TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BÜYÜK VERİ VE AKAN VERİNİN MAHREMİYET KORUMALI ANONİMLEŞTİRİLMESİ

DOKTORA TEZİ

Uğur SOPAOĞLU

Bilgisayar Mühendisliği Anabilim Dalı

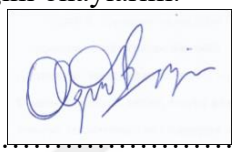
Tez Danışmanı: Doç. Dr. Osman ABUL

NİSAN 2020

Fen Bilimleri Enstitüsü Onayı

……..
Prof. Dr. Osman EROĞUL
Müdür

Bu tezin Doktora derecesinin tüm gereksinimlerini sağladığını onaylarım.



……
Prof. Dr. Oğuz ERGİN
Anabilim Dalı Başkanı

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün 141117006 numaralı Doktora Öğrencisi **Uğur SOPAOĞLU**'nun ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"BÜYÜK VERİ VE AKAN VERİNİN MAHREMİYET KORUMALI ANONİMLEŞTİRİLMESİ"** başlıklı tezi **21.04.2020** tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı : **Doç. Dr. Osman ABUL**
TOBB Ekonomi ve Teknoloji Üniversitesi

Jüri Üyeleri **Prof. Dr. İsmail Hakkı TOROSLU (Başkan)**
Orta Doğu Teknik Üniversitesi

Doç. Dr. Ahmet Murat ÖZBAYOĞLU
TOBB Ekonomi ve Teknoloji Üniversitesi

Doç. Dr. Mehmet TAN
TOBB Ekonomi ve Teknoloji Üniversitesi

Doç. Dr. Hacer KARACAN
Gazi Üniversitesi

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Uğur SOPAOĞLU

ÖZET

Doktora Tezi

BÜYÜK VERİ VE AKAN VERİNİN MAHREMİYET KORUMALI ANONİMLEŞTİRİLMESİ

Uğur SOPAOĞLU

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Osman ABUL

Tarih: Nisan 2020

Geleneksel veri anonimleştirme yöntemleri yalnız statik veri kümeleri için geliştirilmiş olup ölçeklenebilirlik hep ikinci planda kalmıştır. Büyük veri ve akan veri ihtiyaçlarının son yıllarda çeşitlenerek artması ile ölçeklenebilirlik ve verinin dinamikliği unsurları öne çıkmaya başlamıştır. Literatürde büyük veri ve akan veri mahremiyetinin sağlanmasına yönelik bu doğrultuda çalışmalar önerilmiş olsa da problemin çeşitli unsurları nedeniyle daha etkin ve daha kapsamlı veri anonimleştirme yöntemlerine ihtiyaç duyulmaktadır. Bu tez kapsamında büyük veri ve akan veri mahremiyetinin sağlanması için daha etkin ve daha kapsamlı anonimleştirme yöntemleri üzerinde çalışılmıştır.

Apache Spark büyük veri işleme alanında günümüzün en gelişmiş teknoloji ve platformları arasında yer almaktadır. Tezde, büyük veri anonimleştirmeyi de büyük veri işlemenin özel bir durumu olarak ele alıp yarı-tanımlayıcı özniteliklerin alan hiyerarşisi üzerinde yukarıdan-aşağıya özelleşme arama tekniğini kullanan dağıtık bir büyük veri k-anonimleştirme yöntemi önerilmiştir. Arama kriteri olarak bilgi kazancı – mahremiyet kaybı metriği kullanılmıştır. Yöntemin verimliliği ve ölçeklenebilirliği büyütülmüş gerçek veri kümeleri üzerinde gösterilmiştir.

Literatürde akan veriyi k-anonimleştirmeye yönelik geliştirilen çözümler, problemi yarı-tanımlayıcı özniteliklerin bilgi kaybı metriğini minimize etmeye çalışan tek amaçlı optimizasyon problemi olarak formüle eden dar kapsamlı çözümlerdir. Tez kapsamında tespiti yapılan ihtiyaçlara yönelik olarak daha kapsamlı çözümler önerilmiş ve gerçek veri kümeleri üzerinde etkinlikleri geniş kapsamlı deneysel çalışmalarla gösterilmiştir.

İlk olarak, akan veri için bilgi kaybı ile ortalama gecikme süresini beraber minimize etmeye yönelik çok amaçlı bir optimizasyon çatısı önerilmiştir. Böylelikle, akan veri için veri kullanışlılığı, bilgi kaybı metriği ile ölçülen veri kalitesi ve ortalama gecikme süresi metriği ile ölçülen veri güncelliğinin bir fonksiyonu olarak ele alınmıştır. Önerilen yöntemde bu iki bileşen kullanıcı tarafından ağırlıklandırılabilmektedir. İlave olarak, probleme özgü yeni bir bilgi kaybı metriği tanıtılmıştır.

İkinci olarak, veri alıcısının akan anonim veri üzerinde yapacağı analiz işleminden haberdar bir k-anonimleştirme çatısı önerilmiştir. Birçok veri alıcısının anonim veri üzerinde sınıflandırma veri madenciliği görevi çalıştırdığı bilinmektedir. Bu yüzden, bu çalışmada bilgi kaybını minimize etmenin yanında sınıflama doğruluğunu maksimize etmek de bir diğer amaçtır. Hatta akan veride, yarı-tanımlayıcı öznitelikler ve sınıflama hedef özniteliğine ilave olarak hassas öznitelikler olması durumunda bunların hassasiyetinin de en üst düzeyde korunması gerekir. Önerilen yöntem, ağırlıkları kullanıcı tarafından belirlenebilen, bu üç amaçlı optimizasyon problemini çözmektedir.

Anahtar Kelimeler: Veri mahremiyeti, Büyük veri, Akan veri, Anonimleştirme.

ABSTRACT

Doctor of Philosophy

PRIVACY PRESERVING ANONYMIZATION OF BIG DATA AND DATA STREAMS

Uğur SOPAOĞLU

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Osman ABUL

Date: April 2020

Traditional data anonymization methods have been developed only for static datasets, where the scalability has usually been disregarded. With the diversified increase of big data and streaming data needs in recent years, the scalability and dynamic nature of data started to come to the foreground. Although studies have been proposed in the literature to provide big data and streaming data privacy solutions, more effective and high coverage data anonymization methods are needed due to various traits of the problem. Within the scope of this thesis, more effective and high coverage anonymization methods have been studied to ensure big data and streaming data privacy.

Apache Spark is among the most advanced technologies and platforms in the field of big data processing. In this thesis, a distributed big data k-anonymization method is proposed, which takes big data anonymization as a special case of big data processing and uses the top-down specialization search technique on the domain hierarchy of quasi-identifier attributes. Information gain - privacy loss metric is used as the search criteria. The effectiveness and the scalability of the method have been demonstrated on extended real datasets.

The solutions developed for k-anonymization of data streams in the literature are low coverage solutions that formulate the problem as a single-objective optimization problem that tries to minimize the information loss metric on quasi-identifier attributes. High coverage solutions have been proposed for the needs identified within the scope of the thesis and their effectiveness on real data sets has been shown through extensive experimental evaluations.

First, a multi-objective optimization framework is proposed to minimize the information loss and average delay together for streaming data. Thus, the data utility for streaming data is measured as a function of the data quality measured by the information loss metric and the data aging measured by the average delay metric. In the proposed method, the component weights can be tuned by the user. Moreover, a custom information loss metric is introduced.

Secondly, a down-stream data analysis process aware k-anonymization framework is proposed. Many data recipients are known to run classification data mining tasks on the anonymized data. Therefore, in this study, besides minimizing information loss, maximizing classification accuracy is another objective. In fact, in case there exists sensitive attributes in addition to the quasi-identifier and the classification target attributes, the sensitivity of these sensitive attributes should be maintained at the highest level. The proposed method solves this three-objective optimization problem, the weights of which can be tuned by the user.

Keywords: Data privacy, Big data, Data streams, Anonymization.

TEŞEKKÜR

Danışmanım Doç. Dr. Osman ABUL'a, çalışmalarım boyunca bana sağladığı kıymetli desteklerinden ve sabrından dolayı çok teşekkür ederim. Tez izleme komitemde yer alan kıymetli hocalarım Doç. Dr. Murat ÖZBAYOĞLU, Doç. Dr. Mehmet TAN ve Doç. Dr. Hacer KARACAN'a yönlendirmeleri ve değerli fikirlerinden dolayı teşekkür ederim.

Doktora eğitimim süresince her zaman yanımda olduğunu hissettiğim değerli eşim Saide SOPAOĞLU'na ve hayatım boyunca desteklerini benden esirgemeyen annem "Leyla", babam "Zeynel" ve abim Ufuk SOPAOĞLU'na çok teşekkür ederim. Güzel gözleri ve gülüşü ile bana mutluluk veren biricik yeğenim Duru SOPAOĞLU'na teşekkür ederim. Beni zorlu süreçte hiç yalnız bırakmayan Yüksel, Emir, Pelin ve Ahmet ATIGAN'a teşekkürü bir borç bilirim.

Ayrıca doktora eğitimim boyunca bana maddi destek sağlayan TÜBİTAK'a teşekkür ederim.

İÇİNDEKİLER

ŞEKİL LİSTESİ

Sayfa

ÇİZELGE LİSTESİ

Sayfa

KISALTMALAR

AL : Anonimizasyon seviyesi (Anonymization level)
BUG : Aşağıdan yukarıya genelleştirme (Bottom-up generalization)
CAIL : Cardinality aware information loss
IFTDS : İki fazlı yukarıdan aşağıya özelleştirme
ICT : Bilgi ve iletişim teknolojileri
IL : Bilgi kaybı (Informaion loss)
IoT : Nesnelerin interneti (Internet of things)
MB : Megabayt
QI : Yarı tanımlayıcı öznitelik(Quasi-identifier)
QI-grup : Yarı tanımlayıcı özniteliklerden oluşan grup
RAM : Rastgele erişilebilir bellek (Random access memory)
RDD : Resilient distributed datasets
TDS : Yukarıdan aşağıya özelleştirme (Top-down specialization)
UBDSA : Utility based data stream anonymization
ZB : Zettabayt

SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
k	Anonimizasyon değeri
ℓ	Çeşitlilik parametresi
t_i	Kayıt
t_i'	Anonim kayıt
u	QI-grup içerisinde bir özniteliğin üst sınırı
U	Nümerik özniteliklerin tanım kümesi içerisindeki üst sınırı
l	QI-grup içerisinde bir özniteliğin alt sınırı
L	Nümerik özniteliklerin tanım kümesi içerisindeki alt sınırı
v	Bir kategorik öznitelik için tanımlanmış taksonomi ağacında bir düğüm
R	Veri kümesi
R_v	Taksonomi ağacında v değeri altında bulunan yaprak düğüm kümesi
AL_i	Anonimizasyon seviyesi
k^I	Veri parçası için sağlanması gereken anonimizasyon değeri
D	Büyük veri kümesi
D_i	Veri parçası
AL^*	Son anonimizasyon seviyesi
S	Akan veri
S'	Anonimleştirilmiş akan veri
A_i	i'nci öznitelik
SV	Hassas değer
T'	Yarı tanımlayıcı değerleri aynı kayıt kümesi
δ	Gecikme süresi
β	Saklanabilecek anonimleştirilmemiş küme sayısı
μ	Saklanabilecek anonimleştirilmiş küme sayısı
t_o	Sistemde bulunan en eski kayıt

1. GİRİŞ

Bilgi teknolojilerinin gelişmesi ile birlikte üretilen ve saklanan veri miktarı da artış göstermiştir. Bu verilerden anlamlı sonuçlar elde etmek için geliştirilen veri analizi yöntemleri de süreç içerisinde ciddi gelişim göstermişlerdir. Fakat bu süreç kişisel verilerin mahremiyeti problemini de beraberinde getirmiştir. İlk başlarda veri içerisinde bir kişiyi doğrudan tanımlayacak özniteliklerin veri kümesi içerisinden çıkartılması ile bu sorun aşılmaya çalışılsa da, bu işlemin hassas verinin mahremiyetini korumak için yeterli olmadığı görülmüş, akabinde veri mahremiyeti korumak için anonimleştirme ve kriptografi tabanlı birçok yöntem önerilmiştir.

Önerilen yöntemler veri mahremiyetini büyük ölçüde korumasına rağmen ölçeklenebilir çözüm sağlamadığı için veri miktarının artması ilgili yöntemlerin kullanılamamasına neden olmuştur. Büyük veri kavramı ile veri güdümlü (*data-driven*) çözümler ölçeklenilir bir yapıya evrimleşmiş, veri mahremiyeti çalışmaları da bu yönde gelişim göstermeye başlamıştır. Geliştirilen büyük veri işleme platformları yardımı ile verinin dağıtık bir şekilde işlenmesi sağlanmıştır. Literatürde önerilen veri mahremiyeti çalışmalarının da dağıtık bir şekilde yapılmaya başladığı görülmektedir.

Statik veri kümeleri için geliştirilen veri anonimleştirme çözümleri, akan veri için doğrudan kullanılamamaktadır. Akan verinin dinamik yapısı gereği, veri mahremiyeti çalışmalarının da tekrar revize edilmesi gerekmiştir. Akan veri üzerine geliştirilen çalışmalarda, kayıtların kullanılabilirliği ve sistem kaynağının sınırlı olması göz önüne alınarak sisteme gelen kayıtlar için bir maksimum gecikme kısıtı tanımlandığı görülmektedir, öyle ki hiç bir kayıt için bu sınır aşılmamalıdır. Önerilmiş olan yöntemler anonimleştirilmiş akan verinin bilgi kaybı miktarını minimum seviyede tutmayı hedeflemektedirler.

Tez kapsamında büyük veri için Apache Spark tabanlı yukarıdan-aşağıya özelleşme tekniğini kullanan ölçeklenebilir bir anonimleştirme çözümü sunulmaktadır. Ayrıca akan veriler için bilgi kaybı miktarı ile beraber ortalama gecikme süresinin de minimum seviyede tutulmasının hedeflendiği bir akan veri anonimleştirme yöntemi

önerilmektedir. Anonimleştirilen verilerin çoğunlukla sınıflandırma (*classification*) ya da regresyon (*regression*) veri madenciliği görevlerinde kullanıldığı bilinmektedir. Anonimleştirilecek akan verinin sınıflandırma görevlerinde kullanılacağı durumlarda, akan veri için bilgi kaybı miktarının minimize edildiği, sınıflama doğruluğunun maksimum seviyede tutulduğu ve hassas veriler üzerindeki hassasiyetinde en üst düzeyde korunduğu üçüncü bir yöntem tez kapsamında önerilmektedir.

1.1 Tezin Amacı

Statik veri kümeleri için geliştirilmiş veri mahremiyeti çalışmalarında sıklıkla tercih edilen geleneksel anonimleştirme çözümleri birçok kısıt nedeniyle büyük veri ve akan verinin anonimleştirilmesinde kullanılamamaktadır.

Literatürde var olan büyük verinin anonimleştirilmesi ile ilgili geliştirilmiş çalışmalarda ölçeklenebilirlik ve verimlilik problemleri bulunmaktadır. Tez kapsamında sunulacak olan büyük verinin anonimleştirilmesi yöntemi ile daha verimli ve ölçeklenebilir bir çözüm önerilmektedir.

Akan verinin dinamik yapısına uygun bir şekilde anonimleştirme işlemlerinin de dinamik yapılabilmesi gerekmektedir. Ayrıca akan veride anonimleştirilecek kayıtların sistemde, verinin zaman değeri yüzünden, ne kadar süre bekletildiği de önemli bir kriterdir. Bu sürenin uzatılması hem verinin yaşlanması yani önemini kaybetmesine hem de gelecek verinin miktarının bilinmemesinden dolayı sistemde kaynak sıkıntısına neden olmaktadır. Tez kapsamında bu iki kriter dikkate alınarak kayıtların sistemde ortalama gecikme süresi açısından daha az süre tutulduğu ve bilgi kaybı açısından etkin çalışan bir akan veri anonimizasyon yöntemi amaçlanmaktadır.

Anonimleştirilmiş verinin kalite (*quality*) ve kullanılabilirliğinin (*utility*) azaldığı bilinen bir gerçektir. Tez kapsamsında ayrıca anonimleştirilecek akan verinin sınıflandırma görevlerinde kullanılabilirliğini de dikkate alan, bilgi kaybı miktarı ve verinin sınıflandırma için kullanılabilirliği arasında önceliklendirme yapılmasına olanak sağlayacak bir yöntem amaçlanmaktadır.

1.2 Tezin Literatüre Katkıları

Tez içerisinde büyük veri ve akan veri için geliştirilmiş üç yöntem bulunmaktadır ve bu yöntemler ile literatüre yapılan katkılar şu şekildedir:

1. Geliştirilen Apache Spark tabanlı büyük veri çözümü ile ölçeklenebilirlik ve verimlilik açısından başarılı bir yöntem sunulmuştur. Bu yöntem ile verinin bir bilgisayar kümesi üzerinde dağıtılarak, anonimleştirme yönteminin ihtiyaç duyduğu bütün matematiksel işlemlerin dağıtık bir şekilde gerçekleştirilip elde edilen sonuçlara göre veri kümesinin anonimizasyonu sağlanmaktadır.

2. Literatürde bulunan çalışmalarda statik ya da toplu olarak isimlendirilen veri kümeleri ile akan veri arasındaki farklar vurgulanmaktadır. Özellikle akan verinin dinamik yapısı gereği sisteme gelen verinin sistemde uzun süre bekletilmemesi gerektiği için kayıtların sistemde bekleyebilecekleri süre sabit bir değer ile sınırlandırılmıştır ve bu değer anonimleştirme işleminden önce belirlenmektedir. Tez kapsamında önerilen yöntemde süre açısından bir üst sınır olacak şekilde çalışma esnasında dinamik olarak değiştirilerek kayıtların ortalama gecikme süresi minimize edilmektedir. Ayrıca literatürde bulunan çalışmalarda bilgi kaybı ve genişleme gibi kayıtların yakınlığını ölçmekte kullanılan metrikler yerine CAIL (*Cardinality Aware Information Loss*) adında yeni bir metrik tanımlanmıştır. Önemli akan veri anonimleştirme yöntemlerinden birisi olan CASTLE (Cao, 2010) içerisinde kullanılan genişleme (*enlargement*) metriği yerine CAIL kullanıldığında başarısının arttığı gözlenmiştir. Ayrıca literatürde bilgi kaybı açısından en iyi sonuçları veren FADS (Guo, 2013) yöntemi ile karşılaştırıldığında önerilen yöntemin ortalama gecikme süresi açısından daha iyi sonuçlar verdiği görülmektedir. Ayrıca önerilen yöntem için gerçekleştirilen deneylerde kullanılan veri kümelerinden birisi olan TELCO, sonrasında geliştirilecek olan akan veri anonimleştirme çalışmaları için kıyaslama (*benchmark*) veri kümesi olarak literatüre sunulmuştur.

3. Statik veri kümeleri için geliştirilen birçok yöntemde üretilen anonim veri kümesi ile beslenen sınıflandırma algoritmalarının doğruluk oranlarının düştüğü vurgusu yapılmakta ve anonimleştirme algoritmalarının bu yönde değiştiği görülmektedir. Fakat akan veri için önerilen anonimizasyon çözümlerinde bu yönde bir çalışma literatürde bulunmamaktadır. Bu açığı kapatabilmek için yeni bir akan veri anonimleştirme algoritması önerilmektedir. Bu algoritma akan verinin anonimliğini sağlarken üretilecek

çıktı ile beslenecek sınıflandırma algoritmalarının başarısını da korumayı hedeflemektedir. Sunulan bu algoritma ile üretilen anonim akan veri ile beslenen sınıflandırma algoritmalarının başarıları literatürde bilinen diğer algoritmalara kıyasla istatistiksel olarak daha iyi sonuçlar vermektedir.

Tezin geri kalan kısmı beş bölümden oluşmaktadır. Bölüm 2'de veri mahremiyeti problemi açıklanarak bu problemin çözümü için geliştirilmiş yöntemler anlatılacaktır.

Bölüm 3 içerisinde büyük verinin mahremiyeti ile ilgili literatürde varolan çalışmalar anlatılacak ve tez kapsamında hazırlanmış olan büyük verinin anonimleştirilmesi yöntemi sunulacaktır. Bu yöntemin ölçeklenebilirlik ve verimlilik açısından değerlendirilebilmesi için gerçekleştirilen deneylere ait sonuçlar da verilecektir.

Bölüm 4'te akan veri mahmremiyeti ve statik veri kümelerinin mahremiyeti arasındaki farklar açıklanacaktır. Akan veri ile ilgili geliştirilmiş anonimleştirme yöntemlerinden bahsedilecek ve tez kapsamında geliştirilen akan veri anonimleştirme yöntemi açıklanacaktır. Ayrıca bu yöntemin değerlendirilebilmesi için yapılan deneylere yönelik sonuçlar diğer akan veri anonimleştirme çalışmaları ile karşılaştırmalı bir şekilde verilecektir.

Bölüm 5'te anonimleştirilen verinin sınıflandırma algoritmaları tarafından kullanılabilirliğini dikkate alan bir akan veri anonimleştirme yöntemi sunulacaktır. Bu yöntemin etkinliğini göstermek için yapılan deneyler ve bu deneylere ait sonuçlar gösterilecektir. Ayrıca diğer popüler akan veri anonimleştirme yöntemleri ile de karşılaştırılacaktır.

Bölüm 6'da tez kapsamında geliştirilen yöntemlere ait genel bir değerlendirme yapılacaktır. Ayrıca gelecekte yapılabilecek çalışmalar için çeşitli öneriler sunulacaktır.

2. VERİ MAHREMİYETİNİN KORUNMASI

Çeşitlenerek artan veri toplama, işleme ve analizi yöntemleri ile birlikte, veri mahremiyetinin korunması da kritik bir problem haline gelmiştir. Veri mahremiyeti koruma çalışmaları yapısal veriler için iki farklı tehdidi engellemeye çalışmaktadır.

1- Kimlik İfşası: Veri kümesi içerisindeki bir kaydın kime ait olduğunun doğrudan tespit edilebildiği durumlardır. Yayınlanan veri kümesinin başka veri kümeleri ile ortak özniteliklerinin eşleştirilmesi ile bir kaydın kime ait olduğu ortaya çıkarılabilmektedir. Örneğin Samarati ve Sweeney tarafından geliştirilen çalışmada (Samarati, 1998), Birleşik Devletler'de yayınlanan hasta muayene verileri ve seçmen listeleri içerisinde bulunan *Posta Kodu*, *Doğum Tarihi* ve *Cinsiyet* öznitelikleri üzerinden yaptığı eşleştirme sonucunda Birleşik Devletler nüfusunun %87'sinin kimliğinin ortaya çıkarılabildiği belirtilmiştir.

2- Hassas Özniteliğin İfşası: Veri kümesi içerisindeki bir bireyin hassas verisinin tespit edildiği durumlardır. Hassas özniteliğin ifşası için iyi bilinen iki atak türü bulunmaktadır (Machanavajjhala, 2007).

a. Verinin homojenliğine dayalı ataklar

b. Arka plan bilgisine dayalı ataklar

Belirtilen bu saldırılar ile ilgili detaylı bilgi Bölüm 2.1.1'de verilmiştir. Yukarıda bahsedilen tehditlere karşı önlem amaçlı birçok çalışma da yapılmıştır. Bu çalışmalar anonimleştirme ve diferansiyel mahremiyet tabanlı olarak iki başlık altında toplanabilir.

2.1 Anonimleştirme Tabanlı Veri Mahremiyeti

Verinin anonimleştirilmesi, yayınlanacak olan bir veri kümesi içerisindeki kayıtların kime ait olduğu tespit edilemeyecek şekilde kimliksizleştirilmesi işlemine verilen

addır. Yapısal, yarı-yapısal ve yapısal olmayan veri kümeleri için geliştirilmiş birçok anonimleştirme çalışması bulunmaktadır. Bu tez kapsamında yapısal veri kümelerinin anonimleştirilmesi üzerine çalışılmıştır. Yapısal bir veri kümesi içerisindeki öznitelikler dört gruba ayrılmıştır: (i) Doğrudan tanımlayıcı (*Identifier, ID*) öznitelikler: bir kişiyi doğrudan tanımlayan veriler, örneğin TC kimlik numarası veya sosyal güvenlik numarası, (ii) Yarı-tanımlayıcı (*Quasi-Identifier,* QI) öznitelikler: tek başına bir kişiyi tanımlamak için yeterli olmamasına rağmen, başka veri kümeleri ile eşleştirildiğinde bir kişinin tanımlanmasına neden olabilecek özniteliklerdir. Örneğin posta kodu, yaş ve cinsiyet. (iii) Hassas (*Sensitive*) öznitelikler: ortaya çıkması istenmeyecek gizli veriler olarak tanımlanabilir. (iv) Hassas olmayan (*Non-sensitive*) öznitelikler: belirtilen özniteliklerin dışında kalan verilerdir.

Anonimleştirme tabanlı çözümlerde dört aktör bulunmaktadır:

1. Veriyi yayınlayan kişi ya da kurum
2. Verinin sahibi, özne
3. Paylaşılan veriye erişen kişi ya da kurumlar
4. Veriye atak düzenleyen kişi, saldırgan

Anonimliği sağlanan verinin yayınlanması üç farklı şekilde yapılmaktadır:

1. *Tek sürüm yayınlama*: Bir veri kümesi ya da veri kümesi içerisinden bir bölümün daha sonra üzerinde tekrar bir anonimleştirme işlemi yapılmayacak şekilde anonimleştirilip yayınlanmasıdır.
2. *Paralel yayınlama:* Anonimleştirmeden kaynaklı bilgi kaybının azaltılması için yarı tanımlayıcıların farklı gruplar halinde anonimleştirilip yayınlanmasıdır (Yao, 2005), (Kifer, 2006).
3. *Sıralı yayınlama:* Anonimleştirilmiş veri kümesinin artımlı bir şekilde yayınlanmasıdır. Bir örnekle açıklamak gerekirse, elinde bulundurduğu müşteri bilgilerinin anonimliğini sağlayarak paylaşan bir kurumun, süreç içerisinde portföyüne eklediği yeni müşteriler ile veri kümesinin tekrar anonimleştirilerek paylaşılmasıdır. Anonimleştirme işlemi yapılırken, herhangi bir ifşaya neden olmamak için daha önce yayınlanan veri kümeleri dikkate alınarak veri kümesinin son hali anonimleştirilir (Wang, 2006) (Bu, 2008).

Başlıca anonimleştirme tabanlı veri mahremiyeti koruma yöntemleri şunlardır:

- **Genelleştirme (*generalization*):** Veri kümesi içerisinde bulunan verilerin daha genel bir temsili ile değiştirilmesi yöntemidir. Bu tez kapsamında önerilen yöntemlerde, genelleştirme tabanlı çözümler tercih edilmiştir.
- **Suppression:** Bir veri kümesi içerisinde ya da bir yarı tanımlayıcı grup içerisinde bir özniteliğe ait değerlerin silinmesi ya da bu değerlerin ilgili özniteliğin altında bulunan bütün değerleri kapsayacak bir değer ile değiştirilmesidir. Genelleştirme tabanlı anonimleştirme yöntemlerinin birçoğunun içerisinde de kullanılmaktadır.
- **Bucketization**: Bu yöntem içerisinde hassas veriler ve yarı tanımlayıcıların birbirleri arasındaki bağ kopartılarak iki tablo şeklinde yayınlanır. Yarı tanımlayıcılar ve hassas veriler üzerinde herhangi bir değişiklik yapılmamaktadır. Bir yarı tanımlayıcı grup, bir hassas veri grubu ile genelde ID üzerinde eşleştirilir. Fakat yayınlanan veri kümesi içerisinde kayıtlar özelinde bir hassas değer ve yarı tanımlayıcı öznitelikler arasında eşleştirme için kullanılacak bir bilgi bulunmamaktadır (Xiao, 2006).
- **Veri bozulması (*data perturbation*):** Veri kümesi içerisindeki değerlerde değişiklik yapılarak (sentetik veriler ile değiştirme, gürültü ekleme vb.) verinin mahremiyeti korunmaya çalışılmaktadır. Üretilen anonim verinin kullanılabilirliği önemli bir kriter olduğu için değiştirilen değerler ve orijinal değerler arasında istatistiksel açıdan büyük fark olmamasına dikkat edilir. Veri bozulması ile ilgili literatürde birçok çalışma bulunmaktadır (Kargupta, 2003), (Liu, 2005), (Muralidhar, 1999), (Chen, 2007). İyi bilinen anonimleştirme yöntemlerinden birisi olan microaggregation yöntemi de bir veri bozulması yaklaşımıdır. Bu yaklaşım, yarı tanımlayıcı bir grup veriyi ifade edecek özet istatistiksel bilgiler ile değiştirildiği yöntemdir (Domingo-Ferrer, 2002), (Solé, 2012), (Sánchez, 2019), (Domingo-Ferrer, 2010).
- **Sahte anonimleştirme (*pseudonymisation*):** Bir kişiyi doğrudan tanımlayan verilerin sahte veriler ile değiştirilmesi yaklaşımıdır. Verinin mahremiyetini korumak için tam anlamıyla yeterli değildir (Neubauer, 2011), (Riedl, 2007).

