T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DOKTORA TEZİ

BÜYÜK VERİDE ETKİN GİZLİLİK KORUMASI İÇİN YAZILIM TASARIMI

Can EYÜPOĞLU

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

DANIŞMAN
Prof. Dr. Ahmet SERTBAŞ

II. DANIŞMAN
Dr. Öğr. Üyesi Muhammed Ali AYDIN

Haziran, 2018

İSTANBUL

Bu çalışma, 28.06.2018 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Programında Doktora tezi olarak kabul edilmiştir.

Tez Jürisi

Prof. Dr. Ahmet SERTBAŞ(Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi

Prof. Dr. Abdül Halim ZAİM
İstanbul Ticaret Üniversitesi
Mühendislik Fakültesi

Doç. Dr. Atakan KURT
İstanbul Üniversitesi
Mühendislik Fakültesi

Doç. Dr. Berk CANBERK
İstanbul Teknik Üniversitesi
Bilgisayar ve Bilişim Fakültesi

Dr. Öğr. Üyesi Özgür Can TURNA
İstanbul Üniversitesi
Mühendislik Fakültesi

20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Bu çalışma İstanbul Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda yapılan "Büyük Veride Etkin Gizlilik Koruması için Yazılım Tasarımı" adlı doktora tez çalışmasını içermektedir. Tez çalışması kapsamında yayınlanan makalelerin bilgileri aşağıdaki gibidir:

- Eyüpoğlu, C., Aydın, M.A., Zaim, A.H. and Sertbaş, A., 2018, An Efficient Big Data Anonymization Algorithm Based on Chaos and Perturbation Techniques, *Entropy*, 20 (5), 373, 1-18 (Science Citation Index Expanded).
- Eyüpoğlu, C., Aydın, M.A., Sertbaş, A., Zaim, A.H. and Öneş, O., 2017, Preserving Individual Privacy in Big Data, *International Journal of Informatics Technologies*, 10 (2), 177-184 (TÜBİTAK ULAKBİM TR Dizin).

Bu tez çalışması süresince gösterdiği her türlü destek ve yardımlarından dolayı çok değerli danışman hocalarım Prof. Dr. Ahmet SERTBAŞ ve Dr. Öğr. Üyesi Muhammed Ali AYDIN'a en içten dileklerimle teşekkür ederim.

Eğitim hayatım ve akademik kariyerim boyunca sürekli yanımda olan Prof. Dr. Abdül Halim ZAİM'e sonsuz teşekkürlerimi sunarım.

Tez çalışmam boyunca manevi desteklerini esirgemeyen başta Arş. Gör. Erdem YAVUZ, Arş. Gör. Ufuk ŞANVER ve Dr. Öğr. Üyesi Mustafa Cem KASAPBAŞI olmak üzere tüm çalışma arkadaşlarıma çok teşekkür ederim.

Doktora eğitimim süresince de manevi olarak her zaman yanımda olan, desteğini hiçbir zaman esirgemeyen eşim Dr. Şeyda EYÜPOĞLU'na sonsuz teşekkürlerimi sunarım.

Eğitim hayatım boyunca maddi ve manevi desteklerini hiçbir zaman esirgemeyen ve her zaman yanımda olan sevgili aileme en içten dileklerimle teşekkür ederim.

Tezimi sevgili eşime ve canım kızıma armağan ediyorum.

Haziran 2018 Can EYÜPOĞLU

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ

Sayfa No

TABLO LİSTESİ

Sayfa No

SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
$\boldsymbol{A}$	: Çapraz geçerlemede her bir katın doğruluk ölçümü
$\boldsymbol{A_n}$	: Naive Bayes sınıflandırıcıda *n* nitelikli grup üzerinde yapılan *n* ölçümleri
$\boldsymbol{c}$	: Önerilen algoritmada kullanılan bir sayıcı
$\boldsymbol{d}$	: Önerilen algoritmada orijinal giriş veri setinin boyutu
$\boldsymbol{C}$	: Bayes teoreminde bir sınıf
$\boldsymbol{C_i}$	: Naive Bayes sınıflandırıcıda *i*. sınıf
$\boldsymbol{C_m}$	: Naive Bayes sınıflandırıcıda *m* sayıda sınıf
$\boldsymbol{D}$	: Önerilen algoritmada orijinal giriş veri seti/Olasılıksal anonimlikte bir veri seti
$\boldsymbol{D'}$	: Olasılıksal anonimlikte *D*'nin anonimleştirilmiş hali
$\boldsymbol{D_p}$	: Önerilen algoritmada gizliliği korunmuş veri seti
$\boldsymbol{H}$	: Bayes teoreminde $\boldsymbol{X}$ veri grubunun belirtilen bir *C* sınıfına ait olması hipotezi
$\boldsymbol{k}$	: Çapraz geçerlemede kat sayısı/*k*-anonimlikte *k* sayısı
$\boldsymbol{m}$	: Olasılıksal anonimlikte yarı tanımlayıcı nitelik sayısı/Naive Bayes sınıflandırıcıda sınıf sayısı
$\boldsymbol{n}$	: Bayes teoreminde nitelik sayısı
$\boldsymbol{nu_i}$	: Önerilen algoritmada her bir QI_i için benzersiz değer sayısı
$\boldsymbol{N}$	: Sınıflandırmada negatif grupların sayısı
$\boldsymbol{N'}$	: Sınıflandırmada negatif etiketlenen grupların sayısı
$\boldsymbol{P}$	: Sınıflandırmada pozitif grupların sayısı
$\boldsymbol{P'}$	: Sınıflandırmada pozitif etiketlenen grupların sayısı
$\boldsymbol{p(x)}$	: Kullback-Leibler uzaklığında gizliliği korunmuş veri setinin dağılımı
$\boldsymbol{P(C_i)}$	: Naive Bayes sınıflandırıcıda C_i'nin önsel olasılığı
$\boldsymbol{P(C_m)}$	: Naive Bayes sınıflandırıcıda C_m'nin önsel olasılığı
$\boldsymbol{P(C_i\|X)}$	: Naive Bayes sınıflandırıcıda $\boldsymbol{X}$ üzerinde koşullu C_i'nin sonsal olasılığı
$\boldsymbol{P(H)}$	: Bayes teoreminde *H*'nin önsel olasılığı
$\boldsymbol{P(H\|X)}$	: Bayes teoreminde $\boldsymbol{X}$'in nitelik tanımı bilindiğinde, $\boldsymbol{X}$ grubunun *C* sınıfına ait olma olasılığı veya $\boldsymbol{X}$ üzerinde koşullu *H*'nin sonsal olasılığı
$\boldsymbol{P(X)}$	: Bayes teoreminde $\boldsymbol{X}$'in önsel olasılığı/Naive Bayes sınıflandırıcıda $\boldsymbol{X}$'in önsel olasılığı

$\boldsymbol{P(X|C_i)}$: Naive Bayes sınıflandırıcıda C_i üzerinde koşullu $\boldsymbol{X}$'in sonsal olasılığı

$\boldsymbol{P(X|H)}$: Bayes teoreminde H üzerinde koşullu $\boldsymbol{X}$'in sonsal olasılığı

$\boldsymbol{Pa(D')}$: D''nin olasılıksal ananomliği

$\boldsymbol{q}$: Önerilen algoritmada yarı tanımlayıcı nitelik sayısı

$\boldsymbol{q(x)}$: Kullback-Leibler uzaklığında orijinal veri setinin dağılımı

$\boldsymbol{Q_i}$: Olasılıksal anonimlikte D'de i. yarı tanımlayıcı nitelik

$\boldsymbol{QI}$: Önerilen algoritmada yarı tanımlayıcı nitelik

$\boldsymbol{QI_i}$: Önerilen algoritmada i. yarı tanımlayıcı nitelik

$\boldsymbol{QI_q}$: Önerilen algoritmada q sayıda yarı tanımlayıcı

$\boldsymbol{r}$: Önerilen algoritmada kritik benzersiz nitelik değerlerinin sayısı/Olasılıksal anonimlikte D'deki bir kayıt

$\boldsymbol{r_i}$: Önerilen algoritmada i. yarı tanımlayıcı nitelik için kritik benzersiz değerlerinin sayısı

$\boldsymbol{r'}$: Olasılıksal anonimlikte r'nin anonimleştirilmiş hali

$\boldsymbol{r(QI)}$: Olasılıksal anonimlikte r'deki yarı tanımlayıcının değer kombinasyonu

$\boldsymbol{r'(QI)}$: Olasılıksal anonimlikte r'nin anonimleştirilmiş halindeki yarı tanımlayıcının değer kombinasyonu

$\boldsymbol{SA}$: Önerilen algoritmada hassas nitelik

$\boldsymbol{u_{ij}}$: Önerilen algoritmada her bir QI_i için benzersiz değerler

$\mathbf{v}$: Online algılayıcı algoritmasında başlangıç sıfır tahmini vektörü

$\boldsymbol{v_{ij}}$: Önerilen algoritmada benzersiz değer u_{ij}'yi içeren kayıt sayısı

$\mathbf{x}$: Online algılayıcı algoritmasında bir örnek

$\boldsymbol{x_{ij}}$: Önerilen algoritmadaki kaotik fonksiyonun x değerleri

$\boldsymbol{x_k}$: Naive Bayes sınıflandırıcıda $\boldsymbol{X}$ grubunun A_k niteliğinin değerini

$\boldsymbol{X}$: Bayes teoreminde bir veri grubu/Naive Bayes sınıflandırıcıda n-boyutlu nitelik vektörü

$\boldsymbol{y}$: Online algılayıcı algoritmasında bir etiket

$\boldsymbol{\hat{y}}$: Online algılayıcı algoritmasında $\mathbf{x}$ örneğin tahmin edilen etiketi

$\boldsymbol{\lambda}$: Önerilen algoritmadaki kaotik fonksiyonun davranışının bağlı olduğu değer

$\boldsymbol{|C_{i,D}|}$: Naive Bayes sınıflandırıcıda D'deki C_i sınıfının eğitim gruplarının sayısı

$\boldsymbol{|D|}$: Önerilen algoritmada D'nin boyutu/Naive Bayes sınıflandırıcıda D'nin boyutu

$\boldsymbol{|QI|}$: Önerilen algoritmada QI sayısı

Kısaltmalar	Açıklama
ABD	: Amerika Birleşik Devletleri
API	: Application Programming Interface (Uygulama Programlama Arayüzü)
BDAS	: Berkeley Data Analytics Stack (Berkeley Veri Analitiği Yığını)
CCC	: Computing Community Consortium (Hesaplama Topluluğu Birliği)
CDH	: Cloudera's Distribution Including Apache Hadoop (Cloudera'nın Apache Hadoop'u İçeren Dağıtımı)
CDR	: Call Detail Records (Çağrı Detay Kayıtları)
DAG	: Directed Acyclic Graph (Yönlendirilmiş Çevrimsiz Çizge)
DARPA	: Defense Advanced Research Projects Agency (Savunma İleri Araştırma Projeleri Ajansı)
DBMS	: Database Management Systems (Veri Tabanı Yönetim Sistemleri)
DCaS	: Data Cleaning as a Service (Hizmet Olarak Veri Temizleme)
DOD	: The Department of Defense (Savunma Departmanı)
DOE	: The Department of Energy (Enerji Departmanı)
EM2	: Execution Migration Machine (Yürütme Göç Makinesi)
ETL	: Extract, Transform and Load (Çıkar, Dönüştür ve Yükle)
FN	: False Negatives (Yanlış Negatifler)
FP	: False Positives (Yanlış Pozitifler)
GFS	: Google File System (Google Dosya Sistemi)
HDFS	: Hadoop Distributed File System (Hadoop Dağıtık Dosya Sistemi)
HiveQL	: Hive Query Language (Hive Sorgu Dili)
HPCC	: High Performance Computing Cluster (Yüksek Performanslı Hesaplama Kümesi)
ICT	: Information and Communications Technologies (Bilgi ve İletişim Teknolojileri)
ID	: Identifier (Tanımlayıcı)
IDC	: International Data Corporation (Uluslararası Veri Kurumu)
IoT	: Internet of Things (Nesnelerin İnterneti)
IT	: Information Technology (Bilgi Teknolojisi)
JDBC	: Java Database Connectivity (Java Veri Tabanı Bağlantısı)
KL	: Kullback-Leibler
k-NN	: k-Nearest Neighbours (k-En Yakın Komşu)
NB	: Naive Bayes
NIH	: The National Institutes of Health (Ulusal Sağlık Enstitüleri)

NSA : Non-Sensitive Attribute (Hassas Olmayan Nitelik)

NSF : The National Science Foundation (Ulusal Bilim Vakfı)

ODBC : Open Database Connectivity (Açık Veri Tabanı Bağlantısı)

OMB : The Office of Management and Budget (Yönetim ve Bütçe Ofisi)

OneR : One Rule (Tek Kural)

OSTP : The Office of Science and Technology Policy (Bilim ve Teknoloji Politikaları Ofisi)

PPDP : Privacy Preserving Data Publishing (Gizlilik Korumalı Veri Yayınlama)

QI : Quasi-Identifier (Yarı Tanımlayıcı)

RDBMS : Relational Database Management System (İlişkisel Veri Tabanı Yönetim Sistemi)

REST : Representational State Transfer (Temsili Durum Transferi)

SA : Sensitive Attribute (Hassas Nitelik)

SAS : Statistical Analysis System (İstatistiksel Analiz Sistemi)

SDSS : Sloan Digital Sky Survey (Sloan Dijital Gökyüzü Araştırması)

SGI : Silicon Graphics Inc. (Silikon Grafik A.Ş.)

SVM : Support Vector Machine (Destek Vektör Makinesi)

TN : True Negatives (Doğru Negatifler)

TP : True Positives (Doğru Pozitifler)

UDF : User-Defined Function (Kullanıcı Tanımlı Fonksiyon)

USGS : The U.S. Geological Survey (ABD Jeolojik Araştırması)

VP : Voted Perceptron

XML : Extensible Markup Language (Genişletilebilir İşaretleme Dili)

3V : Volume, Velocity and Variety (Hacim, Hız ve Çeşitlilik)

4A : Acquisition/Access, Assembly/Organization, Analyze and Action/Decision (Edinme/Erişim, Birleştirme/Organizasyon, Analiz ve Eylem/Karar)

5V : Volume, Velocity, Variety, Value and Veracity (Hacim, Hız, Çeşitlilik, Değer ve Doğruluk)

ÖZET

DOKTORA TEZİ

BÜYÜK VERİDE ETKİN GİZLİLİK KORUMASI İÇİN YAZILIM TASARIMI

Can EYÜPOĞLU

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman : Prof. Dr. Ahmet SERTBAŞ

II. Danışman : Dr. Öğr. Üyesi Muhammed Ali AYDIN

Büyük veri konusuna son yıllarda giderek artan bir ilgi vardır. Büyük verinin ortaya çıkışı, verilerin paylaşılması ve işlenmesi için gerekli olan veri gizliliği için kullanılan koruma modelleri açısından yeni zorluklara yol açmaktadır. Yayınlanan veri setinin kullanılabilirliğini sürdürürken bireylerin hassas bilgilerini korumak, gizliliğin korunmasındaki en önemli zorluktur. Bu bağlamda, verilerin kimlik ifşası ve bağlantı saldırılarına karşı korunması için veri anonimleştirme yöntemleri kullanılmaktadır. Bu tez çalışmasında, kaos ve pertürbasyon temelli yeni bir veri anonimleştirme algoritması, büyük veride gizlilik ve kullanılabilirlik koruması için önerilmiştir. Ayrıca önerilen algoritma kullanılarak gizliliği korunan büyük veri setleri Hadoop üzerinde dağıtıklaştırılmıştır. Önerilen algoritmanın performansı Kullback-Leibler uzaklığı, olasılıksal anonimlik, sınıflandırma doğruluğu, F-ölçütü, yürütme süresi ve Impala sorguları açısından değerlendirilmiştir. Deneysel sonuçlar, önerilen algoritmanın, etkin ve aynı veri setini kullanan mevcut algoritmaların çoğundan üstün olduğunu göstermektedir. Verilerin karıştırılması için kaosun uygulanması sonucu ortaya çıkan bu başarılı algoritma, gizlilik korumalı veri madenciliği ve veri yayınlama alanlarında kullanılmada gelecek vadetmektedir.

Haziran 2018, 158 sayfa.

Anahtar kelimeler: Büyük Veri, Gizlilik Koruma, Kaos, Pertürbasyon, Hadoop.

SUMMARY

Ph.D. THESIS

SOFTWARE DESIGN FOR EFFICIENT PRIVACY PRESERVING IN BIG DATA

Can EYÜPOĞLU

İstanbul University

Institute of Graduate Studies in Science and Engineering

Department of Computer Engineering

Supervisor : Prof. Dr. Ahmet SERTBAŞ

Co-Supervisor : Assist. Prof. Dr. Muhammed Ali AYDIN

The topic of big data has attracted increasing interest in recent years. The emergence of big data leads to new difficulties in terms of protection models used for data privacy, which is of necessity for sharing and processing data. Protecting individuals' sensitive information while maintaining the usability of the data set published is the most important challenge in privacy preserving. In this regard, data anonymization methods are utilized in order to protect data against identity disclosure and linking attacks. In this study, a novel data anonymization algorithm based on chaos and perturbation has been proposed for privacy and utility preserving in big data. Besides, the big data sets which are privacy preserved using the proposed algorithm are distributed on Hadoop. The performance of the proposed algorithm is evaluated in terms of Kullback-Leibler divergence, probabilistic anonymity, classification accuracy, F-measure, execution time and Impala queries. The experimental results have shown that the proposed algorithm is efficient and superior to most of the existing algorithms using the same data set. Resulting from applying chaos to perturb data, such successful algorithm is promising to be used in privacy preserving data mining and data publishing.

June 2018, 158 pages.

Keywords: Big Data, Privacy Preserving, Chaos, Perturbation, Hadoop.

1. GİRİŞ

Şirketlerin depoladığı sosyal medya, Nesnelerin İnterneti (Internet of Things-IoT) ve multimedya gibi verilerin hacim ve detaylarındaki sürekli artış yapısal ya da yapısal olmayan formatta çok büyük bir veri akışı üretmektedir. Verilerin oluşturulması günümüzde rekor oranlarda gerçekleşmektedir (Villars ve diğ., 2011) ve bu da yaygın olarak bilinen bir akım olan büyük veriyi (big data) ortaya çıkarmıştır. Büyük veri üç yönüyle nitelendirilir: a) Veriler sayısızdır, b) Veriler sıradan ilişkisel veritabanlarında kategorize edilemezler ve c) Veriler hızlı olarak üretilir, tutulur ve işlenirler. Buna ek olarak büyük veri; sağlık, bilim, mühendislik, finans, iş dünyası ve neticede toplumu değiştirmektedir (Hashem ve diğ., 2015).

Büyük veri; akademi, bilimsel araştırma, bilişim teknolojileri endüstrisi, finans ve işletme alanlarında ilgi çeken bir konu haline gelmiştir (Khan ve diğ., 2014; Manyika ve diğ., 2011; Matturdi ve diğ., 2014). Dijital dünyada yaratılan veri miktarı son zamanlarda aşırı derecede artmıştır (McCune, 1998). Uluslararası Veri Kurumunun (International Data Corporation-IDC) yapmış olduğu araştırmaya göre 2011 yılında 1,8 zettabayt veri üretilmiştir ve bu veri miktarı her iki yılda bir ikiye katlanmaktadır (Tankard, 2012). Üretilen veri miktarının 2005 yılından 2020 yılına 300 kat artacağı tahmin edilmektedir (Gantz ve Reinsel, 2013). Sağlık endüstrisi, biyomedikal şirketler, reklamcılık sektörü, özel firmalar ve devlet kurumları tarafından büyük miktarlarda kişisel verinin toplanması, birleştirilmesi ve paylaşılması konusunda birçok yatırım yapılmaktadır (Bamford, 2012).

IoT, verilerin bir nesneler ağı vasıtasıyla taşındığı ve değiştirildiği yeni gelişen bir alandır. IoT; hasta izleme sistemi, trafik kontrol sistemi, envanter yönetimi sistemi gibi çeşitli alanlarda uygulamalara sahiptir. Bu uygulamaların tümünde kullanıcıların etkileşimleri ve hareketlilikleri ile ilgili kimlik ve hassas bilgileri korunmalıdır. Bu sayede bireylerin IoT'deki veri mahremiyetleri garanti altına alınmış olur (Nayahi ve Kavitha, 2017).

Büyük veri; çoğunlukla yetkisiz erişim ve yayınlamaya karşı koruma gerektiren kişiyi tanımlamak için kullanılan hassas bilgiler içermektedir (Ardagna ve Damiani, 2014; Labrinidis ve Jagadish, 2012; Matturdi ve diğ., 2014). Güvenlik açısından bakıldığında, büyük verideki en büyük zorluk bireylerin gizliliğinin korunmasıdır (Eyüpoğlu ve diğ., 2017, 2018; Lafuente, 2015). Bireylerin veri gizliliğini garanti altına almak, özel bilgileri dağıtık ortamlarda (Yuksel

ve diğ., 2017) ya da IoT'de (Henze ve diğ., 2016; Sicari ve diğ., 2015; Yao ve diğ., 2015) paylaşırken gizlilik yasalarına göre zorunludur (Nayahi ve Kavitha, 2017). Gizlilik korumalı veri madenciliği (Aggarwal ve Yu, 2008) ve gizlilik korumalı veri yayınlama yöntemleri (Fahad ve diğ., 2014; Fung ve diğ., 2010) verilerin paylaşılması ve yayınlanması için gereklidir.

Büyük veride, paylaşma veya yayınlamadan önce orijinal verileri modifiye etmek, bireylerin özel bilgilerinin yayınlanan veri setinde olmaması gerektiğinden veri sahibi için çok önemlidir. Hassas verilerin modifikasyonu veri kullanılabilirliğini azaltmaktadır. Aksine veri kullanılabilirliği, verilerin yararlılığını sürdürmek için uygun olmalıdır. Verilerin gizliliği ve kullanılabilirliği için olan bu veri modifikasyon süreci, gizlilik korumalı veri yayınlama olarak adlandırılır ve verileri yayınlarken orijinal veri setlerini korur. Orijinal bir veri seti dört çeşit nitelikten oluşmaktadır. Bireyleri doğrudan tanımlayan ve benzersiz değerlere sahip olan nitelikler, tanımlayıcı olarak adlandırılır. İsim, kimlik numarası ve telefon numarası tanımlayıcı niteliklere örnek olarak gösterilebilir. Hassas nitelikler, verileri yayınlarken ve paylaşırken gizlenmesi gereken niteliklerdir. Maaş ve hastalık hassas nitelik örnekleridir. Bir bireyin kimliğini açığa çıkarmak için kötü niyetli kişiler tarafından kullanılabilecek yaş ve cinsiyet gibi nitelikler yarı tanımlayıcı olarak adlandırılır. Bunların dışındaki diğer nitelikler ise hassas olmayan niteliklerdir. Orijinal veri seti, yayınlanmadan önce tanımlayıcılar silinerek ve yarı tanımlayıcılar modifiye edilerek anonimleştirilir. Böylece bireylerin gizliliği korunmuş olur (Xu ve diğ., 2014).

Gizliliği korumak için; genelleme, gizleme, anatomizasyon, permütasyon ve pertürbasyon olmak üzere beş tür anonimleştirme işlemi vardır. Genelleme işlemi, değerleri daha genel olanlarla değiştirmektedir. Gizleme işlemi, veri setlerinden belirli değerleri kaldırmaktadır (örneğin değerleri "*" gibi belirli karakterle değiştirme). Anatomizasyon işlemi; yarı tanımlayıcılar ve hassas nitelikler arasındaki ilişkileri ortadan kaldırmaktadır. Permütasyon işlemi, bir dizi veri kaydını gruplara ayırarak ve onların hassas değerlerini her grupta karıştırarak, bir yarı tanımlayıcı ve hassas nitelik arasındaki ilişkiyi kesmektedir. Pertürbasyon işlemi ise orijinal değerleri; yer değiştirerek, gürültü ekleyerek veya sentetik veriler oluşturarak yenileriyle değiştirmektedir. Bu anonimleştirme işlemleri, genel olarak bilgi kaybıyla temsil edilen veri kullanılabilirliğini azaltmaktadır. Diğer bir deyişle daha yüksek veri kullanılabilirliği daha düşük bilgi kaybı anlamına gelmektedir (Fung ve diğ., 2010; Xu ve diğ., 2014).

Literatürde şimdiye kadar yukarıda belirtilen işlemleri kullanan çeşitli çalışmalar yapılmıştır. Bu tez çalışmasında, veri kullanılabilirliği ve bilgi kaybı problemlerinin üstesinden gelmek için kaos ve pertürbasyon işlemini kullanan yeni bir anonimleştirme algoritması öne sürülmüştür. Çalışmanın literatüre katkısı; veri seti türünden bağımsız, hem sayısal hem de kategorik niteliklere uygulanabilen kapsamlı bir gizlilik korumalı veri yayınlama algoritmasının geliştirilmesidir. Önerilen algoritma, her bir yarı tanımlayıcı için benzersiz nitelik değerlerinin sıklığının analiz edilmesi, sıklık analizine uygun olarak kritik değerlerin belirlenmesi ve sadece belirlenen bu kritik değerler için pertürbasyon işleminin gerçekleştirilmesi nedeniyle daha yüksek veri kullanılabilirliğine sahiptir. Bu tez çalışmasının diğer bir önemli katkısı, sistemlerin rastlantısallığı için yaygın olarak kullanılan disiplinler arası bir teori olan kaosun, verilerin karıştırılmasındaki etkinliğini ortaya koymaktır. Bilindiği kadarıyla, büyük verinin gizliliğinin korunmasında kaosun bu çerçevede kullanımına ilişkin literatürde başka bir çalışma yoktur. Kaosun rastgeleleştirmedeki büyük başarısı, onun veri pertürbasyonundaki faydasının tez kapsamında araştırılmasına yol açmıştır. Bu tez çalışmasında önerilen algoritmanın farklı metrikler aracılığıyla performansını değerlendiren test sonuçları, algoritmanın önceki çalışmalarla karşılaştırıldığında etkili olduğunu göstermektedir.

Tez çalışmasının geri kalanı şu şekilde düzenlenmiştir: Genel Kısımlar bölümünde; büyük veri, büyük veride gizlilik koruması ve büyük verinin dağıtıklaştırılması ile ilgili genel bilgiler verilmiştir. Ayrıca tez konusu kapsamında literatürde bugüne kadar yapılmış olan çalışmalardan bahsedilmiştir. Malzeme ve Yöntem bölümünde; tez çalışması kapsamında kullanılan veri setleri, veri seti bölümleme yöntemi, performans metrikleri, sınıflandırma yöntemleri, önerilen etkin gizlilik koruma algoritması ve gizliliği korunmuş veri setlerinin dağıtıklaştırılması anlatılmıştır. Bulgular bölümünde; önerilen gizlilik koruma algoritmasının Kullback-Leibler uzaklığı, olasılıksal anonimlik, sınıflandırma doğruluğu, F-ölçütü, yürütme süresi ve Impala sorgu sonuçlarına yer verilmiştir. Son olarak Tartışma ve Sonuç bölümünde; önerilen algoritmanın literatüre katkısından, performans sonuçlarından ve mevcut algoritmalardan hangi açılardan daha etkili olduğundan bahsedilmiştir.

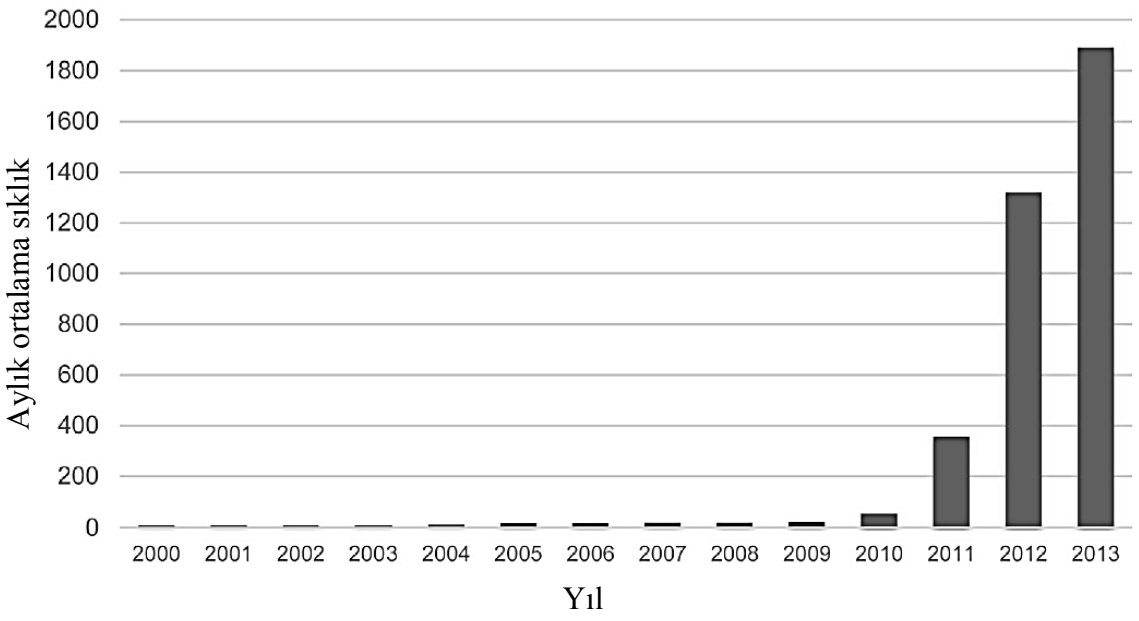
2. GENEL KISIMLAR

2.1. BÜYÜK VERİ

Bu bölümde; büyük verinin tanımı, özellikleri, önemi, kaynakları, sınıflandırılması, yönetimi, altyapısı ve zorlukları ile ilgili genel bilgiler verilmektedir.

2.1.1. Büyük Verinin Tanımı

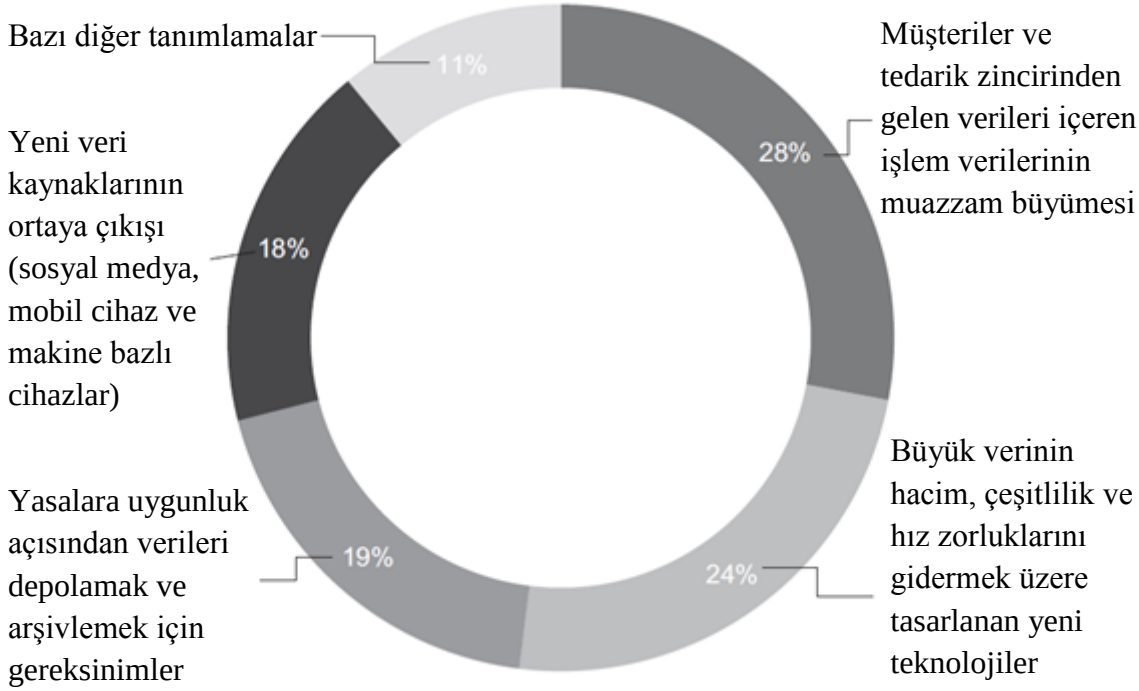
Büyük veri kavramı şu anda herkes tarafından bilinmesine rağmen kökeni belirsizdir. Diebold (2012) büyük veri teriminin John Mashey tarafından belirgin olarak düşünüldüğü ve 1990'ların ortalarında Silikon Grafik A.Ş. (Silicon Graphics Inc.-SGI) şirketinde öğle yemeği sohbeti sırasında ortaya çıktığını iddia etmektedir. Doksanlı yılların ortalarındaki kaynaklara rağmen, Şekil 2.1 (Gandomi ve Haider, 2015) büyük veri teriminin 2011 yılında yaygınlaştığını göstermektedir.



Şekil 2.1: ProQuest araştırma kütüphanesinde büyük veri terimini içeren belgelerin sıklık dağılımı.

Büyük veri tanımları hızla gelişmiştir ve sonucunda bazı karışıklıklar ortaya çıkmıştır. Bu karışıklıklar, 2012 yılının nisan ayında SAP adına Harris Interactive ajansı tarafından yürütülen dünya çapında 154 C-sınıfı yöneticinin çevrimiçi anketinden anlaşılmaktadır (SAP, 2012). Şekil 2.2 (Gandomi ve Haider, 2015), yöneticilerin büyük veriyi anlamada farklılaştıklarını

göstermektedir. Bazı tanımlar büyük verinin ne olduğuna odaklanmakta iken diğerleri büyük verinin ne yaptığını cevaplamaya çalışmaktadır.



Şekil 2.2: SAP'ın anketine dayanan büyük veri tanımları.

Büyük veri; geleneksel veri tabanı teknolojileri ile depolanması, işlenmesi ve analizi zor verilerin hacmindeki artışı belirtmek için kullanılan bir terimdir. Büyük verinin doğası veriyi tanımlamak ve yeni kavramlara dönüştürmek için önemli süreçler içermektedir. Büyük veri terimi Bilgi Teknolojisi (Information Technology-IT) ve işletme alanlarında nispeten yenidir. Ancak bazı araştırmacılar bu terimi daha önce literatürde kullanmıştır. Cox ve Ellsworth (1997) 1997 yılında büyük veriyi görselleştirme için büyük miktarda bilimsel veri olarak tanımlamıştır. Şu anda ise büyük verinin birçok tanımı vardır. Örnek olarak Manyika ve diğ. (2011) büyük veriyi; etkili bir şekilde depolanması, yönetilmesi ve işlenmesi teknoloji kapasitesinin ötesinde olan veri miktarı olarak tanımlamıştır. Bu arada Zikopoulos ve diğ. (2012) ve Berman (2013) büyük veriyi 3V: volume (hacim), velocity (hız) ve variety (çeşitlilik) ile karakterize etmiştir. 3V, büyük veriyi tanımlamak için ortak bir çerçeve olarak ortaya çıkmıştır (Chen ve diğ., 2012; Kwon ve diğ., 2014). Aslında hacim, hız ve çeşitlilik terimleri ilk olarak büyük veri zorluklarını tanımlamak için Gartner tarafından öne sürülmüştür. Gartner büyük veriyi benzer terimlerle; gelişmiş kavrama, karar verme ve süreç otomasyonuna olanak sağlayan uygun maliyetli ve yenilikçi bilgi işleme yapıları gerektiren yüksek hacimli, yüksek hızlı ve/veya yüksek çeşitlilikli

bilgi varlıkları olarak tanımlamıştır (Gartner IT Glossary, 2018). IDC büyük veri teknolojilerini; yüksek hızda yakalama, keşif ve analiz yaparak, geniş kapsamlı çok büyük veri hacimlerinden ekonomik olarak değer çıkarmak için tasarlanan yeni nesil teknolojiler ve mimariler olarak ifade etmiştir (Hashem ve diğ., 2015). TechAmerica Vakfının büyük veri tanımı ise şöyledir: Büyük veri; bilginin yakalanması, depolanması, dağıtımı, yönetimi ve analizi için gelişmiş teknikler ve teknolojiler gerektiren büyük hacimlerdeki yüksek hızlı, karmaşık ve değişken verileri tanımlayan bir terimdir (TechAmerica Foundation's Federal Big Data Commission, 2012).

2.1.2. Büyük Verinin Özellikleri

Büyük veri nedir sorusu göz önünde bulundurulduğunda şüphesiz akla gelen ilk özellik boyuttur. Boyutun yanı sıra büyük verinin diğer özellikleri son zamanlarda ortaya çıkmıştır. Laney (2001); hacim, hız ve çeşitliliği (3V) Şekil 2.3'teki (Mehmood ve diğ., 2016) gibi veri yönetimindeki zorlukların üç boyutu olarak göstermiştir. Demchenko ve diğ. (2013) ise literatürde var olan diğer çalışmaları inceleyerek ve analiz ederek büyük veriyi hacim, hız ve çeşitliliğe ek olarak değer (value) ve doğruluk (veracity) özellikleri ile tanımlamıştır. Bu özellikler literatürde Şekil 2.4'te (Demchenko ve diğ., 2013) görüldüğü üzere büyük verinin 5V'si olarak anılmaktadır (Demchenko ve diğ., 2013, 2014; Sharma ve Mangat, 2015; Terzi ve diğ., 2015). Bazı kaynaklarda ise değişkenlik (variability) ve karmaşıklık (complexity) büyük verinin diğer boyutları olarak incelenmiştir (Bhadani ve Jothimani, 2016; Gandomi ve Haider, 2015).

Şekil 2.3: Büyük verinin 3V'si.

Hacim: Üretilen ve toplanan verilerin büyüklüğünü ifade etmektedir. Veri boyutu terabaytlardan petabaytlara doğru çok hızlı bir oranda artmaktadır. Günümüzde, donanım teknolojisi eksikliği sebebiyle yakalanamayan ve depolanamayan veriler vardır. Bu verilerin yakalanması ve depolanması depolama kapasitelerinin artmasıyla gelecekte mümkün olacaktır. Büyük verinin hacim bazında sınıflandırılması üretilen verinin ve zamanın türüne göre görecelidir. Buna ek olarak genellikle çeşitlilik olarak adlandırılan veri türü büyük veriyi tanımlamaktadır. Örneğin aynı hacimde olan metin ve video verileri farklı veri yönetimi teknolojilerini gerektirebilir (Bhadani ve Jothimani, 2016; Gandomi ve Haider, 2015).

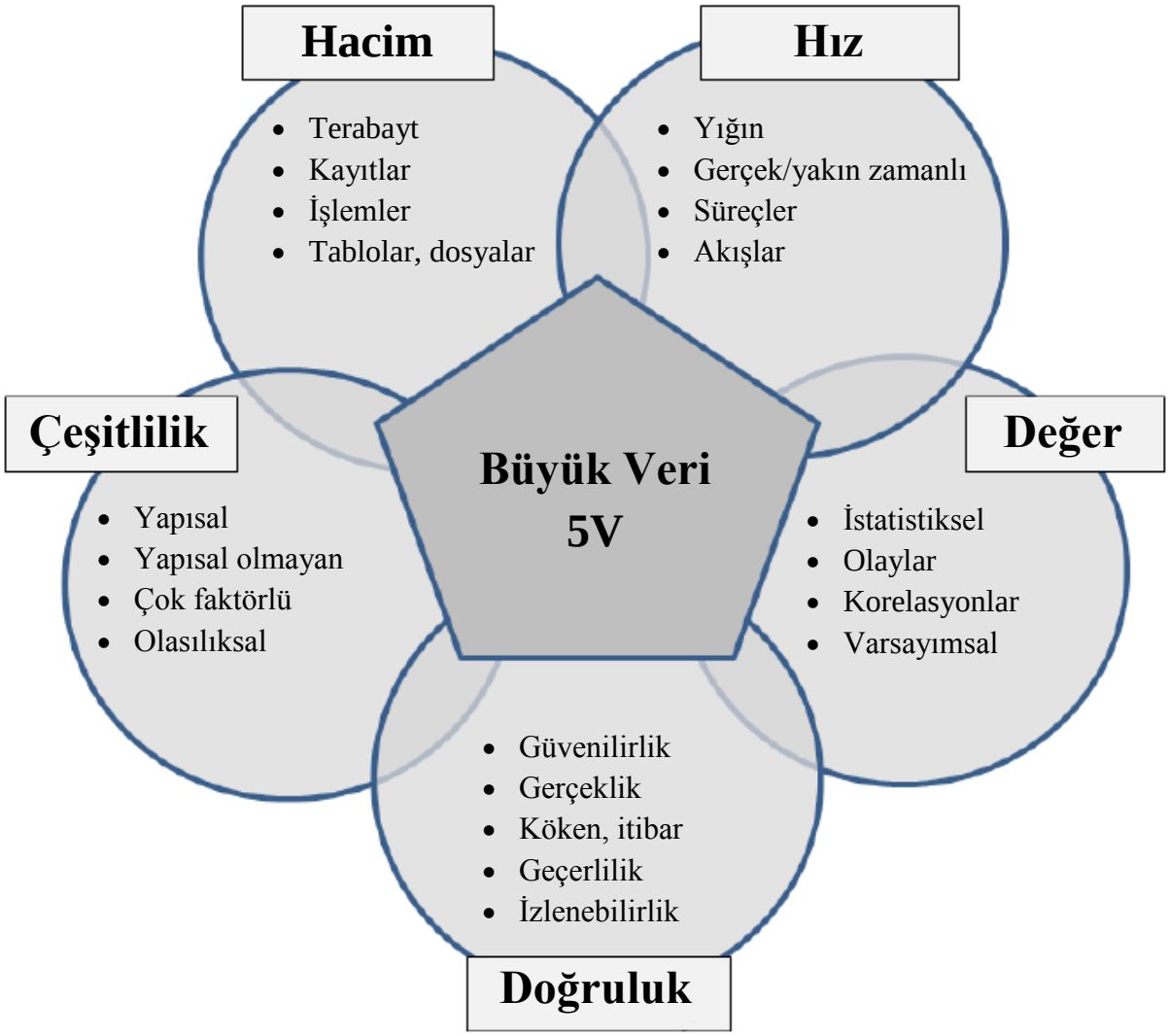
Başka bir deyişle hacim, farklı kaynaklardan üretilen ve büyümeye devam eden tüm veri türlerinin miktarını ifade etmektedir. Çok büyük miktarlarda veriyi bir araya getirmenin yararı, veri analizi yoluyla gizli bilgi ve örüntülerin oluşturulmasını sağlamaktır. Laurila ve diğ. (2012), akıllı mobil cihazlardan gelen boylamsal verilerden oluşan benzersiz bir koleksiyon oluşturmuştur ve bu koleksiyonu araştırma topluluğu için kullanılabilir hale getirmiştir. Söz konusu girişim Nokia tarafından harekete geçirilmiş ve mobil veri zorluğu olarak adlandırılmıştır. Boylamsal veri toplama çok büyük çabalar ve temel yatırımlar

gerektirmektedir. Bununla birlikte bu mobil veri zorluğu, insan davranış örüntülerinin tahmin edilebilirliğinin ya da insan hareketliliği ve görselleştirme tekniklerine dayanan karmaşık verilerin paylaşılması için olan araçların incelenmesindekine benzer ilginç bir sonuç üretmiştir (Hashem ve diğ., 2015).

Hız: Veri üretim oranını ifade etmektedir. Geleneksel veri analitiği günlük, haftalık veya aylık periyodik güncellemelere dayanmaktadır. Büyük veri, artan veri oluşturma oranıyla bilgiye dayalı kararlar vermek için gerçek veya yakın gerçek zamanlı olarak işlenmeli ve analiz edilmelidir. Zamanın rolü burada çok kritiktir (Bhadani ve Jothimani, 2016; Gandomi ve Haider, 2015). Perakende, telekomünikasyon ve finans dâhil olmak üzere az sayıda alan yüksek frekanslı veri üretmektedir. Demografi, coğrafi konum ve işlem geçmişi gibi mobil uygulamalar aracılığıyla oluşturulan veriler, müşterilere kişiselleştirilmiş hizmetler sunmak için gerçek zamanlı olarak kullanılabilir. Bu, müşterileri elde tutmanın yanı sıra servis kalitesini de artırmaya yardımcı olmaktadır (Bhadani ve Jothimani, 2016). Bir başka ifadeyle hız, veri aktarım sürati anlamına gelmektedir. Verilerin içeriği birbirini tamamlayan veri koleksiyonlarının birleşmesi, önceden arşivlenmiş verilerin veya eski koleksiyonların ortaya çıkması ve birden fazla kaynaktan gelen akış verilerinin toplanması sebebiyle sürekli olarak değişmektedir (Berman, 2013; Hashem ve diğ., 2015).

Çeşitlilik: Üretilen ve yakalanan farklı veri türlerini ifade etmektedir. Veri türleri; yapısal, yarı yapısal ve yapısal olmayan olarak kategorize edilirler (Bhadani ve Jothimani, 2016; Gandomi ve Haider, 2015). Önceden tanımlanmış bir veri modeli kullanılarak organize edilebilen veriler, yapısal veri olarak bilinmektedir. İlişkisel veri tabanlarındaki ve Excel'deki tablo verileri, yapısal veri örnekleridir ve mevcut tüm verilerin yalnızca %5'ini oluşturmaktadır (Cukier, 2010). Yapısal olmayan veriler önceden tanımlanmış bir model kullanılarak düzenlenemezler. Video, metin ve ses yapısal olmayan verilere örnek olarak gösterilebilir. Yarı yapısal veriler yapısal ve yapısal olmayan veri türleri arasında yer almaktadır. Genişletilebilir İşaretleme Dili (Extensible Markup Language-XML) bu kategoriye girmektedir (Bhadani ve Jothimani, 2016). Veriler sensörler, akıllı telefonlar veya sosyal ağlar gibi birçok kaynaktan üretilebilir. Mobil uygulamalardan elde edilen verilerin çoğu yapısal olmayan formattadır. Örneğin metin mesajları, çevrimiçi oyunlar, bloglar ve sosyal medya, mobil cihazlar ve sensörler aracılığıyla farklı türlerde yapısal olmayan veriler oluşturmaktadır. Internet kullanıcıları da yapısal ve

yapısal olmayan aşırı derecede çeşitli veri setleri üretmektedir (Hashem ve diğ., 2015; O'Leary, 2013).



Şekil 2.4: Büyük verinin 5V'si.

Değer: Farklı türlerde veri içeren ve hızlı üretimin olduğu büyük veri setlerinden muazzam gizli değerler keşfetme sürecini ifade etmektedir. Ayrıca değer, büyük verinin en önemli yönüdür (Chen ve diğ., 2014; Hashem ve diğ., 2015). Oracle; değeri, büyük veriyi tanımlayan bir özellik olarak göstermiştir. Oracle'ın tanımına göre büyük veri, genellikle "düşük değer yoğunluğu" ile karakterize edilmektedir. Yani orijinal formda alınan veriler genellikle hacmine göre düşük değere sahiptir. Ancak büyük hacimlerdeki bu verilerin analiz edilmesiyle yüksek değer elde edilebilmektedir (Gandomi ve Haider, 2015). Örnek olarak web sitelerindeki loglar

iş değeri elde etmek için ilk formlarında kullanılamazlar. Bu veriler müşteri davranışlarını tahmin etmek için analiz edilmek zorundadırlar (Bhadani ve Jothimani, 2016).

Doğruluk: Veri kaynakları ile ilişkili güvensizliği ifade etmektedir. IBM; doğruluğu, bazı veri kaynaklarının doğasında var olan güvensizliği temsil etmek için öne sürmüştür. Örneğin sosyal medyadaki müşteri görüşleri insan takdiri gerektirmesi sebebiyle doğası gereği belirsizdir. Yine de değerli bilgiler içermektedir. Bu nedenle kesin olmayan verilerle başa çıkma ihtiyacı büyük verinin bir başka yönüdür. Ayrıca bu ihtiyaç, belirsiz verilerin yönetimi ve madenciliği için geliştirilen araçlar ve analitik kullanılarak ele alınmaktadır (Gandomi ve Haider, 2015). Twitter, Facebook, vb. sosyal medya verilerini kullanarak duygu analizi de belirsizliğe tabidir. Güvenilir verilerin belirsiz ve kesin olmayan verilerden ayrılmasına ve verilerle ilgili belirsizliğin yönetilmesine ihtiyaç vardır (Bhadani ve Jothimani, 2016).

Değişkenlik: Veri akış oranlarındaki değişime ifade etmektedir ve SAS (Statistical Analysis System-İstatistiksel Analiz Sistemi) tarafından öne sürülmüştür. Büyük veri hızı, çoğu zaman tutarlı değildir ve periyodik iniş ve çıkışlara sahiptir (Gandomi ve Haider, 2015). Veri akışları artan hızlara ve veri çeşitliliklerine ek olarak periyodik zirvelerle oldukça tutarsız olabilmektedir. Günlük, mevsimsel ve olay tetiklemeli en yüksek veri yüklerini yönetmek zorlu olabilmektedir. Yapısal olmayan verilerle bu daha da zordur (SAS, 2018).

Karmaşıklık: Karmaşıklık da SAS tarafından önerilmiştir ve büyük verinin çok sayıda kaynak aracılığıyla üretildiği gerçeğini ifade etmektedir. Buradaki kritik zorluk farklı kaynaklardan alınan verileri birbirine bağlama, eşleştirme, temizleme ve dönüştürme ihtiyacıdır (Gandomi ve Haider, 2015). Bununla birlikte ilişkilerin, hiyerarşilerin ve çoklu veri bağlantılarının birleştirilmesi ve ilişkilendirilmesi gerekir veya verileriniz hızla kontrolün dışına çıkabilir (SAS, 2018).

2.1.3. Büyük Verinin Önemi

Amerika Birleşik Devletleri'nde (ABD) Beyaz Saray (White House), Yönetim ve Bütçe Ofisi (The Office of Management and Budget-OMB) ve Bilim ve Teknoloji Politikaları Ofisi (The Office of Science and Technology Policy-OSTP) 2010 yılının Ağustos ayında büyük verinin sağlık ve güvenlik için ulusal bir zorluk ve öncelik olduğunu duyurmuştur (American Institute of Physics, 2010). Ulusal Bilim Vakfı (The National Science Foundation-NSF), Ulusal Sağlık Enstitüleri (The National Institutes of Health-NIH), ABD Jeolojik Araştırması (The U.S.

Geological Survey-USGS), Savunma Departmanı (The Department of Defense-DOD), Enerji Departmanı (The Department of Energy-DOE) ve Savunma İleri Araştırma Projeleri Ajansı (Defense Advanced Research Projects Agency-DARPA) 2012 yılının Mart ayında ortak bir Ar-Ge girişimi gerçekleştirdiklerini ve bu girişim ile yeni büyük veri araçları ve teknikleri geliştirmek için 200 milyon dolardan fazla yatırım yapacaklarını duyurmuştur. Bu kuruluşların amacı, çok büyük boyutlardaki bilgilerin işlenmesi ve bu bilgiler üzerinde madencilik yapılması için gerekli olan teknolojilerin anlaşılmasıdır. Ayrıca bu bilgilerin sağlık, enerji, savunma, eğitim ve araştırma gibi diğer bilimsel alanlardaki ulusal hedefleri karşılayacak şekilde uygulanması hedeflenmektedir (Kaisler ve diğ., 2013; Kang, 2011).

Amerika Birleşik Devletleri'nin asıl üzerinde durduğu konu büyük verinin nasıl değer yarattığıdır. Değer, işlemeye uygun bilgileri geliştirmek için verileri analiz etme yeteneğinden ortaya çıkmaktadır. Büyük veri beş genel yol ile kuruluşlar için değer yaratmayı desteklemektedir: 1) İşletme ve fonksiyonel analiz için büyük veriyi açık hale getirerek şeffaflık yaratma (kalite, düşük maliyet, pazarlama süresini kısaltma vb.), 2) Belirli pazar programları gibi kararları veya yaklaşımları test edebilen bireysel konumlarda deneysel analizi destekleme, 3) Pazar bölümlemesini müşteri bilgilerine dayalı olarak daha dar seviyelerde tanımlamada yardımcı olma, 4) Müşterilerden ve gömülü algılayıcılardan gelen veri setlerine uygulanan karmaşık analitiğe dayalı gerçek zamanlı analiz ve kararları destekleme, 5) Müşteri tepkilerini gösteren gömülü ürün algılayıcılarına dayalı ürünlerde bilgisayar destekli yeniliklerin kolaylaştırılması (Kaisler ve diğ., 2013).

Amerikan hükümeti, büyük veri kullanıcılarının daha başarılı, daha üretken ve birçok endüstride farklı etkilere sahip olacağını düşünmektedir. Fakat hükümetin büyük veri ile tam anlamıyla çalışabilecek eğitimli personel ve büyük veri araç eksikliği gibi temel kaygıları vardır. Bazıları genel dizilerin, sosyal medya etkileşimlerinin, sağlık kayıtlarının, telefon loglarının ve devlet kayıtlarının analizinin daha iyi araçlar ve hizmetler yaratmayacağını, bunun aksine yeni gizlilik saldırılarına ve istenmeyen pazarlamalara sebep olabileceğini savunmaktadır (Boyd ve Crawford, 2011). Bu çelişkili kaygılar büyük veri ile nasıl başa çıkılacağı hakkında farklı görüşlere neden olmaktadır (Kaisler ve diğ., 2013).

Tıbbi alandan bir örnek, büyük veri ve analitiğin neden ve nasıl yararlı olabileceğini göstermektedir. Fox (2011), bir hastanın tıbbi kayıtlarındaki güncel verilerin ve mevcut sağlık durumunun sağlık ve hastalık yönetim programlarında hasta katılımını planlamak ve

hedeflemek için nasıl kullanıldığını açıklamaktadır. Fox, doktorların ve sigorta şirketlerinin hastalıktan ziyade hastaları anlaması gerektiğini söylemektedir. Bunu yapmak için bir hastanın tıbbi durumundan başka daha fazla verinin toplanması ve analiz edilmesi gereklidir. Davranış ve sağlığı ilişkilendiren halka açık kritik sosyal ve davranışsal veriler bir hastanın katılma isteğini, sözleşme düzeyini ve uygunluğunu etkilemektedir. Dolayısıyla bu verileri kullanan programlar hastalık sürecinin hedeflenmesini, sürdürülmesini ve hastalığın tedavi edilmesini daha iyi planlayabilirler. Ayrıca bu programlar kronik hastalığı olan hastaların davranışlarını pozitif yönde etkilemeye çalışan doktorlara ve vaka yöneticilerine yardımcı olabilecek tahminleme modellerinden yararlanırlar (Kaisler ve diğ., 2013).

2.1.4. Büyük Veri Kaynakları

En hızlı artış gösteren veri türü yapısal olmayan veridir. Bu veri türü insan bilgisi olarak karakterize edilir. Yüksek çözünürlüklü videolar, filmler, fotoğraflar, bilimsel simülasyonlar, finansal işlemler, telefon kayıtları, genomik veri setleri, sismik görüntüler, coğrafi haritalar, e-postalar, tweetler, Facebook verileri, çağrı merkezi konuşmaları, cep telefonu aramaları, web sitesi tıklamaları, belgeler, sensör verileri, telemetri, tıbbi kayıtlar ve görüntüler, iklim bilimi ve hava durumu kayıtları, log dosyaları ve metinler yapısal olmayan verilere örnek olarak gösterilebilir (Douglas, 2012). Yapısal olmayan bilgiler, kurumlardaki tüm verilerin %70'inden fazlasını oluşturmaktadır (Holzinger ve diğ., 2013). Çoğunlukla sosyal medyadan gelen bu veriler, dünya çapındaki verilerin %80'ini meydana getirir ve büyük verinin %90'ını oluşturur. Şu anda bilgi teknolojileri yöneticilerinin %84'ü yapısal olmayan verileri işlemektedir ve bu yüzdenin yakın gelecekte %44 oranında düşmesi beklenmektedir (Douglas, 2012). Çoğu yapısal olmayan veri modellenmemiştir, rastgeledir ve analiz edilmesi zordur. Bu tür verileri yönetmek için uygun stratejiler birçok kuruluşta geliştirilmelidir (Khan ve diğ., 2014). Endüstriyel Gelişme ve EMC Kurumlarına göre 2020 yılında üretilen veri miktarı 2009 yılındakine göre 44 kat daha fazla yani 40 zettabayt (ZB) olacaktır. Bu artış oranının yıllık olarak %50 ile %60 arasında kalması beklenmektedir (IDC, 2018). Tablo 2.1 (Statistic Brain Research Institure, 2018) verilerin çeşitli şirketlerde hızlı bir şekilde üretildiğini göstermektedir.

Tablo 2.1: Yapısal olmayan verilerin hızlı büyümesi.

