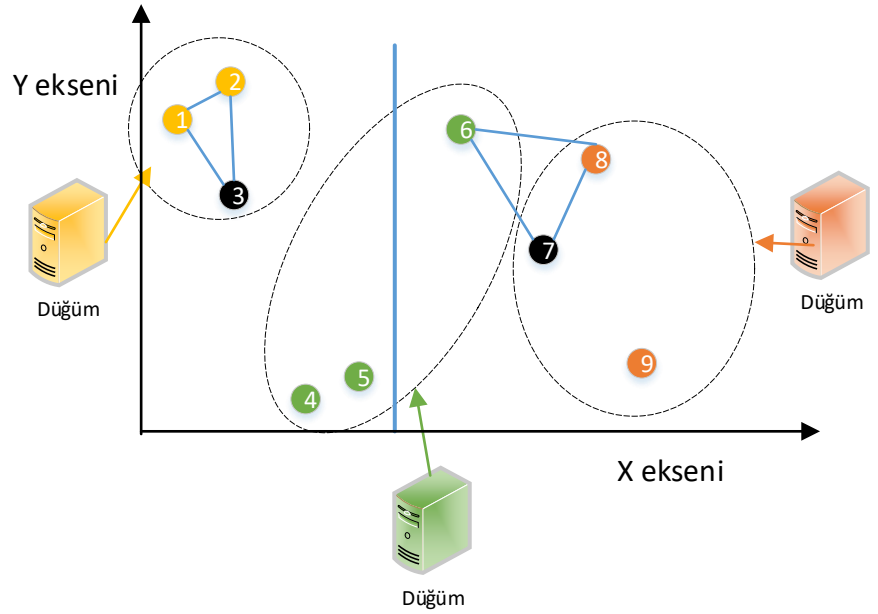
Şekil 7.1. Örnek dağıtık veri kümesi



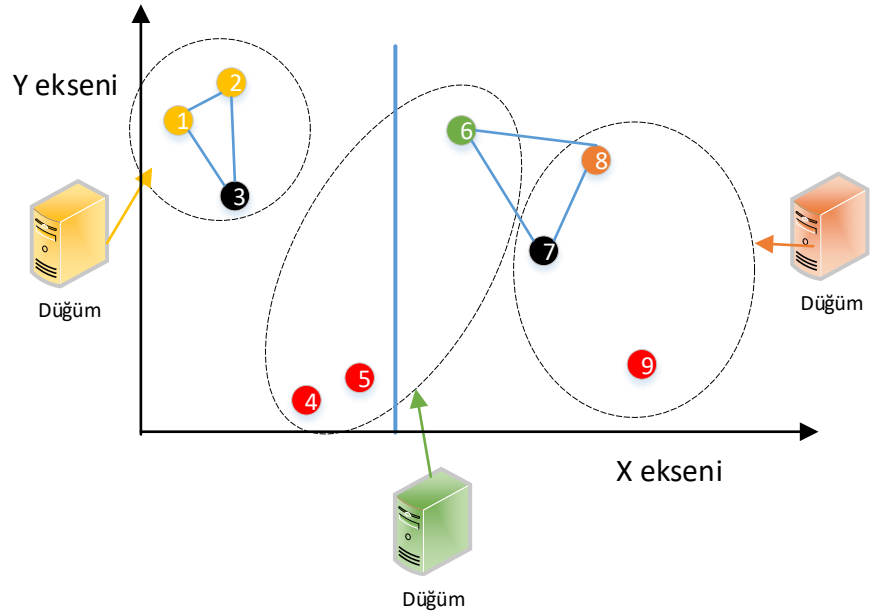
Şekil 7.2. *k*=3 için KD-tree yapısının dağıtık veri uzayını parçalaması



Şekil 7.3. Alt veri gruplarında referans noktalarının belirlenmesi



Şekil 7.4. *k*=3 için referans noktalarına en yakın *k-1* komşulukların bulunması



Şekil 7.5. *k*=3 için aykırı verilerin belirlenmesi

7.2.2. Büyük veri mimarisinde aykırı veri yönetimi

Su-Mondrian modelinde aykırı veri yönetimi, dağıtık aykırı veri kümesinin yinelemeli olarak yeni alt kümelere bölünmesi ve herhangi bir alt kümenin veri kümesinden çıkarılmadan tamamının kullanılmasıyla gerçekleştirilir. Aykırı veri kümesinin yinelemeli olarak bölünmesi, bu verilerin kendi içerisinde daha iyi ayrıştırılmasını sağlayarak veri faydasına olumlu bir katkı sağlar. Buradaki aykırı veri yönetim yaklaşımı, Bölüm 4'de belirtilen yaklaşımla benzer olup, tek farkı dağıtık veri kümeleri üzerinde uygulanmasıdır.

7.3 Önerilen Anonimleştirme Modeli: Su-Mondrian

Bu bölümde, klasik SMondrian modeline aykırı veri konsepti uygulayarak toplam veri faydasını arttırmayı amaçlayan Su-Mondrian modeli tanıtılmış, algoritması, akış şeması ve blok diyagramı sunulmuştur.

Şekil 7.6'da matematiksel ifadesi verilen SMondrian modelinin algoritması şu şekilde çalışır. X, D boyutlu dağıtık bir veri kümesi olsun. İlk aşamada X'e dağıtık KD-tree yapısının uygulanması ile $\{P_1, P_2, \dots, P_T\} \in X$ dağıtık veri parçaları elde edilir. İkinci aşamada ise $\{P_1, P_2, \dots, P_T\} \in X$ içerisindeki her bir P_i için genelleştirme işlemini uygulanarak dağıtık

anonim veri kümesi oluşturulur.

Girdi: dağıtık veri kümesi $X_{RDD} = \{x_1, x_2, \dots, x_N\}$; $x_N \in R^D$, k parametresi
Başlangıç değer ataması: $veri_{RDD} = X$
1) Dağıtık KD-tree uygula:
Yinele:
- Boyut seç; $boyut = \{d \in D,\ d = max_{aralık}(veri_{RDD}, D)\}$
- Frekans hesapla; $f_{boyut} = \left\{\{f_1, f_2, \dots, f_M\} = calc_f(veri_{RDD}, boyut)\right\}$
- Medyan bul; $\mu = \{m \in veri_{boyut}, m = med(f_{boyut})\}$
- Veriyi böl; $lhs_{RDD} = \{l \in veri_{RDD},\ l_{boyut} \leq \mu\}$
$rhs_{RDD} = \{l \in veri_{RDD},\ l_{boyut} > \mu\}$
- Durdurma kriteri sağlanana kadar; $veri_{RDD} = lhs_{RDD}$ ve $veri_{RDD} = lhs_{RDD}$, için tekrarla

Dönder:
- $\{P_{RDD\_1}, P_{RDD\_2}, \dots, P_{RDD\_T}\} \in X$ parçalarını dönder

2) Genelleştir: $\{P_{RDD\_1}, P_{RDD\_2}, \dots, P_{RDD\_T}\} \in X$ içerisindeki her bir P_{RDD_i} için
Yinele:
- Genelleştir; $P^G_{RDD\_i} = gen(P_{RDD\_i})$

Dönder:
- $\left\{P^G_{RDD\_1}, P^G_{RDD\_2}, \dots, P^G_{RDD\_T},\right\} \in X^G_{RDD}$ anonimleştirilmiş veri parçalarını dönder

Çıktı: X^G_{RDD} anonim dağıtık veri kümesi

Şekil 7.6. SMondrian modelinin matematiksel olarak gösterimi

Şekil 7.7'de matematiksel ifadesi verilen Su-Mondrian modeli ise şu şekilde çalışır. X, D boyutlu dağıtık bir veri kümesi olsun. X'e dağıtık KD-tree yapısının uygulanması ile $\{P_1, P_2, \dots, P_T\} \in X$ dağıtık veri parçaları elde edilir. $\{P_1, P_2, \dots, P_T\} \in X$ içerisindeki her bir P_i için, aykırı veriler ve normal veriler belirlenir. Aykırı veriler $AykırıKüme$'de tutulurken, normal veri kümesi $NormalKüme$'de tutulur. Bu işlem belirli bir durdurma kriteri sağlanana kadar devam eder. Durdurma kriteri sağlandığından sonra ise $NormalKüme$ içerisinde bulunan tüm alt gruplar genelleştirilerek anonim dağıtık veri kümesi elde edilir.

Girdi: dağıtık veri kümesi $X_{RDD} = \{x_1, x_2, \dots, x_N\}$; $x_N \in R^D$, k parametresi
Başlangıç değer ataması: $veri_{RDD} = X$
1) Dağıtık KD-tree uygula:
Yinele:
- Boyut seç; $boyut = \{d \in D,\ d = max_{aralık}(veri_{RDD}, D)\}$
- Frekans hesapla; $f_{boyut} = \left\{\{f_1, f_2, \dots, f_M\} = calc_f(veri_{RDD}, boyut)\right\}$
- Medyanı bul; $\mu = \{m \in veri_{boyut}, m = med(f_{boyut})\}$
- Veriyi böl; $lhs_{RDD} = \{l \in veri_{RDD},\ l_{boyut} \leq \mu\}$
  $rhs_{RDD} = \{l \in veri_{RDD},\ l_{boyut} > \mu\}$
- Durdurma kriteri sağlanana kadar; $veri_{RDD} = lhs_{RDD}$ ve $veri_{RDD} = lhs_{RDD}$, için tekrarla

Dönder:
- $\{P_{RDD\_1}, P_{RDD\_2}, \dots, P_{RDD\_T}\} \in X$ parçalarını dönder

2) Dağıtık aykırı verileri belirle: $\{P_{RDD\_1}, P_{RDD\_2}, \dots, P_{RDD\_T}\} \in X$ için
Yinele:
- Skorla; $skor = \left\{\{s_1, s_2, \dots, s_R\} = LOF(P_{RDD\_i})\right\}$
- Referans noktası belirle; $ref_i = \{g = P_{RDD\_i}\{min(skor)\}\}$
- ref_i'nin $k-1$ en yakın komşuluğunu bul; $NN_i^{k-1} = \{x: (k-1) = argmin\|x - ref_i\|\}$
- Normal veriyi belirle; $normal_{RDD\_i} = ref_i \cup NN_i^{k-1}$
- Aykırı veriyi belirle; $aykırı_{RDD\_i} = P_{RDD\_i} - normal_{RDD\_i}$
- $normal_{RDD\_i}$'i $NormalKüme$'ye ekle
- $aykırı_{RDD\_i}$'yı $AykırıKüme$'ye ekle

Değerlendir:
- Eğer $AykırıKüme = \emptyset$ ise Adım 3'e git
  değilse $veri_{RDD} = AykırıKüme$ ve Adım 1'e git

3) Genelleştir: $\{P_{RDD\_1}, P_{RDD\_2}, \dots, P_{RDD\_T}\} \in FaydaKüme$ içerisindeki her bir P_{RDD_i} için
Yinele:
- Genelleştir; $P_{RDD\_i}^G = gen(P_{RDD\_i})$

Dönder:
- $\left\{P_{RDD\_1}^G, P_{RDD\_2}^G, \dots, P_{RDD\_T}^G,\right\} \in X_{RDD}^G$ anonimleştirilmiş veri parçalarını dönder

Çıktı: X_{RDD}^G anonim dağıtık veri kümesi

Şekil 7.7. Su-Mondrian modelinin matematiksel olarak gösterimi

Su-Mondrian modeline ait algoritma Şekil 7.8'de gösterilmektedir. Algoritmada gerçekleştirilen işlemler hazırlık, veri parçalama, aykırı veri belirleme, değerlendirme, genelleştirme ve yayınlama olmak üzere 6 katmanlı bir yapıdan oluşmaktadır. Hazırlık

katmanında anonimleştirilecek dağıtık veri kümesi işlenmeye hazır hale getirilirken, veri parçalama katmanında dağıtık veri kümesi KD-tree yapısı ile dağıtık alt gruplara ayrılır. Aykırı veri belirleme katmanında her bir alt gruptaki dağıtık aykırı ve normal veriler hibrit bir yaklaşımla tespit edilerek ilgili kümelere atanır. Değerlendirme katmanında durdurma kriterinin sağlanıp sağlanmadığı kontrol edilir. Genelleştirme katmanında, normal veri kümesi içerisindeki her alt grup kendi içerisinde genelleştirilerek anonim hale getirilir. Yayınlanma katmanında ise dağıtık anonim veriler birleştirilerek yayınlanır.