- **Permütasyon:** Anonimleştirilecek veri kümesi içerisindeki kayıtlar gruplara dağıtılır ve gruplardaki kayıtların hassas verileri birbirleri arasında değiştirilir (Zhang, 2007).

2.1.1 Genelleştirme tabanlı anonimleştirme yöntemleri

Anonimleştirme tabanlı veri mahremiyeti çalışmalarında genelleştirme yaklaşımı sıklıkla tercih edilmektedir. Bir yarı tanımlayıcı öznitelik içerisinde bulunan bir grup değerin, bu değerleri temsil edecek daha genel bir değer ile değiştirilmesi işlemine genelleştirme denilmektedir. Genelleştirme işlemi için çıktı nümerik ve kategorik değerler için farklılık göstermektedir. Nümerik bir değer için genelleştirme uygulandığında yeni değer bir aralık ile ifade edilirken, kategorik bir değer genelleştirme sonucunda yeni bir değer ile ifade edilir. Genelleştirme temelinde geliştirilmiş ve veri mahremiyeti çalışmalarının birçoğunun temelini oluşturan üç geleneksel yöntem bulunmaktadır: (i) $\mathrm{k}-$anonimlik, (ii) $\ell-$ çeşitlilik ve (iii) $\mathrm{t}-$yakınlık.

2.1.1.1 k-anonimlik (k-*anonymity*)

Samarati ve Sweeney (Samarati, 1998) tarafından veri mahremiyetinin korunabilmesi için önerilen k-anonimlik yöntemi, bu kapsamda geliştirilen ilk yöntemlerden birisidir. Bu yöntemde, veri kümesi içerisinden öncelikli olarak doğrudan tanımlayıcı öznitelikler çıkartılır. Sonrasında, veri kümesi içerisinde bulunan her bir kayıt ile en az $\mathrm{k}-1$ farklı kaydın yarı tanımlayıcı özniteliklerine ait değerleri aynı olana kadar genelleştirilir. Böylece bir kişinin yayınlanan veri kümesi içerisinde tahmin edilme olasılığı en fazla $\frac{1}{\mathrm{k}}$ olmaktadır.

Çizelge 2.1'de beş öznitelikten oluşan ve anonimleştirilmesi istenen yapısal bir veri kümesi verilmiştir. *ID* doğrudan tanımlayıcı bir öznitelikken, *Yaş*, *Eğitim* ve *Cinsiyet* öznitelikleri yarı tanımlayıcı özniteliklerdir. *Teşhis* ise bu veri kümesinin hassas özniteliğidir. Bu veri kümesi için $k-$anonimlik çözümü uygulanmadan önce doğrudan tanımlayıcı öznitelik olan ID veri kümesi içerisinde çıkartılır. 2-anonimlik yaklaşımı veri kümesine uygulanır ve algoritmanın çıktısı Çizelge 2.2'de verilmiştir. Yayınlanan veri kümesine bakıldığında 1. ve 2. kayıt için *Cinsiyet*, 3. ve 4. kayıt için *Yaş* ve *Eğitim*, 5. ve 6. kayıtlar için ise *Yaş*, Cinsiyet ve *Eğitim* alanlarına ait değerler genelleştirilmiştir. Böylece, yayınlanan veri kümesi k-anonimlik yaklaşımının

gereksinimi olan her bir kayıt için veri kümesi içerisinde en az $k-1$ tane aynı yarı tanımlayıcı değere sahip kayıt olması koşulunu sağlamaktadır. Aynı QI değerine sahip gruplara, yarı tanımlayıcı grup (QI-grup) adı verilmektedir.

Çizelge 2.1 k-anonimlik uygulanmadan önce orijinal veri kümesi.

ID	Yaş	Eğitim	Cinsiyet	Teşhis
1	10	4. Sınıf	Erkek	Zatürre
2	10	4. Sınıf	Kadın	Sinüzit
3	12	6. Sınıf	Erkek	Nezle
4	13	7. Sınıf	Erkek	Bronşit
5	17	11. Sınıf	Kadın	Nezle
6	16	12. Sınıf	Erkek	Zatürre

Çizelge 2.2 $2-$anonimlik uygunlandıktan sonra.

Yaş	Eğitim	Cinsiyet	Teşhis
10	4. Sınıf	*	Zatürre
10	4. Sınıf	*	Sinüzit
[12-13]	Ortaokul	Erkek	Nezle
[12-13]	Ortaokul	Erkek	Bronşit
[16-17]	Lise	*	Nezle
[16-17]	Lise	*	Zatürre

$k-$anonimlik yaklaşımı içerisinde her kaydın farklı bir kişiye ait olduğu varsayımı yapılmaktadır. Fakat birçok veri kümesi içerisinde bir kişiye ait birden fazla kayıt bulunabilmektedir. Bu problemin önüne geçebilmek için $(X, Y)-$anonimliği yöntemi önerilmiştir. X ile yarı tanımlayıcı öznitelikler kümesi, Y ise veri kümesi içerisinde bir kişiyi doğrudan tanımlayan öznitelikler kümesini ifade etmektedir. Bu yöntem içerisinde, X kümesi içerisindeki özniteliklerin değerlerinin Y kümesi içerisinden en az k tane farklı veri ile eşleşmesi istenilir (Wang, 2006).

k-anonimlik yöntemi temel alınarak birçok yeni yöntem de geliştirilmiştir (LeFevre, 2005), (Aggarwal, 2005), (Bayardo, 2005). Bu tez kapsamında önerilen yöntemler k-anonimlik tanımının gereksinimlerini karşılamaktadır.

2.1.1.2 $\ell -$ çeşitlilik ($\ell -$*diversity*)

$\ell -$ çeşitlilik çalışması (Machanavajjhala, 2007), $k -$anonimlik yaklaşımının veri mahremiyetini tam anlamı ile koruyamadığı durumlar olduğunu göstermiştir. $k -$anonimlik yöntemi ile anonimleştirilen bir veri kümesine homojenlik ya da geçmiş bilgi atağı düzenlenerek bir kayda ait hassas veriler tespit edilebilmektedir. Bu problem hassas verinin ifşası olarak da adlandırılır.

Homojenlik atağı: Yayınlanan anonim veri kümesi içerisinde herhangi bir QI-grubun hassas verileri eğer aynı değere sahipse, o grupta olduğu tespit edilen bir kişinin hassas bilgisi ortaya çıkmaktadır. Örneğin Çizelge 2.3'te 3-anonimlik uygulanmış bir veri kümesi gösterilmektedir. Bu anonim veri kümesi incelendiğinde QI değerleri aynı olan üçerli gruplar görülmektedir. Bu veri kümesi içerisindeki ilk QI-grubun hassas verisi olan *Teşhis* özniteliği üç kayıt içinde aynı değere sahiptir. Dolayısıyla, o grupta olduğu bilinen bir kişiye *COVID-19* teşhisi konulduğu ortaya çıkmaktadır.

Çizelge 2.3 3 $-$anonimlik uygulandıktan sonra.

Yaş	Eğitim	Cinsiyet	Teşhis
[40-45]	Üniversite	*	COVID-19
[40-45]	Üniversite	*	COVID-19
[40-45]	Üniversite	*	COVID-19
[18-20]	Lise	Erkek	Nezle
[18-20]	Lise	Erkek	Nezle
[18-20]	Lise	Erkek	Bronşit
[30-34]	Üniversite	*	Kanser
[30-34]	Üniversite	*	Viral Enfeksiyon
[30-34]	Üniversite	*	Ülser

Geçmiş bilgi atağı: Bu atak, anonim hale getirilmiş veri kümesini ele geçiren kötü niyetli bir kişinin geçmiş bilgisini kullanarak hassas veriyi tespit etme durumudur. Geçmiş bilgi kullanarak hassas verinin ifşası iki yol ile yapılabilir:

1. Pozitif ifşa: Hassas değerin yüksek bir olasılıkla saldırıyı gerçekleştiren kişi tarafından tahmin edilebildiği durumlardır.

2. Negatif ifşa: Saldırıyı gerçekleştiren kişinin veri kümesi içerisinden tespit etmek istediği kayda ait olmayan hassas değerleri elemesi işlemidir.

Çizelge 2.4 $2-$ Çeşitlilik uygulandıktan sonra.

Yaş	Eğitim	Cinsiyet	Teşhis
≥ 30	Üniversite	*	COVID-19
≥ 30	Üniversite	*	Ülser
≥ 30	Üniversite	*	COVID-19
[18-20]	Lise	Erkek	Nezle
[18-20]	Lise	Erkek	Nezle
[18-20]	Lise	Erkek	Bronşit
≥ 30	Üniversite	*	Kanser
≥ 30	Üniversite	*	Viral Enfeksiyon
≥ 30	Üniversite	*	COVID-19

$\ell-$ çeşitlilik yöntemi (Machanavajjhala, 2007) yukarıda belirtilen problemleri çözmek için geliştirilmiştir. Bunun için yayınlanacak olan her bir QI-grup içerisinde bulunan kayıtların en temel çeşitlilik tanımı ile en az ℓ tane farklı hassas veri barındırması gerekmektedir. Yayınlanan veri kümesinin $\ell-$farklılığın gereksinimlerini sağlaması için ancak bütün QI-grupları içerisinde en az ℓ farklı hassas veri olması gerekmektedir. Çizelge 2.4'de $2-$ çeşitlilik yaklaşımının uygulandığı veri kümesi gösterilmiştir. Görüldüğü üzere, her bir QI-grup içerisinde en az 2 farklı hassas veri bulunmaktadır. Burada bahsi geçen en temel $\ell-$çeşitlilik tanımı yanında bilgi teorisine dayalı tanımları da vardır.

Bilgi kaybı miktarını azaltmak için $\ell^{+}-$ çeşitlilik yöntemi (Liu, 2010) önerilmiştir. Bu yöntem içerisinde, bütün hassas değerler için bir çeşitlilik eşiği belirlemek yerine her bir hassas değer için farklı eşikler belirlenir ve böylece verideki bozulma miktarı azalır.

2.1.1.3 $t-$yakınlık ($t-$*closeness*)

Li ve arkadaşları tarafından önerilen $t-$yakınlık metodu (Li, 2007), $\ell-$ çeşitlilik ve k-anonimlik yaklaşımlarının hassas özniteliğin ifşası ataklarına karşı olan açıklarını kapatmak için geliştirilmiştir. Bir önceki bölümde belirtildiği üzere, k-anonimlik yaklaşımı hassas öznitelikleri açığa çıkartmak için yapılacak saldırılara karşı tam anlamıyla bir koruma sağlamamaktadır. $\ell-$ çeşitlilik yaklaşımı her bir QI-grup için ℓ tane farklı hassas verinin bulunması gerekliliğini savunmuştur. Fakat bu yaklaşımda QI-grup içerisinde bulunacak hassas verilerin anlamsal olarak yakınlıkları dikkate alınmamıştır.

t −yakınlık yaklaşımı aşağıda belirtilen üç faz üzerine geliştirilmiştir:

1. $\boldsymbol{\beta_0}$: Veri kümesi yayınlanmadan önce, saldırı gerçekleştirecek kişinin geçmiş bilgisi.

2. $\boldsymbol{\beta_1}$: Veri kümesi yarı tanımlayıcılar olmadan yayınlandıktan sonra, saldırı gerçekleştirecek kişinin verinin dağılımı ile ilgili sahip olduğu bilgi.

3. $\boldsymbol{\beta_2}$: Veri kümesi yarı tanımlayıcılar ile birlikte yayınlandıktan sonra, saldırı gerçekleştirecek kişinin veri hakkındaki bilgisi.

$\ell -$ çeşitlilik yaklaşımı β_0 ve β_2 arasındaki farkı minimum seviyeye indirmeyi hedeflerken, t −yakınlık yaklaşımı β_1 ve β_2 arasındaki farkı t eşiğinin altında tutmaya çalışmaktadır. Yayınlanan veri kümesi içerisindeki bütün QI-gruplarında bu fark t eşiğinin altında ise, yayınlanan veri kümesi t −yakınlık yöntemine ait gereksinimlerin sağladığı anlamına gelmektedir.

Bu problem çözmek üzere geliştirilmiş (c, k) −güvenliği (Martin, 2007) ve (α, k) −anonimliği (Wong, 2006) gibi başka veri mahremiyeti prensipleri de önerilmiş durumdadır (Sun, 2011), (Chester, 2011), (Brickell, 2008).

Yukarıda bahsedilen anonimleştirme işlemlerinde genelleştirme işlemi bir veri kümesi üzerinde iki farklı katmanda yapılabilir:

- **Global kodlama (*global recoding*):** Bu kapsamda, bir kayıt üzerinde özniteliğin değeri için yapılan değişiklik, bütün veri kümesi içerisinde o değere sahip tüm kayıtlar için geçerlidir.
- **Lokal kodlama (*local recoding*):** Bu kapsamda, bir kayıt üzerinde özniteliğin değeri için yapılan değişiklik, sadece kaydın bulunduğu QI-grup içerisindeki kayıtlar için geçerlidir.

2.2 Diferansiyel Mahremiyet

Diferansiyel mahremiyet (Dwork, 2008), mahremiyet kavramının matematiksel tanımıdır. Bir veri kümesine ait istatistiksel bilgilerin analiz edildiğinde, herhangi bir kişinin veri kümesi içerisinde bulunup bulunmadığının anlaşılamamasıdır. Diferansiyel mahremiyet algoritmalarında, veri kümesine dahil olan veya ayrılan bir kayıt ile sorgu sonuçları neredeyse hiç değişmemektedir. Bu durum, veri kümesi içerisinden bireysel bir veri sızdırılamadığının bir kanıtıdır. Ayrıca veri kümeleri

üzerinde çalıştırılacak sorguların doğruluğunu ve kullanılabilirliğini maksimum seviyede tutmayı hedeflemektedir.

2.3 Anonim Verinin Kullanılabilirliği

Veri mahremiyeti çalışmaları sonucunda kullanılabilir bir veri kümesi üretebilmek bu alanda önemli problemlerden birisi olarak göze çarpmaktadır. Bölüm 2'de belirtilen çözümler ve diğer birçok veri mahremiyetinin korunması için geliştirilen çözümlerde, üretilen anonim verinin kullanılabilirliğini ölçmek için farklı metrikler kullanılmıştır. Bu metrikler şu şekilde özetlenebilir:

- Bilgi kaybı (*information loss*) (Bertino, 2005),
- Minimum bozulma (*minimal distortion*) (Sweeney, 2002),
- Ayırt edilebilirlik metriği (*discernibility metric*) (Li, 2008)
- Sınıflandırma metriği (*classification metric*) (Iyengar, 2002),
- Bilgi dengeleme metriği (*information trade-off metric*) (Fung, 2005),
- Model doğruluğu ve sorgu kalitesi (*model accuracy and query quality*) (Xu, 2006),
- Normalleştirilmiş QI-grup metriği (*normalized equivalence class metric*) (Li, 2008)

Veri mahremiyeti çalışmaları kapsamında sunulan çözümlerin başarısı yukarıda belirtilen metriklere göre değerlendirilmekte ve önerilen yaklaşımlar kendi içlerinde bu metriklere göre karşılaştırılmaktadırlar. Bu tez kapsamında geliştirilen çözümlerde bilgi kaybı (Bertino, 2005) metriği kullanılmıştır. Anonimleştirilen bir veri kümesi için bilgi kaybı Eşitlik (2.1) kullanılarak hesaplanmaktadır.

$$IL = \frac{\sum_{i=1}^{n} InfoLoss(t_i', t_i)}{n} \tag{2.1}$$

burada, n anonimleştirilen toplam kayıt sayısıdır, t_i ise i'nci sıradaki kaydı, t_i' ise anonim halini ifade etmektedir.

$$InfoLoss = \frac{\sum_{j=1}^{m} AttInfoLoss(t_i'(j))}{m} \tag{2.2}$$

burada, m bir kayıt içerisindeki yarı tanımlayıcı öznitelik sayısını temsil ederken, $t_i'(j)$ ile t_i' içerisinde bulunan j'nci öznitelik ifade edilmiştir.

$AttInfoLoss$ fonksiyonu nümerik ve kategorik öznitelikler için farklı formüller kullanılarak hesaplanmaktadır. Bu metriğin kullanılabilmesi için kategorik her bir öznitelik için bir taksonomi ağacı gerekmektedir. Nümerik öznitelikler için $AttInfoLoss$ Eşitlik (2.3) ile hesaplanır.

$$AttInfoLoss = \frac{u - l}{U - L} \tag{2.3}$$

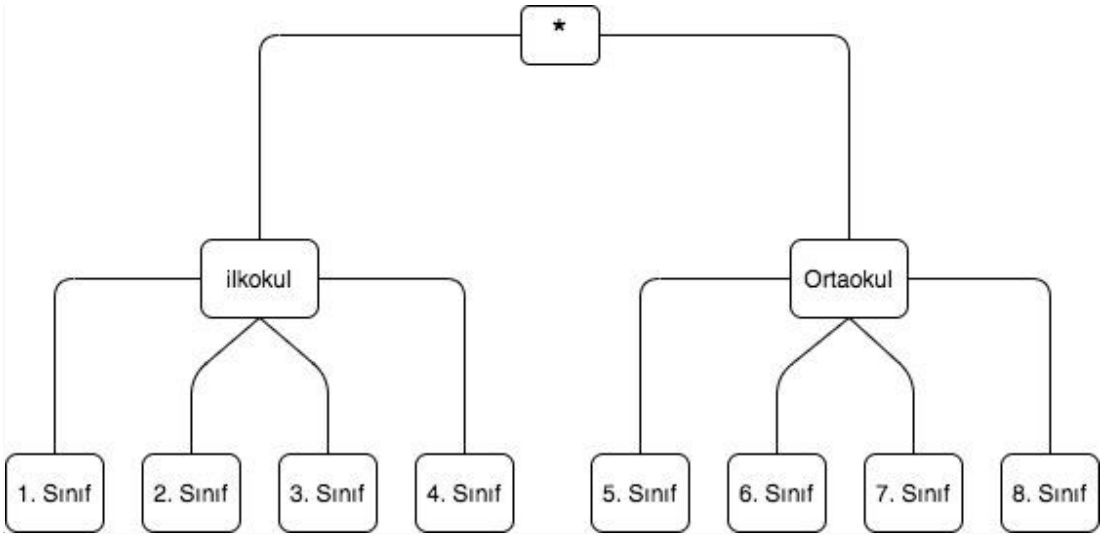
burada, u ve l ile anonimleştirilen QI-grup içerisinde ilgili özniteliğin üst ve alt sınır değerleri ifade edilirken, U ve L ise özniteliğin tanım kümesi içerisindeki maksimum ve minimum değerini temsil etmektedir. $AttInfoLoss$, kategorik değerler için Eşitlik (2.4) kullanılarak hesaplanmaktadır.

$$AttInfoLoss = \frac{|R_v| - 1}{|R| - 1} \tag{2.4}$$

burada, ilgili özniteliğin değerinin taksonomi ağacında bulunduğu düğümün altındaki yaprak düğümler kümesi R_v ile ifade edilmiştir. R ise ilgili özniteliğin tanım kümesindeki bütün değerleri temsil etmektedir.

Nümerik değerler genelleşme sonrası bir sayı aralığı ile ifade edilir. Örneğin Çizelge 2.1'de 3. kaydın *Yaş* değeri *12* iken, genelleşme işleminden sonra bir aralık ile değiştirilip *[12-13]* olmuştur. Diğer bir taraftan, bir kategorik öznitelik olan *Eğitim* ise, 5. kayıt için *11. Sınıf* iken, genelleştirme işlemi sonrası yine bir kategorik değer olan *Lise* ile değiştirilmiştir.

Anonimleştirme işlemi gerçekleştirilirken, birçok çalışmada kategorik değerler içeren öznitelikler için anonimleştirme işleminden önce taksonomi ağaçlarının hazırlanması gerekmektedir. Bu taksonomi ağaçları içerisinde bulunan düğüm değerleri aşağıdan yukarıya doğru gidildikçe genelleşmektedir. Bir özniteliğe ait taksonomi ağacının yaprak düğümleri, özniteliğin tanım kümesi içerisindeki değerlerden oluşmaktadır. Örnek bir taksonomi ağacı Şekil 2.1'de verilmiştir.



Şekil 2.1 Eğitim özniteliği için örnek bir taksonomi ağacı.

3. BÜYÜK VERİ MAHREMİYETİ

Bu bölüm içerisinde, büyük verinin tanımı, getirdiği zorluklar, geleneksel anonimleştirme yöntemlerinin neden büyük veri üzerinde uygulanamadığı, literatürde bulunan büyük veri için geliştirilmiş anonimleştirme çalışmaları, tez kapsamında önerilecek yöntem ve bu yöntem kullanılarak gerçekleştirilen deneyler ve sonuçları bulunmaktadır.

3.1 Büyük Veri (*Big Data*)

Büyük veri yeni bir kavram olmasına karşın, 1960'lı ve 1970'li yıllarda ilk veri merkezlerinin kurulması ve ilişkisel veri tabanlarının ortaya çıkışı ile birlikte büyük veri (*large data*) üzerine çalışmalar başlamıştır (Url-12). Büyük veri, üzerinde herkesin mutabık olduğu bir tanıma sahip olmayan soyut bir kavramdır. Araştırmacıların ve sektörde bulunan firmaların büyük bir kısmının desteklediği tanım Doug Laney (Laney, 2001) tarafından büyük verinin karakteristiğini göstermek için kullanılan 3V özelliğidir. Büyük veriye ait üç özellik şunlardır:

- Hacim (***V**olume*): Hacim ile üretilen ve toplanan veri miktarı kastedilmektedir.
- Hız (***V**elocity*): Verinin üretilme hızını ifade etmektedir.
- Çeşitlilik (***V**ariety*): Farklı yapısal türlere sahip veriler (yapısal, yapısal olmayan ve yarı yapısal) kastedilmektedir.

En çok desteklenen tanım bu olmakla beraber, faklı tanımlarda yapılmıştır. Örneğin bir büyük veri aracı olarak bilinen Apache Hadoop içerisinde büyük veri şöyle tanımlanmaktadır (Chen, 2014): "Yönetimi, işlenmesi ve toplanması kişisel bilgisayarlar ile kabul edilebilir bir zamanda yapılamayan ölçekteki veriler". Ayrıca önde gelen pazar analizi firmalarından biri olan IDC tarafından düzenlenen bir raporda büyük verinin 3V'si üzerine *Değer* (*value*) özelliği de eklenerek karakteristiği 4V ile özetlenmiştir (Chen, 2014).

Büyük verinin yarattığı pazar payı da oldukça büyüktür. Birleşik Devletler'de büyük verinin 2020 yılı için pazar payı 138.9 milyar dolar iken, 2025 yılı pazar payının 229.4 milyar dolar olması beklenmektedir (Url-11).

Günümüzde üretilen veri miktarının ciddi boyutlara ulaşması büyük veri kavramının önemini artmasına önayak olmuştur. Örneğin 2012 yılı itibariyle son iki yılda üretilen veri miktarı o güne kadar üretilen toplam verinin %90'ını oluşturmaktadır (Singh, 2012). 2025 yılında her gün 463 eksabayt veri transferi yapılacağı tahmin ediliyor (Chiusano, 2019). Bu ölçekte üretilen verinin büyük bir kısmı, sensör verileri ve kullanıcı sayısı her gün artış gösteren sosyal medya verilerinden oluşmaktadır (Sopaoglu, 2017). Statista tarafından açıklanan 2020 yılında en çok kullanıcıya sahip ilk 5 sosyal medya platformu ve kullanıcı sayıları Çizelge 3.1'de gösterilmiştir (Url-15).

Çizelge 3.1 2020 yılı için sosyal medya kullanıcı sayıları.

Sosyal Medya Platformu	**Kullanıcı Sayısı**
Facebook	2.4 milyar
Youtube	2 milyar
WhatsApp	1.6 milyar
WeChat	1.1 milyar
Instagram	1 milyar

Hadjar tarafından yapılan çalışmada (Hadjar, 2019), bazı sosyal medya platformlarında üretilen veri miktarları ile ilgili de somut rakamlar verilmiştir:

- Her dakika Youtube'a yüzlerce saatlik video yükleniyor.
- Her gün Twiter'da 400 milyon mesaj paylaşılıyor.
- Her gün 4.75 milyar multimedya içeriği Facebook'ta paylaşılıyor.

Üretilen veri miktarının artış göstermesinin önemli bir diğer sebebi ise nesnelerin interneti (IoT) teknolojisinin gelişmesidir. IDC 2025 yılı için 41.6 milyar IoT cihazın günde 79.4 zettabayt (ZB) veri üreteceğini tahmin etmektedir (Wangyal, 2020).

Bu ölçekte üretilen verinin birçok kullanım alanı da bulunmaktadır. Başlıca kullanım alanları Çizelge 3.2'de verilmiştir (Url-12).

Çizelge 3.2 Büyük veri için örnek kullanım alanları.

Kullanım Alanı	Açıklama
Ürün geliştirme süreçleri	Firmalar yeni ürünlerine karar verirken, geçmiş ve var olan ürünlerinin özelliklerini ve elde edilen ticari başarıyı dikkate alarak modelledikleri sistemleri kullanırlar.
Kestirimci bakım (*predictive maintenance*) uygulamaları	Üretilen büyük hacimli veride önemli bir paya sahip olan sensör verileri, özellikle mekanik yapıları bulunan fabrika ve firmalardan toplanan veriler ile sistemde önceden oluşacak problemleri tespit etmek için analiz edilmektedir.
Müşteri deneyimini anlama	Şirketler, ürünleri ile ilgili sosyal medya yorumları, web sayfası ziyaretleri ve gelen çağrılar gibi farklı kaynaklardan topladıkları verileri anlamlandırarak müşteri memnuniyetini arttırmaya çalışmaktadırlar.
Dolandırıcılık tespiti (*fraud detection*)	Müşteri hareketlerinden toplanan büyük hacimli veriden çıkartılan desenler ile şüpheli hareketlerin tespiti de sağlanabilmektedir.

Büyük veri heterojen yapısı, ölçeklenebilir sistem ihtiyacı, karmaşıklığı ve mahremiyet problemleri nedeniyle verinin toplanması, görselleştirilmesi, modellenmesi ve analiz edilmesi için geliştirilmiş birçok geleneksel yöntemin kullanılabilirliğini sınırlamıştır. Belirtilen bu problemler, büyük veri ile ilgili operasyonların yapılabilmesi için birçok yeni teknolojinin gelişmesine de vesile olmuştur. Çizelge 3.3'te büyük veri kavramı üzerine geliştirilmiş popüler bazı araçlar sunulmaktadır.

Geliştirilen bu araçların da sağladığı kolaylıklar ile şirketler, araştırma enstitüleri, kamu kurum ve kuruluşları ellerinde bulunan verileri analiz ederek alacakları kararlarda kendilerine yardımcı olabilecek sistemler kurmak için çalışmalar gerçekleştirmektedirler. Dünyanın ICT (bilgi ve iletişim teknolojileri) alanında önde gelen ülkeleri, operasyonel etkinliklerini, ekonomik büyüme oranlarını ve halkın refah düzeyini arttırabilmek için büyük veri uygulamaları geliştirme yönünde ilk adımları attılar (Kim, Chung, 2014).

Çizelge 3.3 Büyük veri özelinde geliştirilmiş bazı teknolojiler.