Kaynak	Yıl	Üretim
Facebook	2017	1) Toplam aylık aktif kullanıcı sayısı 1,7 milyardır. 2) Toplam Facebook sayfası sayısı 78,2 milyondur. 3) Günde yüklenen ortalama fotoğraf sayısı 205'dir. 4) Facebook'ta her 20 dakika bir paylaşılan bağlantı sayısı 1 milyon, arkadaş isteği 2 milyon ve gönderilen mesaj sayısı 3 milyondur.
Youtube	2016	1) YouTube'u kullanan toplam kişi sayısı 1,3 milyardır. 2) Her dakika 300 saatlik video yüklenmektedir. 3) Her gün görüntülenen video sayısı yaklaşık 5 milyardır. 4) Her ay 3,2 milyar saat video izlenmektedir.
Twitter	2016	1) Toplam kayıtlı kullanıcı sayısı yaklaşık 700 milyondur. 2) Her gün yeni kaydolan kullanıcı sayısı 135 bindir. 3) Günde ortalama tweet sayısı 58 milyondur. 4) Her saniyede gerçekleşen tweet sayısı 9100'dür.
Google	2017	1) Günlük ortalama arama sayısı 9 milyardır. 2) Yıllık Google arama sayısı yaklaşık 3,3 trilyondur.
Google+	2017	1) Toplam aktif üye sayısı 395 milyondur. 2) Aylık toplam yeni ziyaret sayısı 34 milyondur.
Apple	2017	1) Toplam App Store indirme sayısı 37,2 milyardır. 2) App Store'da bulunan toplam uygulama sayısı 1,3 milyondur.
Instagram	2017	1) Toplam kullanıcı sayısı 715 milyondur. 2) Instagram'da paylaşılan toplam fotoğraf sayısı 34,7 milyardır. 3) Günlük ortalama yüklenen fotoğraf sayısı 52 milyondur. 4) Günlük ortalama beğeni sayısı 1,6 milyardır.
LinkedIn	2017	1) Toplam kullanıcı sayısı 313 milyondur. 2) Aylık yeni ziyaretçi sayısı 187 milyondur. 3) LinkedIn üzerinden görüşme yapan kişi sayısı 122 milyondur.
Tumblr	2016	1) Toplam kullanıcı sayısı 425 milyondur. 2) Toplam Tumblr bloğu sayısı 219 milyondur. 3) Aylık ziyaretçi sayısı 199 milyondur. 4) Toplam post sayısı 101 milyardır.

İçeriklerin endüstriler tarafından dijitalleştirilmesi yeni veri kaynağıdır. Teknolojideki gelişmeler aynı zamanda yüksek veri üretme oranına da neden olmaktadır. Örneğin astronomideki en büyük araştırmalardan biri olan Sloan Dijital Gökyüzü Araştırması (Sloan Digital Sky Survey-SDSS), ilk (2000-2005) ve ikinci araştırmasında (2005-2008) toplam 25

TB veri kaydetmiştir. Teleskopun çözünürlüğündeki ilerlemelerle birlikte, üçüncü anketin sonunda (2008-2014) toplanan veri miktarı 100 TB olmuştur. Akıllı aletlerin kullanımı diğer bir büyük veri kaynağıdır. Enerji sektöründeki akıllı sayaçlar, daha önce yapılan aylık okumalara kıyasla her 15 dakikada bir elektrik kullanım ölçümünü kaydetmektedir. Sosyal medyaya ek olarak IoT şu anda yeni veri kaynağı haline gelmiştir. Veriler, IoT temel alınarak geliştirilen akıllı şehirlerin tarım, sanayi, tıbbi bakım vb. alanlarından alınabilirler. Tablo 2.2 farklı sektörlerde üretilen çeşitli veri türlerini özetlemektedir (Bhadani ve Jothimani, 2016).

Tablo 2.2: Farklı veri kaynakları.

Sektör	**Üretilen veri**	**Kullanım**
Astronomi	Yıldızların, uyduların vb. hareketleri	Asteroitlerin ve uyduların faaliyetlerini izlemek
Finans	Video içeriği, ses, twitter ve haber raporu yoluyla haber içeriği	Ticaret kararları vermek
Sağlık Hizmeti	Elektronik tıbbi kayıtlar ve görüntüler	Kısa vadeli halk sağlığı izleme ve uzun vadeli epidemiyolojik araştırma programlarına yardımcı olmak
Nesnelerin İnterneti	Sensör verileri	Akıllı şehirlerdeki çeşitli faaliyetleri izlemek
Yaşam Bilimleri	Gen dizileri	Genetik çeşitlilik ve potansiyel tedavi etkinliğini analiz etmek
Medya/Eğlence	İçerik ve kullanıcı görüntüleme davranışı	Daha fazla izleyici yakalamak
Sosyal Medya	Blog mesajları, tweetler, sosyal ağ siteleri, günlük log ayrıntıları	Müşteri davranış modelini analiz etmek
Telekomünikasyon	Çağrı Detay Kayıtları (Call Detail Records-CDR)	Müşterinin bir servis sağlayıcısından diğerine geçişini yönetmek
Taşımacılık, Lojistik, Perakende, Kamu Hizmetleri	Filo alıcı vericileri, RFID etiket okuyucuları ve akıllı sayaçlardan üretilen sensör verileri	İşlemleri optimize etmek
Video Gözetimi	CCTV'den IPTV kameralarına ve kayıt sistemlerine yapılan kayıtlar	Hizmet geliştirme ve güvenlik için davranış modellerini analiz etmek

2.1.5. Büyük Verinin Sınıflandırılması

Büyük veri, özelliklerinin daha iyi anlaşılabilmesi için farklı kategoriler halinde sınıflandırılabilir. Tablo 2.3'te görüldüğü üzere büyük veri on iki kategoriye ayrılır: veri tipi, veri kaynağı, veri kullanımı, veri tüketicisi, analiz tipi, işleme metodolojisi, işleme yöntemi,

veri deposu, veri sıklığı, veri hazırlama ve donanım (Hashem ve diğ., 2015; Mysore ve diğ., 2013; Terzi ve diğ., 2015).

Tablo 2.3: Büyük veri sınıflandırması.

Kategori	Alt kategori
Veri tipi	Meta veri, ana veri, tarihsel veri, işlem verisi
İçerik formatı	Yapısal, yarı yapısal, yapısal olmayan
Veri kaynağı	Web ve sosyal medya, IoT, makine tarafından üretilen, insan tarafından üretilen, iç veri kaynakları, işlem verileri, biyometrik veri, veri sağlayıcılarından gelen, veri yaratıcılarından gelen
Veri kullanımı	Sanayi, akademi, devlet, araştırma merkezleri
Veri tüketicisi	İnsani, iş süreci, kurumsal uygulamalar, veri depoları
Analiz tipi	Gerçek zamanlı, yığın, etkileşimli, hibrit
İşleme metodolojisi	Tahmini analiz, analitik, modelleme, sorgu ve raporlama
İşleme yöntemi	Yüksek performanslı hesaplama, dağıtık, paralel, küme, grid
Veri deposu	İlişkisel, graf, belge odaklı, sütun odaklı, anahtar değeri
Veri sıklığı	İsteğe bağlı, sürekli, gerçek zamanlı, zaman serisi
Veri hazırlama	Temizleme, normalizasyon, dönüştürme
Donanım	Ticari donanım, son teknoloji donanım

2.1.6. Büyük Verinin Yönetimi

Temelde veri işleme, son kullanıcılar için yeni bilgi üretmede verilerin toplanması, işlenmesi ve yönetimi olarak görülmektedir. Zamanla, ana zorluklar yüksek verimli verilerin depolanması, taşınması ve işlenmesi ile ilgili hale gelmiştir. Büyük veri zorluklarından farklı olarak anlam karmaşıklığı (ambiguity), belirsizlik (uncertainty) ve çeşitlilik (variety) de göz önünde bulundurulmalıdır (Krishnan, 2013). Sonuç olarak bu gereklilikler, verilerin temizlendiği, etiketlendiği, sınıflandırıldığı ve formatlandığı ek bir adım olduğunu göstermektedir (Agrawal ve diğ., 2012; Krishnan, 2013). Karmasphere şu anda büyük veri analizini dört adıma ayırmıştır: 1) Edinme veya Erişim, 2) Birleştirme veya Organizasyon, 3) Analiz ve 4) Eylem veya Karar. Dolayısıyla bu basamaklar "4A" olarak adlandırılır. Hesaplama Topluluğu Birliği (Computing Community Consortium-CCC) (Krishnan, 2013)'deki gibi

organizasyon adımını bir Çıkarma/Temizleme ve bir Entegrasyon adımına bölmektedir (Emani ve diğ., 2015).

Edinme: Büyük veri mimarisi, çeşitli kaynaklardan (web, DBMS (OLTP), NoSQL, HDFS) yüksek hızlı veri elde etmek ve farklı erişim protokolleriyle başa çıkmak zorundadır. Burada, sadece yararlı olabilecek verileri ya da daha düşük belirsizlik derecesine sahip işlenmemiş (raw) verileri saklamak için bir filtre kurulabilir. Bazı uygulamalarda veri üretme koşulları önemlidir. Dolayısıyla meta verileri yakalamak ve bunları ilgili verilerle depolamak için daha ileri analiz yapma etkileyici olabilir (Agrawal ve diğ., 2012).

Organizasyon: Bu noktada, büyük veri mimarisi, çeşitli veri formatlarının (metin formatları, sıkıştırılmış dosyalar, ayrılmış veriler vs.) üstesinden gelmek zorundadır. Ayrıca bu formatları ayrıştırabilir ve adlandırılmış varlıklar, aralarındaki ilişkiler gibi gerçek bilgileri çıkarabilir olmalıdır. Verilerin temizlenmiş, hesaplanabilir bir moda sokulmuş, yapılandırılmış veya yarı yapılandırılmış, entegre edilmiş ve doğru yerde saklanmış olması gereken nokta burasıdır (mevcut veri ambarları (data warehouse), veri ambarlarının erişim katmanları (data marts), Operasyonel Veri Deposu (Operational Data Store), Kompleks Olay İşleme Motoru (Complex Event Processing engine), NoSQL veri tabanı) (Agrawal ve diğ., 2012). Bu nedenle bir çeşit ETL (Extract (Çıkar), Transform (Dönüştür), Load (Yükle)) yapılmak zorundadır. Büyük veri mimarisinde başarılı temizleme tam anlamıyla garanti edilemez. Aslında büyük verinin hacmi, hızı, çeşitliliği ve değişkenliği verinin tamamen temizlenmesi için gerekli zamanın ayrılmasına engel olabilir (Emani ve diğ., 2015).

Analiz: Burada, yeni anlamlar ve sezgiler bulmak için sorgular çalıştırılmakta, modelleme yapılmakta ve algoritmalar oluşturulmaktadır. Veri madenciliği, entegre, temizlenmiş ve güvenilir verilere ihtiyaç duymaktadır. Aynı zamanda veri madenciliği; verilerin kalitesini ve güvenilirliğini artırmak, semantiği anlamak ve akıllı sorgulama fonksiyonları sağlamak için kullanılabilir (Agrawal ve diğ., 2012).

Karar: Değerli kararlar alabilmek, analiz sonuçlarını etkili bir şekilde yorumlayabilmek demektir. Dolayısıyla kullanıcı için çıktıları/sonuçları anlamak ve doğrulamak çok önemlidir (Agrawal ve diğ., 2012). Buna ek olarak verinin kaynağı (her bir sonucun nasıl ortaya çıktığını açıklayan ek bilgi), kullanıcının elde ettiği şeyi anlamasına yardımcı olmak için sağlanmalıdır (Emani ve diğ., 2015).

Gizlilik; hacim, hız, çeşitlilik ve değişkenliğin dışında büyük veri mimarisindeki akışı etkileyen diğer bir önemli noktadır. Hillard (2012) gizliliğin büyük veri tanımında çok önemli bir yere sahip olduğunu düşünmektedir. Gizlilik; verilerin oluşturulmasında ve analizinde sorunlara neden olabilmektedir (Davis, 2012). Çünkü verilerin birleştirilmesi ya da ilişkilendirilmesi istenildiğinde özel verilere erişilmek zorunda kalınabilir. Gizlilik, ayrıca veri tabanı temizlenmesinde de tutarsızlıklara neden olabilir (Emani ve diğ., 2015).

Özetlemek gerekirse büyük veri ile başa çıkmak için doğrusal ölçeklenebilir, yüksek verimli, hata toleranslı, otomatik olarak kurtarılabilir, yüksek düzeyde paralellikli ve dağıtık veri işleme yapabilen bir altyapıya sahip olmak gerekmektedir (Krishnan, 2013). Büyük veri yönetiminde veri entegrasyonu (erişim, ayrıştırma, normalleştirme, standartlaştırma, entegrasyon, temizleme, çıkarma, eşleme, sınıflandırma, maskeleme ve iletme) büyük veri projelerinin %80'ini temsil etmektedir (Reeve, 2013).

2.1.6.1. Büyük Veri Yönetim Sistemleri

UC Berkeley'deki AMPLab (2018), büyük verilerle başa çıkmak için kurulmuştur. AMPLab yazılım bileşenlerinin entegrasyonu ile büyük verileri kullanmak ve anlamak için açık kaynaklı bir yazılım yığını olan Berkeley Veri Analitiği Yığınını (Berkeley Data Analytics Stack-BDAS) geliştirmiştir. Goethe Üniversitesi Frankfurt da veri analitiğindeki araştırma faaliyetlerini bilgi sistemleri ve bilgisayar bilimi perspektifinden birleştirmek için bir büyük veri analitiği araştırma laboratuvarı kurmuştur. Yaklaşımları; veri yönetimi teknolojileri ve analitik arasındaki disiplinler arası bağlantıya dayanmaktadır. UC Irvine'deki ASTERIX projesi büyük verileri işlemek için doğru bileşen ve katmanların seçimi ile ilgili soruna yönelik bir platform geliştirmiştir (Fang ve diğ., 2015). MIT'nin büyük veri laboratuvarı olan CSAIL (2018), yeni nesil veri zorluklarını çözme konusunda teknolojileri tanımlamak ve geliştirmek için kurulmuştur. CSAIL, büyük veriden gerçek anlamda yararlanabilmek için çoklu uygulama alanlarında yeniden kullanılabilir, ölçeklenebilir ve dağıtımı kolay platformlar geliştirmektedir. Yaklaşımları, gerçek dünya uygulamaları sağlamak için endüstriyle yakından işbirliği yapmak ve büyük veri sorununu temelde çok disiplinli olarak incelemek içindir. CSAIL'in devam eden projeleri arasında Trento akıllı şehirde daha iyi bir yaşam için büyük veri, sağlık hizmeti için büyük veriyi yorumlayan Yürütme Göç Makinesi'nin (Execution Migration Machine-EM2) yürütülmesi, kriminoloji ve büyük veri için doğal dil arayüzü vardır (Fang ve diğ., 2015).

Büyük verinin verimli veri tabanı yönetim platformlarına ihtiyacı vardır. Araştırmacılar hangi sorgu sınıflarının işlenebilir olduğu ve büyük veri üzerinde sorgu yanıtlarının nasıl kolaylaştırabileceği gibi problemleri ele almaktadırlar. Analitik programların ve veri tabanı sistemlerinin statik ve tam zamanında (just-in-time) derlenmesi ve bellek hiyerarşisini verimli bir şekilde kullanan çekirdek dışı algoritmaların otomatik sentezi devam eden araştırma projeleri arasında yer almaktadır (Fang ve diğ., 2015).

Şirketler Tablo 2.4'te görüldüğü gibi büyük veri yönetim sistemleri geliştirmektedirler. Bu yönetim sistemleri arasında en yaygın olarak kullanılan Hadoop Common, Dağıtık Dosya Sistemi, YARN ve MapReduce modüllerinden oluşan Apache Hadoop'tur. Güvenilir, ölçeklenebilir ve dağıtık hesaplamayı destekleyen bir çerçevedir. Özellikle tek bir sunucudan binlerce bilgisayara ölçeklendirme için tasarlanmıştır. Yerel depolama ve hesaplama olanağı sunmakta ve büyük veri setlerinin dağıtık olarak işlenmesine imkân sağlamaktadır (Fang ve diğ., 2015).

Tablo 2.4: Büyük veri yönetim sistemleri.

Şirket	**Sistem**
IBM	Apache Hadoop, InfoSphere
Cloudera	CDH, Cloudera Standard, Cloudera Enterprise
Oracle	Oracle Big Data Appliance
Google	BigTable
Yahoo!	Sherpa
Amazon	SimpleDB
Microsoft	Dryad
Facebook	Apache Cassandra
Hypertable	HyperTable
ASF	Apache CouchDB

Günümüzde büyük veri sektörünün parçası olan çok çeşitli teknolojiler vardır. Büyük veri sağlayıcıları bu teknolojiler üzerinde çalışmaktadır. Tablo 2.5'te büyük veri sağlayıcıları ve bu sağlayıcıların ürün ve hizmetleri gösterilmektedir (Chen ve Zhang, 2015; Proffitt, 2012).

Tablo 2.5: Büyük veri sağlayıcıları ve ürünleri/hizmetleri.

Sağlayıcı	Ürünler/hizmetler	Sağlayıcı	Ürünler/hizmetler
1010 data	Konsolidasyon ve analiz için büyük tablo türü veri yapıları kullanan büyük veri için host edilen analitik platform	Karmasphere	Apache Hadoop veri depoları için veri analitik ve geliştirme hizmetleri
10gen	MongoDB için ticari destek ve hizmetler	Lucid Imagination	Apache Lucene ve Apache Solr dağıtıcısı, LucidWorks arama yazılımı sağlayıcısı
Acxiom	Pazarlama verileri ve hizmetleri üzerinde veri analitiği ve işleme	MapR Technologies	Hizmetleri ve desteği ile Apache Hadoop'un ticari uygulama dağıtıcısı
Amazon Web Services	Bulut tabanlı veri tabanı, depolama, işleme ve sanal ağ hizmetleri sağlayıcısı	MarkLogic	Veri analitik ve görselleştirme hizmetleri
Aster Data	Map/Reduce teknolojisini kullanan veri analitiği hizmetleri	Netezza Corp.	Çok büyük ölçekte paralel işleme veri araçları, analitik hizmetler
Calpont	Çok büyük ölçekte paralel işlem yetenekleri sağlayan bir sütun sıralı veri tabanı olan InfiniDB Kurumsal	Oracle	Büyük veri araçları, MySQL kümesi, Exadata veri tabanı makinesi dâhil olmak üzere çeşitli donanım ve yazılım önerileri
Cloudera	Hizmetleri ve desteği ile Apache Hadoop'un ticari uygulama dağıtıcısı	ParAccel	Sütun deposu teknolojisini kullanan veri analitiği
Couchbase	Couchbase Sunucu MapReduce tabanlı veri tabanının yanı sıra Apache CouchDB ve Memcached'in ticari sponsoru	Pentaho	Apache Hadoop veri depoları için veri analitik hizmetleri
Datameer	Apache Hadoop veri depoları için veri görselleştirme hizmetleri	Pervasive Software	Hive'e dayalı Apache Hadoop veri depoları için veri analitik hizmetleri
DataSift	Sosyal medya verileri analitik hizmetleri, Twitter'ın lisanslı yeniden sendikasyonu	Platform Computing	Hizmetleri ve desteği ile Apache Hadoop'un ticari uygulama dağıtıcısı
DataStax	Hizmetleri ve desteği ile Apache Cassandra'nın ticari uygulama dağıtıcısı	RackSpace	Bulut tabanlı veri tabanı, depolama ve işleme hizmetleri sağlayıcısı
Digital Reasoning	Host edilen ve yerel olan bir iş zekası veri analitiği aracı olan Synthesys	Revolution Analytics	R tabanlı yazılım kullanan veri analitik ve görselleştirme hizmetleri
EMC	Çok büyük ölçekte paralel işleme veri depo/analitik çözümü olan Greenplum'un üreticileri	Splunk	Loglama odaklı yazılım kullanarak veri analitik ve görselleştirme hizmetleri
Esri	Coğrafi bilgi sistemi veri analitik hizmetleri	Tableau Software	İş zekası ve veri analitik yazılımı
FeedZai	Gerçek zamanlı bir iş zekası aracı olan FeedZai Pulse	Talend	Veri tabanı yönetim yazılımı
Hadapt	Apache Hadoop veri depoları için veri analitik hizmetleri	Teradata	Veri tabanı yönetim yazılımı
Hortonworks	Hizmetleri ve desteği ile Apache Hadoop'un ticari uygulama dağıtıcısı	Vertica Systems	Sütun deposu tabanlı teknolojiler kullanarak veri analitiği
HPCC Systems	Açık kaynaklı büyük çapta paralel işleme hesaplama veri tabanı olan HPCC (High Performance Computing Cluster-Yüksek Performanslı Hesaplama Kümesi)	Apache Flume	Büyük miktarlarda log verilerinin verimli bir şekilde toplanması, bir araya getirilmesi ve taşınması için dağıtık, güvenilir ve kullanışlı bir hizmet olan Flume
IBM	Donanım, veri analitik hizmetleri ve büyük çapta paralel işleme veritabanı db2	Facebook Scribe	Gerçek zamanlı akış log verilerinin bir araya getirilmesi
Impetus	Apache Hadoop veri depoları için veri analitik ve yönetim hizmetleri	Google's Dremel	Salt okunur iç içe geçmiş verilerin analizi için ölçeklenebilir, etkileşimli bir geçici (ad hoc) sorgu sistemi
InfoBright	Hizmetler ve destek ile bir sütun deposu veri tabanı olan InfoBright	Apache Drill	Google'ın Dremel'e dayanan büyük ölçekli veri setlerinin etkileşimli analizi için dağıtık sistem
Jaspersoft	Apache Hadoop veri depoları için veri analitik hizmetleri		

2.1.7. Büyük Verinin Altyapısı

Büyük verinin farklı boyutlarını hacim, hız ve çeşitlilik açısından ele almak için farklı kaynaklardan gelen ve çok yüksek hıza ulaşan büyük miktardaki veriyi işlemek için etkin ve etkili sistemlerin tasarlanması gerekmektedir. Veriler günümüzde dağıtıktır ve büyük veri depolarını saklamak ve işlemek için yeni teknolojiler geliştirilmektedir. Örneğin büyük veri depolama ve işleme için Hadoop MapReduce gibi bulut bilişim teknolojileri araştırılmaktadır (Mehmood ve diğ., 2016). Bu bölümde büyük veri yaşam döngüsü açıklanmaktadır. Ayrıca büyük verinin depolanması ve işlenmesi için bulut bilişim kullanıldığında büyük verinin bulut bilişimin teknolojilerinden nasıl faydalandığı anlatılmaktadır.

2.1.7.1. Büyük Veri Yaşam Döngüsü

Büyük veri, yaşam döngüsü boyunca Şekil 2.5'te (Mehmood ve diğ., 2016) görüldüğü üzere çeşitli aşamalardan geçmek zorundadır.

Şekil 2.5: Büyük veri yaşam döngüsü.

Veri üretimi: Veriler birçok dağıtık kaynaktan oluşturulabilir. Son birkaç yılda insanlar ve makineler tarafından üretilen veri miktarı aşırı derecede artmıştır. Örneğin web'de günde 2,5 kentilyon (quintillion) bayt veri üretilmektedir ve dünyadaki verilerin %90'ı son birkaç yıl içinde üretilmiştir. Bir sosyal paylaşım sitesi olan Facebook tek başına günde 25 TB yeni veri üretmektedir. Genellikle üretilen veriler büyük, çeşitli ve karmaşıktır. Bu nedenle geleneksel sistemlerin bu verilerle başa çıkması zordur. Üretilen veriler genellikle iş, internet, araştırma vb. gibi belirli bir alanla ilişkilendirilmektedir (Mehmood ve diğ., 2016).

Veri depolama: Bu aşama, büyük ölçekli veri setlerinin depolanması ve yönetilmesini ifade etmektedir. Bir veri depolama sistemi donanım altyapısı ve veri yönetimi olmak üzere iki bölümden oluşur (Hu ve diğ., 2014). Donanım altyapısı, dağıtık depolama gibi çeşitli görevler

için Bilgi ve İletişim Teknolojileri (Information and Communications Technologies, ICT) kaynaklarını kullanmayı işaret etmektedir. Veri yönetimi, büyük ölçekli veri setlerini yönetmek ve sorgulamak için donanım altyapısının üzerine konuşlandırılan yazılım setini ifade eder. Ayrıca depolanan veriler ile etkileşimde bulunmak ve verileri analiz etmek için birçok arayüz sağlamalıdır (Mehmood ve diğ., 2016).

Veri işleme: Veri işleme aşaması temel olarak veri toplama, veri iletimi, ön işleme ve yararlı bilgileri çıkarma sürecini ifade etmektedir. Veri toplama; veriler metin, görüntü ve video gibi farklı kaynaklardan gelebileceğinden gereklidir. Veri toplama aşamasında, veriler özel veri toplama teknolojisi kullanılarak belirli veri üretim ortamından elde edilir. Veri iletimi aşamasında, belirli bir veri üretim ortamından işlenmemiş veriler toplandıktan sonra çeşitli analitik uygulamalar için verileri uygun bir depoya iletmek için yüksek hızlı bir iletim mekanizmasına ihtiyaç vardır. Son olarak ön işleme aşaması, verilerin anlamsız ve gereksiz kısımlarını ortadan kaldırmayı amaçlamaktadır. Böylece daha fazla depolama alanı kurtarılabilir (Mehmood ve diğ., 2016).

Veri ve alana özgü analitik yöntemler, birçok uygulamada anlamlı bilgiler elde etmek için kullanılır. Veri analitiğindeki farklı alanlar değişik veri karakteristikleri gerektirse de bu alanlardan birkaçı, verilerden değer çıkarmada verileri incelemek, dönüştürmek ve modellemek için benzer temel teknolojiden yararlanabilir. Gelişmekte olan veri analitiği araştırmaları altı teknik alana ayrılabilir: yapısal veri analitiği, metin analitiği, multimedya analitiği, web analitiği, ağ analitiği ve mobil analitik (Hu ve diğ., 2014; Mehmood ve diğ., 2016).

2.1.7.2. Bulut Bilişim ve Büyük Veri

Büyük veri, bulut bilişim için de bir gereksinim olan büyük çapta hesaplama ve depolamaya ihtiyaç duymaktadır. Bulut bilişim; maliyet tasarrufu ve ölçeklenebilirlik gibi birçok avantajı sayesinde şirketlerin ve işletmelerin bulutu benimsemelerini sağlamaktadır. Aynı zamanda çok büyük işleme gücü ve depolama kapasitesi sunmaktadır. Sanallaştırma, dağıtık depolama ve işleme gibi bulut bilişimde kullanılan teknolojiler, konvansiyonel sistemlerde zor olarak görülen görevleri gerçekleştirmeyi mümkün hale getirmektedir. Diğer taraftan bulut bilişim, buluta özgü önemli gizlilik sorunlara neden olmaktadır. İnsanlar, verilerinin bulut üzerinde güvenli olacağından emin olmadıkça özel veya hassas verilerini buluta aktarmaktan çekinmektedirler. Bulut üzerinde güvenilir ve güvenli bir büyük veri depolama ve işleme

sistemi oluşturmak için bazı zorluklar vardır. Bunlar dış kaynak kullanımı (outsourcing), çoklu kullanım (multi-tenancy) ve büyük çapta hesaplamadır (massive computation) (Mehmood ve diğ., 2016; Xiao ve Xiao, 2013).

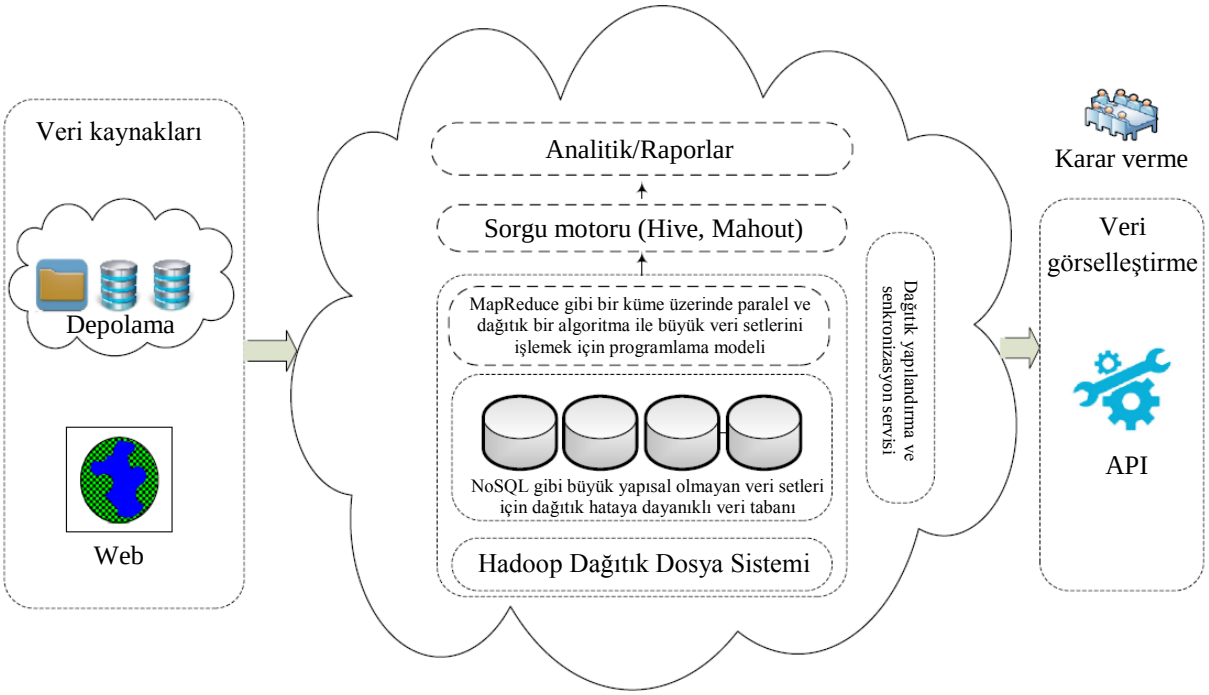
Dış kaynak kullanımı: Şirketler günümüzde sermaye ve işlemsel giderleri azaltmak için verilerini dış kaynak kullanarak bulutta tutarlar. Ancak verileri dış kaynak kullanarak bulutta tutma müşterilerin, verilerinin fiziksel kontrolünü kaybedeceği anlamına gelmektedir. Veriler üzerindeki kontrol kaybı, bulut güvensizliğinin ana nedenlerinden biri haline gelmiştir. Güvensizlik, bulut bilişim müşterilerinin gizliliklerine ciddi zararlar verebilir. Bu sorunlar, güvenli hesaplama ortamı ve veri depolaması sağlanarak ele alınabilir. Buna ek olarak bulutta tutulan veriler gizlilik ve bütünlük açısından da müşterilere doğrulanabilir olmalıdır.

Çoklu kullanım: Sanallaştırma, aynı bulut platformunun birden fazla müşteri tarafından paylaşılmasını mümkün kılmıştır. Farklı bulut kullanıcılarına ait olan veriler, bazı kaynak ayırma ilkeleri ile aynı fiziksel depolama alanına yerleştirilebilir. Böyle bir ortamda, kötü niyetli bir kullanıcının kendisine ait olmayan verilere yasadışı olarak erişmesi nispeten kolaydır. Ayrıca bu ortamda veri ihlali ve hesaplama ihlali gibi bir dizi sorun ortaya çıkabilir. Bu nedenle potansiyel gizlilik ve güvenlik riskleri ile başa çıkmak için mekanizmalar tasarlamak çok önemlidir.

Büyük çapta hesaplama: Bulut bilişimin büyük çapta veri depolama ve yoğun hesaplama kabiliyeti nedeniyle, bireylerin mahremiyetini korumak için olan geleneksel mekanizmalar yeterli değildir (Mehmood ve diğ., 2016).

2.1.7.3. Büyük Veride Bulut Bilişim Kullanımı

Bulut bilişim ve büyük veri ayrılmaz bir bütündür. Büyük veri kullanıcılara çoklu veri setlerinde dağıtık sorguları işlemek ve sonuç setlerini zamanında döndürmek için emtia (commodity) hesaplama kullanma olanağı sağlamaktadır. Bulut bilişim, dağıtık veri işleme platformlarından oluşan bir sınıf olan Hadoop'un kullanımıyla temeli oluşturan motoru sağlar (Hashem ve diğ., 2015). Büyük veride bulut bilişimin kullanımı Şekil 2.6'da (Hashem ve diğ., 2015) gösterilmektedir.



Şekil 2.6: Büyük veride bulut bilişim kullanımı.

Şekilde görüldüğü üzere bulut ve Web'den gelen büyük veri kaynakları, dağıtık hataya dayanıklı bir veri tabanında saklanır ve bir kümede paralel dağıtık algoritma ile büyük veri setleri için olan bir programlama modeli ile işlenirler. Veri görselleştirmenin temel amacı, karar vermede farklı grafikler aracılığıyla görsel olarak sunulan analitik sonuçlarının görüntülenmesidir (Hashem ve diğ., 2015).

Büyük veri, bir bilgisayara veya elektronik cihaza bağlı yerel depolama yerine, bulut bilişime dayalı dağıtık depolama teknolojisini kullanmaktadır. Büyük verinin değerlendirilmesi, sanallaştırılmış teknolojiler kullanılarak geliştirilen hızlı büyüyen bulut tabanlı uygulamalar tarafından yürütülmektedir. Bu nedenle bulut bilişim sadece büyük verinin hesaplanması ve işlenmesi için olanaklar sağlamakla kalmaz, aynı zamanda bir servis modeli olarak da hizmet vermektedir. Birtakım büyük veri bulut sağlayıcısının karşılaştırılması Tablo 2.6'da gösterilmektedir (Hashem ve diğ., 2015).

Tablo 2.6: Birtakım büyük veri bulut sağlayıcısının karşılaştırılması.

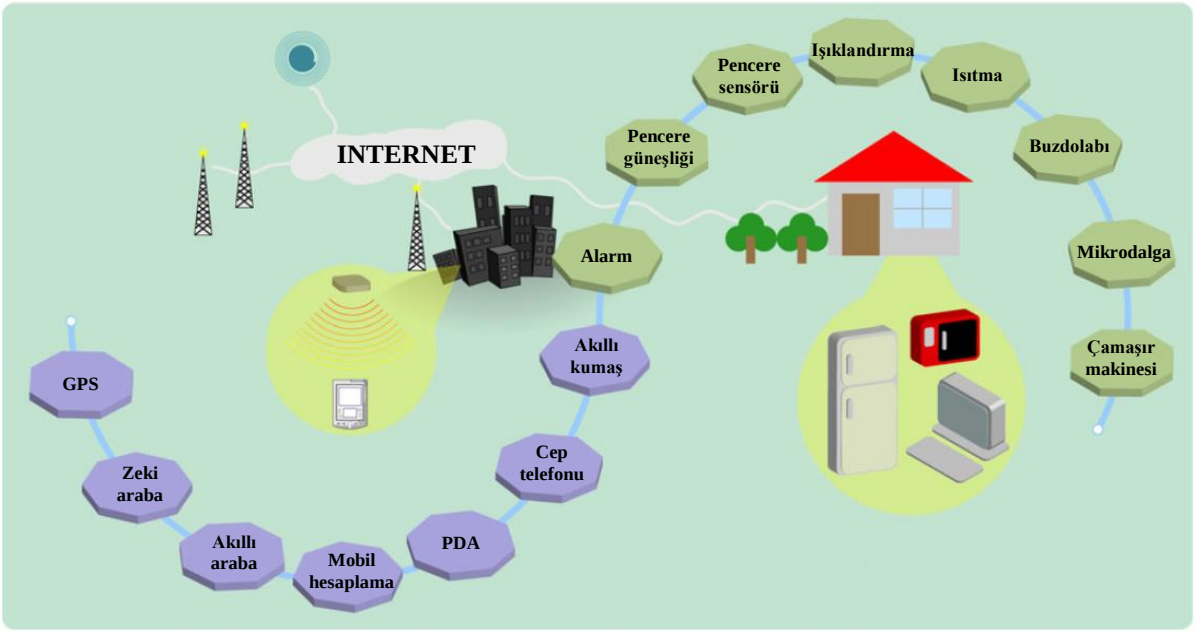
Özellik	Büyük veri bulut sağlayıcısı			
	Google	**Microsoft**	**Amazon**	**Cloudera**
Büyük veri depolama	Google bulut servisleri	Azure	S3	-
MapReduce	AppEngine	Azure üzerinde Hadoop	Esnek MapReduce (Hadoop)	MapReduce YARN
Büyük veri analitiği	BigQuery	Azure üzerinde Hadoop	Esnek MapReduce (Hadoop)	Esnek MapReduce (Hadoop)
İlişkisel veri tabanı	Bulut SQL	SQL Azure	MySQL veya Oracle	MySQL, Oracle, PostgresSQL
NoSQL veri tabanı	AppEngine veri deposu	Tablo depolama	DynamoDB	Apache Accumulo
Akış işleme	Arama API'si	StreamInsight	Önceden paketlenmiş hiçbir şey	Apache Spark
Makine öğrenmesi	Tahmin API'si	Hadoop + Mahout	Hadoop + Mahout	Hadoop + Oryx
Veri içe aktarım	Ağ	Ağ	Ağ	Ağ
Veri kaynakları	Birkaç örnek veri kümesi	Windows Azure marketplace	Açık veri setleri	Açık veri setleri
Elverişlilik	Özel betadaki bazı servisler	Özel betadaki bazı servisler	Açık üretim	Endüstriler

Talia (2013), veri türlerinin karmaşıklığını ve çeşitliliğini incelemiş ve büyük veri setleri üzerinde analiz yapmak için gerekli olan işlem gücünü araştırmıştır. Bulut bilişim altyapısı, büyük veri analizini gerçekleştirmek için gerekli veri deposunu karşılamada etkili bir platform olarak hizmet verebilmektedir. Bulut tabanlı teknolojiler bu yeni ortamla başa çıkmak zorundadır. Çünkü eş zamanlı işleme için büyük verinin üstesinden gelmek, giderek daha karmaşık hale gelmektedir (Ji ve diğ., 2012). MapReduce (Dean ve Ghemawat, 2008), bulut ortamında büyük veri işleme için iyi bir örnektir ve kümede paralel olarak depolanan büyük boyutlardaki veri setlerinin işlenmesine olanak sağlamaktadır. Küme hesaplama (cluster computing); bilgisayar gücü, depolama ve ağ iletişimi gibi dağıtık sistem ortamlarında iyi performans sergilemektedir. Benzer şekilde Bollier ve Firestone (2010), küme hesaplamanın veri büyümesi için iyi bir ortam sağlaması yeteneğini vurgulamıştır. Bununla birlikte Miller

(2013) veri elverişliliğinin (availability) eksikliğinin masraflı olduğunu savunmuştur. Çünkü analitik metotların yanlış kullanımı veya yöntemlerdeki doğal zayıflıklar yanlış ve maliyetli kararlar üretebilmektedir. Veri Tabanı Yönetim Sistemleri (Database Management Systems-DBMS) mevcut bulut bilişim mimarisinin bir parçası olarak kabul edilmektedir ve uygulamaların eski kurumsal altyapılardan yeni bulut altyapısı mimarilerine kolay geçişini sağlamak için önemli bir rol oynamaktadır. Kurumların büyük veri depolama ve işleme gereksinimlerinin zorluğuyla başa çıkmak için teknolojiyi hızla benimseme ve uygulama baskısı, beklenmedik riskleri ve sonuçları beraberinde getirmektedir. Literatürde bulut bilişim teknolojilerinin kullanımı ile büyük veriyi ele alan birçok çalışma vardır (Aluru ve Simmhan, 2015; Demirkan ve Delen, 2013; Hipgrave, 2013; Kwon ve diğ., 2014; Lee ve diğ., 2012; Pandey ve Nepal, 2013; Schadt ve diğ., 2011; Schnase ve diğ., 2017; Singh ve diğ., 2014; Srirama ve diğ., 2012; Talia, 2013; Tannahill ve Jamshidi, 2014; Yan ve diğ., 2013; Zhang ve diğ., 2013).

2.1.7.4. Nesnelerin İnterneti ve Büyük Veri

Nesnelerin İnterneti (Internet of Things-IoT) yaklaşımında muazzam miktarda ağ sensörü, gerçek dünyada çeşitli cihazlara ve makinelere yerleştirilmiştir. Farklı alanlarda konuşlandırılan bu tür sensörler; çevresel, coğrafi, astronomik ve lojistik gibi çok çeşitli veriler toplayabilmektedir (Chen ve diğ., 2014). Mobil cihazlar, ulaşım vasıtaları, kamu tesisleri ve ev aletlerinin hepsi Şekil 2.7'de (Chen ve diğ., 2014) görüldüğü üzere IoT'de veri toplama aracı olabilmektedir.



Şekil 2.7: IoT'de veri toplama araçları.

IoT tarafından üretilen büyük veri, toplanan farklı veri türleri nedeniyle genel büyük verilerle karşılaştırıldığında farklı özelliklere sahiptir. Bu verilerin en klasik özellikleri heterojenlik, çeşitlilik, yapısal olmayan özellik, gürültü ve yüksek fazlalıktır. Mevcut IoT verileri büyük verinin baskın kısmı olmasa da, sensörlerin sayısı 2030'a kadar bir trilyona ulaşacaktır. Bu durumda HP'nin tahminine göre IoT verileri büyük verinin en önemli parçası olacaktır. Intel'in raporu IoT'deki büyük verilerin, büyük veri yaklaşıma uyan üç özelliğe sahip olduğuna dikkat çekmektedir: 1) veri yığınları üreten terminaller, 2) IoT tarafından üretilen veriler genellikle yarı yapısal veya yapısal olmayandır, 3) IoT verileri sadece analiz edildiğinde yararlıdır (Chen ve diğ., 2014).

Günümüzde IoT'nin veri işleme kapasitesi, toplanan verilerin gerisinde kalmıştır ve IoT'nin gelişimini desteklemek için büyük veri teknolojilerinin önerilmesini hızlandırmak son derece önemlidir. IoT'nin başarısı büyük veri ve bulut bilişimin etkili entegrasyonuna bağlı olduğu için IoT operatörleri büyük verinin öneminin farkına varmıştır. IoT'nin geniş çapta konuşlandırılması, aynı zamanda birçok şehri büyük veri çağına da taşıyacaktır (Chen ve diğ., 2014).

Büyük verinin gelişimi çoktan geride kalmış iken IoT uygulamaları için büyük veriyi benimsemeye zorlayıcı bir ihtiyaç vardır. Büyük veri ve IoT teknolojilerinin birbirine bağımlı olması ve birlikte geliştirilmelerinin gerektiği yaygın olarak kabul edilmiştir. Bir yandan

IoT'nin yaygın olarak konuşlanması, hem miktar hem de kategorideki yüksek veri büyümesine yol açmaktadır. Böylece büyük veri uygulama ve geliştirme imkânı sağlanmaktadır. Diğer yandan büyük veri teknolojisinin IoT'ye uygulanması da IoT'nin araştırma gelişmelerini ve iş modellerini hızlandırmaktadır (Chen ve diğ., 2014).

2.1.8. Büyük Verinin Zorlukları

Büyük veri çağında hızlı bir şekilde artan veri seli; veri toplama, depolama, yönetim ve analiz konularında büyük zorluklar doğurmaktadır. Geleneksel veri yönetim ve analiz sistemleri İlişkisel Veri Tabanı Yönetim Sistemi'ne (Relational Database Management System-RDBMS) dayanmaktadır. Ancak bu RDBMS'ler sadece yarı yapısal veya yapısal olmayan veriler dışındaki yapısal veriler için geçerlidir. Ek olarak RDBMS'ler giderek daha pahalı donanımlar kullanmaktadırlar. Görüldüğü üzere geleneksel RDBMS'ler, büyük verinin devasa hacmi ve heterojenliği ile başa çıkamamaktadır. Araştırmacılar bu sorunlar için farklı perspektiflerden bazı çözümler önermişlerdir. Örneğin büyük verinin maliyet verimliliği, esneklik ve kolay sürüm yükseltme (upgrading)/sürüm düşürme (downgrading) gibi altyapı gereksinimlerini karşılamak için bulut bilişim kullanılmaktadır. Büyük ölçekli düzensiz veri setlerinin kalıcı depolanması ve yönetimi çözümleri için dağıtık dosya sistemleri (Howard ve diğ., 1988) ve NoSQL (Cattell, 2011) veritabanları iyi seçimlerdir. Bu tür programlama çerçeveleri, özellikle web sayfası sıralaması için kümelenmiş görevleri işlemede büyük başarı göstermektedir. Bu yenilikçi teknolojilere veya platformlara dayanan çeşitli büyük veri uygulamaları geliştirilebilir. Ayrıca büyük veri analiz sistemlerini konuşlandırmak kolay değildir (Chen ve diğ., 2014).

Literatürde var olan bazı çalışmalar (Agrawal ve diğ., 2012; Chaudhuri ve diğ., 2011; Chen ve diğ., 2014; Labrinidis ve Jagadish, 2012) büyük veri uygulamalarının geliştirilmesindeki zorlukları ve engelleri ele almıştır. Temel zorluklar aşağıdaki gibi sıralanabilir:

Veri gösterimi: Birçok veri setinin tür, yapı, semantik, organizasyon, taneciklilik (granularity) ve erişilebilirlik açısından belirli düzeyde heterojenliği vardır. Veri gösterimi (data representation); bilgisayar analizi ve kullanıcı yorumlaması için verileri daha anlamlı hale getirmeyi amaçlamaktadır. Fakat yanlış bir veri gösterimi, orijinal verilerin değerini azaltır ve hatta etkili veri analizini engelleyebilir. Etkili veri temsili, farklı veri setleri üzerinde verimli

operasyonlar gerçekleştirebilmek için veri yapısını, sınıfını, türünü ve entegre teknolojileri yansıtmalıdır.

Fazlalık azaltma ve veri sıkıştırma: Veri setlerinde genellikle yüksek düzeyde bir fazlalık vardır. Fazlalığın azaltılması (redundancy reduction) ve verilerin sıkıştırılması (data compression), tüm sistemin dolaylı maliyetinin verilerin potansiyel değerlerinin etkilenmemesi gerekçesiyle azaltılması için yararlıdır. Örneğin algılayıcı (sensor) ağlar tarafından üretilen çoğu veri son derece gereksizdir. Bu veriler büyüklük sırasına göre filtrelenebilir ve sıkıştırılabilir.

Veri yaşam döngüsü yönetimi: Yaygın (pervasive) algılama ve hesaplama, depolama sistemlerinin nispeten yavaş ilerlemesiyle karşılaştırıldığında görülmemiş oranlarda ve hacimlerde veri üretmektedir. Birinin mevcut depolama sistemlerinin bu kadar büyük çapta veriyi destekleyememesi olduğu birçok zorluk vardır. Genel anlamda büyük veride saklanan değerler veri yeniliğine bağlıdır. Bu nedenle hangi verinin depolanacağına ve hangi verinin çıkarılacağına karar vermek için analitik değere ilişkin bir veri önceliği ilkesi oluşturulmalıdır.

Analitik mekanizma: Büyük verinin analitik sistemi, sınırlı bir süre içinde heterojen veri yığınlarını işlemek zorundadır. Ancak geleneksel RDBMS'ler, performans gereksinimlerini karşılayamayan ölçeklenebilirlik ve genişletilebilirlik eksikliği ile tasarlanmıştır. İlişkisel olmayan veri tabanları, yapısal olmayan verilerin işlenmesinde benzersiz avantajlar göstermiştir ve büyük veri analizinde ana akım olmaya başlamıştır. Böyle olmasına rağmen ilişkisel olmayan veri tabanlarının, performanslarında ve belirli uygulamalarında hala problemler vardır. RDBMS'ler ve ilişkisel olmayan veritabanları arasında bir uzlaşmacı çözüm bulunmalıdır. Örnek olarak Facebook ve Taobao gibi bazı şirketler her iki veri tabanı türünün avantajlarını birleştiren hibrit bir veri tabanı mimarisi kullanmaktadır. Bellek içi veri tabanında (in memory database) ve tahmini analize (approximate analysis) dayalı örnek veriler üzerinde daha fazla araştırma yapmaya ihtiyaç vardır.

Veri gizliliği: Günümüzdeki büyük veri hizmeti sağlayıcıları veya sahipleri, sınırlı kapasiteleri nedeniyle büyük veri setlerini etkili bir şekilde koruyamamış ve analiz edememiştir. Veri sağlayıcıları potansiyel güvenlik risklerini artıran bu tür verileri analiz etmek için profesyonellere veya araçlara güvenmek zorundadırlar. Örneğin işlem (transactional) veri setleri genellikle kilit iş süreçlerini yürütmek için bir dizi eksiksiz işletim verisi içermektedir.

Bu veriler en düşük taneciklilik ayrıntıları ve kredi kartı numaraları gibi bazı hassas bilgileri içermektedir. Bu nedenle büyük verinin analizi, yalnızca bu tür hassas verileri korumak için uygun önleyici tedbirler alındığında işlenmek üzere üçüncü bir şahısa iletilebilir.

Enerji yönetimi: Ana bilgisayar (mainframe) hesaplama sistemlerinin enerji tüketimi hem ekonomi hem de çevre açısından çok ilgi çekmektedir. Veri hacminin ve analitik gereksinimlerin artmasıyla büyük verinin işlenmesi, depolanması ve iletilmesi kaçınılmaz olarak daha fazla elektrik enerjisi tüketecektir. Bu sebeple genişletilebilirlik ve erişilebilirlik sağlanırken, büyük veri için sistem düzeyinde güç tüketimi kontrol ve yönetim mekanizması oluşturulmak zorundadır.

Genişletilebilirlik ve ölçeklenebilirlik: Büyük verinin analitik sistemi mevcut ve gelecekteki veri setlerini desteklemelidir. Analitik algoritmalar giderek genişleyen ve daha karmaşık olan veri setlerini işleyebiliyor olmalıdır.

İşbirliği: Büyük veri analizi, farklı alanlardaki uzmanların büyük verinin potansiyelini ortaya çıkarmak için işbirliği yapmasını gerektiren disiplinler arası bir araştırma alanıdır. Çeşitli alanlardaki bilim insanlarının ve mühendislerinin farklı veri türlerine erişmelerine ve uzmanlıklarından tam olarak yararlanılabilmesine yardımcı olmak için kapsamlı bir büyük veri ağı mimarisi oluşturulmalıdır.

2.2. BÜYÜK VERİDE GİZLİLİK KORUMASI

Büyük veri işlemede gizlilik koruması; gizlilik korumalı veri yayınlama ve veriden bilgi çıkarma olarak iki aşamaya ayrılabilir. İlk aşamada, toplanan veriler veri sahibi hakkında hassas bilgiler içerebileceğinden, bilgileri istenmeyen ifşalardan korumak hedeflenmektedir. İkinci aşamadaki amaç ise gizlilik ihlali yapmadan veriden anlamlı bilgi çıkarmaktır (Mehmood ve diğ., 2016). Bu bölümde; gizlilik korumalı veri yayınlama, veriden bilgi çıkarma, gizlilik korumaya yönelik çalışmalar ve saldırı türleri ile ilgili genel bilgiler verilmektedir.

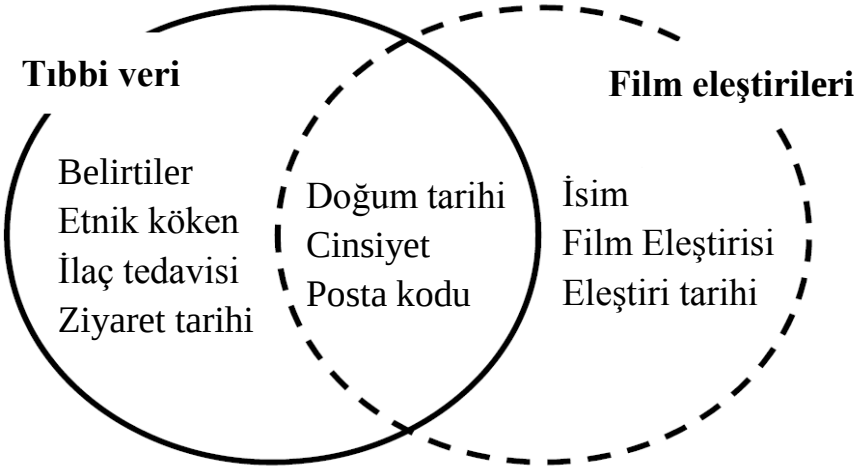
2.2.1. Gizlilik Korumalı Veri Yayınlama

Gizlilik Korumalı Veri Yayınlama (Privacy Preserving Data Publishing-PPDP) sırasında, toplanan veriler veri sahibi hakkında hassas bilgiler içerebilir. Bilgileri daha ileri işlemler için doğrudan yayınlama veri sahibinin gizliliğini ihlal edebilir. Dolayısıyla veri değişikliği (data

modification), veri sahibiyle ilgili kişisel bilgileri ifşa etmeyecek şekilde yapılmalıdır. Öte yandan değiştirilen veriler, veri yayınlamanın asıl amacını ihlal etmemek için değişiklikten sonra da kullanışlı olmalıdır. Veri gizliliği ve kullanılabilirliği (utility) birbiriyle ters ilişkilidir. Verileri daha sonraki işlemler için yayınlamadan ya da depolamadan önce değiştirmek için birçok çalışma yapılmıştır (Fung ve diğ., 2010; Wong ve Fu, 2010). PPDP, bir kullanıcının gizliliğini korumak için genellikle anonimleştirme tekniklerini kullanmaktadır. Orijinal veriler hassas ve özel olarak kabul edilmektedir ve birden fazla kayıttan oluşmaktadır. Her bir kayıt aşağıdaki dört nitelikten oluşabilir (Fung ve diğ., 2010).

Tanımlayıcı (Identifier-ID): Bir kişiyi benzersiz (eşsiz) şekilde tanımlamak için kullanılabilen niteliklerdir. İsim, ehliyet numarası ve telefon numarası vb. örnek olabilir.

Yarı tanımlayıcı (Quasi-identifier-QI): Bir kaydı eşsiz olarak tek başına tanımlayamayan ancak bazı harici veri setleriyle bağlantılıysa, kayıtları yeniden tanımlayabilen niteliklerdir. Yarı tanımlayıcı örneği Şekil 2.8'de (Mehmood ve diğ., 2016) gösterilmektedir.



Şekil 2.8: Yarı tanımlayıcılar ve bağlantılı kayıtlar.

Hassas nitelik (Sensitive attribute-SA): Bir kişinin gizlemek isteyebileceği niteliklerdir. Maaş ve hastalık örnek olarak gösterilebilir.

Hassas olmayan nitelik (Non-sensitive attribute-NSA): İfşa edilmesi durumunda kullanıcının gizliliğini ihlal etmeyecek olan niteliklerdir. Tanımlayıcı, yarı tanımlayıcı ve hassas nitelikler dışındaki tüm nitelikler hassas olmayan nitelik olarak sınıflandırılır.

Veriler, daha sonraki işlemler için yayınlanmadan veya depolanmadan önce tanımlayıcılar kaldırılarak ve yarı tanımlayıcılar değiştirilerek anonim hale getirilir. Anonimleştirme sonucunda veri sahibinin kimliği ve hassas değerler kötü niyetli kişilerden gizlenir. Anonimleştirilmesi gereken veri miktarı, esas olarak anonim hale getirilecek veride gizliliğin ne kadar korunmak istenildiğine bağlıdır. Gizlilik modelleri temel olarak bir saldırganın bir bireyi tanımlama yeteneğine göre iki kategoriye ayrılır (Fung ve diğ., 2010). Birinci kategori saldırganın kayıtları, harici veri kaynaklarıyla bağlayarak belirli bir kullanıcının kayıtlarını tanımlayabileceği varsayımına dayanmaktadır. İkinci kategori ise saldırganın olasılıksal saldırıları (probabilistic attacks) gerçekleştirmek için yeterli geçmiş bilgisine sahip olduğu varsayımına dayanmaktadır. Diğer bir deyişle saldırgan, belirli bir kullanıcının kayıtlarının veri tabanında olup olmadığı konusunda kendinden emin bir tahmin yapabilir. Bu sorunlarla başa çıkmak için önerilen birçok model vardır. Bunlardan bazıları şöyledir: Kayıt bağlamayı (record linkage) önlemek için *k*-anonimlik (*k*-anonymity), nitelik bağlama (attribute linkage) ve kayıt bağlamayı önlemek için *l*-çeşitlilik (*l*-diversity), olasılıksal saldırılar ve nitelik bağlamayı önlemek için *t*-yakınlık (*t*-closeness) (Fung ve diğ., 2010).

2.2.1.1. Anonimleştirme Teknikleri

Orijinal veri setleri yayınlanmadan önce belirtilen gizlilik gereksinimlerini karşılayacak şekilde değiştirilir. Aşağıda yer alan anonimleştirme işlemlerinden biri gizliliği korumak için verilere uygulanır (Fung ve diğ., 2010).