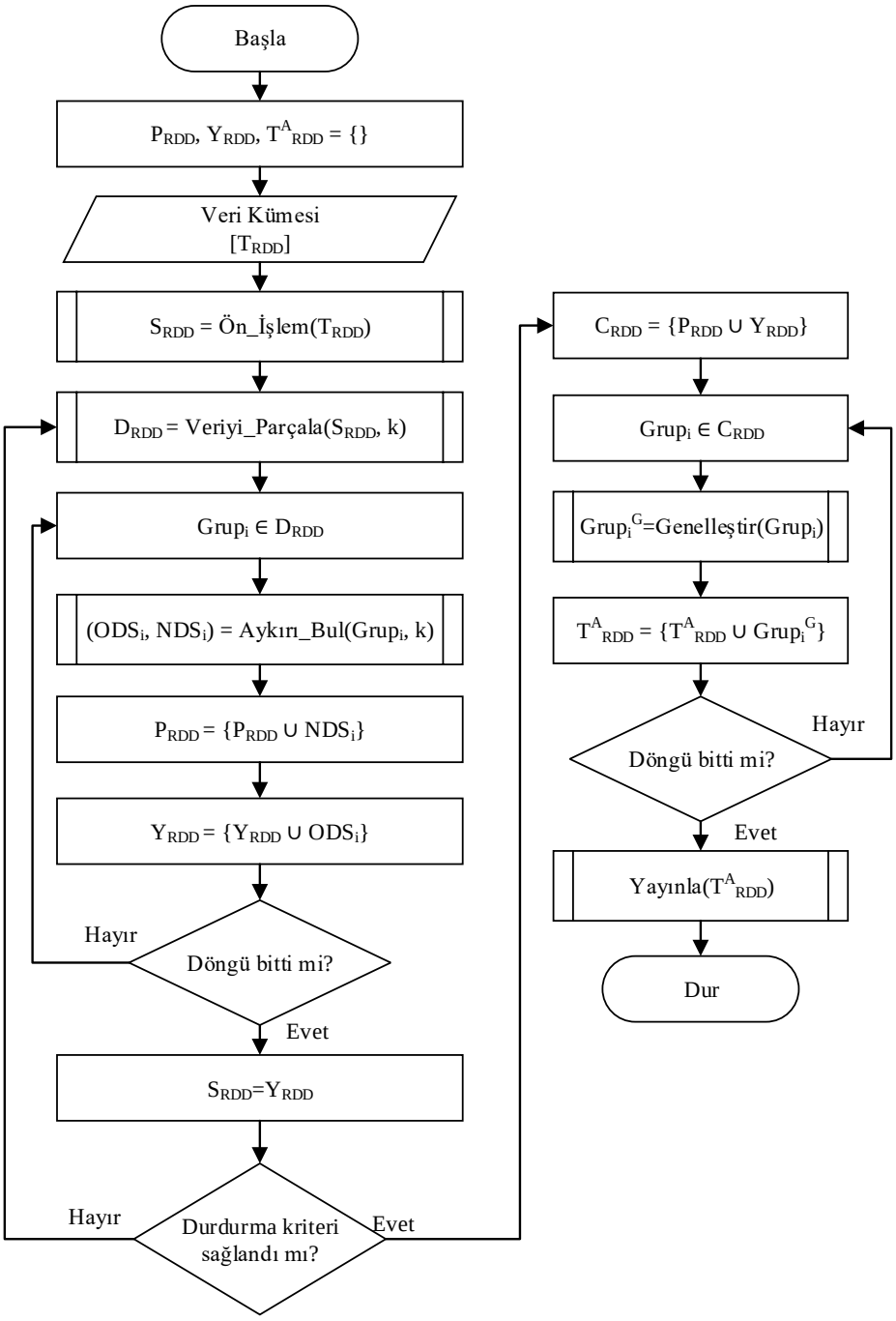
1: $P_{RDD}, Y_{RDD}, T^A_{RDD} \leftarrow \emptyset$
2: $S_{RDD} \leftarrow Hazirlik(Veri_{RDD})$
3: **function** $Su\_Mondrian(S_{RDD})$
4: **while** $durdurma\_kriteri == false$ **do**
5: $D_{RDD} \leftarrow Veriyi\_Parcala(S_{RDD}, k)$
6: **for all** $grup \in D_{RDD}$ **do**
7: $(ODS_i, NDS_i) \leftarrow Aykiri\_Verileri\_Bul(grup, k)$
8: $P_{RDD} \leftarrow P_{RDD} \cup NDS_i$
9: $Y_{RDD} \leftarrow Y_{RDD} \cup ODS_i$
10: **end for**
11: $S_{RDD} \leftarrow Y$
12: **end while**
13: $C_{RDD} \leftarrow P_{RDD} \cup Y_{RDD}$
14: **for all** $grup \in C_{RDD}$ **do**
15: $grup^G \leftarrow Genellestir(grup)$
16: $T^A_{RDD} \leftarrow T^A_{RDD} \cup grup^G$
17: **end for**
18: $Yayinla(T^A_{RDD})$
19: **end function**

Şekil 7.8. Su-Mondrian algoritması

Su-Mondrian modeline ait akış şeması Şekil 7.9'da gösterilmekte olup, akıştaki her adım sırasıyla aşağıda açıklanmıştır;

1. HDFS'de tutulan T_{RDD} dağıtık veri kümesi modele girdi olarak verilir,
2. Ön işlem aşamasında özniteliklerin sınıfları belirlenir, normalizasyon işlemi yapılır, durdurma kriteri belirlenir ve S_{RDD} veri kümesi elde edilir,
3. S_{RDD} veri kümesi KD-tree yapısı ile alt gruplara bölünerek D_{RDD} veri kümesi elde edilir,
4. D_{RDD} kümesi içerisindeki her bir grup için;
    a. Aykırı veriler ve normal veriler belirlenerek NDS_i ve ODS_i kümeleri oluşturulur,
    b. NDS_i, P_{RDD} kümesine eklenir,
    c. ODS_i, Y_{RDD} kümesine eklenir,

5. Eğer durdurma kriteri sağlanmadıysa irdelenecek yeni veri kümesi $S_{RDD} = Y_{RDD}$ olur ve Adım 3'e gidilir,
6. Eğer durdurma kriteri sağlandıysa P_{RDD}, Y_{RDD} ile birleştirilerek C_{RDD} kümesi elde edilir,
7. C_{RDD} kümesindeki her bir grup için;
    a. Gruplar genelleştirilir,
    b. Genelleştirilen grup T^{A}_{RDD} kümesine eklenir
8. T^{A}_{RDD} anonim veri kümesi yayınlanır.



Şekil 7.9. Su-Mondrian modelinin akış diyagramı

7.3.1. Büyük veri temelli hazırlık katmanı

Özniteliklerin sınıflandırılması, dağıtık verinin normalizasyonu ve durdurma kriterinin belirlenmesi bu katmanda gerçekleştirilir. Bu katmana ait algoritma Şekil 7.10'da gösterilmiş olup, yapılan işlemler aşağıda açıklanmıştır;

- Özniteliklerin sınıflandırılması: veri içerisindeki özniteliklerin; tam-tanımlayıcı, yarı-tanımlayıcı, hassas ve hassas olmayan olmak üzere sınıflandırıldığı aşamadır. *k*-Anonimlik modeli yarı-tanımlayıcılar üzerinden işlem yaptığı için bu aşamada dağıtık veri kümesindeki yarı-tanımlayıcı öznitelikler seçilerek işlenmeye hazır yeni bir dağıtık veri kümesi oluşturulur.
- Dağıtık veri normalizasyonu: her biri farklı değer aralıklarına sahip öznitelikler normalizasyon işlemine tabi tutulur. Dağıtık halde tutulan veri kümesi min-max yöntemi ile normalize edilebilir.
- Durdurma kriterinin belirlenmesi: iterasyon sayısı, çeşitli metrik değerleri veya farklı parametreler durdurma kriteri olarak belirlenebilir.

```
1: function Hazirlik(Veri_RDD)
2:     T_RDD ← Oznitelik_Siniflandir(Veri_RDD)
3:     S_RDD ← Normalize_Et(T_RDD)
4:     Durdurma_Kriteri_Belirle()
5:     return S
6: end function
7: function Oznitelik_Siniflandir(Veri_RDD)
8:     A ← Veri_RDD.toDF.select("yarı − tanımlayıcılar").rdd
9:     return A
10: end function
11: function Normalize_Et(T_RDD)
12:     for all boyut ∈ D do
13:         min ← T_RDD.toDF.select(min(boyut)).first.getDouble(0)
14:         max ← T_RDD.toDF.select(max(boyut)).first.getDouble(0)
15:         B ← T_RDD.toDF.map{row => ((row.getDouble(0) − min)/(max − min), ...}
16:         return B
17:     end for
18: end function
```

Şekil 7.10. Hazırlık katmanı algoritması

7.3.2. Büyük veri temelli veri parçalama katmanı

Bu katmanda, SMondrian algoritmasında kullanılan dağıtık KD-tree ağacı ile veri kümesi her birinde minimum *k* maksimum *p* adet veri olacak şekilde alt gruplara parçalanır. Bu şekilde veri uzayı alt uzaylara bölünerek birbirine olabildiği kadar yakın eleman grupları oluşturulur. Şekil 7.11'de veri parçalama katmanı için sunulan algoritma gösterilmekte olup, bu algoritmadaki ilgili fonksiyonlar için detaylar Şekil 7.12'de gösterilmiştir.