Teknoloji	**Açıklama**
Apache Hadoop	Büyük veri için geliştirilmiş ilk uygulamalardan birisidir. Açık kaynak kodlu Apache Hadoop verinin dağıtık bir şekilde tutulmasına ve işlenmesine olanak sağlamaktadır (Url-7).
Apache Storm	Apache Storm açık kaynak kodlu olup, gerçek zamanlı akan veri üzerinde hesaplama olanağı sunar (Url-17).
Apache Spark	Büyük verinin dağıtık bir şekilde işlenmesini hedefleyen açık kaynak kodlu bir projedir (Url-14).
Apache Cassandra	Çok büyük miktarda veriyi depolamak için tasarlanmış, açık kaynak kodlu dağıtık bir NoSQL veri tabanı yönetim sistemidir (Url-4).
Statwing	Büyük veri analizi için geliştirilmiş bir araçtır (Url-16).
Open Refine	Büyük verinin temizlenmesi, verinin bir formattan başka bir formata dönüşümüne olanak sağlayan bir büyük veri aracıdır (Url-13).

Gelişen analiz yöntemleri ile beraber, elimizde bulunan sağlık verisi, sosyal medya verisi ya da e-ticaret verisi gibi büyük veri kaynaklarını kullanarak içlerinden anlamlı sayılabilecek sonuçlar çıkartabiliriz. Fakat bu veriler çoğu zaman kişiler ya da kurumlar hakkında hassas sayılabilecek veriler içermektedirler. Bugün ülkemizde geçerli olan Kişisel Verileri Koruma Kanunu (KVKK) ile birlikte, kişi ya da kurumlar için hassas olabilecek veriler de koruma altına alınmıştır (Url-10).

Büyük veri için verinin mahremiyetinin sağlanması zorlu bir problem olarak karşımıza çıkmaktadır. Bu tez kapsamında büyük veri mahremiyetini korumak için geliştirilmiş yöntem Bölüm 3.2'de sunulmuştur.

3.2 Büyük Veri Mahremiyeti

Bu bölümde, büyük verinin mahremiyeti için geliştirilmiş anonimleştirme tabanlı yöntemlerden bahsedilecektir ve sonrasında bu tez kapsamında geliştirilen yöntem açıklanacak ve yapılan deneyler sonucu elde edilen bulgular sunulacaktır.

Veri mahremiyetinin korunması ile ilgili birçok yöntem geliştirilmiştir. Geliştirilen yöntemlerin bir kısmı Bölüm 2'de anlatılmıştır. Anonimleştirilmesi gereken veri miktarı terabayt hatta petabayt seviyelerine çıkabilmektedir. Bu boyutta bir verinin geleneksel anonimleştirme yöntemleri ve kişisel bilgisayarlar ya da geleneksel sunucular kullanılarak anonimleştirilmesi mümkün değildir. Bu durum daha etkili ve ölçeklenebilir yöntemler geliştirilmesi ihtiyacını beraberinde getirmiştir. Büyük verinin mahremiyeti ile ilgili geliştirilen birçok çalışmada dağıtık bir şekilde verinin işlenmesi ile ilgili çözümler sunulmaktadır.

Bulut bilişim teknolojisinin gelişmesi de büyük verinin mahremiyeti için yeni yöntemler geliştirilmesini hızlandırmıştır. Bulut bilişim teknolojisinin gelişmesi araştırmalar ve bilgi teknolojileri üzerinde de ciddi bir etki yarattı (Hayes, 2008), (Wang, 2011). Bulut bilişim teknolojisinin kullanıcılarına yüksek miktarda kaynak (CPU, RAM ve vb.) sağlayabilmesi birçok IT firmasını sunucu satın almak ve onların üzerinde kurulumlar ile uğraşmaktan ziyade, bulut hizmet sağlayıcılarında hizmet kiralama modeline yönlendirmiştir (Agmon, 2014). Fakat birçok potansiyel bulut müşterisi hala bulutta tutulacak verinin güvenliği ve mahremiyeti ile ilgili korkularından dolayı bu teknolojiye geçmek konusunda kararsızdırlar (Zhang X., 2013), (Sadiku, 2014). Büyük verinin mahremiyetinin sağlanması bir diğer açıdan bulut teknolojisine insanların güvenini arttırmak için de önem kazanmıştır.

3.2.1 Büyük veri mahremiyeti için geliştirilen yöntemler

Büyük veri özelinde geliştirilen ilk çalışmalar aslında bir kişisel bilgisayar üzerinde işlem yapacak büyüklükte veriler (*large data*) için geliştirilmiş yöntemlerin üzerine kurgulanmıştır. $k-$ anonimleştirme yöntemi ve bu prensip üzerine geliştirilmiş diğer yöntemlerin çözmeye çalıştığı problem NP-Hard bir problemdir (Meyerson, 2004). Dolayısıyla çeşitli sezgisel yaklaşımlar ile iyileştirmeler ve optimuma yakın çözümler elde edilmeye çalışılmaktadır. Bu çalışmaların en önemli iki örneği TDS (Fung, 2005) ve BUG (Wang, 2004) yöntemleridir. Bu çalışmalar algoritmaları içerisinde yaptıkları çeşitli optimizasyonlar ile kişisel bir bilgisayar üzerinde etkili bir şekilde

anonimleştirme işleminin yapılmasına olanak sağlamışlardır. (LeFevre, 2007) çalışmasında ise Mondrian algoritmasının büyük veri (*large data*) kümeleri için çalışabilecek şekilde güncellemiştir.

Verinin miktarının artması ve performans problemleri nedeniyle verinin bir bilgisayar kümesine dağıtılarak işlenmesi temelinde çözümler de geliştirilmiştir. Jiang tarafından geliştirilen çalışmada (Jiang, 2006) veri kümesini oluşturan öznitelik grubunun iki farklı kaynaktan geldiği ve bu iki kaynaktan gelen verinin bir merkezde birleştirildiği ve verinin burada anonimleştirildiği bir senaryo için geliştirilen çözümde performansı arttırabilmek için dağıtık mimari ile çalışan bir çözüm önerilmiştir.

Hadoop MapReduce teknolojisinin hayatımıza girmesi ve popülerleşmesi ile büyük verinin anonimleştirilmesi üzerine yapılan çalışmalar da artış göstermiştir. Hadoop MapReduce, Google tarafından geliştirilmiş bir MapReduce çerçevesidir (*framework*). Bu çerçeve içerisinde iki temel fonksiyon bulundurmaktadır bunlar: *map* ve *reduce* fonksiyonlarıdır. Bir Hadoop MapReduce işi girdi olarak anahtar-değer (*key-value*) çifti alır. Bu çiftler üzerinde işlem yapılmak istendiğinde *map* fonksiyonu her bir kayıt için çağırılır ve bu fonksiyon çıktı olarak yeni küçük veri parçaları oluşturur. *Reduce* fonksiyonu ise *map* fonksiyonu ile üretilen parçaların toplanması ve yeni anahtar-değer çiftlerinin oluşturulmasından sorumludur. Hadoop MapReduce kullanıcılarının sadece gerekli *map* ve *reduce* fonksiyonlarını belirtmeleri yeterlidir. Paralelleştirme ve hata durumları bu çerçeve tarafından kontrol edilmektedir. Hadoop MapReduce ile geliştirilmiş çözümlerde veri kümeleri Hadoop Dağıtık Dosya Sisteminde (HDFS) tutulmaktadır. Hadoop MapReduce veri okuma ve yazma işlemlerini disk üzerine yaptığı için performans olarak yeterince hızlı değildir.

Büyük veri özelinde geliştirilen çalışmaları detaylandırmadan önce, birçok çalışmanın ve tez kapsamında önerilecek çözümün temelini oluşturan TDS yaklaşımı öncelikli olarak anlatılacaktır.

TDS Yaklaşımının Detayları

Birçok genelleştirme tabanlı anonimizasyon çalışmasında olduğu gibi, TDS yaklaşımı da anonimleştirme işlemine başlanmadan önce kategorik öznitelikler için taksonomi ağaçlarının tanımlanmış olmasını beklemektedir. Her bir yarı tanımlayıcı için belirlenmiş olan bu ağaçların üzerinde TDS algoritması yukarıdan aşağıya eş zamanlı arama yapmaktadır.

Algoritmanın ilk adımından, veri kümesinden bir kaydın doğrudan tanınmasına neden olacak öznitelikler çıkartılır. Aynı yarı tanımlayıcı değerlerine sahip kayıtlar bir araya toplanmaktadır. Çizelge 3.4'te bu işlem ile ilgili bir örnek veri kümesi verilmiştir. Bu veri kümesi içerisinde *Yaş*, *Eğitim* ve *Cinsiyet* yarı tanımlayıcı özniteliklerken, *Gelir* ise hassas veri olarak belirlenmiştir. *Gelir* özniteliğinde yıllık geliri 50 bin TL altı olan kişiler Hayır (H) değeri ile gösterilirken, 50 bin TL üzeri gelir ise Evet (E) değeri ile ifade edilmiştir. Gösterilen örnek veri kümesi k=5 için anonim değildir.

Algoritmanın ikinci adımı genelleştirme işlemidir. Bütün veri kümesinin anonimizasyon seviyesi AL ile ifade edilmektedir. Anonimizasyon seviyesi içerisinde yarı tanımlayıcı öznitelikler için önceden tanımlanmış taksonomi ağaçları bulunmaktadır. Başlangıçta veri kümesi için anonimizasyon seviyesi taksonomi ağaçlarının kök değerleridir ve veri kümesi içerisindeki kayıtların yeni değerlerine anonimizasyonun seviyesine bakılarak karar verilmektedir. Dolayısıyla kayıtlar bütün öznitelikler için en genel değerleri olan taksonomi ağaçlarının kök düğümleri ile başlarlar. Anonimizasyon seviyesine bağlı olarak en az eleman bulunduran QI-grup, veri kümesi için mevcut k değerini belirler ve bu $mevcutK$ olarak ifade edilmektedir. En üst seviyeden başlayan anonimizasyon seviyesi her bir adımda özelleştirilecektir. Bu işlem $mevcutK < k$ olana kadar devam etmektedir. $mevcutK > k$ durumu, anonimizasyon seviyesinin çok genel olduğu anlamına gelmektedir.

Çizelge 3.4 Ön işlemeden sonra oluşan örnek veri kümesi.

Yaş	**Eğitim**	**Cinsiyet**	**Gelir (Kişi)**	**Kayıt Sayısı**
35	Üniversite	Erkek	6E2H	8
42	Yüksek Lisans	Kadın	1E4H	5
38	Üniversite	Erkek	0E2H	2
24	Lisans	Kadın	3E1H	4
30	Lisans	Kadın	2E2H	4
40	Doktora	Erkek	5E0H	5

Özelleştirmenin devamı için, bir taksonomi ağacı $v \in AL$ seçilir ve anonimizasyon seviyesi ilgili öznitelik için çocuk düğümleri ile değiştirilir. Değişme işlemi, seçilen taksonomi ağacının kök değeri AL'den çıkartılır ve alt ağaçların kök değerleri AL'e eklenir. Daha iyi açıklayabilmek için özelleştirme işlemi yapılacak olan düğümün altında x adet çocuk düğüm olduğunu varsayarsak, AL içerisinden mevcut ağaç

çıkartılacak ve yerine x adet yeni ağaç eklenecektir dolayısıyla başlangıçta $|AL|$ yarı tanımlayıcıların sayısına eşitken, her bir özelleştirme işlem sonrası boyutu büyümektedir. Taksonomi ağacı altında çocuk düğümler olduğu sürece o öznitelik için özelleştirme işlemi devam edebilir.

TDS yönteminin her bir iterasyonunda, özelleştirilebilecek aday sayısı $|AL|$ kadardır ve bu iterasyonlarda bir aday özelleştirilmek üzere seçilir. Seçim işlemi iki parametreye göre yapılmaktadır (i) bilgi kazancı (*information gain*) , (ii) anonimlik kaybı (*anonymity loss*). Bu parametreler bütün adaylar için hesaplanmaktadır.

$$InfoGain(v) = I(R_v) - \sum_{c \in child\ (v)} \frac{|R_{v=c}|}{R_v} I(R_{v=c}) \tag{3.1}$$

burada, veri kümesi içerisinde öznitelik değeri v taksonomi ağacının altında bulunan kayıtlar kümesi R_v ile ifade edilmektedir. $R_{v=c}$ ise v ağacı kullanılarak yapılacak olan özelleştirmeden sonra c değerine sahip kayıt kümesini göstermektedir. $I(R_v)$, R_v kümesinin entropi değeri anlamına gelmektedir. Entropi hesaplanırken Eşitlik (3.2) kullanılmaktadır.

$$I(R_v) = -\sum_{sv \in Hassas\ Veriler} \frac{|R_v, sv|}{|R_v|} \times \log_2 \frac{|R_v, sv|}{|R_v|} \tag{3.2}$$

burada $|R_v, sv|$ ile R_v içerisinde sv hassas değerine sahip kayıtların toplam sayısı ifade edilmektedir.

Bir diğer parametre olan anonimlik kaybı ise Eşitlik (3.3) kullanılarak hesaplanmaktadır.

$$AnonymityLoss = A(v) - A'(v) \tag{3.3}$$

burada, özelleştirme işlemi yapılmadan önce anonimlik seviyesi $A(v) = mevcutK$ ile gösterilirken, $A'(v) = mevcut'K$ ise özelleştirme sonrasındaki anonimlik seviyesini göstermektedir.

Anonimleştirilen veri kümelerinin kullanılabilirliği düşmektedir. Dolayısıyla kullanılabilirlik ve anonimlik arasında negatif korelasyon vardır. Bu iki metriği dengeleyebilmek için bilgi kazanımı ve anonimlik kaybı metrikleri kullanılarak yeni bir skor hesaplanmakta ve en yüksek skoru üreten $v \in AL$ taksonomi ağacı için

özelleştirme işlemi gerçekleştirilmektedir. Bu skorun hesaplamasında Eşitlik (3.4) kullanılmaktadır.

$$Score(v) = \begin{cases} \dfrac{InfoGain(v)}{AnonymityLoss(v)}, & eğer\ AnonymityLoss(v)\ \neq 0 \\ InfoGain(v), & aksi\ halde \end{cases} \quad (3.4)$$

Ölçeklenebilir İki Fazlı Yukarıdan Aşağıya Özelleştirme: Hadoop MapReduce kullanılarak geliştirilen yöntem (Zhang X., 2013), yukarıdan aşağıya özelleştirme algoritmasını temel almaktadır. TDS, anonimleştirme yaklaşımının ölçeklenebilirliğini arttırmak için geliştirilmiştir. TDS yöntemi indeksleme veri yapısı kullanan ölçeklenebilir bir algoritma olarak sunulmuştur. Fakat, bu büyük ölçekli veriler için geçerli değildir. Çünkü bu yaklaşım bütün verinin hafızaya sığdığı varsayımı ile geliştirilmiştir. Ayrıca indeksleme yapısından dolayı ekstra hafıza tüketmektedir. Bu nedenle büyük hacimli verilerde bu yaklaşım kullanılamamaktadır. İki fazlı TDS yöntemi bu eksiklikler üzerine geliştirilmiş bir çözümdür.

İki Fazlı TDS yaklaşımı aşağıdaki bileşenlerden oluşmaktadır:

1. *Verinin Parçalara Ayrılması*: Bu aşamada, veri kümesi p tane bölüme ayrılmaktadır. Veri kümesi içerisinde bulunan her bir kayıt için 1 ile p arasında rastgele bir sayı üretilir ve bu sayı her bir kaydın hangi kümede olacağını gösterir. Oluşturulan veri bölümlerinin, veri kümesi ile benzer dağılımda olması istendiğinden böyle bir yaklaşım uygulanmıştır.
2. *Anonimleştirilen Parçaların Birleştirilmesi*: Her bir parça için MRTDS (MapReduce tabanlı TDS) yaklaşımı anonimlik değeri k^I ile çalıştırılır. k^I değeri kullanıcı tarafından değil geliştirici tarafından anonimleştirme işlemi başlamadan önce belirlenen bir değerdir ve bu değer ($k^I > k$) koşulunu sağlamalıdır. Eşitlik (3.5) de belirtilen MRTDS fonksiyonu veri bölümünü, anonimlik parametresi k^I değerini ve başlangıç anonimizasyon seviyesi AL^0 parametre olarak almaktadır ve MRTDS yaklaşımının çalıştırıldığı ilgili bölüm için (D_i) yeni anonimizasyon seviyesi (AL_i) çıktı olarak üretilir.

$$MRTDS(D_i, k^I, AL^0) \rightarrow AL_i' \quad (3.5)$$

Her bir parça için üretilen yeni anonimizasyon seviyesi (AL_i') için birleştirme işlemi çalıştırılır ve bütün veri kümesi için tek bir anonimizasyon seviyesi elde edilir (AL^I). Birleştirme konservatif bir işlemdir dolayısıyla birleştirme işleminden sonra bölümler için anonimlik seviyesi azalmaz.

$$merge(AL_1', AL_2', \dots, AL_p') \rightarrow AL^I \tag{3.6}$$

3. *Verinin Özelleştirilmesi*: Bu aşamada, MRTDS fonksiyonu bütün veri kümesi üzerinde k değeri ve AL^I ile çalıştırılır ve yeni anonimizasyon seviyesi AL^* elde edilir.

$$MRTDS(D, k, AL^I) \rightarrow AL^* \tag{3.7}$$

AL^* değeri bulunduktan sonra algoritma veri kümesi içerisinde her bir kayıt için yarı tanımlayıcıların değerlerini anonimizasyon seviyesine göre değiştirir.

Yukarıdan Aşağıya Özelleştirme ve Aşağıdan Yukarı Genelleştirme Yöntemlerinin Kombinasyonu: TDS ve BUG yaklaşımları indeksleme veri yapısı kullanmaktadırlar. Bu veri yapısı büyük ölçekli verilerin anonimleştirilmesinde yeterli değildir. Burada da bir önceki yöntemde olduğu gibi Hadoop MapReduce kullanılarak ölçeklenebilir bir sistem oluşturulmuştur. TDS ve BUG yaklaşımları için farklı k değerleri ile testler yapılmış ve yapılan testler sonucunda k değeri büyük olduğu durumlarda TDS yaklaşımı daha iyi sonuç verirken, k değeri küçük olduğu durumlarda BUG yöntemi daha iyi sonuçlar vermektedir. Burada hibrit bir yaklaşım geliştirilmiştir (Zhang, 2013) ve sistemde belirlenen k değerine göre uygulanacak yönteme karar verilmektedir. TDS ve BUG yöntemleri MapReduce yaklaşımı kullanılarak yeniden implemente edildiği için hibrit yöntemin ölçeklenebilirliği sağlanmıştır. Elde edilen sonuçlara göre, var olan yöntemlerden istatistiksel olarak önemli derecede ölçeklenebilirlik açısından daha iyi sonuçlar vermektedir.

Büyük Veride Çok Boyutlu Anonimleştirme:

Çok boyutlu anonimleştirme uygulamalarında da ölçeklenebilirlik sorunu bulunmaktadır. Bu çalışma (Zhang X., 2013), ölçeklenebilir bir çok boyutlu anonimleştirme çözümü sağlıyor. Önceki belirtilen yöntemlerde olduğu gibi Hadoop MapReduce tabanlı bir çözüm geliştirilmiştir. Çok boyutlu veriler için anonimleştirme

yapılırken nümerik özniteliklerde medyan hesaplanmaktadır. Nümerik öznitelikler için medyan bulmak, çok boyutlu anonimleştirmede temel bir bileşendir. Bu bileşenin sisteme getirdiği yük, veri miktarının artması ile dolaylı olarak artmaktadır bu durum çalışmanın motivasyonu da bu problemdir. Medyan değeri Hadoop MapReduce yardımı ile dağıtık bir şekilde hesaplanarak ölçeklenebilir bir çözüm olarak sunulmuştur.

3.2.2 Apache Spark ile yukarıdan aşağıya özelleştirme

Bu bölümde tez kapsamında büyük verinin anonimleştirilmesi için geliştirilmiş yöntem açıklanmaktadır.

Yukarıdan aşağıya anonimleştirme işlemi için iki farklı çözüm anlatılmıştır. Birinci çözüm, kişisel bir bilgisayar üzerinde işlenebilecek ölçekte olan verinin anonimleştirilmesi üzerinedir. Diğer bir taraftan, ikinci çözümde büyük verinin anonimleştirilmesi için geliştirilmiş olan iki fazlı TDS yaklaşımıdır (Zhang X., 2013). Bu yaklaşım Hadoop MapReduce çerçevesi kullanılarak geliştirilmiştir. Bu yöntem büyük verinin ölçeklenebilir bir şekilde anonimleştirilmesini sağlamasına rağmen, Hadoop MapReduce'nin okuma/yazma (I/O) işlemlerini disk üzerinden yapıyor olması sistemin kullanılabilirliğini oldukça düşürmektedir. Bu çalışmanın motivasyonu iki fazlı TDS yaklaşımının I/O performansının düşük olmasından kaynaklı yöntemin kullanılabilirliğinin düşüklüğüdür. Bu problem popüler büyük veri işleme araçlarından birisi olan Apache Spark yardımı ile tez kapsamında çözüldü. TDS yaklaşımının büyük veri çalışmalarında etkili bir şekilde kullanılabilmesi için Apache Spark yardımı ile yeniden tasarlanmıştır. Apache Spark günümüzde geliştirilen birçok büyük veri projesinde aktif olarak kullanılmakta olan bir araçtır ve Hadoop MapReduce ile karşılaştırıldığında 100 kat daha hızlı olduğu iddia edilmektedir (Url-14). Bundan dolayı, tez kapsamında sunulan TDS yaklaşımı ölçeklenebilirlik açısından daha önce geliştirilen TDS yaklaşımlarına göre daha iyi sonuçlar vermektedir.

3.2.2.1 Apache Spark ile dağıtık TDS çözümü

Apache Spark verinin dağıtık bir şekilde işlenmesine ve üzerinde hesaplamalar yapılmasına olanak sağlayan bir büyük veri aracıdır. Hadoop MapReduce'nin eksiklerini ve performans problemlerini (Bu, 2010) çözmek için geliştirilmiş olan bu araç verinin RAM (*Random Access Memory*) üzerinde tutulması ve işlemlerin RAM

üzerinde yapılmasıyla Hadoop MapReduce'ye göre 100 kat daha hızlı çalışabilmektedir. Verinin belleğe yüklendikten sonra, Apache Spark platformu veri kümesi üzerinde sorguların tekrar tekrar çalıştırılmasına olanak sağlanmaktadır. Bu özellik Apache Spark'ın yinelemeli işlemler gerektiren büyük veri problemlerinde tercih edilen bir araç olmasını sağlamıştır. Dağıtılan veri Apache Spark tarafından tanımlanmış ve üzerinde paralel işlemler yapmaya olanak sağlayan RDD (*Resilient Distributed Datasets*) veri yapısında tutulmaktadır. Apache Spark içerisinde ana düğüm (*master node*) veriyi işçi düğümlere (*worker nodes*) dağıtır ve veri ile ilgili yapılacak bütün işlemler işçi düğümler üzerinde gerçekleştirilir.

Bu çalışmanın motivasyonu Apache Spark kullanarak daha verimli ve ölçeklenebilir bir anonimleştirme çözümü elde etmektir. Bizim çözümümüz TDS ve iki fazlı TDS yaklaşımlarının üzerine geliştirilmesine rağmen, implemantasyon açısından Hadoop tabanlı çözümden tamamen farklıdır.

Tez kapsamında geliştirilen çözümümüz ile ilgili iş akışı Şekil 3.1'de gösterilmiş ve kullanılan notasyon da Çizelge 3.5'te verilmiştir.

Algoritma girdi olarak veri kümesini ve anonimizasyon parametresi olan k değerini alır. Veri kümesi sürücü düğüm tarafından p parçaya ayrılır ve bu parçalar işçi düğümlere atanır. İşçi düğümler kendilerine atanan parçalar üzerinde gerekli hesaplamaları (R_p, R_c, (R_p, hv) ve (R_c, hv)) yapar. Bütün işçi düğümlerde yapılan hesaplamaların sonuçları sürücü düğümde toplanır. Böylece toplanan lokal değerler ile bütün veri kümesi için entropi, bilgi kazanımı ve anonimlik kaybı hesaplanır. Hafızada tek başına tutulamayacak büyüklükte olan veri kümesi için gereken işlemler dağıtık bir mimari ile hesaplanmıştır. Algoritmanın son adımında bilgi kazanımı anonimlik kaybı (IGPL) özelleştirilebilecek bütün adaylar için hesaplanır ve özelleştirme işlemi gerçekleştirilir. Eğer oluşan yeni durumda $mevcutK < k$ sağlanıyorsa son yapılan özelleştirme işlemi geri alınır ve veri kümesi yayınlanır. Eğer koşul sağlanmıyorsa, aynı işlem koşul sağlanana kadar tekrarlanır.

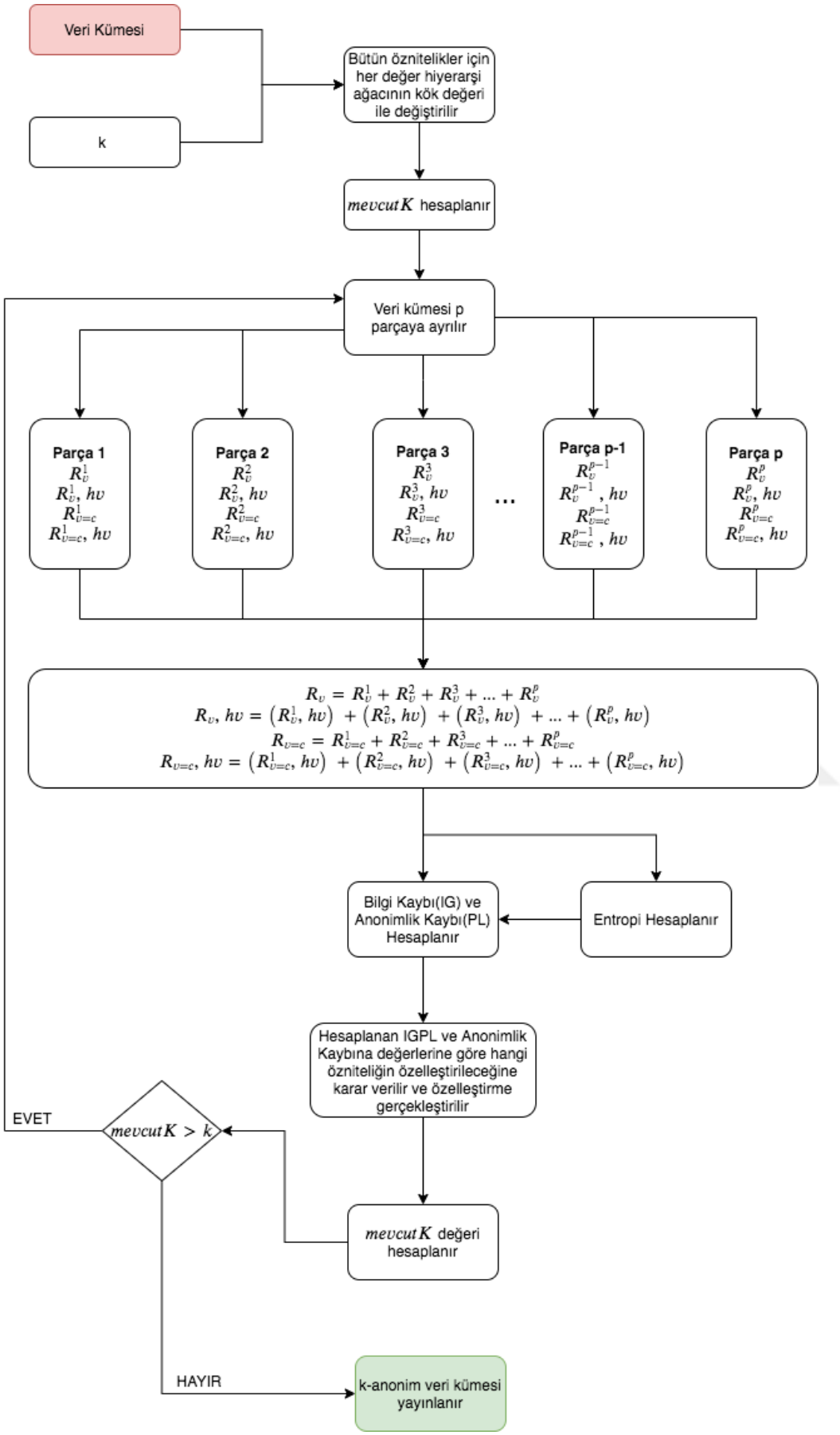
Çizelge 3.5 Dağıtık TDS yaklaşımında kullanılan notasyon.