Genelleme: Genelleme işlemi (generalization), belirli yarı tanımlayıcı niteliklerin değerlerinin daha az belirli betimlemeler ile değiştirilmesidir. Bu işlemde bazı değerler bir niteliğin sınıflandırılmasında bir üst değer ile değiştirilir. Örnek olarak bir meslek niteliği şarkıcı veya oyuncu yerine sanatçı ile temsil edilebilir. Tam alan genelleme (full domain generalization), alt ağaç genelleme (subtree generalization), çok boyutlu genelleme (multidimensional generalization), kardeş genelleme (sibling generalization) ve hücre genelleme (cell generalization) genelleme teknikleri türleridir.

Gizleme: Gizleme işleminde (suppression) bazı değerler özel bir karakter (örneğin "*") ile değiştirilir ve bu da değiştirilen değerin ifşa edilmediğini gösterir. Gizleme şemalarına örnek olarak kayıt gizleme, değer gizleme ve hücre gizleme gösterilebilir.

Anatomizasyon: Anatomizasyon işlemi (anatomization), yarı tanımlayıcı ya da hassas nitelikleri değiştirmek yerine bunlar arasındaki bağlantıyı koparır. Bu işlemde yarı tanımlayıcı ve hassas nitelikteki veriler iki ayrı tabloda yayınlanır. Bir tablo yarı tanımlayıcıları, diğer tablo ise hassas nitelikleri içerir. Her iki tabloda da genelde grup numarası adı verilen ortak bir nitelik bulunur. Aynı grup, gruptaki hassas değerlerle bağlantılı olan grup numarası için aynı değere sahip olacaktır.

Permütasyon: Permütasyon işleminde (permutation) yarı tanımlayıcı ve hassas nitelik arasındaki ilişki, bir dizi kayıt gruplara ayrılarak ve her bir grup içindeki hassas değerler karıştırılarak bozulur.

Pertürbasyon: Pertürbasyon işleminde (perturbation) orijinal veri değerleri bazı sentetik veri değerleri ile değiştirilir. Böylece değiştirilmiş verilerden hesaplanan istatistiksel bilgiler, orijinal veriden hesaplanan istatistiksel bilgilerle önemli ölçüde farklılık göstermez. Pertürbasyon işlemine örnek olarak gürültü ekleme, verilerin karşılıklı olarak değiştirilmesi (swapping) ve sentetik veri oluşturma verilebilir. Pertürbasyon ile ilgili sorun, yayınlanan kayıtların sentetik olması ve gerçek dünyada herhangi bir şey ifade etmemesidir.

2.2.1.2. Gizlilik ve Kullanılabilirlik Dengesi

Yüksek seviyeli veri anonimleştirme, gizliliğinin iyi korunmuş olduğunu göstermektedir. Ancak verinin kullanılabilirliğini etkileyebilmektedir. Bu da veriden daha az değer çıkarılabileceği anlamına gelmektedir. Bu nedenle gizlilik ve kullanılabilirlik arasındaki dengeyi sağlamak büyük veri uygulamalarında çok önemlidir. Veri kullanılabilirliğindeki azalma, bilgi kaybı (information loss) ile temsil edilmektedir. Literatürde bilgi kaybını ölçmek için çeşitli yöntemler önerilmiştir. Bu yöntemlere örnek olarak Kullback-Leibler uzaklığı (Kullback ve Leibler, 1951), minimal bozulma (minimal distortion) (Sweeney, 2002), belirlenebilirlik metriği (discernibility metric) (Bayardo ve Agrawal, 2005), normalize ortalama eşdeğerlik sınıfı boyutu metriği (normalized average equivalence class size metric) (LeFevre ve diğ., 2006), ağırlıklı kesinlik cezası (weighted certainty penalty) (Xu ve diğ., 2006) ve bilgi teorik metrikleri (Gionis ve Tassa, 2009; Xu ve diğ., 2015) verilebilir. PPDP algoritmaları gizlilik ve kullanılabilirlik arasındaki denge sorunlarını çözmek için genellikle açgözlü (greedy) yaklaşım kullanırlar. Bu algoritmalar, gizlilik koruması ve bilgi kaybı için verilen metrikleri kullanarak çoklu tablolar oluştururlar ve anonimleştirme süreci boyunca belirli gizlilik modeli

gereksinimlerini sağlarlar. Açgözlü bir algoritmanın çıkışı minimum bilgi kaybı olan tablodur (Mehmood ve diğ., 2016).

Gizliliğin ölçülmesi çok zor bir görevdir. Örneğin bir parça verinin bir veri sahibinden toplandığı bir senaryo düşünelim. Veri sahibi, üçüncü bir şahısla (third party) ne kadar ve ne tür bir bilgi paylaşmak istediğini karar vermekte özgürdür. Veriler üçüncü şahıslara verildikten sonra bazı gizlilik kayıpları ortaya çıkabilir. Farklı veri sahipleri aynı verileri üçüncü şahıslara verebilir. Fakat gizlilik ifşası olduğunda gizliliği ciddiye alan bireyler, gizlilik hakkında az endişe duyanlardan daha fazla kayıp görebilirler (Mehmood ve diğ., 2016).

2.2.2. Veriden Bilgi Çıkarma

Gizlilik korumalı veri madenciliği teknikleri gizliliği ihlal etmeden büyük veriden yararlı bilgiler çıkarmak için geliştirilmiştir. Bilgi çıkarma işlemini verilerin örüntülerini ve yönelimlerini belirleyerek yaparlar. Büyük veri; büyük, karmaşık ve dinamik olarak değişken veriler içerebileceğinden, bu teknikler büyük verilere doğrudan uygulanamazlar. Gizlilik korumalı veri madenciliği teknikleri, büyük veri ile etkili bir şekilde başa çıkmak için modifiye edilmelidir ya da bazı özel teknikler kullanılmalıdır. Buna ek olarak modifiye edilen bu teknikler gizlilikle ilgili kaygıları ele almalıdır. Büyük ölçekli ve karmaşık verileri analiz etmek için önerilen birkaç teknik vardır. Bu teknikler genel olarak kümeleme, sınıflandırma ve birliktelik kuralı tabanlı teknikler olarak gruplanabilir (Mehmood ve diğ., 2016).

2.2.2.1. Gizlilik Korumalı Kümeleme

Kümeleme, bilinmeyen verileri analiz etme yeteneğine sahip popüler veri işleme tekniklerinden biridir. Kümelemenin ardındaki temel fikir, işaretlenmemiş giriş verilerini birçok farklı gruba ayırmaktır (Jain ve diğ., 1999). Geleneksel kümeleme algoritmaları, verilerin aynı formatta olmasını ve tek bir işlem birimine yüklenmesini gerektirir. Bu, büyük veri işleme için uygun değildir. Bu konu ile ilgili son on yılda birçok çözüm sunulmuştur (Fahad ve diğ., 2014; Xu ve Wunsch, 2009). Fakat bu çözümlerin büyük verinin doğası sebebiyle çok sayıda dezavantajı vardır. Bunlar arasında hesaplama karmaşıklığı ve gizlilik kaygısı başlıca problemlerdir. Hesaplama karmaşıklığı sorununu ele almak için Shirkhorshidi ve diğ. (2014) tek makineli kümeleme için örnekleme ve boyut küçültme çözümlerini, çoklu makine kümeleme için ise paralel ve MapReduce (HaritalaİndirgeHaritalaİndirge) çözümlerini öne sürmüştür. Xu ve diğ. (2012), verimliliği artırmak için bulut bilişim tabanlı paralel işleme yöntemi önermiştir. Feldman ve

diğ. (2013), çok büyük veri setlerinde kümelemeyi mümkün kılmak için ana setlerin bir ağaç yapısı kullanılarak oluşturulduğu bir paralel işleme yaklaşımı sunmuştur. Bu çalışmada geleneksel kümeleme algoritmalarına kıyasla işlem süresi ve gerekli enerji miktarı önemli ölçüde azaltılmıştır. Bununla birlikte tüm bu yöntemlerde (Elmisery ve Fu, 2010; Fahad ve diğ., 2014; Feldman ve diğ., 2013; Shirkhorshidi ve diğ., 2014; Xiao-Dan ve diğ., 2007; Xu ve diğ., 2012; Xu ve Wunsch, 2009) gizlilik büyük bir kaygıdır. Büyük hacimli karmaşık veriler söz konusu olduğunda kümelemede gizliliğin korunması zorlu bir sorundur. İlk zamanlarda kümelemede gizliliği korumak için hibrit geometrik veri dönüştürme tabanlı yöntemler (Oliveira ve Zaiane, 2010) önerilmiştir. Ancak bu yöntemler öteleme (translation), ölçekleme (scaling) ve döndürme (rotation) işlemleriyle sayısal nitelikleri değiştirirler. Belirli bir gizlilik seviyesi sağlanabilmesine rağmen veri kullanılabilirliği genellikle azalır. Dolayısıyla bu yöntemler pratik olarak uygulanabilir değildir. Oliveira ve Zaiane (2004), boyutsal indirgeme ve nesne benzerliğine dayalı temsil kullanarak, merkezi veri için bir yöntem önermiştir. Bu yöntem özellikle merkezi veri için tasarlandığından, daha yaygın olarak mevcut olan merkezi olmayan büyük verilerle kullanılamaz. Xiao-Dan ve diğ. (2007), bilinmeyen yeni verilerde kümelemenin etkinliğini artırmak için örneklemeye dayalı olasılıksal bir dağıtık model öne sürmüştür. Elmisery and Fu (2010), karmaşık ve dağıtık verilerle başa çıkmak için dağıtık yerel kümeleme olarak adlandırılan yeni bir algoritma sunmuştur. Bu çalışmada gizlilik koruması sağlamak için homomorfik şifreleme gibi güvenli çok taraflı (multi-party) hesaplama tabanlı teknikler kullanılmıştır. Kümeleme, yukarıda bahsedilen yöntemlerde düşük dereceli istatistik kullanılarak yapılmaktadır. Giriş verileri karmaşık olduğunda, bu daha düşük dereceli istatistikler yetersizdir ve zayıf kümelenme sonuçları verebilir. Bu problemin üstesinden gelebilmek için Shen ve Li (2014) yeni bir metot öne sürmüştür. Çalışmada bir doğrusal ve çekirdek dağıtık kümeleme algoritması oluşturmak için maliyet fonksiyonu olarak bilgi teorik ölçülerini kullanarak bir kümeleme yöntemi geliştirmişlerdir. Bu yöntemde düğümler, komşularıyla orijinal veriler yerine sadece birkaç parametre değiştirirler (Mehmood ve diğ., 2016).

2.2.2.2. Gizlilik Korumalı Sınıflandırma

Sınıflandırma, yeni bir giriş verisinin ait olduğu önceden tanımlanmış grubunun belirlenmesi için bir tekniktir. Sınıflandırma algoritmaları kümeleme algoritmalarına benzer şekilde geleneksel olarak merkezi ortamlarda çalışmak üzere tasarlanmıştır. Geleneksel sınıflandırma

algoritmaları, büyük verinin gereksinimleri ile başa çıkmak için paralel hesaplama ortamına uyacak şekilde modifiye edilirler. Örnek olarak Tekin ve Van Der Schaar (2013), verileri iki yönlü işlemek için "sınıflandır veya sınıflandırma için gönder" olarak bilinen bir sınıflandırma algoritması tasarlamıştır. Bu algoritma verileri kendi sınıflandırır ya da giriş verilerini başka bir sınıflandırıcıya iletir. Özellikle büyük ve karmaşık verileri ele alırken hesaplama açısından verimlidir. Bir başka yeni sınıflandırma algoritmasında Rebentrost ve diğ. (2014), büyük veri sınıflandırması için kuantum tabanlı bir Destek Vektör Makinesi (Support Vector Machine-SVM) önermiştir. Bu yöntem, hesaplama karmaşıklığını ve gerekli eğitim verisini azaltmaktadır. Yöntemin ana sınırlaması kuantum hesaplamadaki gelişmemiş donanım teknolojileridir. Büyük veri için geliştirilen sınıflandırma algoritmaları makul bir performans seviyesine ulaşabilmesine rağmen bu algoritmalar veri gizliliğine çok fazla önem vermemektedir. Agrawal ve Srikant (2000), veriden bilgi keşfi için gizlilik korumalı bir sınıflandırma algoritması önermiştir. Bu algoritmada orijinal veriler, rastgele ofsetler eklenerek değiştirilir. Daha sonra Bayesian formülü, karar ağacını yeniden oluşturmak için orijinal verilerin yoğunluk fonksiyonunu türetmede kullanılır. Bu yöntemle ilgili temel sorun sadece merkezi veriler için uygun olmasıdır. Başka bir gizlilik korumalı veri madenciliği algoritması Evfimievski ve diğ. (2003) tarafından rastgele yeniden oluşturma (reconstruction) teknikleri kullanılarak öne sürüşmüştür. Algoritmadaki rastgele işlem, veri karıştırma (scrambling) yoluyla orijinal verilerin gizliliğini korumaktadır. Fakat bu yöntem farklı veriler için uygun değildir. Agrawal ve Haritsa (2005) tarafından Agrawal ve Srikant (2000) ve Evfimievski ve diğ. (2003)'den farklı olarak dağıtık veri tabanları için bir gizlilik koruma yöntemi önerilmiştir. Bu çalışmada, verilerin gizliliğini koruma altına almak için rastgele bir pertürbasyon matrisi kullanılmıştır. Değiştirilen veri setinden orijinal veri kümesinin, algoritmanın doğası gereği yeniden oluşturulması gerekmektedir. Bu, algoritmanın doğruluğunu önemli ölçüde azaltmaktadır. Weiping ve diğ. (2006), doğruluğu artırmak için tek nitelikli veri rastgele matrisi kullanarak bir algoritma geliştirmişlerdir. Bu matris, verileri çok az oranda modifiye etmek için kullanılır ve orijinal veri setinin yeniden oluşturulması, çok nitelikli ortak dağılım (joint distribution) matrisinin kullanılmasıyla geliştirilir. Bu yöntem gizlilik pahasına doğruluğu artırır. Zhang ve Bi (2010), çok nitelikli ortak dağılım matrisinin avantajını kullanarak sınıflandırma için doğruluk ve gizliliği çok az oranda iyileştiren bir gizlilik koruma yöntemi önermiştir. Ancak bu yöntem büyük ve karmaşık verilerle başa çıkamamaktadır (Mehmood ve diğ., 2016).

2.2.2.3. Gizlilik Korumalı Birliktelik Kuralı Madenciliği

Kümeleme ve sınıflandırma giriş verilerini gruplandırmaya çalışırken, birliktelik kuralları, giriş verileri arasındaki önemli ilişkileri veya örüntüleri bulmak için tasarlanmıştır. Çok büyük veri setlerindeki ilişkileri bulma uzun yıllardır çalışılmaktadır. İlk başlarda örüntü bulmak için FP-ağacı (Han ve diğ., 2000) gibi ağaç yapıları kullanılmıştır. Önceki algoritmalar, paralel hesaplama ve bulut bilişim teknolojileri kullanılması sebebiyle büyük ve farklı veriler için uygun değildir. Büyük ve karmaşık verilerle etkili bir şekilde başa çıkmak için MapReduce kullanılarak birçok yöntem (Leung ve diğ., 2014; Lin ve diğ., 2012; Riondato ve diğ., 2012) geliştirilmiştir. MapReduce kavramı bulut tabanlı birliktelik kuralı bulma algoritmaları için ideal olarak uygundur. Fakat Leung ve diğ. (2014), Lin ve diğ. (2012) ve Riondato ve diğ. (2012) tarafından yapılan çalışmalarda önerilen birliktelik kuralı madenciliği yöntemleri, giriş verilerinin gizliliğini dikkate almamaktadır. Birliktelik kuralı madenciliğinde gizlilik koruma, veriden hassas bilgilerin çıkarılmasının korunması işlemidir. Örneğin Agrawal ve Srikant (2000) tarafından yapılan çalışmada gizlilik, orijinal veriler bozularak korunmaktadır. Veriler, orijinal verilerdeki değerler açığa çıkarılmadan bozulmaktadır. Bozulmuş veriler orijinal veri dağılımını kestirmek için kullanılabilir. Bu yaklaşımda gizlilik seviyesi nispeten düşüktür. Bu nedenle Agrawal ve Aggarwal (2001), gizliliği artırmak ve gizlilik sızıntısını (leakage) azaltmaya yönelik daha zorlu koşullar uygulamıştır. Son zamanlarda Evfimievski ve diğ. (2004) ve Rizvi ve Haritsa (2002) tarafından yapılan çalışmalarda gizlilik koruma teknikleri Boolean birliktelik kurallarına uygulanmıştır. Bu çalışmalarda da diğer yöntemlere benzer olarak orijinal veriler bozulmuştur. Lindell ve Pinkas (2000)'ın çalışmasındaki gibi bazı yöntemlerde karar ağaçlarını oluşturmak için kriptografik teknikler kullanılmaktadır. Gizliliği korumalı veri madenciliği, güvenli çok taraflı hesaplamanın bir parçası olarak kabul edilmektedir. Bu yöntemler bazı gizlilik ve doğruluk seviyelerine ulaşmasına rağmen büyük ve karmaşık verileri ele alma konusunda tam olarak yeterli değildir (Mehmood ve diğ., 2016).

2.2.3. Gizlilik Korumaya Yönelik Çalışmalar

Gizlilik korumalı veri madenciliği ve veri yayınlamada gizlilik koruması; veri anonimleştirme (Nayahi ve Kavitha, 2015, 2017; Samarati, 2001; Samarati ve Sweeney, 1998; Sweeney, 1997, 1998, 2002a, 2002b), veri pertürbasyonu (Chen ve diğ., 2013; Domingo-Ferrer ve diğ., 2001; Herranz ve diğ., 2010; Kim ve Winkler, 2003; Liu ve diğ., 2006; Yang and Qiao, 2010; Zhu ve diğ., 2009), veri rastgeleleştirme (Chen ve diğ., 2007; Chen ve Liu, 2009, 2011; Islam ve

Brankovic, 2011) ve kriptografi (Liu ve diğ., 2015; Pinkas, 2002) gibi çeşitli yöntemler kullanılarak sağlanmaktadır. Bu yöntemler arasında *k*-anonimlik (Sweeney, 2002b) ve Datafly (Sweeney, 1998), Incognito (LeFevre ve diğ., 2005), Mondrian (LeFevre ve diğ., 2006) gibi *k*-anonimlik tabanlı algoritmalar en yaygın olarak kullanılan tekniklerdir. *k*-anonimleştirme, yarı tanımlayıcıların değerlerinin değiştirildiği ve böylece anonimleştirilmiş veri setindeki herhangi bir bireyin en az $k-1$ sayıda diğer bireyler tarafından ayırt edilemediği bir süreçtir (Xu ve diğ., 2014). Tablo 2.7; yaş, cinsiyet ve posta kodunun yarı tanımlayıcı, hastalığın ise hassas nitelik olduğu örnek bir orijinal veri setini göstermektedir.

Tablo 2.7: Örnek bir orijinal veri seti.

Yaş	Cinsiyet	Posta kodu	Hastalık
32	Kadın	34200	Meme kanseri
38	Kadın	34800	Böbrek kanseri
64	Erkek	40008	Cilt kanseri
69	Kadın	40001	Kemik kanseri
53	Erkek	65330	Cilt kanseri
56	Erkek	65380	Böbrek kanseri
75	Kadın	20005	Meme kanseri
76	Erkek	20009	Prostat kanseri
41	Erkek	85000	Akciğer kanseri
47	Erkek	87000	Akciğer kanseri

k-anonimleştirme kullanılarak elde edilen bu orijinal veri setinin 2-anonim hali Tablo 2.8'de gösterilmektedir. Şekilde görüldüğü üzere genelleme ve gizleme işlemleri kullanılarak aynı değerlere sahip beş eşdeğerlik sınıfı elde edilmiştir. Bu 2-anonim gruplar kimlik ifşası ve bağlantı saldırılarıyla mücadele etmektedir.

Tablo 2.8: Orijinal veri setinin 2-anonim hali.

Yaş	Cinsiyet	Posta kodu	Hastalık
[30-40]	Kadın	34***	Meme kanseri
[30-40]	Kadın	34***	Böbrek kanseri
[60-70]	*	4000*	Cilt kanseri
[60-70]	*	4000*	Kemik kanseri
[50-60]	Erkek	653**	Cilt kanseri
[50-60]	Erkek	653**	Böbrek kanseri
[70-80]	*	2000*	Meme kanseri
[70-80]	*	2000*	Prostat kanseri
[40-50]	Erkek	8****	Akciğer kanseri
[40-50]	Erkek	8****	Akciğer kanseri

Machanavajjhala ve diğ. (2007), hassas niteliklerin çeşitlilikten yoksun olduğu *k*-anonimliğini geliştirmek için *l*-çeşitlilik (*l*-diversity) ilkesini öne sürmüştür. *l*-çeşitlilik; yarı tanımlayıcılar ve hassas niteliklerin arasındaki ilişkilere odaklanmaktadır. Yarı tanımlayıcı bir grup, en az *l* sayıda iyi temsil edilen hassas nitelik değeri içerirse *l*-çeşitliliği sağlamış olur. Buna ek olarak hassas niteliklerin entropisi, bir veri setindeki her yarı tanımlayıcı grup için ln *l*'den büyükse entropi *l*-çeşitlilik sağlanır. Li ve diğ. (2007), *l*-çeşitlilik ilkesinin sınırlamalarının üstesinden gelmek için nitelik ifşası ve benzerlik saldırıları ile başa çıkan *t*-yakınlık (*t*-closeness) ilkesini önermiştir. Sun ve diğ. (2011), *l*-çeşitlilik ve entropi *l*-çeşitliliği geliştirerek yukarıdan aşağı (top-down) bir anonimleştirme modeli sunmuştur.

Agrawal ve Srikant (2000), orijinal veri setine Gauss dağılımından rastgele gürültü ekleyerek gizliliği korumak için bir değer bozma yöntemi sunmuştur. Bu yöntem daha sonra daha iyi bir dağılım oluşturmak için Agrawal ve Aggarwal (2001) tarafından geliştirilmiştir.

Evfimievski ve diğ. (2004), verileri rastgeleleştirerek bir birliktelik kuralı madenciliği çerçevesi önermiştir. Daha sonra Evfimievski ve diğ. (2003), gizlilik ihlallerini veri dağıtım bilgileri olmadan sınırlandırmak için bu çerçeveyi modifiye etmiştir. Buna ek olarak Rizvi ve Haritsa (2002), gizlilik sağlamak için olasılıksal bozulma tabanlı bir şema sunmuştur.

Yang ve Qiao (2010), gizlilik koruması ve bilgi muhafazası için yarı tanımlayıcılar ve hassas nitelikler arasındaki bağlantıları rastgele olarak koparan bir anonimleştirme yöntemi önermiştir. Chen ve diğ. (2013), gizlilik korumalı veri madenciliğindeki bilgi ve veri bozulma problemini çözmek için geri dönüşümlü (reversible) veri ve fark gizlemeyi birleştiren bir veri pertürbasyon yöntemi sunmuştur.

Dwork (2006), gizlilik korumalı veri yayınlamada (Yang ve diğ., 2012; Zhang ve diğ., 2017) geçmiş bilgisi saldırılarına (background knowledge attack) karşı koymak için yaygın olarak kullanılan bir yaklaşım olan diferansiyel gizliliği (differential privacy) öne sürmüştür. Diferansiyel gizlilik yaklaşımı, bireysel kayıtlar içeren ve bilgi keşfini desteklemeyi amaçlayan gizlilik korumalı istatistiksel veritabanları alanında kişiye özel verilere ilişkin değerlere gürültü ekleyerek gizliliği korumaktadır (Gazeau ve diğ., 2016). Laplace dağılımından örneklenen rastgele gürültüyü kayıt sayılarına ekleyen Laplace mekanizması (Dwork ve diğ., 2006), diferansiyel gizliliği sağlamak için en sık kullanılan yaklaşımdır (Li ve diğ., 2017). Ayrıca McSherry ve Talwar (2007), diferansiyel gizliliği başarmak için çıktı kalitesini garanti eden üstel bir mekanizma sunmuştur.

Mohammed ve diğ. (2011), ileri sınıflandırma analizi için diferansiyel gizliliği ve bilgi korumayı garanti altına alan ilk genelleme tabanlı gizlilik korumalı veri yayınlama algoritmasını öne sürmüştür. Chen ve diğ. (2011), diferansiyel gizlilik gereksinimleriyle ilk yörünge veri yayınlama yaklaşımını önermiştir. Li ve diğ. (2012), diferansiyel gizliliği sağlamak için gizleme ve örnekleme işlemlerini kullanan bir *k*-anonimleştirme tekniği sunmuştur. Soria-Comas ve diğ. (2014), veri kullanılabilirliğini iyileştirmek için *k*-anonimliği ve diferansiyel gizliliği birleştiren mikro birleştirme tabanlı bir *k*-anonimlik yaklaşımı önermiştir. Fouad ve diğ. (2014), süpermodülerite ve rastgele örneklemeye dayalı bir diferansiyel gizlilik korumalı algoritma ileri sürmüştür. Wang ve Jin (2016), *kd*-ağacı algoritmasından (Xiao ve diğ., 2010) uyarlanan bir diferansiyel gizlilik çok boyutlu veri yayınlama modeli önermiştir. Zaman ve diğ. (2017), veri temizleme için genelleme işlemi ve Laplace mekanizmasını kullanan 2-katmanlı bir diferansiyel gizlilik koruma tekniği sunmuştur. Koufogiannis ve Pappas (2017), dinamik sistemlerin korunması için diferansiyel gizlilik tabanlı bir gizlilik koruma mekanizması öne sürmüştür. Li-Xin ve diğ. (2017) ise diferansiyel gizlilik veri koruması ve yayınlaması için duyarsız bir kümeleme algoritması sunmuştur.

Dong ve diğ. (2014), kurumların veri setlerini ve veri temizleme taleplerini üçüncü taraf hizmet sağlayıcılarına dışarıdan temin etmelerini sağlayan Hizmet Olarak Veri Temizleme (Data Cleaning as a Service-DCaS) için iki etkili gizlilik korumalı veri tekilleştirme (deduplication) tekniği öne sürmüştür. Bu teknikler sıklık analizi ve bilinen-şema (known-scheme) saldırılarına karşı koymaktadır.

Son zamanlarda yapılmış bir çalışmada, Nayahi ve Kavitha (2015), verilerin anonimleştirilmesi ve hassas niteliklerin korunması için benzerlik saldırılarına karşı dirençli olan (G, S) kümeleme algoritmasını önermiştir. Devamında yazarlar bu algoritmayı modifiye ederek hassas verileri; olasılıksal çıkarım saldırısı (probabilistic inference attack), bağlantı saldırısı (linking attack), homojenlik saldırısı (homogeneity attack) ve benzerlik saldırısına (similarity attack) karşı korumak için *k*-En Yakın Komşu (*k*-Nearest Neighbours-*k*-NN) tekniğini kullanan KNN-(G, S) kümeleme algoritmasını ileri sürmüştür. Bu tez çalışmasında, yukarıda bahsedilen yöntemlerden farklı olarak, büyük veride gizlilik ve kullanılabilirliği korumak için yeni bir kaos ve pertürbasyon tabanlı anonimleştirme algoritması önerilmiştir.

2.2.3.1. k-Anonimleştirme Tabanlı Teknikler

Han ve diğ. (2014), hem sayısal hem de kategorik veriler için uygun iki adımlı kümeleme tabanlı *k*-anonimleştirme algoritması önermiştir. Bu yöntem, anonimleştirme için yaygın olarak kullanılan metotlardan ikisi olan genelleme (generalization) ve mikro birleştirmeyi (microaggregation) bir araya getirmektedir. Mortazavi ve Jalili (2014), anonimleştirme için hızlı veri odaklı mikro birleştirme algoritması önermiştir. Bu algoritmada bölmeler (partitions), bilgi kaybı minimize edilerek ve *k* anonimleştirme parametresi sağlanarak oluşturulmaktadır. Gkoulalas-Divanis ve diğ. (2014), gizlilik ve kullanılabilirlik korumalı anonimleştirme algoritması önermiştir ve anonimleştirmenin kimlik ifşası saldırılarının (identity disclosure attacks) üstesinden gelmek için bilinen en iyi mekanizma olduğunu kanıtlamıştır. Wimmera ve diğ. (2016), *k*-anonimleştirmeyi kullanarak tıbbı veri üzerinde çok vekilli (multiagent) gizlilik korumalı karar verme yöntemi geliştirmiştir. Birden fazla kaynaktan entegre edilen veriler, kabul edilebilir düzeyde doğrulukla anonimleştirilmiştir. *k*-anonimleştirme; homojenlik saldırısı, benzerlik saldırısı, geçmiş bilgisi saldırısı ve olasılıksal çıkarım saldırısı gibi yaygın saldırılara karşı açıktır.

Machanavajjhala ve diğ. (2007), her eşdeğerlik (equivalence) sınıfı grubunda *l* sayıda iyi temsil edilen hassas değerleri sağlamak için *l*-çeşitlilik ilkesini önermiştir. Li ve diğ. (2007), *l*-çeşitlilik ilkesinin sınırlamalarını aşmak için *t*-yakınlık ilkesini öne sürmüştür. *l*-çeşitlilik, olasılıksal çıkarım saldırılarına karşı koruma sağlamak için tek başına yeterli değildir ve bazı hassas değerler diğerlerinden daha sık olduğunda entropi çeşitliliği (entropy diversity) elde etmek mümkün olmayacaktır. *t*-yakınlık ilkesi, oluşturulan her bir eşdeğerlik sınıfındaki hassas değerlerin dağılımını orijinal veri setindeki hassas değerlerin dağılımına yakın olmasını sağlar. Ancak *t*-yakın eşdeğerlik sınıflarını belirlemek pratik olarak mümkün değildir. Domingo-Ferrer ve Soria-Comas (2015), *k*-anonimlik ve ϵ-diferansiyel gizliliği birleştiren bir stokastik *t*-yakınlık ilkesi önermiştir. Yazarlar önerdikleri yöntem ile deterministik *t*-yakınlık ilkesinin eksikliklerinin üstesinden gelmiştir. Soria-Comas ve diğ. (2015), *k*-anonim *t*-yakın eşdeğerlik sınıflarını elde etmek için üç farklı mikro birleştirme algoritması önermiştir. *t*-yakın eşdeğerlik sınıflarını oluşturmak için kullanılan kümeleme algoritmalarından ilki minimum bilgi kaybı göstermektedir. Bu algoritmalar sadece sayısal veriler için uygundur.

Geleneksel yöntemler yaygın saldırılara karşı açıktır ve gizlilik ile kullanılabilirlik arasında daha iyi bir denge sağlamada başarısız olurlar. Anonimleştirme yapmak için veri madenciliği tekniklerinin kullanılması, bu verilerden elde edilen bilgilerin doğruluğunu artıracaktır (Ghinita ve diğ., 2011; Kisilevich ve diğ., 2011). Wen-Yang ve diğ. (2016), açgözlü bir kümeleme algoritması kullanarak olumsuz ilaç reaksiyonu verisi için MS (k, θ^2) gizlilik modeli önermiştir. Önerilen algoritma, k anonimleştirme parametresini karşılayan anonimleştirilmiş verileri oluşturur ve veri kullanılabilirliğini muhafaza eder (Nayahi ve Kavitha, 2017).

2.2.3.2. Yaygın Saldırıları Ele Alan Yöntemler

Wong ve diğ. (2006), gelişmiş bir *k*-anonimlik modeli olan (α, k) anonimliği öne sürmüştür. Bu yöntem mahremiyet korumalı veri yayınlama için *k*-anonimlik ve α-ilişkilendirme bozma (α-deassociation) özelliklerini sağlar. Her eşdeğerlik sınıfında en sık ortaya çıkan hassas değerin göreceli frekansı, kullanıcı tarafından tanımlanan α eşiğinden küçük veya ona eşit olmalıdır. Ayrıca yazarlar (α, k) anonimleştirmenin sağlanmasının NP-zor olduğunu göstermiştir. Traian ve diğ. (2007), veri yayınlama için (p, α) ve (p+, α) hassas *k*-anonimliği önermiştir. (p, α) hassas özelliği, en az α toplam ağırlıklı her bir eşdeğerlik sınıfında p farklı hassas değeri elde etmek için önerilmiştir. (p+, α) hassas özelliği ise en az α toplam ağırlıklı hassas nitelik değerlerinin en az p farklı kategorilerini elde etmek için öne sürülmüştür. (p, α)

hassas özelliği bile verileri benzerlik saldırısına karşı korumak için yeterli olmayabilir. Bu nedenle Sun ve diğ. (2008), *p*-hassas *k*-anonimliği geliştirmiştir ve hassas nitelik değerlerini en gizliden gizli olmayana doğru olacak şekilde dört kategoriye ayırmıştır. Ardından her kategoriden çeşitli hassas değerlere sahip eşdeğerlik sınıfları oluşturmuştur. Sun ve diğ. (2011), *l*-çeşitlilik ve hassas nitelik değerlerinin farklı mahremiyet düzeylerine sınıflandırılmasına dayanan (L, α) çeşitlilik gizlilik modelini önermiştir. Tian and Zhang (2011), fonksiyonel (τ, l) çeşitlilik kullanarak *l*-çeşitliliği geliştirmiştir. Hassas değerler, (τ, l) çeşitlilik sağlanıncaya kadar QI nitelikleri ile birlikte genellenirler. Yukarıdaki algoritmalarda bahsedilen tüm parametreler kullanıcı tarafından tanımlanır. Bunun yanı sıra bu parametreler eşdeğerlik sınıfının büyüklüğü için uygun eşik değerini ve farklı hassas nitelik değerlerinin sayısını ayarlamak için kullanılır. Giriş veri seti hassas nitelik için yalnızca birkaç olası değer içeriyorsa ya da bazı hassas değerler daha sık geçiyorsa, mevcut tekniklerle benzerlik saldırısının üstesinden gelmek mümkün değildir.

Sattar ve diğ. (2013), saldırganların bir bireyin hassas bilgileri üzerindeki güvenini gizlemek için örnekleme ile genellemeye dayanan gizlilik korumalı veri yayınlama yöntemi önermiştir. Amiri ve diğ. (2016), veriyi kimlik ve nitelik ifşası saldırılarına karşı korumak için *k*-anonim *β*-benzerlik (*β*-likeness) modelini öne sürmüştür. Yazarlar *k*-anonim *β*-benzerlik gizlilik modelini oluşturmak için iki kümeleme algoritması geliştirmiş ve geçmiş bilgisi saldırılarının üstesinden gelmiştir. Komishani ve diğ. (2016), yörünge verileri için anonimleştirme temelli kişiselleştirilmiş gizlilik modeli önermiştir. Bu modelde yörünge verileri gizlenirken hassas nitelikler genelleştirilmektedir. Yöntemin bağlantı ve benzerlik saldırılarının üstesinden geldiği söylenmektedir. Zhang ve diğ. (2015), anonimleştirme kullanarak bulut üzerinde veri gizliliğini korumak için iki aşamalı bir kümeleme algoritması önermiştir. Bu algoritmada hassas nitelik değerinin semantik çeşitliliği, farklı hassas değerler ile eşdeğerlik sınıflarını oluşturmak için göz önünde bulundurulmuştur (Nayahi ve Kavitha, 2017).

2.2.3.3. Güvenli Çok Taraflı Hesaplama Metotları

Literatürde aynı zamanda verilerin gizliliğini dağıtık ortamda korumaya dayalı çalışmalar da yapılmaktadır. Güvenli çok taraflı hesaplama (Lindell ve Pinkas, 2009), hem yarı dürüst hem de kötü niyetli saldırganların varlığında veri gizliliğini sağlamak için kullanılan kriptografik bir tekniktir. Bu yöntemde bir nitelik değerinin sıklığı ya da belirli bir nitelik ile ilgili istatistikler gibi yalnızca belirli bir dağıtık hesaplama için gerekli olan veriler, taraflar arasında güvenli bir

şekilde değiştirilir. Bir takım veri madenciliği algoritması, dağıtık veritabanları üzerinde yani yatay olarak bölümlendirilmiş veritabanları (Kantarcioglu ve Clifton, 2004; Xiao ve diğ., 2006) ve dikey olarak bölümlendirilmiş veritabanları (Fang ve Yang, 2008; Kantarcioglu ve Vaidya, 2003; Vaidya, 2004; Vaidya ve Clifton, 2002, 2003, 2009; Vaidya ve diğ., 2008; Yu ve diğ., 2006) üzerinde uygulanabilecek şekilde geliştirilmiştir. Yang ve diğ. (2005), sık öğe seti madenciliği (frequent itemset mining), Naive Bayes sınıflandırma ve ID3 sınıflandırmayı gerçekleştirmek için güvenli protokoller önermiştir. Yukarıdaki algoritma sınıflarında tüm veri dağıtık ortamdaki düğümler arasında asla değiştirilmez. Buna ek olarak değiştirilmiş veri yalnızca belirli bir veri madenciliği görevinde kullanılır (Nayahi ve Kavitha, 2017).

2.2.3.4. Hibrit Yaklaşımlar

Son zamanlarda araştırmacılar gizlilik sağlamak için anonimleştirmeyi diğer tekniklerle birleştiren hibrit yaklaşımlara odaklanmıştır. Kohlmayer ve diğ. (2014), dağıtık verileri anonimleştirmek için esnek bir yaklaşım öne sürmüştür. Bu yaklaşım veri gizliliğini korumak için güvenli çok taraflı hesaplama ve anonimleştirmeyi kullanmaktadır. Her katılımcı taraf şifrelenmiş veriyi bir sonraki tarafa gönderir ve nihai entegre veri bağlantı saldırılarından korunmak için anonimleştirilir. Domingo-Ferrer ve Muralidhar (2016); veriyi, saldırganı ve anonimleştirmenin şeffaflığını göz önünde bulundurarak permütasyon tabanlı veri anonimleştirme yöntemi önermiştir. Bu yöntem aynı eşdeğerlik sınıfındaki verinin çeşitliliğini hesaba katmaktadır ve dolayısıyla benzerlik saldırılarının üstesinden gelmektedir. Fakat şu an sadece sayısal veriler için uygulanabilmektedir. Goryczka ve diğ. (2014), işbirliğine dayalı veri yayınlama için anonimleştirme temelli *m*-gizlilik modelini öne sürmüştür. *m*-gizlilik modeli, anonimleştirilmiş veri setini kendi verileri ve anonimleştirilmiş veri hakkında bilgi sahibi olan *m* saldırgandan korumaktadır. Ji-Jiang ve diğ. (2015), bulut ortamında veri gizliliğini korumak için şifreleme ve anonimleştirmeyi birleştiren hibrit bir yaklaşım önermiştir. Bu yaklaşımda veriler; belirgin (explicit) tanımlayıcılar, yarı tanımlayıcı nitelikler ve tıbbi bilgiler olmak üzere üç kısma ayrılmaktadır. Tıbbi bilgiler düz metin olarak kalırken belirgin tanımlayıcılar şifrelenir ve yarı tanımlayıcılar anonimleştirilir. Ancak şifrelenmiş tanımlayıcılar ya da anonimleştirilmiş yarı tanımlayıcılar tıbbi bilgiler ile eşleştirilirse bu durum kimlik ve nitelik ifşasına yol açar. Chen-Yi (2016), verileri bozmak için geri dönüşümlü bir veri dönüştürme algoritması önermiştir. Bu algoritmada damgalama (watermarking) bozulmuş verilerin değiştirilip değiştirilmediğini kontrol etmek için kullanılır (Nayahi ve Kavitha, 2017).

2.2.4. Saldırı Türleri

Veri sahipleri tarafından yayınlanan mikro veriler açık ve gizli niteliklerden oluşmaktadır. Halka açık olan ve herkes tarafından bilinen nitelikler açık niteliklerdir. Bir bireyin kimliğini açığa çıkarmak için kötü niyetli kişiler tarafından kullanılan nitelikler yarı tanımlayıcı nitelikler olarak adlandırılır. Hassas nitelikler, veri yayınlama sırasında değerleri gizli tutulması gereken niteliklerdir (Nayahi ve Kavitha, 2017). Tablo 2.9; no'nun açık nitelik; yaş, cinsiyet ve posta kodunun yarı tanımlayıcı nitelik ve hastalığın hassas nitelik olduğu bir mikro veriyi göstermektedir.

Tablo 2.9: Örnek mikro veri.

No	Yaş	Cinsiyet	Posta kodu	Hastalık
1211	44	Kadın	50004	Gastrit
1298	42	Erkek	50006	Mide ülseri
1306	48	Kadın	50004	Akciğer kanseri
1552	46	Kadın	50009	Akciğer kanseri
1623	53	Erkek	50006	Diyabet
1744	54	Erkek	50088	Gastrit
1875	56	Erkek	50004	Mide kanseri
1939	59	Erkek	50006	Mide kanseri

2.2.4.1. Kimlik İfşası/Bağlantı Saldırısı

Saldırganlar, bir kişinin hassas nitelik değerini o kişinin yarı tanımlayıcı nitelik değerleri hakkındaki bilgisini kullanarak açığa çıkarmaya çalışmaktadır (Nayahi ve Kavitha, 2017). Örnek olarak bir saldırganın kendi mahallesinde yaşayan 53 yaşındaki bir erkeğin özel bir hastanede tedavi gördüğünü bildiğini ve hastane tarafından yayınlanan mikro veriye (Tablo 2.9) erişebildiğini varsayalım. Saldırgan bu verilere bakarak kişinin diyabet hastası olduğu sonucuna varabilir. Bu saldırı türü, kimlik ifşası veya bağlantı saldırısı olarak bilinmektedir. Tablo 2.9'da verilen mikro veriden *k*-anonimleştirme kullanılarak elde edilen 2-anonim tablo Tablo 2.10'da gösterilmektedir.

Tablo 2.10: 2-anonim gruplar.

Yaş	Cinsiyet	Posta kodu	Hastalık
[40-45]	*	5000*	Gastrit
[40-45]	*	5000*	Mide ülseri
[45-50]	Kadın	5000*	Akciğer kanseri
[45-50]	Kadın	5000*	Akciğer kanseri
[50-55]	Erkek	5000*	Diyabet
[50-55]	Erkek	500**	Gastrit
[55-60]	Erkek	5000*	Mide kanseri
[55-60]	Erkek	5000*	Mide kanseri

Tabloda genelleme ve gizleme işlemleri kullanılarak elde edilen yarı tanımlayıcı nitelikler için her biri aynı değerlere sahip dört eşdeğerlik sınıfı vardır. Anonimleştirme sırasında gizlenen değerler "*" olarak temsil edilmektedir. Her 2-anonim grup, bağlantı saldırısının üstesinden gelmektedir. 53 yaşındaki erkeği tanıyan aynı saldırganın, anonimleştirme sonrasında kişinin diyabet hastası olduğu sonucuna varması için %50 şansı vardır. Erkeğin gastrit olması için de eşit bir şans vardır. Tablo 2.10'daki tablo, bağlantı saldırısına karşı dayanıklı olmasına rağmen aşağıda verilen diğer saldırıların şansı vardır.

2.2.4.2. Homojenlik Saldırısı

Anonim bir gruptaki tüm kayıtlar hassas nitelik için aynı değerleri içerdiğinde, *k*-anonimleştirme işlemi anlamsız olmaktadır. Bu saldırı biçimine homojenlik saldırısı adı verilir (Nayahi ve Kavitha, 2017). Tablo 2.10'daki hassas nitelikleri için aynı değerlere sahip olan ikinci ve dördüncü eşdeğerlik sınıfları homojenlik saldırısına yol açmaktadır. *l*-çeşitlilik ilkesi mümkün olduğunca homojenlik saldırısının üstesinden gelmektedir. Tablo 2.11, her biri 3 çeşitli hassas değerli ($l = 3$) 4-anonim 3-çeşitli eşdeğerlik sınıfı gruplarını göstermektedir.

Tablo 2.11: 4-anonim 3-çeşitli gruplar.

Yaş	Cinsiyet	Posta kodu	Hastalık
[40-50]	*	5000*	Gastrit
[40-50]	*	5000*	Mide ülseri
[40-50]	*	5000*	Akciğer kanseri
[40-50]	*	5000*	Akciğer kanseri
[50-60]	Erkek	500**	Diyabet
[50-60]	Erkek	500**	Gastrit
[50-60]	Erkek	500**	Mide kanseri
[50-60]	Erkek	500**	Mide kanseri

2.2.4.3. Benzerlik Saldırısı

Anonim bir gruptaki hassas değerler birbiriyle aynı olmasa da, değerler birbirine benzeyebilir. Bu, değerler *l*-çeşitli olmasına rağmen hassas nitelik ifşasına yol açacaktır. Bu tür saldırılar benzerlik saldırısı olarak adlandırılır (Nayahi ve Kavitha, 2017). Tablo 2.10'daki ilk eşdeğerlik sınıfı, her iki kaydın hassas değerleri için benzer değerlere sahiptir. Saldırgan kişinin mide ile ilgili bir problemi olduğu sonucuna kolayca varabilir.

2.2.4.4. Geçmiş Bilgisi Saldırısı

Hassas nitelik veya veri setinin genel özellikleri hakkında bilgiye sahip olan bir saldırgan, bu bilgileri bir eşdeğerlik sınıfı grubundaki olası hassas değer sayısını daraltmak (sınırlandırmak) için kullanır. Bu saldırı biçimine geçmiş bilgisi saldırısı denir (Nayahi ve Kavitha, 2017).

2.2.4.5. Olasılıksal Çıkarım Saldırısı

Bir hassas nitelik değeri aynı eşdeğerlik sınıfındaki diğer değerlerden daha sık olduğunda olasılıksal çıkarım saldırısı için bir ihtimal vardır. Bu, saldırganın bir hassas niteliğin belirli QI değerleri kombinasyonu arasında daha yaygın olduğunu öğrenmesine yardımcı olur (Nayahi ve Kavitha, 2017). Bu tez çalışması kapsamında önerilen etkin gizlilik koruma algoritması, yukarıda bahsedilen tüm saldırılara karşı dayanıklıdır.

2.3. BÜYÜK VERİNİN DAĞITIKLAŞTIRILMASI

Bu bölümde Hadoop ve Cloudera ile ilgili genel bilgiler verilmektedir.

2.3.1. Hadoop

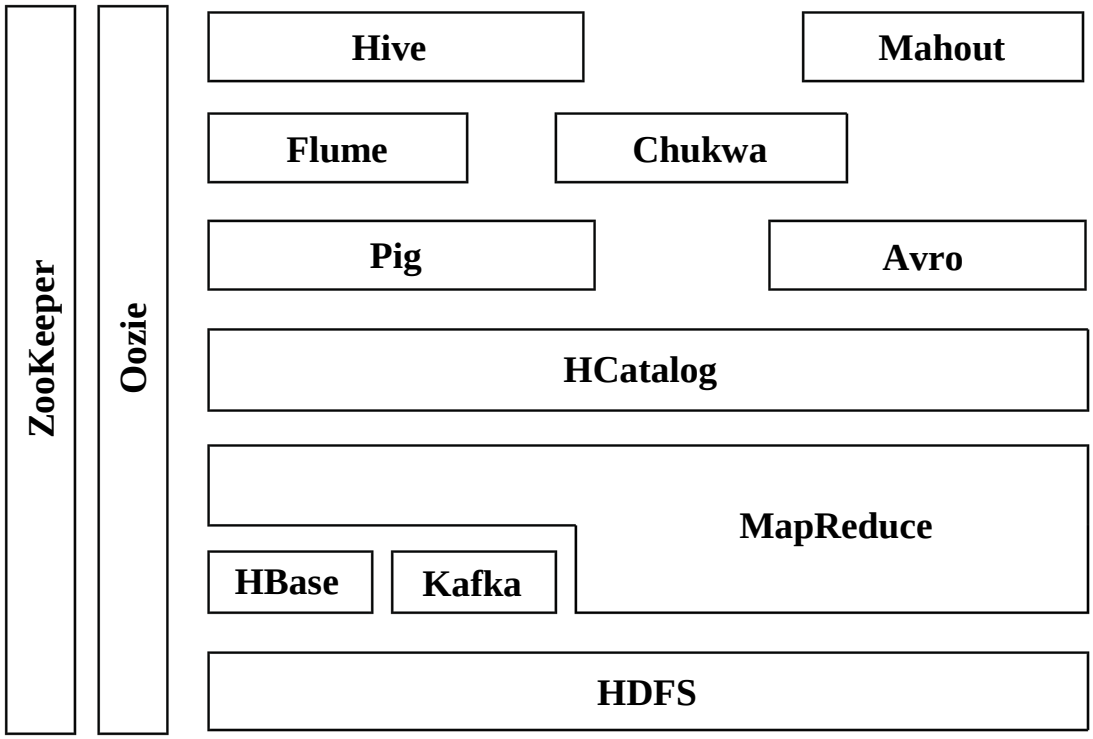
Hadoop, 2006 yılında başlayan üst düzey bir Apache projesidir ve Java'da yazılmıştır (Khan ve diğ., 2014). Apache Hadoop projesi; güvenilir, ölçeklenebilir ve dağıtık hesaplama için açık kaynaklı bir yazılım geliştirmektedir. Apache Hadoop yazılım kütüphanesi, basit programlama modellerini kullanarak bilgisayar kümeleri arasında büyük veri setlerinin dağıtık işlenmesini sağlayan bir çerçevedir. Tekil sunuculardan, her biri yerel hesaplama ve depolama sunan binlerce makineye ölçeklendirmek için tasarlanmıştır. Yüksek elverişlilik sağlamak için donanıma güvenmek yerine kütüphanenin kendisi uygulama katmanındaki hataları tespit etmek ve onlarla başa çıkmak için tasarlanmıştır. Bu nedenle her biri arızalara eğilimli olabilecek bir bilgisayar kümesi üzerinde yüksek elverişli bir hizmet sunmaktadır. Projenin içerdiği ana modüller şunlardır (Hadoop, 2018):

- Hadoop Common: Diğer Hadoop modüllerini destekleyen ortak araçlar.
- Hadoop Dağıtık Dosya Sistemi (Hadoop Distributed File System-HDFS): Uygulama verilerine yüksek verimli erişim sağlayan dağıtık dosya sistemi.
- Hadoop YARN: İş planlama ve küme kaynak yönetimi için bir çerçeve.
- Hadoop MapReduce: Büyük veri setlerinin paralel işlenmesi için YARN tabanlı bir sistem.

Apache Hadoop (Hadoop, 2018), emtia donanım kümelerini kullanarak büyük veri setlerini saklamak ve işlemek için açık kaynaklı bir çerçevedir. Hadoop, yüzlerce hatta binlerce düğümü ölçeklendirmek için tasarlanmıştır ve aynı zamanda oldukça hata toleranslıdır (Singh ve Reddy, 2014). Doug Cutting, Hadoop'u Google MapReduce programlama ortamının dağıtık bir sistemde uygulanabileceği açık kaynaklı projelerden oluşan bir koleksiyon olarak geliştirmiştir ve halen büyük miktarlardaki veriler üzerinde kullanılmaktadır. İşletmeler, daha önce yönetilmesi ve analiz edilmesi zor olan verilerden Hadoop ile yararlanabilmektedir. Hadoop, kuruluşların yaklaşık %63'ü tarafından çok sayıda yapısal olmayan logları ve olayları yönetmek için kullanılmaktadır (Khan ve diğ., 2014).

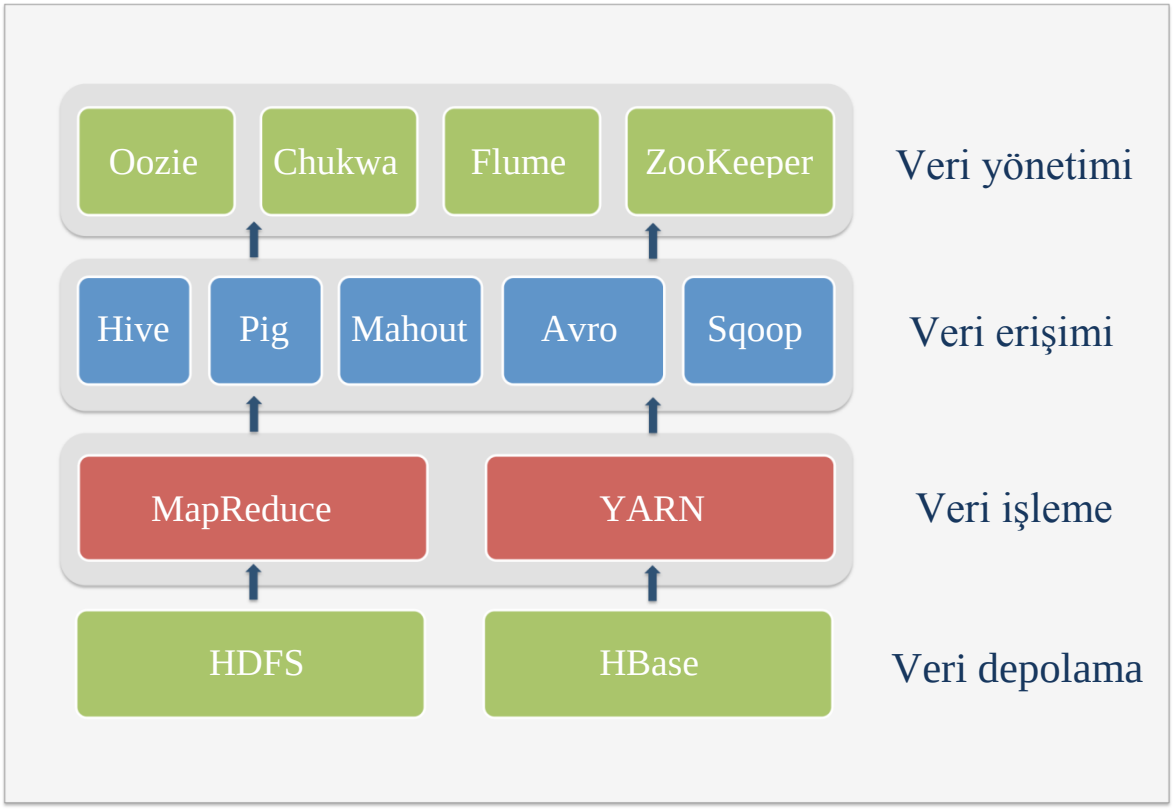
2.3.1.1. Hadoop Ekosistemi

Hadoop; HDFS ve MapReduce başta olmak üzere birçok bileşenden oluşmaktadır. Büyük veri için en yaygın ve bilinen bileşenler HDFS ve MapReduce'dur (Khan ve diğ., 2014). Hadoop ekosistemi ve bileşenlerin birbiriyle olan ilişkileri Şekil 2.9 (Khan ve diğ., 2014) ve Şekil 2.10'da (SAVVYCOM, 2018) gösterilmektedir.



Şekil 2.9: Hadoop ekosistemi.

Hadoop, özellikle değişken yapıda veya hiç bir yapıya sahip olmayan çok büyük hacimli verileri işleyebilmektedir. Aşağıdaki bölümde HDFS ve MapReduce dışındaki çeşitli Hadoop projeleri ve bunların birbirleriyle bağlantıları (Aho, 2012; Bakshi, 2012; Katal ve diğ., 2013; Khan ve diğ., 2014; Pastorelli ve diğ., 2013; Sagiroglu ve Sinanc, 2013; Wang, 2011) kaynakları incelenerek detaylandırılmıştır.



Şekil 2.10: Hadoop ekosistemi ve bileşenlerin ilişkisi.

HBase: Büyük tablolar için yapısal veri depolamayı destekleyen ölçeklenebilir, dağıtık bir veri tabanıdır (Hadoop, 2018). HBase, Google'ın BigTable'ı temelinde açık kaynaklı, sürümlü ve dağıtık bir yönetim sistemidir. Bu sistem, büyük veri setleri boyunca benzer değerler üzerindeki işlemlerin performansını hızlandıran satır tabanlıdan ziyade sütun tabanlı bir sistemdir. Örneğin okuma ve yazma işlemleri tüm satırları, ancak tüm sütunların yalnızca küçük bir alt kümesini içermektedir. HBase'e Thrift, Java ve Temsili Durum Transferi (Representational State Transfer-REST) gibi Uygulama Programlama Arayüzleri (Application Programming Interface-API) aracılığıyla erişilebilir. Bu API'lerin kendi sorguları veya komut dizisi oluşturma dilleri yoktur. HBase, varsayılan olarak tamamen bir ZooKeeper örneğine bağlıdır.

ZooKeeper: Dağıtık uygulamalar için yüksek performanslı bir koordinasyon servisidir (Hadoop, 2018). ZooKeeper, büyük miktarda veriyi sürdürür, yapılandırır ve adlandırır. Ayrıca dağıtık senkronizasyon ve grup hizmetleri sağlamaktadır. Bu örnek, bir dosya sistemi gibi paylaşılan ve hiyerarşik olan veri saklayıcılarının (register) bir ad uzayı aracılığıyla dağıtık işlemlerin birbirini yönetmesine ve katkıda bulunmasına olanak sağlamaktadır. Tek başına

ZooKeeper, efendi (master) ve köle (slave) düğümleri içeren ve yapılandırma bilgilerini depolayan dağıtık bir hizmettir.