1: **function** $Veriyi\_Parcala(S_{RDD}, k)$
2: $D_{RDD} \leftarrow KD\_Tree(S_{RDD}, k)$
3: **return** (D_{RDD})
4: **end function**
5: **function** $KD\_Tree(S_{RDD}, k)$
6: **while** $bolunme == true$ **do**
7: $boyut \leftarrow Boyut\_Sec(S_{RDD})$
8: $frekans \leftarrow Frekans\_Hesapla(S_{RDD}, boyut)$
9: $medyan \leftarrow Medyan\_Bul(frekans)$
10: $(SolParca_{RDD}, SagParca_{RDD}) \leftarrow Veriyi\_Bol(S_{RDD}, medyan)$
11: $KD\_Tree(SolParca_{RDD}, k)$
12: $KD\_Tree(SagParca_{RDD}, k)$
13: **end while**
14: **return** $parcalar_{RDD}$
15: **end function**

Şekil 7.11. Veri parçalama katmanı algoritması

```
function Boyut_Sec(S_RDD)
    df ← S_RDD.toDF
    max = 0
    dim = len(df.columns)
    while i < dimdim do
        gen = Boyut_Genislik(df, dim)
        if max < gen then
            max = gen
            max_dim = dim
            return max_dim
        end if
    end while
end function
function Boyut_Genislik(df, dim)
    min = df.select(min(boyut)).first.getDouble(0)
    max = df.select(max(boyut)).first.getDouble(0)
    return max − min
end function
function Frekans_Hesapla(S_RDD, dim)
    df ← S_RDD.toDF
    frekanslar = df.select(dim).groupBu(dim).count.sort(dim)
    frekans_kumesi = frekanslar.rdd.map(Row(x, frekans) => (x, frekans)).collect
    return frekans_kumesi
end function
function Medyan_Bul(frekans_kumesi)
    x = frekans_kumesi.map(_._2).sum
    medyan = x/2
    return medyan_indis
end function
function Veriyi_Bol(S_RDD, medyan)
    df = S_RDD.toDF
    boyut = df.select(dim).sort.rdd.collect
    SolParca_RDD ← df.join(broadcast(boyut(0, medyan))).rdd
    SagParca_RDD ← df.join(broadcast(boyut(medyan, son))).rdd
    return SolParca_RDD, SagParca_RDD
end function
```

Şekil 7.12. Veri parçalama katmanı alt algoritmaları

7.3.3. Büyük veri temelli aykırı veri belirleme katmanı

Aykırı verilerin belirlenerek dağıtık veri kümesinin, normal ve aykırı olmak üzere iki alt kümeye bölündüğü katmandır. Şekil 7.13'de algoritması sunulan bu katman için, veri parçalama katmanı sonrası elde edilen her bir alt grupta sırasıyla şu işlemler yapılır;

1. alt gruptaki verilere dağıtık LOF algoritması uygulanarak LOF skorları elde edilir,
2. minimum LOF skoruna sahip veri referans noktası olarak belirlenir,
3. referans noktasına en yakın *k-1* adet veri tespit edilir,
4. referans noktası da dahil *k* adet veri NDS_{RDD} normal veri kümesine, arta kalan veriler ise ODS_{RDD} aykırı veri kümesine atanır,

5. tüm gruplar için işlemler yapıldıktan sonra elde edilen NDS_{RDD} ve ODS_{RDD} veri kümeleri geri gönderilir.

```
1: function Aykiri_Verileri_Bul(D_RDD_i, k)
2:     for all grup_RDD ∈ D_RDD do
3:         t ← LOF(grup_RDD)
4:         referans ← grup_RDD(min(t))
5:         N_RDD ← kNN_Bul(grup_RDD, referans)
6:         NDS_RDD ← NDS_RDD ∪ N_RDD
7:         ODS_RDD ← grup_RDD − N_RDD
8:     end for
9:     return (NDS_RDD, ODS_RDD)
10: end function
```

Şekil 7.13. Aykırı veri tespit katmanı algoritması

Şekil 7.13'de verilen algoritma için; yoğunluk skorlama fonksiyonu olarak kullanılan LOF algoritmasının dağıtık versiyonu Şekil 7.14'de, referans noktasına en yakın komşulukların bulunması için de geliştirilen algoritma da Şekil 7.15'de verilmiştir.

```
1: function LOF(Veri_RDD)
2:     komsuluk_RDD = Range(0, numPartitions).map{outIdx =>
3:     outData = Veri_RDD.mapPartitionWithIndex{(inIdx, iter) =>
4:     {if(inIdx == outIdx) iter}}.collect
5:     Veri_RDD.mapPartitions{inPart =>
6:     sc.broadcast(outPart).value.foreach{(idx, vec) =>
7:     komsuluk = kKomsulukHesapla(inPart, vec, minpts)
8:     list.append(idx, komsuluk)}} list.iterator}
9:     reduceByKey(komsuluklariBirlestir)}reduce(_.union(_))
10:    t = komsuluk_RDD.flatMap{(outIdx, komsuluk) =>
11:    komsuluk.map{(inIdx, uzaklik) => (inIdx, (outIdx, uzaklik))}.groupByKey()
12:    a = t.cogroup(komsuluk_RDD).flatMap{(outIdx, k, v) =>
13:    k_Uzaklik = v.head.last._2
14:    k.head.filter(_._1 != outIdx).map{(inIdx, uzaklik) =>
15:    (inIdx, (outIdx, max(uzaklik, k_Uzaklik)))}groupByKey()
16:    b = a.map{(idx, m) => num = m.size sum = m.map(_._2).sum (idx, num/sum)}}
17:    c = b.cogroup(t).flatMap{(outIdx, k, v) => lrd = k.head
18:    v.head.map{(inIdx, uzaklik) => (inIdx, (outIdx, lrd))}.groupByKey
19:    d = c.map{(idx, r) =>
20:    lrd = r.find(_._1 = idx).get._2
21:    sum = r.filter(_._1 != idx).map(_._2).sum
22:    (idx, sum/lrd/(r.size − 1))}
23:    return a
24: end function
```

Şekil 7.14. Dağıtık skorlama fonksiyonu

```
1: function kNN_Bul(Veri_RDD, referans)
2:     df = Veri_RDD
3:     data = Veri_RDD.collect()
4:     data.foreach{(idx, vec) => list.append((idx, sqdist(referans, vec)))})
5:     normal_idx = list.sortBy(x = x._2).get(x._1)[0, k]
6:     normal = df.filter(idx == normal_idx).rdd
7:     return normal
8: end function
```

Şekil 7.15. Dağıtık komşuluk bulma fonksiyonu

7.3.4. Büyük veri temelli değerlendirme katmanı

Bir durdurma kriterine bağlı olarak dağıtık olarak çalışan aykırı veri belirleme, anonimleştirme ve yayınlama katmanlarının çalışmasını kontrol eden katmandır. Belirlenen durdurma kriteri sağlanmazsa veri parçalama ve aykırı veri belirleme katmanlarına gidilirken bu kriterin sağlanması durumunda genelleştirme ve yayınlama katmanlarına gidilir. Şekil 7.16'da bu katmana ait algoritma gösterilmektedir. İterasyon sayısı ve farklı metrikler durdurma kriteri olarak kullanılabileceği gibi, büyük veri temelli farklı yapıların da kullanılması mümkündür.

```
1: while durdurma_kriteri == false do
2:     devam
3:     ....
4: end while
```

Şekil 7.16. Değerlendirme katmanı algoritması

7.3.5. Büyük veri temelli genelleştirme katmanı

Genelleştirme operatöründen faydalanılarak dağıtık veri kümesinin anonimleştirildiği katmandır. Aykırı veri belirleme katmanı sonrası elde edilen normal veri grupları bu aşamada anonim hale getirilir. Her bir alt veri grubunun kendi içerisinde genelleştirildiği bu katmana ait algoritma Şekil 7.17'de sunulmuştur.

```
1: function Genellestir(grup_RDD)
2:     data = grup_RDD.toDF
3:     min = broadcast(data.agg(min("yarı − tanımlayıcılar"))).head()
4:     max = broadcast(data.agg(max("yarı − tanımlayıcılar"))).head()
5:     gen = grup_RDD.map{r => while(i < len(boyut))
6:     Array(i) = Row(min(i), max(i))}
7:     Row(field.toSeq)}
8:     return gen_RDD
9: end function
```

Şekil 7.17. Anonimleştirme katmanı algoritması

7.3.6. Büyük veri temelli yayınlama katmanı

Faydası arttırılmış olan anonim büyük verinin yayınlandığı son katmandır. Son aşamada elde edilen T^{A}_{RDD} anonim dağıtık veri kümesi bu katmanda yayınlanır.

7.4. Deneysel Çalışmalar

Bu bölümde, önerilen Su-Mondrian modelini test etmek ve doğrulamak için gerçekleştirilen deneysel çalışmalar açıklanmış ve elde edilen sonuçlar sunulmuştur.

Önerilen model, Databricks platformu üzerinde Spark anaçatısı kullanılarak geliştirilmiştir. Literatürdeki geleneksel veride yapılan mahremiyet çalışmalarında olduğu gibi büyük veri anonimleştirme çalışmalarında da defakto-standart olarak kullanılan Adult veri kümesi bu deneyde kullanılmıştır. Eksik veri yönetimi kapsamında Adult veri kümesi içerisinde bulunan 18680 adet eksik veri kümesinden çıkarılmıştır. Arta kalan veri kümesi ise 10 defa satır seviyesinde alt alta eklenerek çoğaltılmış ve testler 301620 adet veri üzerinde yapılmıştır. Yapılan deneylerde, yaş, kilo, gelir bilgisi, gider bilgisi ve haftalık çalışma saati öznitelikleri yarı-tanımlayıcı olarak seçilmiştir. Elde edilen sonuçlar tablolar halinde sunulmuş, son olarak da bu sonuçların daha kolay yorumlanması için grafiklerle gösterilmiştir.

Deneylerde DM, GCP ve AECS metrikleri ve eşlenik sınıf sayısını ölçmek için ES fonksiyonu kullanılmış, bu metriklerle veri faydası ölçülerek ilgili karar kuralları için değerlendirmeler yapılmıştır.

Yapılan deneyde, durdurma kriteri iterasyon sayısı olarak belirlenmiş ve bu sayı 5'e sabitlenmiştir. Tüm deneylerde, önerilen modelin SMondrian modeline göre veri faydasında önemli artış gerçekleştirdiği görülmektedir. Tüm çizelgelerde, SMondrian kısaltılarak SM ile Su-Mondrian ise kısaltılarak Su-M ile temsil edilmiştir.

Çizelge 7.1'de gösterilen sonuçlar incelendiğinde; *k*=100 için DM, GCP ve AECS metriklerine göre SMondrian sırasıyla 40753000, 197408,06 ve 130,23 değerlerini sunarken, Su-Mondrian sırasıyla 30150000, 179049,92 ve 100,44 değerlerini sunmaktadır. İlgili karar kurallarına göre, Su-Mondrian modelinin SMondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 7.1. *k*=100 için Su-Mondrian ve SMondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	301620	0	2316	2316	40753000	40753000	197408,06	197408,06	130,23	SM
1	231600	70020	2316	2316	23160000	23160000	123933,45	123933,45	100	Su-M
2	54600	15420	546	2862	5460000	28620000	39398,06	163331,51	100	
3	11500	3920	115	2977	1150000	29770000	10931,95	174263,46	100	
4	3000	920	30	3007	300000	30070000	3492,81	177756,27	100	
5	920	0	9	3016	80000	30150000	1293,65	179049,92	102,22	

Çizelge 7.2'de gösterilen sonuçlar incelendiğinde; *k*=200 için DM, GCP ve AECS metriklerine göre SMondrian sırasıyla 81431400, 239880,89 ve 260,91 değerlerini sunarken, Su-Mondrian sırasıyla 60280000, 204020,76 ve 200,08 değerlerini sunmaktadır. İlgili karar kurallarına göre, Su-Mondrian modelinin SMondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 7.2. *k*=200 için Su-Mondrian ve SMondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	301620	0	1156	1156	81431400	81431400	239880,89	239880,89	260,91	SM
1	231200	70420	1156	1156	46240000	46240000	135559,30	135559,30	200	Su-M
2	55000	15420	275	1431	11000000	57240000	49993,48	185552,78	200	
3	10800	4620	54	1485	2160000	59400000	11040,68	196593,46	200	
4	3600	1020	18	1503	720000	60120000	5661,20	202254,66	200	
5	1020	0	5	1506	160000	60280000	1766,10	204020,76	204	

Çizelge 7.3'de gösterilen sonuçlar incelendiğinde; *k*=300 için DM, GCP ve AECS metriklerine göre SMondrian sırasıyla 135127600, 267864,00 ve 430,27 değerlerini sunarken, Su-Mondrian sırasıyla 90360000, 221533,22 ve 304,80 değerlerini sunmaktadır. İlgili karar kurallarına göre, Su-Mondrian modelinin SMondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 7.3. *k*=300 için Su-Mondrian ve SMondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	301620	0	701	701	135127600	135127600	267864,00	267864,00	430,27	SM
1	210300	91320	701	701	63090000	63090000	117256,78	117256,78	300	Su-M
2	69000	22320	230	931	20700000	83790000	65896,85	183153,63	300	
3	17700	4620	59	990	5310000	89100000	26273,35	209426,98	300	
4	3000	1620	10	1000	900000	90000000	7034,22	216461,20	300	
5	1620	0	5	1005	360000	90360000	5072,02	221533,22	324	