Notasyon	Açıklama
$mevcutK$	Her bir iterasyonda sonrasında veri kümesi için hesaplanan k değeridir.
p	Veri kümesinin kaç parçaya ayrılacağı bilgisini belirtmektedir.
R_v	Özelleştirme işlemi uygulanacak v taksonomi ağacı altında bulunan kayıt kümesi
R_v, sv	Özelleştirme işlemi uygulanacak v taksonomi ağacı altında bulunan ve sv hassas değerine sahip kayıt kümesi
IG	Bilgi kazanımı
PL	Anonimlik kaybı
$IGPL$	Bilgi kazanımı anonimlik kaybı

3.2.3 Deneysel değerlendirme

Önerilen çalışmayı değerlendirmek için yapılan deneylerde ADULT (Url-1) veri kümesi kullanılmıştır. ADULT veri mahremiyeti çalışmalarında sıklıkla kullanılan bir veri kümesidir. Çizelge 3.6'da veri kümesinin bu çalışma özelinde kullanılan yarı tanımlayıcıları ve tanım kümelerinin büyüklükleri verilmiştir ve bu versiyonu bölüm kapsamında ADULT olarak ifade edilecektir.

Yapılan çalışmayı ölçeklenebilirlik açısında değerlendirebilmek için ADULT veri kümesi sırasıyla 10, 50, 100 ve 194 kat büyütülerek deneyler bu veri kümeleri üzerinde gerçekleştirilmiştir. Bu büyütme işlemi sonrası elde edilen veri kümelerinin boyutları sırasıyla 27 MB, 135 MB, 270 MB ve 500 MB olmuştur. Veri kümesi genişletme işlemi (Mohammed, 2010) çalışmasında belirtilen yönteme göre yapılmıştır.



Şekil 3.1 Önerilen çözüm yolu için iş akış şeması

Çizelge 3.6 ADULT veri kümesi içerisinde kullanılan yarı tanımlayıcılar.

Öznitelik	Tanım kümesinin büyüklüğü
Age	74
Education	16
Gender	2
Occupation	14
Race	5
Relationship	6
Marital status	7
Nation	41

Öznitelikler için ihtiyaç duyulan taksonomi ağaçları (Fung, 2005) çalışmasından alınmıştır.

Bu çalışma kapsamında gerçekleştirilen deneyler Çankaya Üniversitesi'nde bulunan 32 çekirdek ve 128 GB RAM kapasiteli bir bilgisayar kümesinde gerçekleştirilmiştir. Apache Spark için gerekli olan sürücü ve işçi düğümler Docker konteynerleri üzerinde koşturulmuştur. Farklı sayıda işçi konteynerler (1, 2, 4, 8 ve 16) ile deneyler gerçekleştirilmiştir. Her bir konteynere 2 GB RAM kapasitesi ve 2 GHz çalışma hızında bir çekirdekli işlemci verilmiştir. Sunulan yöntemin başarısını ölçmek için üç farklı değişkene bağlı sonuçlar değerlendirilmiştir:

1. İşçi düğüm sayısı
2. Veri kümesinin büyüklüğü
3. Anonimleştirme parametresi k

Gerçekleştirilen deneylerde anonimleştirme işlemine ait tamamlanma süreleri (milisaniye cinsinden) verimlilik ölçütü olarak kullanılmaktadır.

3.2.4 Deney sonuçları

Şekil 3.2, Şekil 3.3 ve Şekil 3.4'te farklı k değerleri (k=10, 50 ve 100) ve farklı boyutta veri kümeleri (27 MB, 135MB ve 270 MB) kullanılarak önerilen yöntemin verimliliği gösterilmektedir. Verilen sonuçlardan anlaşıldığı üzere, işçi düğüm sayısının artması her zaman çalışma süresini azaltmamaktadır. Bunun nedeni veriyi parçalara bölmenin getirdiği ek yüktür. Örneğin, tek işçi düğüm üzerinde veriyi parçalara ayırmak bazen

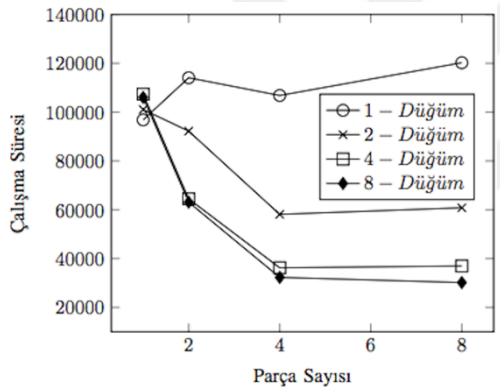
çalışma zamanını arttırmaktadır. Bununla birlikte, genel bir eğilim olarak çalışma zamanı, artan bölüm sayısı ve işçi düğüm sayısı ile önemli ölçüde düşer.

Şekil 3.5 ve Şekil 3.6'ya bakıldığında, farklı k değerleri ve sabit sayıda bölüm ve işçi sayısı ile yöntem çalıştırıldığında, çalışma zamanının bundan etkilendiği gözükmektedir. Bunun nedeni, mevcut anonimlik değerlerinin her yinelemede keskin bir şekilde azalması ve yineleme sayısının k = 10, 50, 100 değerleri üzerinde aşağı yukarı aynı olmasıdır.

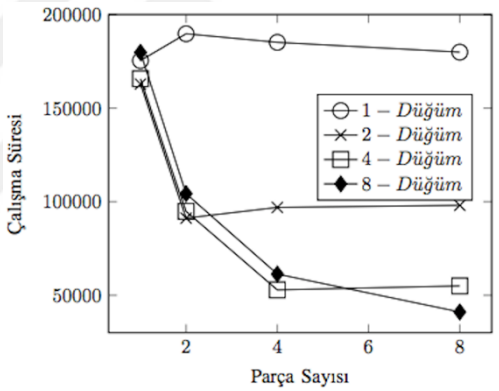
Yöntemin ölçeklenebilirliğini gösterebilmek için 500 MB boyutunda bir veri kümesi üzerinde de testler gerçekleştirilmiştir. Ölçeklenebilirlik ile ilgili yapılan deneylerin sonuçları Şekil 3.5 ve Şekil 3.6'da verilmiştir. Deneylerde bölüm sayısı ve işçi düğüm sayısı için aynı değerler kullanılmıştır. Yani deney kapsamında 2 bölüm için verilen sonuca bakıyorsanız, ilgili deneyde 2 işçi düğüm kullanılmaktadır. Öncesinde iddia edildiği gibi, bölüm sayısının ve işçi düğüm sayısının arttırılması özellikle büyük veri kümelerinde performans avantajı sağlamaktadır ve önemli ölçüde sonuçlar iyileşmektedir. TDS yaklaşımı 500 MB boyutlu bir veri kümesi üzerinde çalışamamaktadır (Fung, 2005), Apache Spark tabanlı geliştirdiğimiz yöntem daha büyük veri kümeleri için de umut vericidir.

3.3 Değerlendirmeler

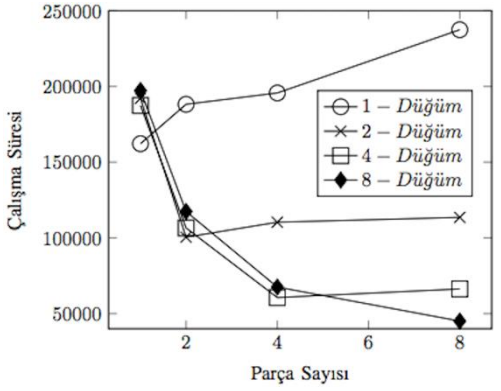
Büyük veri çağında yaşadığımız bu günlerde üretilen veri miktarı da üstel bir şekilde artış göstermektedir. Ölçeklenebilirliği sağlamak adına veri miktarının artması ile birlikte Apache Spark, Apache Hadoop gibi büyük veri araçlarının sayısı da hızla artmaktadır. Bu sebeple birçok algoritma ve uygulama Apache Hadoop, Apache Spark gibi büyük veri teknolojilerine uygun bir şekilde güncellenmektedir. Aksi takdirde bu uygulama ve algoritmalar pratikte kullanılamayacaklardır. Veri mahremiyeti için geliştirilmiş uygulamalar da bu durumdan etkilenmektedirler. Bu açıdan, geliştirilen yeni anonimleştirme yöntemlerinde bu problem dile getirilmektedir. Popüler anonimleştirme yöntemleri arasında olan TDS yaklaşımı Hadoop ile büyük veriler için uygulanabilir bir hale gelmesine rağmen, Apache Spark yardımı ile bu yaklaşım ölçeklenebilirlik açısından daha verimli bir hale getirilmiştir. Önerilen yöntem üzerinde gerçekleştirilen deney sonuçları incelendiğinde yüksek oranda ölçeklenebilir olduğu sonucuna ulaşılmaktadır.



(a) $k = 10$

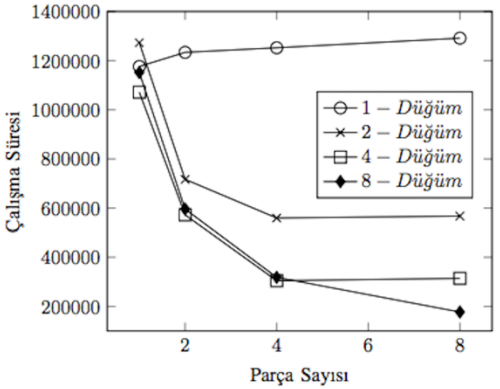


(b) $k = 50$

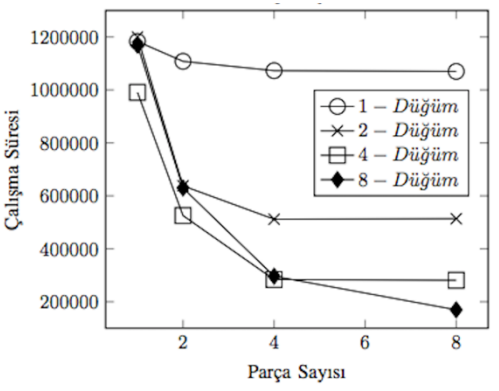


(c) $k = 100$

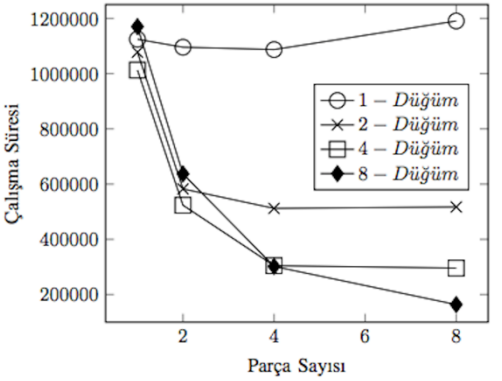
Şekil 3.2 27MB boyutunda olan veri kümesi üzerinde önerilen yöntemin verimi



(a) $k = 10$

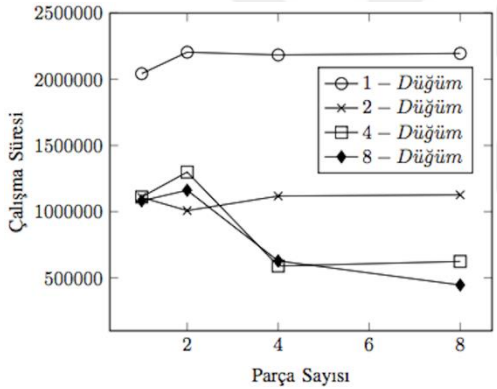


(b) $k = 50$

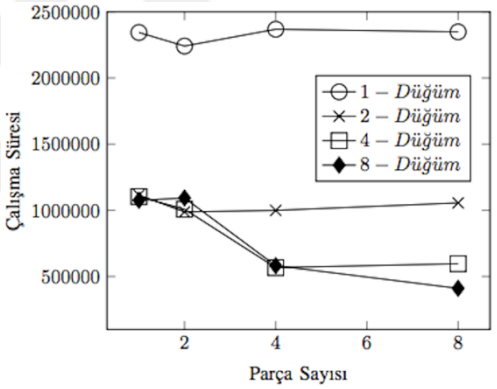


(c) $k = 100$

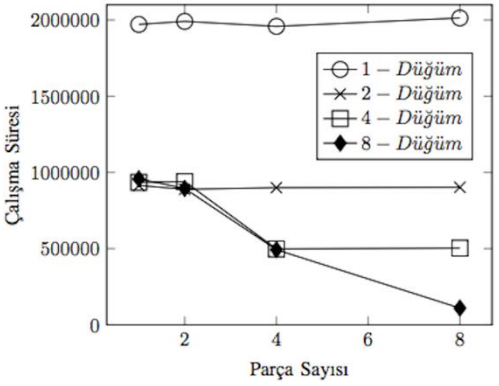
Şekil 3.3 135MB boyutunda olan veri kümesi üzerinde önerilen yöntemin verimi



(a) $k = 10$

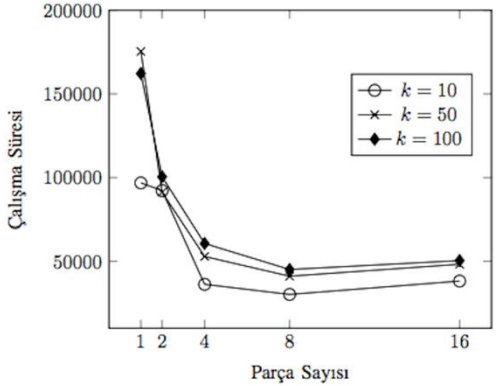


(a) $k = 50$

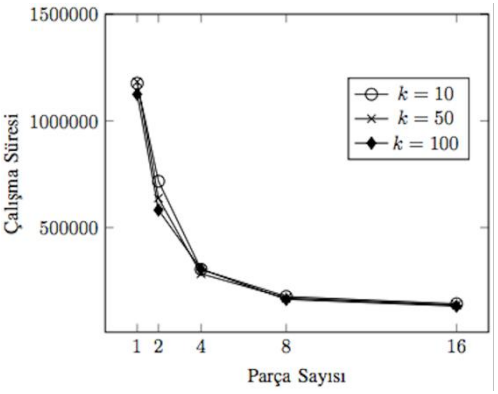


(a) $k = 100$

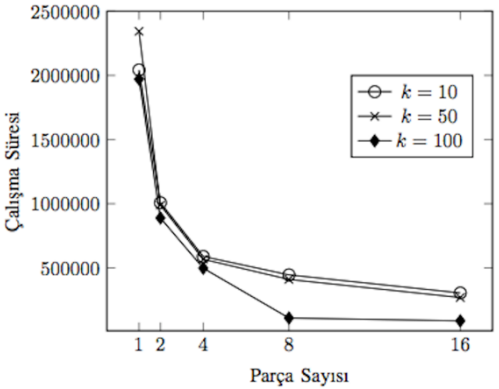
Şekil 3.4 270MB boyutunda olan veri kümesi üzerinde önerilen yöntemin verimi



(a) 27 MB

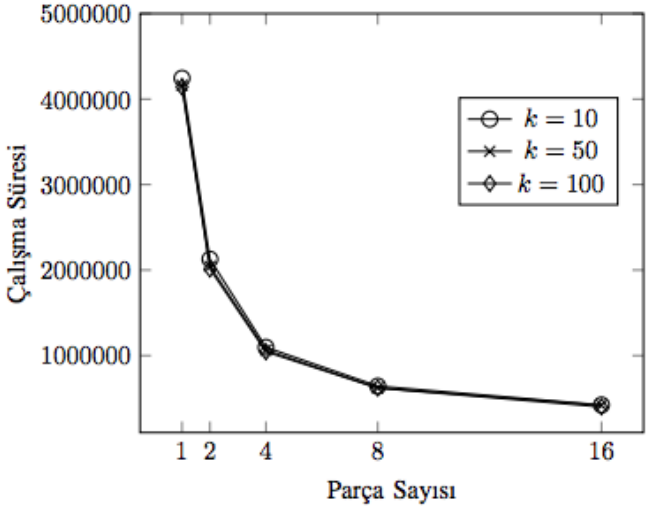


(b) 135 MB



(c) 270 MB

Şekil 3.5 27MB, 135 MB, 270MB'lık veri kümeleri üzerinde ölçeklenebilirlik deneyine ait sonuçlar



Şekil 3.6 500 MB'lık bir veri kümesi üzerinde ölçeklenebilirlik deney sonucu

4. AKAN VERİ MAHREMİYETİ

Bu bölümde öncelikle akan veri kavramının ne olduğu ve nerelerde kullanıldığı anlatılacaktır. Akan veri ve statik veri arasındaki farklar belirtilecektir. Akan verinin getirdiği zorluklar ile beraber gelişen akan veri teknolojileri vurgulanacaktır.
Akan veri mahremiyetinin sağlanması ile ilgili yapılmış çalışmalardan bahsedilerek, tez kapsamında geliştirilmiş olan UBDSA (*Utility Based Data Stream Anonymization*) yöntemi açıklanacaktır. UBDSA yöntemi kullanılarak yapılan deneylere ait sonuçlar sunulacak ve elde edilen sonuçlar literatürde var olan ve bilinen akan veri anonimleştirme yöntemleri ile karşılaştırılacaktır.

4.1 Akan Veri

Akan veri farklı kaynaklardan sürekli üretilen veriler olarak tanımlanabilir. Örneğin yazılımlar tarafından üretilen günlükler (*log*), banka işlemleri, telekomünikasyon verileri, sosyal medya üzerinden yapılan paylaşımlar, e-ticaret işlemleri ve birçok IoT verisi gibi veriler olabilir.

Akan verinin kullanım alanları oldukça geniştir. Aşağıda akan verinin kullanımları ile ilgili örnekler verilmiştir:

- E-ticaret siteleri üzerinde kullanıcıların tıklama eğilimleri takip edilerek, tıklama akışında anormal bir davranış gözlemlendiğinde uyarı üretilir (Wang, 2013), (Hofgesang, 2005).
- Haber kaynaklarından toplanacak tıklanma verileri ve bu verilerin demografik bilgiler ile zenginleştirilmesi sayesinde kitlelerin demografik bilgilerine uygun haberler sunulabilmektedir (De Bock, 2010).
- Mekanik bir sistemden toplanan sensör verileri sürekli olarak analiz edilir ve sistemde oluşabilecek hatalar önceden tespit edilebilmektedir (*predictive maintenance*) (Hashemian, 2010).

Akan veri ve statik veri kümeleri arasındaki farklar aşağıdaki gibidir (Babcock, 2002):

- Veriler çevrim içi olarak toplanır.
- Sistemin hangi verinin önce geleceği konusunda bir kontrolü yoktur.
- Akan verinin boyutu için bir sınır yoktur.
- Akan veri içerisinden bir kayıt ile ilgili işlem yapıldıktan sonra genellikle atılır.

Akan veri ve statik veri kümeleri arasında detaylı bir karşılaştırma Çizelge 4.1'de verilmiştir.

Çizelge 4.1 Statik veri ile akan verinin karşılaştırılması (Wares, 2019).

Geleneksel statik veri	**Akan veri**
Çevrim dışı	Çevrim içi
Yavaş veri üretimi	Hızlı veri üretimi
Veriler kalıcı olarak depolanır.	Veriler geçici olarak depolanır.
İşlemler bütün veri üzerinde gerçekleştirilir.	İşlemler bir grup veri üzerinde gerçekleştirilir.
Sürekli kullanılabilir.	Sınırlı kullanılabilir.
Boyutu bellidir.	Boyutu için bir sınır yoktur.
Verinin karakteristiği bilinir.	Verinin karakteristiği öngörülemez.

Geleneksel veriler üzerinde işlemler toplu olarak yapılır ve bu işlemler genellikle büyük hacimli veriler üzerinde aynı anda ve uzun gecikme süreleri ile yapılır. Diğer taraftan akan veri ile ilgili yapılacak işlemlerde verinin sırayla ve artan bir şekilde işlenmesi gerekmektedir ve bunun için popüler iki yöntem izlenmektedir: (i) kayıt bazında işlemler yapmak, (ii) kayan bir pencere üzerinde işlemler yapmak. Belirtilen farklar ve akan veri ile ortaya çıkan yeni gereksinimler ile birçok yeni araçta geliştirilmiştir. Çizelge 4.2'de akan veri için geliştirilen bazı araçlar verilmiştir.

Çizelge 4.2 Akan veri için kullanılan araçlar.

Araç	**Açıklama**
Apache Kafka	Apache Kafka, uygulamalar ve akan veri arasındaki entegrasyonu sağlayan dağıtık bir yayınla/abone ol mesajlaşma sistemidir (Url-8).
Apache Flink	Açık kaynak kodlu bir akan veri işleme aracıdır (Url-6).
Apache Storm	Apache Storm açık kaynak kodlu olup, gerçek zamanlı olarak akan veri üzerinde işlem yapma olanağı sağlar (Url-17).
Amazon Kinesis Data Streams	Gerçek zamanlı olarak büyük ölçekli akan veri toplama ve işleme aracıdır (Url-3).
Google Cloud DataFlow	Akan veri analiz servisi (Url-5).

4.2 Akan Veri Mahremiyeti

Gelişen veri madenciliği teknikleri ve analiz araçları ile verinin mahremiyetini korumak zorlaşmaktadır. Bu bağlamada veri kümelerine birçok farklı atak düzenlenebilmektedir. Daha önceki bölümlerde anlatıldığı üzere veri mahremiyetinin korunması için birçok yöntem bulunmaktadır. Fakat bu yöntemler çeşitli koşullar altında her zaman kullanılamamaktadır (verinin boyutu, verinin tipi, vb.) Akan veri günümüzün dinamik verileri olarak adlandırılabilir.

Geliştirilen geleneksel veri anonimleştirme yöntemleri akan veriler için aşağıdaki sebeplerden dolayı kullanılamamaktadır:

1. Geleneksel anonimleştirme yöntemleri statik veri kümelerinin mahremiyetini sağlamak için tasarlanmıştır.
2. Geleneksel yöntemlerde anonimleştirilecek veri kümesi içerisinde bir kişiye ait bir kayıt olduğu varsayımı yapılmaktadır. Bu varsayım akan veriler için söylenemez.
3. Statik veri kümelerinin boyutu anonimleştirme işleminden önce bilinirken, akan verinin dinamik yapısı nedeniyle boyutu ile ilgili bir bilgiye sahip değiliz. Dolayısıyla geliştirilen yöntemin kaynak kullanımı konusunda daha dikkatli olması gerekmektedir.

Belirtilen nedenlerden dolayı, akan veri için yeni anonimleştirme yöntemleri gerekmektedir. SWAF (Wang, 2007) ve SKY (Li, 2008) akan verinin anonimleştirilmesi için geliştirilmiş ilk yöntemlerdendir. Bu iki yöntemde sisteme gelen veriler arabellekte geçici olarak tutulmaktadır. Bir kayıt önceden tanımlanmış maksimum gecikme eşiği kadar sistemde kalabilir ve bu eşiğe ulaşan bir kayıt için bilgi kaybının minimum seviyede tutulması hedeflenerek yarı tanımlayıcı değerleri üzerinden $k-$anonim gruplar oluşturarak anonimleştirilir. Bu iki yöntem, yarı tanımlayıcı öznitelikler için hazırlanmış taksonomi ağaçları üzerinde yukarıdan aşağıya özelleştirme yöntemini kullanarak akan verinin anonimliğini sağlamaktadır.

Akan verinin anonimliğini sağlamak için geliştirilmiş en popüler yaklaşımlardan biri olan CASTLE (Cao, 2010), verinin mahremiyetini k-anonimlik ve $\ell-$ çeşitlilik prensipleri ile korumaktadır. Sisteme gelen kayıtlar daha sonra anonimleştirilmek üzere kümelere ayrılır. Bir kaydın hangi kümeye atanacağına, küme ile kayıt arasındaki mesafeyi ölçmek için tanımlanmış genişleme (*enlargement*) metriği kullanılarak karar verilmektedir. Genişleme metriği, bir kaydın kümeye dahil olduktan sonra o küme üzerinde ne kadar bilgi kaybına neden olduğunu ölçmektedir. Yöntemin temel amacı birbirine yakın olan kayıtları aynı kümelere toplamaktır, böylece genelleştirmeden kaynaklanan veride bozulma mümkün olduğunca küçük olacaktır.

SWAF ve SKY yöntemlerine benzer şekilde, CASTLE yaklaşımında da her kayıt sistemde belirli bir süre kalabilmektedir. Önceden belirlenen bu süre bir kayıt için dolduğunda, o kaydın bulunduğu küme anonimleştirilir.

CASTLE yaklaşımı üzerinde yapılan deneyler sonucunda, bir süre sonra kayıtların büyük bir kısmını aynı küme içerisine toplama eğilime girdiği gözlenmiştir ve bu durum o kümedeki kayıtlar için bozulma miktarını arttırmaktadır. Ayrıca diğer kümelerde de yeterli kayıt olmadığı için genelleştirme işlemi doğrudan küme üzerinde gerçekleştirilememektedir. Bu yüzden kümenin başka kümeler ile birleştirilmesi ya da süresi dolmuş kayıtlar için gizleme (*suppression*) yönteminin uygulanması gerekmektedir. Bu durum anonimleştirilen akan veri için bilgi kaybı miktarının atmasına neden olmaktadır. Dolayısıyla genişleme (*enlargement*) metriği verileri, kümelere dağıtmak için her zaman uygun bir metrik olmayabilir. CASTLE yaklaşımının bir varyantı olan B-CASTLE (Wang, 2010) yöntemi, kümelerin alabileceği kayıt sayısını sınırlayarak, CASTLE'da meydana gelen kayıtların bir kümede toplanması problemini çözmek istemiştir.

FAANST (Zakerzadeh, 2010) bir diğer kümeleme tabanlı akan veri anonimleştirme yöntemidir. Bu yöntem sadece nümerik yarı tanımlayıcıların anonimleştirilmesini sağlamaktadır. Kategorik veriler için uygulanamamaktadır. FAANST yaklaşımında sisteme gelen kayıtlar bir arabellekte saklanır. Bir kayıt için maksimum gecikme süresi dolduğunda sistemde bulunan kayıtlardan k-means kümeleme yaklaşımı ile QI-gruplar oluşturulur ve boyutu k'den büyük olan gruplar genelleştirme yolu ile anonim hale getirilir.

Bir diğer yöntem olan FADS (Guo, 2013), literatürde bilgi kaybı açısından en iyi sonuçları veren yaklaşımdır. FADS yöntemi de kümeleme tabanlı bir çözüm sunmaktadır. Sisteme gelen veriler FAANST algoritmasında olduğu gibi arabellekte saklanır. Bir kayıt için maksimum gecikme süresi dolduktan sonra, ilgili kayda en yakın $k-1$ kayıt saklanan veriler içerisinden seçilir ve bir QI-grup oluşturulur. Bu QI-grup genelleştirme yöntemi ile anonimleştirilir. Bu yaklaşımda anonimleştirilen bütün kümeler için boyut k olarak sabitlenir. FADS basit ve anlaşılır bir algoritma olmasının yanı sıra verimli ve bilgi kaybı açısından etkili bir yöntemdir.

FADS yöntemi, CASTLE, B-CASTLE ve FAANST yöntemlerinden bilgi kaybı açısından daha iyi sonuçlar elde ettiğini çalışmasında göstermektedir. Fakat, FADS yaklaşımında kayıtların ortalama gecikme süreleri CASTLE yöntemine göre yaklaşık 40% daha fazladır. Sisteme gelen kayıtların gecikmeleri farklı açılardan literatürde değerlendirilmiştir. FAST (Mohammadian, 2014) çalışması, FADS (Guo, 2013) yaklaşımını çok iş parçacıklı (*multi-threaded*) bir şekilde implemente ederek büyük akan veri için bir anonimleştirme sağlamaktadır. Çok iş parçacıklı implementasyon sisteme gelen kayıtlar için ortalama gecikmeyi azaltmaktadır. Bu bağlamda, FAST yönteminde bilgi kaybı ve gecikme miktarının ağırlıklı bir şekilde toplamını içeren bir maliyet fonksiyonu tanımlanmıştır. Fakat tanımlanan fonksiyon ve anonimleştirme algoritması arasındaki ilişki açık bir şekilde ifade edilmemiştir. Maliyet fonksiyonu çok iş parçacıklı implementasyonun kullanılabilirliğini değerlendirici bir metrik olarak sunulmaktadır. FAANST yaklaşımının bir varyantı olarak sunulan FAANST-delay (Zakerzadeh, 2013) yöntemi, sisteme gelen kayıtların zaman ekseninde gecikme miktarını sınırlayan bir yöntem olarak sunulmuştur. Bu amaçla, anonimleştirme işleminin her bir iterasyonunda bir sonraki turda gecikme süresi sona erebilecek olan kayıtları tespit edecek proaktif bir sezgisel zaman aşımı yöntemi önerilmiştir.