HCatalog: HCatalog HDFS'yi yönetir, meta verileri depolar ve büyük miktarda veri için tablo oluşturur. HCatalog, Hive meta depoya bağlıdır ve meta depoyu ortak bir veri modeli kullanarak MapReduce ve Pig dâhil olmak üzere diğer hizmetlerle entegre etmektedir. HCatalog ayrıca bu veri modeli ile HBase'e genişleyebilmektedir. HCatalog, kullanıcı iletişimini HDFS verilerini kullanarak basitleştirir ve araçlar ile yürütme platformları arasında bir veri paylaşımı kaynağıdır.

Hive: Veri özetlemesi ve geçici sorgulama sağlayan bir veri ambarı altyapısıdır (Hadoop, 2018). Hive; HDFS'deki ve Amazon S3 gibi diğer giriş kaynaklarındaki ambarları yapılandırır. Hive, Hadoop ekosistemindeki bir alt platformdur ve kendi sorgu dilini (HiveQL) üretmektedir. Bu dil, MapReduce tarafından derlenmiştir ve Kullanıcı Tanımlı Fonksiyonlara (User-Defined Function-UDF) olanak sağlamaktadır. Hive platformu temel olarak üç ilgili veri yapısına dayanmaktadır. Bunlar tablolar, bölüntüler (partitions) ve kovalardır (buckets). Tablolar, HDFS dizinlerine (directories) karşılık gelir ve çeşitli bölüntülerde ve sonunda kovalarda dağıtılabilir.

Pig: Paralel hesaplama için yüksek seviyeli bir veri akışı dili ve yürütme çerçevesidir (Hadoop, 2018). Pig çerçevesi, yüksek seviyeli bir komut dizisi oluşturma dili (Pig Latin) üretir ve kullanıcıların Hadoop'ta MapReduce'u çalıştırmasını sağlayan bir yürütüm zamanı (run time) platformu kullanır. Pig, veri modeli göz önüne alındığında potansiyel veri formatı bakımından Hive'den daha esnektir. Pig'in JSON ve XML de dâhil olmak üzere yarı yapısal verileri temsil eden kendi veri türü, yani haritası vardır.

Mahout: Ölçeklenebilir bir makine öğrenmesi ve veri madenciliği kütüphanesidir (Hadoop, 2018). Toplu filtreleme, kategorize etme, kümeleme ve paralel sık desenlerin madenciliği olmak üzere dört ana gruba ayrılmıştır. Mahout kütüphanesi; dağıtık biçimde ve MapReduce ile yürütülebilen alt kümeye aittir.

Oozie: Apache Hadoop işlerini yönetmek için bir iş akışı planlayıcı sistemidir (Hadoop, 2018). Oozie, Hadoop sisteminde iş akışını koordine eder, yürütür ve yönetir. Hive, Pig, Java MapReduce, Akış MapReduce ve Distcp Sqoop gibi diğer Apache Hadoop çerçevelerine dâhil edilmiştir. Oozie, eylemleri birleştirir ve Yönlendirilmiş Çevrimsiz Çizge (Directed Acyclic

Graph-DAG) kullanarak Hadoop görevlerini düzenler. Bu model, çeşitli görevler için yaygın olarak kullanılmaktadır.

Avro: Bir veri serileştirme sistemidir (Hadoop, 2018). Avro verileri seri hale getirir, uzak prosedür çağrılarını yürütür ve verileri bir programdan veya dilden diğerine geçirir. Bu çerçevede, veriler kendi kendini tanımlamaktadır ve her zaman kendi şemalarına göre depolanmaktadır. Çünkü bu nitelikler özellikle Pig gibi komut dizisi oluşturma dilleri için uygundur.

Chukwa: Büyük dağıtık sistemleri yönetmek için bir veri toplama sistemidir (Hadoop, 2018). MapReduce ve HDFS ile ilgili veri toplama ve analiz için bir çerçevedir ve şu anda geliştirme aşamasındadır. Chukwa, dağıtık sistemlerden veri toplar, işler ve bu verileri Hadoop'ta depolar. Chukwa bağımsız bir modül olarak Apache Hadoop'un dağıtımına dâhildir.

Flume: Büyük miktarlardaki log verilerinin verimli bir şekilde toplanması, bir araya getirilmesi ve taşınması için dağıtık, güvenilir ve elverişli bir hizmettir (Flume, 2018). Flume, Hadoop içine ve dışına büyük miktarlarda log verisi toplamak ve aktarmak için özel olarak kullanılır. Flume, kaynaklar (sources) ve alıcılar (çıkış düğümü-sinks) olmak üzere iki kanal kullanmaktadır. Kaynaklar; Avro, dosyalar ve sistem loglarını içerirken, alıcılar; HDFS ve HBase'i ifade etmektedir. Flume, sorgu işleme için kendi kişisel motoru sayesinde, her yeni büyük veri yığınını alıcıya yerleştirilmeden önce dönüştürür (Khan ve diğ., 2014).

Kafka: Kafka; hızlı, ölçeklenebilir, dayanıklı ve hata toleranslı bir mesajlaşma sistemidir. Kafka, gerçek zamanlı analiz ve akış verilerinin işlenmesi için Apache Storm, Apache HBase ve Apache Spark ile birlikte çalışmaktadır. Örnek olarak Kafka, kasalı kamyon filolarından gelen coğrafi verileri ya da ofis binalarındaki ısıtma ve soğutma ekipmanlarından gelen sensör verilerini mesajlaştırabilmektedir. Kafka; endüstri veya kullanım durumu ne olursa olsun, Kurumsal Apache Hadoop'ta düşük gecikmeli analiz için büyük mesaj akışlarına aracılık yapmaktadır. Kafka; yüksek verimlilik, güvenilir dağıtım ve yatay ölçeklenebilirliğin önemli olduğu senaryolar için genel amaçlı bir mesajlaşma sistemi olarak çok çeşitli kullanım durumlarını desteklemektedir. Apache Storm ve Apache HBase'in her ikisi de Kafka ile birlikte çok iyi çalışmaktadır. Akış işleme, Web sitesi aktivite takibi, metrik toplama ve izleme ve log toplama yaygın kullanım durumlarıdır (Kafka, 2018).

Ambari: Hadoop HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig ve Sqoop için destek içeren Apache Hadoop kümelerinin hazırlanması, yönetilmesi ve izlenmesi için web tabanlı bir araçtır. Ambari, ısı haritaları gibi küme sağlığını görüntülemek için bir gösterge tablosu sunmaktadır. Ayrıca MapReduce, Pig ve Hive uygulamalarının performans özelliklerinin kullanışlı bir şekilde teşhis edilmesi için görsel olarak görüntüleme yeteneği sağlamaktadır (Hadoop, 2018).

Cassandra: Tek bir hata noktası olmayan, ölçeklenebilir bir çok yöneticili veri tabanıdır (Hadoop, 2018). Apache Cassandra veri tabanı, performanstan ödün vermeden ölçeklenebilirliğe ve yüksek elverişliliğe ihtiyaç duyulduğunda doğru seçimdir. Emtia donanımı veya bulut altyapısı üzerinde doğrusal ölçeklenebilirlik ve kanıtlanmış hata toleransı, Cassandra'yı kritik görev verileri için mükemmel bir platform haline getirmektedir. Cassandra'nın çoklu veri merkezlerinde çoğaltma desteği, kullanıcılar için daha düşük gecikme süresi ve bölgesel kesintilerden kurtulabilme açısından sınıfının en iyisidir (Cassandra, 2018).

Spark: Hadoop verileri için hızlı ve genel bir hesaplama motorudur. Spark; ETL, makine öğrenmesi, akış işleme ve grafik hesaplama gibi çok çeşitli uygulamaları destekleyen basit ve etkileyici bir programlama modeli sunmaktadır (Hadoop, 2018).

Tez: Hadoop YARN üzerine kurulu, hem toplu hem de etkileşimli kullanım durumlarında verileri işlemek için görevlerin rastgele bir DAG'ını yürütmek üzere güçlü ve esnek bir motor sağlayan genelleştirilmiş bir veri akışı programlama çerçevesidir. Tez, Hadoop ekosistemindeki Hive, Pig ve diğer çerçeveler tarafından ve aynı zamanda Hadoop MapReduce'un temel yürütme motoru olarak değiştirilmesi için ETL araçları gibi başka ticari yazılımlar tarafından benimsenmektedir (Hadoop, 2018).

Sqoop: Apache Hadoop ve ilişkisel veritabanları gibi yapısal veri depoları arasında yığınsal verileri etkin şekilde aktarmak için tasarlanmış bir araçtır (Sqoop, 2018).

Yukarıda bahsedilen çeşitli Hadoop bileşenlerinin işlevselliği Tablo 2.12'de özetlenmektedir.

Tablo 2.12: Hadoop bileşenleri ve işlevleri.

Hadoop bileşenleri	İşlevler
HDFS	Dağıtık dosya sistemi, depolama ve çoğaltma
MapReduce	Küme yönetimi, dağıtık işleme ve hata toleransı
HBase	Yapısal veri depolama, hızlı okuma/yazma erişimi
ZooKeeper	Koordinasyon ve yönetim
HCatalog	Meta veri depolama ve tablo oluşturma
Hive	SQL, veri özetlemesi ve sorgulama
Pig	Veri akışı dili ve komut dizisi oluşturma
Mahout	Makine öğrenmesi ve veri madenciliği
Oozie	İş akışı planlama ve izleme
Avro	Veri serileştirme
Chukwa	Veri toplama ve analiz
Flume	Log verisi bir araya getirme ve taşıma
Kafka	Mesajlaşma ve veri entegrasyonu
Ambari	Kümelerin hazırlanması, yönetilmesi ve izlenmesi
Cassandra	Ölçeklenebilir ve hata toleranslı veri tabanı
Spark	Hesaplama motoru
Tez	Veri akışı programlama çerçevesi
Sqoop	Hadoop ve yapısal veri depoları arasında veri aktarımı

Hadoop büyük veri ile spam filtreleme, ağ arama, tıklama analizi ve sosyal öneri gibi endüstriyel uygulamalarda yaygın olarak kullanılmaktadır. Yahoo; spam filtreleme ve arama gibi ürün ve hizmetlerini dağıtmak için 2012 yılının haziran ayından itibaren Hadoop'u dört veri merkezinde 42000 sunucuda çalıştırmaktadır. 2014 yılında, en büyük Hadoop kümesi 4000 düğüm içermektedir. Hadoop 2.0 sürümü ile bu kümenin 10000 düğüme çıkması beklenmektedir (Chen ve diğ., 2014). Facebook, Hadoop kümesinin 2012 yılının kasım ayı itibariyle günde 0,5 PB oranında artışla 100 PB veri işlediğini açıklamıştır. Wiki 2013 yılında bazı tanınmış şirketlerin ve ajansların da dağıtık hesaplamayı desteklemek için Hadoop'u kullandığını belirtmiştir. Ayrıca Cloudera, EMC, MapR, IBM ve Oracle gibi çeşitli şirketler Hadoop'u ticari olarak uygulamaktadır ve/veya destek sağlamaktadır (Khan ve diğ., 2014).

SYS-CON Media şirketinin 2011 yılındaki açıklamasına göre kullanıcıların %94'ü Hadoop ile büyük miktarlardaki verileri analiz edebilmektedir. Buna ek olarak kullanıcıların %88'i veriler üzerinde ayrıntılı analiz yapabilmekte ve %82'si de daha fazla veriyi muhafaza edebilmektedir. Her şirket kendi ihtiyacına göre Tablo 2.12'deki Hadoop projeleri arasından birini kullanmaktadır. Facebook, Hadoop kullanarak 100 PB boyutunda hem yapısal hem de yapısal olmayan veri depolamaktadır. IBM ise öncelikli olarak erişilebilir, ölçeklenebilir, etkili ve kullanıcı dostu bir Hadoop platformu oluşturmayı amaçlamaktadır. Ayrıca gelişmiş bir analitik uygulama için geliştirme ve çalışma zamanı ortamları oluşturarak büyük veri analitiği ile ilişkili zaman-değer eğrisini düzleştirmeyi ve işletme kullanıcıları için büyük veri analitik araçları sağlamayı hedeflemektedir. Tablo 2.13 (Khan ve diğ., 2014), şirketlerin Hadoop'u hangi alanlarda kullandığını göstermektedir.

Tablo 2.13: Şirketlerin Hadoop'u kullanım alanları.

Kullanım alanı	**Kullanan şirket**
Arama	Yahoo, Amazon, Zvents
Log işleme	Facebook, Yahoo, ContexWeb.Joost, Last.fm
Video ve görüntü analizi	New York Times, Eyelike
Veri ambarı	Facebook, AOL
Tavsiye sistemleri	Facebook

2.3.1.2. HDFS ve MapReduce

Hadoop, Java ile yazılmış ve büyük veri setlerinin emtia kümeleri boyunca dağıtık işlenmesini sağlayan açık kaynaklı bir Apache projesidir. Hadoop'un, HDFS ve MapReduce programlama çerçevesi olmak üzere iki ana bileşeni vardır. Hadoop'un en önemli özelliği HDFS ve MapReduce'un birbiriyle yakından ilişkili olmasıdır. Her biri, tek bir küme üretilecek şekilde birlikte uygulanır (White, 2009). Bu nedenle depolama sistemi fiziksel olarak işleme sisteminden ayrı değildir (Hashem ve diğ., 2015).

HDFS, küme düğümlerinin yerel dosya sistemleri üstünde çalışmak ve akış verilerine erişmek için çok büyük dosyaları depolamak üzere tasarlanmış dağıtık bir dosya sistemidir. Oldukça hataya dayanıklıdır ve tek bir sunucudan, her biri yerel hesaplama ve depolama sunan binlerce makineye kadar ölçeklenebilir. HDFS, efendi adında bir isim düğümü ve köleler olarak

adlandırılan çeşitli veri düğümleri olmak üzere iki tür düğümden oluşmaktadır. Ayrıca ikincil isim düğümlerini de içerebilir. İsim düğümü; dosya sistemleri ve yönetici ad uzayının (namespace) hiyerarşisini yönetmektedir. Dosya sistemleri; erişim zamanı, değişiklik, izin ve disk alanı kotaları gibi nitelikleri kaydeden bir isim düğümü biçiminde sunulmaktadır. Dosya içeriği büyük bloklara bölünür ve dosyanın her bir bloğu yedekleme için veri düğümleri arasında bağımsız olarak çoğaltılır ve mevcut tüm blokların bir raporu isim düğümüne periyodik olarak gönderilir (Hashem ve diğ., 2015; Shvachko ve diğ., 2010).

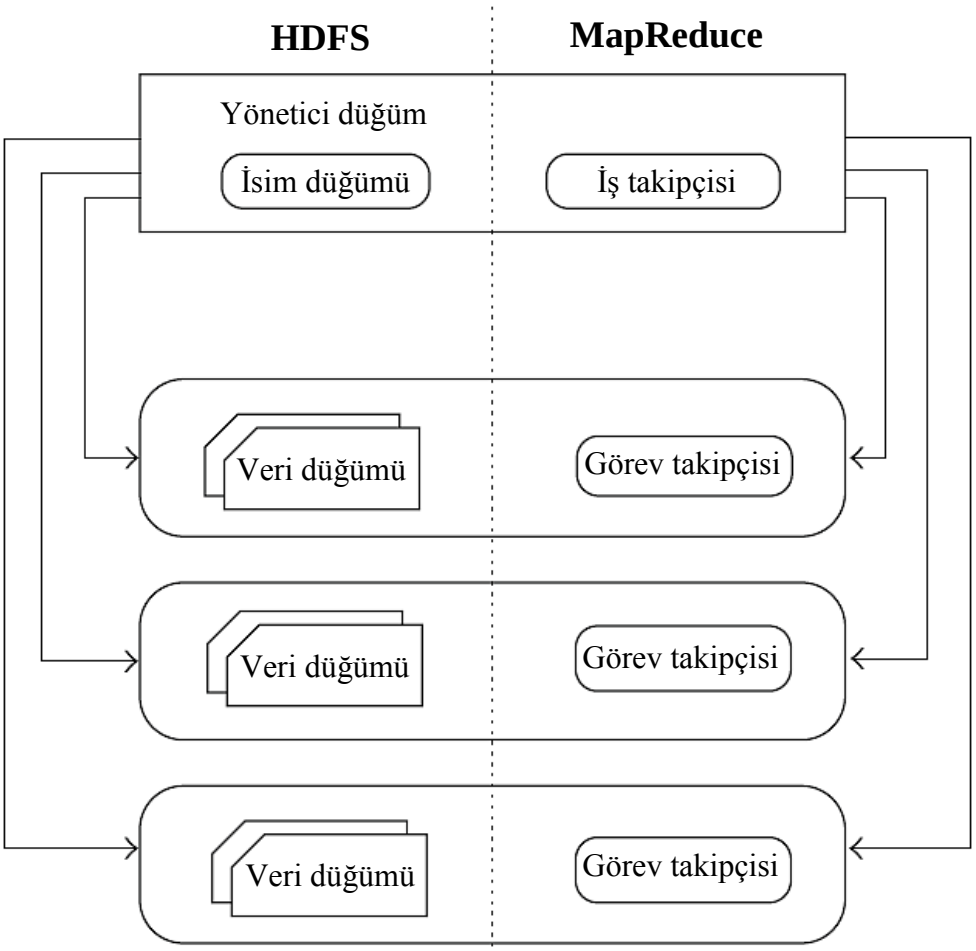
MapReduce, yoğun veri uygulamalarında çok sayıda veri setini işlemek için Google tarafından öncülük edilen basitleştirilmiş bir programlama modelidir. MapReduce modeli, Google Dosya Sistemine (Google File System-GFS) (Ghemawat ve diğ., 2003) dayalı olarak geliştirilmiş ve açık kaynaklı Hadoop uygulaması ile kullanılmıştır. MapReduce, deneyimsiz bir programcının paralel programlar geliştirmesine ve bulutta bilgisayar kullanabilen bir program oluşturmasına olanak tanımaktadır. Programcıların çoğu durumda sadece iki fonksiyonu belirlemesi gerekmektedir. Bunlar fonksiyonel programlamada yaygın olarak kullanılan haritalama fonksiyonu (haritalayıcı, mapper) ve indirgeme fonksiyonudur (indirgeyici, reducer). Haritalayıcı, anahtar/değer çiftini girdi olarak kabul eder ve ara anahtar/değer çiftleri üretir. İndirgeyici aynı ara anahtarla ilişkili tüm çiftleri birleştirir ve daha sonra bir çıkış üretir (Dean ve Ghemawat, 2008). Tablo 2.14 (Hashem ve diğ., 2015), haritala (map)/indirge (reduce) fonksiyon sürecini özetlemektedir.

Tablo 2.14: Haritala/indirge fonksiyon sürecinin özeti.

Haritala/indirge fonksiyonu
Haritalayıcı (anahtar1, değer1) → Liste [(anahtar2, değer2)]
İndirgeyici [anahtar2, liste (değer2)] → Liste (anahtar3, değer3)

Tablo 2.14'te görüldüğü üzere haritalama fonksiyonu, giriş alanının üretilen çıkış çiftleri listesinden (anahtar2, değer2) farklı olduğu her bir girişe (anahtar1, değer1) uygulanır. Listenin elemanları (anahtar2, değer2) daha sonra bir anahtarla gruplandırılır. Gruplandırmadan sonra liste (anahtar2, değer2) çeşitli listelere [anahtar2, liste (değer2)] bölünür ve indirgeme fonksiyonu en son sonuç listesini (anahtar3, değer3) oluşturmak için her bir listeye [anahtar2, liste (değer2)] uygulanır (Hashem ve diğ., 2015).

Haritala ve indirge fonksiyonları büyük verinin işlenmesini ölçeklendirmek için büyük veri setlerinin küçük alt setleri üzerinde gerçekleştirilebilir (Azzini ve Ceravolo, 2013; O'Driscoll ve diğ., 2013). Veriler bir Hadoop kümesinde daha küçük bloklara dönüştürülür. Bu bloklar küme boyunca dağıtılır. HDFS bu işlevi sağlar ve tasarımında dağıtık dosya sistemi olan GFS'den aşırı derecede esinlenilmiştir. HDFS ve MapReduce mimarileri Şekil 2.11'de (Khan ve diğ., 2014) gösterilmektedir.



Şekil 2.11: HDFS ve MapReduce'un sistem mimarileri.

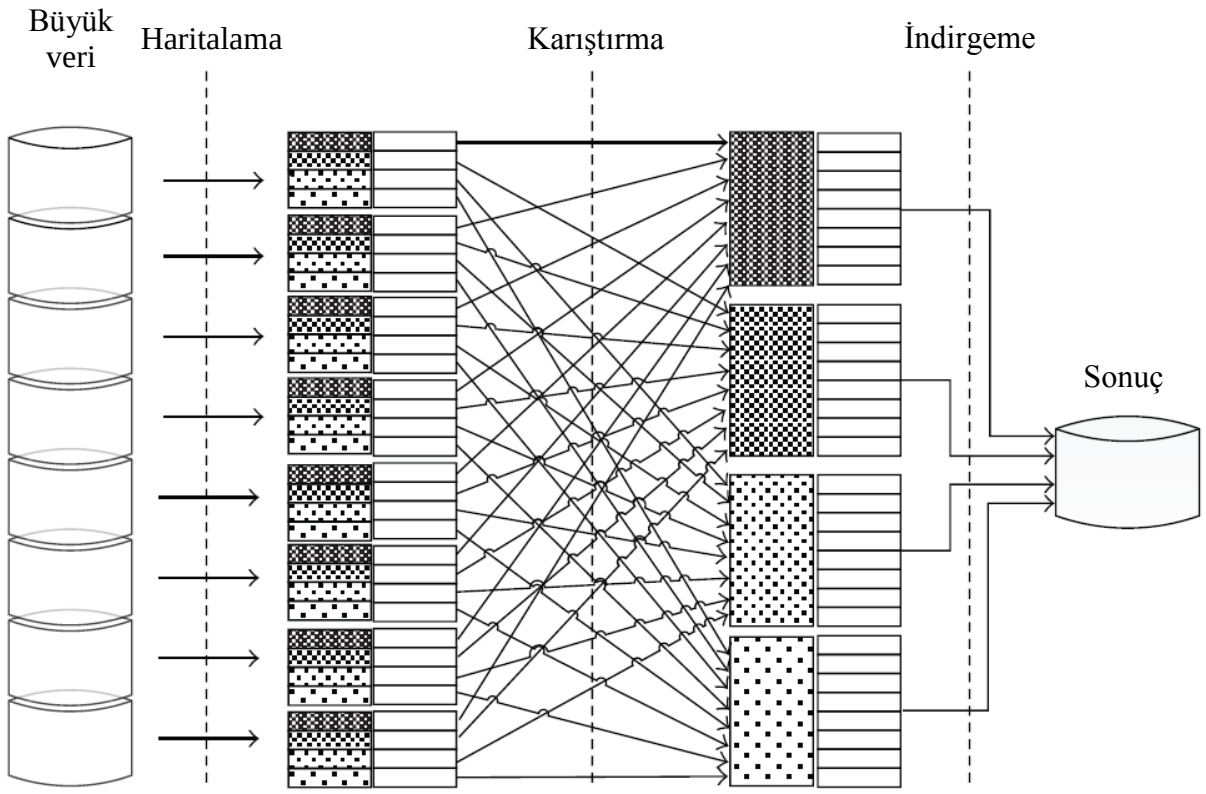
MapReduce, Hadoop'un merkezidir ve Hadoop kümesinde çok sayıda sunucuda toplu ölçeklenebilirlik sağlayan bir programlama yaklaşımıdır. Her sunucu bu kümede pahalı olmayan bir dizi dahili disk sürücüsü içermektedir. MapReduce, performansı artırmak için işlenmiş verilerin depolandığı sunuculara iş yükleri atamaktadır. Veri işleme küme düğümlerine bağlı olarak planlanır. Bir düğüme, bu düğüme yabancı veri gerektiren bir görev atanabilir. MapReduce'un işlevselliği (Azzini ve Ceravolo, 2013; O'Driscoll ve diğ., 2013)'de ayrıntılı olarak ele alınmıştır.

MapReduce aslında Hadoop programları tarafından gerçekleştirilen iki ayrı işe karşılık gelmektedir. Birincisi, bir veri setini elde etmeyi ve onu başka bir veri setine dönüştürmeyi içeren haritalama işidir. Bireysel bileşenler bu veri setlerinde anahtar/değer çiftlerine yani gruplara (tuples) dönüştürülür. İndirgeme görevi, haritalama çıktılarından gelen girdileri alır ve daha sonra veri gruplarını küçük grup setlerine böler. Bu nedenle indirgeme görevi her zaman haritalama işinden sonra gerçekleştirilir. MapReduce görevleri Tablo 2.15'te (Khan ve diğ., 2014) özetlenmektedir.

Tablo 2.15: MapReduce görevleri.

Adımlar	**Görevler**
Giriş	1) Veriler bloklar halinde HDFS'ye yüklenir ve veri düğümlerine dağıtılır 2) Bloklar hata durumunda çoğaltılır 3) İsim düğümü blokları ve veri düğümlerini izler
İş	1) İşi ve ayrıntılarını iş takipçisine gönderir
İş başlatma	1) İş takipçisi her veri düğümünde görev takipçisi ile etkileşime girer 2) Tüm görevler planlanır
Haritalama	1) Haritacı veri bloklarını işler 2) Anahtar değer çiftleri listelenir
Sıralama	1) İşi ve ayrıntılarını iş takipçisine gönderir
Karıştırma	1) Haritalanmış çıkış indirgeyiciye aktarılır 2) Değerler sıralı biçimde yeniden düzenlenir
İndirgeme	1) İndirgeyici nihai sonucu oluşturmak için anahtar değer çiftleri listesini birleştirir
Sonuç	1) Değerler HDFS'de depolanır 2) Sonuçlar yapılandırmaya göre çoğaltılır 3) İstemciler HDFS'den sonuçları okur

Fazlalık (redundant) veriler küme üzerinde birden çok alanda saklanmaktadır. Programlama modeli, programın bölümlerini kümedeki çeşitli sunucularda çalıştırarak hataları otomatik olarak çözmektedir. Veriler, veri fazlalığı göz önüne alındığında ilişkili programlama ile birlikte çok büyük bir emtia bileşeni kümesi boyunca dağıtılabilir. Bu fazlalık ayrıca hataları tolere eder ve özellikle büyük miktarda veri göz önünde bulundurulduğunda emtia donanım bileşeni bozulursa, Hadoop kümesinin kendini tamir etmesini sağlar. Hadoop, bu süreçte büyük veri sorunlarıyla ilgili iş yüklerini makul makinelerin büyük kümeleri arasında devredebilir. MapReduce mimarisi Şekil 2.12'de (Khan ve diğ., 2014) gösterilmektedir.



Şekil 2.12: MapReduce mimarisi.

2.3.2. Cloudera

Çeşitli bilgi teknolojileri sağlayıcıları ve toplulukları Hadoop altyapısını, araçlarını ve hizmetlerini geliştirmek ve zenginleştirmek için çalışmaktadır. Açık kaynaklı modüller aracılığıyla büyük veri yeniliklerinin paylaşılması yararlıdır ve büyük veri teknolojilerini desteklemektedir. Buradaki olumsuzluk, kullanıcıların farklı kaynaklardan çeşitli modül versiyonlarından oluşan bir Hadoop platformu ile karşı karşıya kalabilmesidir. Her bir Hadoop modülünün kendi olgunluk eğrisi olması sebebiyle Hadoop platformunda sürüm uyumsuzluğu riski vardır. Ayrıca çeşitli teknolojilerin aynı platformda entegrasyonu da güvenlik risklerini artırmaktadır. Genellikle her modül test edilmektedir. Ancak çoğu zaman farklı kaynaklardan gelen teknolojilerin birleşimi, tam olarak araştırılmayan veya test edilmeyen gizli riskler getirebilmektedir. Cloudera, IBM, MapR ve Hortonworks gibi pek çok bilgi teknolojisi sağlayıcısı bu sorunlara karşı koymak için kendi modüllerini geliştirmekte ve bu modülleri dağıtımlarında bir araya getirmektedir. Hedeflerden biri, tüm birleştirilmiş modüllerin

uyumluluğunu, güvenliğini ve performansını garanti altına almaktır. Mevcut Hadoop dağıtımlarının çoğu adım adım zenginleştirilmektedir. Bu dağıtımlar; dağıtık depolama sistemleri, kaynak yönetimi, koordinasyon hizmetleri, etkileşimli arama araçları, gelişmiş zeka analiz araçları gibi çeşitli hizmetler içermektedir. Ayrıca Hadoop dağıtım sağlayıcıları kendi ticari desteklerini sunmaktadır (Oussous ve diğ., 2017).

Cloudera (Azarmi, 2016), en çok kullanılan Hadoop dağıtımlarından biridir. Hadoop tarafından desteklenen bir kurumsal veri merkezi'nin konuşlandırılmasına ve yönetilmesine olanak vermektedir. Ayrıca merkezi yönetim aracı, birleşik yığın işleme, etkileşimli SQL ve rol tabanlı erişim kontrolü gibi birçok fayda sağlamaktadır. Buna ek olarak Cloudera çözümleri çok çeşitli mevcut altyapıya entegre edilebilir ve tek bir sistemdeki farklı iş yükleri ve veri formatları ile başa çıkabilir. Cloudera, Hadoop'taki verileri taramak ve sorgulamak için kolay bir yol sunmaktadır. Aslında gerçek zamanlı etkileşimli bir sorgulamanın gerçekleştirilmesi ve sonucunun uygun bir şekilde görselleştirilmesi mümkündür. Bununla birlikte güvenlik ve veri yönetimini desteklemek için çeşitli araçlar mevcuttur (Oussous ve diğ., 2017).

Cloudera'nın temel modüllerinden biri Impala'dır (Sakr, 2016). Hadoop ile uyumlu olan ilginç bir sorgu dili modülü oluşturmaktadır. Impala, verileri sütun biçimli bir veri formatında yapılandırmaktadır. Büyük veri üzerinde etkileşimli ve gerçek zamanlı analizin üstesinden gelmek için olanak sağlamaktadır. Impala, Hive'in aksine MapReduce çerçevesini kullanmamaktadır. MapReduce yerine büyük veri setleri üzerinde hızlı sorguları sağlamak için kendi bellek içi işleme motorunu kullanmaktadır. Bu nedenle Impala, sorgulama sonuçları döndürmede Hive'den daha hızlıdır. Aslında AMPLab Shark projesi gibi Impala (Manoochehri, 2013), mevcut HDFS ve HBase kaynaklarından gelen verileri doğrudan kullanabilmektedir. Böylece veri hareketini en aza indirir ve dolayısıyla büyük sorguların yürütme süresini azaltır. Impala, hata toleransı için depoları çift kopya olarak çoğaltır ve başlıca iş zekası araçlarıyla entegrasyona olanak tanır. Kimlik doğrulama için Kerberos ve rol tabanlı yetkilendirme için Apache Sentry'e dayanan yerel Hadoop güvenliğini de içermektedir (Menon, 2014; Oussous ve diğ., 2017).

Cloudera (Prasad ve Agarwal, 2016) ayrıca yapısal ve yapısal olmayan verileri destekleyen esnek bir model sağlamaktadır. Cloudera Hive'den daha hızlıdır. Örnek olarak Hive/MapReduce'dan en az 10 kat daha hızlı sorgu yürütmektedir. Hive Sorgu Dili (Hive Query Language-HiveQL) ile karşılaştırıldığında, Cloudera en az bir joinli sorgular için 7 ila 45 kat

performans kazancı sağlamaktadır. Birleştirme (aggregation) sorguları bile yaklaşık 20-90 kat hızlandırılmıştır. Cloudera ayrıca gerçek zamanlı yanıt verme açısından HiveQL veya MapReduce'u geride bırakmaktadır. Aslında Cloudera Enterprise (Kurumsal) sürümü HiveQL veya MapReduce kullanarak sorguların yanıt süresini dakika yerine saniyelere düşürmektedir (Oussous ve diğ., 2017).

3. MALZEME VE YÖNTEM

Bu bölüm; tez çalışması kapsamında kullanılan veri setlerini, veri seti bölümleme yöntemini, performans metriklerini, sınıflandırma yöntemlerini, önerilen etkin gizlilik koruma algoritmasını ve gizliliği korunmuş veri setlerinin dağıtıklaştırılmasını içermektedir.

3.1. VERİ SETLERİ

Önerilen algoritmanın performansı, 1994 ABD nüfus sayımı veri tabanından çıkarılan Yetişkin veri seti (Adult data set) (Kohavi ve Becker, 1996) üzerinde değerlendirilmiştir. Bu veri setinin bu çalışmada kullanılmasının nedeni, literatürde algoritmaların gizlilik analizi için benchmark olarak kullanılmasıdır. Ayrıca veri seti California Üniversitesi-Irvine, Makine Öğrenmesi Deposunda (Machine Learning Repository) (Lichman, 2013) online olarak mevcuttur. Yetişkin veri seti 32561 kayıt içermektedir ve eksik değeri olmayan toplam kayıt sayısı 30162'dir. Veri seti bu çalışmada kullanılmadan önce eksik değer içeren kayıtlar silinmiştir. Veri setindeki nitelik sayısı 6 sayısal ve 9 kategorik olmak üzere 15'tir. Veri setinde 7508 örnek "> 50K" sınıfında, 22654 örnek ise "≤ 50K" sınıfındadır. Yetişkin veri setinin detaylı tanıtımı Tablo 3.1'de gösterilmektedir.

Yetişkin veri seti, önerilen algoritmanın büyük veri üzerinde ölçeklenebilirliğini göstermek için sırasıyla ~60K, ~120K, ~240K ve ~480K kayıtlı dört veri seti olacak şekilde eşit oranda genişletilmiştir. Ayrıca verilerin iki katına çıkarılması, Yetişkin veri setinin *k*-anonim formları üzerinde $k = 2, 4, 8$ ve 16'yı sağlayacak şekilde önerilen algoritmanın sınıflandırma doğruluğu, F-ölçütü ve yürütme süresi performansını değerlendirmek için veri bütünlüğünü bozmadan eşit olarak gerçekleştirilir. Önerilen algoritmanın performansını mevcut algoritmalar ile karşılaştırmak için "yaş", "ırk" ve "cinsiyet" nitelikleri yarı tanımlayıcı olarak seçilmiştir. Ek olarak "gelir" ve "meslek" nitelikleri ise iki ayrı test durumu için hassas nitelik olarak belirlenmiştir. Yetişkin veri seti üzerindeki test durumları Tablo 3.2'de gösterilmektedir.

Tablo 3.1: Yetişkin veri setinin detaylı tanıtımı.

Nitelik	Nitelik türü	Tanım kümesi
Yaş (age)	Sayısal	[17-90]
İş sınıfı (workclass)	Kategorik	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked
Fnlwgt	Sayısal	[19214-1226583]
Eğitim (education)	Kategorik	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool
Eğitim numarası (education-num)	Sayısal	[1-16]
Medeni hal (marital-status)	Kategorik	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
Meslek (occupation)	Kategorik	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces
İlişki durumu (relationship)	Kategorik	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried
Irk (race)	Kategorik	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
Cinsiyet (sex)	Kategorik	Female, Male
Sermaye kazancı (capital-gain)	Sayısal	[0-99999]
Sermaye zararı (capital-loss)	Sayısal	[0-4356]
Haftalık saat (hours-per-week)	Sayısal	[1-99]
Ana vatan (native-country)	Kategorik	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands
Gelir (income) (sınıf niteliği)	Kategorik	"> 50K" ve "≤ 50K"

Tablo 3.2: Yetişkin veri seti üzerindeki test durumları.

Test durumu	Yarı tanımlayıcı		Hassas nitelik	
	İsim	**Benzersiz değer sayısı**	**İsim**	**Benzersiz değer sayısı**
I	Yaş	72	Gelir	2
	Irk	5		
	Cinsiyet	2		
II	Yaş	72	Meslek	14
	Irk	5		
	Cinsiyet	2		

3.2. VERİ SETİ BÖLÜMLEME

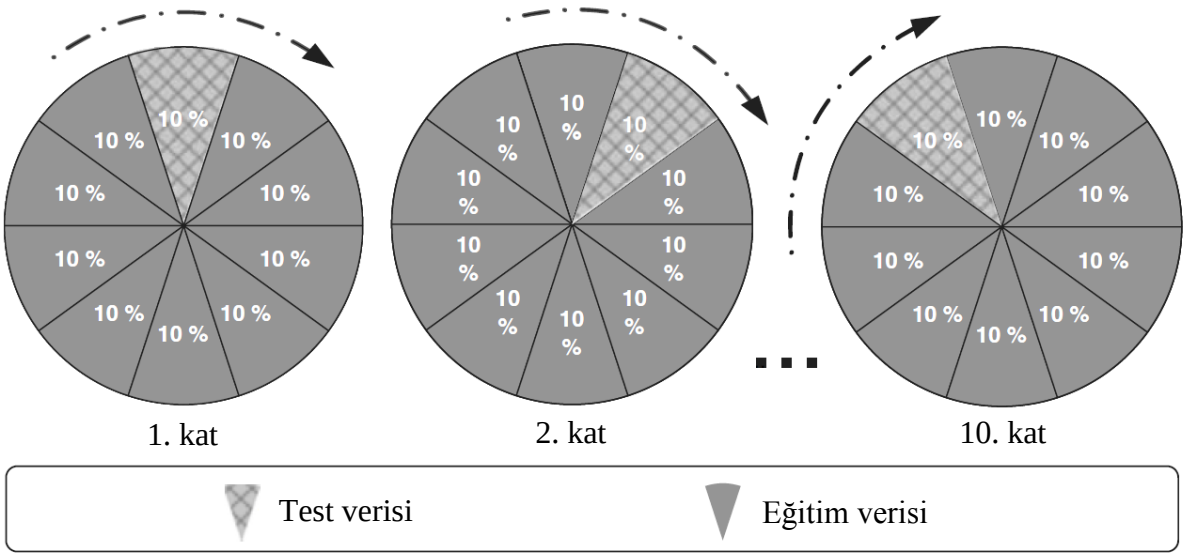
Bu çalışmada önerilen algoritmanın sınıflandırma doğruluğu ve F-ölçütü performansları değerlendirilirken kullanılan veri setleri *k*-kat çapraz geçerleme yöntemi ile bölümlendirilmiştir.

3.2.1. k-Kat Çapraz Geçerleme

k-kat çapraz geçerlemede (*k*-fold cross validation), tüm veri seti *k* sayıda eşit boyutlu ayrışık alt setlere (katlar) rastgele olarak bölünür. Sınıflandırma modeli eğitilir ve *k* kez test edilir. Her defasında *k* – 1 kat eğitim ve geriye kalan 1 kat test için kullanılır. Bir modelin genel doğruluğunun çapraz geçerleme tahmini, *k* sayıda ayrı ölçümlerin basitçe ortalaması alınarak hesaplanır (3.1).

$$\text{Çapraz geçerleme doğruluğu} = \frac{1}{k}\sum_{i=1}^{k} A_i \tag{3.1}$$

Burada *k*, kullanılan kat sayısı ve *A*, her bir katın doğruluk ölçümüdür. *k*-kat çapraz geçerlemede *k* değerinin 10 olarak seçilmesi en yaygın uygulamadır. Deneysel (empirical) çalışmalar ideal (optimal) *k* kat sayısının 10 olduğunu göstermektedir (Olson ve Delen, 2008). *k*'nın 10 olduğu *k*-kat çapraz geçerlemenin görselleştirilmesi Şekil 3.1'de (Olson ve Delen, 2008) verilmektedir.



Şekil 3.1: 10-kat çapraz geçerlemenin görselleştirilmesi.

k-kat çapraz geçerlemenin gerçekleştirilmesinde izlenecek süreç aşağıdaki gibidir:

Adım 1: Tüm veri seti her biri neredeyse aynı sayıda kayıt içeren *k* ayrık alt sete (kat) rastgele olarak bölünür. Örnekleme, sınıfların orantılı temsilinin orijinal veri setindekilerle yaklaşık olarak aynı olmasını sağlamak için sınıf etiketleriyle katmanlaştırılır.

Adım 2: Her bir kat için bir sınıflandırıcı mevcut kattaki kayıtlar hariç tüm kayıtlar kullanılarak oluşturulur. Daha sonra sınıflandırıcı, hata oranının çapraz geçerleme tahminini elde etmek için mevcut kat üzerinde test edilir ve sonuç kaydedilir.

Adım 3: 10 kat için adım 2 tekrarlandıktan sonra her bir model tipinin toplu sınıflandırma doğruluğu tahminini bulmak için 10 çapraz geçerleme tahminlerinin ortalaması alınır.

10-kat çapraz geçerleme, geleneksel tek bölme (single split) (2/3 eğitim ve 1/3 test) deneyine kıyasla daha fazla veri gerektirmez. Aslında veri madenciliği topluluğunda, nispeten küçük veri setleriyle yapılan yöntem karşılaştırma çalışmaları için *k* kat tipi deney metotları önerilmektedir. Esas itibariyle 10-kat (ya da herhangi bir sayıda katın) çapraz geçerlemenin ana avantajı, her seferinde verilerin ayrı bir bölümünün holdout örneği olarak kullanılmasıyla deneyi 10 kez tekrar ederek eğitim ve holdout veri örneklerinin rastgele örneklemesine ilişkin sapmayı (bias) azaltmaktır (Olson ve Delen, 2008).

3.3. PERFORMANS METRİKLERİ

Bu bölümde, önerilen gizlilik koruma algoritmasının değerlendirilmesi için kullanılan performans metrikleri sunulmaktadır. Bu metrikler Kullback-Leibler uzaklığı (bağıl entropi), olasılıksal anonimlik, sınıflandırma doğruluğu, F-ölçütü, yürütme süresi ve Impala sorgularıdır.

3.3.1. Kullback-Leibler Uzaklığı

Kullback-Leibler uzaklığı (Kullback-Leibler divergence) iki dağılım arasındaki farkı ölçmek için kullanılmaktadır (Kullback ve Leibler, 1951; Sun ve diğ., 2011). Ayrıca gizlilik korumasında, orijinal ve gizliliği korunmuş veri setleri arasındaki mesafeyi hesaplamak için kullanılır. KL uzaklık metriği şu şekilde tanımlanır;

$$\text{KL uzaklığı} = \sum_{x} p(x)\log\frac{p(x)}{q(x)} \tag{3.2}$$

Burada $p(x)$ ve $q(x)$ iki dağılımdır (Nayahi ve Kavitha, 2015). KL uzaklığı negatif değildir ve iki dağılımın aynı olması durumunda 0'dır (Li ve diğ., 2007). Bu çalışmada, $p(x)$ ve $q(x)$ dağılımları sırasıyla gizliliği korunmuş ve orijinal veri setleri için kullanılmıştır.

3.3.2. Olasılıksal Anonimlik

Olasılıksal anonimlik, gizlilik veya anonimlik için tanımlanmış ve ispatlanmış bir istatistiksel ölçümdür (Yang ve Qiao, 2010). Saldırganlar, gizliliği korunmuş bir veri setinde orijinal ilişkileri karşılık gelen ilişkilerden açığa çıkaramazlar. Olasılıksal anonimlik çıkarım yapamamayı ölçmektedir.

Tanım 1 (Olasılıksal anonimlik): Bir veri seti D'nin verildiği ve D'nin anonimleştirilmiş halinin D' olduğu varsayılsın. r, D'de bir kayıt ve $r' \in D'$, r'nin anonimleştirilmiş versiyonu olsun. $r(QI)$, r'deki yarı tanımlayıcının değer kombinasyonu olarak sembolize edilir. D''nin olasılıksal anonimliği $1/P(r(QI)|r'(QI))$ olarak tanımlanır. $P(r(QI)|r'(QI))$, tüm $r \in D$ için $r(QI)$'nin verilen $r'(QI)$'den açığa çıkarılabilme olasılığıdır. Q_i, $i = 1,\ldots, m$ D'de i. yarı tanımlayıcı nitelik ve $Entropy(Q_i)$, Q_i'nin entropi değeri olsun. D''nin olasılıksal ananomliği $Pa(D')$ olarak sembolize edilir ve şu şekilde tanımlanır:

$$Pa(D') = e^{Entropy(Q_i)} \tag{3.3}$$

$Pa(D')$, (3.4) olduğunda en yüksek değere ulaşır.

$$p_i = \frac{e^{Entropy(Q_i)}}{\sum_{j=1}^{m} e^{Entropy(Q_i)}} \tag{3.4}$$

Bu ifade olasılıksal anonimliğin hesaplanması için genel bir ölçüm olarak kullanılabilir. Ölçeklenmiş $Pa(D')$ ile ilgili bir kestirim, (3.5) olduğunda tüm yarı tanımlayıcı niteliklerin geometrik ortalamasının hesaplanmasıyla (3.6)'daki gibi yapılabilir:

$$p_i = \frac{1}{m},\ i = 1,\ldots,m \tag{3.5}$$

$$\ln Pa(D') = \ln m + \sum_{i=1}^{m} (\frac{1}{m} Entropy(Q_i)) = \ln (m(\prod_{i=1}^{m} Diversity_i)^{\frac{1}{m}}) \tag{3.6}$$

Burada $Diversity_i$ (3.7)'deki gibidir:

$$Diversity_i = e^{Entropy(Q_i)} \tag{3.7}$$

D'de rastgele bir kayıt için bir yarı tanımlayıcının orijinal değerini tahmin etme olasılığı $1/Pa(D')$ olarak hesaplanır. Ayrıca bu olasılık, bir kullanıcının hassas bir değeri bir bireyle ilişkilendirmek için olan güvenini göstermektedir. Denklem (3.6)'ya göre, $Pa(D')$, genellikle tüm yarı tanımlayıcı niteliklerin çeşitliliklerinin geometrik ortalamasından daha büyüktür. Benzer şekilde, $Pa(D')$, çoğunlukla hassas niteliğin çeşitliliğinden de daha büyüktür. Bir hassas niteliğin çeşitliliği $Diversity_s$ olsun. Bir kullanıcı, bir bireyin veri setinde bulunduğundan emin olduğunda bile kullanıcının ilgili hassasiyeti ortaya çıkarma konusundaki maksimum güveni $1/Diversity_s$'dir. İspat ve daha ayrıntılı bilgi için (Yang ve Qiao, 2010)'den yararlanılabilir.

3.3.3. Sınıflandırma Doğruluğu

Sınıflandırma doğruluğu, doğru olarak sınıflandırılmış test seti gruplarının (tuples) yüzdesidir ve şöyle tanımlanır:

$$\text{Sınıflandırma doğruluğu} = \frac{TP + TN}{P + N} \tag{3.8}$$

P, pozitif grupların sayısıdır. *N*, negatif grupların sayısıdır. Doğru pozitifler (true positives-*TP*), doğru olarak etiketlenen pozitif gruplardır. Doğru negatifler (true negatives-*TN*), doğru olarak etiketlenen negatif gruplardır. Yanlış pozitifler (false positives-*FP*), yanlış olarak pozitif etiketlenen negatif gruplardır. Yanlış negatifler (false negatives-*FN*), yanlış olarak negatif etiketlenen pozitif gruplardır. *P'*, pozitif etiketlenen grupların sayısıdır ve *N'*, negatif etiketlenen grupların sayısıdır (Han ve diğ., 2012). Şekil 3.2, bu terimlerin özeti olan konfüzyon matrisini göstermektedir.

Tahmin edilen sınıf

Gerçek sınıf	**Pozitif**	**Negatif**	**Toplam**
Pozitif	*TP*	*FN*	*P*
Negatif	*FP*	*TN*	*N*
Toplam	*P'*	*N'*	*P + N*

Şekil 3.2: Konfüzyon matrisi.

3.3.4. F-Ölçütü

F-skoru ve F_1 skoru olarak da bilinen F-ölçütü, bir testin doğruluğu için bir ölçümdür ve sınıflandırma tekniklerini değerlendirmek için kullanılmaktadır. F-ölçütü şu şekilde tanımlanır:

$$\text{F-ölçütü} = \frac{2 \text{ x } precision \text{ x } recall}{precision + recall} \tag{3.9}$$

Burada *precision* ve *recall* sırasıyla kesinlik ve bütünlük ölçümleridir. Bu ölçümler şöyle hesaplanır (Han ve diğ., 2012):

$$Precision = \frac{TP}{TP + FP} \tag{3.10}$$

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{3.11}$$

3.3.5. Yürütme Süresi

Yürütme süresi (execution time, run time), bir programın çalışması ve sonlanması arasında geçen zamanı ifade etmektedir. Bu çalışmada, büyük veri üzerinde çalışılması sebebiyle tez kapsamında önerilen algoritmanın veya yazılımın yürütme süresi açısından etkin olması gerekmektedir.

3.3.6. Impala Sorguları

Cloudera Impala, HDFS veya HBase'de saklanan Hadoop verilerine direkt olarak hızlı ve etkileşimli SQL sorgularının çekilmesini sağlamaktadır. Impala, aynı birleşik depolama platformunu kullanmanın yanı sıra aynı meta verileri, SQL sözdizimini (Hive SQL), Açık Veri Tabanı Bağlantısı (Open Database Connectivity-ODBC) sürücüsünü ve kullanıcı arayüzünü (Hue'daki Cloudera Impala sorgu kullanıcı arayüzü) Apache Hive olarak kullanmaktadır. Bu, gerçek zamanlı veya yığın odaklı sorgular için alışıldık ve birleşik bir platform sağlamaktadır. Impala, büyük verileri sorgulamak için mevcut olan araçlara ek niteliğindedir ve MapReduce üzerinde oluşturulan yığın işleme çerçevelerini (frameworks) Hive gibi değiştirmez. MapReduce üzerinde oluşturulan Hive ve diğer çerçeveler, ETL türü işlerin yığın işlenmesini içerenler gibi uzun süren yığın işlerde en uygunudur. Impala'nın faydaları şu şekilde sıralanabilir (Cloudera Impala, 2018):

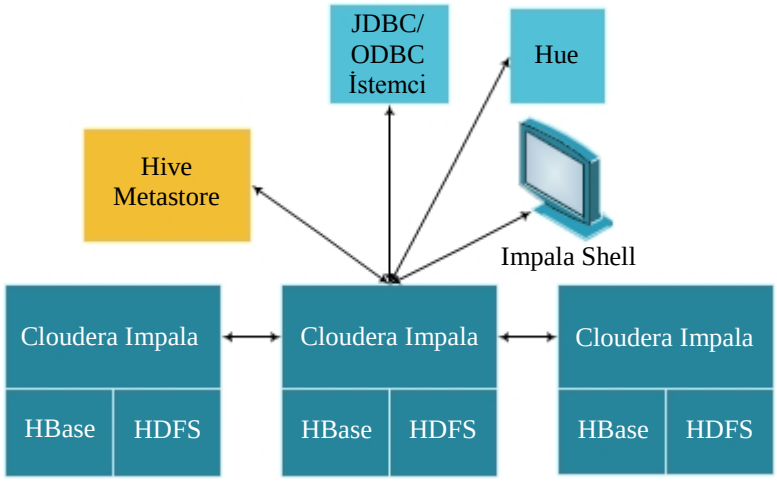
- Veri bilimcileri ve analistlerinin bildiği alışıldık SQL arayüzü.
- Apache Hadoop'taki yüksek hacimli verileri ("büyük veri") sorgulama becerisi.

- Uygun ölçekleme ve etkin maliyetli emtia donanımının kullanımı için küme ortamında dağıtık sorgular.

- Veri dosyalarını farklı bileşenler arasında kopyalama veya dışa aktarma/içe aktarma adımı olmadan paylaşma becerisi. Örneğin Pig ile yazmak, Hive ile dönüştürmek ve Impala ile sorgulamak. Impala, Hive tablolarından okuyabilir ve yazabilir. Bu, Hive'de üretilen veriler üzerinde analiz için Impala'yı kullanarak basit veri değişimini mümkün kılmaktadır.

- Büyük veri işleme ve analitiği için tek sistem. Dolayısıyla müşteriler sadece analitik için maliyetli modelleme ve ETL'den kaçınabilir.

Impala'nın daha geniş Cloudera ortamında nasıl konumlandığı Şekil 3.3'te (Cloudera Impala, 2018) gösterilmektedir.



Şekil 3.3: Impala'nın daha geniş Cloudera ortamında konumlanması.

Impala çözümü aşağıdaki bileşenlerden oluşmaktadır (Cloudera Impala, 2018):

- İstemciler: Hue, ODBC istemcileri, JDBC (Java Database Connectivity-Java Veri Tabanı Bağlantısı) istemcileri ve Impala Shell varlıklarının tümü Impala ile etkileşime girebilir. Bu arayüzler genellikle sorgu çekmek veya Impala'ya bağlanmak gibi yönetimsel görevleri tamamlamak için kullanılır.

- Hive Metastore: Impala'nın kullanabileceği veriler hakkında bilgi depolar. Örneğin metastore (metadepo), Impala'nın hangi veri tabanlarının mevcut olduğunu ve bu veri

tabanlarının yapısının ne olduğunu bilmesini sağlar. Şema nesneleri oluştururken, silerken veya değiştirirken, verileri tablolara yüklerken veya buna benzer Impala SQL ifadelerini yerine getirirken, ilgili meta veri değişiklikleri Impala 1.2'de öne sürülen özel katalog hizmeti tarafından tüm Impala düğümlerine otomatik olarak yayınlanır.

- Cloudera Impala: DataNodes (VeriDüğümleri) üzerinde çalışan bu süreç, sorguları koordine eder ve yürütür. Impala'nın her örneği; Impala istemcilerinin sorgularını alabilir, planlayabilir ve koordine edebilir. Sorgular, Impala düğümleri arasında dağıtılır ve bu düğümler daha sonra paralel sorgu parçaları yürüten çalışanlar olarak görev yapar.

- HBase ve HDFS: Sorgulanacak verilerin depolanması.

Impala kullanılarak yürütülen sorgular şu şekilde ele alınır (Cloudera Impala, 2018):

1. Kullanıcı uygulamaları, standart sorgulama arayüzleri sağlayan ODBC veya JDBC üzerinden Impala'ya SQL sorguları gönderir. Kullanıcı uygulaması, kümedeki herhangi bir impalad'a bağlanabilir. Bu impalad sorgunun koordinatörü olur.

2. Impala, sorguyu ayrıştırır (çözümler) ve küme boyunca impalad örnekleri tarafından hangi görevlerin gerçekleştirilmesi gerektiğini belirlemek için analiz eder. Yürütme en iyi verim için planlanır.

3. HDFS ve HBase gibi servislere, veri sağlamak için yerel impalad örnekler tarafından erişilir.

4. Her bir impalad, verileri koordine edici impalada döndürür ve bu koordine edici impalad sonuçları istemciye gönderir.

3.4. SINIFLANDIRMA YÖNTEMLERİ

Bu tez çalışmasında, önerilen algoritmanın sınıflandırma doğruluğu performansı Voted Perceptron, OneR, Naive Bayes ve C4.5 (J48) Karar Ağacı sınıflandırıcıları kullanılarak incelenmiştir.

3.4.1. Voted Perceptron Algoritması

Voted Perceptron algoritması, Freund ve Schapire (1999) tarafından büyük marjinlerle doğrusal olarak ayrılabilen verilerden yararlanan doğrusal sınıflandırma için öne sürülmüştür. Bu algoritma, Rosenblatt'ın iyi bilinen algılayıcı (perceptron) algoritmasına (Rosenblatt, 1958, 1962) ve Helmbold ve Warmuth (1995) tarafından geliştirilen online öğrenme algoritmalarının yığın öğrenme algoritmalarına dönüşümüne dayanmaktadır.

Voted Perceptron algoritmasının temelini oluşturan klasik algılayıcı algoritması, en doğal olarak online öğrenme modellerinde çalışılan çok basit bir algoritmadır. Algoritmada tüm noktaların $\mathbf{x} \in \mathbb{R}^n$ ve y etiketlerinin $\{-1, +1\}$ aralığında olduğu varsayılır. Online algılayıcı algoritması, bir başlangıç sıfır tahmini vektörü $\mathbf{v} = 0$ ile başlar. Yeni bir $\mathbf{x}$ örneğinin etiketini $\hat{y} = \text{sign}(\mathbf{v} \cdot \mathbf{x})$ olarak tahmin eder. Bu tahmin y etiketinden farklıysa, tahmin vektörünü $\mathbf{v} = \mathbf{v} + y\mathbf{x}$ olarak günceller. Tahmin doğruysa $\mathbf{v}$ değişmez. Aynı süreç sonraki örnekle tekrarlanır.

Algılayıcı algoritmasının bir eğitim örneği yığınından öğrenmek için kullanılmasının en yaygın yolu, eğitim setinin tümü üzerinde doğru olan bir tahmin vektörü bulana kadar algoritmayı eğitim seti boyunca sürekli olarak çalıştırmaktır. Bu tahmin kuralı daha sonra test setindeki etiketlerin tahmin edilmesi için kullanılır.