Çizelge 7.4'de gösterilen sonuçlar incelendiğinde; *k*=400 için DM, GCP ve AECS metriklerine göre SMondrian sırasıyla 161831200, 276947,60 ve 521,83 değerlerini sunarken, Su-Mondrian sırasıyla 120480000, 235156,64 ve 401,00 değerlerini sunmaktadır. İlgili karar kurallarına göre, Su-Mondrian modelinin SMondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 7.4. *k*=400 için Su-Mondrian ve SMondrian modellerinin test sonuçları

i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	301620	0	578	578	161831200	161831200	276947,60	276947,60	521,83	SM
1	231200	70420	578	578	92480000	92480000	157965,79	157965,79	400	Su-M
2	51600	18820	129	707	20640000	113120000	49290,27	207256,06	400	
3	13600	5220	34	741	5440000	118560000	17492,95	224749,01	400	
4	3600	1620	9	750	1440000	120000000	6048,07	230797,08	400	
5	1620	0	4	754	480000	120480000	4359,56	235156,64	405	

Çizelge 7.5'de gösterilen sonuçlar incelendiğinde; *k*=500 için DM, GCP ve AECS metriklerine göre SMondrian sırasıyla 220892400, 292666,33 ve 691,78 değerlerini sunarken, Su-Mondrian bu metrikler için sırasıyla 150750000, 249506,83 ve 512,00 değerlerini sunmaktadır. İlgili karar kurallarına göre, Su-Mondrian modelinin SMondrian modeline göre veri faydasını arttırdığı görülmektedir.

Çizelge 7.5. *k*=500 için Su-Mondrian ve SMondrian modellerinin test sonuçları

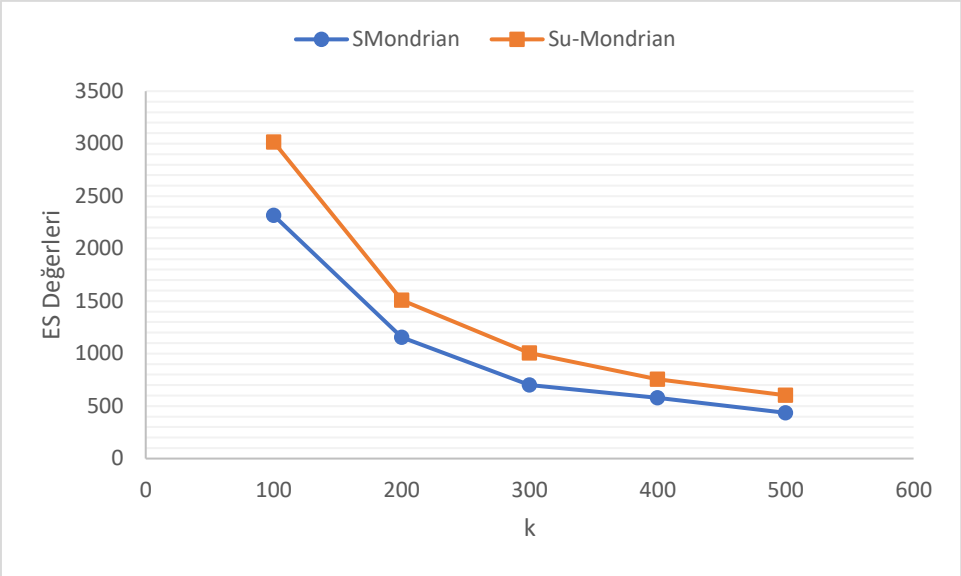
i	|N|	|O|	ES	ES_T	DM	DM_T	GCP	GCP_T	AECS	Model
-	301620	0	436	436	220892400	220892400	292666,33	292666,33	691,78	SM
1	218000	83620	436	436	109000000	109000000	140386,45	140386,45	500	Su-M
2	66500	17120	133	569	33250000	142250000	75272,27	215658,72	500	
3	12000	5120	24	593	6000000	148250000	18914,54	234573,26	500	
4	4000	1120	8	601	2000000	150250000	10652,68	245225,94	500	
5	1120	0	2	603	500000	150750000	4280,89	249506,83	560	

Çizelge 7.6'da Su-Mondrian ile SMondrian modellerinin ES, DM, GCP ve AECS sonuçları gösterilmektedir. Elde edilen sonuçlara ve karar kurallarına göre, önerilen modelin SMondrian modeline göre tüm *k* değerleri için veri faydasını iyileştirdiği görülmektedir.

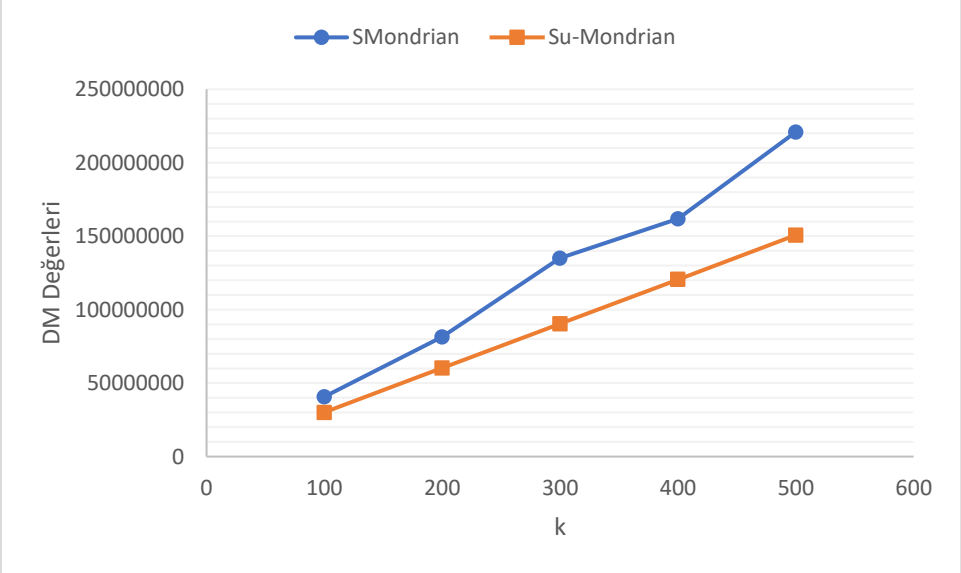
Çizelge 7.6. Su-Mondrian ve SMondrian modellerinin genel test sonuçları

k	ES	DM	GCP	AECS	Model
100	2316	40753000	197408,06	130,23	SM
	3016	30150000	179049,92	100,44	Su-M
200	1156	81431400	239880,89	260,91	SM
	1508	60280000	204020,76	200,80	Su-M
300	701	135127600	267864,00	430,27	SM
	1005	90360000	221533,23	304,8	Su-M
400	578	161831200	276947,60	521,83	SM
	754	120480000	235156,64	401,00	Su-M
500	436	220892400	292666,33	691,78	SM
	603	150750000	249506,83	512,00	Su-M

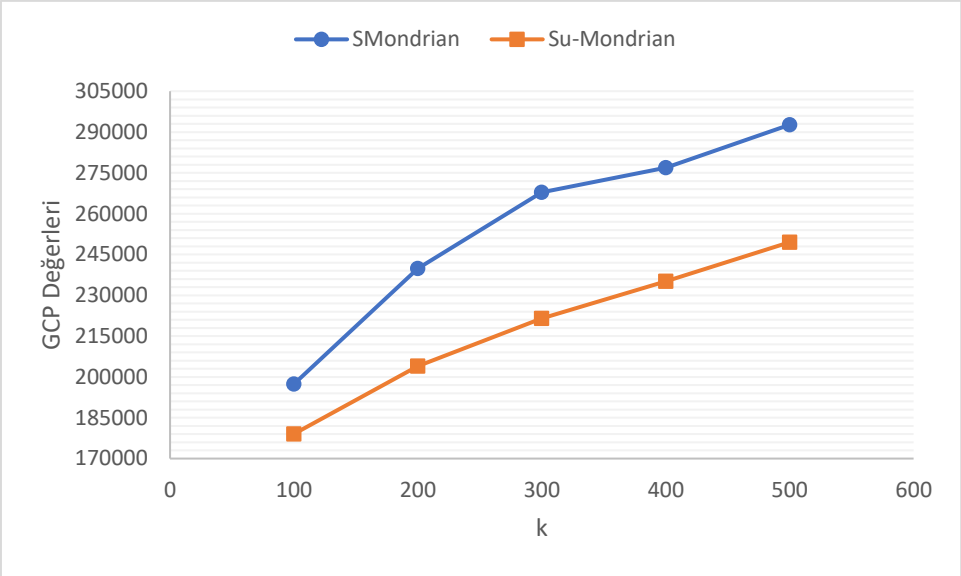
Şekil 7.18, Şekil 7.19, Şekil 7.20 ve Şekil 7.21'de iki model için farklı *k* değerlerine göre ölçülen ES, DM, GCP ve AECS değerleri sırasıyla gösterilmiştir. Sunulan tüm bu şekillerde, Su-Mondrian modelinin SMondrian modeline göre daha yüksek veri faydası sunduğu açıkça görülmektedir.



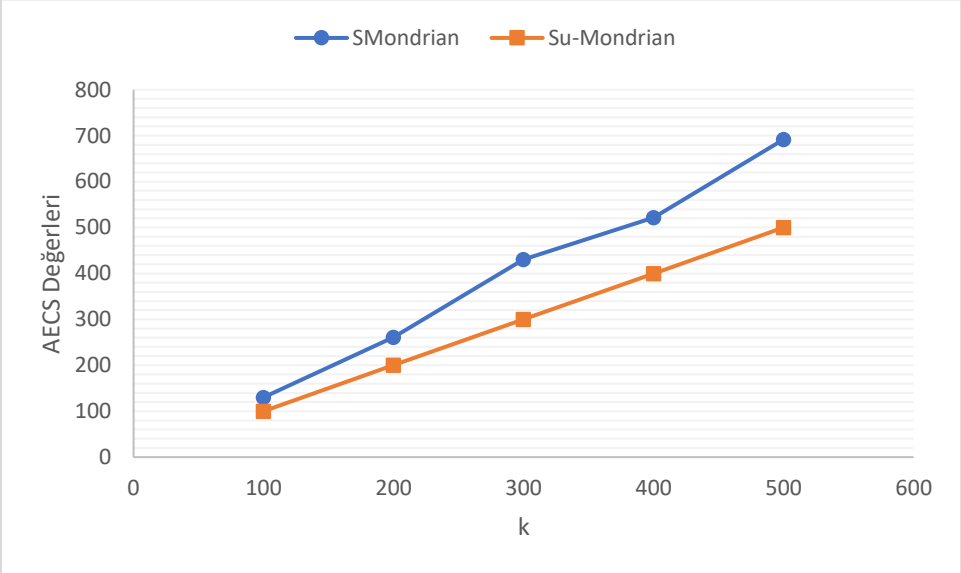
Şekil 7.18. Farklı *k* değerleri için elde edilen ES değerleri



Şekil 7.19. Farklı *k* değerleri için elde edilen DM değerleri



Şekil 7.20. Farklı *k* değerleri için elde edilen GCP değerleri



Şekil 7.21. Farklı *k* değerleri için elde edilen AECS değerleri

Çizelge 7.6'da sunulan sonuçlara göre Su-Mondrian modelinin SMondrian'a göre veri faydasında yaptığı iyileştirmeler oransal olarak Çizelge 7.7'de gösterilmiştir. Bu sonuçlara göre, Su-Mondrian modeli SMondrian modeline göre DM, GCP ve AECS değerlerinde sırasıyla %25,55-%33,12, %9,29-%17,29 ve %22,83-%29,16 aralıklarında daha yüksek veri faydası sunduğu görülmektedir.