FAANST-delay yönteminin motivasyonu maksimum gecikme sınırını zaman bazlı olarak tanımlamaktır.

Tez kapsamında, kayıtlar için ortalama gecikme süreleri ve bilgi kaybı arasında olan negatif korelasyonu dikkate alınarak bir yöntem önerilmektedir. UBDSA olarak adlandırılan bu yaklaşım, bilgi kaybı ve ortalama gecikme arasında bir denge sağlamayı ve hatta iki değer arasında bir önceliklendirme yapabilmeyi hedeflemektedir. Literatürde önerilen yöntemler incelendiğinde, tez kapsamında önerilen çalışmanın motivasyonu ile örtüşen bir çalışma bulunmamaktadır. Fakat akan verinin tutulduğu arabellek boyutu ile ilgili dinamik bir düzenleme yapan SWET (Sakpere, 2015) yöntemi bilgi kaybını minimize etmek için bunu yapmaktadır.

Geliştirilen yöntemlerde sisteme gelen verilere belirli bir gecikme kısıtı uygulanmakla birlikte genel olarak bu yöntemlerde kümeleme tabanlı çözümler uygulanmıştır.

Tez kapsamında önerilen yöntemimiz (UBDSA) kategorik ve nümerik yarı tanımlayıcılar için çalışabilmektedir. Önerilen yöntem, CASTLE çalışmasının üzerine kurgulanmış olup algoritmasında benzer adımlar izlenmektedir. Fakat CASTLE yönteminden farklı olarak iki önemli katkı sağlamaktadır:

1. Üretilen anonim akan verinin kullanılabilirliği[1] arttırılmaktadır.
2. Sistem gelen kayıtlar için CAIL adında yeni bir atama metriği tanımlanmıştır.

Literatürde bulunan popüler kümeleme tabanlı akan veri anonimleştirme çalışmaları üç farklı kriterde karşılaştırılmıştır ve detayları Çizelge 4.3'de verilmiştir.

Çizelge 4.3 Akan veri anonimleştirme algoritmalarının karşılaştırılması.

Yöntem	**Küme oluşturulma zamanı**	**Küme seçim metriği**	**Optimizasyon hedefi**
CASTLE	Kayıt geldiğinde	Genişleme	Bilgi kaybı
FAANST	Kayıt anonimleştirilirken	Bilgi kaybı	Bilgi kaybı
FADS	Kayıt anonimleştirilirken	Bilgi kaybı	Bilgi kaybı
B-CASTLE	Kayıt geldiğinde	Genişleme	Bilgi kaybı
UBDSA	Kayıt geldiğinde	CAIL	Bilgi kaybı ve ortalama gecikme

[1] Veri kullanılabilirliği iki açıdan ele alınmıştır (i) anonimleştirilen verinin bilgi kaybı, (ii) verinin sistemde kalma süresi.

4.3 Akan Veri Anonimleştirme Çerçevesi

Bir akan veri S sürekli bir kayıt dizisi olarak tanımlanabilir, yani $S = (t_1, t_2, t_3, \dots)$. Alternatif olarak, S sıralı veri kümeleri olarak düşünülebilir ve her kayıt $t_j \in S$ benzersiz bir sisteme geliş sırasına(j) sahiptir. Ayrıca, her kayıt $t_j \in S$ aynı öznitelik şemasına $(A_1, A_2, \dots, A_j, SV)$ ve bu öznitelikler için aynı tanım kümelerine $(D_1, D_2, \dots, D_j, D_{SV})$ sahiptir. $QI = (A_1, A_2, \dots, A_j)$ öznitelik kümesindeki yarı tanımlayıcıları ifade etmektedir ve SV ise hassas özniteliği ifade etmektedir. Bir kayıt $t \in S$ için öznitelikler $t = (a_1, a_2, \dots, a_j, sv)$ şeklinde gösterilmektedir. Bir kaydın t herhangi bir öznitelik değeri a_m nokta notasyonu $t.a_m$ ile ifade edilir. Öznitelik kümesinden bir kaydı doğrudan tanımlayan özniteliklerin öncesinde çıkarıldığını unutmayalım.

Özniteliklerin tanım kümeleri birbirinden bağımsızdır ve nümerik ya da kategorik olabilirler. Fakat, her kategorik öznitelik A_i için önceden tanımlanmış taksonomi ağacı TT_{A_i} olduğu varsayımında bulunulmaktadır. Aynı şekilde nümerik alanlar için tanım kümesinin sınırlarının önceden bilindiği varsayılmaktadır. Ayrıca bir taksonomi ağacının TT_{A_i} yaprak değerleri ilgili kategorik özniteliğin A_i tanım kümesinden D_{A_i} oluşmaktadır.

4.3.1 Akan veri anonimizasyonu için problem tanımı

Akan veri anonimleştirme algoritmaları, girdi olarak aldığı akan veriyi S, anonim akan veri çıktısı olarak S' sağlar. j zamanında, sistemde bulunan kayıtlar $S_j = (t_1, t_2, \dots, t_j)$ şeklinde ifade edilmektedir ve j zamanında anonimleştirilen QI-grup $S'_j \in S_j$ ile gösterilmektedir. Eğer $|S'_j| < |S_j|$ ise, arabellekte anonimleştirilmemiş kayıtlar olduğu anlamına gelmektedir. Anonimleştirilen QI-grup içerisinde bulunan kayıtlar için herhangi bir sıra söz konusu değildir. Ayrıca anonimleştirilen QI-grup içerisindeki kayıtlar için sistemden ayrılma sıraları aynıdır. Bir kaydın sistemden ayrılma sırası $t.ro$ ile ifade edilir. Akan veri ile sürekli sisteme gelen veri ifade edilmektedir, dolayısıyla kayıtların sistemde uzun süre bekletilmemeleri gerekmektedir. Bir kaydın sistemde kalabileceği süre için gecikme eşiği ya da gecikme kısıtı ifadesi kullanılmaktadır ve δ ile gösterilmektedir.

Tanım 4.1 (QI-grup) Yarı tanımlayıcı değerleri aynı olan bir kayıt kümesi $T' \subseteq S'$ bir QI-grup oluşturabilir.

Tanım 4.2 (k −anonim QI-grup) Boş olmayan bir QI-grup için aşağıdaki şartları sağladığı takdirde k −anonim denilir:

- Öncesinde tanımlanan bir k değeri için $|T'| \geq k$ ya da,
- T' içerisinde bulunan kayıtların bütün yarı tanımlayıcı değerleri en üst seviyeden genelleştirildiyse (*suppression*).

Yukarıdaki tanımda, en üst seviyeden genelleştirilmiş bir kayıt kümesi, tanım gereği k −anonim olduğu unutulmamalıdır. Açıkçası, bu durum bir kaydın yarı tanımlayıcı değerleri hakkında herhangi bir bilgi vermemektedir.

Tanım 4.3 (k −anonim akan veri) j zamanında, eğer S'_j k-anonim bir akan veri ise, S'_j içerisinde bulunan bütün QI-grupların k −anonim olması gerekmektedir.

Birçok çalışmada maksimum gecikme eşiği olarak arabellek boyutu kullanılmıştır. Dolayısıyla gereksiz parametre kullanmamak için maksimum gecikme eşiği ve arabellek boyutu δ ifade edilmiştir.

Problem 4.1 (k −anonim akan veri paylaşımı) Belirli bir maksimum gecikme eşiği için δ, herhangi bir zamanda $j = 1, 2, 3, \ldots$, k −anonim akan veri paylaşımı için aşağıdaki koşulların sağlandığından emin olunmalıdır:

i. $S'_j, k -$ anonim olmalıdır.

ii. Hiç bir kayıt δ'dan fazla sistemde beklememelidir, $\forall t'_i \in S', t'_i.ro - i < \delta$,

iii. Ortalama gecikme $AverageDelay(S'_j)$ minimum seviyede tutulmalıdır.

iv. Ortalama bilgi kaybı $InformationLoss(S'_j)$ minimum seviyede tutulmalıdır.

Ortalama gecikme süresi bir akan veri için Eşitlik (4.1) ile hesaplanır.

$$AverageDelay(S'_j) = \frac{\sum_{t'_i \in S'_j} t'_i.ro - i}{|S'_j|} \tag{4.1}$$

Ortalama bilgi kaybı ise Eşitlik (2.1) ile hesaplanmaktadır.

Teorem 1 k −anonim akan veri yayınlama problemi NP-Hard bir problemdir.

İspat Problem 1'den (ii) ve (iii) numaralı kısıtlar çıkartıldığında problem k-anonimlik problemine indirgenmektedir. k-anonimlik yaklaşımının NP-Hard olduğu bilinmektedir (Aggarwal, 2005).

Akan verinin anonimleştirilmesi NP-Hard bir problem olduğu için tez kapsamında önerilecek yöntem kümeleme tabanlı sezgisel bir yaklaşımdır.

4.4 Fayda Tabanlı Akan Veri Anonimleştirme Algoritması

Problem 4.1'de iki adet optimizasyon hedefi belirtilmiştir: (i) ortalama gecikme süresini minimize ederek verinin yaşlanmasını en aza indirgemek, (ii) bilgi kaybı miktarını azaltarak verinin kalitesini arttırmak. Tez kapsamında verinin yararlılığını, verinin kalitesini ve verinin yaşlanmasını kullanacak bir fonksiyon olarak tanımlıyoruz. Bu bölümde sunulacak olan fayda tabanlı akan veri anonimleştirme algoritması (*Utility Based Data Stream Anonymization,* UBDSA) verinin sağlayacağı faydayı maksimuma çıkarmayı amaçlamaktadır.

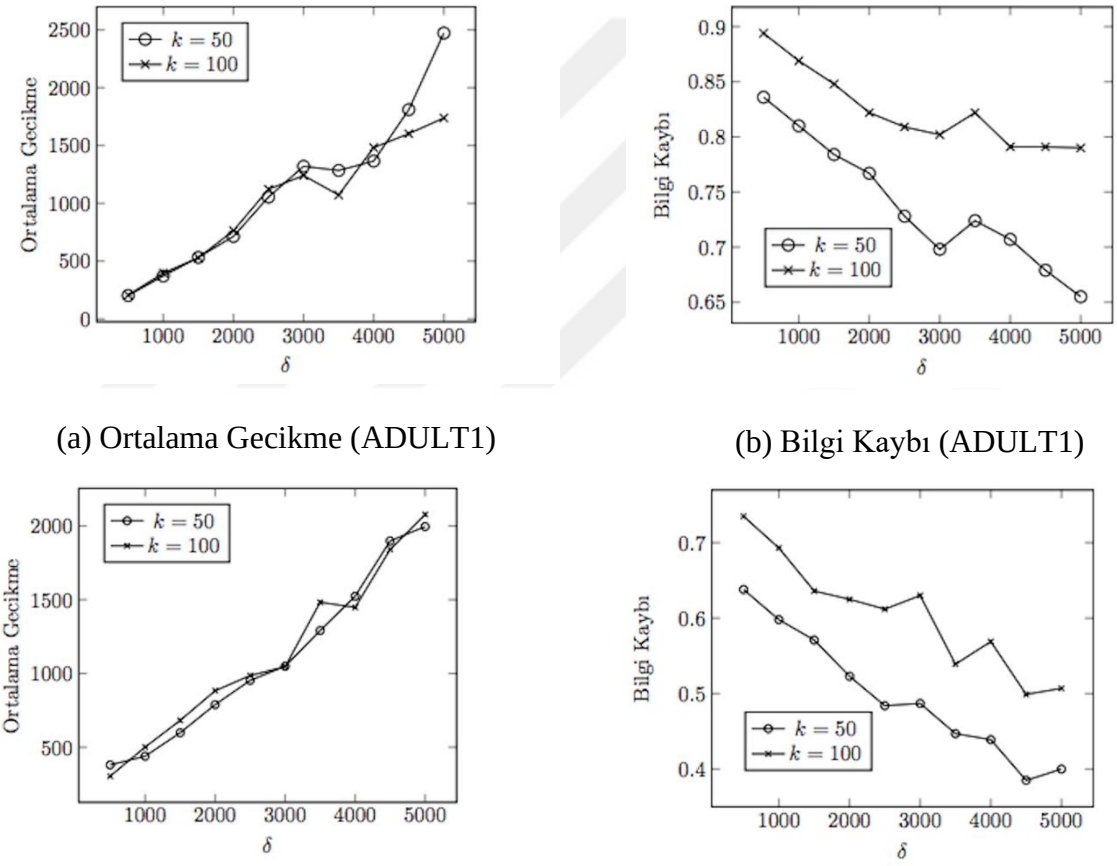
UBDSA, Problem 4.1'de belirtilen sorunları çözmek için geliştirilmiş kümeleme tabanlı bir akan veri anonimleştirme algoritmasıdır. UBDSA metodu, sisteme gelen her kaydı bir kümeye atamaktadır ve problem tanımında belirtilmiş olan gecikme sınırı hiçbir kayıt için aşılmamaktadır. Ayrıca sistem içerisinde tanımlanmış arabellek kapasitesi de aşılmamaktadır. UBDSA yönteminde, sisteme ulaşan kayıt doğrudan bir kümeye dahil olur ya da o kayıt üzerinde yeni bir küme oluşturulur. UBDSA yönteminde, kategorik özniteliklerin anonimleştirilmesi için önceden tanımlanmış taksonomi ağaçlarına ihtiyaç duyulmaktadır.

UBDSA yönteminin hedeflediği iki kriter arasında negatif bir korelasyon bulunmaktadır. Yani, ortalama gecikme süresini arttırmak bilgi kaybı miktarını azaltırken, ortalama gecikme süresini azaltmak bilgi kaybı miktarını arttırmaktadır. Bu iddiayı Şekil 4.1'de doğrulamaktadır. UBDSA yaklaşımının temel motivasyonu olan bu problem için iki metrik arasında dengenin sağlanması ya da metrikler için bir önceliklendirme hedeflenmektedir.

Geliştirilen akan veri anonimleştirme yöntemleri içerisinde δ değeri sabit tutulmaktadır. Tez kapsamında önerilen çalışmada δ değeri çalışma sırasında güncellenebilmektedir. δ_c ile o iterasyonda geçerli olan gecikme eşiği ifade edilmektedir. δ_c değeri k ile δ arasında değişmektedir ($k \leq \delta_c \leq \delta$). δ_c değerinin

arttırılması bilgi kaybı miktarını minimize ederken, azaltılması verinin yaşlanmasının önüne geçmektedir. δ_c değeri önceden sistemde belirlenmiş adım boyutu (*stepsize*) kadar arttırılıp azaltılmaktadır. Kısaca, δ değeri uyulması gereken kesin bir kısıt ve üst sınırken, δ_c dinamik olarak değişebilen bir kısıttır.

UBDSA algoritmasının ana metodu Şekil 4.2'de verilmiştir. j anında, sisteme gelen kayıt t_j dir ve yayınlanmamış veriler *UnpublishedTuples* listesi içerisinde tutulmaktadır. *UnpublisedTuples* listesindeki kayıtlar QI-grup kümeleri (*NonAnonyCluster)* içerisinde organize edilmektedir.



(a) Ortalama Gecikme (ADULT1)

(b) Bilgi Kaybı (ADULT1)

(c) Ortalama Gecikme (ADULT2)

(d) Bilgi Kaybı (ADULT2)

Şekil 4.1 CASTLE algoritması ve iki farklı k değeri kullanılarak, ortalama gecikme ve bilgi kaybı arasında olan negatif ilişki gösterilmektedir.

Input: $S, \delta, k, w, stepsize$
Output: S'
1: $AnonClusters \leftarrow \emptyset$
2: $NonAnonClusters \leftarrow \emptyset$
3: $j \leftarrow 1$
4: $\delta_c \leftarrow \delta$
5: **while** S_j has next tuple t_j **do**
6: **AssignCluster**(t_j, $NonAnonClusters$)
7: $UnPublishedTuples \leftarrow S_j \setminus S'_j$
8: $t_o \leftarrow$ oldest tuple from $UnPublishedTuples$
9: **while** $j - o \geq \delta_c$ **do**
10: **Publish**($t_o, AnonClusters, NonAnonClusters$)
11: **UpdateDelta**($\delta_c, \delta, k, stepsize, AnonClusters$)
12: Shift $AnonClusters$
13: $t_o \leftarrow$ oldest tuple from $UnPublishedTuples$
14: $j \leftarrow j + 1$

Şekil 4.2 UBDSA algoritması

AssignCluster metodu, t_j'i anonimleştirilmemiş kümelerden birisine (*NonAnonClusters*) atar ya da sadece t_j'i içeren yeni bir küme oluşturur ve bu yeni kümeyi anonimleştirilmemiş küme listesi ekler, hangi işlemin uygulanacağı ile ilgili karar daha sonra anlatılacak olan bir değerlendirme fonksiyonuna dayanmaktadır. Hiçbir kayıt δ_c'den daha fazla geciktirilemediğinden, önce yayınlanmamış en eski kayıt t_o belirlenir (satır 8) ve ardından mevcut δ_c eşiğine karşı kaydın bekleme süresi kontrol edelir (satır 9). Sürenin dolmuş olması durumunda, ilgili kayıt aynı küme içerisinde bulunduğu kayıtlar ile birlikte ya da tek başına hemen anonimleştirilir (10. satırda çağırılan *Publish* metodu içerisinde). Detayları sonra verilecek olan *Publish* prosedürü, t_0'ın anonimleştirilmesini üç farklı şekilde gerçekleştirebilmektedir.

1. t_o'nun içerisinde bulunduğu kümeyi genelleştirme yaklaşımı ile anonimleştirebilir.

2. *AnonCluster* (daha önce anonimleştirilmiş kümelerin prototip bilgileri de tutulmaktadır) içerisinde t_o ile eşleşen bir küme olması durumunda, var olan kümenin genelleşme seviyesini kullanarak t_o'yu anonim hale getirebilir.

3. t_o için gizleme (*suppression*) yaklaşımı uygulayabilir.

UpdateDelta metodu, δ_c değerini en son anonimleştirilen $2w$ kümenin içeriğine bağlı olarak *stepsize* kadar arttırarak ya da azaltarak güncellemektedir. Pencere boyutu (w) ve adım boyutu ($stepsize$) parametreleri kullanılarak ortalama bilgi kaybı ve ortalama gecikme arasındaki denge ayarlanabilmektedir. Son olarak, anonimleştirilen kümeler *AnonClusters* listesine eklenir.

Maksimum oluşturulabilecek anonimleştirilmemiş küme sayısı β ile sınırlandırılırken, anonimleştirilmiş en fazla μ tane kümenin prototipleri saklanmaktadır. Bizim yaklaşımımızda, *AnonClusters*'ın boyutu μ'e ulaştığında akan bir pencere gibi çalışıp en eski küme listeden çıkartılır.

AssignCluster prosedürü: Yayınlanmamış kayıtlar, *NonAnonyCluster* içerisinde bulunan kümelerde tutulmaktadır. *AssignCluster* metodu, sisteme gelen t_j'i bir kümeye atayabilmek için uygun bir küme olup olmadığını kontrol eder. Olması durumunda bu kaydı ilgili kümeye atar. Aksi durumda, bu kayıttan yeni bir küme oluşturur ve oluşan kümeyi *NonAnonyCluster*'e ekler.

Algoritma ile ilgili detaylar Şekil 4.3'te verilmiştir. t_j ile anonimleştirilmemiş kümeler arasında uzaklık Eşitlik (4.2) ile hesaplanmaktadır ve minimum sonuç veren kümeler $C_{minDist}$ içerisinde toplanmaktadır (satır 1). Kardinaliteye duyarlı bilgi kaybı (CAIL) olarak isimlendirilen uzaklık metriği bu çalışmanın önemli katkılarından biridir. Uygulanan bu metrik ile t_j için uygun olmayan kümeler filtrelenir. Son anonimleştirilen μ kümenin ortalama bilgi kaybı τ ile ifade edilmektedir. $C_{minDist}$ içerisinde kümelerin bilgi kaybı miktarına bakılarak bir filtre daha uygulanır ve sonuçlar C_{minIL} içerisinde tutulur (satır 2). Sonuç olarak C_{minIL} boş değilse, içerisinde kayıt sayısı en az olan kümeler belirlenir (satır 4) ve bu liste içerisinde rastgele bir küme seçilir ve t_j kümeye eklenir (satır 5-6). C_{minIL} 'nin boş olması durumunda, anonimleştirilmemiş küme sayısı eğer β'a ulaşmadıysa t_j'den yeni bir küme oluşturulur (satır 12). Aksi takdirde, t_j'i atamak için $C_{minDist}$ içerisinden rastgele bir küme seçilir (satır 9-10).

Bir kayıt t ve bir küme C arasındaki uzaklık (4.2) ile hesaplanmaktadır. Eşitlikte, IL fonksiyonu küme içerisindeki bütün kayıtların anonimleştirilmesi sonrası oluşan bilgi kaybı miktarını ifade etmektedir. Bu amaçla $IL(C)$ ve $IL(C \cup \{t\})$ sırasıyla bir kaydın (t) bir kümeye (C) eklenmesinden önce ve sonraki bilgi kaybını göstermektedir.

$$CAILDistance(t, C) = IL(C \cup \{t\}) + \big(IL(C \cup \{t\}) - IL(C)\big) * \log(|C|) \quad (4.2)$$

Input: t_j, $NonAnonClusters$
Output: updated $NonAnonClusters$

1: $C_{minDist} \leftarrow argmin_c\{CAILDistance(t \cdot c) \mid c \in NonAnonClusters\}$
2: $C_{minIL} \leftarrow \{c \mid c \in C_{minDist}\ and\ IL(\{t_j\} \bigcup c) \leq \tau\}$
3: **if** $C_{minIL} \neq \emptyset$ **then**
4: $\quad C_{cand} \leftarrow argmin_c\{|c| \mid c \in C_{minIL}\}$
5: $\quad c_{pick} \leftarrow$ A random cluster from C_{cand}
6: $\quad c_{pick} \leftarrow c_{pick} \bigcup \{t_j\}$
7: **else**
8: $\quad$ **if** $|NonAnonClusters| \geq \beta$ **then**
9: $\quad\quad c_{pick} \leftarrow$ A random cluster from $C_{minDist}$
10: $\quad\quad c_{pick} \leftarrow c_{pick} \bigcup \{t_j\}$
11: $\quad$ **else**
12: $\quad\quad NonAnonClusters \leftarrow NonAnonClusters \bigcup \{t_j\}$

Şekil 4.3 AssignCluster prosedürü

Publish prosedürü: *Publish* metodunun amacı, anonimleştirilen QI-grupların yayınlanmasıdır. t_o anonimleştirilmesi gereken temel kayıt olarak değerlendirilir ve anonimleştirilecek QI-grubun onu içermesi sağlanır. Bu amaçla, anonimleştirilmemiş küme listesi içerisinden t_o'yu içeren c_{t_0} kümesi belirlenir (satır 1). Eğer kümedeki kayıt sayısı QI-grup olmak için yeterli ise, yani en az k tane kayıt varsa, küme yayınlanır (satır 3-9). Ayrıca, eğer kümedeki eleman sayısı $2k$'den fazla ise, *splitCluster* metodu çağrılır ve küme k-NN algoritması kullanılarak her biri en az k tane kayıt içerecek şekilde parçalanır. Kümenin küçük parçalara ayrılma sebebi genelleme miktarını düşürerek veri kalitesi arttırmaktır. *OutputCluster* metodu Tanım 4.2'de verilen parametrelere göre küme içerisindeki kayıtları genelleştirir. Sonrasında bütün kayıtlar beraber yayınlanır. c_{t_o} kümesi içerisinde yeterince kayıt olmadığı durumda 11-25 arası kod bloğu çalıştırılır.

FindFittingClusters prosedürü içerisinde, anonim küme listesi (*AnonClusters*) içerisinden t_o ile eşleşen kümeler çıkartılır (C) ve içerisinden rastgele bir küme seçilir (c_{pick}) ve t_o kaydı c_{pick} ile anonimleştirilir (satır 13-14). Burada, t_o'ya en yakın küme yerine rastgele bir küme seçildiği unutulmamalıdır. Bu işlem t_o karşı düzenlenebilecek olan kontra atak saldırılarını önlemek için yapılmaktadır. *outputTuple* prosedürü ile t_o'ın c_{pick}'e genelleştirilmiş hali yayınlanır (satır 14). Eğer t_o için daha önce anonimleştirilmiş kümeler arasında eşleşen bir küme bulunamazsa ($C = \emptyset$), 17 ile 25 satırları arasındaki kod bloğu çalıştırılır.

Anonimleştirilmemiş kayıt sayısı k'den küçük olduğu durumda, bu kayıtlardan bir QI-grup oluşturulamamaktadır. Bu durumda anonimleştirilmesi gereken kaydın bütün yarı tanımlayıcı değerleri en üst seviyeye genelleştirilir (*suppression*) ve öyle yayınlanır (satır 21). Aynı yaklaşım c_{t_0}'ın içerdiği kayıt sayısı, bütün kümelerin kayıt sayısının medyanından küçükse de uygulanır (Cao, 2010). Eğer arabellekte yeterince anonimleştirilmemiş kayıt varsa, 23 ile 25 arasında bulunan kod bloğu çalıştırılır. *mergeCluster* prosedürü, *NonAnonyCluster* içerisinden c_{t_o}'a en yakın küme ile birleştirir bu işlem kümenin boyutu k değerine ulaşana kadar devam etmektedir (satır 23). Birleştirilen kümelerden bir QI-grup yaratılır ve yayınlanır (satır 24).

UpdateDelta prosedürü: Girdi olarak *AnonyClusters* kümesinden en son anonimleştirilen $2w$ boyutunda QI-grup listesi alır. Prosedür en son yayınlanmış $2w$'lik liste içerisinden yeni w'lik QI-grup ve eski w'lik QI-grup için ortalama bilgi kaybını hesaplar ve bu değerleri karşılaştırır. Eğer eski olan QI-grup için bilgi kaybı miktarı daha küçükse, δ_c değeri *stepsize* kadar arttırılır (δ_c değerinin $k \leq \delta_c \leq \delta$ koşulunu sağladığı kontrol edilir). δ_c değerinin arttırılması verinin yaşlanmasına neden olacaktır. Diğer bir taraftan δ_c değerinin düşürülmesi ise daha fazla bilgi kaybına neden olacaktır. İki durumda verinin kullanılabilirliğini etkilemektedir. Verinin kullanılabilirliği açısından bu çalışma bilgi kaybı ile ortalama gecikme arasındaki dengeyi sağlamaya çalışmaktadır. w uzunluğu, artış / azalma sıklığının ayarlanması için kullanılırken, *stepsize* değeri büyüklüğü belirler.

4.4.1 UBDSA algoritmasının karmaşıklığı

$(A_1, A_2, \ldots, A_n)$ yarı tanımlayıcı özniteliklerken, $(D_1, D_2, \ldots, D_n)$ bu özniteliklere bağlı tanım kümeleridir. β ve μ sırasıyla hafızada tutulabilecek maksimum anonimleştirilmemiş küme sayısını ve maksimum anonimleştirilmiş küme sayısını ifade etmektedir. Ayrıca, arabellek boyutu da δ ile gösterilmektedir.

UBDSA ana metodu *AssignCluster* prosedürünü her iterasyonda kesinlikle bir kez, *Publish* ve *UpdateDelta* prosedürünü ise her iterasyonda bir kez (amortize edilmiş) çalıştırmaktadır.