Voted Perceptron algoritmasında, eğitim sırasında daha fazla bilgi saklanmaktadır ve daha sonra test verileri üzerinde daha iyi tahminler üretmek için bu ayrıntılı bilgiler kullanılmaktadır. Eğitim sırasında sürdürülen bilgiler her bir hatadan sonra üretilen tüm tahmin vektörlerinin listesidir. Her bir vektör için bir sonraki hata yapılıncaya kadar "ayakta kaldığı" iterasyon sayısı sayılır. Bu sayım tahmin vektörünün "ağırlığı" olarak ifade edilir. Bir tahminin hesaplanması için tahmin vektörlerinin her birinin ikili (binary) tahmini hesaplanır ve tüm bu tahminler ağırlıklı bir çoğunluk oyuyla birleştirilir. Kullanılan ağırlıklar yukarıda açıklanan ayakta kalma süreleridir. Bu, sezgisel anlamda "iyi" tahmin vektörlerinin uzun bir süre ayakta kalma eğiliminde olduğu ve dolayısıyla çoğunluk oyunda daha fazla ağırlığa sahip olduğu anlamına gelmektedir (Freund ve Schapire, 1999).

3.4.2. OneR Sınıflandırıcı

OneR (one rule, tek kural) sınıflandırıcı, tümünün belirli bir niteliği test ettiği bir dizi kural biçiminde ifade edilen tek seviyeli bir karar ağacı oluşturan basit bir sınıflandırma algoritmasıdır. OneR, genellikle verideki yapıyı karakterize etmek için oldukça iyi kurallar içermektedir. OneR sınıflandırıcısının performansıyla ilgili yapılan kapsamlı çalışmalar, OneR'ın insanların yorumlaması için basit kurallar üretirken en yeni öğrenme şemalarından sadece biraz daha az doğru kurallar oluşturduğunu göstermektedir. OneR, basit bir algoritma olmasına rağmen eksik değerler ve sayısal niteliklerle başa çıkabilmektedir. Bu da algoritmanın uyarlanabilirliğini göstermektedir (Waguih, 2013).

OneR algoritması, eğitim verisindeki her nitelik için bir kural oluşturur. Ardından kuralı, "tek kural" olarak en küçük hata oranıyla seçer. Bir nitelik için kural oluştururken her nitelik değeri için en sık sınıf belirlenmek zorundadır. En sık sınıf, basitçe bu nitelik değeri için en sık görünen sınıftır. Bir kural, çoğunluk sınıflarına bağlı bir dizi nitelik değeridir (Witten and Frank, 2005).

OneR algoritması şu şekilde çalışmaktadır (Muda ve diğ., 2011):

Adım 1: Kümelenmiş setten her nitelik tahminleyicisinin (predictor) her bir değeri için bir kural seti oluşturulur.

1. Hedef sınıfın her bir değerinin ne sıklıkla ortaya çıktığı sayılır.
2. En sık görülen sınıf bulunur.
3. Bir kural seti oluşturulur ve bu sınıf nitelik tahminleyicisinin bu değerine atanır.
4. Her nitelik tahminleyicisi için ayarlanan kurallarda oluşan toplam hata hesaplanır.

Adım 2: En küçük toplam hataya sahip olan en iyi nitelik tahminleyicileri seçilir.

3.4.3. Naive Bayes Sınıflandırıcı

Bayes sınıflandırıcıları istatistiksel sınıflandırıcılardır ve verilen bir grubun (tuple) belirli bir sınıfa ait olma olasılığı gibi sınıf üyelik olasılıklarını tahmin edebilirler. Bayes sınıflandırma Bayes teoremine dayanmaktadır. Sınıflandırma algoritmalarını karşılaştıran çalışmalar, karar ağacı ve seçilen sinir ağı sınıflandırıcılarıyla performans açısından karşılaştırılabilir olması için

Naive Bayes sınıflandırıcı olarak bilinen basit bir Bayes sınıflandırıcısı bulmuştur. Bayes sınıflandırıcılar, büyük veritabanlarına uygulandığında yüksek doğruluk ve hız sergilemektedir. Naive Bayes sınıflandırıcıları, verilen bir sınıftaki bir nitelik değerinin etkisinin diğer niteliklerin değerlerinden bağımsız olduğunu varsaymaktadır. Bu varsayım, "sınıf-koşullu bağımsızlık" olarak adlandırılır. Sınıf-koşullu bağımsızlık, söz konusu hesaplamaları basitleştirmek için yapılır ve bu bağlamda "naive" olarak kabul edilir (Han ve diğ., 2012).

Bayes teoremi, 18. yüzyılda olasılık ve karar teorisinde çalışmalar yapan, yenilikçi bir İngiliz papaz olan Thomas Bayes'in adını almıştır. $\boldsymbol{X}$ bir veri grubu olsun. $\boldsymbol{X}$, Bayes terimlerinde "kanıt (evidence)" olarak kabul edilir. Doğal olarak bir dizi n niteliği üzerinde yapılan ölçümlerle tanımlanır. H, $\boldsymbol{X}$ veri grubunun belirtilen bir C sınıfına ait olması gibi bir hipotez olsun. H hipotezinin "kanıt" ya da gözlenen $\boldsymbol{X}$ veri grubunu vermesi olasılığı olan $P(H|\boldsymbol{X})$, sınıflandırma problemleri için belirlenmek istenmektedir. Diğer bir deyişle $\boldsymbol{X}$'in nitelik tanımı bilindiğinde, $\boldsymbol{X}$ grubunun C sınıfına ait olma olasılığı aranmaktadır (Han ve diğ., 2012).

$P(H|\boldsymbol{X})$, $\boldsymbol{X}$ üzerinde koşullu H'nin sonsal (posterior ya da posteriori) olasılığıdır. Tersine, $P(H)$, H'nin önsel (prior veya priori) olasılığıdır. $P(H|\boldsymbol{X})$ sonsal olasılığı, $\boldsymbol{X}$'den bağımsız olan $P(H)$ önsel olasılığından daha fazla bilgiye dayanmaktadır. Benzer şekilde $P(\boldsymbol{X}|H)$, H üzerinde koşullu $\boldsymbol{X}$'in sonsal olasılığıdır. $P(\boldsymbol{X})$, $\boldsymbol{X}$'in önsel olasılığıdır. Bayes teoremi; $P(H)$, $P(\boldsymbol{X}|H)$ ve $P(\boldsymbol{X})$'den $P(H|\boldsymbol{X})$ sonsal olasılığının hesaplanması için bir yol sağlaması açısından yararlıdır. Bayes teoremi:

$$P(H|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|H)P(H)}{P(\boldsymbol{X})} \tag{3.12}$$

Bayes teoremine dayanan Naive Bayes sınıflandırıcı aşağıdaki gibi çalışmaktadır (Han ve diğ., 2012):

1. D, grupların bir eğitim seti ve ilişkili sınıf etiketleri olsun. Her zaman olduğu gibi, her bir grup bir n-boyutlu nitelik vektörü $\boldsymbol{X} = (x_1, x_2, \ldots, x_n)$ ile temsil edilir ve sırasıyla $A_1, A_2, \ldots, A_n$ olarak n nitelikli grup üzerinde yapılan n ölçümlerini gösterir.

2. $C_1, C_2, \ldots, C_m$ olarak m sınıf olduğu varsayılsın. Bir $\boldsymbol{X}$ grubu verildiğinde, sınıflandırıcı, $\boldsymbol{X}$'in $\boldsymbol{X}$ koşullu en yüksek sonsal olasılığa sahip olan sınıfa ait olduğunu öngörecektir. Yani Naive

Bayes sınıflandırıcı $\boldsymbol{X}$ grubunun C_i sınıfına ait olduğunu ancak ve ancak $1 \leq j \leq m, j \neq i$ için $P(C_i|\boldsymbol{X}) > P(C_j|\boldsymbol{X})$ olduğunda tahmin eder. Böylece $P(C_i|\boldsymbol{X})$ maksimuma çıkarılır. $P(C_i|\boldsymbol{X})$'nin maksimuma çıkarıldığı C_i sınıfı "maksimum sonsal hipotezi" olarak adlandırılır. Bayes teoreminden (3.12) denklem (3.13) elde edilir:

$$P(C_i|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|C_i)P(C_i)}{P(\boldsymbol{X})} \tag{3.13}$$

3. $P(\boldsymbol{X})$ tüm sınıflar için sabitken, sadece $P(\boldsymbol{X}|C_i)P(C_i)$ maksimuma çıkarılmaya ihtiyaç duyar. Sınıf önsel olasılıkları bilinmese de yaygın olarak sınıfların eşit derecede olası olduğu varsayılır. Yani $P(C_1) = P(C_2) = \cdots = P(C_m)$ ve bu nedenle $P(\boldsymbol{X}|C_i)$ maksimuma çıkarılır. Aksi durumda $P(\boldsymbol{X}|C_i)P(C_i)$ maksimuma çıkarılır. Ayrıca sınıf önsel olasılıklarının $P(C_i) = |C_{i,D}|/|D|$ ile tahmin edilebileceği dikkate alınmalıdır. Burada $|C_{i,D}|$, D'deki C_i sınıfının eğitim gruplarının sayısıdır.

4. Çok nitelikli veri setleri verildiğinde, $P(\boldsymbol{X}|C_i)$'yi hesaplamak hesaplama açısından son derece masraflı olacaktır. $P(\boldsymbol{X}|C_i)$'yi değerlendirirken hesaplamayı azaltmak için sınıf-koşullu bağımsızlığın naive varsayımı yapılır. Bu, grubun sınıf etiketi verildiğinde niteliklerin değerlerinin koşullu olarak birbirinden bağımsız olduğunu varsayar (nitelikler arasında hiçbir bağımlılık ilişkisi yoktur). Böylece:

$$P(\boldsymbol{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \tag{3.14}$$

Eğitim gruplarından $P(x_1|C_i)$, $P(x_2|C_i)$,…, $P(x_n|C_i)$ olasılıkları kolay bir şekilde tahmin edilebilir. Burada x_k, $\boldsymbol{X}$ grubunun A_k niteliğinin değerini ifade etmektedir.

5. $\boldsymbol{X}$'in sınıf etiketini tahmin etmek için her bir C_i sınıfı için $P(\boldsymbol{X}|C_i)P(C_i)$ hesaplanır. Sınıflandırıcı $\boldsymbol{X}$ grubunun sınıf etiketinin C_i sınıfı olduğunu ancak ve ancak $1 \leq j \leq m, j \neq i$ için $P(\boldsymbol{X}|C_i)P(C_i) > P(\boldsymbol{X}|C_j)P(C_j)$ olduğunda tahmin eder. Diğer bir deyişle tahmin edilen sınıf etiketi $P(\boldsymbol{X}|C_i)P(C_i)$'nin maksimum olduğu C_i sınıfıdır.

Bayes sınıflandırıcılarının karar ağacı ve sinir ağı sınıflandırıcılarına kıyasla ne kadar etkili olduğunu görebilmek için yapılan çeşitli deneysel çalışmalar, sınıflandırıcının bazı alanlarda karşılaştırılabilir olduğunu göstermektedir. Teorik olarak, Bayes sınıflandırıcıları diğer tüm sınıflandırıcılara kıyasla minimum hata oranına sahiptir. Ancak pratikte bu, sınıf-koşullu bağımsızlık ve mevcut olasılık verilerinin eksikliği gibi kullanımı için yapılan varsayımlardaki yanlışlıklardan dolayı her zaman geçerli değildir.

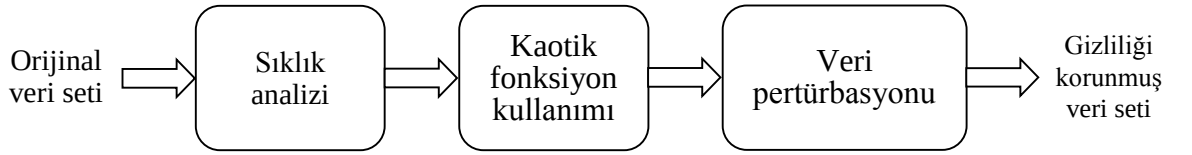
Bayes sınıflandırıcıları, Bayes teoremini açık bir şekilde kullanmayan diğer sınıflandırıcılar için teorik bir doğrulama sağladığı için de yararlıdır. Örneğin birçok sinir ağı ve eğri-uydurma (curve-fitting) algoritmasının belli varsayımlar altında maksimum sonsal hipotezini Naive Bayes sınıflandırıcısının yaptığı gibi ortaya çıkardığı gösterilebilir (Han ve diğ., 2012).

3.4.4. C4.5 (J48) Karar Ağacı Algoritması

J48, ID3 algoritmasının (Quinlan, 1990) ardılı olan C4.5 karar ağacı algoritmasının (Quinlan, 1993) açık kaynaklı Weka uygulamasıdır. C4.5 algoritması, verilen bir veri seti için verilerin özyinelemeli (recursive) bölümlenmesi ile bir sınıflandırma karar ağacı oluşturur. Karar, derin öncelikli (depth-first) stratejisi kullanılarak büyür. Algoritma, veri setini bölebilen tüm olası testleri göz önünde bulundurur ve en iyi bilgi kazancını (information gain) veren bir testi seçer. Niteliğin farklı değerlerinin sayısı kadar çıktılı bir test her bir ayrık nitelik için dikkate alınır. Niteliğin her farklı değerini içeren ikili testler her bir sürekli nitelik için hesaba katılır. Bütün bu ikili testlerin entropi kazancını verimli bir şekilde bir araya getirmek için söz konusu düğümün ait olduğu eğitim veri seti sürekli nitelik değerleri için sıralanır ve her bir farklı değere dayanan ikili kesimin entropi kazancı sıralanan verilerin bir taramasında hesaplanır. Bu süreç her sürekli nitelik için tekrarlanır (Zhao ve Zhang, 2008). Bu algoritma ile ilgili daha ayrıntılı bilgi için (Mitchell, 1997; Quinlan, 1986)'den yararlanılabilir.

3.5. ÖNERİLEN ETKİN GİZLİLİK KORUMA ALGORİTMASI

Bu tez çalışmasında, gizlilik ve kullanılabilirlik koruması kaos ve veri pertürbasyonu kullanılarak sağlanmıştır. Önerilen algoritmanın genel blok diyagramı Şekil 3.4'te gösterilen üç ana aşamadan oluşmaktadır. İlk aşama, her bir yarı tanımlayıcı için benzersiz nitelik değerlerinin sıklığının analiz edilmesi ve daha sonra bu sıklık analizine göre kritik değerlerin bulunmasıdır. İkinci aşamada, seçilen kritik değerler için yeni değerler belirlemede kaotik bir fonksiyon kullanılır. Son aşamada ise veri pertürbasyonu gerçekleştirilir.



Şekil 3.4: Önerilen algoritmanın genel blok diyagramı.

Tez kapsamında önerilen etkin gizlilik koruma algoritması Algoritma 1'de sunulmaktadır. Bu algoritma aşağıda yer alan sekiz adımdan oluşmaktadır:

Adım 1: Orijinal giriş veri seti D, yarı tanımlayıcı nitelikler QI (QI_1, QI_2,..., QI_q) ve hassas nitelik SA belirlenir.

Adım 2: Her bir QI için benzersiz nitelik değerleri bulunur. $|D|$, giriş veri seti D'nin boyutudur ve $|QI|$, yarı tanımlayıcı niteliklerin QI sayısıdır.

Adım 3: Benzersiz nitelik değerlerini içeren kayıtların sayısı her bir QI için hesaplanır.

Adım 4: Benzersiz nitelik değerleri, sıklığa göre küçükten büyüğe sıralanır.

Adım 5: Benzersiz nitelik değerlerinin D'deki kayıt yerleri, sonraki rastgeleleştirme ve yenisiyle değiştirme işlemleri için bulunur.

Adım 6: Kritik benzersiz nitelik değerlerinin sayısı her bir QI için denklem (3.15) kullanılarak hesaplanır.

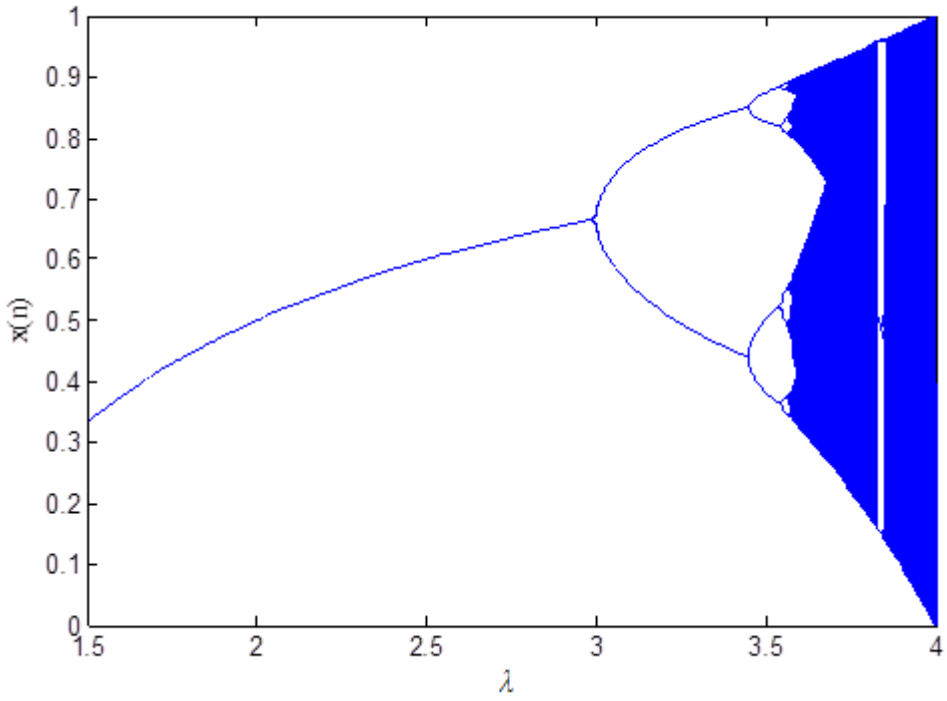
$$r = \text{round}\,(\log_2 \textit{benzersiz nitelik değerlerinin sayısı}) \tag{3.15}$$

Belirli bir QI için benzersiz nitelik değerlerinin sayısı ne kadar az olursa kimlik ifşası ve bağlantı saldırıları için o kadar kritik olur. Bu nitelikler bireylerin hassas niteliğini açığa çıkarmak için kötü niyetli veya izinsiz kişiler tarafından kullanılabilir.

Adım 7: Seçilen kritik benzersiz değerler için yeni nitelik değerleri, lojistik harita (logistic map) olarak bilinen kaotik bir fonksiyon kullanılarak belirlenir (3.16).

$$f(x) = \lambda x(1 - x) \tag{3.16}$$

Bu fonksiyonda λ, $3.57 < \lambda < 4$ aralığındadır. Fonksiyonun kaotik davranışı tamamen λ değerine bağlıdır. λ, fonksiyonun en kaotik bölgede işleyebilmesi için 3.99 ve 4 aralığında tanımlanmalıdır (Yavuz ve diğ., 2016). Lojistik haritanın dallanma diyagramı Şekil 3.5'te gösterilmektedir. Şekilde görüldüğü üzere fonksiyonun çıkışı λ değeri 4'e yaklaştığında 0 ve 1 aralığında farklı değerler almaktadır. Bu çalışmada lojistik haritanın kullanılma amacı, veri pertürbasyonu için lojistik haritanın bilinen kaotik davranışından yararlanmaktır.



Şekil 3.5: Lojistik haritanın dallanma diyagramı.

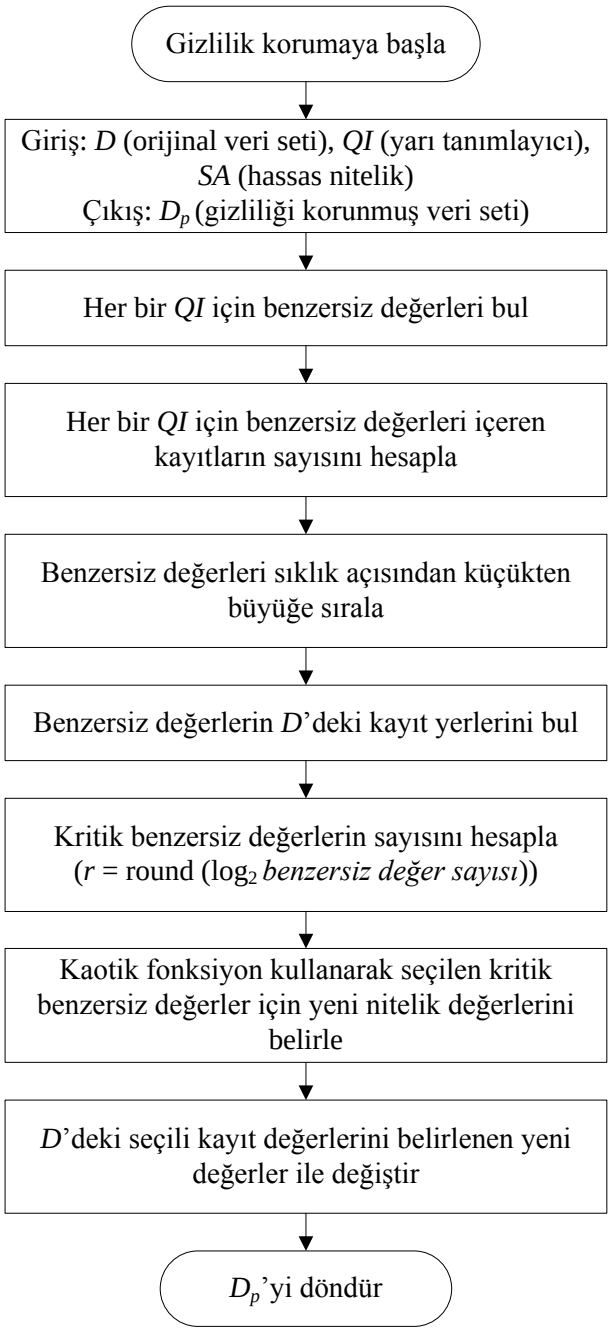
Adım 8: D'deki seçili kayıt değerleri, belirlenen yeni nitelikleri değerleri ile değiştirilir. En sonunda, gizliliği korunmuş veri seti D_p elde edilir.

Gizlilik koruma sürecinin akış diyagramı, algoritmayı daha açık bir şekilde anlatmak için Şekil 3.6'da gösterilmiştir.

Algoritma 1: Önerilen Etkin Gizlilik Koruma Algoritması

Giriş: Orijinal giriş veri seti D, yarı tanımlayıcı nitelikler QI $(QI_1, QI_2, \ldots, QI_q)$ ve hassas nitelik SA
Çıkış: Gizliliği korunmuş veri seti D_p
Başlangıç atamaları: $c = 0$, $\lambda = 3.99$, $iterasyon = 400$

1: $d = |D|$
2: $q = |QI|$
3: **for** $i = 1$ to q **do**
4: nu_i = her bir QI_i için benzersiz değer sayısı
5: **for** $j = 1$ to nu_i **do**
6: u_{ij} = her bir QI_i için benzersiz değerler
7: v_{ij} = benzersiz değer u_{ij}'yi içeren kayıt sayısı
8: **end for**
9: **end for**
10: Her bir QI_i için v_{ij}'ye bağlı olarak u_{ij}'yi küçükten büyüğe sırala
11: $kayıt\_yeri_i = Ø$ (her bir QI_i için boyut $d \times nu_i$)
12: **for** $i = 1$ to q **do**
13: **for** $j = 1$ to nu_i **do**
14: **for** $k = 1$ to d **do**
15: **if** QI_i'deki k. kayıt değeri == sıralı u_{ij}'deki j. değer **then**
16: c++
17: $kayıt\_yeri_i(c, j) = j$
18: **else**
19: continue
20: **end if**
21: **end for**
22: $c = 0$
23: **end for**
24: **end for**
25: **for** $i = 1$ to q **do**
26: $r_i = \text{round}(\log_2 nu_i)$
27: **end for**
28: **for** $i = 1$ to q **do**
29: $x_{i1} = 0.1$
30: **for** $j = 1$ to $iterasyon$ **do**
31: $x_{ij+1} = \lambda \times x_{ij} \times (1 - x_{ij})$
32: **end for**
33: **end for**
34: Her bir QI_i için kayıt yerleri x_{ij}'ye bağlı olarak sıralı benzersiz değerler u_{ij}'de ilk r_i değer için yeni nitelik değerlerini belirle
35: D'de, seçilen kayıt değerlerini belirlenen yeni değerler ile değiştir
36: D_p'yi döndür

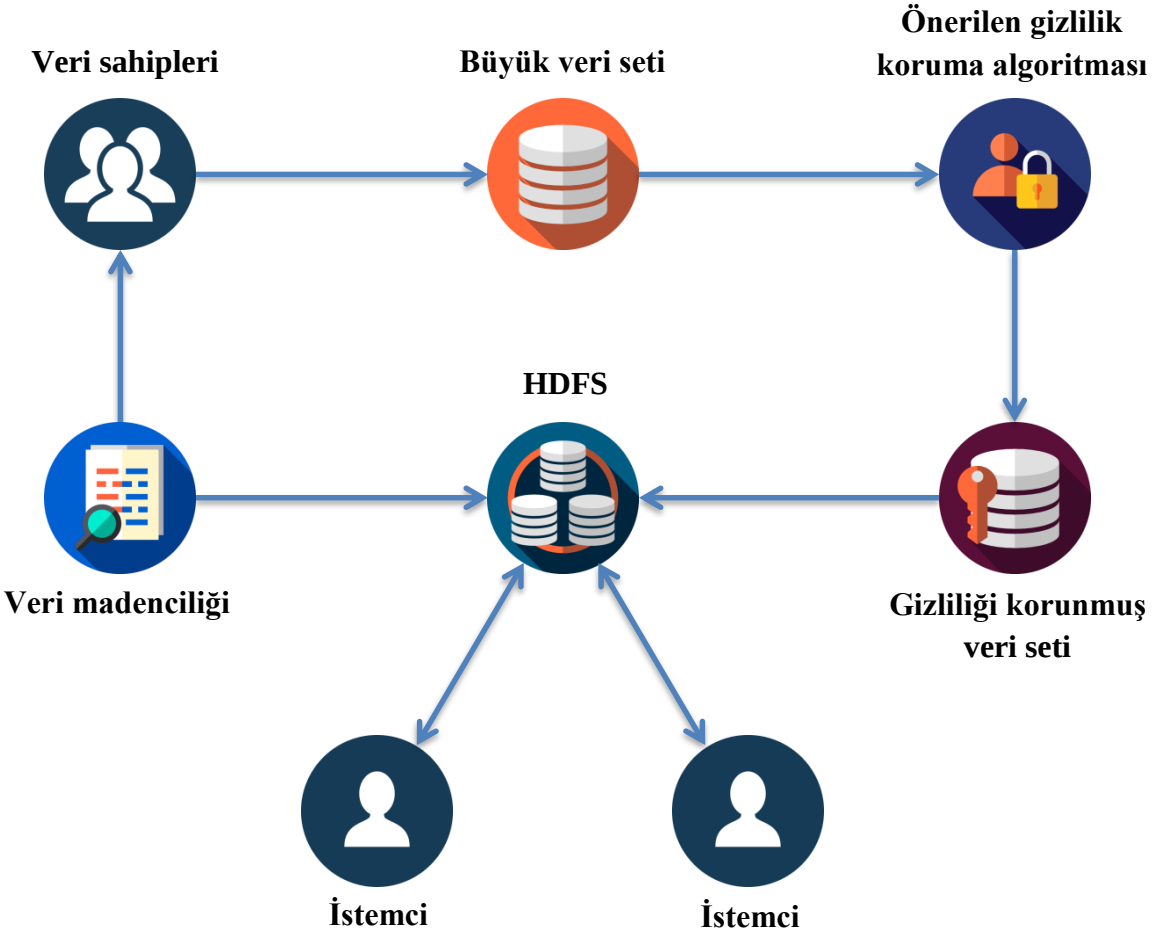


Şekil 3.6: Önerilen algoritmanın işlevsel akış diyagramı.

3.6. GİZLİLİĞİ KORUNMUŞ VERİ SETLERİNİN DAĞITIKLAŞTIRILMASI

Bu tez çalışmasında önerilen etkin gizlilik koruma algoritmasının diğer bir amacı, veri setlerini farklı düğümler arasında güvenli bir şekilde dağıtmaktır. Hadoop, çok büyük veri setlerini işlemek için kullanılan dağıtık bir çerçeve sağlamaktadır (Song ve diğ., 2015). Hadoop; HDFS

(Shafer ve diğ., 2010; Shvachko ve diğ., 2010) ve Mapreduce (Gu ve diğ., 2014) çerçevesinden oluşmaktadır. Veriler parçalara (chunks) ayrılır ve Hadoop kümesinde (cluster) bulunan düğümler arasında dağıtılır. Veri işleme paralel bir şekilde yapılır ve bu nedenle sistemin ölçeklenebilirliği çok yüksektir (Nayahi ve Kavitha, 2017). Tez kapsamında önerilen algoritma kullanılarak anonimleştirilen veriler güvenli bir şekilde HDFS üzerinde dağıtıklaştırılmaktadır. Tez kapsamında kullanılan Hadoop üzerinde gizlilik korumalı veri dağıtıklaştırma mimarisi Şekil 3.7'de gösterilmektedir.



Şekil 3.7: Hadoop üzerinde gizlilik korumalı veri dağıtıklaştırma mimarisi.

Veri sahipleri, önerilen gizlilik koruma algoritmasını giriş veri seti üzerinde bağımsız olarak uygularlar ve gizliliği korunmuş veri setini Hadoop üzerinde dağıtırlar. İstemciler, gizliliği korunmuş veriler için okuma isteğinde bulunabilir. Ardından herhangi bir veri madenciliği algoritması, Mapreduce hizmeti paralel olarak kullanılarak bu dağıtık veriler üzerinde uygulanabilir ve çıkarılan bilgi veya örüntüler veri sahipleri ile paylaşılır. Hadoop üzerinde

gizliliği korunmuş verileri görebilen herhangi bir saldırgan, bir kaydı belirli bir bireyle eşleştiremez. Böylece önerilen etkin gizlilik koruma algoritması, verileri Hadoop kümesi üzerinde gizlilik koruma amacıyla veri sahipleri arasında ek hesaplama karmaşıklığı ve iletişim yükü olmadan saklamak için etkili bir mekanizma olarak hizmet verebilir. Bir veri için istekte bulunan istemci bir mesajı $O(1)$ ile değiş tokuş etmektedir ve yanıt, istenen gizliliği korunmuş veri için $O(n)$ olacaktır. Burada n veri seti boyutudur.

Tez kapsamında önerilen algoritma ile gizliliği korunmuş veri setlerinin Hadoop üzerinde dağıtımı için CDH (Cloudera's distribution including Apache Hadoop-Cloudera'nın Apache Hadoop'u içeren dağıtımı) kullanılmıştır. CDH, Cloudera'nın Hadoop ekosistemini içeren %100 açık kaynaklı platform dağıtımıdır ve dünyanın en popüler Hadoop platformudur. CDH, tamamen açık standartlarda inşa edilmiştir. Ayrıca sınırsız veri depolama, işleme, keşfetme, modelleme ve sunma için tüm ana bileşenlere sahiptir (Cloudera, 2018). CDH sürümü olarak ücretsiz olan Cloudera QuickStart VM 5.12 kullanılmıştır. Cloudera QuickStart VM, VMware Workstation Pro 12 üzerinde sanal makine olarak kurulmuştur. Gizliliği korunmuş veri setlerinin CDH üzerinde dağıtımı adım adım şu şekilde gerçekleştirilir:

1. Cloudera Quickstart VM 5.12'nin VMware platformu için olan sürümü "zip" dosyası olarak indirilir ve bu dosya dışarıya çıkarılır.

2. Çıkarılan zip dosyası içerisinde "vmx" uzantılı dosya açılarak sanal makinenin otomatik olarak VMware Workstation Pro arayüzünde görülmesi sağlanır.

3. VMware Workstation Pro üzerindeki sanal makinenin CPU ve RAM gibi özellikleri ayarlanır (Şekil 3.8).

4. Sanal makine çalıştırılarak ayağa kaldırılır.

5. Sanal makine çalıştıktan sonra Cloudera QuickStart VM'nize hoş geldiniz sayfası açılır (Şekil 3.9). Bu sayfada küme (cluster) bilgileri yer almaktadır:

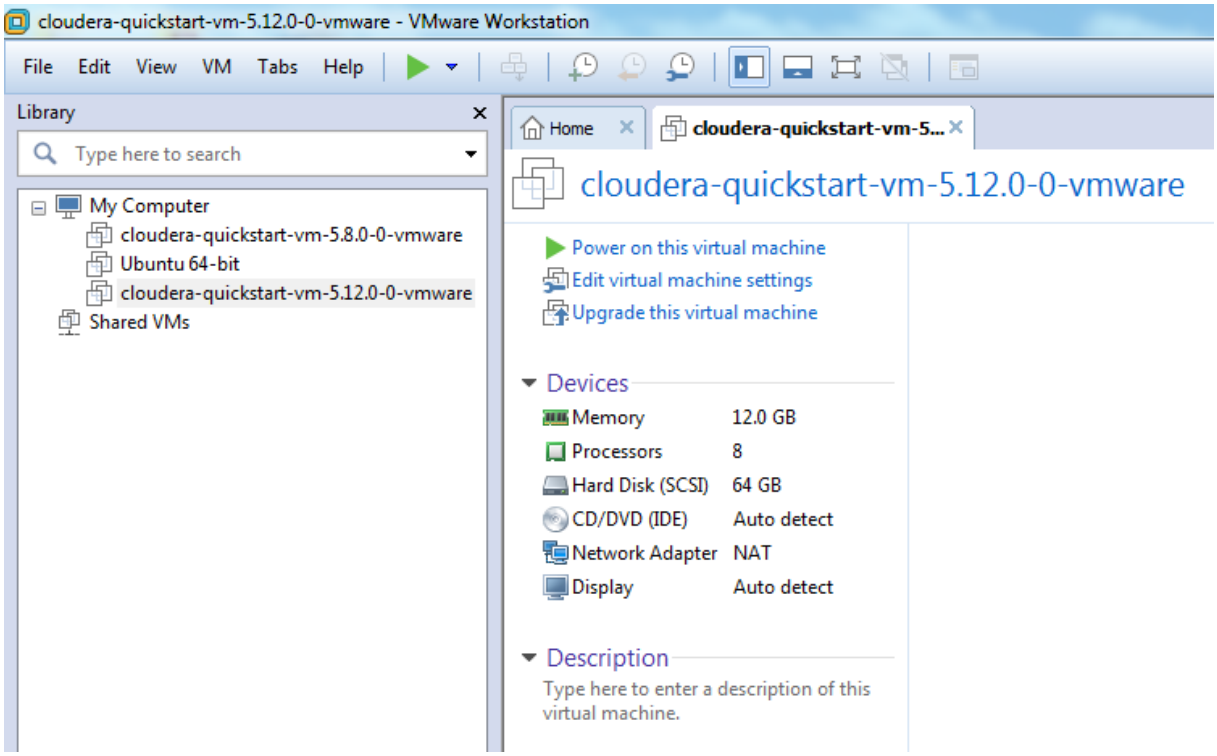
    - Yönetici düğüm adresi (manager node address): 192.168.135.130

    - İşçi düğüm 1 adresi (worker node 1 address): 192.168.135.130

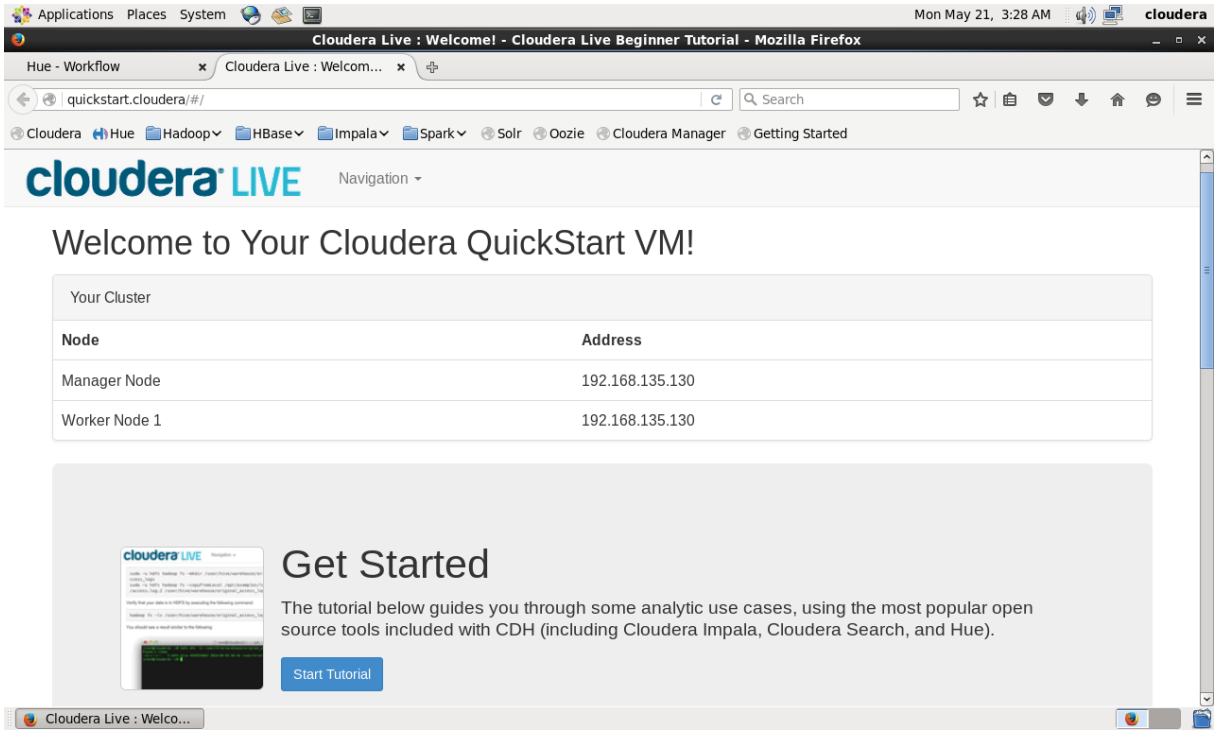
6. Ardından Cloudera QuickStart VM arayüzünden eğitime başlanır (Şekil 3.10).

7. Cloudera QuickStart VM sanal makinesini çalıştırmak için Cloudera Express (ücretsiz) ya da Cloudera Enterprise (ücretsiz 60 günlük deneme sürümü) gerekmektedir.
    - Cloudera Express en az 8GB RAM ve en az 2 sanal CPU gerektirmektedir.
    - Cloudera Enterprise en az 10GB RAM ve en az 2 sanal CPU gerektirmektedir.
8. Cloudera Express başlatılır. Cloudera Express'i bir kez başlatmak yeterlidir ve sanal makinenin kapatılıp tekrar açılması durumunda da çalışmaktadır.
9. Ortam (environment) doğrulanır ve Cloudera Manager açılır (Şekil 3.11).
10. Servislerin (HDFS, Hive, Hue, Impala ve YARN) çalışıp çalışmadığı kontrol edilir. Cloudera Manager ana sayfa durum görünümünde, her bir servisin durum göstergesi yeşil olmalıdır (Şekil 3.12).
11. Hue giriş sayfası arayüzünden kullanıcı adı ve şifre "cloudera" olarak girilir (Şekil 3.13). Sonrasında Hue ana sayfası (Impala) açılır (Şekil 3.14)
12. Hue arayüzüne girildikten sonra büyük veri seti Hadoop ortamına yüklenir: Browsers → Files → Upload → "gizliliği_korunmuş_büyük_veri_seti.txt" (Şekil 3.15 ve 3.16).
13. Hadoop ortamına yüklenecek gizliliği korunmuş büyük veri seti seçilir (Şekil 3.17 ve 3.18).
14. Hadoop ortamına yüklenen veri setinden (Şekil 3.19) tablo oluşturulur: Documents → Tables → Create a new table (Şekil 3.20).
15. Oluşturulacak tablo için Hadoop ortamına yüklenen veri setinin yolu (path) verilir (Şekil 3.21 ve 3.22).
16. Oluşturulacak tablo için alan (field) ayırıcı ve kayıt (record) ayırıcı ayarlanır (Şekil 3.23 ve 3.24).
17. Oluşturulacak tablo için alan türleri belirlenir (Şekil 3.25 ve 3.26).
18. Hadoop ortamı üzerinde oluşturulan tablonun bilgilerini içeren tablo açılır (Şekil 3.27 ve 3.28).

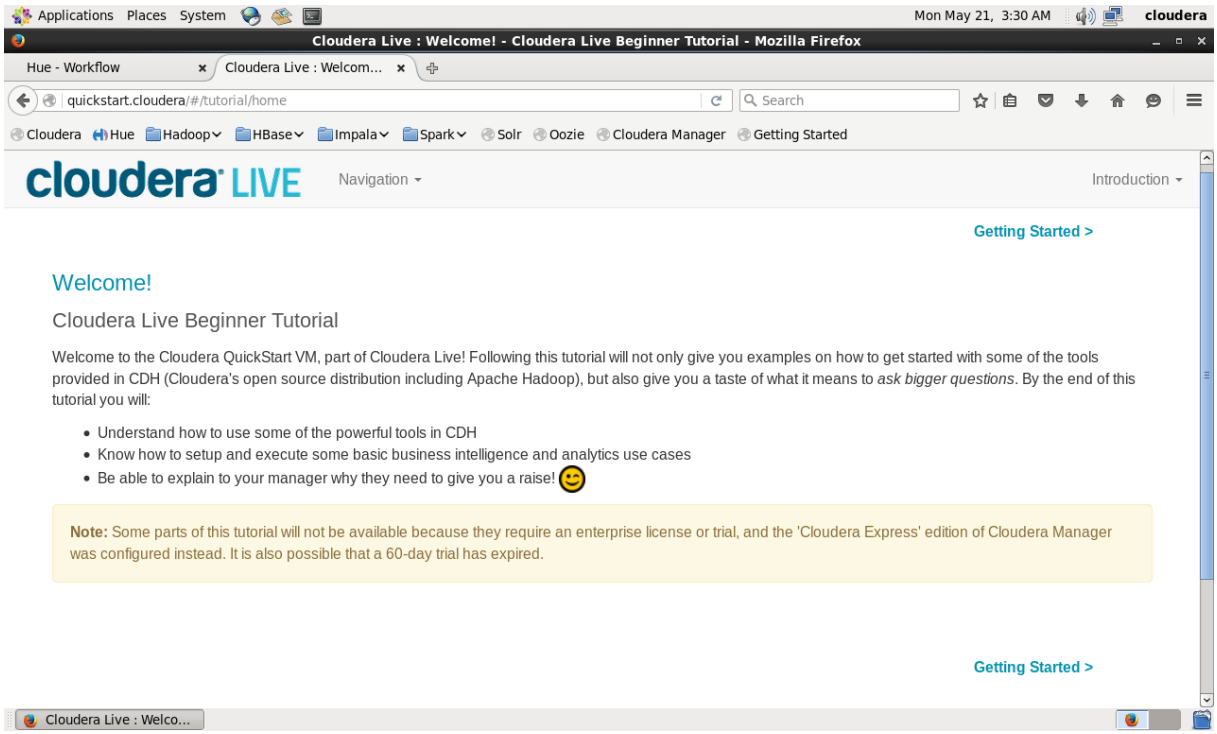
19. Son olarak Hadoop ortamındaki gizliliği korunmuş büyük veri setleri üzerinde Impala ile sorgu çekilir (Şekil 3.29).



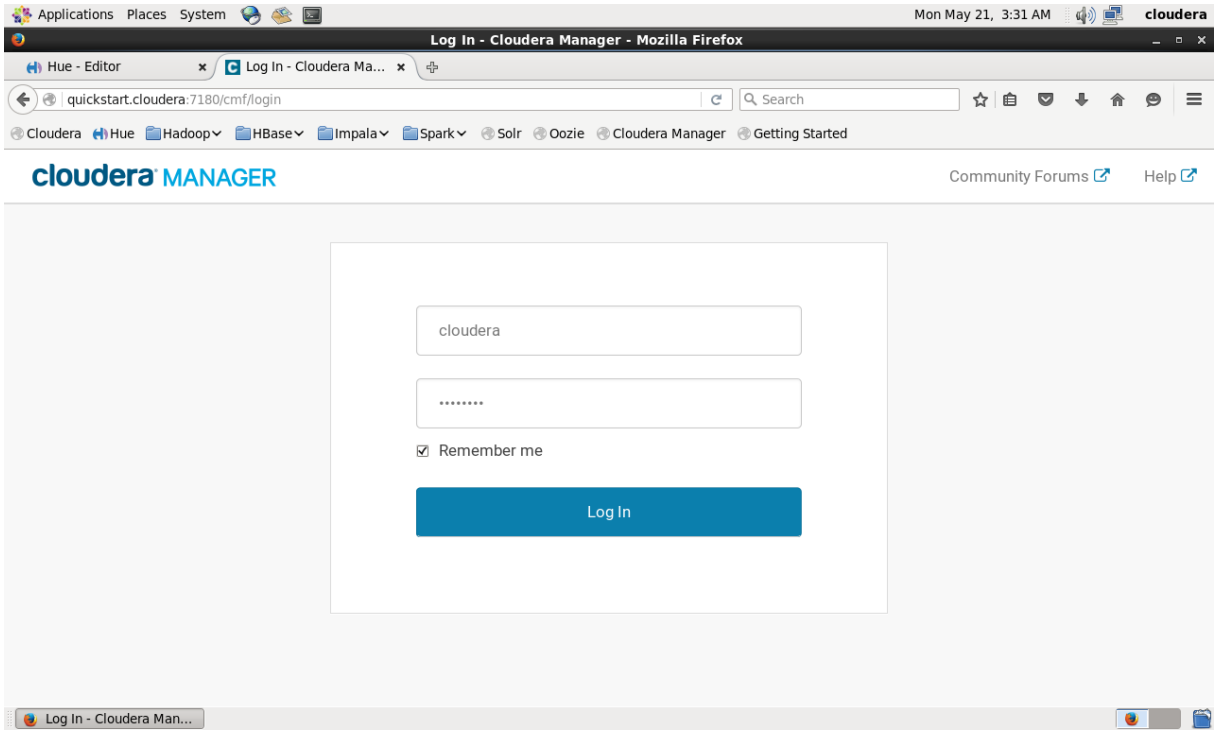
Şekil 3.8: Sanal makinenin özellikleri.



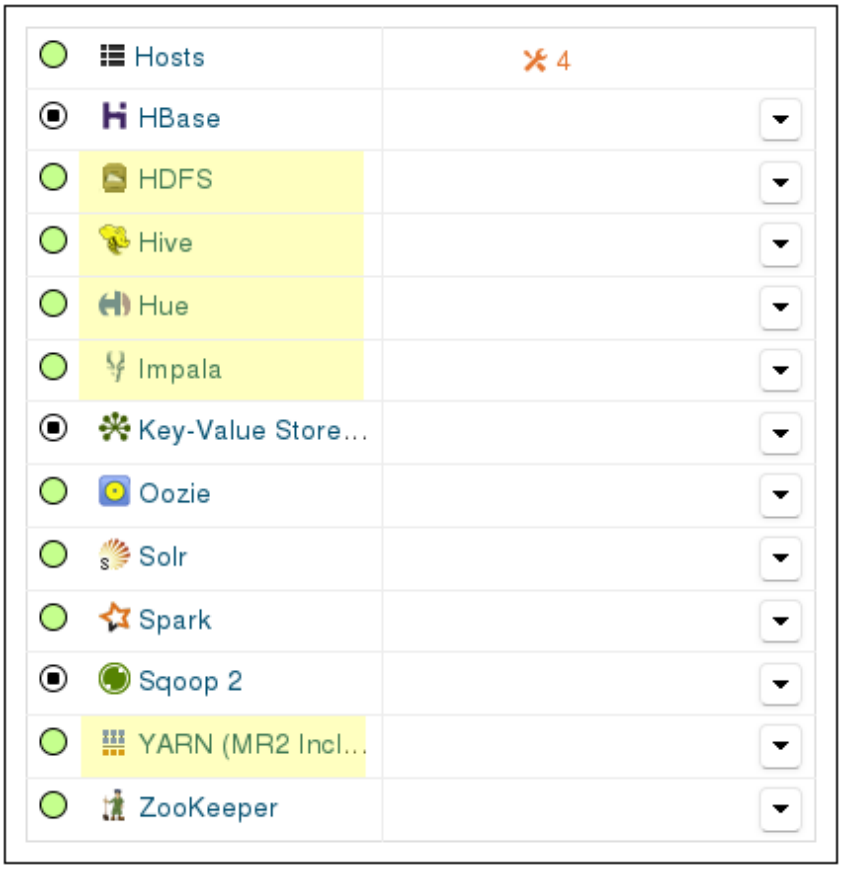
Şekil 3.9: Cloudera QuickStart VM'nize hoş geldiniz sayfası.



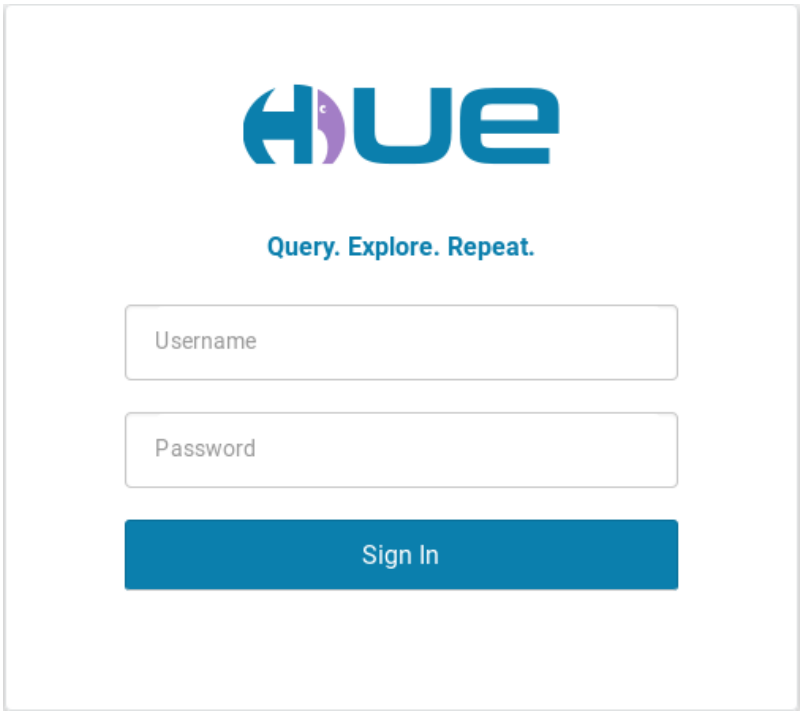
Şekil 3.10: Cloudera QuickStart VM eğitime başlama arayüzü.



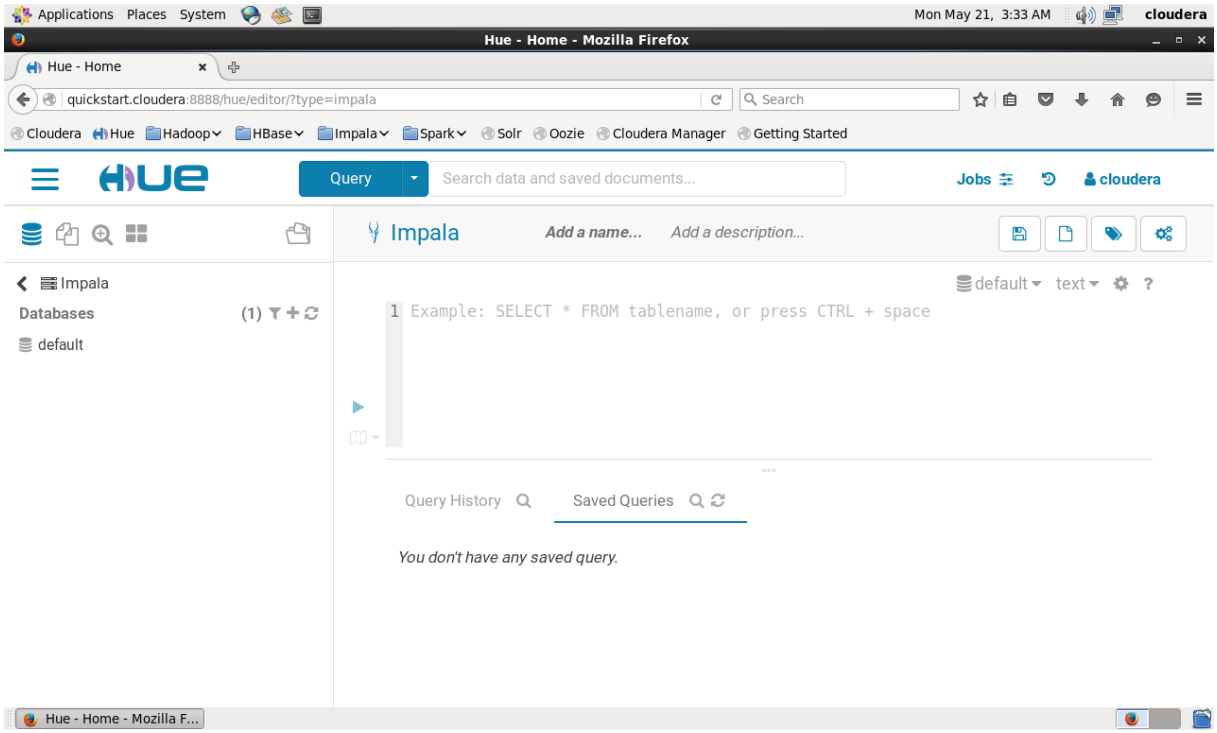
Şekil 3.11: Cloudera Manager giriş sayfası arayüzü.



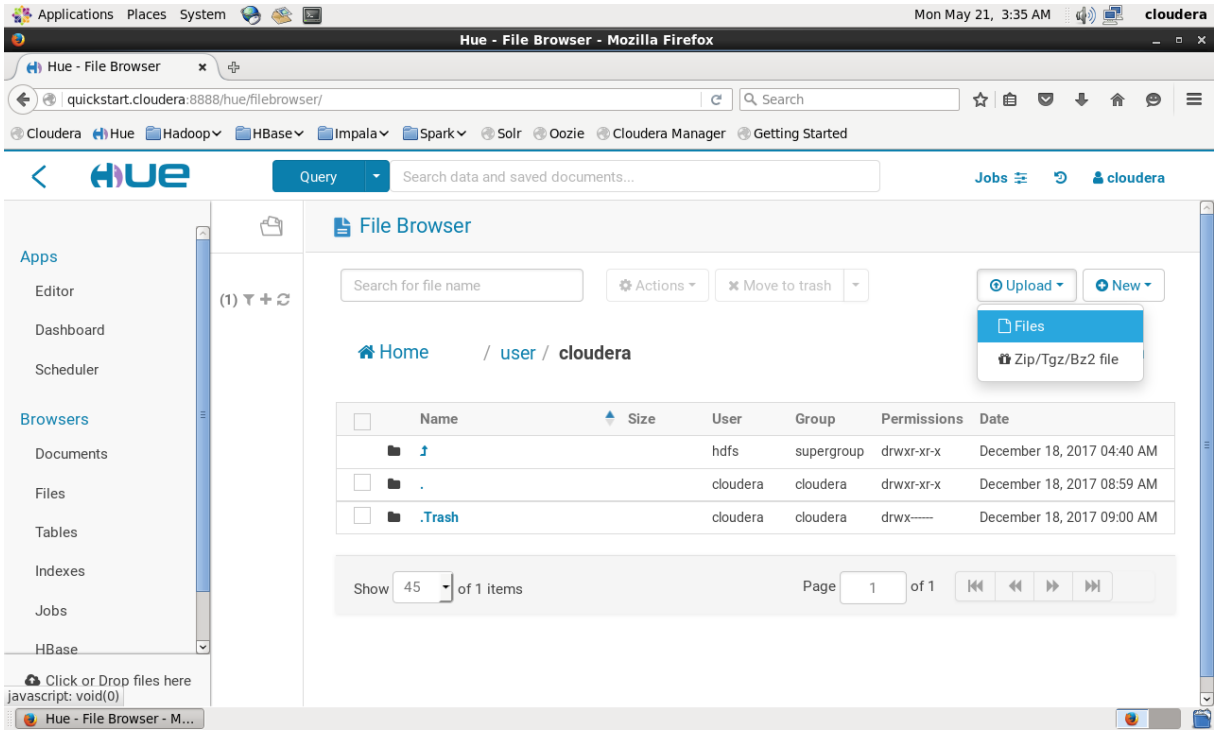
Şekil 3.12: Hadoop servisleri.