Çizelge 7.7. Su-Mondrian modelinin SMondrian'a göre veri faydası iyileştirme oranları

k	DM (%)	GCP (%)	AECS (%)
100	26,01	9,29	22,83
200	25,97	14,94	23,04
300	33,12	17,29	29,16
400	25,55	15,08	23,15
500	31,75	14,74	25,98

7.5. Bölümde Sunulan Katkılar

Bu bölümde sunulan katkılar aşağıda listelenmiştir;

- Aykırı veri yönetimi yaparak tüm aykırı verilerden fayda üreten büyük veri temelli bir anonimleştirme modeli ilk defa bu çalışmada önerilmiş, uygulanmış ve başarı ile test edilmiştir,
- Büyük veri anonimleştirmede, yoğunluk ve yakınlık tabanlı yaklaşımları birleştirilerek aykırı verilerin belirlenmesini sağlayan hibrit bir yaklaşım ilk defa bu çalışmada önerilmiş, uygulanmış ve başarı ile test edilmiştir,
- SMondrian modeli için aykırı veri ve normal veri tanımları ilk defa bu çalışmada yapılmıştır.

8. SONUÇ VE DEĞERLENDİRMELER

Bu tez kapsamında;

- aykırı verileri yöneterek toplam veri faydasını arttırmayı amaçlayan geleneksel mimari temelli iki yeni anonimleştirme modeli (u-Mondrian ve u-Canon),
- veri uzayını parçalamak için VP-tree yapısını kullanan ve Mondrian modelinden daha yüksek veri faydası sunduğu test sonuçları ile ortaya konulan yeni bir anonimleştirme modeli (Canon),
- büyük veri mimarisinde aykırı veri yönetimi yaparak toplam veri faydasını arttıran yeni bir anonimleştirme modeli (Su-Mondrian) ilk defa bu tez çalışması kapsamında önerilmiş, başarılı bir şekilde geliştirilmiş, uygulanmış ve test edilmiştir.

Elde edilen analiz sonuçları, önerilen dört farklı modelin yüksek başarımlı modeller olduğunu, anonimleştirme çözümü olarak kullanılabileceklerini ve maksimum veri faydası elde etmede önemli modeller olduğunu göstermiştir.

Tez kapsamında elde edilen bulgular değerlendirildiğinde aşağıdaki hususlar elde edilmiştir;

1. Anonimleştirme kapsamında aykırı veri konseptine yeni bir bakış açısı getirilmiştir,
2. Önerilen u-Mondrian modeli, Mondrian modeline göre daha yüksek veri faydası sunmaktadır,
3. Önerilen Canon modeli, Mondrian modeline göre daha yüksek veri faydası sunmaktadır,
4. Önerilen u-Canon modeli, Canon modeline göre daha yüksek veri faydası sunmaktadır,
5. Önerilen Su-Mondrian modeli, SMondrian modeline göre daha yüksek veri faydası sunmaktadır,
6. Aykırı veri yönetimi ile veri faydası arasındaki ilişkiyi göstermek adına ilk defa kapsamlı karar kuralları oluşturulmuş ve bu kuralların önerilen modellerde başarı ile kullanılabileceği gösterilmiştir,
7. Anonimleştirmede aykırı ve normal veri tespitinde kullanılmak üzere ilk defa hem yoğunluğu hem de yakınlığı dikkate alan yeni bir hibrit yaklaşım geliştirilmiştir.

Tez kapsamında elde edilen sonuçlar Şekil 8.1, Şekil 8.2, Şekil 8.3 ve Şekil 8.4'de gösterilmekte olup verilen bu şekillerde; Mondrian-R ve Mondrian-S sırasıyla Relaxed ve Strict stratejili Mondrian modellerini, u-Mondrian-R ve u-Mondrian-S ise sırasıyla Relaxed

ve Strict stratejili u-Mondrian modellerini temsil etmektedir.

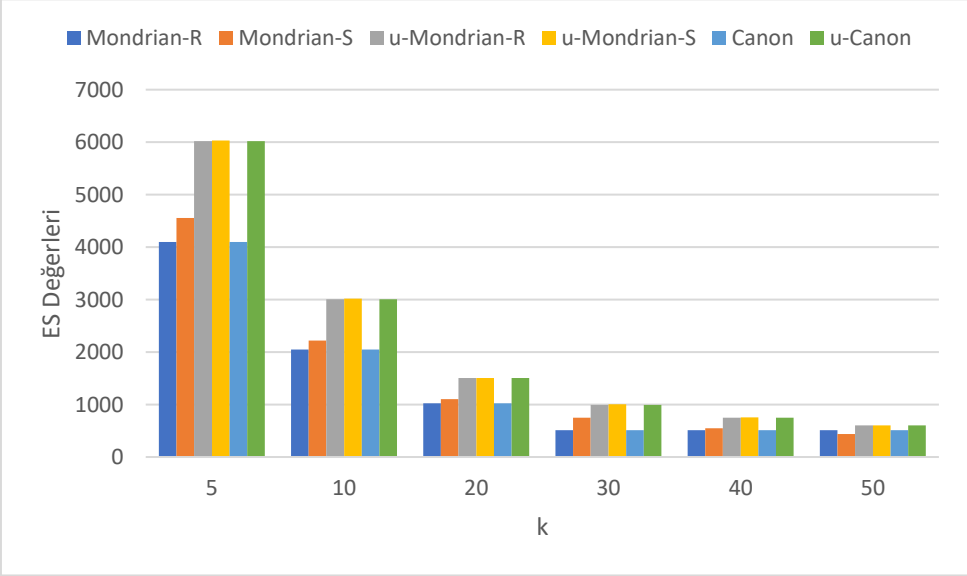
Yapılan testlerde elde edilen ES, DM ve AECS değerlerine göre;

- Mondrian-R ile Canon modelleri birbirine yakın sonuçlar üretmektedir,
- u-Mondrian-S, u-Mondrian-R ve u-Canon modelleri birbirine yakın sonuçlar üretmekte ve bu sonuçların ise literatürdeki mevcut sonuçlardan daha iyi olduğu görülmektedir

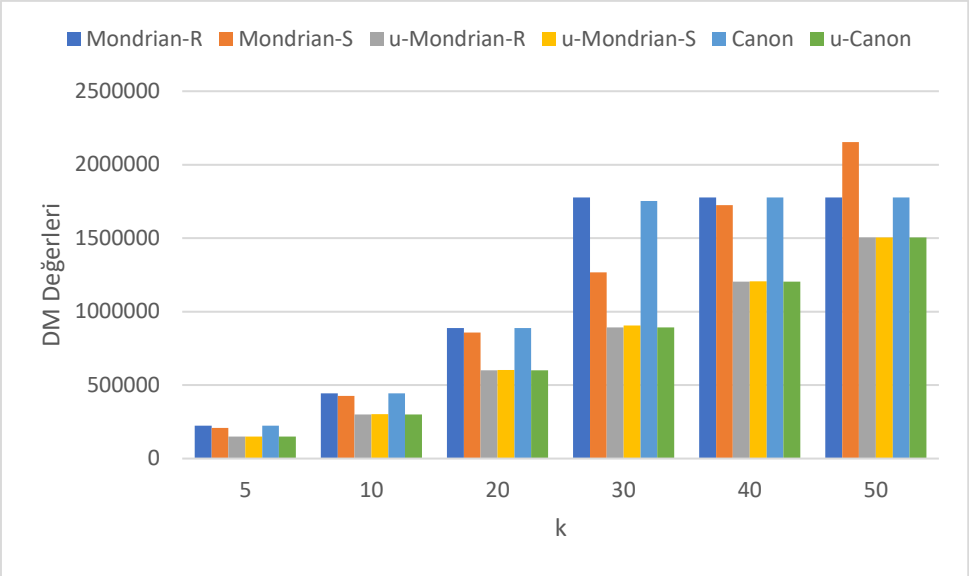
GCP metriğine göre elde edilen sonuçlara bakıldığında ise;

- u-Mondrian-R modeli, Mondiran-R modeline göre daha yüksek veri faydası sunmaktadır,
- Canon modeli Mondrian-S modeline göre daha iyi sonuçlar vermektedir,
- u-Mondrian-S ile u-Canon modellerinin sunduğu veri faydası birbirine çok yakın olmakla beraber, bu modellerin geliştirilen diğer modeller temel alındığında en yüksek veri faydası sunan modeller olduğu görülmektedir.

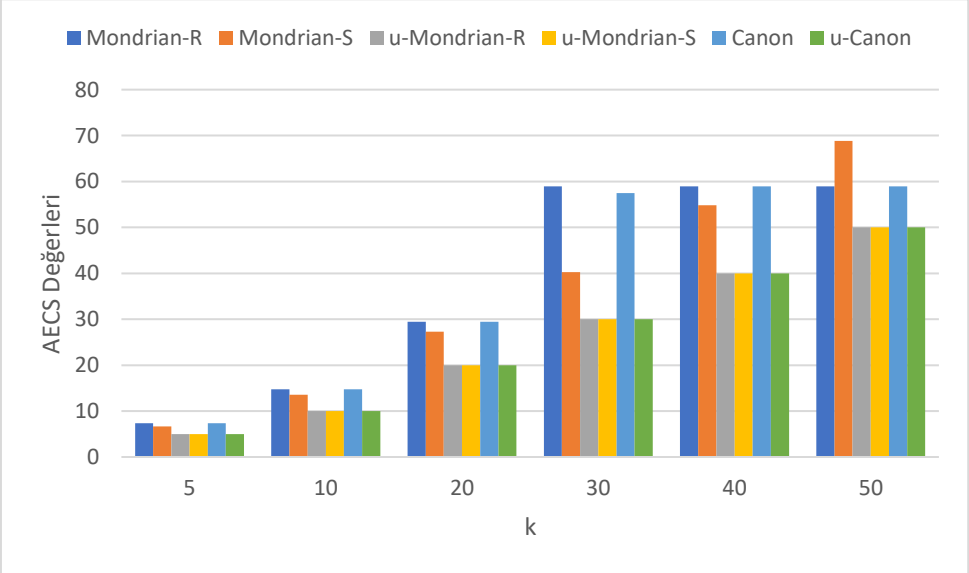
Elde edilen sonuçlara göre, geliştirilen modellerin başarılı ve kabul edilebilir sonuçlar verdiği, u-Canon ve u-Mondrian-S modellerinin en yüksek veri faydası sunan modeller olduğu görülmektedir.