Input: t_o, $AnonClusters$, $NonAnonClusters$
Output: updated $AnonClusters$, $NonAnonClusters$

1: $c_{t_o} \leftarrow \{c \mid c \in NonAnonClusters \text{ and } t_o \in c\}$
2: **if** $|c_{t_o}| \geq k$ **then**
3: $\quad NonAnonClusters \leftarrow NonAnonClusters \setminus \{c_{t_o}\}$
4: $\quad$ **if** $|c_{t_o}| \geq 2k$ **then**
5: $\quad\quad C \leftarrow$ **splitCluster**(c_{t_o})
6: **for each** $c_i \in C$ **do**
7: $\quad\quad$ **outputCluster**(c_i)
8: $\quad$ **else**
9: $\quad\quad$ **outputCluster**(c_{t_o})
10: **else**
11: $\quad C \leftarrow$ **findFittingClusters**$(t_o, AnonClusters)$
12: $\quad$ **if** $C \neq \emptyset$ **then**
13: $\quad\quad c_{pick} \leftarrow$ A random cluster from C
14: $\quad\quad$ **outputTuple**(**generalize**(t_o, c_{pick}))
15: $\quad\quad c_{t_o} \leftarrow c_{t_o} \setminus \{t_o\}$
16: $\quad$ **else**
17: $\quad\quad nonAnonT \leftarrow \{t : c \in NonAnonClusters \text{ and } t \in c\}$
18: $\quad\quad smallC \leftarrow \{c : c \in NonAnonClusters \text{ and } |c| < |c_{t_o}|\}$
19: $\quad\quad$ **if** $|nonAnonT| < k$ OR $|smallC| \leq |NonAnonClusters|/2$ **then**
20: $\quad\quad\quad c_{t_o} \leftarrow c_{t_o} \setminus \{t_o\}$
21: $\quad\quad\quad$ **outputTuple**(**generalize**$(t_o, *)$)
22: $\quad\quad$ **else**
23: $\quad\quad\quad$ **mergeCluster**$(c_{t_o}, NonAnonClusters)$
24: $\quad\quad\quad$ **outputCluster**(c_{t_o})
25: $\quad\quad\quad NonAnonClusters \leftarrow NonAnonClusters \setminus \{c_{t_o}\}$

Şekil 4.4 UpdateDelta prosedürü

AssignCluster algoritmasının 1. satırı prosedürü domine etmektedir, bu satırda CAIL metriği $O(\beta)$ kez hesaplanmaktadır. CAIL metriği hesaplanırken bilgi kaybının her bir öznitelik için hesaplanması gerekmektedir. Bir özniteliğin bilgi kaybı hesaplanırken öznitelik için tanımlanmış taksonomi ağacında küme ve kayıt için en yakın ortak ata düğümün belirlenmesi gerekmektedir. Bu işlem $O(|D|)$ maaliyet ile gerçekleşmektedir. Dolayısıyla *AssignCluster* için toplam çalışma maliyeti $O(\beta n|D|)$ dir. Fakat, $O(|D|\log(|D|))$ ön bir maaliyet ile, en yakın ortak ata düğümü bulma işlemi dinamik programlama yaklaşımı kullanılarak $O(log|D|)$ maaliyet ile yapılabilmektedir. Sonuç olarak, *AssignCluster* prosedürü $O(\beta nlog|D|)$ zaman maaliyeti ile implemente edilebilir.

Publish prosedürü *splitCluster*, *findFittingClusters*, *mergeClusters* ve genelleştirme alt prosedürleri en fazla bir kez çalıştırmaktadır. Kümeleme işlemi için doğrusal

(*linear*) zamanda çalışabilecek bir algoritma kullanılabilir ve çalışma zamanı $O(\delta\beta nlog|D|)$ olur. Biz aç gözlü bir kümeleme yaklaşımı kullanıyoruz. Diğer üç prosedürün *splitCluster* ile karşılaştırıldığında çalışma zamanları önemsizdir.

UpdateDelta prosedüründe, kademeli olarak ortalama hesaplamak kayan pencereden (*sliding window*) dolayı çok hızlıdır ve $O(1)$ zamanında uygulanabilir.

Sonuç olarak en büyük maliyete neden olan bileşen alındığında algoritma $O(\delta\beta nlog(|D|))$ zaman karmaşıklığı ile çalışmaktadır. Bu karmaşıklık ifadesindeki parametreler işlem sırasında değişmemektedir. Dolayısıyla her adımda çalışma süresi için üst sınırlar sabit değerler ile kısıtlanmıştır.

4.5 UBDSA Algoritmasının Deneysel Değerlendirilmesi

Bu bölümde, UBDSA yaklaşımı kullanılarak yapılan deneysel değerlendirmeler sunulacak ve iyi bilinen akan veri anonimleştirme yöntemleri CASTLE ve FADS ile UBDSA yöntemi karşılaştırılacaktır. Bu çalışma kapsamından bütün implementasyonlar Java programlama dili ile geliştirilmiştir ve deneyler Intel i7 2.2 GHz CPU ve 16 GB RAM bulunan bir bilgisayar üzerinde gerçekleştirilmiştir. Üç farklı veri kümesi kullanılmıştır. Kullanılan veri kümeleri içerisinde kategorik değerler olduğu için FAANST yaklaşımına deneylerde yer verilememiştir. Algoritmaların performansları iki farklı metrik üzerinden karşılaştırılmıştır: (i) ortalama gecikme, (ii) ortalama bilgi kaybı.

4.5.1 Veri kümeleri

UBDSA yaklaşımının performansı ADULT (Url-1), TELCO (Url-9) ve NURSERY (Url-2) veri kümeleri üzerinde yapılan deneyler ile değerlendirilmiştir. ADULT veri kümesi veri mahremiyeti çalışmalarında sıklıkla kullanılan bir veri kümesidir. Yarı tanımlayıcı sayısının etkisini anlayabilmek için ADULT veri kümesini farklı sayılarda yarı tanımlayıcı içerecek şekilde ADULT1 ve ADULT2 olarak iki veri kümesi gibi kullanıldı. "*Nation*" özniteliği hassas veri olarak belirlendi. ADULT1 ve ADULT2 veri kümelerinin deneyler içersinde kullanılan öznitelikleri Çizelge 4.4'te verilmiştir. TELCO veri kümesi 21 öznitelik ve 7043 kayıttan oluşmaktadır. TELCO veri kümesi için "Monthly charges", "Total charges" ve "Churn" öznitelikleri hassas olarak belirlenmiştir.

TELCO veri kümesi özelinde deneylerde kullanılan öznitelikler Çizelge 4.5'te verilmiştir. Gerçekçi bir e-ticaret (müşteri karmaşası) veri kümesi olan TELCO veri kümesinin, akan veri anonimleştirme kapsamında ADULT veri kümesinden daha uygun olduğunu düşünüyoruz. Bildiğimiz kadarıyla, akan veri anonimleştirme çalışmaları içerisinde, TELCO veri kümesi ilk bu çalışmada kullanılmıştır. NURSERY veri kümesi de birçok veri mahremiyetinin korunmasına yönelik çalışmalarda kullanılmış veri kümelerinden birisidir. Bu veri kümesinin kullanılan özniteliklerine ait bilgileri Çizelge 4.6'da verilmiştir ve veri kümesi belirtilen öznitelikler ile kullanılan versiyonu bu bölüm içerisinde NURSERY olarak isimlendirilecektir. Veri kümesi 12960 kayıttan ve 9 öznitelikten oluşmaktadır. Üç veri kümesi de statik olduğundan, akan veri şeklinde modelleyebilmek için bunların sıralı olarak geldiği simüle edilmektedir.

Veri kümeleri içerisinde kategorik ve nümerik yarı tanımlayıcı öznitelikler bulunmaktadır. ADULT veri kümesi için literatürde bulunan standart taksonomi ağaçları kullanılmıştır. TELCO ve NURSERY veri kümeleri içerisinde bulunan kategorik öznitelikler için taksonomi ağaçları tarafımızca üretilmiştir. Neyse ki, öznitelikler için tanım küme boyutları oldukça küçük olduğundan taksonomi ağaçları alan uzmanlığı gerekmeksizin kendiliğinden oluşturulabilmektedir.

Çizelge 4.4 ADULT1 ve ADULT2 veri kümelerinde kullanılan öznitelikler.

ADULT1 öznitelikleri	**ADULT2 öznitelikleri**	**Tanım kümesinin boyutu**
Age	Age	100
Education	Education	16
Status	Status	7
Relationship	Relationship	6
Race	Race	5
Gender	Gender	2
Workclass	Workclass	8
Occupation		14
Hours per week		100
Capital loss		5000
Nation		41

Çizelge 4.5 TELCO veri kümesinde kullanılan öznitelikler.

TELCO öznitelikler	**Tanım kümesinin boyutu**
Tenure	72
Gender	2
Contract	3
Dependents	2
Device protection	3
Internet service	3
Multiple lines	3
Online backup	3
Online security	3
Paperless billing	2
Partner	2
Payment method	4
Phone service	2
Senior citizen	2
Streaming movies	3
Streaming TV	3
Tech support	3
Monthly charges	*Decimal*
Total charges	*Decimal*
Churn	*2*

Çizelge 4.6 NURSERY veri kümesinde kullanılan öznitelikler.

NURSERY öznitelikler	**Tanım kümesinin boyutu**
Parents	3
Has_nurs	5
Form	4
Children	4
Housing	3
Finance	2
Social	3
Health	3
Class	5

4.5.2 Ön deneyler

Bu çalışmanın motivasyonu için ADULT1 ve ADULT2 üzerinde çeşitli deneyler gerçekleştirilmiştir. Şekil 4.1'de ortalama gecikme ve bilgi kaybı arasındaki negatif korelasyon gösterilirken, Şekil 4.5'te önerilen CAIL metriği CASTLE yaklaşımı içerisinde kullanılan genişleme metriği ile değiştirildiğinde ve ortalama bilgi kaybı miktarının azaldığı gözlenmiştir. Deneyler sonucunda elde edilen bulgular bu çalışmanın motivasyonu olmuştur.

4.5.3 Deney sonuçları

Akan veri anonimleştirme algoritmaları ortalama gecikme ve bilgi kaybı metriklerine göre değerlendirilmektedir. Bu bölümde anonimleştirme algoritmaları farklı parametre değerleri ile test edilecektir. Bu şekilde verinin yaşlanması ve kalitesi arasındaki denge gözlemlenebilecektir.

4.5.3.1 Hafıza boyutunun performans üzerindeki etkisi

UBDSA algoritmasının model boyutu, hiper parametreler olan β ve μ (anonimleştirilmiş ve anonimleştirilmemiş kümelerin maksimum sayısı) ile belirlenir. Hiper parametrelerin performans üzerindeki etkisini anlayabilmek için ADULT1 üzerinde deneyler yapılmıştır. Deneyler düşük (β = μ = 10), orta (β = μ = 50) ve yüksek (β = μ = 100) değerler ile gerçekleştirilmiştir ve sonuçlar Çizelge 4.7'de verilmiştir. Sonuçlar incelendiğinde, orta seviyedeki değerlerin bilgi kaybı ve ortalama gecikme açısında genel olarak daha iyi sonuçlar verdiği gözükmektedir.

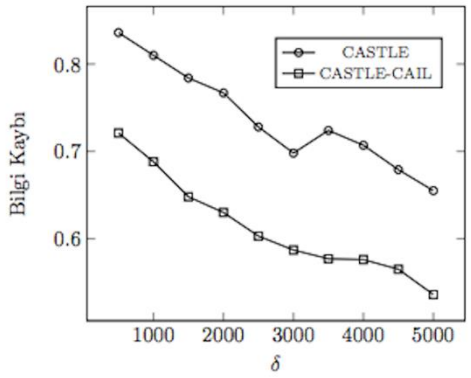
Çizelge 4.7 UBDSA algoritmasında β ve μ metriklerinin bilgi kaybı ve ortalama gecikme üzerindeki etkisi

k	**Bilgi kaybı**			**Ortalama gecikme**		
	β = μ = 10	β = μ = 50	β = μ = 100	β = μ = 10	β = μ = 50	β = μ = 100
50	0.566	0.528	0.547	2509	2380	2397
100	0.627	0.624	0.634	2469	2471	2603

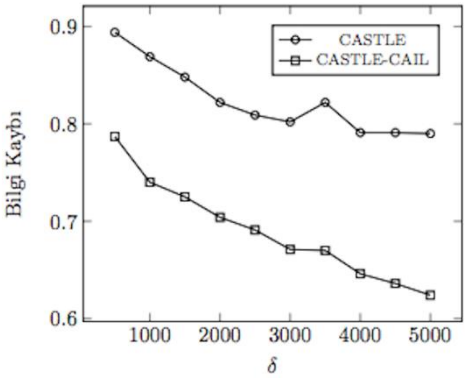
Elde edilen sonuçlar ışığında, deneylerde CASTLE ve UBDSA tarafından kullanılan hiper parametreler aşağıdaki değerlere sabitlenmiştir:

- μ = 50
- β = 50

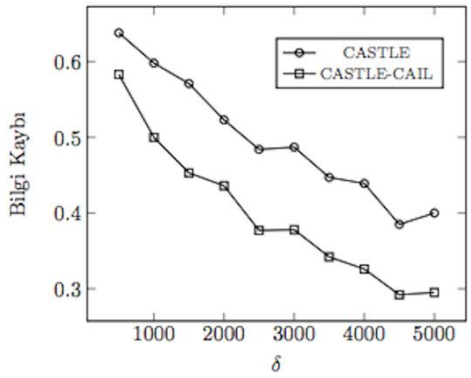
ve FADS'ın ihtiyaç duyduğu $T_{kc} = 200$ olarak belirlenmiştir.



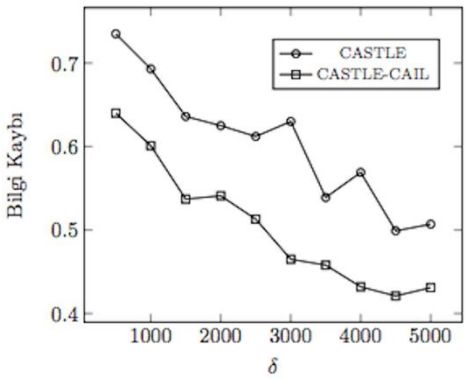
(a) ADULT1 ($k = 50$)



(b) ADULT1 ($k = 100$)



(c) ADULT2 ($k = 50$)



(d) *ADULT2* ($k = 100$)

Şekil 4.5 CASTLE ve CASLTE-CAIL'ın bilgi kaybı açısından karşılaştırılması.

4.5.3.2 Literatür ile karşılaştırma

Bu bölümde UBDSA yaklaşımı literatürde bulunan FADS ve CASTLE yöntemleri ile anonimleştirilen verinin kullanılabilirliği açısından karşılaştırılacaktır. UBDSA yönteminde üretilen verinin kullanılabilirliği dikkate alındığında, üç farklı pencere boyutu değeri ile deneylerde sunulacaktır.

- UBDSA1 (UBDSA ve w = 0)
- UBDSA2 (UBDSA ve w = 1)
- UBDSA3 (UBDSA ve w = 3)

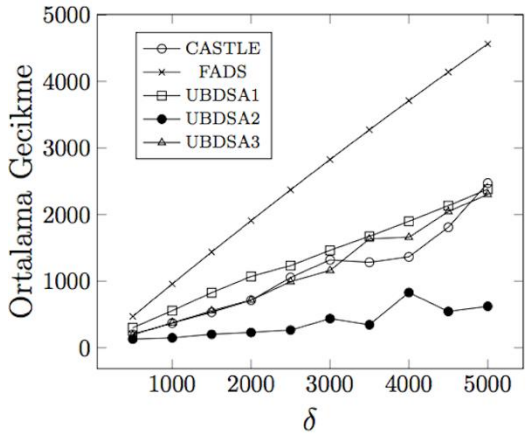
Sonuçlar Şekil 4.6, Şekil 4.7, Şekil 4.8 ve Şekil 4.9'da gösterilmiştir. Deneylerde *stepsize* parametresi ADULT1 ve ADULT2 için 50, TELCO veri kümesi için 10 ve NURSERY veri kümesi için stepsize 25 olarak sabitlenmiştir. TELCO ve NURSERY veri kümesi içerisindeki kayıt sayısı az olduğu için *stepsize*, δ ve k değerleri için küçük

değerler seçilmiştir. Sonuçlar incelendiğinde, UBDSA bütün konfigürasyonlarında ortalama gecikme açısından FADS'dan daha iyi sonuçlar vermektedir. Diğer bir taraftan, ortalama gecikme açısından CASTLE yaklaşımı UBDSA1'den daha iyi sonuçlar verirken, UBDSA2'den sonuçları daha kötüdür. UBDSA2 yaklaşımı diğer yöntemler ile karşılaştırıldığında en iyi ortalama gecikme değerlerini elde etmiştir. Bilgi kaybı açısından karşılaştırdığımızda, ADULT1 üzerinde yapılan deneylerde en iyi sonucu UBDSA1 vermektedir. Fakat ADULT2 ve TELCO veri kümeleri üzerinde yapılan deneylerde FADS en iyi sonucu sağlamaktadır. NURSERY veri kümesi üzerinde yapılan deneylerde FADS genel olarak iyi sonuçlar vermesine rağmen UBDSA1 ile bilgi kaybı açısında yakın sonuçlar üretmektedir. Fakat ADULT2 üzerinde $k = 100$ parametresi ile yapılan deneylerde UBDSA ve FADS birbirlerine yakın sonuçlar üretmektedir.

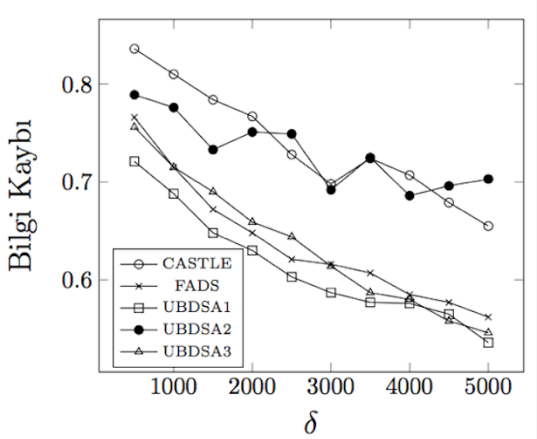
Ortalama gecikme ve bilgi kaybı açısından sonuçlarını özetlersek, UBDSA1 ve UBDSA3'ün, üç veri kümesi üzerinde ürettiği sonuçlar için bir denge sağladıkları gözükmektedir. En önemlisi sonuçlar, (i) bu çalışmanın motivasyonu deneysel olarak doğrulanmaktadır ve (ii) UBDSA'nın gerçekten de veri kalitesi ve veri yaşlanmasının önemini ölçebilen iyi bir yaklaşım olduğunu göstermektedir.

4.5.3.3 Pencere boyutunun performans üzerindeki etkisi

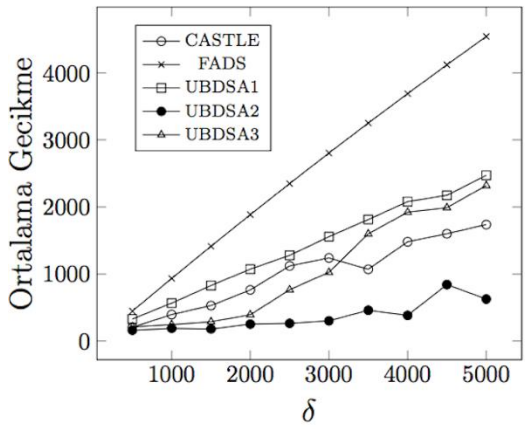
Pencere boyutunun UBDSA'nın performansı üzerindeki etkisini anlayabilmek için çeşitli adım boyutlarında deney yapılmıştır. Bu deneylere ait sonuçlar Şekil 4.10 ve Şekil 4.11'de verilmiştir. İlk olarak, sonuçlar açıkça tüm konfigürasyonlarda ortalama gecikme ve bilgi kaybı arasında çok yüksek bir negatif korelasyon olduğunu göstermektedir. Bu nedenle, sadece ortalama gecikme sonuçları üzerine yorum yapmayı tercih ediyoruz. δ_c , k ile δ arasında değerler alabilmektedir ve δ_c'nin değeri δ ile başlar ve rastgele yürüyüş modeline benzer bir akışla hareket eder. Küçük pencere boyutu ile artan ya da azalan geçiş sayıları daha fazla olacaktır, daha küçük pencere boyutları ile ortalama gecikme miktarının düşük olması beklenilir. Sonuçlar da bu iddiayı desteklemektedir.



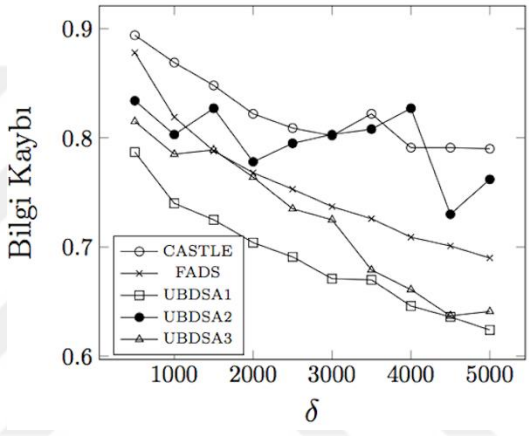
(a) k=50



(b) k=50

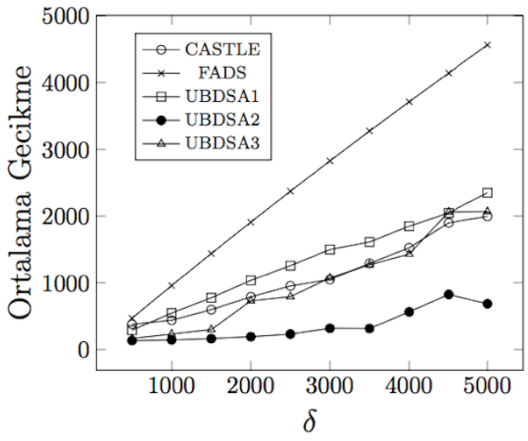


(c) k=100

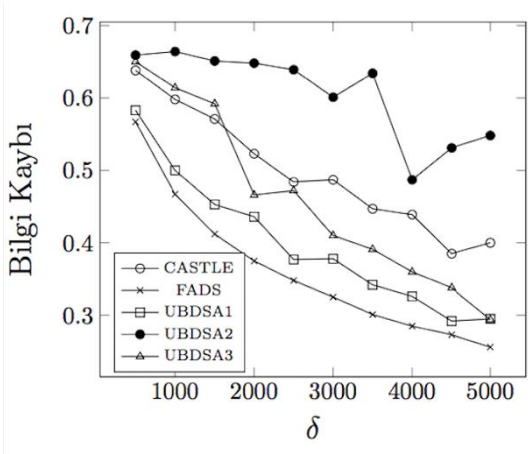


(c) k=100

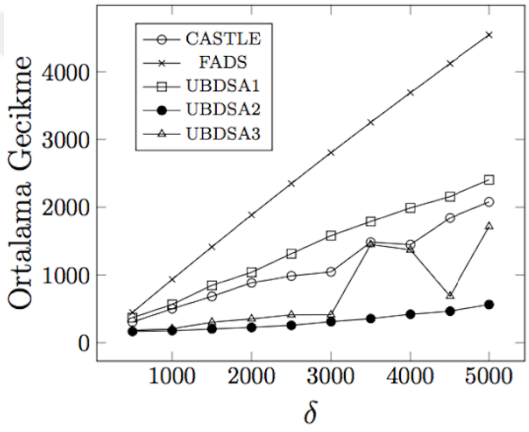
Şekil 4.6 ADULT1 veri kümesi üzerinde UBDSA, CASTLE ve FADS için anonim veri kullanılabilirliği sonuçları



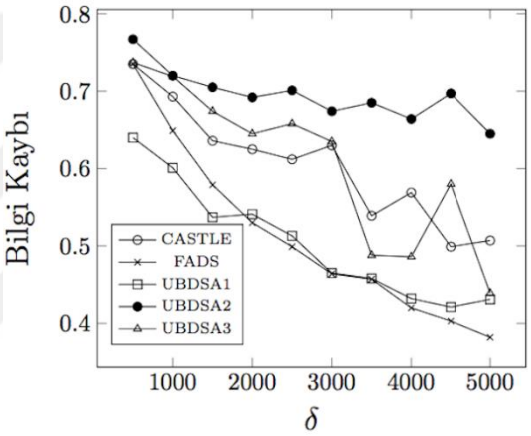
(a) k=50



(b) k=50

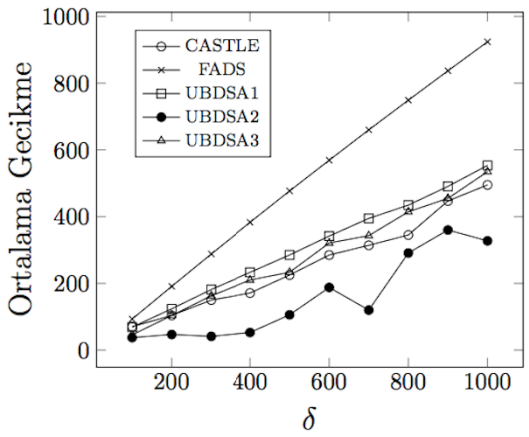


(c) k=100

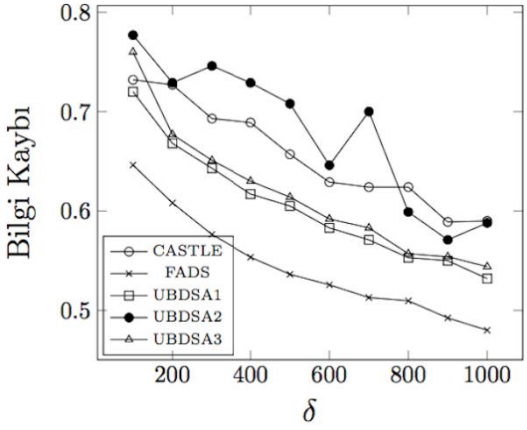


(c) k=100

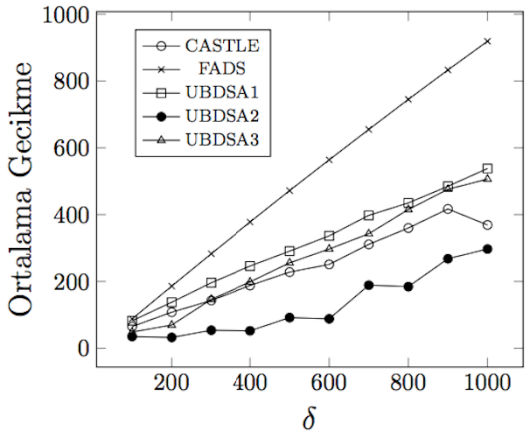
Şekil 4.7 ADULT2 veri kümesi üzerinde UBDSA, CASTLE ve FADS için anonim veri kullanılabilirliği sonuçları