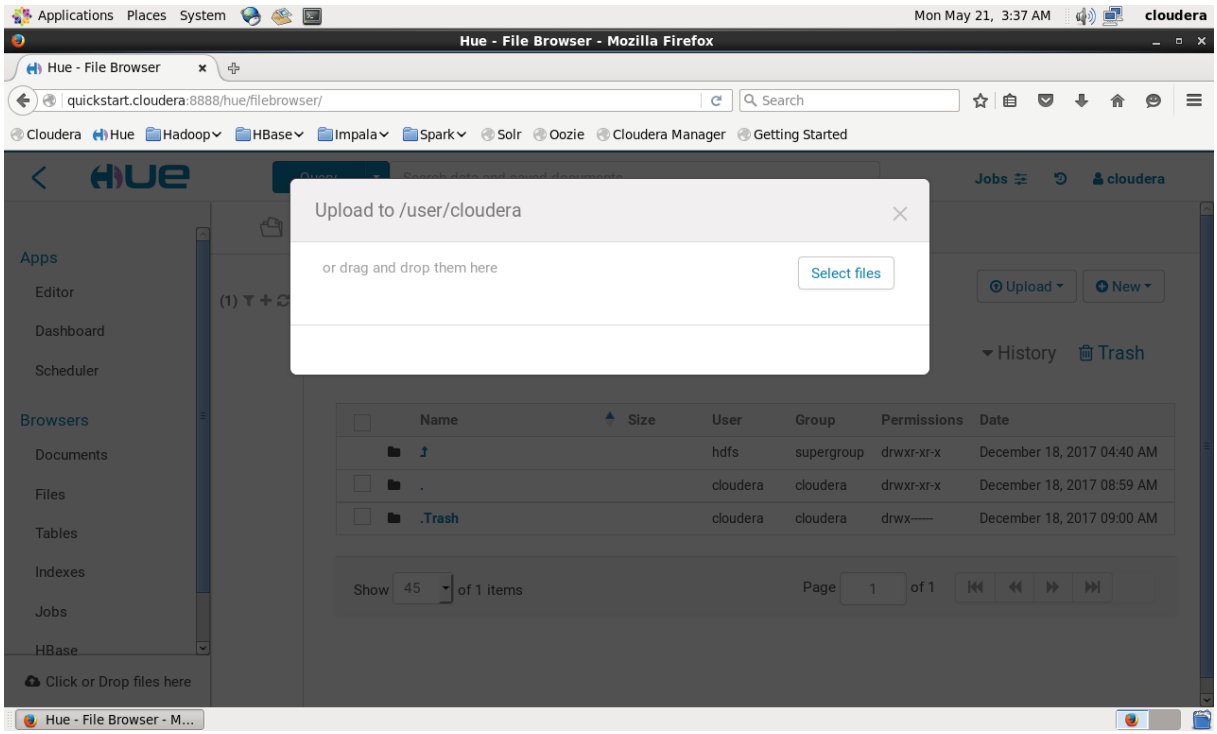
Şekil 3.13: Hue giriş sayfası arayüzü.



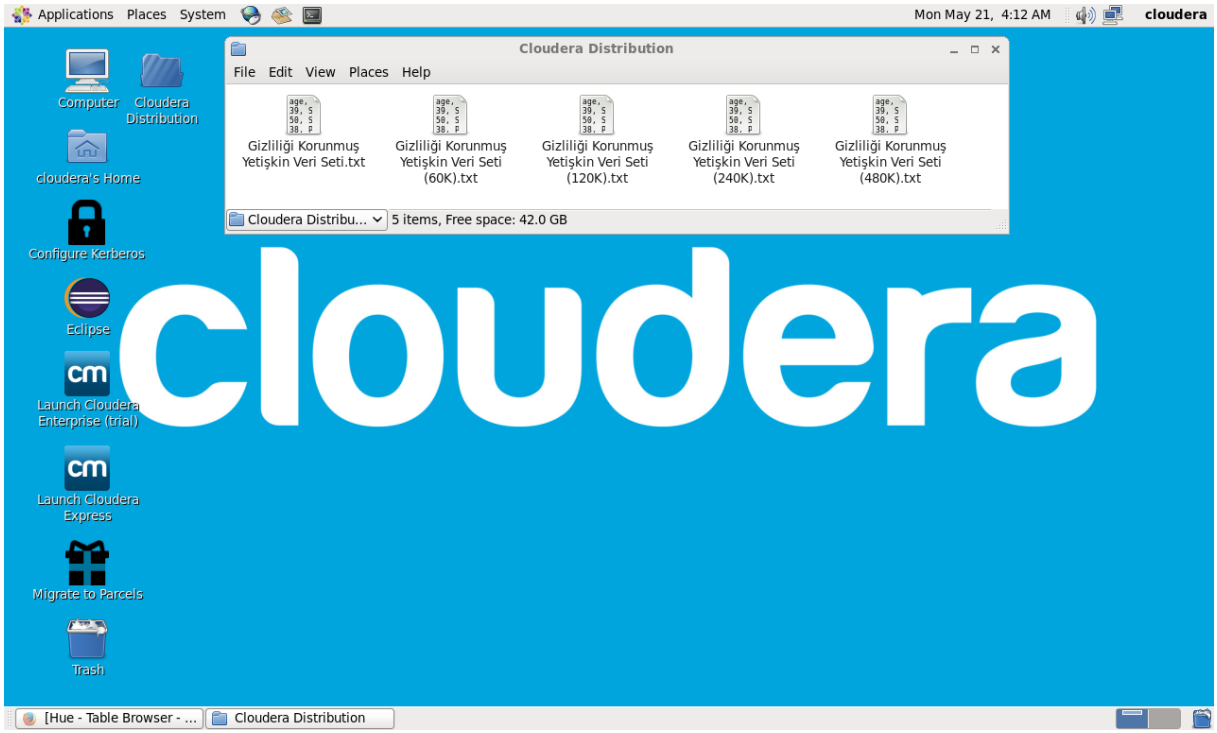
Şekil 3.14: Hue ana sayfa (Impala).



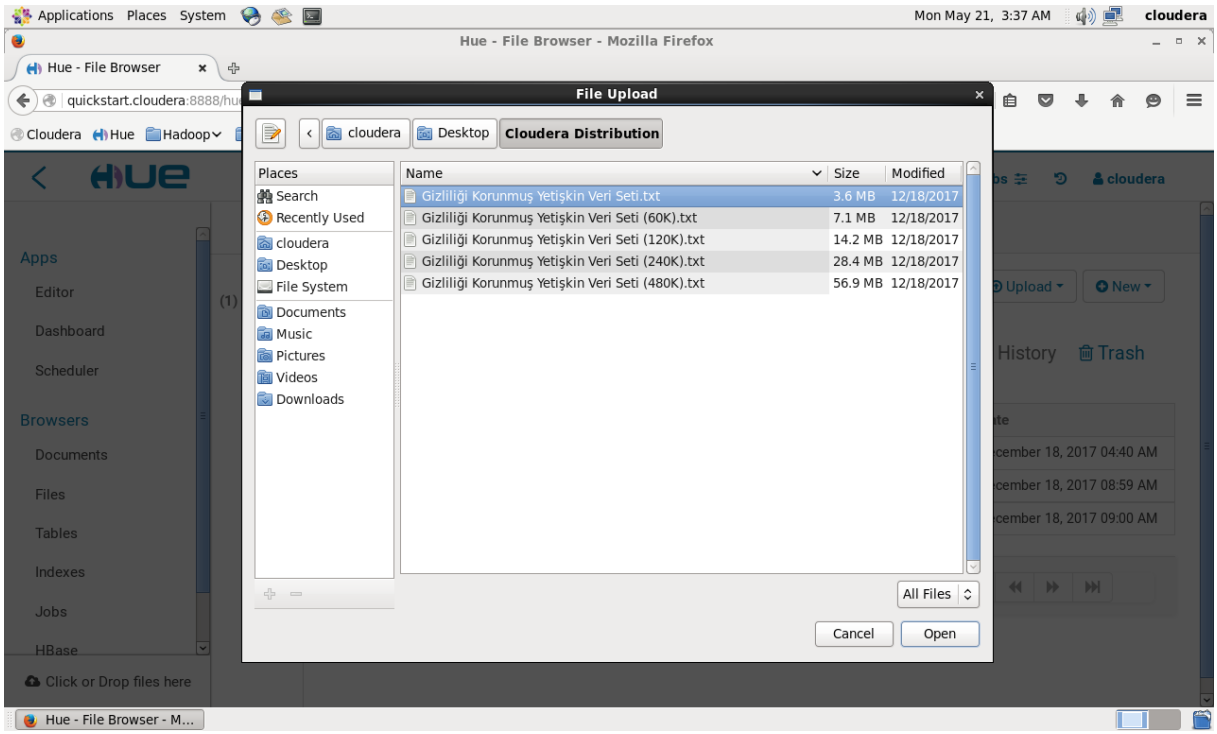
Şekil 3.15: Büyük veri setinin Hadoop ortamına yüklenmesi (1).



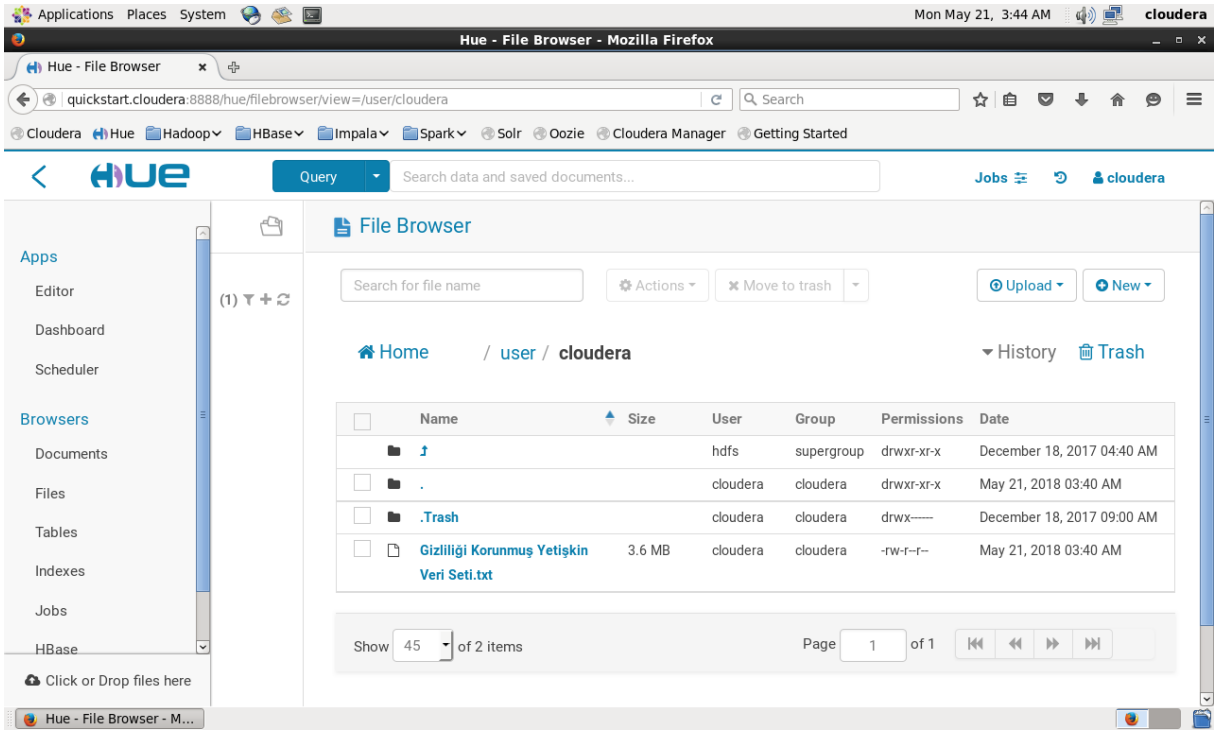
Şekil 3.16: Büyük veri setinin Hadoop ortamına yüklenmesi (2).



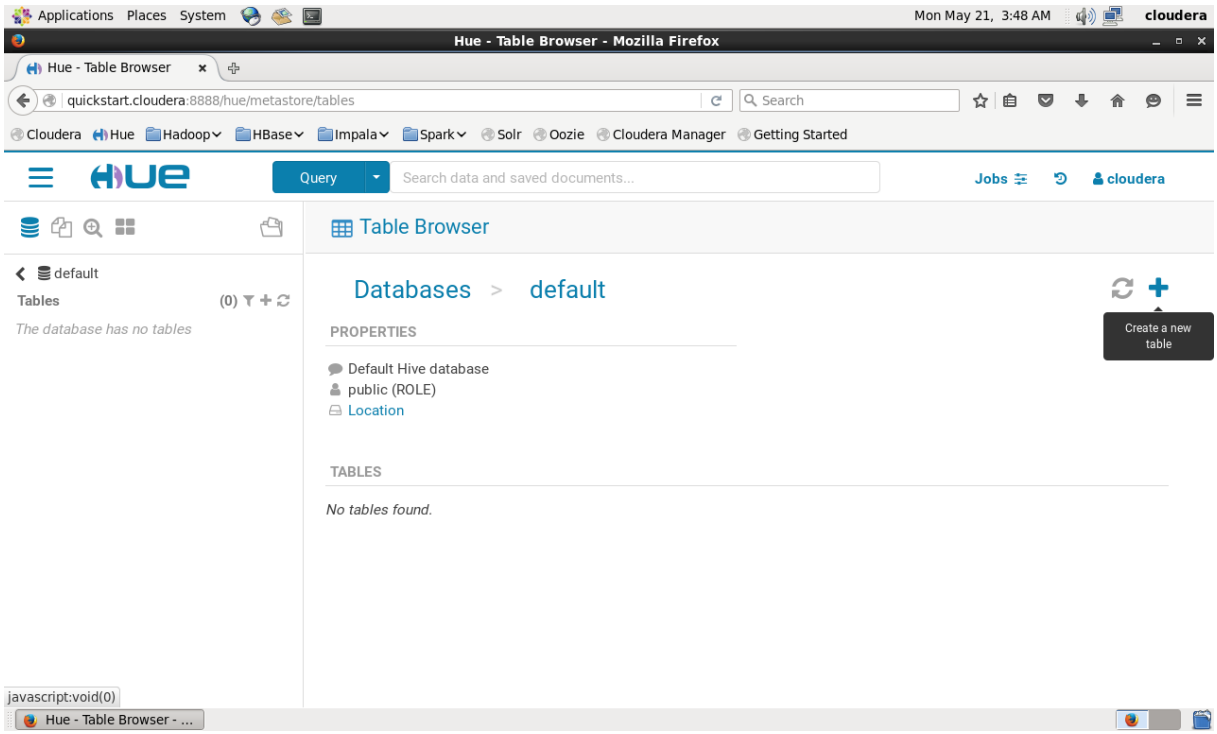
Şekil 3.17: Gizliliği korunmuş büyük veri setleri.



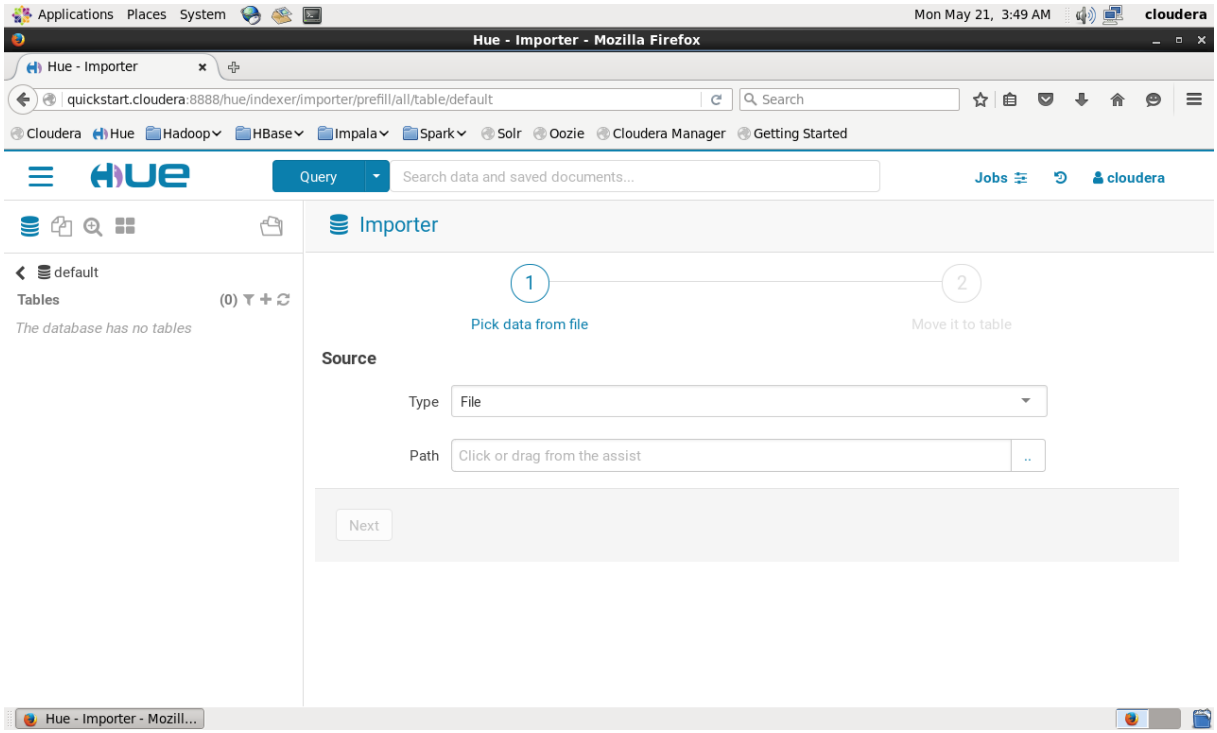
Şekil 3.18: Hadoop ortamına yüklenecek büyük veri setinin seçilmesi.



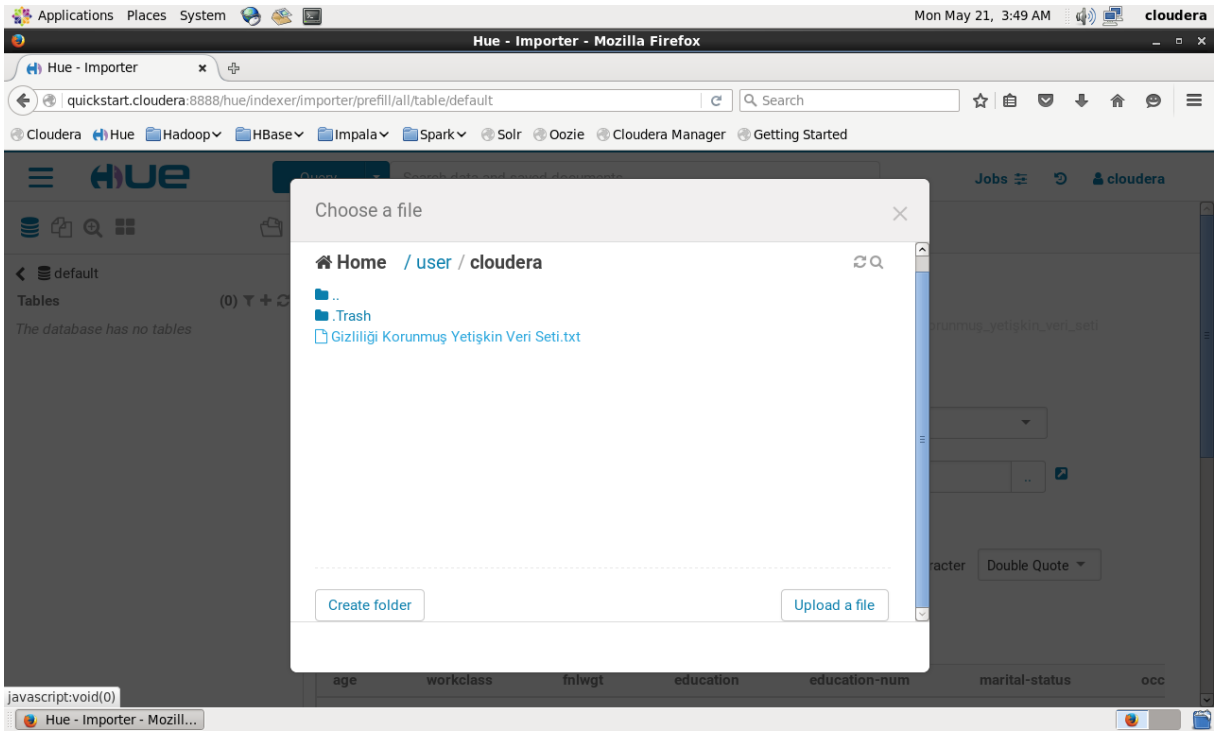
Şekil 3.19: Hadoop ortamındaki gizliliği korunmuş büyük veri seti.



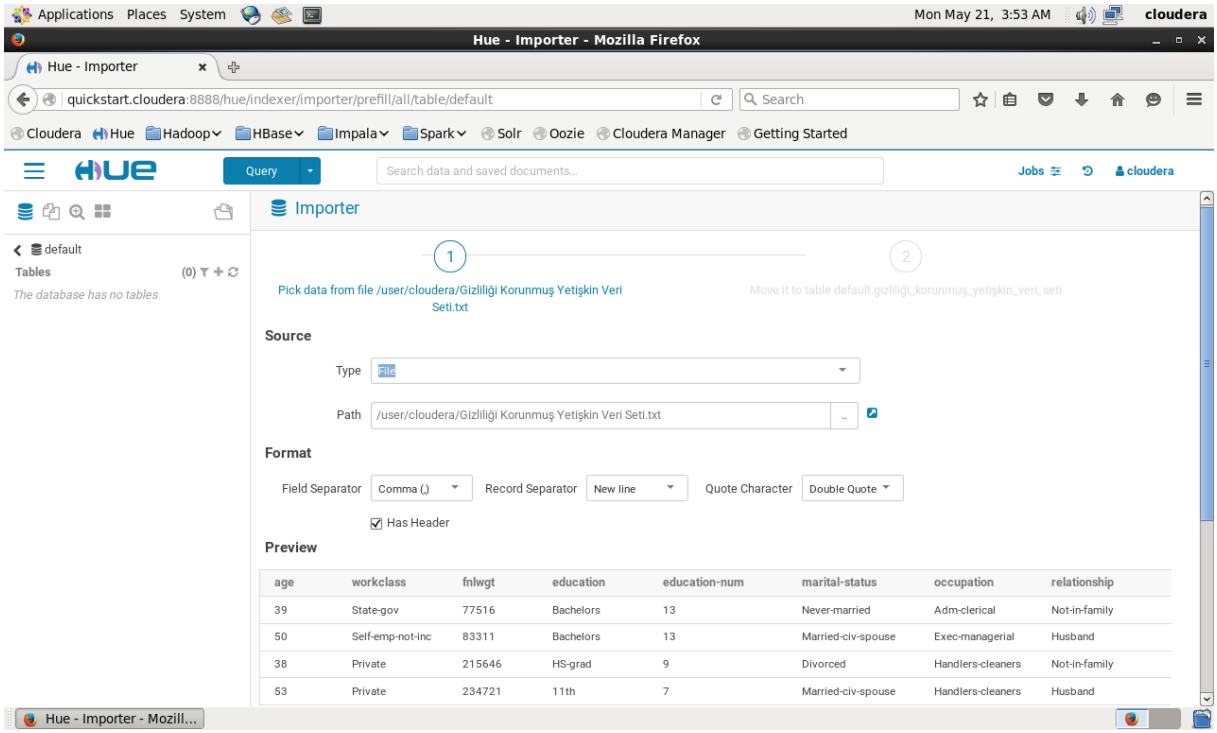
Şekil 3.20: Hadoop ortamına yüklenen veri setinden tablonun oluşturulması.



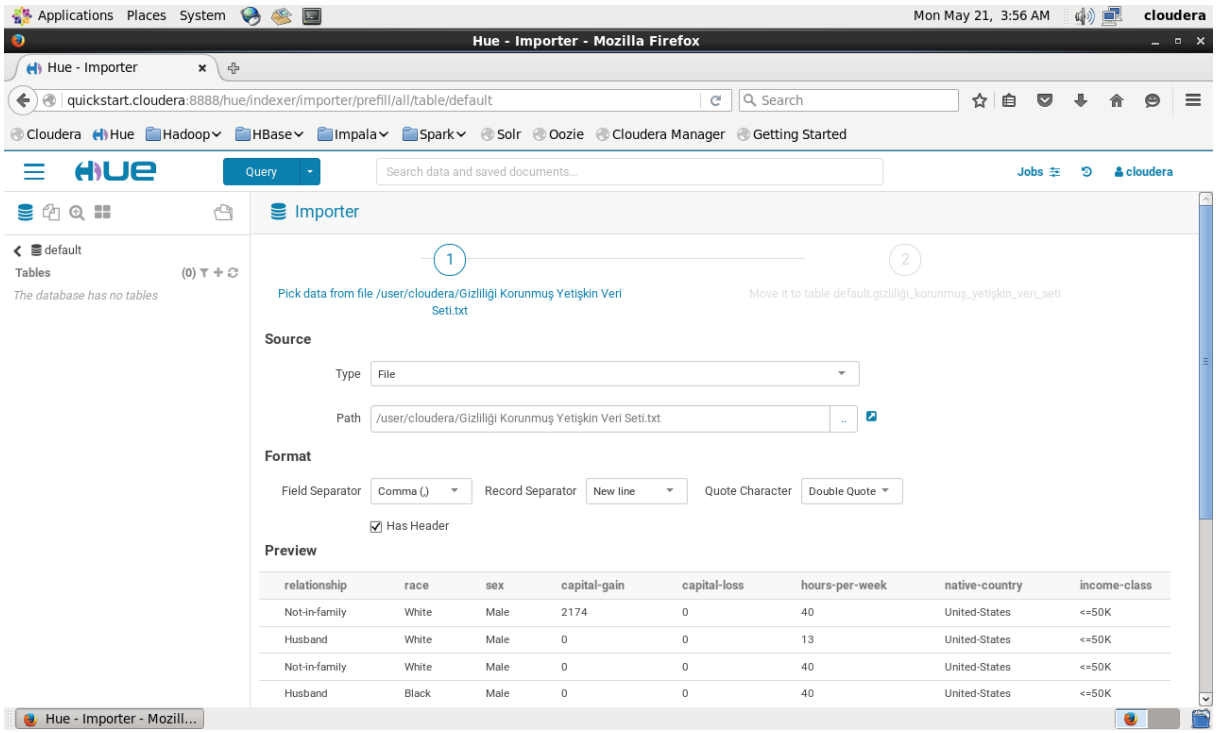
Şekil 3.21: Oluşturulacak tablo için Hadoop ortamına yüklenen veri setinin yolunun verilmesi (1).



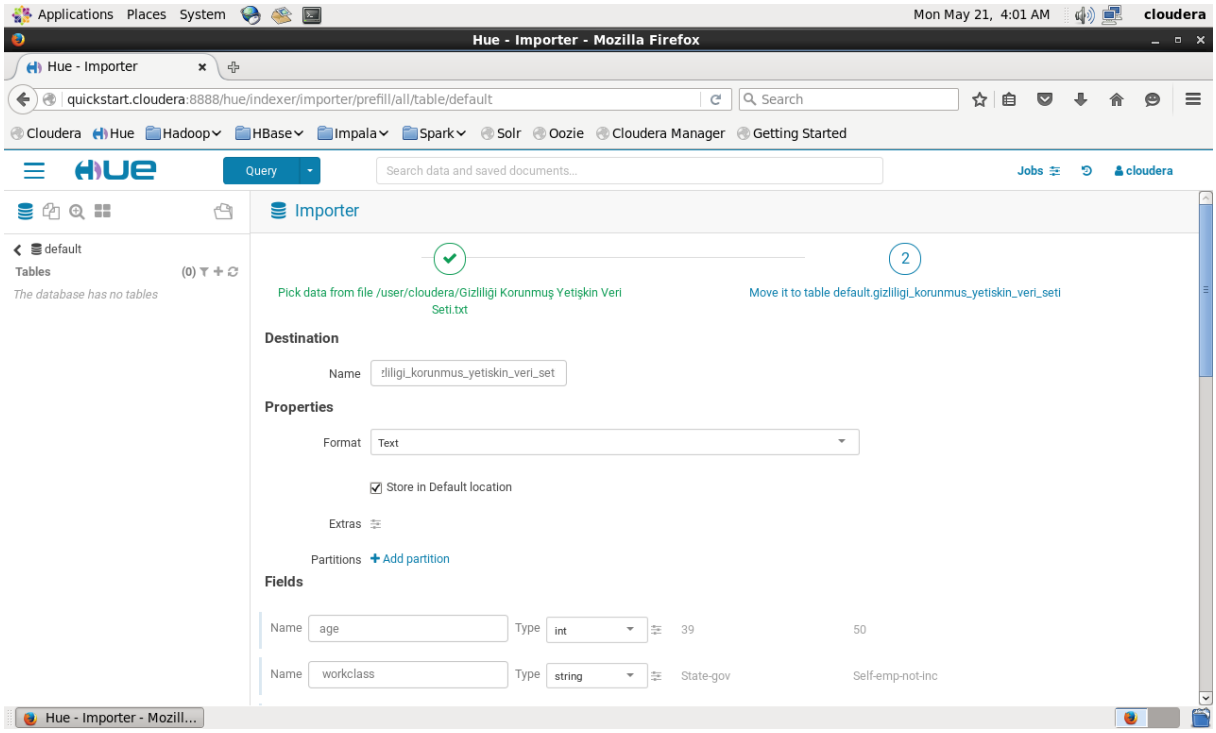
Şekil 3.22: Oluşturulacak tablo için Hadoop ortamına yüklenen veri setinin yolunun verilmesi (2).



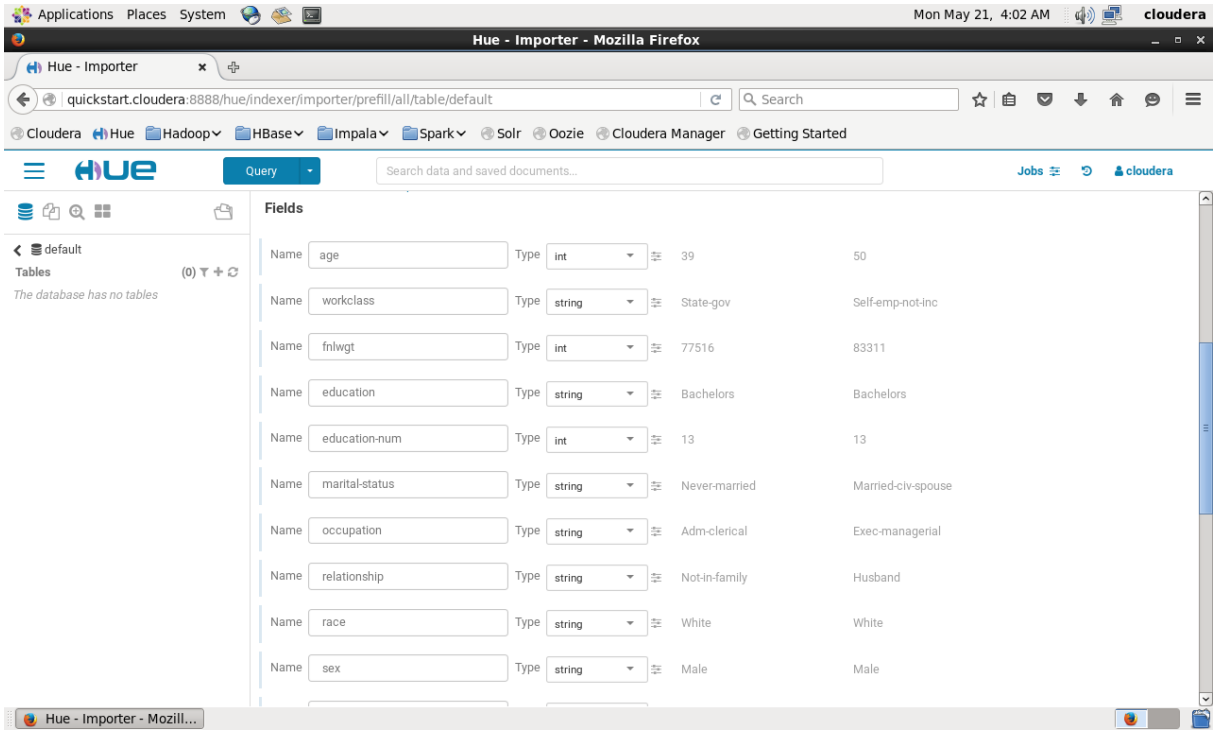
Şekil 3.23: Oluşturulacak tablo için alan ayırıcı ve kayıt ayırıcının ayarlanması (1).



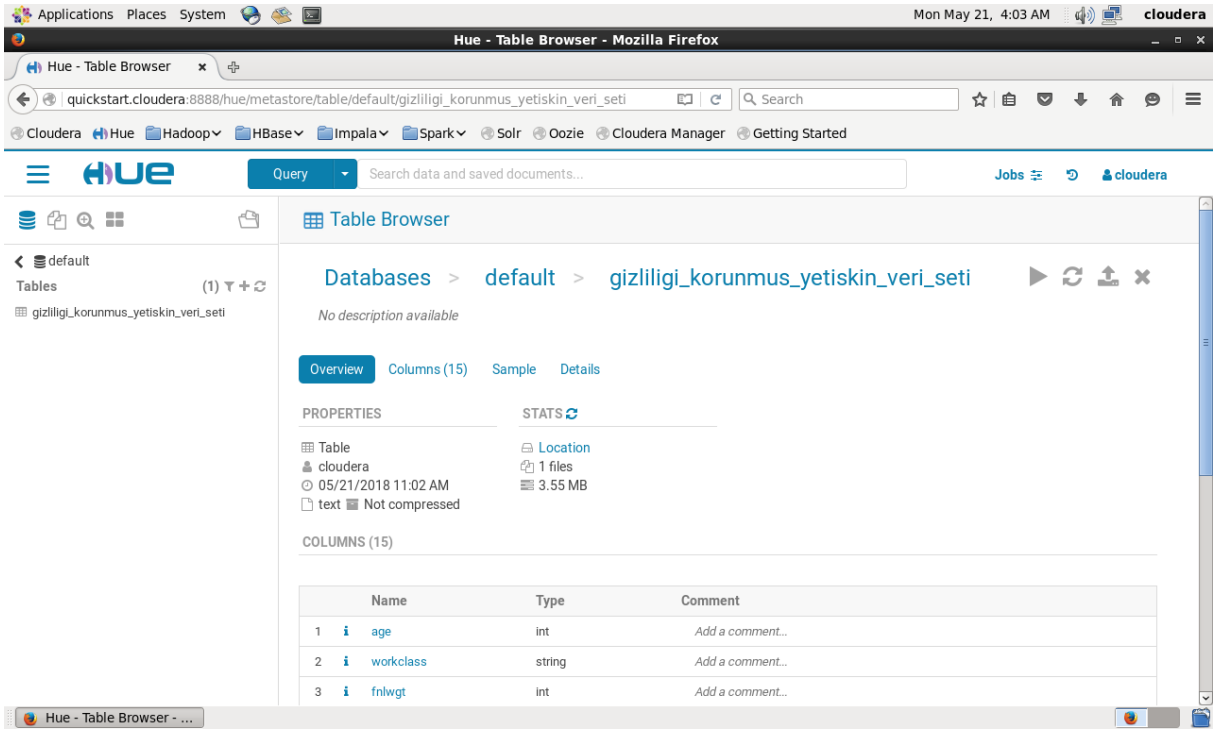
Şekil 3.24: Oluşturulacak tablo için alan ayırıcı ve kayıt ayırıcının ayarlanması (2).



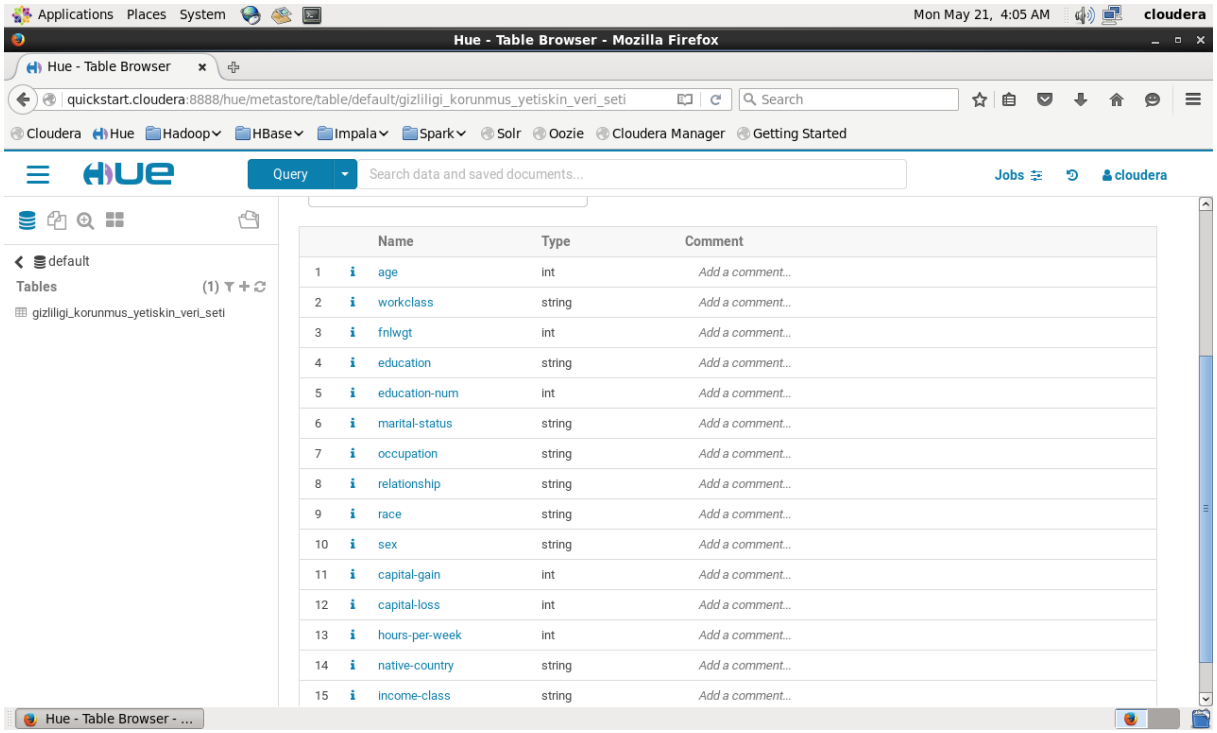
Şekil 3.25: Oluşturulacak tablo için alan türlerinin belirlenmesi (1).



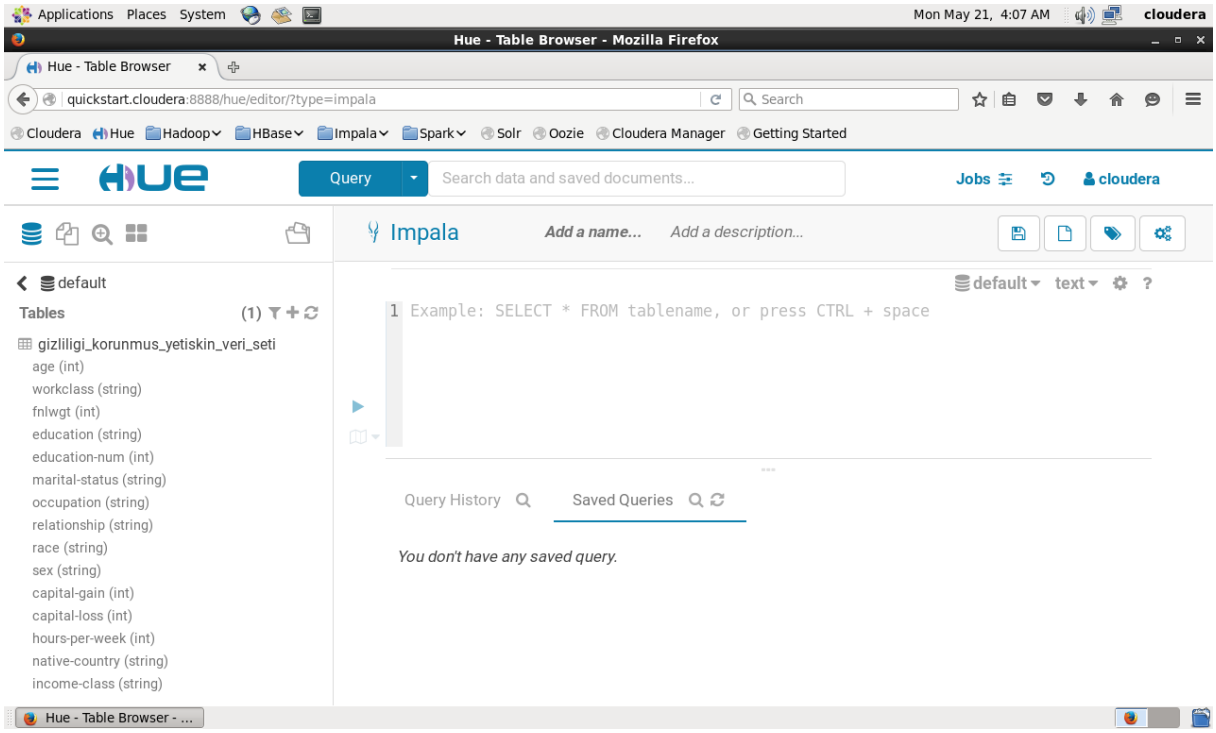
Şekil 3.26: Oluşturulacak tablo için alan türlerinin belirlenmesi (2).



Şekil 3.27: Hadoop ortamı üzerinde oluşturulan tablonun bilgileri (1).



Şekil 3.28: Hadoop ortamı üzerinde oluşturulan tablonun bilgileri (2).



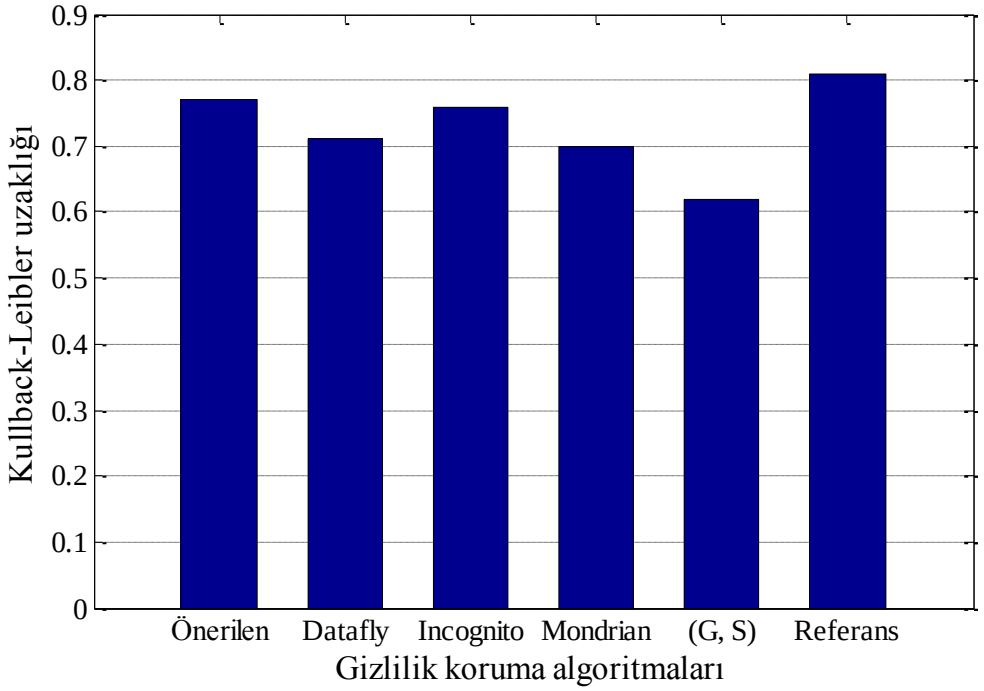
Şekil 3.29: Hadoop ortamındaki büyük veri setleri üzerinde Impala ile sorgu çekilmesi.

4. BULGULAR

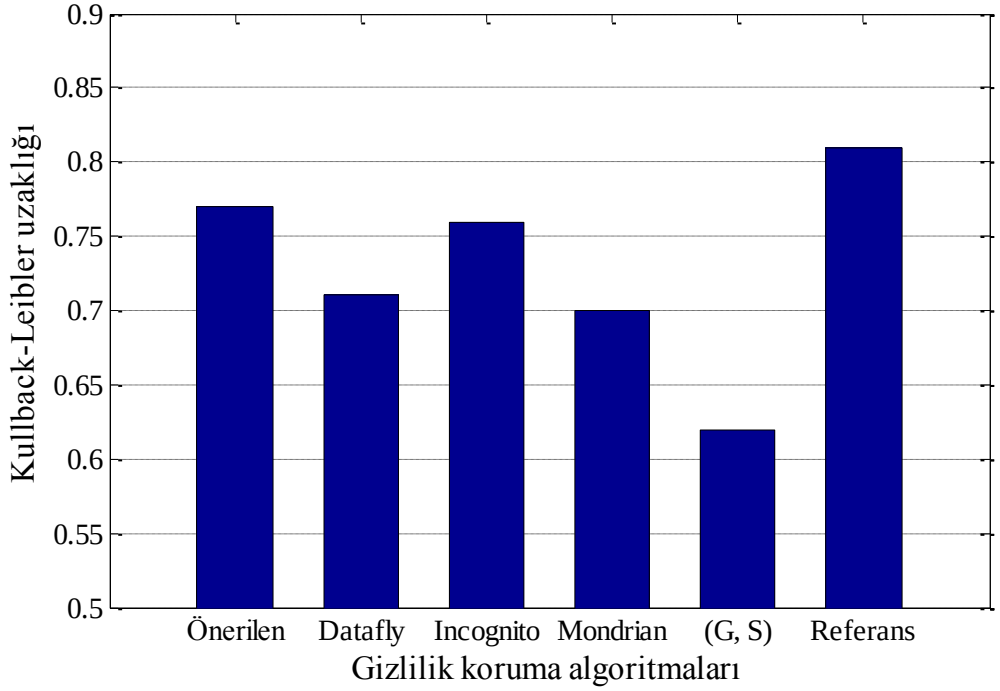
Tez kapsamında önerilen algoritma 16GB RAM ve Intel Core i7-3820 (3.60 GHz) işlemcili bir bilgisayarda Windows 7 64-bit işletim sistemi üzerinde çalışan MATLAB R2016a'da gerçeklenmiştir. Önerilen algoritmanın sınıflandırma doğruluğu ve F-ölçütü sonuçları, Weka 3.8'de çeşitli sınıflandırma teknikleri kullanılarak elde edilmiştir. Bu bölümde; önerilen gizlilik koruma algoritmasının Kullback-Leibler uzaklığı, olasılıksal anonimlik, sınıflandırma doğruluğu, F-ölçütü, yürütme süresi ve Impala sorgu sonuçları yer almaktadır.

4.1. KULLBACK-LEIBLER UZAKLIĞI SONUÇLARI

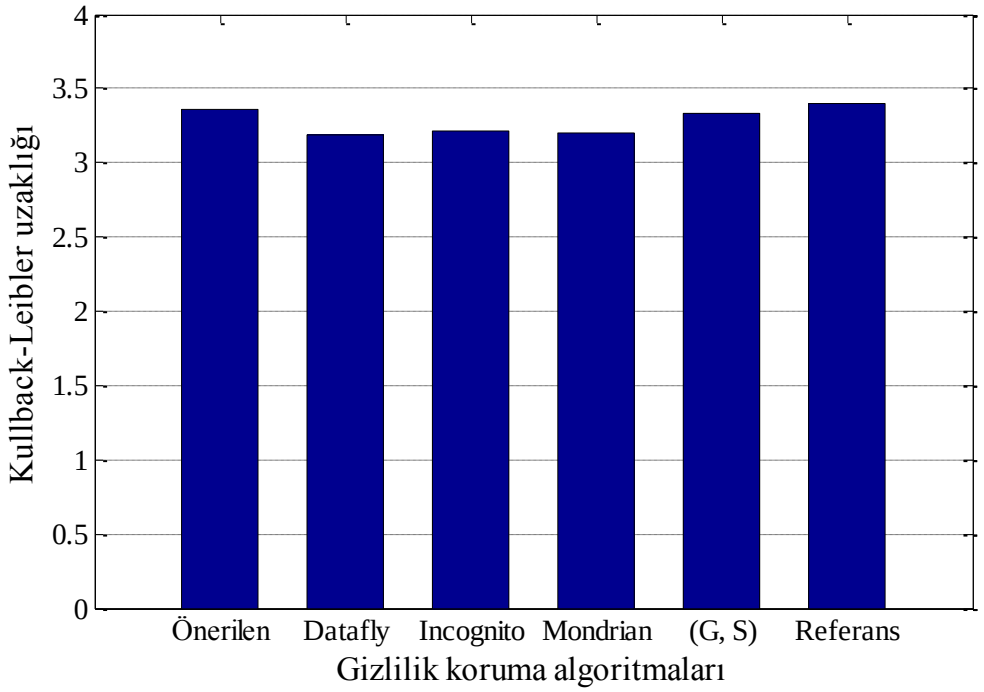
Bu çalışmada önerilen gizlilik koruma algoritmasının Kullback-Leibler uzaklığı performansı literatürde var olan Datafly (Sweeney, 1998), Incognito (LeFevre ve diğ., 2005), Mondrian (LeFevre ve diğ., 2006) ve (G, S) (Nayahi ve Kavitha, 2015) algoritmaları ile kıyaslanmıştır. Önerilen algoritmanın Kullback-Leibler uzaklığının mevcut algoritmalarla karşılaştırılması test durumu I ve II için Şekil 4.1-4.4'te gösterilmektedir.



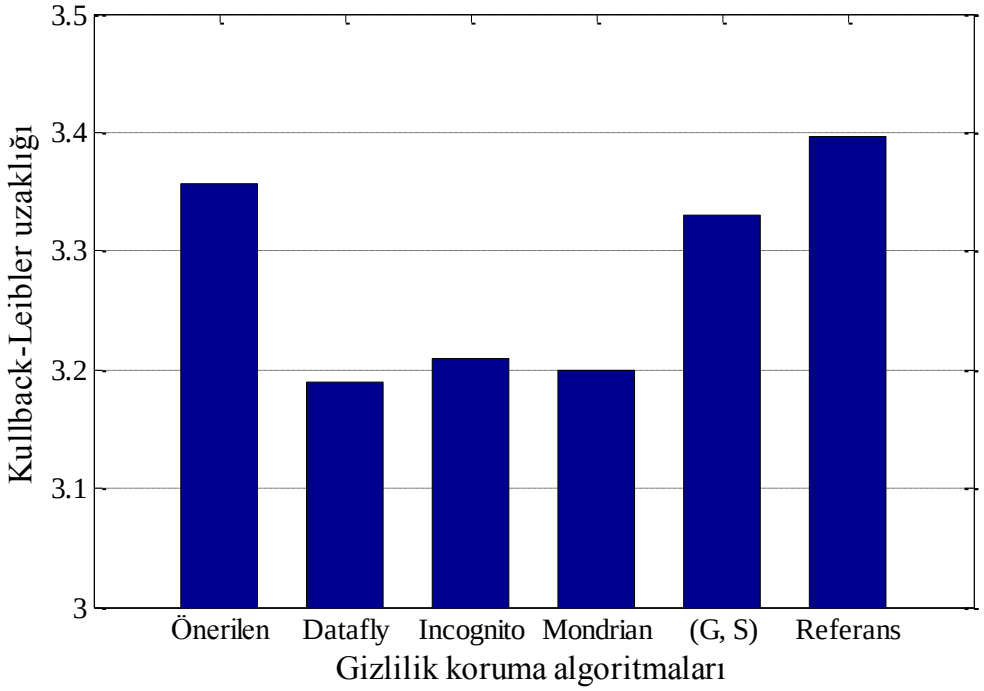
Şekil 4.1: Önerilen algoritmanın Kullback-Leibler uzaklığının mevcut yöntemlerle karşılaştırılması (Test durumu I).



Şekil 4.2: Önerilen algoritmanın Kullback-Leibler uzaklığının mevcut yöntemlerle karşılaştırılması (Test durumu I, KL uzaklığının 0.5-0.9 aralığında gösterimi).



Şekil 4.3: Önerilen algoritmanın Kullback-Leibler uzaklığının mevcut yöntemlerle karşılaştırılması (Test durumu II).



Şekil 4.4: Önerilen algoritmanın Kullback-Leibler uzaklığının mevcut yöntemlerle karşılaştırılması (Test durumu II, KL uzaklığının 3.0-3.5 aralığında gösterimi).

Referans değeri (baseline), orijinal Yetişkin veri setindeki hassas niteliğin entropisidir. Şekil 4.1-4.4'te görüldüğü üzere önerilen algoritmanın KL uzaklığı mevcut algoritmalardan daha iyidir ve referans değerine çok yakındır. Bu sonuç, önerilen algoritmanın orijinal veri setini çok az oranda bozduğunu göstermektedir. Ayrıca önerilen algoritma, her bir yarı tanımlayıcı için benzersiz nitelik değerlerinin sıklık analizine bağlı olarak sadece belirlenen kritik değerler için pertürbasyon işleminin gerçekleştirilmesi sebebiyle daha yüksek veri kullanılabilirliğine sahiptir.

4.2. OLASILIKSAL ANONİMLİK SONUÇLARI

Önerilen algoritmanın Yetişkin veri seti için olasılıksal anonimliği denklem (3.6) kullanılarak hesaplanmıştır ve buna karşılık gelen değer 24,53'tür. Yetişkin veri setindeki rastgele bir kayıt için bir yarı tanımlayıcının orijinal değerinin tahmin olasılığı 0,04'tür. Bu sonuçlar önerilen algoritmanın olasılıksal anonimliğinin oldukça iyi olduğunu göstermektedir.

4.3. SINIFLANDIRMA DOĞRULUĞU SONUÇLARI

Önerilen algoritmanın sınıflandırma doğruluğu; Voted Perceptron (VP), OneR, Naive Bayes (NB) ve C4.5 (J48) Karar Ağacı olmak üzere dört farklı sınıflandırıcı kullanılarak incelenmiştir. Önerilen algoritmanın farklı boyutlara sahip beş veri seti için *k*-kat çapraz geçerleme tekniği kullanılarak elde edilen sınıflandırma doğruluğu sonuçları Tablo 4.1-4.3'te verilmiştir. 2-kat, 5-kat ve 10-kat çapraz geçerleme her bir sınıflandırıcı için gerçekleştirilmiştir. Önerilen algoritmanın uygulandığı veri setlerinin orijinal ve gizliliği korunmuş formlarının sınıflandırma doğrulukları, önerilen algoritmanın performansını değerlendirmek için birbirleriyle karşılaştırılmıştır. Daha yüksek sınıflandırma doğruluğu değerleri tercih edilmektedir. Ayrıca orijinal değerlere daha yakın olan sınıflandırma doğruluğu değerleri, bilgi kaybının düşük olduğu anlamına gelmektedir ve bu da daha yüksek veri kullanılabilirliği olduğunu göstermektedir.

Tablo 4.1: Önerilen algoritmanın çeşitli veri setleri için sınıflandırma doğruluğu sonuçları (2-kat çapraz geçerleme).

Veri setleri		2-kat çapraz geçerleme			
		VP	**OneR**	**NB**	**J48**
Yetişkin	**Orijinal**	77,84	80,21	82,75	85,03
	Gizliliği korunmuş	77,84	80,21	82,55	85,14
~60K	**Orijinal**	78,41	80,24	82,78	87,19
	Gizliliği korunmuş	78,41	80,24	82,58	86,92
~120K	**Orijinal**	78,45	78,16	82,83	92,15
	Gizliliği korunmuş	78,45	78,16	82,62	92,31
~240K	**Orijinal**	78,47	83,24	82,87	98,41
	Gizliliği korunmuş	78,47	83,24	82,65	98,39
~480K	**Orijinal**	78,40	88,69	82,90	99,86
	Gizliliği korunmuş	78,40	88,69	82,66	99,85

Tablo 4.2: Önerilen algoritmanın çeşitli veri setleri için sınıflandırma doğruluğu sonuçları (5-kat çapraz geçerleme).

Veri setleri		5-kat çapraz geçerleme			
		VP	**OneR**	**NB**	**J48**
Yetişkin	**Orijinal**	78,36	80,21	82,84	85,71
	Gizliliği korunmuş	78,36	80,21	82,59	85,54
~60K	**Orijinal**	78,44	75,45	82,90	88,94
	Gizliliği korunmuş	78,44	75,45	82,65	88,73
~120K	**Orijinal**	78,46	81,20	82,88	96,95
	Gizliliği korunmuş	78,46	81,20	82,66	96,86
~240K	**Orijinal**	78,43	86,04	82,90	99,84
	Gizliliği korunmuş	78,43	86,04	82,65	99,83
~480K	**Orijinal**	78,42	89,30	82,90	99,98
	Gizliliği korunmuş	78,42	89,30	82,66	99,98

Tablo 4.3: Önerilen algoritmanın çeşitli veri setleri için sınıflandırma doğruluğu sonuçları (10-kat çapraz geçerleme).