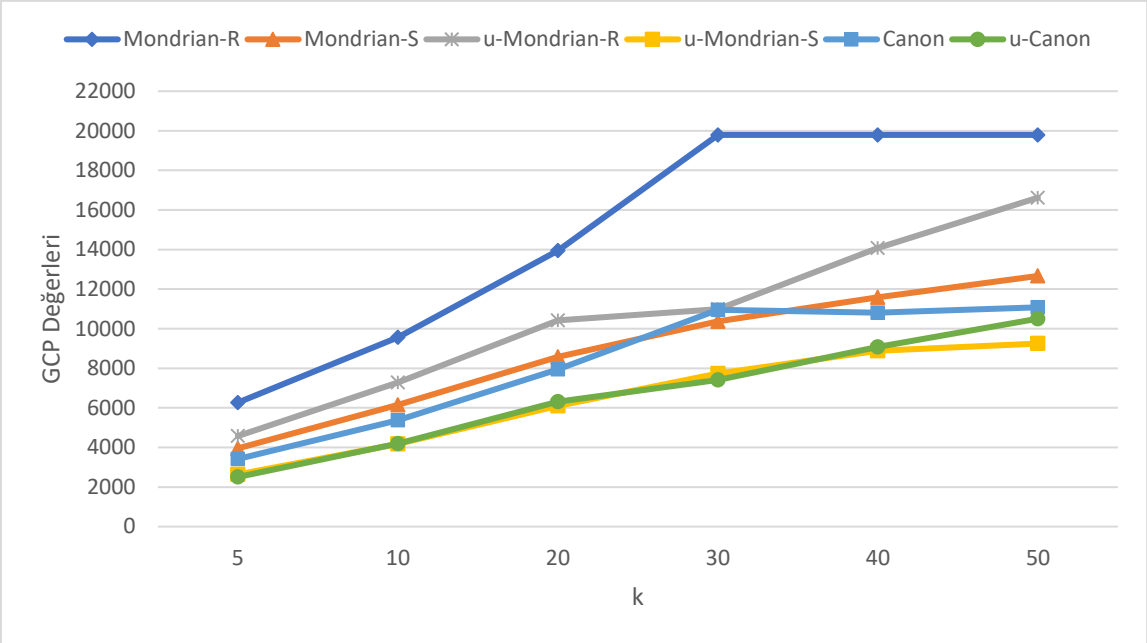
Şekil 8.1. Tez kapsamında geliştirilen tüm modellerin ES değerlerine göre karşılaştırılması



Şekil 8.2. Tez kapsamında geliştirilen tüm modellerin DM değerlerine göre karşılaştırılması



Şekil 8.3. Tez kapsamında geliştirilen tüm modellerin AECS değerlerine göre karşılaştırılması



Şekil 8.4. Tez kapsamında geliştirilen tüm modellerin GCP değerlerine göre karşılaştırılması

Büyük veri mimarisi için geliştirilen Su-Mondrian modeli ise klasik SMondrian modeline göre daha yüksek veri faydası sunmuş ve bu alanda geliştirilen ilk model olduğu için farklı modeller ile karşılaştırılamamasına rağmen SMondrian modeli ile karşılaştırmanın da bu aşamada yeterli olduğu değerlendirilmiştir.

Bu tez çalışmasında geliştirilen modellerin iyileştirmesi için yapılabilecek yeni çalışmalara dair değerlendirmeler aşağıda sunulmuştur;

1. u-Mondrian modeli için;
    - aykırı veri belirlemede farklı yaklaşımlar kullanılabilir,
    - bölme değerinin belirlenmesi için farklı yaklaşımlar kullanılabilir,
    - aykırı verilere ek olarak normal verilerin de bir yönetim işlemine tabi tutulmasıyla veri faydası arttırılabilir,
2. Canon modeli için;
    - bölme değeri belirlemede farklı yaklaşımlar kullanılabilir,
    - referans noktası belirlemede daha etkin yaklaşımlar kullanılabilir,
    - uzaklıkların belirlenmesinde farklı fonksiyonlar kullanılabilir,
3. u-Canon modeli için;
    - aykırı veri belirlemede farklı yaklaşımlar kullanılabilir,

- bölme değerinin belirlenmesi için farklı yaklaşımlar kullanılabilir,
- aykırı verilere ek olarak normal verilerin de bir yönetim işlemine tabi tutulmasıyla veri faydası arttırılabilir,

4. Su-Mondrian modeli için;
    - büyük veri mimarisinde aykırı veri belirlemede farklı yaklaşımlar kullanılabilir,
    - bölme değerinin belirlenmesi için farklı yaklaşımlar kullanılabilir,
    - büyük veri mimarisine uygun bir şekilde normal verilerin de bir yönetim işlemine tabi tutulmasıyla veri faydası arttırılabilir.

Bu tez çalışmasında karşılaşılan zorluklar ise aşağıda verilmiştir;

- veri mahremiyeti konusu hassas bir konu oluğu için tez kapsamında gerçek verilere ulaşılması ve tez kapsamında test edilmesinde zorluklar yaşanmıştır,
- büyük veri ortamlarının beraberinde getirdiği teknoloji, mimari, alt yapı ve programlama paradigmaları karmaşık ve zor yapılardır,
- büyük veri ortamlarında iteratif yapıya sahip anonimleştirme modellerin geliştirilmesi ve uygulanması zordur,
- Mondrian modeli kapsamında aykırı veri ve normal veri için gerekli tanımlar literatürde bulunmamaktadır,
- aykırı verilerin veri faydasına etkisini ölçmede kullanılabilecek kapsamlı karar kuralları literatürde mevcut değildir,
- Mondrian modeli kapsamında aykırı verilerin belirlenmesinde kullanılabilecek bir yaklaşım mevcut değildir ve
- Mondrian modeline aykırı veri konsepti uygulayan başka bir çalışma literatürde yoktur.

KAYNAKLAR

1. De Capitani Di Vimercati, S., Foresti, S., Livraga, G. and Samarati, P. (2012). Data Privacy: Definitions and Techniques. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* 20(06), 793-817.

2. Fung, B., Wang, K., Chen, R. and Yu, P. (2010). Privacy-Preserving Data Publishing: A Survey of Recent Developments. *Computing Surveys,* 42(4), 14.

3. El Emam, K. and Arbuckle, L. (2013). *Anonymizing Health Data: Case Studies and Methods To Get You Started*. O'Reilly Media, 20.

4. Sweeney, L. (2001). *Computational Disclosure Control: A Primer on Data Privacy Protection*, Ph. D. Thesis, Massachusetts Institute of Technology Department of Electrical Engineering and Computer Science, USA.

5. Meyerson, A. and Williams, R. (2004). *On the Complexity of Optimal k-Anonymity*. ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 223-228, Paris, France.

6. Sweeney, L. (2002). k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* 10(5), 557-570.

7. Wang, H.W. and Liu, R. (2015). Hiding Outliers into Crowd: Privacy-Preserving Data Publishing with Outliers. *Data & Knowledge Engineering,* 100, 94-115.

8. Vural, Y. (2017). *p-Kazanım: Mahremiyet Korumalı Fayda Temelli Veri Yayınlama Modeli*, Doktora Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

9. Fung, B.C., Wang, K., Fu, A.W. and Philip, S.Y. (2010). *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. USA: CRC Press, 43.

10. Kohlmayer, F., Prasser, F. and Kuhn, K.A. (2015). The Cost of Quality: Implementing Generalization and Suppression for Anonymizing Biomedical Data with Minimal Information Loss. *Journal of Biomedical Informatics*, 58, 37-48.

11. Prasser, F., Eicher, J., Bild, R., Spengler, H. and Kuhn, K.A. (2017). *A Tool for Optimizing De-Identified Health Data for Use in Statistical Classification*. IEEE International Symposium on Computer-Based Medical Systems, 169-174, Thessaloniki, Greece.

12. Prasser, F., Bild, R., Eicher, J., Spengler, H., Kohlmayer, F. and Kuhn, K.A. (2016). Lightning: Utility-Driven Anonymization of High-Dimensional Data. *Transactions on Data Privacy,* 9(2), 161-185.

13. Vural, Y. and Aydos, M. (2017). *A New Approach to Utility-Based Privacy Preserving in Data Publishing*. IEEE International Conference on Computer and Information Technology, 204-209, Helsinki, Finland.

14. Ramana, K.V., Kumari, V. and Raju, K. (2011). *Impact of Outliers on Anonymized Categorical Data*. Advances in Digital Image Processing and Information Technology, 326-335, Berlin, Germany.

15. Aggarwal, C.C. and Philip, S.Y. (2008). *Privacy-Preserving Data Mining: Models and Algorithms*. USA: Springer Science & Business Media, 39.

16. Hu, H., Wen, Y., Chua, T.S. and Li, X. (2014). Toward Scalable Systems for Big Data Analytics: A Technology Tutorial. *IEEE Access*, 2, 652-687.

17. Warren, S.D. and Brandeis, L.D. (1890). The Right to Privacy. *Harvard Law Review*, 193-220.

18. Chibba, M. and Cavoukian, A. (2015). *Privacy, Consumer Trust and Big Data: Privacy by Design and the 3 C's*. ITU Kaleidoscope: Trust in the Information Society, 1-5, Barcelona, Spain.

19. Jain, P., Gyanchandani, M. and Khare, N. (2016). Big Data Privacy: A Technological Perspective and Review. *Journal of Big Data*, 3(1), 25.

20. Pentland, A.S. (2013). The data-driven society. *Scientific American*, 309(4), 78-83.

21. Cavanillas, J.M., Curry, E. and Wahlster, W. (2016). *New Horizons for a Data-Driven Economy: a Roadmap for Usage and Exploitation of Big Data in Europe*. İsviçre: Springer,

22. Jurczyk, P. and Xiong, L. (2009). Distributed Anonymization: Achieving Privacy for Both Data Subjects and Data Providers. *Data and Applications Security XXIII*, 5645, 191-207.

23. İnternet: Sweeney, L. Simple Demographics Often Identify People Uniquely. *Data Privacy Working Paper 3*.URL: https://dataprivacylab.org, Son Erişim Tarihi: 19.02.2018.

24. Machanavajjhala, A., Gehrke, J., Kifer, D. and Venkitasubramaniam, M. (2006). *l-Diversity: Privacy Beyond k-Anonymity*. International Conference on Data Engineering, 24-24, Atlanta, USA.

25. Motwani, R. and Nabar, S. (2008). Anonymizing Unstructured Data. *arXiv:0810.5582*.

26. Li, N., Li, T. and Venkatasubramanian, S. (2007). *t-Closeness: Privacy Beyond k-Anonymity and l-Diversity*. IEEE International Conference on Data Engineering, 106-115, Istanbul, Turkey.

27. Aggarwal, C. (2005). *On k-Anonymity and the Curse of Dimensionality*. International Conference on Very Large Data Bases, 901-909, Trondheim, Norway.

28. Fayyoumi, E. and Oommen, B.J. (2010). A Survey on Statistical Disclosure Control and Micro-Aggregation Techniques for Secure Statistical Databases. *Software: Practice and Experience*, 40(12), 1161-1188.

29. Agrawal, R. and Srikant, R. (2000). *Privacy-preserving data mining*. ACM SIGMOD Record, 439-450.

30. Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D. and Zhu, A. (2005). *Anonymizing Tables*. International Conference on Database Theory, 246-258, Edinburgh, UK.

31. Samarati, P. (2001). Protecting Respondents Identities in Microdata Release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6), 1010-1027.

32. Wong, R.C., Li, J., Fu, A.W. and Wang, K. (2006). *(α, k)-Anonymity: An Enhanced k-Anonymity Model for Privacy Preserving Data Publishing*. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 754-759, Philadelphia, USA.

33. Bayardo, R.J. and Agrawal, R. (2005). *Data Privacy Through Optimal k-Anonymization*. International Conference on Data Engineering, 217-228, Tokyo, Japan.

34. Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D. and Zhu, A. (2005). Approximation Algorithms for k-Anonymity. *Journal of Privacy Technology*, 1-18.

35. LeFevre, K., DeWitt, D. and Ramakrishnan, R. (2006). *Mondrian Multidimensional k-Anonymity*. International Conference on Data Engineering, 25-25, Atlanta, USA.

36. Sun, X., Sun, L. and Wang, H. (2011). Extended k-Anonymity Models Against Sensitive Attribute Disclosure. *Computer Communications*, 34(4), 526-535.