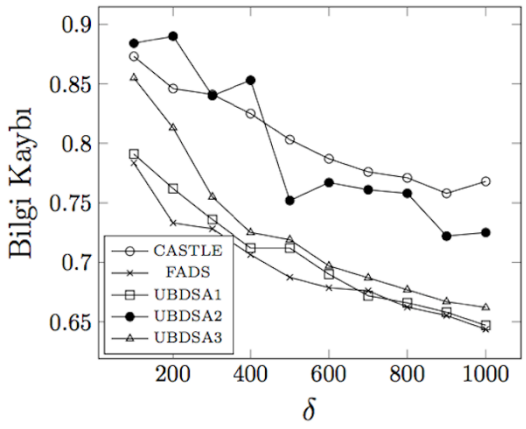
(a) k=10



(b) k=10

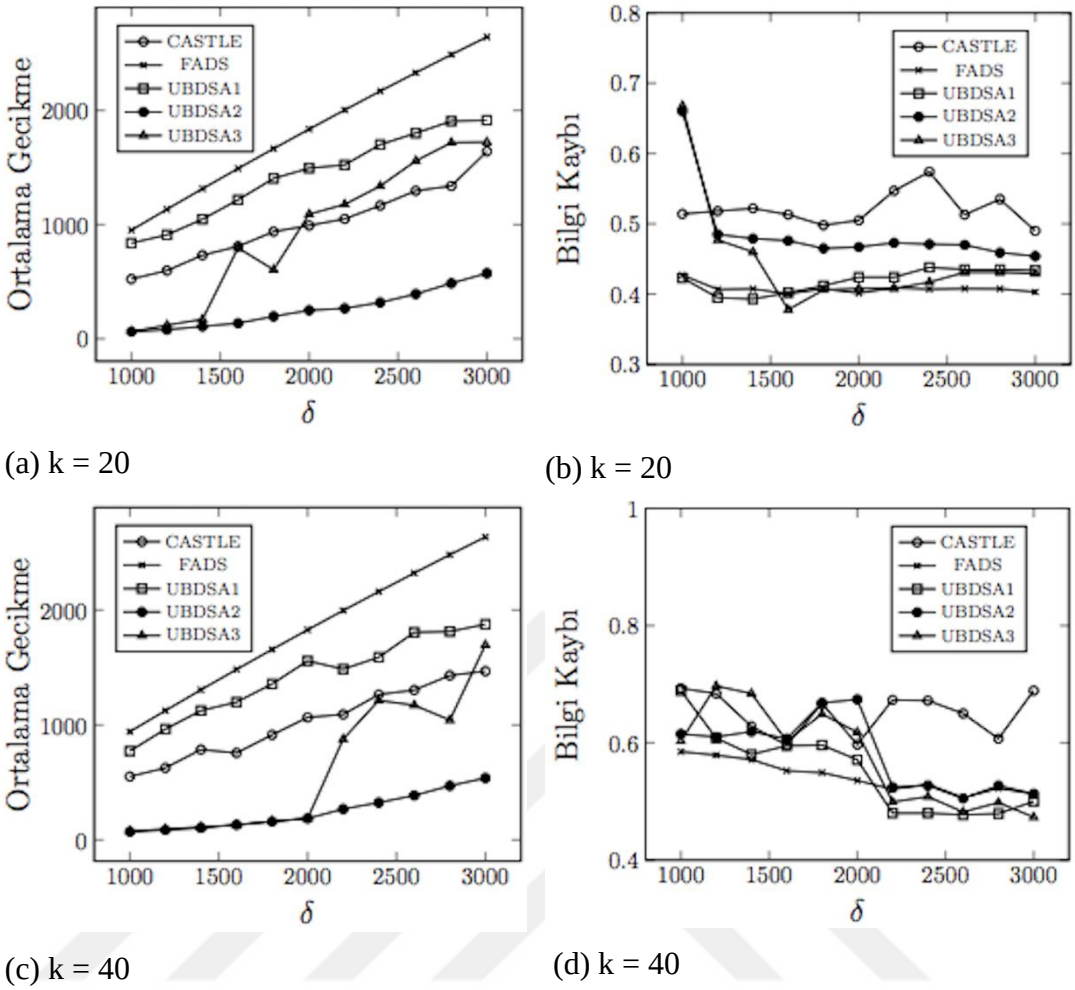


(c) k=20



(d) k=20

Şekil 4.8 TELCO veri kümesi üzerinde UBDSA, CASTLE ve FADS için anonim veri kullanılabilirliği sonuçları



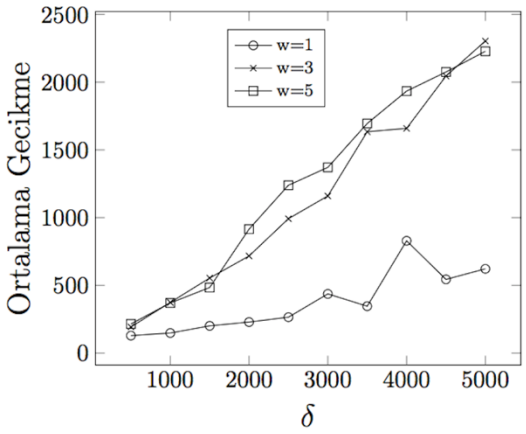
(a) k = 20

(b) k = 20

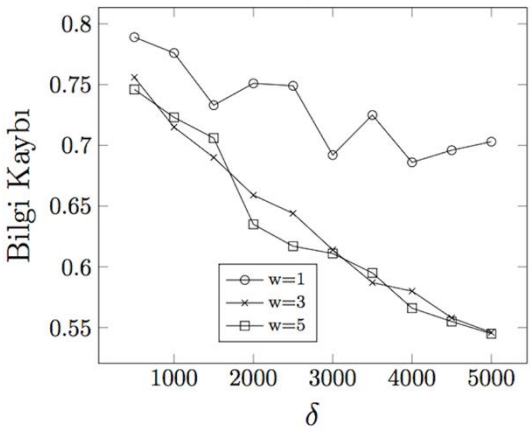
(c) k = 40

(d) k = 40

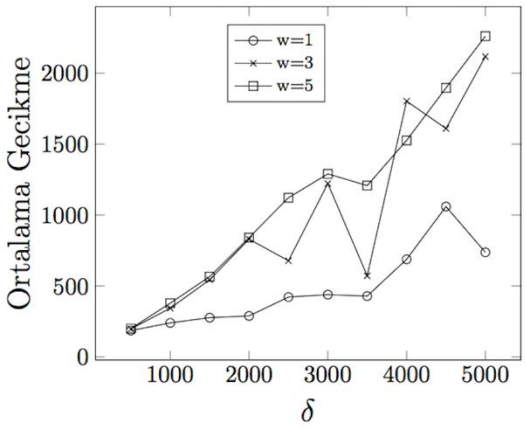
Şekil 4.9 NURSERY veri kümesi üzerinde UBDSA, CASTLE ve FADS için anonim veri kullanılabilirliği sonuçları



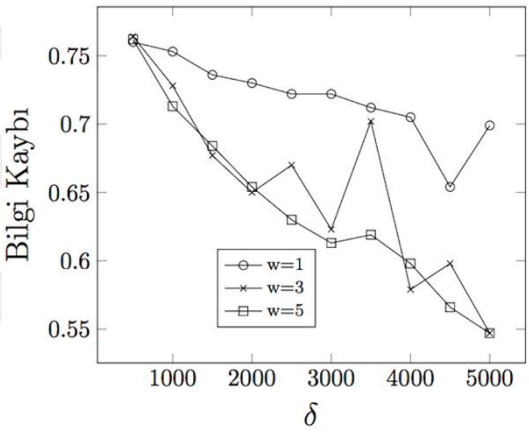
(a) stepsize = 50



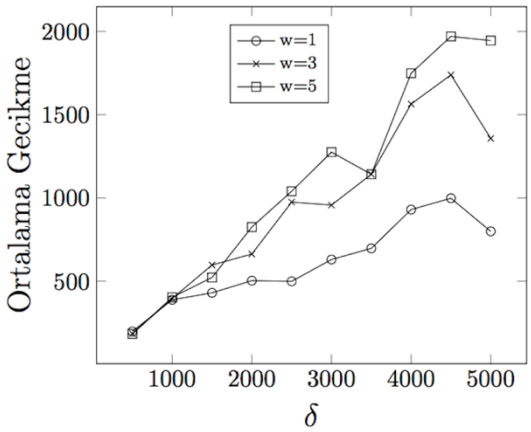
(b) stepsize = 50



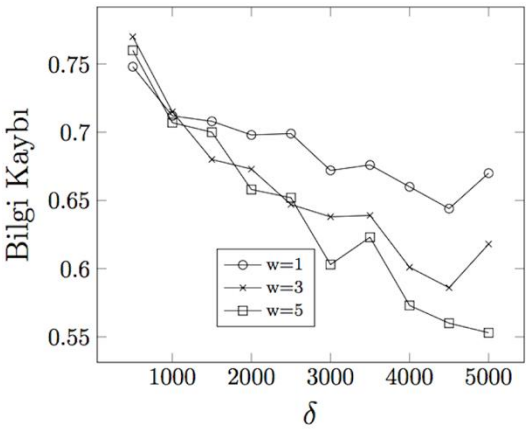
(c) stepsize = 100



(d) stepsize = 100

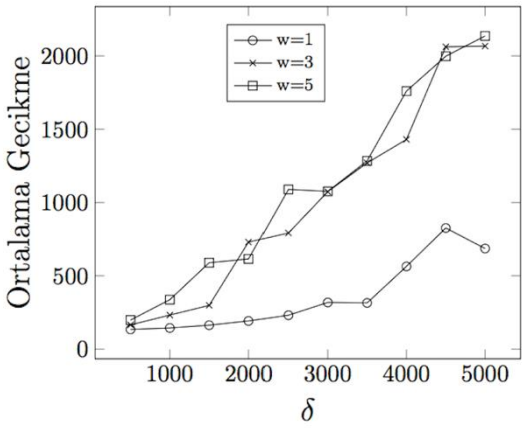


(e) stepsize = 200

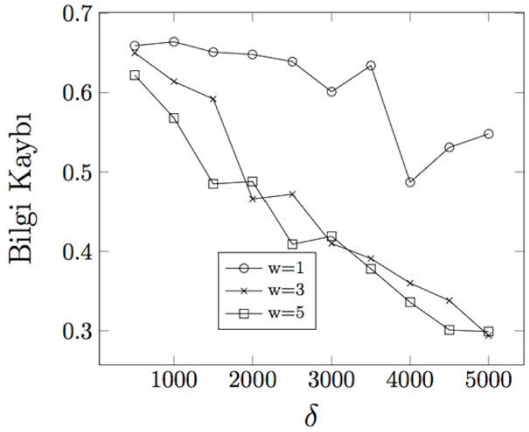


(f) stepsize = 200

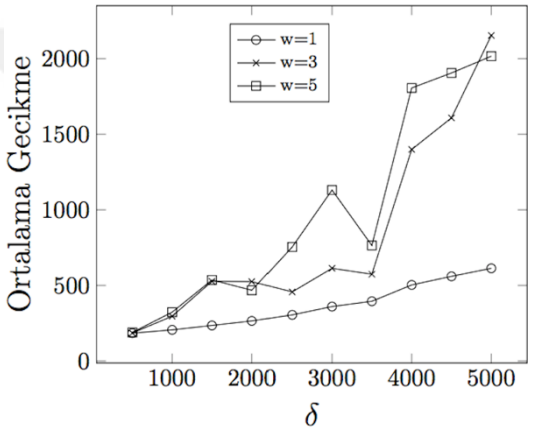
Şekil 4.10 Pencere boyutunun UBDSA yönteminde veri kullanılabilirliğine etkisi (ADULT1 ve k=50)



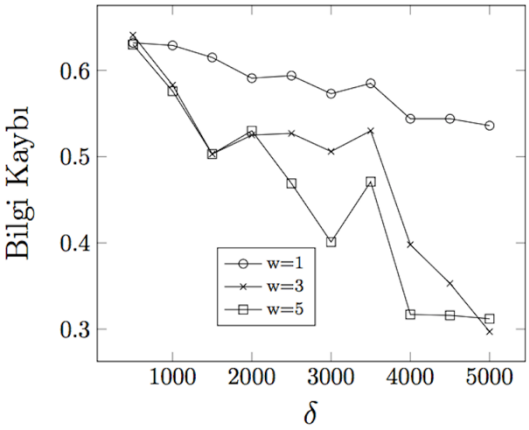
(a) stepsize = 50



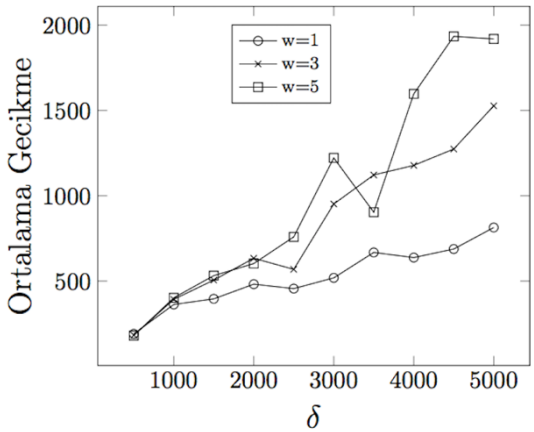
(b) stepsize = 50



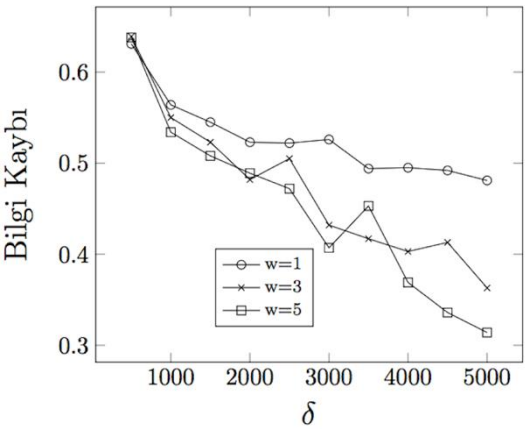
(c) stepsize = 100



(d) stepsize = 100



(e) stepsize = 200



(f) stepsize = 200

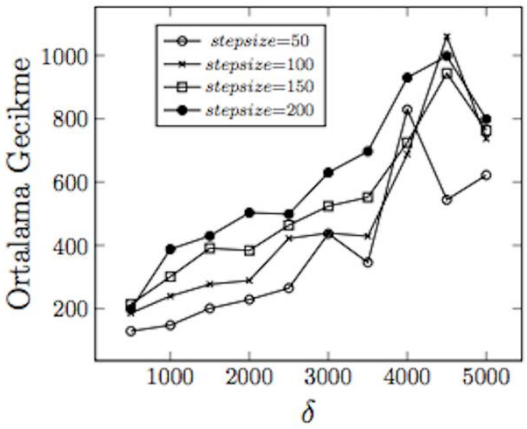
Şekil 4.11 Pencere boyutunun UBDSA yönteminde veri kullanılabilirliğine etkisi (ADULT2 ve k=50)

4.5.3.4 Adım sayısının performansa etkisi

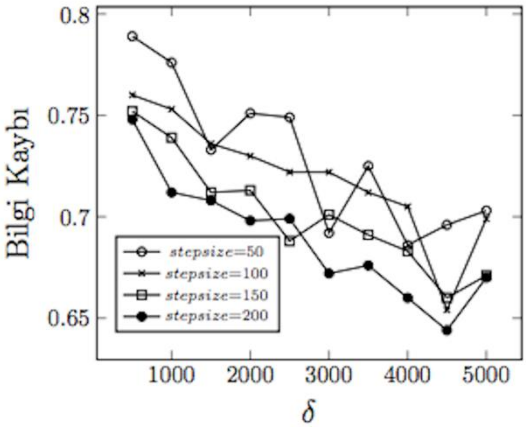
Adım boyutunun UBDSA'nın performansı üzerindeki etkisini anlayabilmek için farklı pencere boyutları ile testler gerçekleştirilmiştir. Bu testlere ait sonuçlar Şekil 4.12 ve Şekil 4.13'te sunulmuştur. Pencere boyutu deneyinde olduğu gibi, ortalama gecikme ve bilgi kaybı arasında çok kuvvetli bir negatif bir korelasyon olduğu bütün konfigürasyonlarda açıkça görülmektedir. Her iki veri kümesi üzerinde yapılan deneylerde pencere boyutu arttıkça değerler birbirine yaklaşmaktadır. Örneğin, ortalama gecikme değerleri w = 5 iken birbirlerine çok yakındır. Fakat w = 1 iken değerler birbirlerinden oldukça farklıdır. Bu aslında daha yüksek pencere boyutu değerlerinin dengeyi arttırdığını göstermektedir ve adım boyutunu etkisini azaltmaktadır. Sonuçlar ayrıca adım ve pencere boyutu parametrelerinin ortogonal (*orthogonal*) olmadığını göstermektedir. $stepsize = 50$ ve $w = 1$ olduğu durumda ortalama ortalama gecikmenin en küçük, ve $w = 5$ olduğunda en yüksek olduğu göz önünde bulundurulmalıdır.

4.5.3.5 UBDSA yönteminde önceliklendirme

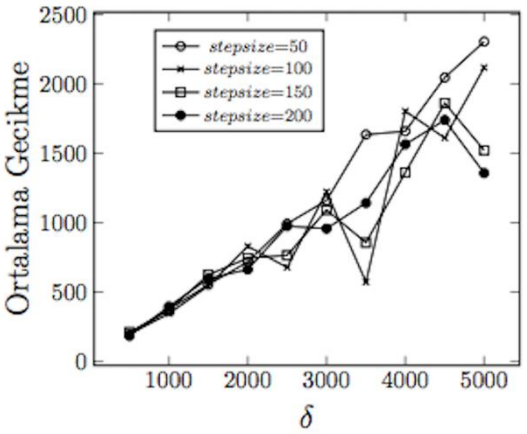
Önerilen modelde adım ve pencere boyutuna nasıl karar verileceği önemli bir sorudur. Bu bağlamda, pencere ve adım boyutu arasındaki etkileşimi göstermek için üç farklı anonimlik seviyesi (k=50, k=150 ve k=250) altında ADULT1 veri kümesi üzerinde yapılan deneylerde oluşan ortalama gecikme ve bilgi kaybı değerlerinden çıkarılan ısı haritaları Şekil 4.14'te sunulmuştur. Ek olarak, $k = 50$, $k = 150$ ve $k = 250$ için ortalama değerlerden elde edilen ısı haritası da Şekil 4.14'te verilmiştir. Sonuçlar, UBDSA yaklaşımının beş farklı adım boyutu ($stepsize = 100, 200, 300, 400, 500$) ve beş farklı pencere boyutu ($w = 1, 3, 5, 7, 9$) üzerinde test edilmesi sonucu elde edilmiştir. Isı haritaları incelendiğinde, ortalama gecikme ağırlıklandırılırsa sonuçlar "Ortalama Gecikme" tarafında olumlu çıkarken, bilgi kaybı önceliklendirildiğinde (bilgi kaybı için ağırlık daha fazla verildiğinde) sonuçlarda "Bilgi Kaybı" değerinin minimize olduğu görülmektedir. Dengeli bir sonuç için bilgi kaybı ve ortalama gecikme için belirlenen ağırlıkların birbirlerine yakın olması gerekmektedir.



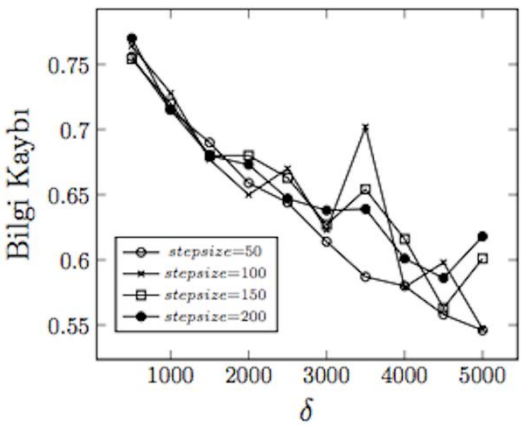
(a) w = 1



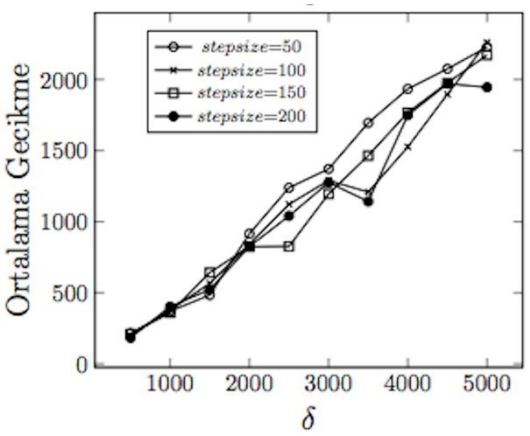
(b) w = 1



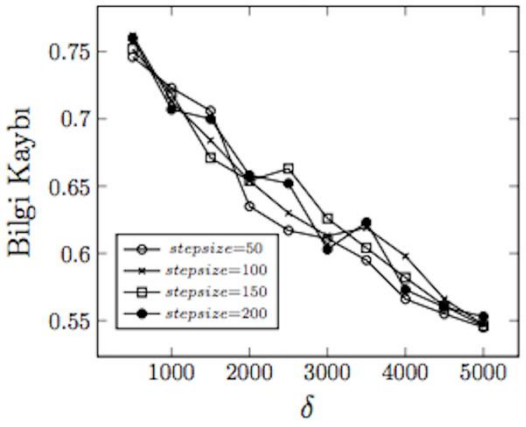
(c) w = 3



(d) w = 3

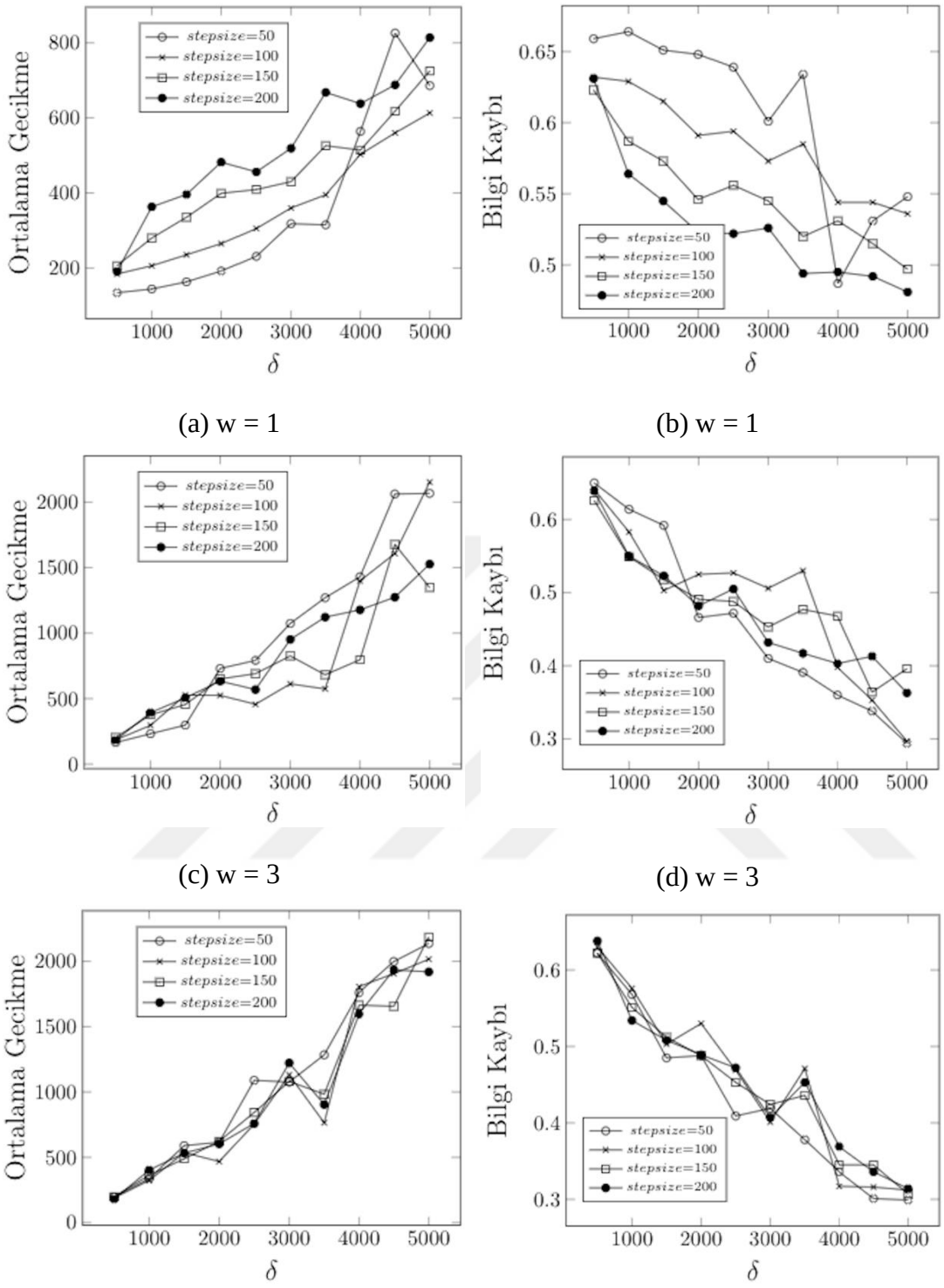


(e) w = 5



(f) w = 5

Şekil 4.12 Adım boyutunun UBDSA yönteminde veri kullanılabilirliğine etkisi (ADULT1 ve k=50)



(a) w = 1 (b) w = 1

(c) w = 3 (d) w = 3

(e) w = 5 (f) w = 5

Şekil 4.13 Adım boyutunun UBDSA yönteminde veri kullanılabilirliğine etkisi (ADULT2 ve k=50)

Akan veri mahremiyetinden sorumlu kişinin belirleyeceği bir ağırlıklandırma ile bilgi kaybı ve ortalama gecikme arasında bir önceliklendirme yapılabilmektedir. "Balanced" sütununda verilen değerler, "Average Delay" ve "Information Loss" sütunlarının kendi içlerinde normalize (0 ile 1 arasında) edilmiş değerlerinin ortalaması alınarak hesaplanmıştır. Isı haritası, UBDSA performansının hem adım boyutu hem de pencere boyutu parametreleri ile ilişkili olduğunu doğrular. Sonuçlar $k-$anonimlik seviyesine duyarlıdır. Bununla birlikte, genel eğilim (i) ortalama gecikme için küçük pencere boyutu ve küçük adım boyutu, (ii) bilgi kaybı için büyük pencere boyutu küçük adım boyutu yönündedir. "Balance" şeması için her k değerinde farklı bazı ara değerler daha iyi sonuçlar üretmektedir. Metrikler için tanımlanacak olan ağırlıklara karar verirken parametre değerlerinin seçimi için Çizelge 4.8'den faydalanılabilir. Bu önerilerin daha önce belirlenmiş olan hiper parametreler $\mu = \beta = 50$ değerleri ile test elde edildiği unutulmamalıdır.

4.5.3.6 Değerlendirmeler

Günümüzde işletmelerin bilgi işlem altyapılarının hızla ilerliyor olması, verilerin bazen akış şeklinde paylaşılmasına sebep oldu. Verinin mahremiyeti ile ilgili kaygıları gidermek ve hassas verileri korumak için akan verinin anonimleştirilmesi gerekmektedir. Bu bağlamada akan verinin anonimleştirilmesi için geliştirilmiş CASTLE ve FADS gibi birçok anonimleştirme yöntemi bulunmaktadır. Bu çalışmaların genel amacı, maksimum gecikmen kısıtı altında veri kalitesinin bir ölçüsü olan bilgi kaybını minimum seviyede tutmaktır. Birçok alanda verinin yaşlanmasını istenmeyen bir durum olduğundan, bu çalışmanın motivasyonu ortalama gecikme ve bilgi kaybını beraber minimize etmektir. Fakat, bilgi kaybı ve ortalama gecikme arasında negatif bir korelasyon bulunmaktadır ve bu durum çalışma kapsamında gösterilmiştir. Dolayısıyla, UBDSA iki hedef arasındaki dengeyi dinamik olarak kontrol ederek korumayı amaçlamaktadır. Sonuç olarak UBDSA konfigüre edilebilir bir çözüm sunmaktadır.

Ayrıca CASTLE metoduna benzer kümeleme tabanlı bir anonimleştirme sağlayan UBDSA içerisinde CAIL adı verilen yeni bir mesafe metriği de geliştirilmiştir. CAIL, CASTLE içerisinde sunulan enlargement adı verilen mesafe metriğinden önemli ölçüde daha iyi performans göstermektedir.