Veri setleri		10-kat çapraz geçerleme			
		VP	**OneR**	**NB**	**J48**
Yetişkin	**Orijinal**	78,42	80,22	82,88	85,73
	Gizliliği korunmuş	78,42	80,22	82,64	85,69
~60K	**Orijinal**	78,43	75,54	82,87	89,43
	Gizliliği korunmuş	78,43	75,54	82,64	89,31
~120K	**Orijinal**	78,45	82,31	82,89	98,13
	Gizliliği korunmuş	78,45	82,31	82,65	98,18
~240K	**Orijinal**	78,44	87,09	82,90	99,89
	Gizliliği korunmuş	78,44	87,09	82,65	99,89
~480K	**Orijinal**	78,44	88,73	82,90	99,99
	Gizliliği korunmuş	78,44	88,73	82,66	99,99

Tablo 4.1-4.3'te görüldüğü üzere *k* (kat sayısı) değerindeki artış her bir veri seti için genel olarak sınıflandırma doğruluğunda küçük bir artışa neden olmaktadır. Gizliliği korunmuş veri setlerinin sınıflandırma doğruluğu her bir veri seti için orijinal doğruluklarla aynıdır veya çok yakındır. Orijinal ve gizliliği korunmuş veri setlerinin sınıflandırma doğrulukları, Voted Perceptron ve OneR sınıflandırıcıları için eşittir. Naive Bayes ve J48 sınıflandırıcıları için ise neredeyse aynıdır. Ek olarak en iyi doğruluk değerleri her bir veri seti için J48 sınıflandırıcı kullanılarak elde edilmiştir.

Önerilen algoritmanın sınıflandırma doğruluğunun; aynı veri seti, yarı tanımlayıcılar, hassas nitelik ve sınıflandırma algoritmaları için literatürde var olan metotlar olan Datafly (Sweeney, 1998), Incognito (LeFevre ve diğ., 2005), Mondrian (LeFevre ve diğ., 2006), Entropi *l*-çeşitlilik (Machanavajjhala ve diğ., 2007), (G, S) (Nayahi ve Kavitha, 2015) ve KNN-(G, S) (Nayahi ve Kavitha, 2017) yöntemleri ile 10-kat çapraz geçerleme tekniği kullanılarak karşılaştırılması Tablo 4.4'te gösterilmektedir.

Tablo 4.4: Önerilen algoritmanın sınıflandırma doğruluğunun mevcut yöntemler ile karşılaştırılması.

Gizlilik koruma algoritmaları	*k*	Sınıflandırma algoritmaları			
		VP	**OneR**	**NB**	**J48**
Orijinal Yetişkin veri seti	-	78,42	80,22	82,88	85,73
Datafly	5	78,36	80,18	82,85	85,35
Incognito	5	78,38	80,17	82,75	85,30
Mondrian	5	78,38	80,17	82,83	85,00
Entropi *l*-çeşitlilik (*l* = 2)	5	78,38	80,17	82,40	85,42
(G, S)	5	78,43	80,21	83,46	85,16
KNN-(G, S)	5	78,38	80,16	82,72	85,26
Datafly	10	78,38	80,18	82,85	85,35
Incognito	10	78,38	80,15	82,44	85,30
Mondrian	10	78,38	80,17	82,83	84,97
Entropi *l*-çeşitlilik (*l* = 2)	10	78,37	80,18	82,40	85,40
(G, S)	10	78,43	80,21	83,46	85,16
KNN-(G, S)	10	78,38	80,16	83,72	85,26
Datafly	25	78,38	80,18	82,85	85,38
Incognito	25	78,38	80,17	82,71	85,31
Mondrian	25	78,38	80,17	82,84	84,99
Entropi *l*-çeşitlilik (*l* = 2)	25	78,38	80,17	82,40	85,42
(G, S)	25	78,44	80,20	82,12	85,16
KNN-(G, S)	25	78,39	80,19	83,01	85,40
Datafly	50	78,38	80,17	83,11	85,37
Incognito	50	78,38	80,17	82,71	85,31
Mondrian	50	78,38	80,17	82,85	85,05
Entropi *l*-çeşitlilik (*l* = 2)	50	78,38	80,17	82,40	84,42
(G, S)	50	78,42	80,17	83,44	85,35
KNN-(G, S)	50	78,39	80,11	83,50	85,69
Önerilen algoritma	**-**	**78,42**	**80,22**	**82,64**	**85,69**

Önerilen algoritmanın sınıflandırma doğruluğunun Voted Perceptron, OneR ve J48 sınıflandırıcılarının tüm *k* (*k*-anonimlik) durumları için mevcut algoritmalardan daha iyi olduğu Tablo 4.4'ten görülebilmektedir. Burada, önerilen algoritmanın sınıflandırma doğruluğu sadece *k*'nın 50 olduğu durumda KNN-(G, S) algoritması ile aynıdır. Önerilen gizlilik koruma

algoritmasının performansı, Voted Perceptron ve OneR sınıflandırıcılarında orijinal Yetişkin veri setiyle aynıdır. Buna ek olarak önerilen algoritmanın sınıflandırma doğruluğu Naive Bayes ve J48 sınıflandırıcılarında orijinal değer ile neredeyse eşittir. J48 sınıflandırıcısı tüm algoritmalar için en iyi doğruluk sonuçlarını vermektedir. Ayrıca önerilen algoritmanın Yetişkin veri seti için 10-kat çapraz geçerlemede Voted Perceptron, OneR, Naive Bayes ve J48 sınıflandırıcıları kullanılarak elde dilen konfüzyon matrisleri Şekil 4.5'te gösterilmektedir.

Tahmin edilen sınıf

Gerçek sınıf	**> 50K**	**≤ 50K**	**Toplam**
> 50K	1172	6336	7508
≤ 50K	174	22480	22654
Toplam	1346	28816	30162

(a)

Tahmin edilen sınıf

Gerçek sınıf	**> 50K**	**≤ 50K**	**Toplam**
> 50K	1584	5924	7508
≤ 50K	42	22612	22654
Toplam	1626	28536	30162

(b)

Tahmin edilen sınıf

Gerçek sınıf	**> 50K**	**≤ 50K**	**Toplam**
> 50K	3741	3767	7508
≤ 50K	1468	21186	22654
Toplam	5209	24953	30162

(c)

Tahmin edilen sınıf

Gerçek sınıf	**> 50K**	**≤ 50K**	**Toplam**
> 50K	4778	2730	7508
≤ 50K	1586	21068	22654
Toplam	6364	23798	30162

(d)

Şekil 4.5: Konfüzyon matrisleri: (a) Voted Perceptron, (b) OneR, (c) Naive Bayes, (d) J48.

4.4. F-ÖLÇÜTÜ SONUÇLARI

Önerilen algoritmanın F-ölçütü performansını analiz etmek için dört sınıflandırma yöntemi kullanılmıştır. Farklı boyutlu beş veri seti için önerilen algoritmanın *k*-kat çapraz geçerleme tekniği kullanılarak elde edilen F-ölçütü sonuçları Tablo 4.5-4.7'de gösterilmektedir. Her bir sınıflandırma yöntemi için 2-kat, 5-kat ve 10-kat çapraz geçerleme gerçekleştirilmiştir. Veri setlerinin orijinal ve gizliliği korunmuş versiyonlarının F-ölçütü değerleri, önerilen algoritmanın performansını değerlendirmek için birbirleriyle karşılaştırılmıştır. Daha yüksek F-ölçütü değerleri tercih edilmektedir ve orijinallere daha yakın F-ölçütü değerleri daha iyidir.

Tablo 4.5: Önerilen algoritmanın çeşitli veri setleri için F-ölçütü sonuçları (2-kat çapraz geçerleme).

Veri setleri		2-kat çapraz geçerleme			
		VP	**OneR**	**NB**	**J48**
Yetişkin	**Orijinal**	0,709	0,750	0,817	0,845
	Gizliliği korunmuş	0,709	0,750	0,814	0,845
~60K	**Orijinal**	0,723	0,750	0,818	0,869
	Gizliliği korunmuş	0,723	0,750	0,814	0,866
~120K	**Orijinal**	0,724	0,765	0,818	0,920
	Gizliliği korunmuş	0,724	0,765	0,815	0,922
~240K	**Orijinal**	0,724	0,825	0,819	0,984
	Gizliliği korunmuş	0,724	0,825	0,815	0,984
~480K	**Orijinal**	0,722	0,886	0,819	0,999
	Gizliliği korunmuş	0,722	0,886	0,815	0,998

Tablo 4.6: Önerilen algoritmanın çeşitli veri setleri için F-ölçütü sonuçları (5-kat çapraz geçerleme).

Veri setleri		5-kat çapraz geçerleme			
		VP	**OneR**	**NB**	**J48**
Yetişkin	**Orijinal**	0,721	0,750	0,818	0,853
	Gizliliği korunmuş	0,721	0,750	0,814	0,851
~60K	**Orijinal**	0,723	0,729	0,819	0,887
	Gizliliği korunmuş	0,723	0,729	0,815	0,885
~120K	**Orijinal**	0,724	0,803	0,819	0,969
	Gizliliği korunmuş	0,724	0,803	0,815	0,968
~240K	**Orijinal**	0,723	0,858	0,819	0,998
	Gizliliği korunmuş	0,723	0,858	0,815	0,998
~480K	**Orijinal**	0,723	0,893	0,819	1,000
	Gizliliği korunmuş	0,723	0,893	0,815	1,000

Tablo 4.7: Önerilen algoritmanın çeşitli veri setleri için F-ölçütü sonuçları (10-kat çapraz geçerleme).

Veri setleri		10-kat çapraz geçerleme			
		VP	**OneR**	**NB**	**J48**
Yetişkin	**Orijinal**	0,722	0,750	0,819	0,853
	Gizliliği korunmuş	0,722	0,750	0,815	0,853
~60K	**Orijinal**	0,723	0,731	0,819	0,892
	Gizliliği korunmuş	0,723	0,731	0,815	0,891
~120K	**Orijinal**	0,723	0,816	0,819	0,981
	Gizliliği korunmuş	0,723	0,816	0,815	0,982
~240K	**Orijinal**	0,723	0,870	0,819	0,999
	Gizliliği korunmuş	0,723	0,870	0,815	0,999
~480K	**Orijinal**	0,723	0,887	0,819	1,000
	Gizliliği korunmuş	0,723	0,887	0,815	1,000

Tablo 4.5-4.7'nin analizinden görüldüğü üzere F-ölçütü değerleri, *k* (kat sayısı) değerinin artışıyla genel olarak her bir veri seti için az oranda artmaktadır. Önerilen algoritma; Voted Perceptron, OneR ve Naive Bayes ile karşılaştırıldığında J48 sınıflandırma tekniği ile en iyi F-ölçütü değerlerini elde etmektedir. Gizliliği korunmuş veri setlerinin F-ölçütü, tüm veri setleri için orijinal değerlerle aynıdır veya çok yakındır. Orijinal ve gizliliği korunmuş veri setlerinin F-ölçütleri, Voted Perceptron ve OneR sınıflandırıcıları için eşittir. Naive Bayes ve J48 sınıflandırıcıları için ise neredeyse aynıdır. Önerilen algoritmanın F-ölçütünün, 10-kat çapraz geçerlemede aynı deney koşulları için mevcut yöntemler ile karşılaştırması Tablo 4.8'de gösterilmektedir.

Tablo 4.8: Önerilen algoritmanın F-ölçütünün mevcut yöntemler ile karşılaştırılması.

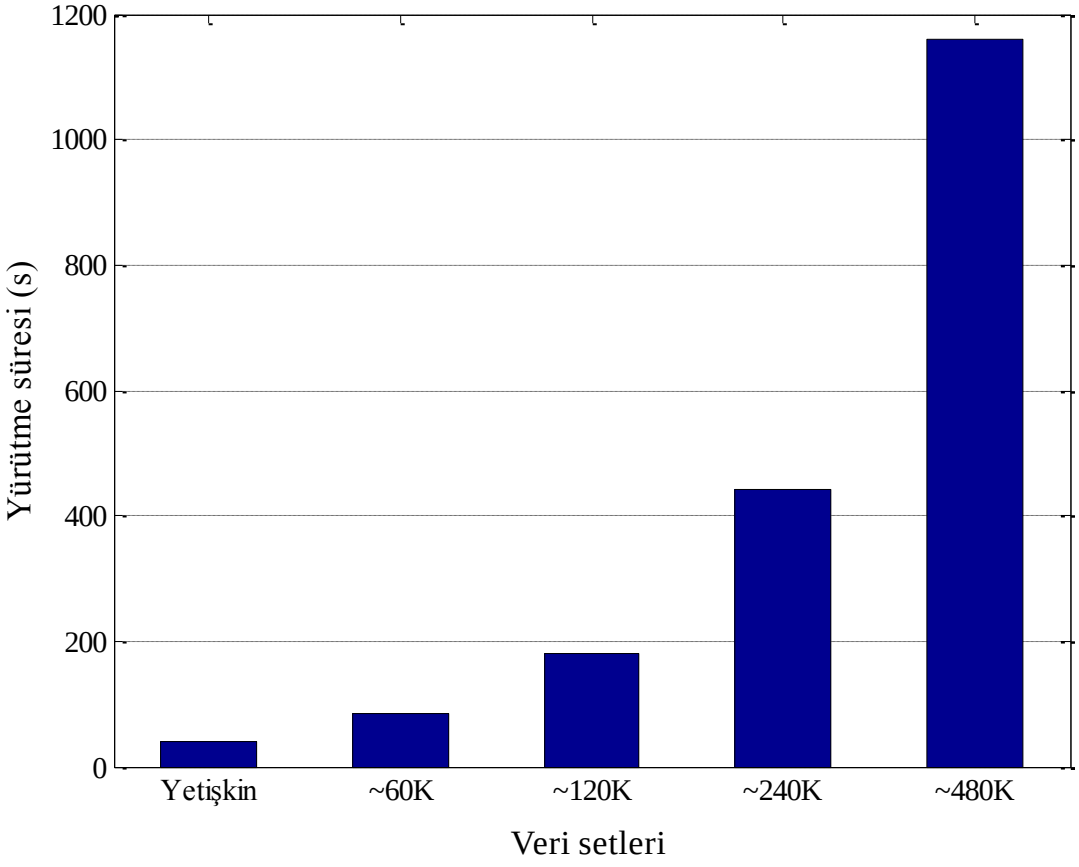
Gizlilik koruma algoritmaları	*k*	Sınıflandırma algoritmaları			
		VP	**OneR**	**NB**	**J48**
Orijinal Yetişkin veri seti	-	0,722	0,750	0,819	0,853
Datafly	5	0,722	0,750	0,819	0,850
Incognito	5	0,722	0,749	0,818	0,847
Mondrian	5	0,722	0,749	0,818	0,843
Entropi *l*-çeşitlilik (*l* = 2)	5	0,722	0,749	0,808	0,849
(G, S)	5	0,723	0,750	0,829	0,845
KNN-(G, S)	5	0,722	0,749	0,817	0,847
Datafly	10	0,722	0,749	0,819	0,849
Incognito	10	0,722	0,749	0,812	0,848
Mondrian	10	0,722	0,749	0,818	0,840
Entropi *l*-çeşitlilik (*l* = 2)	10	0,722	0,750	0,808	0,849
(G, S)	10	0,723	0,750	0,829	0,845
KNN-(G, S)	10	0,722	0,749	0,817	0,847
Datafly	25	0,722	0,749	0,819	0,849
Incognito	25	0,722	0,749	0,817	0,847
Mondrian	25	0,722	0,749	0,818	0,840
Entropi *l*-çeşitlilik (*l* = 2)	25	0,722	0,749	0,808	0,849
(G, S)	25	0,723	0,750	0,808	0,845
KNN-(G, S)	25	0,722	0,749	0,822	0,849
Datafly	50	0,722	0,749	0,825	0,848
Incognito	50	0,722	0,749	0,817	0,847
Mondrian	50	0,722	0,749	0,818	0,842
Entropi *l*-çeşitlilik (*l* = 2)	50	0,722	0,749	0,808	0,849
(G, S)	50	0,723	0,749	0,830	0,848
KNN-(G, S)	50	0,722	0,749	0,836	0,853
Önerilen algoritma	**-**	**0,722**	**0,750**	**0,815**	**0,853**

Tablo 4.8'den görüldüğü üzere önerilen algoritma, Voted Perceptron ve OneR sınıflandırma algoritmalarında mevcut algoritmalara kıyasla daha iyi veya eşit performans göstermektedir. J48 sınıflandırıcısında, önerilen algoritmanın performansı tüm mevcut algoritmalardan daha iyidir ve orijinal Yetişkin veri seti ile aynıdır. Burada, önerilen algoritmanın F-ölçütü değeri

sadece *k*'nın 50 olduğu durumda KNN-(G, S) algoritmasının F-ölçütü değeri ile eşittir. Önerilen algoritmanın F-ölçütü, Naive Bayes sınıflandırıcısında orijinale çok yakındır. Ayrıca J48 sınıflandırıcı, tüm algoritmalar için F-ölçütü açısından diğer üç sınıflandırıcıdan daha iyidir.

4.5. YÜRÜTME SÜRESİ SONUÇLARI

Bu çalışmada, farklı boyutlarda beş veri seti, önerilen algoritmanın büyük veri üzerindeki uygulanabilirliğini ve ölçeklenebilirliğini göstermek için kullanılmıştır. Önerilen algoritmanın yürütme süresi performansı, Yetişkin veri seti ve bu veri setinin ~60K, ~120K, ~240K ve ~480K kayıt içeren dört genişletilmiş versiyonu kullanılarak incelenmiştir (Şekil 4.6).



Şekil 4.6: Önerilen algoritmanın çeşitli veri setleri için yürütme süresi performansı.

Şekilde görüldüğü üzere veri setlerindeki kayıt sayısı arttıkça önerilen algoritmanın yürütme süresi de artmaktadır. Buna ek olarak her bir veri seti için yürütme süresi sonuçları, önerilen algoritmanın uygulanabilirlik ve ölçeklenebilirlik açısından ideal olduğunu göstermektedir.

4.6. IMPALA SORGU SONUÇLARI

Bu çalışmada önerilen algoritma ile gizliliği korunmuş veri setleri, Hadoop üzerinde dağıtıklaştırılmıştır. Daha sonra Hadoop ortamındaki bu veri setlerinin incelenmesi için Impala ile sorgular çekilmiştir. Gizliliği korunmuş veri setleri üzerinde örnek Impala sorguları ve sonuçları Şekil 4.7-4.17'de görülebilmektedir.

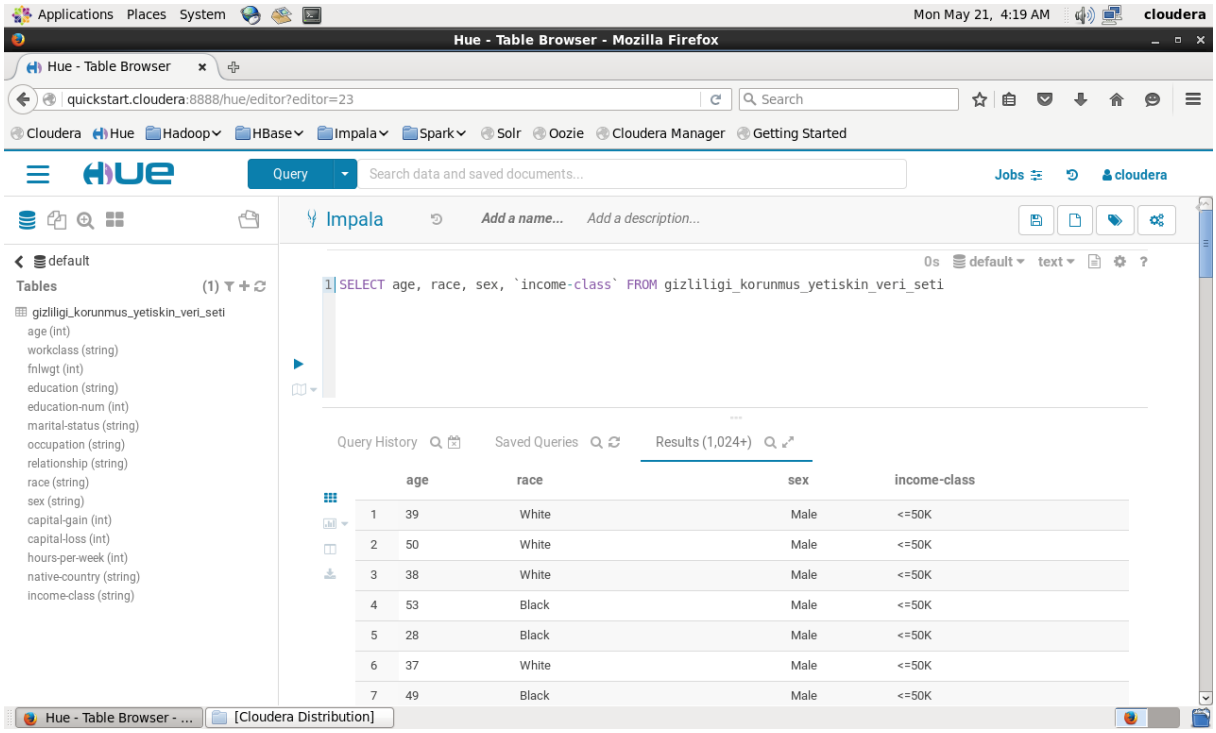
- Impala sorgu örneği (1): SELECT age, race, sex, 'income-class' FROM gizliligi_korunmus_yetiskin_veri_seti (Şekil 4.5).
- Impala sorgu örneği (2): SELECT age, race, sex, occupation, 'income-class' FROM gizliligi_korunmus_yetiskin_veri_seti WHERE age = 34 (Şekil 4.6).
- Impala sorgu örneği (3): SELECT age, race, sex, occupation, 'income-class' FROM gizliligi_korunmus_yetiskin_veri_seti WHERE age = 67 AND race = 'Asian-Pac-Islander' (Şekil 4.7).
- Impala sorgu örneği (4): SELECT age, race, sex, occupation, 'income-class' FROM gizliligi_korunmus_yetiskin_veri_seti WHERE age = 78 AND race = 'White' (Şekil 4.8).
- Impala sorgu örneği (5): SELECT age, race, sex, occupation, 'income-class' FROM gizliligi_korunmus_yetiskin_veri_seti WHERE age = 75 AND race = 'Black' (Şekil 4.9).
- Impala sorgu örneği (6): SELECT age, race, sex, occupation FROM gizliligi_korunmus_yetiskin_veri_seti WHERE age = 70 AND race = 'Black' (Şekil 4.10).
- Impala sorgu örneği (7): SELECT age, race, sex, occupation FROM gizliligi_korunmus_yetiskin_veri_seti WHERE age = 69 AND race = 'Black' (Şekil 4.11).
- Impala sorgu örneği (8): SELECT age, race, sex, occupation FROM gizliligi_korunmus_yetiskin_veri_seti WHERE age = 70 AND race = 'Asian-Pac-Islander' (Şekil 4.12).

- Impala sorgu örneği (9): SELECT age, race, sex, ‘income-class’ FROM gizliligi_korunmus_yetiskin_veri_seti WHERE age = 75 AND race = ‘Asian-Pac-Islander’ (Şekil 4.13).

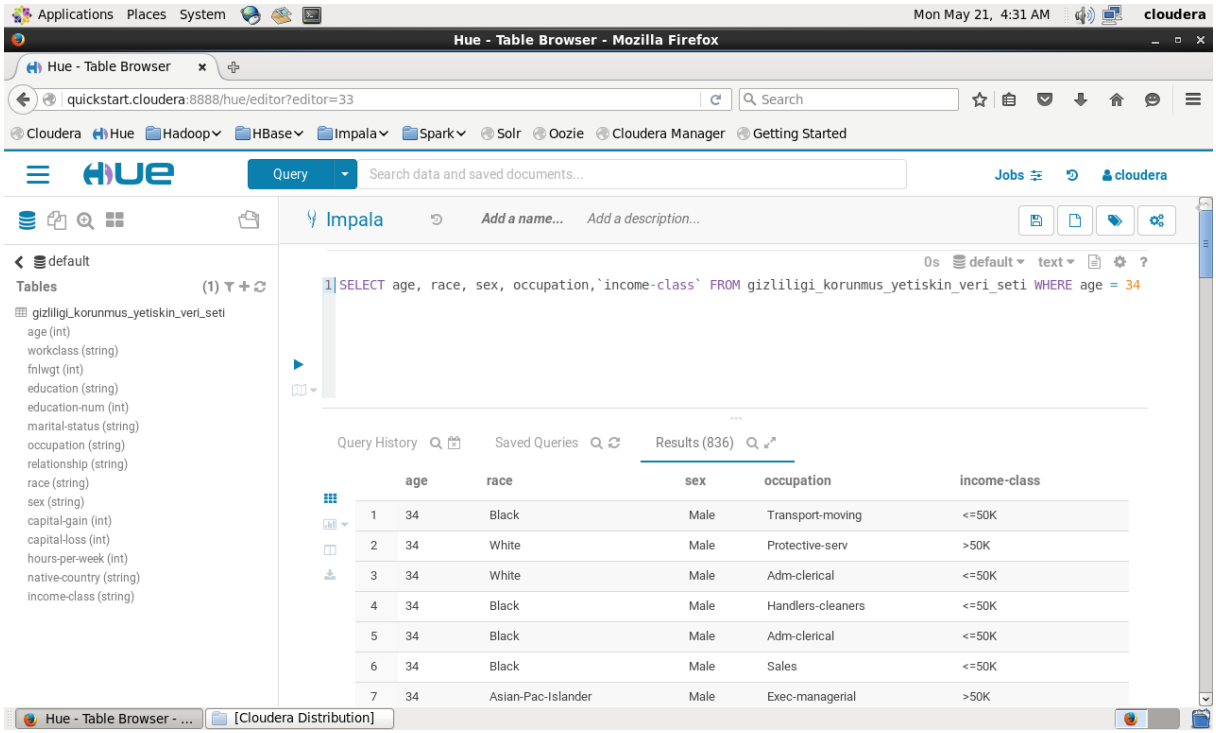
- Impala sorgu örneği (10): SELECT age, race, sex, ‘income-class’ FROM gizliligi_korunmus_yetiskin_veri_seti WHERE age = 75 AND race = ‘Black’ (Şekil 4.14).

- Impala sorgu örneği (11): SELECT age, race, sex, ‘income-class’ FROM gizliligi_korunmus_yetiskin_veri_seti WHERE age = 65 AND race = ‘Black’ (Şekil 4.15).

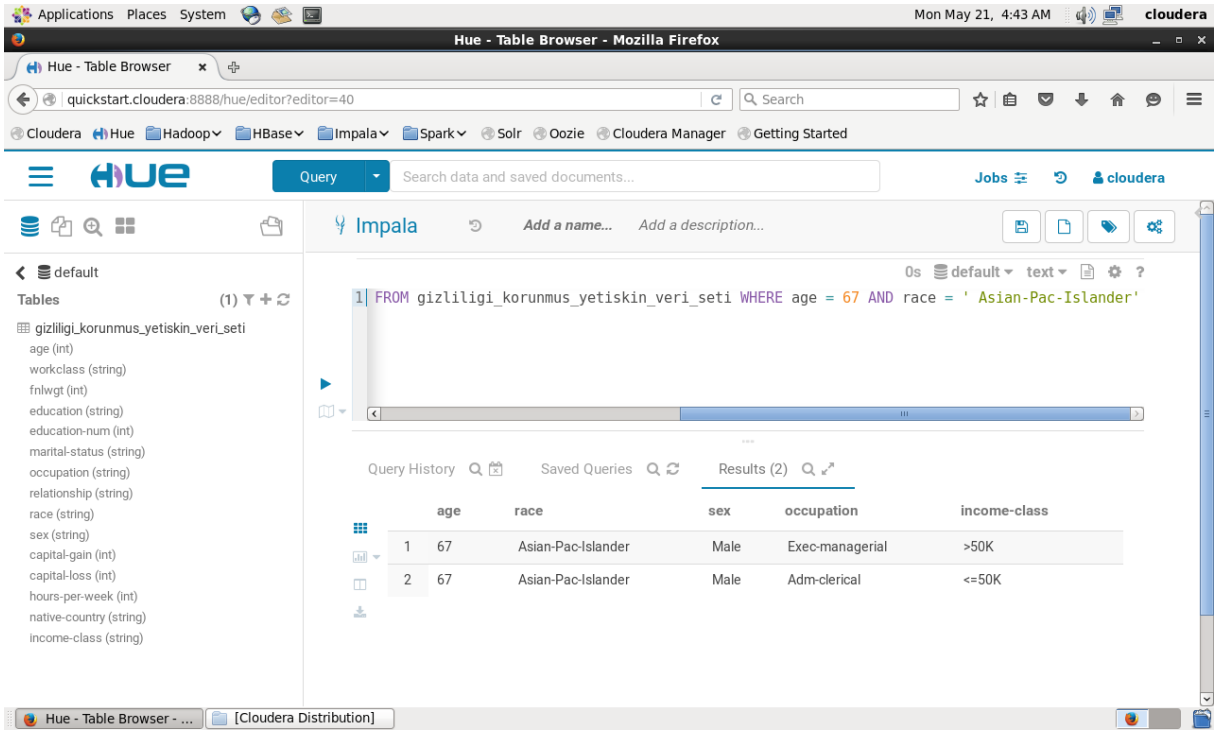
Impala sorgu örneklerinin sonuçlarından görüldüğü üzere önerilen algoritma büyük veri setlerinin gizliliğini etkili bir şekilde korumaktadır.



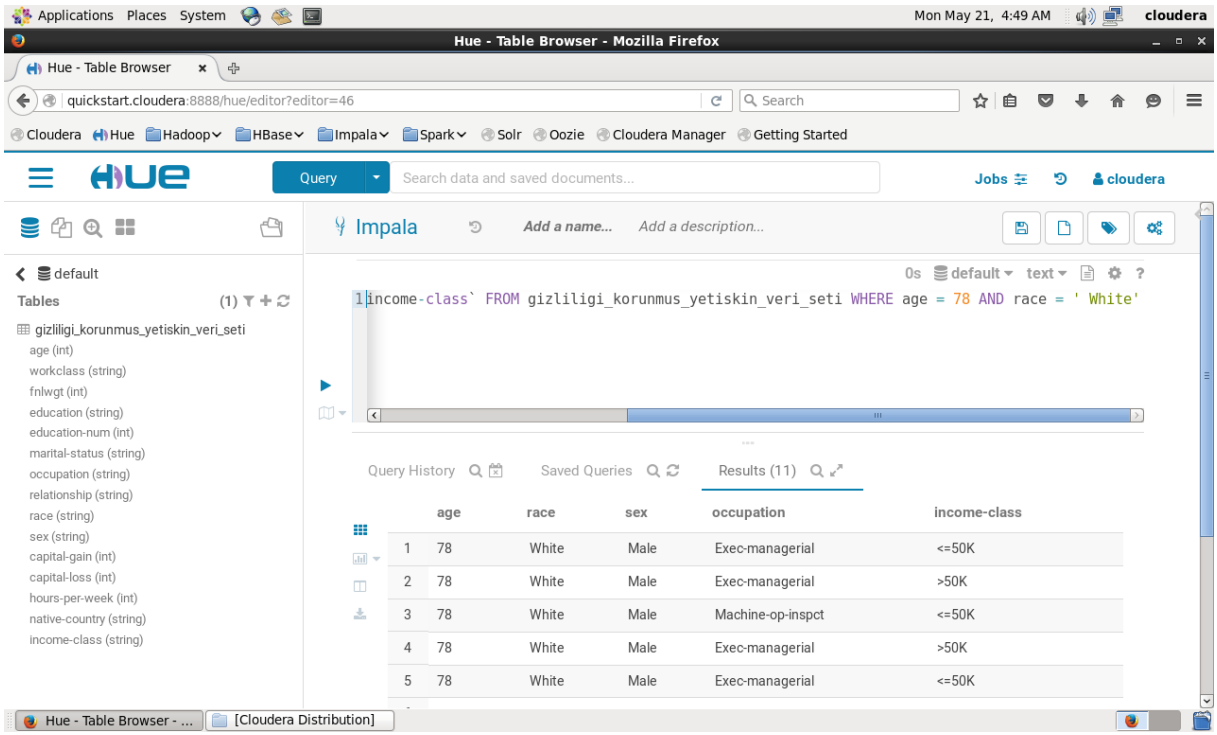
Şekil 4.7: Impala sorgu örneği (1).



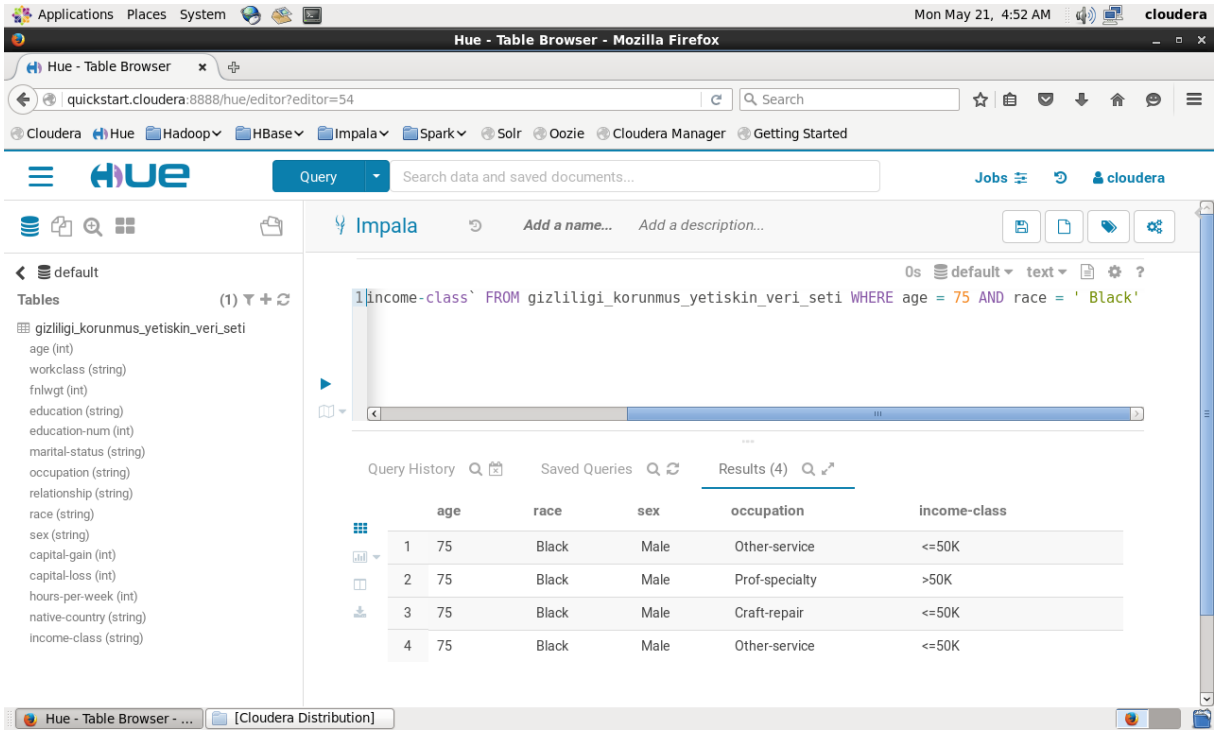
Şekil 4.8: Impala sorgu örneği (2).



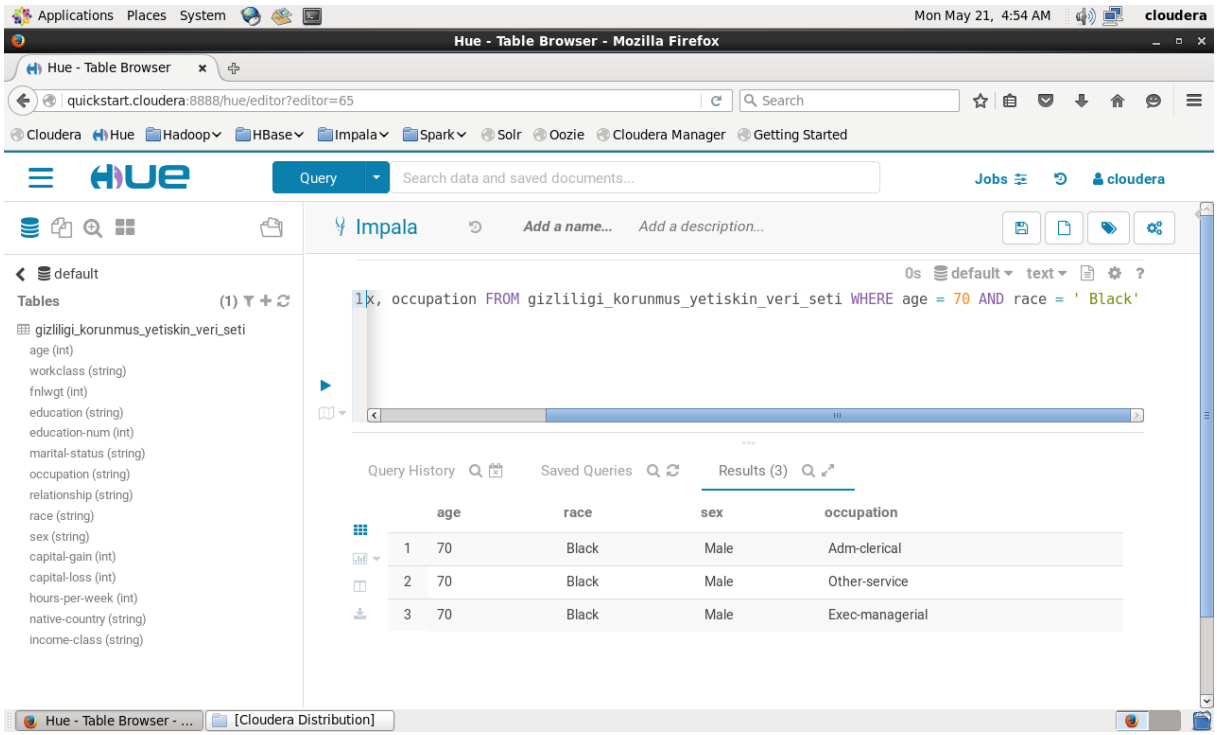
Şekil 4.9: Impala sorgu örneği (3).



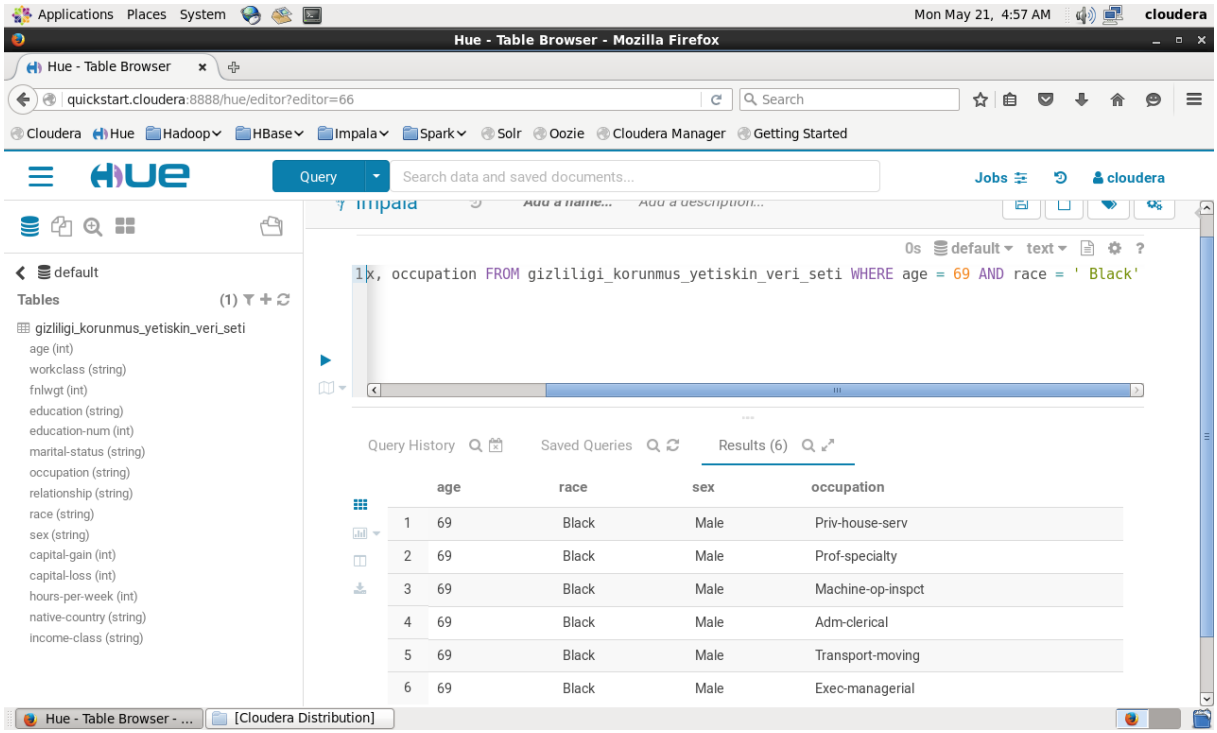
Şekil 4.10: Impala sorgu örneği (4).



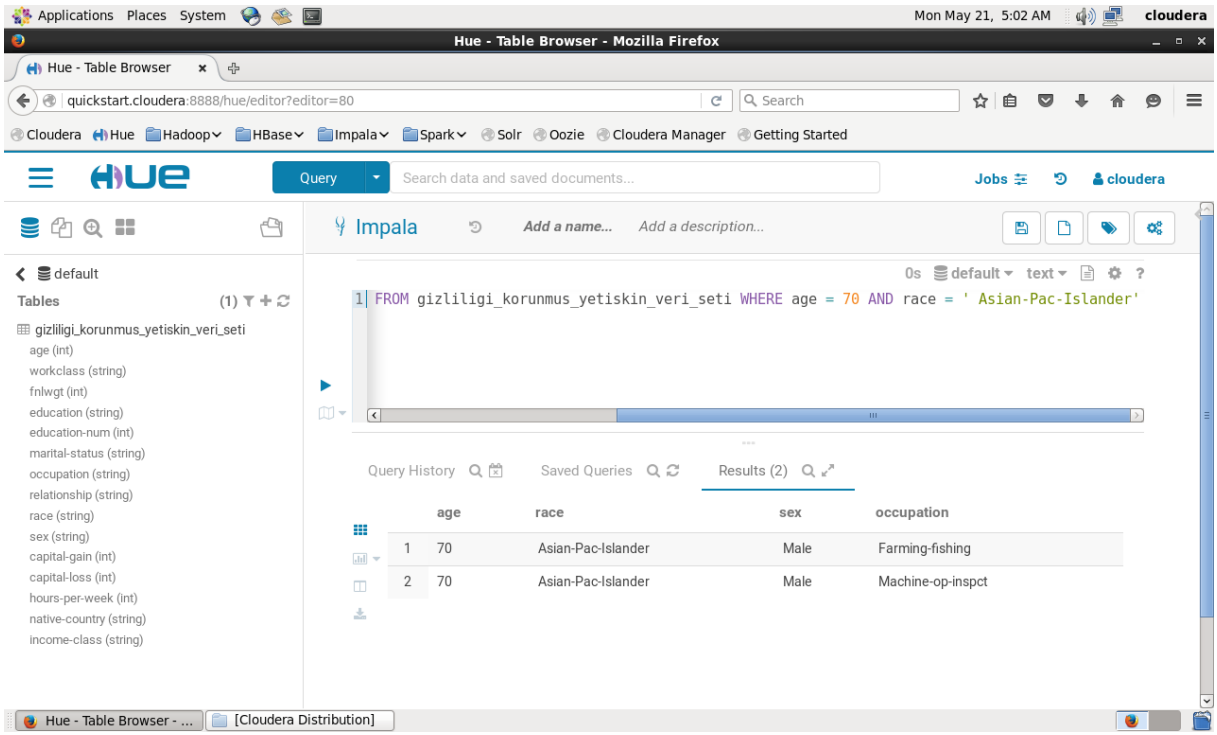
Şekil 4.11: Impala sorgu örneği (5).



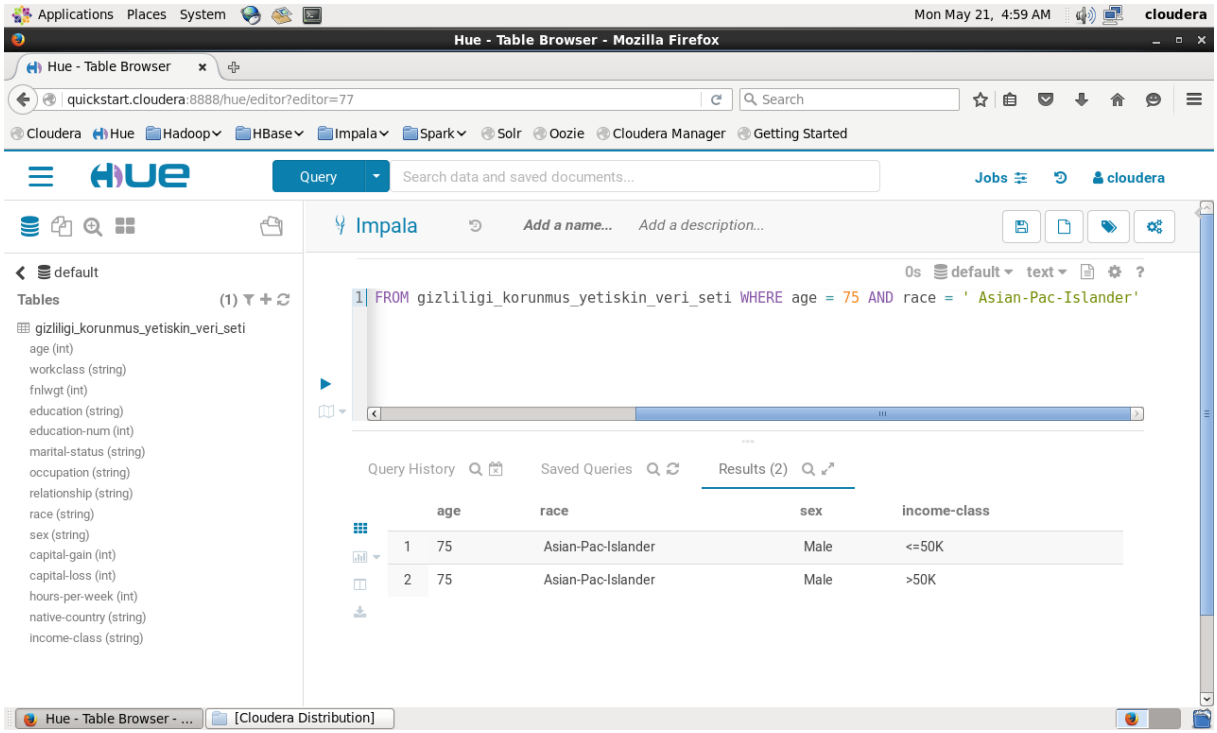
Şekil 4.12: Impala sorgu örneği (6).



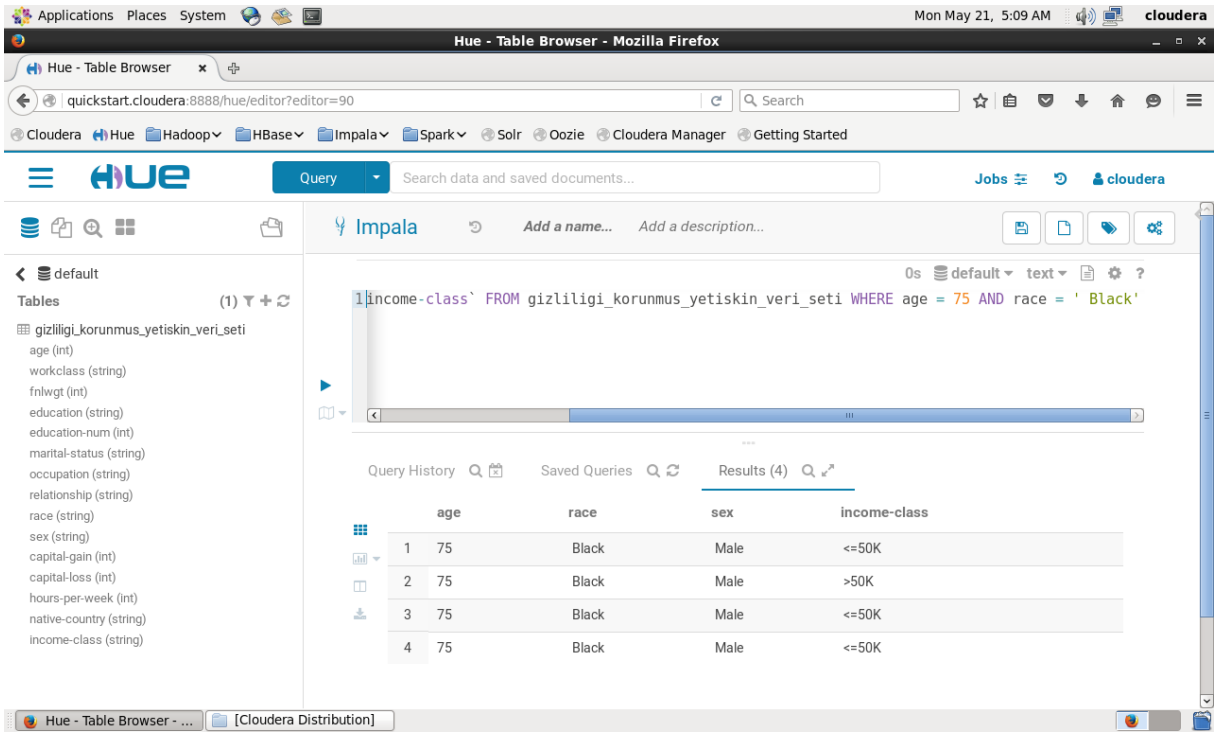
Şekil 4.13: Impala sorgu örneği (7).



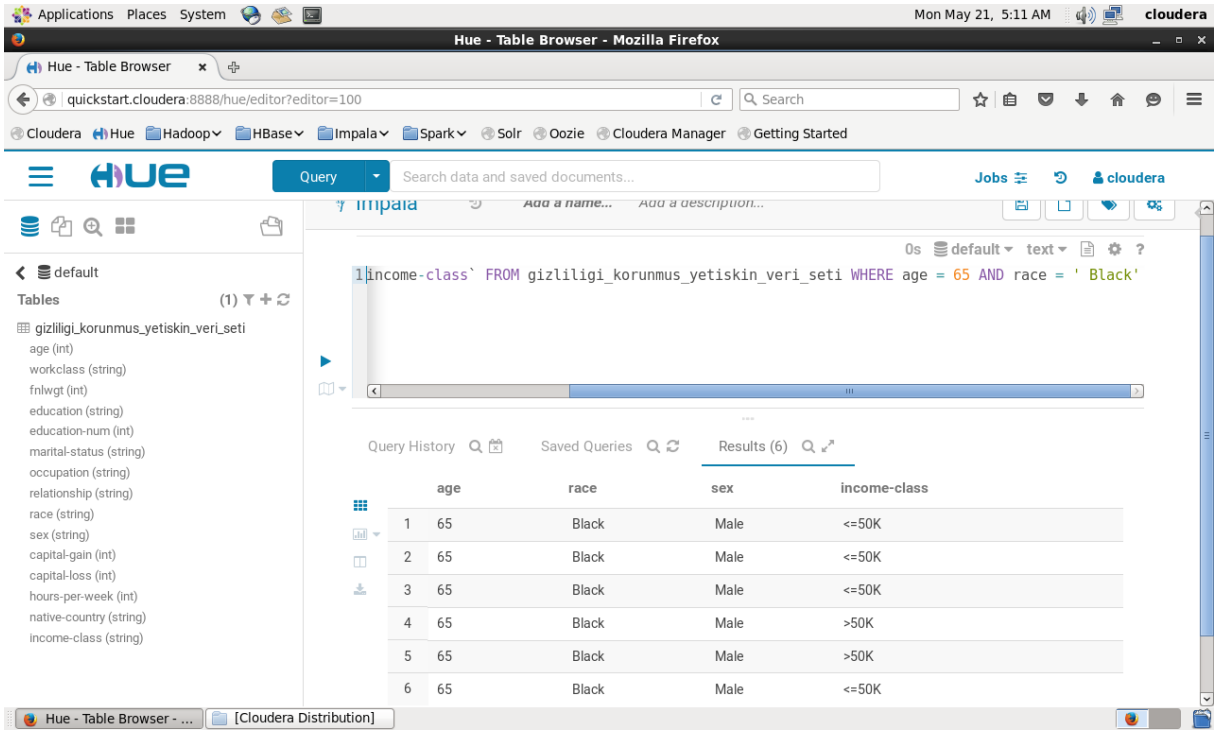
Şekil 4.14: Impala sorgu örneği (8).



Şekil 4.15: Impala sorgu örneği (9).



Şekil 4.16: Impala sorgu örneği (10).



Şekil 4.17: Impala sorgu örneği (11).

5. TARTIŞMA VE SONUÇ

Sosyal ağlar, bulut bilişim, IoT ve veri analitiği gibi yeni teknolojiler günümüzde büyük miktarlarda veri toplamaya olanak sağlamaktadır. Bununla birlikte verilerin tam olarak kullanılabilmesi için son otuz yılda oldukça araştırılan veri gizliliği ve güvenliği konuları büyük önem taşımaktadır. Fakat verileri güvenli hale getirme ve koruma konusunda yeni sorunlar ortaya çıkmıştır. Bu da yeni zorlu araştırma alanlarının oluşmasına sebep olmuştur. Bu zorluklardan bazıları büyük miktardaki verinin kullanımıyla ilgili olarak mahremiyet kaygılarının artması ve mahremiyetin verilerin kullanımı ile bağdaştırılması gerekliliğinden kaynaklanmaktadır. Diğer zorlukların ortaya çıkma sebebi IoT sistemlerinde kullanılanlar gibi yeni veri toplama ve işleme cihazlarının yaygınlaşmasının saldırı potansiyelini artırmasıdır.

Günümüzde veriler, doğası gereği coğrafi olarak çeşitli yerlere dağıtılırlar. Araştırmacılar, bu yerlerde bulunan entegre veri topluluğundan elde edilen bilgilerle ilgilenirler. Örnek olarak toplumda yayılan ani bir salgın hastalık vakası göz önünde bulundurulduğunda, araştırmacılar o hastalığın nedenlerini, belirtilerini ve tedavi yöntemlerini araştırırlar. Dünya ya da ülke çapındaki hastanelerden toplanan hasta verileri üzerinde yapılan veri madenciliği, tek bir hastaneden alınan veriler üzerinde yapılandan daha etkili ve verimli olacaktır. Bu noktada büyük veri kavramı ortaya çıkmaktadır.

Büyük verinin doğuşu, verilerin işlenmesi ve yayınlanması sırasında veri mahremiyeti için kullanılan mevcut koruma yöntemleri açısından yeni zorluklara neden olmaktadır. Gizlilik korumasındaki en büyük zorluk, yayınlanacak ya da paylaşılacak veri setlerinin kullanılabilirliğini sağlarken kişilerin özel bilgilerini korumaktır.

Bu tez çalışmasında, büyük veride veri kullanılabilirliği ve bilgi kaybı sorunları ile başa çıkmak için yeni bir kaos ve pertürbasyon temelli gizlilik koruma algoritması öne sürülmüştür. Veri seti türünden bağımsız, hem sayısal hem de kategorik niteliklere uygulanabilen kapsamlı bir gizlilik korumalı veri yayınlama algoritmasının geliştirilmesi, çalışmanın literatüre katkısıdır. Önerilen algoritmanın daha yüksek veri kullanılabilirliğine sahip olmasının sebebi; her bir yarı tanımlayıcı için benzersiz nitelik değerlerinin sıklığının analiz edilmesi, sıklık analizine uygun olarak kritik değerlerin belirlenmesi ve sadece belirlenen bu kritik değerler için pertürbasyon işleminin gerçekleştirilmesidir. Sistemlerin rastlantısallığı için yaygın olarak kullanılan disiplinler arası bir teori olan kaosun verilerin karıştırılmasındaki etkinliğini ortaya koymak, bu

çalışmanın diğer bir önemli katkısıdır. Bilindiği kadarıyla, literatürde büyük verinin gizliliğinin korunmasında kaosun bu çerçevede kullanımına ilişkin başka bir çalışma yoktur. Kaos rastgeleleştirmede çok başarılıdır. Bu nedenle kaosun veri pertürbasyonundaki faydası tez kapsamında araştırılmıştır.