37. Bonizzoni, P., Della Vedova, G. and Dondi, R. (2011). Anonymizing Binary and Small Tables is Hard to Approximate. *Journal of Combinatorial Optimization*, 22(1), 97-119.

38. Blocki, J. and Williams, R. (2010). *Resolving the Complexity of Some Data Privacy Problems*. International Colloquium on Automata, Languages, and Programming, 393-404, Bordeaux, France.

39. Scott, A., Srinivasan, V. and Stege, U. (2015). k-Attribute-Anonymity is Hard Even for k=2. *Information Processing Letters*, 115(2), 368-370.

40. Aggarwal, C.C. (2007). *On Randomization, Public Information and the Curse of Dimensionality*. IEEE International Conference on Data Engineering, 136-145, Istanbul, Turkey.

41. Sweeney, L. (1998). Datafly: A System for Providing Anonymity in Medical Data. *Database Security XI*. USA. Springer, 356-381.

42. Wang, K., Yu, P.S. and Chakraborty, S. (2004). *Bottom-Up Generalization: A Data Mining Solution to Privacy Protection*. IEEE International Conference on Data Mining, 249-256, Brighton, UK.

43. Fung, B.C., Wang, K. and Yu, P. (2005). *Top-Down Specialization for Information and Privacy Preservation*. International Conference on Data Engineering, 205-216, Tokyo, Japan.

44. Beyer, M.A. and Laney, D. (2012). *Gartner*.

45. James, M., Michael, C., Brad, B., Jacques, B., Richard, D., Charles, R. and Angela, H. (2011). Big Data: The Next Frontier for Innovation, Competition, and Productivity. *The McKinsey Global Institute*.

46. Gantz, J. and Reinsel, D. (2011). Extracting value from chaos. *IDC Iview*, 1142(2011), 1-12.

47. Sun, J. and Reddy, C. (2013). *Big Data Analytics for Healthcare*. International Conference on Knowledge Discovery and Data Mining, Illinois, USA.

48. Mehmood, A., Natgunanathan, I., Xiang, Y., Hua, G. and Guo, S. (2016). Protection of Big Data Privacy. *IEEE Access*, 4, 1821-1834.

49. Nandini Prasaad, K. and Pratheek, T. (2015). *Providing Anonymity Using Top Down Specialization on Big Data Using Hadoop Framework*. IEEE Annual India Conference, 1-6, New Delhi, India.

50. Patil, H.K. and Seshadri, R. (2014). *Big Data Security and Privacy Issues in Healthcare*. IEEE International Congress on Big Data 762-765, Anchorage, AK, USA.

51. Victor, N., Lopez, D. and Abawajy, J.H. (2016). Privacy Models for Big Data: A Survey. *International Journal of Big Data Intelligence*, 3(1), 61-75.

52. Zhang, X., Yang, C., Nepal, S., Liu, C., Dou, W. and Chen, J. (2013). *A Mapreduce Based Approach of Scalable Multidimensional Anonymization for Big Data Privacy Preservation on Cloud*. International Conference on Cloud and Green Computing, 105-112, Karlsruhe, Germany.

53. Li, W. and Li, H. (2016). *LRDM: Local Record-Driving Mechanism for Big Data Privacy Preservation in Social Networks*. IEEE International Conference on Data Science in Cyberspace, 556-560, Changsha, China.

54. Olaronke, I. and Oluwaseun, O. (2016). *Big Data in Healthcare: Prospects, Challenges and Resolutions*. Future Technologies Conference, 1152-1157, San Francisco, USA.

55. Tanwar, M., Duggal, R. and Khatri, S.K. (2015). *Unravelling Unstructured Data: A Wealth of Information in Big Data*. International Conference on Reliability, Infocom Technologies and Optimization, 1-6, Noida, India.

56. İnternet: Vorhies, W. How Many V's in Big Data? The Characteristics that Define Big Data. URL: https://www.datasciencecentral.com/profiles/blogs/how-many-v-s-in-big-data-the-characteristics-that-define-big-data, Son Erişim Tarihi: 10.10.2017.

57. İnternet: Big Data: Data Wrangling Boot Camp Big Data Vs. URL: http://www.cs.odu.edu/~ccartled/Teaching/2017-Spring/DataWrangling/Presentations/030-bigDataVs.pdf, Son Erişim Tarihi: 05.08.2017.

58. İnternet: Divakar, M., Shrikant, K. and Shweta, J. Introduction to Big Data Classification and Architecture. URL: https://www.ibm.com/developerworks/analytics/library/bd-archpatterns1/index.html, Son Erişim Tarihi: 23.02.2018.

59. Furht, B. and Villanustre, F. (2016). *Big Data Technologies and Applications*. İsviçre: Springer, 4.

60. Tanenbaum, A. and Van Steen, M. (2007). *Distributed Systems: Principles and Paradigms*. USA: Prentice-Hall, 17.

61. İnternet: Matt Turck, D.O. Great Power, Great Responsibility: The 2018 Big Data & AI Landscape. URL: http://mattturck.com/bigdata2018/, Son Erişim Tarihi: 05.01.2018.

62. Ankam, V. (2016). *Big Data Analytics*. UK: Packt Publishing, 13.

63. Samadi, Y., Zbakh, M. and Tadonki, C. (2016). *Comparative Study Between Hadoop and Spark Based on Hibench Benchmarks*. International Conference on Cloud Computing Technologies and Applications, 267-275, Marrakech, Morocco.

64. Lee, M., Kim, E., Nam, C. and Shin, D. (2017). *Design of Educational Big Data Application Using Spark*. International Conference on Advanced Communication Technology, 355-357, Bongpyeong, South Korea.

65. Jam, M.R., Khanli, L.M., Javan, M.S. and Akbari, M.K. (2014). *A Survey on Security of Hadoop*. International eConference onComputer and Knowledge Engineering, 716-721, Mashhad, Iran.

66. Sogodekar, M., Pandey, S., Tupkari, I. and Manekar, A. (2016). *Big Data Analytics: Hadoop and Tools*. IEEE Bombay Section Symposium, 1-6, Baramati, India.

67. Mohanty, H., Bhuyan, P. and Chenthati, D. (2015). *Big Data: A Primer*. Switzerland: Springer, 61.

68. Karambelkar, H.V. (2018). *Apache Hadoop 3 Quick Start Guide*. USA: Packt Publishing, 1.

69. Miner, D. and Shook, A. (2012). *Mapreduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems*. USA: O'Reilly Media, 1.

70. Buyya, R., Calheiros, R.N. and Dastjerdi, A.V. (2016). *Big Data: Principles and Paradigms*. USA: Morgan Kaufmann, 1.

71. White, T. (2012). *Hadoop: The Definitive Guide*. USA: O'Reilly Media, 343.

72. Karau, H., Konwinski, A., Wendell, P. and Zaharia, M. (2015). *Learning Spark: Lightning-Fast Big Data Analysis*. USA: O'Reilly Media, 3.

73. Mehrotra, S. and Grade, A. (2019). *Apache Spark Quick Start Guide*. USA: Packt Publishing, 38.

74. İnternet: Apache Spark. URL: http://spark.apache.org/, Son Erişim Tarihi: 06.06.2017.

75. Frampton, M. (2015). *Mastering Apache Spark*. USA: Packt Publishing, 1.

76. Zomaya, A.Y. and Sakr, S. (2017). *Handbook of Big Data Technologies*. Switzerland: Springer, 1.

77. Canbay, Y., Vural, Y. and Sagiroglu, S. (2018). *Privacy Preserving Big Data Publishing*. IEEE International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism, 24-29, Ankara, Turkey.

78. Wong, R.C., Fu, A.W., Wang, K. and Pei, J. (2007). *Minimality Attack In Privacy Preserving Data Publishing*. International Conference on Very Large Data Bases, 543-554, Vienna, Austria.

79. Duncan, G. and Lambert, D. (1989). The risk of disclosure for microdata. *Journal of Business & Economic Statistics*, 7(2), 207-217.

80. Skinner, C. and Holmes, D.J. (1998). Estimating the Re-Identification Risk per Record in Microdata. *Journal of Official Statistics*, 14(4), 361-372.

81. Dankar, F.K., El Emam, K., Neisa, A. and Roffey, T. (2012). Estimating the Re-Identification Risk of Clinical Data Sets. *Bmc Medical Informatics and Decision Making*, 12(1), 66.

82. Winkler, W. (2004). *Masking and Re-Identification Methods for Public-Use Microdata: Overview and Research Problems*. International Workshop on Privacy in Statistical Databases, 231-246, Barcelona, Spain.

83. Domingo-Ferrer, J. and Torra, V. (2008). *A Critique of k-Anonymity and Some of Its Enhancements*. 2008 Third International Conference on Availability, Reliability and Security, 990-993, Barcelona, Spain.

84. Nergiz, M.E., Atzori, M. and Clifton, C. (2007). *Hiding the Presence of Individuals from Shared Databases*. ACM SIGMOD International Conference on Management of Data, 665-676, Beijing, China.

85. Chen, B., LeFevre, K. and Ramakrishnan, R. (2007). *Privacy Skyline: Privacy with Multidimensional Adversarial Knowledge*. International Conference on Very Large Data Bases, 770-781, Vienna, Austria.

86. Fang, W., Wen, X.Z., Zheng, Y. and Zhou, M. (2017). A Survey of Big Data Security and Privacy Preserving. *IETE Technical Review,* 34(5), 544-560.

87. Zhang, X., Yang, L.T., Liu, C. and Chen, J. (2014). A Scalable Two-Phase Top-Down Specialization Approach for Data Anonymization Using Mapreduce on Cloud. *IEEE Transactions on Parallel and Distributed Systems,* 25(2), 363-373.

88. Sweeney, L. (2002). Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* 10(5), 571-588.

89. Ciriani, V., Di Vimercati De Capitani, S., Foresti, S. and Samarati, P. (2008). k-anonymous data mining: A survey. *Privacy-preserving Data Mining*. Springer, 105-136.

90. Samarati, P. and Sweeney, L. (1998). *SRI International Technical Report.*

91. Tao, Y., Tong, Y., Tan, S., Tang, S. and Yang, D. (2008). *Protecting the Publishing Identity in Multiple Tuples*. Annual Conference on Data and Applications Security and Privacy, 205-218, London, UK.

92. Kenig, B. and Tassa, T. (2012). A practical approximation algorithm for optimal k-anonymity. *Data Mining and Knowledge Discovery,* 25(1), 134-168.

93. Li, N., Li, T. and Venkatasubramanian, S. (2010). Closeness: A New Privacy Measure for Data Publishing. *IEEE Transactions on Knowledge and Data Engineering,* 22(7), 943-956.

94. Gkoulalas Divanis, A. and Loukides, G. (2015). *Medical Data Privacy Handbook*. Switzerland: Springer, 21.

95. LeFevre, K., DeWitt, D.J. and Ramakrishnan, R. (2005). *Incognito: Efficient Full-domain k-Anonymity*. ACM SIGMOD International Conference on Management of Data, 49-60, Baltimore, Maryland.

96. Kohlmayer, F., Prasser, F., Eckert, C., Kemper, A. and Kuhn, K.A. (2012). *Flash: Efficient, Stable and Optimal k-Anonymity*. International Conference on Privacy, Security, Risk and Trust and International Confernece on Social Computing, 708-717, Amsterdam, Netherlands.

97. Bentley, J.L. (1975). Multidimensional Binary Search Trees Used for Associative Searching. *Communications of the ACM,* 18(9), 509-517.

98. Tang, Q., Wu, Y. and Wang, X. (2010). *New Algorithm with Lower Upper Size Bound for k-Anonymity*. International Conference on Communication Systems, Networks and Applications, 421-424, Hong Kong, China.