Önerilen algoritmanın ölçeklenebilirliği ve uygulanabilirliği, farklı boyutlarda veri setleri kullanılarak değerlendirilmiştir. Önerilen algoritma kullanılarak gizliliği korunan büyük veri setleri Hadoop üzerinde dağıtıklaştırılmıştır. Kullback-Leibler uzaklığı, olasılıksal anonimlik, sınıflandırma doğruluğu, F-ölçütü, yürütme süresi ve Impala sorgu sonuçları değerlendirme metrikleri olarak kullanılmıştır. Gizlilik analizleri ve deneysel sonuçlar önerilen algoritmanın aynı deney koşullarında Kullback-Leibler uzaklığı, sınıflandırma doğruluğu ve F-ölçütü açısından önceki çalışmalardan daha iyi performans gösterdiğini kanıtlamaktadır. Önerilen algoritmanın olasılıksal anonimliği ve yürütme süresi performansı oldukça elverişlidir. Ayrıca Impala sorgu sonuçları önerilen algoritmanın büyük veri setlerinin gizliliğini etkili bir şekilde koruduğunu göstermektedir. Veri pertürbasyonu için kaotik bir fonksiyonun kullanılmasından kaynaklanan önerilen algoritmanın başarısı göz önünde bulundurulduğunda, algoritma; verilerin yayınlanması ve paylaşılmasından önce bireylerin gizliliğinin korunması için uygunluğunu garanti etmektedir.

Tez kapsamında, büyük veri setlerinin önerilen algoritma kullanılarak gizliliği korunduktan sonra Hadoop üzerinde dağıtılması sağlanmıştır. Bu çalışmanın ileriki aşamalarında, önerilen kaos ve pertürbasyona dayalı büyük veri anonimleştirme algoritmasının direkt olarak Hadoop ortamında gerçeklenmesi planlanmaktadır.

KAYNAKLAR

Aggarwal, C.C. and Yu, P.S., 2008, *Privacy-Preserving Data Mining: Models and Algorithms*, Springer Publication, Heidelberg, Berlin, Germany.

Agrawal, D. and Aggarwal, C.C., 2001, On the Design and Quantification of Privacy Preserving Data Mining Algorithms, *The Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 21-24 May 2001, Santa Barbara, California, USA, 247-255.

Agrawal, D., Bernstein, P., Bertino, E., Davidson, S., Dayal, U., Franklin, M., Gehrke, J., Haas, L., Halevy, A., Han, J., Jagadish, H.V., Labrinidis, A., Madden, S., Papakonstantinou, Y., Patel, J.M., Ramakrishnan, R., Ross, K., Shahabi, C., Suciu, D., Vaithyanathan, S. and Widom, J., 2012, *Challenges and Opportunities with Big Data*, White Paper, Computing Community Consortium, Computing Research Association, https://cra.org/ccc/resources/ccc-led-whitepapers/, [Ziyaret tarihi: 21 Nisan 2018].

Agrawal, R. and Srikant, R., 2000, Privacy-preserving data mining, *ACM SIGMOD Record*, 29 (2), 439-450.

Agrawal, S. and Haritsa, J.R., 2005, A framework for high-accuracy privacy-preserving mining, *21st International Conference on Data Engineering (ICDE 2005)*, 5-8 April 2005, Tokoyo, Japan, Japan, 193-204.

Aho, A.V., 2012, Computation and Computational Thinking, *The Computer Journal*, 55 (7), 832-835.

Aluru, S. and Simmhan, Y., 2015, A Special Issue of Journal of Parallel and Distributed Computing: Scalable Systems for Big Data Management and Analytics, *Journal of Parallel and Distributed Computing*, 73 (6), 896.

American Institute of Physics, 2010, https://www.aip.org/fyi/2010, [Ziyaret tarihi: 21 Nisan 2018].

Amiri, F., Yazdani, N., Shakery, A. and Chinaei, A.H., 2016, Hierarchical anonymization algorithms against background knowledge attack in data releasing, *Knowledge-Based Systems*, 101, 71-89.

AMPLab, 2018, *Big Data*, UC Berkeley, https://amplab.cs.berkeley.edu/, [Ziyaret tarihi: 21 Nisan 2018].

Ardagna, C.A. and Damiani, E., 2014, Business Intelligence meets Big Data: An Overview on Security and Privacy, *NSF Workshop on Big Data Security and Privacy*, 16-17 September 2014, Dallas, TX, USA, 1-6.

Azarmi, B., 2016, *Scalable Big Data Architecture: A practitioners guide to choosing relevant Big Data architecture*, Springer, Apress, Berkeley, CA, ISBN: 978-1-4842-1327-8.

Azzini, A. and Ceravolo, P., 2013, Consistent Process Mining over Big Data Triple Stores, *IEEE International Congress on Big Data (BigData Congress)*, 27 June-2 July 2013, Santa Clara, CA, USA, 54-61.

Bakshi, K., 2012, Considerations for big data: Architecture and approach, *2012 IEEE Aerospace Conference*, 3-10 March 2012, Big Sky, MT, USA, 1-7.

Bamford, J., 2012, *The NSA Is Building the Country's Biggest Spy Center (Watch What You Say)*, Wired, https://www.wired.com/2012/03/ff_nsadatacenter/all/1/, [Ziyaret tarihi: 21 Nisan 2018].

Bayardo, R.J. and Agrawal, R., 2005, Data privacy through optimal k-anonymization, *21st International Conference on Data Engineering (ICDE 2005)*, 5-8 April 2005, Tokyo, Japan, Japan, 217-228.

Berman, J.J., 2013, *Principles of Big Data: Preparing, Sharing, and Analyzing Complex Information*, Morgan Kaufmann, Waltham, MA, USA, ISBN: 978-0-12-404576-7.

Bhadani, A.K. and Jothimani, D., 2016, *Big Data: Challenges, Opportunities, and Realities*, Effective Big Data Management and Opportunities for Implementation, In: Singh, M.K. and Kumar, D.G. (Eds.), Chapter 1, IGI Global, Pennsylvania, USA, 1-24.

Bollier, D. and Firestone, C.M., 2010, *The Promise and Peril of Big Data*, The Aspen Institute, Communications and Society Program, Washington, DC, USA, ISBN: 0-89843-516-1.

Boyd, D. and Crawford, 2011, Six Provocations for Big Data, *A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society*, 21-24 September 2011, Oxford, UK.

Cassandra, 2018, *Apache Cassandra*, http://cassandra.apache.org/, [Ziyaret tarihi: 21 Nisan 2018].

Cattell, R., 2011, Scalable SQL and NoSQL data stores, *ACM Sigmod Record*, 39 (4), 12-27.

Chaudhuri, S., Dayal, U. and Narasayya, V., 2011, An Overview of Business Intelligence Technology, *Communications of the ACM*, 54 (8), 88-98.

Chen, C.P. and Zhang, C.Y., 2014, Data-intensive applications, challenges, techniques and technologies: A survey on Big Data, *Information Sciences*, 275, 314-347.

Chen, H., Chiang, R.H.L. and Storey, V.C., 2012, Business intelligence and analytics: from big data to big impact, *MIS Quarterly*, 36 (4), 1165-1188.

Chen, K. and Liu, L., 2009, Privacy-Preserving Multiparty Collaborative Mining with Geometric Data Perturbation, *IEEE Transactions on Parallel and Distributed Systems*, 20 (12), 1764-1776.

Chen, K. and Liu, L., 2011, Geometric data perturbation for privacy preserving outsourced data mining, *Knowledge and Information Systems*, 29 (3), 657-695.

Chen, K., Sun, G. and Liu, L., 2007, Towards Attack-Resilient Geometric Data Perturbation, *2007 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, 78-89.

Chen, M., Mao, S. and Liu, Y., 2014, Big Data: A survey, *Mobile Networks and Applications*, 19 (2), 171-209.

Chen, R., Fung, B. and Desai, B.C., 2011, Differentially Private Trajectory Data Publication, *arXiv*, Cornell University Library, preprint arXiv:1112.2020.

Chen, T.S., Lee, W.B., Chen, J., Kao, Y.H. and Hou, P.W., 2013, Reversible privacy preserving data mining: a combination of difference expansion and privacy preserving, *The Journal of Supercomputing*, 66 (2), 907-917.

Cloudera, 2018, *Cloudera*, https://www.cloudera.com/, [Ziyaret tarihi: 21 Nisan 2018].

Cloudera Impala, 2018, *Cloudera Impala Overview*, https://www.cloudera.com/documentation/enterprise/5-3-x/topics/impala_intro.html, [Ziyaret tarihi: 21 Nisan 2018].

Cox, M. and Ellsworth, D., 1997, Managing Big Data for Scientific Visualization, *ACM SIGGRAPH '97 Course #4, Exploring Gigabyte Datasets in Real-Time: Algorithms, Data Management, and Time-Critical Design*, 3-8 August 1997, Los Angeles, CA, USA, 146-162.

CSAIL, 2018, *Big Data*, MIT Big Data Initiative, http://bigdata.csail.mit.edu/, [Ziyaret tarihi: 21 Nisan 2018].

Cukier, K., 2010, *Data, Data Everywhere: A Special Report on Managing Information*, The Economist, Economist Newspaper, https://www.economist.com/node/15557443, [Ziyaret tarihi: 21 Nisan 2018].

Davis, K, 2012, *Ethics of Big Data: Balancing Risk and Innovation*, O'Reilly Media, ISBN: 978-1-449-31179-7.

Dean, J. and Ghemawat, S., 2008, MapReduce: Simplified Data Processing on Large Clusters, *Communications of the ACM*, 51 (1), 107-113.

Demchenko, Y., De Laat, C. and Membrey, P., 2014, Defining Architecture Components of the Big Data Ecosystem, *2014 International Conference on Collaboration Technologies and Systems (CTS)*, 19-23 May 2014, Minneapolis, MN, USA, 104-112.

Demchenko, Y., Grosso, P., De Laat, C. and Membrey, P., 2013, Addressing Big Data Issues in Scientific Data Infrastructure, *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 20-24 May 2013, San Diego, CA, USA, 48-55.

Demirkan, H. and Delen, D., 2013, Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud, *Decision Support Systems*, 55 (1), 412-421.

Diebold, F.X., 2012, *A Personal Perspective on the Origin(s) and Development of 'Big Data': The Phenomenon, the Term, and the Discipline*, Social Science Research Network, Penn Institute for Economic Research (PIER), Research Paper Series, Paper No. 13-003 https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2202843, [Ziyaret tarihi: 21 Nisan 2018].

Domingo-Ferrer, J., Mateo-Sanz, J.M. and Torra, V., 2001, Comparing SDC Methods for Microdata on the Basis of Information Loss and Disclosure, *International Conference on New Techniques and Technologies for Statistics: Exchange of Technology and Knowhow*, 807-826.

Domingo-Ferrer, J. and Muralidhar, K., 2016, New directions in anonymization: Permutation paradigm, verifiability by subjects and intruders, transparency to users, *Information Sciences*, 337-338, 11-24.

Domingo-Ferrer, J. and Soria-Comas, J., 2015, From *t*-closeness to differential privacy and vice versa in data anonymization, *Knowledge-Based Systems*, 74, 151-158.

Dong, B., Liu, R. and Wang, W.H., 2014, PraDa: Privacy-preserving Data-Deduplication-as-a-Service, *23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM'14)*, 3-7 November 2014, Shanghai, China, 1559-1568.

Douglas, K., 2012, *Infographic: Big Data Brings Marketing Big Numbers*, https://martech.zone/ibm-big-data-marketing/, [Ziyaret tarihi: 21 Nisan 2018].

Dwork, C., 2006, Differential privacy, *33rd international conference on Automata, Languages and Programming (ICALP'06)*, 10-14 July 2006, Venice, Italy, 1-12.

Dwork, C., McSherry, F., Nissim, K. and Smith, A., 2006, Calibrating noise to sensitivity in private data analysis, *Third Theory of Cryptography Conference (TCC 2006)*, 4-7 March 2006, New York, NY, USA, 265-284.

Elmisery, A.M. and Fu, H., 2010, Privacy Preserving Distributed Learning Clustering of HealthCare Data Using Cryptography Protocols, *2010 IEEE 34th Annual Computer Software and Applications Conference Workshops (COMPSACW)*, 19-23 July 2010, Seoul, South Korea, 140-145.

Emani, C.K., Cullot, N. and Nicolle, C., 2015, Understandable Big Data: A Survey, *Computer Science Review*, 17, 70-81.

Evfimievski, A., Gehrke, J. and Srikant, R., 2003, Limiting privacy breaches in privacy preserving data mining, *Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 9-11 June 2003, New York, NY, USA, 211-222.

Evfimievski, A., Srikant, R., Agrawal, R. and Gehrke, J., 2004, Privacy preserving mining of association rules, *Information Systems*, 29 (4), 343-364.

Eyüpoğlu, C., Aydın, M.A., Sertbaş, A., Zaim, A.H. and Öneş, O., 2017, Preserving Individual Privacy in Big Data, *International Journal of Informatics Technologies*, 10 (2), 177-184.

Eyüpoğlu, C., Aydın, M.A., Zaim, A.H. and Sertbaş, A., 2018, An Efficient Big Data Anonymization Algorithm Based on Chaos and Perturbation Techniques, *Entropy*, 20 (5), 373, 1-18.

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S. and Bouras, A., 2014, A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis, *IEEE Transactions on Emerging Topics in Computing*, 2 (3), 267-279.

Fahad, A., Tari, Z., Almalawi, A., Goscinski, A., Khalil, I. and Mahmood, A., 2014, "PPFSCADA: Privacy preserving framework for SCADA data publishing, *Future Generation Computer Systems*, 37, 496-511.

Fang, H., Zhang, Z., Wang, C.J., Daneshmand, M., Wang, C. and Wang, H., 2015, A Survey of Big Data Research, *IEEE Network*, 29 (5), 6-9.

Fang, W. and Yang, B., 2008, Privacy Preserving Decision Tree Learning over Vertically Partitioned Data, *2008 International Conference on Computer Science and Software Engineering*, 12-14 December 2008, Hubei, China, 1049-1052.

Feldman, D., Schmidt, M. and Sohler, C., 2013, Turning Big data into tiny data: Constant-size coresets for *k*-means, PCA and projective clustering, *Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, 6-8 January 2013, Philadelphia, Pennsylvania, USA, 1434-1453.

Flume, 2018, *Apache Flume*, https://flume.apache.org/, [Ziyaret tarihi: 21 Nisan 2018].

Fouad, M.R., Elbassioni, K. and Bertino, E., 2014, A Supermodularity-Based Differential Privacy Preserving Algorithm for Data Anonymization, *IEEE Transactions on Knowledge and Data Engineering*, 26 (7), 1591-1601.

Fox, B., 2011, *Leveraging Big Data for Big Impact*, Health Management Technology, https://www.healthmgttech.com/, [Ziyaret tarihi: 21 Nisan 2018].

Freund, Y. and Schapire, R.E., 1999, Large Margin Classification Using the Perceptron Algorithm, *Machine learning*, 37 (3), 277-296.

Fung, B.C.M., Wang, K., Chen, R. and Yu, P.S., 2010, Privacy preserving data publishing: A survey on recent developments, *ACM Computing Surveys*, 42 (4), 14, 1-53.

Gandomi, A. and Haider, M., 2015, Beyond the hype: Big data concepts, methods, and analytics, *International Journal of Information Management*, 35 (2), 137-144.

Gantz, J. and Reinsel, D., 2013, *The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East–United States*, IDC Country Brief, IDC Analyze the Future, Framingham, MA, USA, https://www.emc.com/collateral/analyst-reports/idc-digital-universe-united-states.pdf, [Ziyaret tarihi: 21 Nisan 2018].

Gartner IT Glossary, 2018, *Big data*, https://www.gartner.com/it-glossary/big-data/, [Ziyaret tarihi: 21 Nisan 2018].

Gazeau, I., Miller, D. and Palamidessi, C., 2016, Preserving differential privacy under finite-precision semantics, *Theoretical Computer Science*, 655, 92-108.

Ghemawat, S., Gobioff, H. and Leung, S.T., 2003, The Google file system, *The Nineteenth ACM Symposium on Operating Systems Principles (SOSP'03)*, 19-22 October 2003, Bolton Landing, NY, USA, 29-43.

Ghinita, G., Kalnis, P. and Tao, Y., 2011, Anonymous Publication of Sensitive Transactional Data, *IEEE Transactions on Knowledge and Data Engineering*, 23 (2), 161-174.

Gionis, A. and Tassa, T., 2009, k-Anonymization with Minimal Loss of Information, *IEEE Transactions on Knowledge and Data Engineering*, 21 (2), 206-219.

Gkoulalas-Divanis, A., Loukides, G. and Sun, J., 2014, Toward smarter healthcare: Anonymizing medical data to support research studies, *IBM Journal of Research and Development*, 58 (1), 9:1-9:11.

Goryczka, S., Xiong, L. and Fung, B.C., 2014, *m*-Privacy for Collaborative Data Publishing, *IEEE Transactions on Knowledge and Data Engineering*, 26 (10), 2520-2533.

Gu, R., Yang, X., Yan, J., Sun, Y., Wang, B., Yuan, C. and Huang, Y., 2014, SHadoop: Improving MapReduce performance by optimizing job execution mechanism in Hadoop clusters, *Journal of Parallel and Distributed Computing*, 74 (3), 2166-2179.

Hadoop, 2018, *What Is Apache Hadoop?*, http://hadoop.apache.org/, [Ziyaret tarihi: 21 Nisan 2018].

Han, J., Kamber, M. and Pei, J., 2012, *Data Mining Concepts and Techniques*, 3rd ed., Morgan Kaufmann, San Francisco, CA, USA, ISBN: 978-0-12-381479-1.

Han, J., Pei, J. and Yin, Y., 2000, Mining frequent patterns without candidate generation, *ACM SIGMOD Record*, 29 (2), 1-12.

Han, J., Yu, J., Mo, Y., Lu, J. and Liu, H., 2014, MAGE: A semantics retaining *K*-anonymization method for mixed data, *Knowledge-Based Systems*, 55, 75-86.

Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A. and Khan, S.U., 2015, The rise of "big data" on cloud computing: Review and open research issues, *Information Systems*, 47, 98-115.

Helmbold, D.P. and Warmuth, M.K., 1995, On Weak Learning, *Journal of Computer and System Sciences*, 50 (3), 551-573.

Henze, M., Hermerschmidt, L., Kerpen, D., Häußling, R., Rumpe, B. and Wehrle, K., 2016, A comprehensive approach to privacy in the cloud-based Internet of things, *Future Generation Computer Systems*, 56, 701-718.

Herranz, J., Matwin, S., Nin, J. and Torra, V., 2010, Classifying data from protected statistical datasets, *Computers & Security*, 29 (8), 875-890.

Hillard, R., 2012, *It's time for a new definition of big data*, http://mike2.openmethodology.org/blogs/information-development/2012/03/18/its-time-for-a-new-definition-of-big-data/, [Ziyaret tarihi: 21 Nisan 2018].

Hipgrave, S., 2013, Smarter fraud investigations with big data analytics, *Network Security*, 2013 (12), 7-9.

Holzinger, A., Stocker, C., Ofner, B., Prohaska, G., Brabenetz, A. and Hofmann-Wellenhof, R., 2013, *Combining HCI, Natural Language Processing, and Knowledge Discovery-Potential of IBM Content Analytics as an Assistive Technology in the Biomedical Field*, Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data, In Holzinger, A. and Pasi, G. (Eds.), Springer, Berlin, Heidelberg, 13-24.

Howard, J.H., Kazar, M.L., Menees, S.G., Nichols, D.A., Satyanarayanan, M., Sidebotham, R. N. and West, M.J., 1988, Scale and performance in a distributed file system, *ACM Transactions on Computer Systems*, 6 (1), 51-81.

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.

IDC, 2018, *Analyze the future*, http://www.idc.com/, [Ziyaret tarihi: 21 Nisan 2018].

Islam, M.Z. and Brankovic, L., 2011, Privacy preserving data mining: A noise addition framework using a novel clustering technique, *Knowledge-Based Systems*, 24 (8), 1214-1223.

Jain, A.K., Murty, M.N. and Flynn, P.J., 1999, Data Clustering: A Review, *ACM Computing Surveys*, 31 (3), 264-323.

Ji, C., Li, Y., Qiu, W., Awada, U. and Li, K., 2012, Big Data Processing in Cloud Computing Environments, *2012 12th International Symposium on Pervasive Systems, Algorithms and Networks (ISPAN)*, 13-15 December 2012, San Marcos, TX, USA, 17-23.

Kafka, 2018, *Apache Kafka*, https://hortonworks.com/apache/kafka/, [Ziyaret tarihi: 21 Nisan 2018].

Kaisler, S., Armour, F., Espinosa, J.A. and Money, W., 2013, Big Data: Issues and Challenges Moving Forward, *2013 46th Hawaii International Conference on System Sciences (HICSS)*, 7-10 January 2013, Wailea, Maui, HI, USA, 995-1004.

Kang, U., 2011, *Mining Tera-Scale Graphs with MapReduce: Theory, Engineering and Discoveries*, Thesis (PhD), University of Chicago.

Kantarcioglu, M. and Clifton, C., 2004, Privacy-preserving distributed mining of association rules on horizontally partitioned data, *IEEE Transactions on Knowledge and Data Engineering*, 16 (9), 1026-1037.

Kantarcıoglu, M., Vaidya, J. and Clifton, C., 2003, Privacy Preserving Naive Bayes Classifier for Horizontally Partitioned Data, *IEEE ICDM Workshop on Privacy Preserving Data Mining*, 3-9.

Katal, A., Wazid, M. and Goudar, R.H., 2013, Big data: Issues, challenges, tools and good practices, *2013 Sixth International Conference on Contemporary Computing (IC3)*, 8-10 August 2013, Noida, India, 404-409.

Khan, N., Yaqoob, I., Hashem, I.A.T., Inayat, Z., Ali, W.K.M., Alam, M., Shiraz, M. and Gani, A., 2014, Big Data: Survey, Technologies, Opportunities, and Challenges, *The Scientific World Journal*, 2014, 1-18.

Kim, J. and Winkler, W., 2003, *Multiplicative Noise for Masking Continuous Data*, Research Report Series, Statistical Research Division, U.S. Bureau of the Census, Washington, DC, USA.

Kisilevich, S., Rokach, L., Elovici, Y. and Shapira, B., 2010, Efficient Multidimensional Suppression for K-Anonymity, *IEEE Transactions on Knowledge and Data Engineering*, 22 (3), 334-347.

Kohavi, R. and Becker, B., 1996, *Adult Data Set*, Data Mining and Visualization Silicon Graphics, https://archive.ics.uci.edu/ml/datasets/adult, [Ziyaret tarihi: 21 Nisan 2018].

Kohlmayer, F., Prasser, F., Eckert, C. and Kuhn, K.A., 2014, A flexible approach to distributed data anonymization, *Journal of Biomedical Informatics*, 50, 62-76.

Komishani, E.G., Abadi, M. and Deldar, F., 2016, PPTD: Preserving personalized privacy in trajectory data publishing by sensitive attribute generalization and trajectory local suppression, *Knowledge-Based Systems*, 94, 43-59.

Koufogiannis, F. and Pappas, G.J., 2017, Differential privacy for dynamical sensitive data, *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 12-15 December 2017, Melbourne, VIC, Australia, 1118-1125.

Krishnan, K., 2013, *Data Warehousing in the Age of Big Data*, Morgan Kaufmann, Waltham, MA, USA, ISBN: 978-0-12-405891-0.

Kullback, S. and Leibler, R.A., 1951, On Information and Sufficiency, *The Annals of Mathematical Statistics*, 22 (1), 79-86.

Kwon, O., Lee, N. and Shin, B., 2014, Data quality management, data usage experience and acquisition intention of big data analytics, *International Journal of Information Management*, 34 (3), 387-394.

Labrinidis, A. and Jagadish, H.V., 2012, Challenges and Opportunities with Big Data, *VLDB Endowment*, 5 (12), 2032-2033.

Lafuente, G., 2015, The big data security challenge, *Network Security*, 2015 (1), 12-14.

Laney, D., 2001, *3D Data Management: Controlling Data Volume, Velocity, and Variety*, Application Delivery Strategies by META Group Inc., https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf, [Ziyaret tarihi: 21 Nisan 2018].

Laurila, J.K., Gatica-Perez, D., Aad, I., Blom, J., Bornet, O., Do, T.M.T., Dousse, O., Eberle, J. and Miettinen, M., 2012, The Mobile Data Challenge: Big Data for Mobile Computing Research, *10th International Conferenceon Pervasive Computing*, 18-22 June 2012, Newcastle, UK, 1-8.

Lee, S., Park, H. and Shin, Y., 2012, Cloud Computing Availability: Multi-clouds for Big Data Service, *6th International Conference on Convergence and Hybrid Information Technology (ICHIT 2012)*, 23-25 August 2012, Daejeon, Korea, 799-806.

LeFevre, K., DeWitt, D.J. and Ramakrishnan, R., 2005, Incognito: Efficient full-domain k-anonymity, *2005 ACM SIGMOD International Conference on Management of Data*, 14-16 June 2005, Baltimore, Maryland, ABD, 49-60.

LeFevre, K., DeWitt, D.J. and Ramakrishnan, R., 2006, Mondrian multidimensional k-anonymity, *22nd International Conference on Data Engineering (ICDE'06)*, 3-7 April 2006, Atlanta, GA, USA, 25.

Leung, C.K.S., MacKinnon, R.K. and Jiang, F., 2014, Reducing the Search Space for Big Data Mining for Interesting Patterns from Uncertain Data, *2014 IEEE International Congress on Big Data (BigData Congress)*, 27 June-2 July 2014, Anchorage, AK, USA, 315-322.

Li, M., Zhu, L., Zhang, Z. and Xu, R., 2017, Achieving differential privacy of trajectory data publishing in participatory sensing, *Information Sciences*, 400-401, 1-13.

Li, N., Li, T. and Venkatasubramanian, S., 2007, *t*-Closeness: Privacy Beyond *k*-Anonymity and *l*-Diversity, *IEEE 23rd International Conference on Data Engineering (ICDE 2007)*, 15-20 April 2007, Istanbul, Turkey, 106-115.

Li, N., Qardaji, W. and Su, D., 2012, On sampling, anonymization, and differential privacy or, *k*-anonymization meets differential privacy, *7th ACM Symposium on Information, Computer and Communications Security (ASIACCS'12)*, 2-4 May 2012, Seoul, Korea, 32-33.

Li-Xin, L., Yong-Shan, D. and Jia-Yan, W., 2017, Differential Privacy Data Protection Method Based on Clustering, *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 12-14 October 2017, Nanjing, China, 11-16.

Lichman, M., 2013, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, Irvine, CA, USA, https://archive.ics.uci.edu/ml/index.php, [Ziyaret tarihi: 21 Nisan 2018].

Lin, C.Y., 2016, A reversible data transform algorithm using integer transform for privacy-preserving data mining, *Journal of Systems and Software*, 117, 104-112.

Lin, M.Y., Lee, P.Y. and Hsueh, S.C., 2012, Apriori-based frequent itemset mining algorithms on MapReduce, *6th International Conference on Ubiquitous Information Management and Communication (ICUIMC'12)*, 20-22 February 2012, Kuala Lumpur, Malaysia, 76.

Lin, W.Y., Yang, D.C. and Wang, J.T., 2016, Privacy preserving data anonymization of spontaneous ADE reporting system dataset, *BMC Medical Informatics and Decision Making*, 16 (1), 58.

Lindell, Y. and Pinkas, B., 2000, Privacy Preserving Data Mining, *20th Annual International Cryptology Conference*, 20-24 August, Santa Barbara, California, USA, 36-54.

Lindell, Y. and Pinkas, B., 2009, Secure Multiparty Computation for Privacy Preserving Data Mining, *Journal of Privacy and Confidentiality*, 1, 59-98.

Liu, K., Kargupta, H. and Ryan, J., 2006, Random projection-based multiplicative data perturbation for privacy preserving distributed data mining, *IEEE Transactions on Knowledge and Data Engineering*, 18 (1), 92-106.

Liu, J., Huang, X. and Liu, J.K., 2015, Secure sharing of Personal Health Records in cloud computing: Ciphertext-Policy Attribute-Based Signcryption, *Future Generation Computer Systems*, 52, 67-76.

Machanavajjhala, A., Gehrke, J., Kifer, D. and Venkitasubramaniam, M., 2007, *l*-diversity: Privacy Beyond *k*-Anonymity, *ACM Transactions on Knowledge Discovery from Data*, 1 (1), 3.

Manoochehri, M., 2013, *Data Just Right: Introduction to Large-Scale Data & Analytics*, Addison-Wesley Data & Analytics Series, Addison-Wesley, Crawfordsville, Indiana, USA, ISBN: 978-0-321-89865-4.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. and Byers, A.H., 2011, *Big Data: The Next Frontier for Innovation, Competition, and Productivity*, McKinsey Global Institute, McKinsey&Company, New York, NY, USA, https://bigdatawg.nist.gov/pdf/MGI_big_data_full_report.pdf, [Ziyaret tarihi: 21 Nisan 2018].

Matturdi, B., Xianwei, Z., Shuai, L. and Fuhong, L., 2014, Big Data Security and Privacy: A Review, *China Communications*, 11 (4), 135-145.

McCune, J.C., 1998, Data, data everywhere, *Management Review*, 87 (10), 10-12.

McSherry, F. and Talwar, K, 2007, Mechanism Design via Differential Privacy, *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, 21-23 October 2007, Providence, RI, USA, 94-103.

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

Menon, R., 2014, *Cloudera Administration Handbook*, Packt Publishing, Birmingham, UK, ISBN: 978-1-78355-896-4.

Miller, H.E., 2013, Big Data in Cloud Computing: A Taxonomy of Risks, *Information Research*, 18 (1), 1-19.

Mitchell, T.M., 1997, *Machine Learning*, McGraw-Hill Science/Engineering/Math, ISBN: 0-07042-807-7.

Mohammed, N., Chen, R., Fung, B. and Yu, P.S., 2011, Differentially private data release for data mining, *17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 21-24 August 2011, San Diego, California, USA, 493-501.

Mortazavi, R. and Jalili, S., 2014, Fast data-oriented microaggregation algorithm for large numerical datasets, *Knowledge-Based Systems*, 67, 195-205.

Muda, Z., Yassin, W., Sulaiman, M.N. and Udzir, N.I., 2011, Intrusion detection based on k-means clustering and OneR classification, *2011 7th International Conference on Information Assurance and Security (IAS)*, 5-8 December 2011, Melaka, Malaysia, 192-197.

Mysore, D., Khupat, S. and Jain, S., 2013, *Big data architecture and patterns, Part 1: Introductionto big data classification and architecture, How to classify big data into categories*, IBM, developerWorks, https://www.ibm.com/developerworks/analytics/library/bd-archpatterns1/index.html, [Ziyaret tarihi: 21 Nisan 2018].

Nayahi, J.J.V. and Kavitha, V., 2015, An Efficient Clustering for Anonymizing Data and Protecting Sensitive Label, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 23 (5), 685-714.

Nayahi, J.J.V. and Kavitha, V., 2017, Privacy and utility preserving data clustering for data anonymization and distribution on Hadoop, *Future Generation Computer Systems*, 74, 393-408.

O'Driscoll, A., Daugelaite, J. and Sleator, R.D., 2013, 'Big data', Hadoop and cloud computing in genomics, *Journal of Biomedical Informatics*, 46 (5), 774-781.

O'Leary, D.E., 2013, Artificial intelligence and big data. *IEEE Intelligent Systems*, 28 (2), 96-99.

Oliveira, S.R. and Zaiane, O.R., 2004, Privacy-Preserving Clustering by Object Similarity-Based Representation and Dimensionality Reduction Transformation, *Workshop on Privacy and Security Aspects of Data Mining (PSADM'04) in Conjunction with the Fourth IEEE International Conference on Data Mining (ICDM'04)*, 01-04 November 2004, Brighton, UK, 21-30.

Oliveira, S.R. and Zaiane, O.R., 2010, Privacy Preserving Clustering by Data Transformation, *Journal of Information and Data Management*, 1 (1), 37-51.

Olson, D.L. and Delen, D, 2008, *Advanced Data Mining Techniques*, Springer Science & Business Media, Heidelberg, Berlin, Germany, ISBN: 978-3-540-76916-3.

Oussous, A., Benjelloun, F.Z., Lahcen, A.A. and Belfkih, S., 2017, Big Data technologies: A survey, *Journal of King Saud University-Computer and Information Sciences*, In Press, Corrected Proof.

Pandey, S. and Nepal, S., 2013, Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond, *Future Generation Computer Systems*, 29 (7), 1774-1776.

Pastorelli, M., Barbuzzi, A., Carra, D., Dell'Amico, M. and Michiardi, P., 2013, HFSP: Size-based scheduling for Hadoop, *2013 IEEE International Conference on Big Data*, 6-9 October 2013, Silicon Valley, CA, USA, 51-59.

Pinkas, B., 2002, Cryptographic techniques for privacy-preserving data mining, *ACM SIGKDD Explorations Newsletter*, 4 (2), 12-19.

Prasad, B.R. and Agarwal, S., 2016, Comparative Study of Big Data Computing and Storage Tools: A Review, *International Journal of Database Theory and Application*, 9 (1), 45-66.

Proffitt, B., 2012, *Big data tools and vendors, Who's who in big data: How the tools and vendors break down*, ITworld, https://www.itworld.com/article/2729487/big-data/big-data-tools-and-vendors.html, [Ziyaret tarihi: 21 Nisan 2018].

Quinlan, J.R., 1990, Decision trees and decision-making, *IEEE Transactions on Systems, Man, and Cybernetics*, 20 (2), 339-346.

Quinlan, J.R., 1993, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA, USA, ISBN: 1-55860-238-0.

Quinlan, J.R., 1986, Induction of Decision Trees, *Machine Learning*, 1 (1), 81-106.

Rebentrost, P., Mohseni, M. and Lloyd, S., 2014, Quantum Support Vector Machine for Big Data Classification, *Physical Review Letters*, 113 (13-26), 130503.

Reeve, A., 2013, *Managing Data in Motion: Data Integration Best Practice Techniques and Technologies*, Morgan Kaufmann, Waltham, MA, USA, ISBN: 978-0-12-397167-8.

Riondato, M., DeBrabant, J.A., Fonseca, R. and Upfal, E., 2012, PARMA: a parallel randomized algorithm for approximate association rules mining in MapReduce, *21st ACM International Conference on Information and Knowledge Management*, 29 October-2 November 2012, Maui, Hawaii, USA,85-94.

Rizvi, S.J. and Haritsa, J.R., 2002, Maintaining data privacy in association rule mining, *28th International Conference on Very Large Databases (VLDB'02)*, 20-23 August 2002, Hong Kong, China, 682-693.

Rosenblatt, F., 1958, The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, 65 (6), 386.

Rosenblatt, F., 1962, *Principles of Neurodynamics*, Spartan Books, Washington, DC, USA.

Sagiroglu, S. and Sinanc, D., 2013, Big data: A review, *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 20-24 May 2013, San Diego, CA, USA, 42-47.

Sakr, S., 2016, *Big Data 2.0 Processing Systems: A Survey*, Springer, Cham, ISBN: 978-3-319-38775-8

Samarati, P., 2001, Protecting respondents identities in microdata release, *IEEE Transactions on Knowledge and Data Engineering*, 13 (6), 1010-1027.

Samarati, P. and Sweeney, L., 1998, *Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression*, Technical Report, SRI International, 101-132.

SAP, 2012, *Small and Midsize Companies Look to Make Big Gains With "Big Data," According to Recent Poll Conducted on Behalf of SAP*, http://global.sap.com/news-reader/index.epx?PressID=19188, [Ziyaret tarihi: 21 Nisan 2018].

SAS, 2018, *Big Data History and Current Considerations*, https://www.sas.com/en_us/insights/big-data/what-is-big-data.html, [Ziyaret tarihi: 21 Nisan 2018].

Sattar, A.S., Li, J., Ding, X., Liu, J. and Vincent, M., 2013, A general framework for privacy preserving data publishing, *Knowledge-Based Systems*, 54, 276-287.

SAVVYCOM, 2018, https://savvycomsoftware.com/, [Ziyaret tarihi: 21 Nisan 2018].

Schadt, E.E., Linderman, M.D., Sorenson, J., Lee, L. and Nolan, G.P., 2011, Cloud and heterogeneous computing solutions exist today for the emerging big data problems in biology, *Nature Reviews Genetics*, 12, 224.

Schnase, J.L., Duffy, D.Q., Tamkin, G.S., Nadeau, D., Thompson, J.H., Grieg, C.M., McInerney, M.A. and Webster, W.P., 2017, MERRA Analytic Services: Meeting the Big Data challenges of climate science through cloud-enabled Climate Analytics-as-a-Service, *Computers, Environment and Urban Systems*, 61, 198-211.

Shafer, J., Rixner, S. and Cox, A.L., 2010, The hadoop distributed filesystem: Balancing portability and performance, *2010 IEEE International Symposium on Performance Analysis of Systems & Software (ISPASS)*, 28-30 March 2010, White Plains, NY, USA, 122-133.

Sharma, S. and Mangat, V., 2015, Technology and Trends to Handle Big Data: Survey, *2015 Fifth International Conference on Advanced Computing & Communication Technologies (ACCT)*, 21-22 February 2015, Haryana, India, 266-271.

Shen, P. and Li, C., 2014, Distributed Information Theoretic Clustering, IEEE Transactions on Signal Processing, 62 (13), 3442-3453.

Shirkhorshidi, A.S., Aghabozorgi, S., Wah, T.Y. and Herawan, T., 2014, Big Data Clustering: A Review, *14th International Conference on Computational Science and Its Applications (ICCSA 2014)*, 30 June-3 July 2014, Guimarães, Portugal, 707-720.

Shvachko, K., Kuang, H., Radia, S. and Chansler, R., 2010, The Hadoop Distributed File System, *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 3-7 May 2010, Incline Village, NV, USA, 1-10.

Sicari, S., Rizzardi, A., Grieco, L.A. and Coen-Porisini, A., 2015, Security, privacy and trust in Internet of Things: The road ahead, *Computer Networks*, 76, 146-164.

Singh, K., Guntuku, S.C., Thakur, A. and Hota, C., 2014, Big Data Analytics framework for Peer-to-Peer Botnet Detection using Random Forests, *Information Sciences*, 278, 488-497.

Singh, D. and Reddy, C.K., 2015, A survey on platforms for big data analytics, *Journal of Big Data*, 2 (1), 8.

Song, J., Guo, C., Wang, Z., Zhang, Y., Yu, G. and Pierson, J.M., 2015, HaoLap: A Hadoop based OLAP system for big data, *Journal of Systems and Software*, 102, 167-181.

Soria-Comas, J., Domingo-Ferrer, J., Sánchez, D. and Martínez, S., 2014, Enhancing data utility in differential privacy via microaggregation-based *k*-anonymity, *The VLDB Journal*, 23 (5), 771-794.

Soria-Comas, J., Domingo-Ferrer, J., Sánchez, D. and Martinez, S., 2015, t-Closeness through Microaggregation: Strict Privacy with Enhanced Utility Preservation, *IEEE Transactions on Knowledge and Data Engineering*, 27 (11), 3098-3110.

Sqoop, 2018, *Apache Sqoop*, http://sqoop.apache.org/, [Ziyaret tarihi: 21 Nisan 2018].

Srirama, S.N., Jakovits, P. and Vainikko, E., 2012, Adapting scientific computing problems to clouds using MapReduce, *Future Generation Computer Systems*, 28 (1), 184-192.

Statistic Brain Research Institute, 2018, https://www.statisticbrain.com/, [Ziyaret tarihi: 21 Nisan 2018].

Sun, X., Li, M. and Wang, H., 2011, A family of enhanced (L, α)-diversity models for privacy preserving data publishing, *Future Generation Computer Systems*, 27 (3), 348-356.

Sun, X., Wang, H., Li, J. and Truta, T.M., 2008, Enhanced *p*-sensitive *k*-anonymity models for privacy preserving data publishing, *Transactions on Data Privacy*, 1 (2), 53-66.

Sweeney, L., 1997, Guaranteeing anonymity when sharing medical data, the Datafly System, *AMIA Annual Fall Symposium*, American Medical Informatics Association, 51-55.

Sweeney, L., 1998, *Datafly: A system for providing anonymity in medical data*, Database Security XI, IFIP Advances in Information and Communication Technology, In: Lin, T.Y. and Qian, S. (Eds.), Springer, Boston, MA, 356-381.

Sweeney, L., 2002a, Achieving *k*-anonymity privacy protection using generalization and suppression, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10 (5), 571-588.

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

Talia, D., 2013, Clouds for Scalable Big Data Analytics, *Computer*, 46 (5), 98-101.

Tankard, C., 2012, Big data security, *Network Security*, 2012 (7), 5-8.

Tannahill, B.K. and Jamshidi, M., 2014, System of Systems and Big Data analytics–Bridging the gap, *Computers & Electrical Engineering*, 40 (1), 2-15.

TechAmerica Foundation's Federal Big Data Commission, 2012, *Demystifying Big Data: A Practical Guide to Transforming the Business of Government*, http://www.techamerica.org/Docs/fileManager.cfm?f=techamerica-bigdatareport-final.pdf, [Ziyaret tarihi: 21 Nisan 2018].

Tekin, C. and Van Der Schaar, M., 2013, Distributed online big data classification using context information, *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2-4 October 2013, Monticello, IL, USA, 1435-1442.

Terzi, D.S., Terzi, R. and Sagiroglu, S., 2015, A survey on Security and Privacy Issues in Big Data, *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, 14-16 December 2015, London, UK, 202-207.

Tian, H. and Zhang, W., 2011, Extending *l*-diversity to generalize sensitive data, *Data & Knowledge Engineering*, 70 (1), 101-126.

Truta, T.M., Campan, A. and Meyer, P., 2007, Generating Microdata with *P*-Sensitive *K*-Anonymity Property, *4th VLDB Workshop on Secure Data Management*, 23-24 September 2007, Vienna, Austria, 124-141.

Vaidya, J., 2004, *Privacy Preserving Data Mining over Vertically Partitioned Data*, Thesis (PhD), Purde University.

Vaidya, J. and Clifton, C., 2002, Privacy preserving association rule mining in vertically partitioned data, *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 23-26 July 2002, Edmonton, Alberta, Canada, 639-644.

Vaidya, J. and Clifton, C., 2003, Privacy-preserving *k*-means clustering over vertically partitioned data, *The ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 24-27 August 2003, Washington, DC, USA, 206-215.

Vaidya, J. and Clifton, C., 2009, Privacy-Preserving Kth Element Score over Vertically Partitioned Data, *IEEE Transactions on Knowledge and Data Engineering*, 21 (2), 253-258.

Vaidya, J., Kantarcıoğlu, M. and Clifton, C., 2008, Privacy-preserving Naive bayes classification, *The VLDB Journal*, 17 (4), 879-898.

Villars, R.L., Olofson, C.W. and Eastwood, M., 2011, *Big Data: What It Is and Why You Should Care*, White Paper, IDC Analyze the Future, MA, USA.

Waguih, H., 2013, A Data Mining Approach for the Detection of Denial of Service Attack, *IAES International Journal of Artificial Intelligence*, 2 (2), 99-106.

Wang, D., 2011, An Efficient Cloud Storage Model for Heterogeneous Cloud Infrastructures, *Procedia Engineering*, 23, 510-515.

Wang, X. and Jin, Z., 2016, A differential privacy multidimensional data release model, *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 14-17 October 2016, Chengdu, China, 171-174.

Weiping, G., Wei, W., Haofeng, Z. and Baile, S., 2006, Privacy Preserving Classification Mining, *Journal of Computer Research and Development*, 43 (1), 39-45.

White, T., 2009, *Hadoop: The Definitive Guide*, O'Reilly Media, Sebastapol, CA, ISBN: 978-1-449-31152-0.

Wimmer, H., Yoon, V.Y. and Sugumaran, V., 2016, A multi-agent system to support evidence based medicine and clinical decision making via data sharing and data privacy, *Decision Support Systems*, 88, 51-66.

Witten, I.H. and Frank, E., 2005, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, San Francisco, California, USA.

Wong, R.C.W. and Fu, A.W.C., 2010, Privacy-Preserving Data Publishing: An Overview, *Synthesis Lectures on Data Management*, 2 (1), 1-138.

Wong, R.C.W., Li, J., Fu, A.W.C. and Wang, K., 2006, (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing, *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 20-23 August 2006, Philadelphia, PA, USA, 754-759.

Xiao, M.J., Han, K., Huang, L.S. and Li, J.Y., 2006, Privacy Preserving C4.5 Algorithm Over Horizontally Partitioned Data, *Fifth International Conference on Grid and Cooperative Computing (GCC 2006)*, 21-23 October 2006, Hunan, China, 78-85.

Xiao, Y., Xiong, L. and Yuan, C., 2010, Differentially Private Data Release through Multidimensional Partitioning, *7th VLDB Workshop on Secure Data Management*, 17 September 2010, Singapore, 150-168.

Xiao, Z. and Xiao, Y., 2013, Security and Privacy in Cloud Computing, *IEEE Communications Surveys & Tutorials*, 15 (2), 843-859.

Xiao-Dan, W., Dian-Min, Y., Feng-Li, L. and Chao-Hsien, C., 2007, Distributed Model Based Sampling Technique for Privacy Preserving Clustering, *International Conference on Management Science and Engineering (ICMSE 2007)*, 20-22 August 2007, Harbin, China, 192-197.

Xu, H., Li, Z., Guo, S. and Chen, K., 2012, Cloudvista: interactive and economical visual cluster analysis for big data in the cloud, *Proceedings of the VLDB Endowment*, 5 (12), 1886-1889.

Xu, J., Wang, W., Pei, J., Wang, X., Shi, B. and Fu, A.W.C., 2006, Utility-Based Anonymization for Privacy Preservation with Less Information Loss, *ACM SIGKDD Explorations Newsletter*, 8 (2), 21-30.

Xu, L., Jiang, C., Chen, Y., Ren, Y. and Liu, K.R., 2015, Privacy or Utility in Data Collection? A Contract Theoretic Approach, *IEEE Journal of Selected Topics in Signal Processing*, 9 (7), 1256-1269.

Xu, L., Jiang, C., Wang, J., Yuan, J. and Ren, A.Y., 2014, Information Security in Big Data: Privacy and Data Mining, *IEEE Access*, 2, 1149-1176.

Xu, R. and Wunsch, D., 2009, *Clustering*, John Wiley & Sons, Hoboken, NJ, USA, ISBN: 978-0-470-27680-8.

Yan, W., Brahmakshatriya, U., Xue, Y., Gilder, M. and Wise, B., 2013, p-PIC: Parallel power iteration clustering for big data, *Journal of Parallel and Distributed computing*, 73 (3), 352-359.

Yang, J.J., Li, J.Q. and Niu, Y., 2015, A hybrid solution for privacy preserving medical data sharing in the cloud environment, *Future Generation Computer Systems*, 43-44, 74-86.

Yang, W. and Qiao, S., 2010, A novel anonymization algorithm: Privacy protection and knowledge preservation, *Expert Systems with Applications*, 37 (1), 756-766.

Yang, Y., Zhang, Z., Miklau, G., Winslett, M. and Xiao, X., 2012, Differential privacy in data publication and analysis, *2012 ACM SIGMOD International Conference on Management of Data*, 20-24 May 2012, Scottsdale, Arizona, USA, 601-606.

Yang, Z., Zhong, S. and Wright, R.N., 2005, Privacy-Preserving Classification of Customer Data without Loss of Accuracy, *2005 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, 92-102.

Yao, X., Chen, Z. and Tian, Y., 2015, A lightweight attribute-based encryption scheme for the Internet of Things, *Future Generation Computer Systems*, 49, 104-112.

Yavuz, E., Yazıcı, R., Kasapbaşı, M.C. and Yamaç, E., 2016, A chaos-based image encryption algorithm with simple logical functions, *Computers & Electrical Engineering*, 54, 471-483.

Yu, H., Vaidya, J. and Jiang, X., 2006, Privacy-Preserving SVM Classification on Vertically Partitioned Data, *10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2006)*, 9-12 April 2006, Singapore, 647-656.

Yuksel, B., Kupcu, A. and Ozkasap, O., 2017, Research issues for privacy and security of electronic health services, *Future Generation Computer Systems*, 68, 1-13.

Zaman, A.N.K., Obimbo, C. and Dara, R.A., 2017, An improved differential privacy algorithm to protect re-identification of data, *2017 IEEE Canada International Humanitarian Technology Conference (IHTC)*, 21-22 July 2017, Toronto, ON, Canada, 133-138.

Zhang, L., Wu, C., Li, Z., Guo, C., Chen, M. and Lau, F.C., 2013, Moving Big Data to The Cloud: An Online Cost-Minimizing Approach, *IEEE Journal on Selected Areas in Communications*, 31 (12), 2710-2721.

Zhang, X. and Bi, H., 2010, Research on Privacy Preserving Classification Data Mining Based on Random Perturbation, *2010 International Conference on Information Networking and Automation (ICINA)*, 18-19 October 2010, Kunming, China, 173-178.

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.

Zhang, X., Qi, L., Dou, W., He, Q., Leckie, C., Ramamohanarao, K. and Salcic, Z., 2017, MRMondrian: Scalable Multidimensional Anonymisation for Big Data Privacy Preservation, *IEEE Transactions on Big Data*, Early Access.

Zhao, Y. and Zhang, Y., 2008, Comparison of decision tree methods for finding active objects, *Advances in Space Research*, 41 (12), 1955-1959.

Zhu, D., Li, X.B. and Wu, S., 2009, Identity disclosure protection: A data reconstruction approach for privacy-preserving data mining, *Decision Support Systems*, 48 (1), 133-140.

Zikopoulos, P., DeRoos, D., Parasuraman, K., Deutsch, T., Corrigan, D. and Giles, J., 2013, *Harness the Power of Big Data The IBM Big Data Platform*, McGraw-Hill, ISBN: 978-0-07180818-7.

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Can EYÜPOĞLU
Doğum Yeri	İstanbul
Doğum Tarihi	1989
Uyruğu	☑ T.C. ❒ Diğer:
Telefon	0212 440 00 00
E-Posta Adresi	caneyupoglu@gmail.com
Web Adresi	http://orcid.org/0000-0002-6133-8617

Eğitim Bilgileri	
Lisans	
Üniversite	İstanbul Kültür Üniversitesi
Fakülte	Mühendislik Fakültesi
Bölümü	Bilgisayar Mühendisliği Bölümü (İngilizce)
Mezuniyet Yılı	2012

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Bilgisayar Mühendisliği Anabilim Dalı
Programı	Bilgisayar Mühendisliği Programı

Doktora	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Bilgisayar Mühendisliği Anabilim Dalı
Programı	Bilgisayar Mühendisliği Programı

Makale ve Bildiriler

Eyüpoğlu, C., Aydın, M.A., Zaim, A.H. and Sertbaş, A., 2018, An Efficient Big Data Anonymization Algorithm Based on Chaos and Perturbation Techniques, *Entropy*, 20 (5), 373, 1-18.

Yavuz, E., Kasapbaşı, M.C., Eyüpoğlu, C. and Yazıcı, R., 2018, An epileptic seizure detection system based on cepstral analysis and generalized regression neural network, *Biocybernetics and Biomedical Engineering*, 38 (2), 201-216.

Eyupoglu, S., Kut, D., Girisgin, A.O., Eyupoglu, C., Ozuicli, M., Dayioglu, H., Civan, M. and Aydin, L., 2018, Investigation of the bee-repellent properties of cotton fabrics treated with microencapsulated essential oils, *Textile Research Journal*, Accepted.

Eyupoglu, C., 2018, Breast Cancer Classification Using k-Nearest Neighbors Algorithm, *The Online Journal of Science and Technology*, Accepted.

Sanver, U., Yavuz, E. and Eyupoglu, C., 2018, Design and Implementation of a Programmable Logic Controller Using PIC18F4580, *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 29 January-1 February 2018, St. Petersburg, Russia.

Kasapbasi, M.C., Eyupoglu, C., Urut, T.B., Turan, Y., Karaagacli, A.O. and Sezgenc, Y.C., 2018, Implementation of Virtual Reality Enhanced Continuous Performance Test Designed for Attention Deficit Hyperactivity Disorder Diagnosis, *Journal of Selcuk-Technic*, 2018 (Special Issue), 1-10.

Eyüpoğlu, C., Aydın, M.A., Sertbaş, A., Zaim, A.H. and Öneş, O., 2017, Preserving Individual Privacy in Big Data, *International Journal of Informatics Technologies*, 10 (2), 177-184.

Eyupoglu, C., 2017, Implementation of Bernsen's Locally Adaptive Binarization Method for Gray Scale Images, *The Online Journal of Science and Technology*, 7 (2), 68-72.

Kasapbasi, M.C., Eyupoglu, C., Urut, T.B., Turan, Y., Karaagacli, A.O. and Sezgenc, Y.C., 2017, Implementation of Virtual Reality Enhanced Continuous Performance Test Designed for Attention Deficit Hyperactivity Disorder Diagnosis, *International Conference on Engineering Technologies (ICENTE'17)*, 7-9 December 2017, Konya, Turkey.

Yavuz, E., Eyupoglu, C., Sanver, U. and Yazici, R., 2017, An Ensemble of Neural Networks for Breast Cancer Diagnosis, *IEEE 2nd International Conference on Computer Science and Engineering (UBMK'17)*, 5-8 October 2017, Antalya, Turkey.

Eyupoglu, C., 2017, Breast Cancer Classification Using k-Nearest Neighbors Algorithm, *8th International Science and Technology Conference (ISTEC 2017)*, 17-19 July 2017, Berlin, Germany.

Eyupoglu, S., Sanver, U. and Eyupoglu, C., 2017, Use of Fibrous Materials in Acoustic Insulation Applications, *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 1-3 February 2017, St. Petersburg, Russia.

Sanver, U., Yavuz, E. and Eyupoglu, C., 2017, An Image Processing Application to Detect Faulty Bottle Packaging, *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 1-3 February 2017, St. Petersburg, Russia.

Eyupoglu, C., 2016, Clustering of Mitochondrial D-loop Sequences Using Similarity Matrix, PCA and K-means Algorithm, *International Journal of Intelligent Systems and Applications in Engineering*, 4 (Special Issue-1), 244-248.

Eyupoglu, C. and Yesilyurt, U., 2016, Modelling and Implementation of Network Coding for Video, *International Journal of Computer Network and Information Security*, 8 (8), 1-10.

Eyupoglu, C., 2016, Clustering of Mitochondrial D-loop Sequences Using Similarity Matrix, PCA and K-means Algorithm, *3rd International Conference on Advanced Technology & Sciences (ICAT'16)*, 1-3 September 2016, Konya, Turkey.

Eyupoglu, C., 2016, Implementation of Bernsen's Locally Adaptive Binarization Method for Gray Scale Images, *7th International Science and Technology Conference (ISTEC 2016)*, 13-15 July 2016, Vienna, Austria.

Eyupoglu, C., 2016, Implementation of Color Face Recognition Using PCA and k-NN Classifier, *2016 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (2016 ElConRusNW)*, 2-3 February 2016, St. Petersburg, Russia.

Eyupoglu, C., 2015, Investigation of the Performance of Nikhilam Multiplication Algorithm, *Procedia – Social and Behavioral Sciences*, 195, 1959-1965.

Eyupoglu, C., 2015, Performance Analysis of Karatsuba Multiplication Algorithm for Different Bit Lengths, *Procedia – Social and Behavioral Sciences*, 195, 1860-1864.

Eyupoglu, C. and Aydin, M.A., 2015, Energy Efficiency in Backbone Networks, *Procedia – Social and Behavioral Sciences*, 195, 1966-1970.

Kilinc, M., Canbolat, S., Eyupoglu, C. and Kut, D., 2015, The Evaluation with Statistical Analyses of the Effect of Different Storage Condition and Type of Gas on the Properties of Plasma Treated Cotton Fabrics, *Procedia – Social and Behavioral Sciences*, 195, 2170-2176.

Eyüpoğlu, C. and Sertbaş, A., 2015, Performance Comparison of Karatsuba and Nikhilam Multiplication Algorithms for Different Bit Lengths, *Istanbul Commerce University Journal of Science*, 14 (27), 55-64.

Eyupoglu, C., 2015, Investigation of the Performance of Nikhilam Multiplication Algorithm, *World Conference on Technology, Innovation and Entrepreneurship*, 28-30 May 2015, Istanbul, Turkey.

Eyupoglu, C., 2015, Performance Analysis of Karatsuba Multiplication Algorithm for Different Bit Lengths, *World Conference on Technology, Innovation and Entrepreneurship*, 28-30 May 2015, Istanbul, Turkey.

Eyupoglu, C. and Aydin, M.A., 2015, Energy Efficiency in Backbone Networks, *World Conference on Technology, Innovation and Entrepreneurship*, 28-30 May 2015, Istanbul, Turkey.

Kilinc, M., Canbolat, S., Eyupoglu, C. and Kut, D., 2015, The Evaluation with Statistical Analyses of the Effect of Different Storage Condition and Type of Gas on the Properties of Plasma Treated Cotton Fabrics, *World Conference on Technology, Innovation and Entrepreneurship*, 28-30 May 2015, Istanbul, Turkey.

Eyüpoğlu, C., Aydın, M.A. and Zaim, A.H., 2014, Investigation of Slotted Optical Switching Techniques, *Istanbul Commerce University Journal of Science*, 13 (25), 45-78.

Eyüpoğlu, C., Aydın, M.A. and Zaim, A.H., 2014, Slotted Optical Burst and Packet Switching Techniques, *XVI. Academic Informatics Conference*, 5-7 February 2014, Mersin, Turkey.