99. Liu, K.C., Kuo, C.W., Liao, W.C. and Wang, P.C. (2018). *Optimized Data de-Identification Using Multidimensional k-Anonymity*. IEEE International Conference On Trust, Security And Privacy In Computing And Communication, IEEE International Conference On Big Data Science And Engineering, 1610-1614, Newyork, USA.

100. Gong, Q., Yang, M., Chen, Z. and Luo, J. (2016). *Utility Enhanced Anonymization for Incomplete Microdata*. International Conference on Computer Supported Cooperative Work in Design, 74-79, Nanchang, China.

101. Ruggieri, S. (2014). Using t-Closeness Anonymity to Control for Non-Discrimination. *Transaction on Data Privacy,* 7(2), 99-129.

102. Nergiz, M.E. and Gök, M.Z. (2014). Hybrid k-Anonymity. *Computers & Security*, 44, 51-63.

103. LeFevre, K., DeWitt, D. and Ramakrishnan, R. (2006). *Workload-aware Anonymization*. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 277-286, Philadelphia, USA.

104. Xiao, X. and Tao, Y. (2006). *Personalized Privacy Preservation*. ACM SIGMOD International Conference on Management of Data, 229-240, Chicago, USA.

105. Skowron, A. and Rauszer, C. (1992). The Discernibility Matrices and Functions in Information Systems. *Intelligent Decision Support*. Springer, 331-362.

106. Ghinita, G., Karras, P., Kalnis, P. and Mamoulis, N. (2007). *Fast Data Anonymization with Low Information Loss*. International Conference on Very Large Databases, 758-769, Vienna, Austria.

107. Witten, I.H., Frank, E., Hall, M.A. and Pal, C.J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. USA: Morgan Kaufmann, 1.

108. Han, J., Pei, J. and Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Elsevier, 1.

109. Aggarwal, C.C. (2017). *Outlier Analysis*. USA: Springer, Cham, 1.

110. Vural, Y. and Aydos, M. (2018). ρ-Gain: Utility Based Data Publishing Model. *Journal of the Faculty of Engineering and Architecture of Gazi University,* 33(4), 1355-1368.

111. Lee, H., Kim, S., Kim, J.W. and Chung, Y.D. (2017). Utility-Preserving Anonymization for Health Data Publishing. *BMC Medical Informatics and Decision Making*, 17(104), 1-12.

112. Wang, H.W. and Liu, R. (2009). *Hiding Distinguished Ones into Crowd: Privacy-Preserving Publishing Data with Outliers*. International Conference on Extending Database Technology: Advances in Database Technology, 624-635, Saint Petersburg, Russia.

113. Majeed, A. (in press). Attribute-Centric Anonymization Scheme for Improving User Privacy And Utility of Publishing E-Health Data. *Journal of King Saud University-Computer and Information Sciences*.

114. Nergiz, M.E., Gök, M.Z. and Özkanlı, U. (2013). *Preservation of Utility Through Hybrid k-Anonymization*. International Conference on Trust, Privacy and Security in Digital Business, 97-111, Prague, Czech Republic.

115. Prasser, F., Kohlmayer, F., Lautenschlaeger, R. and Kuhn, K.A. (2014). *Arx-A Comprehensive Tool for Anonymizing Biomedical Data*. AMIA Annual Symposium Proceedings, 984-993, Washington, USA.

116. Canbay, Y., Vural, Y. and Sağıroğlu, Ş. (baskıda). Oan: Aykırı Veri Yönelimli Fayda Temelli Mahremiyet Koruma Modeli. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*.

117. Breunig, M.M., Kriegel, H.P., Ng, R.T. and Sander, J. (2000). *LOF: Identifying Density-Based Local Outliers*. ACM International Conference on Management of Data, 93-104, Dallas, USA.

118. Zhang, X., Liu, C., Nepal, S., Yang, C., Dou, W. and Chen, J. (2014). A Hybrid Approach for Scalable Sub-Tree Anonymization Over Big Data Using Mapreduce on Cloud. *Journal of Computer and System Sciences*, 80(5), 1008-1020.

119. Zhang, X., Dou, W., Pei, J., Nepal, S., Yang, C., Liu, C. and Chen, J. (2015). Proximity-Aware Local-Recoding Anonymization with Mapreduce for Scalable Big Data Privacy Preservation in Cloud. *IEEE Transactions on Computers*, 64(8), 2293-2307.

120. Krizan, T., Brakus, M. and Vukelic, D. (2015). *In-situ Anonymization of Big Data*. International Convention on Information and Communication Technology, Electronics and Microelectronics, 292-298, Opatija, Croatia.

121. Kavitha, S., Yamini, S. and Vadhana, R. (2015). *An Evaluation on Big Data Generalization Using k-Anonymity Algorithm on Cloud*. International Conference on Intelligent Systems and Control, 1-5, Coimbatore, India.

122. Al-Zobbi, M., Shahrestani, S. and Ruan, C. (2016). *Sensitivity-based Anonymization of Big Data*. IEEE Conference on Local Computer Networks, 58-64, Dubai, United Arab Emirates.

123. Zakerzadeh, H., Aggarwal, C.C. and Barker, K. (2015). *Privacy-Preserving Big Data Publishing*. International Conference on Scientific and Statistical Database Management, 26, California, USA.

124. Zhang, X., Leckie, C., Dou, W., Chen, J., Kotagiri, R. and Salcic, Z. (2016). *Scalable Local-Recoding Anonymization using Locality Sensitive Hashing for Big Data Privacy Preservation*. ACM International on Conference on Information and Knowledge Management, 1793-1802, Indiana, USA.

125. Canbay, Y. and Sağıroğlu, S. (2017). *Big Data Anonymization with Spark*. International Conference on Computer Science and Engineering, 833-838, Antalya, Turkey.

126. Mehta, B.B. and Rao, U.P. (2017). Privacy Preserving Big Data Publishing: A Scalable k-Anonymization Approach Using Mapreduce. *IET Software*, 11(5), 271-276.

127. Mehta, B.B. and Rao, U.P. (in press). Improved l-Diversity: Scalable Anonymization Approach for Privacy Preserving Big Data Publishing. *Journal of King Saud University-Computer and Information Sciences*.

128. İnternet: Dheeru, D. and Taniskidou, E.K. UCI Machine Learning Repository. URL: http://archive.ics.uci.edu/ml, Son Erişim Tarihi: 25.03.2019.

129. Kumar, N., Zhang, L. and Nayar, S. (2008). *What is a Good Nearest Neighbors Algorithm for Finding Similar Patches in Images?* European Conference on Computer Vision, 364-378, Marseille, France.

130. Dolatshah, M., Hadian, A. and Minaei-Bidgoli, B. (2015). Ball*-Tree: Efficient Spatial Indexing for Constrained Nearest-Neighbor Search in Metric Spaces. *Preprint arXiv:1511.00628*.

131. Wang, J., Wang, N., Jia, Y., Li, J., Zeng, G., Zha, H. and Hua, X. (2014). Trinary-Projection Trees for Approximate Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2), 388-403.

132. Yianilos, P.N. (1993). *Data Structures and Algorithms for Nearest Neighbor Search In General Metric Spaces*. Symposium on Discrete Algorithms, 311-321, Texas, USA.

133. Samet, H. (2006). *Foundations of Multidimensional and Metric Data Structures*. USA: Morgan Kaufmann, 1.

134. Fu, A.W., Chan, P.M., Cheung, Y. and Moon, Y. (2000). Dynamic Vp-Tree Indexing for N-Nearest Neighbor Search Given Pair-Wise Distances. *International Journal on Very Large Data Bases*, 9(2), 154-173.

135. Bozkaya, T. and Ozsoyoglu, M. (1997). *Distance-Based Indexing for High-Dimensional Metric Spaces*. ACM SIGMOD International Conference on Management of Data, 357-368, Arizona, USA.

136. DeFreitas, T., Saddiki, H. and Flaherty, P. (2016). GEMINI: A Computationally-Efficient Search Engine for Large Gene Expression Datasets. *BMC Bioinformatics*, 17(1), 102.

137. Böhm, C., Berchtold, S. and Keim, D.A. (2001). Searching in High-Dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases. *ACM Computing Surveys*, 33(3), 322-373.

138. Bozkaya, T. and Ozsoyoglu, M. (1999). Indexing Large Metric Spaces for Similarity Search Queries. *ACM Transactions on Database Systems,* 24(3), 361-404.

139. Zhang, F., Di, P., Zhou, H., Liao, X. and Xue, J. (2016). *RegTT: Accelerating Tree Traversals on GPUs by Exploiting Regularities*. International Conference on Parallel Processing, 562-571, Philadelphia, USA.

140. Kristensen, T.G. and Pedersen, C.N. (2010). *Data Structures for Accelerating Tanimoto Queries on Real Valued Vectors*. International Workshop on Algorithms in Bioinformatics, 28-39, Liverpool, UK.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : CANBAY, Yavuz
Uyruğu : T.C.
Doğum tarihi ve yeri : 29.03.1988, Elazığ
Medeni hali : Evli
Telefon : 0 (539) 379 42 39
e-mail : yavuzcanbay@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Doktora	Gazi Üniversitesi / Bilgisayar Mühendisliği	Devam Ediyor
Yüksek lisans	Erciyes Üniversitesi / Bilgisayar Mühendisliği	2013
Lisans	Harran Üniversitesi / Bilgisayar Mühendisliği	2011
Lise	Elazığ Balakgazi Lisesi (YDA)	2006

İş Deneyimi

Yıl	Yer	Görev
2013-Halen	Gazi Üniversitesi	Araştırma Görevlisi
2012-2013	Erciyes Üniversitesi	Araştırma Görevlisi
2012-2012	Kahramanmaraş Sütçü İmam Üniversitesi	Araştırma Görevlisi

Yabancı Dil

İngilizce

Yayınlar

1. Canbay, Y., Vural, Y. ve Sağıroğlu, S. (baskıda). Oan: aykırı veri yönelimli fayda temelli mahremiyet koruma modeli. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*.
2. Canbay, Y., Vural, Y. ve Sağıroğlu, S. (baskıda). Mahremiyet Korumalı Büyük Veri Yayınlama İçin Kavramsal Model Önerileri. *Politeknik Dergisi*

3. Canbay, Y., Sağıroğlu, S. and Vural, Y. (hakem değerlendirmesinde). u-Mondrian: A New Utility-aware Mondrian-based Anonymization Model. *IEEE Transactions on Knowledge and Data Engineering*.
4. Canbay, Y. and Sağıroğlu, S. (2017). *Big data anonymization with spark*. IEEE International Conference on Computer Science and Engineering, 24-29, Antalya, Turkey.
5. Canbay, Y., Vural, Y. and Sağıroğlu, S. (2018). *Privacy Preserving Big Data Publishing*. International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism, Ankara, 833-838, Turkey.
6. Canbay, Y., Sağıroğlu, S. and Vural, Y. (baskıda). *A Mondrian-based Utility Optimization Model for Anonymization*. International Conference on Computer Science and Engineering, Samsun, Turkey.
7. Canbay, Y., Vural, Y. ve Sağıroğlu, S. (2017). *Büyük Veri Mahremiyetinde Kullanılan Kavramlar ve Yöntemler*. Ankara: Grafiker Yayınevi, 31.
8. Canbay, Y., Vural, Y. ve Sağıroğlu, S. (2017). *Büyük Verilerde Mahremiyet İhlali Endişesi, Açık Veri Yaklaşımları ve Alınabilecek Önlemler*. Ankara: Grafiker Yayınevi, 61.

Hobiler

Yüzme, Dijital oyunlar

GAZİ GELECEKTİR..