	Average Delay					Information Loss					Balanced				
	w					w					w				
stepsize	1	3	5	7	9	1	3	5	7	9	1	3	5	7	9
100	736	2116	2260	2206	2289	0,70	0,55	0,55	0,55	0,55	0,50	0,45	0,50	0,47	0,50
200	798	1356	1944	2152	2262	0,67	0,62	0,55	0,55	0,55	0,43	0,44	0,41	0,46	0,50
300	994	1570	1753	2034	2224	0,65	0,60	0,59	0,56	0,57	0,43	0,45	0,47	0,47	0,54
400	1213	1766	1882	1368	1713	0,63	0,59	0,57	0,61	0,58	0,44	0,46	0,46	0,40	0,43
500	1495	1801	1631	1578	2099	0,62	0,59	0,59	0,60	0,57	0,48	0,49	0,44	0,43	0,53

(a) $k = 50$

	Average Delay					Information Loss					Balanced				
	w					w					w				
stepsize	1	3	5	7	9	1	3	5	7	9	1	3	5	7	9
100	1062	848	2030	2174	2044	0,77	0,76	0,68	0,68	0,68	0,56	0,42	0,49	0,50	0,46
200	1048	1353	1488	1846	2320	0,78	0,72	0,71	0,69	0,67	0,57	0,41	0,42	0,44	0,54
300	1160	1774	1708	2015	2208	0,75	0,71	0,70	0,69	0,66	0,49	0,52	0,46	0,49	0,46
400	1243	1780	1730	1662	1654	0,75	0,71	0,69	0,67	0,70	0,50	0,54	0,41	0,31	0,43
500	1369	1858	2005	1734	2001	0,74	0,70	0,69	0,70	0,70	0,49	0,48	0,50	0,46	0,54

(b) $k = 150$

	Average Delay					Information Loss					Balanced				
	w					w					w				
stepsize	1	3	5	7	9	1	3	5	7	9	1	3	5	7	9
100	847	1833	1591	2189	2241	0,83	0,74	0,75	0,73	0,75	0,50	0,41	0,40	0,48	0,62
200	932	1319	1147	1890	2189	0,79	0,77	0,81	0,75	0,75	0,35	0,38	0,52	0,51	0,62
300	1365	1285	1283	1743	1300	0,76	0,78	0,80	0,76	0,79	0,36	0,40	0,50	0,50	0,48
400	1072	1360	1631	2092	1621	0,79	0,78	0,75	0,75	0,77	0,39	0,43	0,39	0,57	0,49
500	1256	1555	1435	1390	1850	0,77	0,76	0,77	0,77	0,76	0,36	0,44	0,44	0,43	0,50

(c) $k = 250$

	Average Delay					Information Loss					Balanced				
	w					w					w				
stepsize	1	3	5	7	9	1	3	5	7	9	1	3	5	7	9
100	882	1599	1960	2190	2191	0,77	0,68	0,66	0,65	0,66	0,52	0,43	0,46	0,49	0,53
200	926	1343	1526	1963	2257	0,75	0,70	0,69	0,66	0,66	0,45	0,41	0,45	0,47	0,55
300	1173	1543	1581	1931	1911	0,72	0,69	0,70	0,67	0,67	0,43	0,45	0,48	0,49	0,49
400	1176	1635	1748	1707	1663	0,72	0,69	0,67	0,68	0,68	0,45	0,48	0,42	0,43	0,45
500	1373	1738	1690	1567	1983	0,71	0,68	0,68	0,69	0,68	0,45	0,47	0,46	0,44	0,53

(d) $k = 50$, $k = 150$ ve $k = 250$ için gerçekleştirilen deneylerin ortalaması

Şekil 4.14 Adım boyutu ve pencere boyutu metriklerinin performansa etkisinin ısı haritası üzerindeki gösterimi

Çizelge 4.8 Tercih edilen anonimlik seviyesine göre önerilen parametre değerleri.

Anonimlik seviyesi	**Ortalama gecikme**	**Bilgi kaybı**	**Denge durumu**
Küçük k	$stepsize = 100, w = 1$	$stepsize = 100, w = 5$	$stepsize = 400, w = 7$
Orta k	$stepsize = 100, w = 3$	$stepsize = 300, w = 9$	$stepsize = 400, w = 7$
Büyük k	$stepsize = 100, w = 1$	$stepsize = 100, w = 7$	$stepsize = 200, w = 1$
Belirsiz k	$stepsize = 100, w = 1$	$stepsize = 100, w = 7$	$stepsize = 200, w = 3$

ADULT ve TELCO veri kümeleri üzerinde gerçekleştirilen deneylerin sonuçları bilgi kaybı ve ortalama gecikme açısından FADS ve CASTLE ile karşılaştırılmıştır. Ayrıca UBDSA yöntemi farklı konfigürasyon parametreleri ile test edilmiştir. Sonuçlar, (i) diğer yöntemler ile karşılaştırıldığında UBDSA'nın daha dengeli sonuçlar ürettiği gözükmektedir, (ii) UBDSA bilgi kaybı ve ortalama gecikme arasında bir önceliklendirme yapma fırsatı da sunmaktadır. Gelecekteki çalışmalarımız için (i) aynı çerçevede $l-$çeşitliliği gibi diğer gizlilik modellerini incelemek ve (ii) UBDSA'yı Hadoop MapReduce ve Apache Spark gibi büyük veri platformları ile yüksek hacimli / hızlı akan veriler için uyarlamaktır.

5. AKAN VERİNİN SINIFLANDIRMA GÖREVİ HABERDAR ANONİMLEŞTİRİLMESİ

Veri mahremiyetinin sağlanması için birçok yöntem geliştirilmiştir. Verinin türüne, boyutuna göre anonimleştirme işlemlerinin gereksinimleri de değişmektedir. Örneğin statik veri kümeleri için geliştirilmiş anonimleştirme yöntemleri doğrudan akan veriler üzerinde kullanılamamaktadır.

Veri mahremiyetini sağlamak için sunulan bu yöntemler ile anonimleştirilen veri kümeleri ya da akan veriler daha sonra kullanılmak üzere çeşitli kurum, kuruluş ya da araştırmacılar ile paylaşılmaktadır. Fakat anonimleştirme işleminde sonra verinin kullanılabilirliği azalmaktadır. Anonim veri kümelerinin kullanılabilirliği birçok çalışmada bilgi kaybı adı verilen bir metrik ile ölçülmektedir. Bu metriğe bağlı olarak geliştirilen yöntemlerin öncelikli amacı genel analiz maksatlı olarak bilgi kaybı miktarını minimize etmektir.

Bu bölüm kapsamında, geliştirilen akan veri anonimleştirme yöntemi ile üretilecek anonim veri ile eğitilen sınıflandırma modelinin başarısının korunması hedeflenmektedir.

Bu bölümde öncelikli olarak üretilen anonim verinin kullanılabilirliği baz alınarak geliştirilmiş anonimleştirme yöntemlerinden bahsedilecektir. Sonrasında tez kapsamında geliştirilmiş olan sınıflandırma başarısı öncelikli akan veri anonimleştirme yöntemi (CUDSA) açıklanacaktır. Bu yöntem için kullanılan algoritma verilecek olup, veri kümeleri üzerinde gerçekleştirilen deneyler ve bu deneylere ait sonuçlar sunulacaktır.

5.1 Anonim Verinin Sınıflama Öncelikli Anonimleştirme Yaklaşımları

Üretilen anonim veri kümeleri üçüncü partiler tarafından kullanılmaktadır ve bu anonim veri kümeleri üzerinde oluşan aşırı bilgi kaybı ya da bozulmalar veri kümesinin kullanılabilirliğini azaltmaktadır. Bu problemi dikkate alarak birçok çalışma gerçekleştirilmiştir. Bu bölümde, bir veri kümesinin anonimleştirilmesi

sırasında, üretilecek çıktının kullanılabilirlik açısından kalitesini dikkate alan başlıca anonimleştirme çalışmalarından bahsedilmektedir.

5.1.1 Aşağıdan-yukarıya genelleştirme (Bottom-up generalization)

Aşağıdan-yukarıya genelleştirme (BUG) çalışması (Wang, 2004), anonimleştirme işleminden sonra verinin kullanılabilirliğinin azalmasını motivasyon olarak belirlemiş ve bu probleme çözüm olarak önerilen ilk yaklaşımlardan birisidir.

Bu çalışmada sadece kategorik değerlerin anonimleştirilmesi üzerine bir model önerilmiştir. Her bir kategorik öznitelik için önceden tanımlanmış özniteliğin tanım kümesi içerisindeki değerlerden oluşan bir taksonomi ağacının sisteme verilmesi gerekmektedir. Şekil 2.1'de örnek bir taksonomi ağacı verilmiştir.

Önerilen çözümde, bütün öznitelikler için tanımlanmış taksonomi ağaçlarının en alt seviyelerinden anonimleştirme işlemine başlanılır ve süreç veri kümesi için k-anonimlik sağlanana kadar devam etmektedir. Bu çalışmanın sonuçları incelendiğinde, sınıflandırma başarısı anlamında diğer algoritmalar ile yaklaşık olarak aynı sonucu vermektedir. Fakat ölçeklenebilirlik açısından diğer çalışmalardan daha iyi sonuçlar vermektedir.

5.1.2 Yukarıdan-aşağıya özelleştirme (Top-down specialization)

Bir diğer çalışma olan yukarıdan-aşağıya özelleştirme (TDS) yöntemi (Fung, 2005) üretilen anonim veri kümesi ile eğitilmiş bir sınıflandırma modelinin başarısının, orijinal veri kümesi ile eğitilen modele yakın olmasını hedeflemektedir. Önerilen model, BUG yaklaşımının tam tersidir. Model çalıştırılmaya başlanmadan önce özniteliklerin başlangıç değerleri taksonomi ağaçlarının kök değerleri olarak belirlenmektedir. Taksonomi ağaçları üzerinde yukarıdan aşağıya doğru gidildikçe değerler genelden özele doğru gitmektedir. TDS yaklaşımında, öznitelikler üzerinde özelleştirme yapılarak k-anonimliğin gereksinimlerini karşılayan anonim veri kümesi elde edilmeye çalışılmaktadır. Özelleştirme yapılacak özniteliğe bilgi kazanımı (*information gain*) ve anonimlik kaybı (*anonymity loss*) değerlerine bakılarak karar verilir. Özelleştirme işlemi k-anonimlik gereksinimleri sağlanamadığı duruma kadar devam etmektedir. Sonra bir önceki duruma göre bütün kayıtlar anonimleştirilir.

TDS metodunun sonuçları incelendiğinde, öncesinde raporlanmış diğer sonuçlara göre daha iyi sonuçlar vermektedir. TDS yaklaşımı ile ilgili detaylı bilgi Bölüm 3.2.1'de verilmektedir.

5.1.3 Bilgi tabanlı veri mahremiyeti

Li tarafından geliştirilen çalışma (Li, 2011), veri mahremiyetini korumanın yanı sıra anonimleştirilen verinin sınıflandırma modelleri gibi uygulamalar içinde kullanılabilir kalmasını hedeflemektedir. Bu amaçla, bir öznitelik için genelleştirme işlemine karar verilirken mahremiyet gereksinimlerinden ziyade çıkacak sonucun sınıflandırma kabiliyetine etkisi dikkate alınmaktadır. Hangi özniteliğin genelleştirileceği ile ilgili karar ortak bilgi (*mutual information*) metriği ile seçilmektedir ve amaç sınıflandırma kabiliyetini maksimum seviyede tutmaktır. Bu yöntemde, genelleşme işlemi global olarak bütün veri tabanına uygulanırken, gizleme (*suppression*) işleme lokal boyutta (QI-grup içerisinde) gerçekleştirilmektedir.

5.1.4 Anonimleştirilmiş verinin sınıflandırma modelinde kullanımı

Bu çalışma (Inan, 2009), diğer çalışmalardan farklı olarak bütün anonimleştirme yöntemlerinin içerisine eklenebilecek bir çözüm sunmaktadır. Yöntem, anonimleştirilen veriler ile birlikte çeşitli basit istatistiksel bilgilerin sağlanmasını önermektedir. Örneğin, her bir QI-grup içerisinde sayısal veriler için ortalama ve varyans değerleri verilirken, kategorik veriler için ise her bir değerin grup içerisinde bulunma olasılığı belirtilmektedir. Bu şekilde sağlanan anonim veri kümeleri için önerilen bir SVM (*support vector machine*) yaklaşımı ile anonim verinin kullanılabilir olması hedeflenmektedir.

Yukarıda bahsedilen çalışmalar dışında birçok faklı çalışmada verinin kullanılabilirliği üzerine yöntemler önermiştir (LeFevre, 2008), (Gachanga, 2019), (Majeed, 2019), (Ye, 2013).

Belirtilen bu yöntemlerin hepsi, statik veri kümeleri için uygulanan yöntemlerdir. Akan veri anonimleştirme yöntemlerinde, üretilen anonim veri ile eğitilen bir sınıflandırma modelinin başarısını korumayı hedefleyen bir yöntem bildiğimiz kadarıyla bulunmamaktadır. Bu tez kapsamında sınıflandırma başarısı öncelikli bir akan veri anonimleştirme yöntemi önerilmektedir.

5.2 Sınıflandırma Başarısı Öncelikli Akan Verinin Anonimleştirilmesi

Akan verinin anonimleştirilmesi ile ilgili detaylı bilgi Bölüm 4.2'de verilmiştir. Bu bölümde kullanılacak olan notasyon ile ilgili bilginin bir kısmı da Bölüm 4.3'te verilmiştir. Ek olarak bu bölüm özelinde kullanılacak olan CT ile sınıflandırma çalışmalarında kullanılacak olan veri kümesi içerisinde bulunan sınıflandırma hedef özniteliğini ifade edilmektedir.

Akan veri anonimleştirme algoritmalarının temel amacı orijinal akan veriden (S), üçüncü partiler ile paylaşılabilecek ve içerisinde bulunan mahrem verilerin kime ait olduğu tespit edilemeyecek şekilde anonim bir akan veri (S') elde etmektir.

Bir akan veri anonimleştirme algoritması Tanım 4.3'te belirtilen hususları sağlamak zorundadır. Bu gereksinimlerin sağlanması akan veri mahremiyetini korumak için yeterli olmasına rağmen, üretilen bu anonim akan veri ile eğitilen sınıflandırma modellerinin başarıları oldukça düşüktür.

Problem 5.1 (Sınıflandırma çalışmaları için k-anonim hassas akan veri paylaşımı): Belirlenen bir gecikme kısıtı (δ) ile, k-anonim akan veri paylaşım probleminde herhangi bir zamanda $j = 1,2, \ldots$, aşağıdaki kısıtlar sağlamalıdır:

1. S'_j, k-anonimlik gereksinimlerini sağlanmalıdır.
2. Hiçbir kayıt için gecikme kısıtının (δ) aşılmamalıdır, $\forall t'_i \in S'_j . t'_i . ro - i \leq \delta$
3. Yarı tanımlayıcılar üzerindeki bilgi kaybı en aza indirilir.
4. CT özniteliği için sınıflandırma doğruluğu en yüksek seviyede tutulmalıdır.
5. QI-gruplar için hassas özniteliğin farklılığı maksimum seviyede tutulmalıdır.

Yukarıda problem tanımında iki farklı kısıt tipi mevcuttur: zorunlu (i ve ii), zorunlu olmayan (iii, iv ve v). Zorunlu olmayan kısıtlar aslında optimizasyon hedefidir.
Anonimleştirilen akan veride, yarı tanımlayıcılarının kalitesinin yüksek olması yani bozulmanın az olması istenir. Verinin kalitesi ortalama bilgi kaybı metriği ile hesaplanmaktadır. Ortalama bilgi kaybı Eşitlik (2.1) kullanılarak hesaplanmaktadır.
Ayrıca sınıflandırma işlerine girdi olarak verilen anonim yarı tanımlayıcılar ve hedef öznitelik olarak belirlenen CT için sınıflandırma modelinin doğruluğunun ($Accuracy(f_{CT})$) da yüksek tutulması amaçlanmaktadır. Bu bağlamda tahminleme problemi $f_{CT} : Q'_1 \times Q'_2 \times \ldots \times Q'_n \rightarrow CT$ fonksiyonu ile ifade edilir.
Anonim veri kümesi üzerinden herhangi bir kayda ait hassas bilgi, atak düzenleyecek bir kişi tarafından (attacker), ortaya çıkarılabilir. Hassas bilginin ortaya çıkmasını

engellemek için her bir QI-grup ($qidg$) içerisindeki hassas değerlerin (SV) farklılığının sağlanması gerekir. QI-grup içerisinde SV özniteliği için değerlerin farklılığı entropi metriği (Eşitlik (5.1)) kullanılarak hesaplanmaktadır.

$$H_{qidg}(SV) = -\sum_{sv \in D_{SV}} p_{qidg}(sv) \log(p_{qidg}(sv)) \tag{5.1}$$

Bu eşitlikte, $p_{qidg}(sv)$ ile QI-grup içerisinde bulunan her bir hassas değerin bulunma olasılığı ifade edilmektedir. Anonimleştirilen akan veriler için birden fazla QI-grup oluşmaktadır. Dolayısıyla genel farklılığı QI-gruplar için hesaplanmış $H_{qidg}(SV)$ değerlerinin ortalamasını alarak hesaplıyoruz. Bu işlem Eşitlik (5.2)'de gösterilmektedir.

$$H(SV) = \frac{1}{|QIDG|} \sum_{qidg \in QIDG} H_{qidg}(SV) \tag{5.2}$$

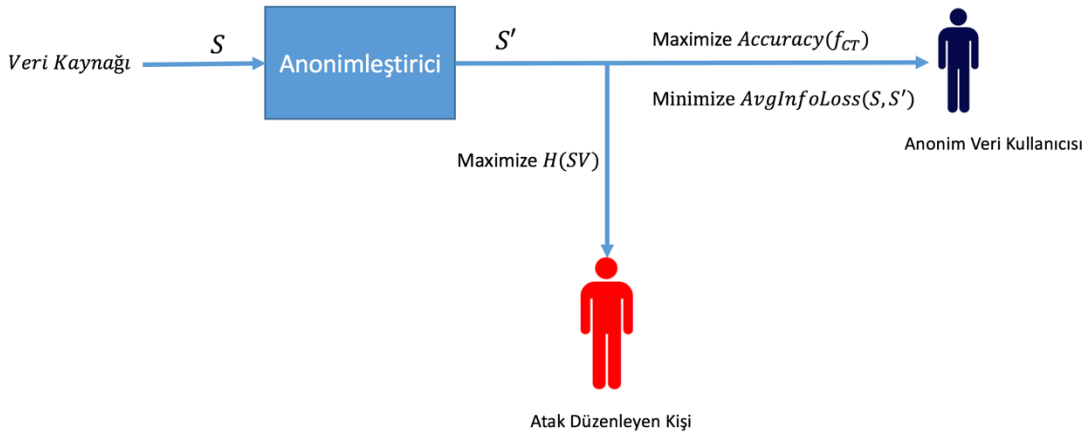
Bu eşitlikte, $QIDG$ ile QI-grup kümesi ifade edilmektedir. $H(SV)$ değerinin yüksek olması durumunda bir kaydın dahil olduğu QI-grup tespit edilse bile o kayda ait hassas bilginin ortaya çıkma olasılığı düşük olacaktır.

Çeşitlilik ölçütümüz, $\ell-$ çeşitlilik (Machanavajjhala, 2007) veya (α, k) −anonimlik (Wong, 2006) anonimleştirme yaklaşımlarında kullanılan formüllere benzese de, temel fark çeşitliliğin azami düzeyde sağlanması gereken fakat zorunlu olmayan bir kısıtlama olmasıdır.

Özetlemek gerekirse, Problem 5.1'de verinin kullanıcısı için bilgi kaybının ($AvgInfoLoss(S, S')$) minimum seviyede tutulduğu, sınıflandırma başarısının ($Accuracy(f_{CT})$) maksimum seviyede tutulduğu ve veriye karşı düzenlenebilecek saldırılara karşı da hassas veride çeşitliliğin ($H(SV)$) maksimize edildiği bir çözüm istenmektedir (Şekil 5.1). Problem açıkça görüldüğü gibi bir çok amaçlı optimizasyon problemidir. Çok-amaçlı optimizasyon yaklaşımlarının temel hedefi karar vericinin tercihlerini modellemektir. Karar vericinin tercihleri ifade etme şekline bağlı olarak çok amaçlı optimizasyon yöntemleri üçe ayrılır (Marler, 2004). Bunlar (i) amaçlar için belirlenecek görece öncelikler karar verici tarafından işlem gerçekleştirilmeden önce belirlenir, (ii) sunulan potansiyel çözüm kümeleri içersinden karar vericinin bir tercih yaptığı durumdur, (iii) yöntemin çalışması sırasında karar vericinin sürekli olarak girdi sağladığı ve tercihlerin bu girdilere paralel olarak

güncellendiği yöntemdir. Tez kapsamında önerilen çok-amaçlı optimizasyon yöntemi için problemin NP-Hard olması ve işlemlerin kısıtlı süre içinde hızlı gerçekleştirilmesi ihtiyacı sebebiyle en uygun seçenek (i) olarak belirlenmiştir.

Problem (5.1)'de Belirtilen bu üç hedefte 0 ile 1 arasında değerler aldığından, çok amaçlı optimizasyon problemini tek amaçlı bir minimizasyon problemine indirgenebilmektedir bunun için Eşitlik (5.3)'te belirtilen metrik kullanılabilir. Ağırlıklar kullanıcının üretilecek anonim akan veri üzerindeki önceliklerine göre belirlenmektedir.



Şekil 5.1 Akan veri anonimleştiricisi için optimizasyon hedefleri

$$\arg\min_{S'} CombinedLoss(S,S') \tag{5.3}$$
$$= ILW \times AvgInfoLoss(S,S')$$
$$+ CW \times \left(1 - Accuracy(f_{CT})\right) + SW \times (1 - H(SV))$$

Eşitlik (5.3) anlaşılabilir bir metrik olamasına rağmen $Accuracy(f_{CT})$'i bileşenini hesaplamak bir sınıflandırma modelini eğitmeyi gerektirir ve ayrıca oldukça sık çalıştırılması gereken bir algoritmayı içerir dolayısıyla pratikte mümkün değildir. Neyse ki, $Accuracy(f_{CT})$ ile her bir QI grup içindeki CT özniteliğinin saflığı (purity) arasında anlamlı bir korelasyon gözlendiği (bkz: deneysel değerlendirme) için bu şekilde maaliyetli hesaplamalardan kaçınabiliriz. Bunun ardındaki mantık, QI-gruplar içerisindeki kayıtların aynı yarı tanımlayıcı değerler ile ifade ediliyor olmasıdır. QI-grup içerisindeki CT değerleri farklıysa, aynı kestirici (*predictor*) değerler farklı hedef

etiketlere sahip olacağından, yüksek doğruluklu bir sınıflandırma modeli eğitmek oldukça zor bir hale gelir. Aksine, her bir QI grubu içindeki CT değerlerinin saflığı sağlandığında hedef etiketler aynı kestirici değerler için aynı olacaktır. Eşitlik (5.4)'te görüldüğü gibi saflık, $Accuracy(f_{CT})$ ile değiştirebilmek için "bir eksi entropi" olarak tanımlanmıştır.

$$Accuracy(f_{CT}) \approx Purity(CT) = 1 - H(CT) \tag{5.4}$$

Eşitlik (5.3)'te bulunan $Accuracy(f_{CT})$'i $1 - H(CT)$ ile ifade ettiğimizde, Eşitlik (5.5)'i elde etmekteyiz.

$$\arg\min_{S'} CombinedLoss(S, S') \tag{5.5}$$
$$= ILW \times AvgInfoLoss(S, S') + CW \times H(CT) + SW \times (1 - H(SV))$$

5.3 Sınıflandırma Başarısı Öncelikli Akan Veri Anonimleştirme Algoritması (CUDSA)

Anonim akan verinin kullanılabilirliğini arttırmak için Problem 5.1'de tanımlanan çok amaçlı akan veri anonimleştirme yaklaşımı: (i) bilgi kaybı miktarını minimize etmek, (ii) CT özniteliğinin saflığını minimize etmek, (iii) hassas özniteliğin entropisini maksimum seviyede tutmak istemektedir. Akan veri çevrimiçi bir şekilde sisteme gelir ve kayıtlar eşik değerinden daha fazla sistemde tutulmamaktadır. Bundan dolayı, hızlı bir çok amaçlı anonimleştirme çözümünün sağlanması gerekir. Diğer bir deyişle, alternatif anonim akan veriler için $CombinedLoss(S, S')$ eşitliğinin hızlı bir şekilde hesaplanması gerekir.

Eşitlik (5.5)'te belirlenen ağırlıklar ile problem için ayarlanabilir (*tunable*) bir çözüm sağlanmaktadır. Önerilen çözümde, uygulama hedeflerine bağlı olarak kullanıcı tarafından önemleri göreceli olarak belirlenir. Ağırlıklar anonimleştirme işlemi başlamadan tanımlanması gereken konfigürasyon parametreleridir.

Çalışmada kümeleme tabanlı bir anonimleştirme yaklaşımı önerilmiştir. Arabellekte bekleyen en az k kayıttan bir küme oluşturulur. Bu amaçla, küme oluştururken $CombinedLoss$ metriğinin değeri minimum seviyede tutulmaya çalışılmaktadır.

Tez kapsamında önerilen çözüm kategorik ve nümerik değerlerin anonimleştirilmesi için kullanılabilmektedir. Fakat kategorik değerler için taksonomi ağaçlarının önceden

tanımlanması gerekmektedir. Örnek bir taksonomi ağacı Şekil 2.1'de verilmiştir. Ayrıca nümerik değerler için ilgili özniteliğin tanım kümesinin önceden belirtilmesi gerekmektedir.

Kümeleme algoritmasına ait detaylar Şekil 5.2'de verilmiştir. k ve δ değerleri zorunlu kısıtlarken, diğer parametereler tez kapsamında önerilerin çözümün getirdiği hiper parametrelerdir. T_{kc} daha sonra tekrar kullanılmak için kaç tane anonimleştirilmiş küme saklanacağına dair bir limittir, ws pencere boyutu ve diğerleri (ILW, CW ve SW) Eşitlik (5.5)'te tanımlanan ağırlık parametreleridir.

Main prosedürü: Sisteme gelen akan veriler *buffer* içerisinde saklanırken, anonimleştirilmiş QI-grupların prototipleri ise *AnonyCls* içerisinde tutulmaktadır. *buffer*'de tutulan kayıt sayısı δ değerine ulaştığında (satır 6), sistemde bulunan en eski kayıt (t_o) için anonimleştirme işlemi *Publish* prosedürü çağrılarak başlar. Anonimleştirilen küme *AnonyCls* listesine eklenir (satır 11). *AnonyCls* içerisinde saklanabilecek anonim küme sayısı T_{kc} ile sınırlandırılmıştır ve *AnonyCls* dolduğunda en eski küme listeden çıkartılır (satır 12-14).

Input: $S, \delta, k, ws, T_{kc}, ILW, CW, SW$
Output: S'

1: $AnonyCls \leftarrow \emptyset$
2: $buffer \leftarrow \emptyset$
3: $j \leftarrow 1$
4: **while** $t_j \in S$ **do**
5: $\quad buffer \leftarrow buffer \bigcup \{t_j\}$
6: $\quad$ **if** $|buffer| \geq \delta$ **then**
7: $\quad\quad t_o \leftarrow$ oldest tuple from $buffer$
8: $\quad\quad buffer \leftarrow buffer \setminus \{t_o\}$
9: $\quad\quad NewCluster \leftarrow$ **Publish**$(t_o, buffer, ws, k, ILW, CW, SW, AnonyCls)$
10: $\quad\quad$ **if** $NewCluster$ is not Empty **then**
11: $\quad\quad\quad AnonyCls \leftarrow AnonyCls \bigcup \{newCluster\}$
12: $\quad$ **if** $|AnonyCls| \geq T_{kc}$ **then**
13: $\quad\quad c_i \leftarrow$ oldest cluster from $AnonyCls$
14: $\quad\quad AnonyCls \leftarrow AnonyCls \setminus \{c_i\}$
15: $\quad j \leftarrow j + 1$

Şekil 5.2 Main prosedürü

Publish prosedürü: Bir kaydın anonimleştirilmesi ile ilgili detaylar Publish prosedürü içerisinde yer almaktadır. Eğer *buffer*'da bulunan toplam kayıt sayısı k'den fazla ise (satır 3), t_o ve diğer kayıtlar arasındaki bilgi kaybı açısından yakınlık hesaplanır ve bu değere göre bir minimum yığın (min-heap) oluşturulur (satır 4-6). Bu yığın yardımı ile yarı tanımlayıcı kümesine dahil olacak kayıtlara karar verebilmek için

TupleSelection prosedürü çağırılır (satır 7). Eğer anonimleştirilmesi gereken kayıt, daha önce anonimleştirilmiş kümelerden birisi ile eşleşirse, bu küme ile *TupleSelection* prosedürü tarafından üretilen küme bilgi kaybı açısından karşılaştırılır ve minimum bilgi kaybına neden olan küme ile ilgili kayıt anonimleştirilir (satır 12-14). Eğer eşleşen bir küme bulunamazsa *TupleSelection* prosedürü tarafından üretilen küme anonimleştirilir (satır 15-17). Diğer bir taraftan, *TupleSelection* prosedürü içerisinde ilgili kayıt (t_o) için bir küme oluşturulamazsa *SuppressOrReuse* prosedürü çağırılır (satır 8-10).

TupleSelection prosedürü: Kayıtlar içerisinden bir küme oluşturmak için TupleSelection prosedürü kullanılmaktadır. İlk adım olarak yeni bir küme oluşturulup içerisine t_o eklenmektedir (satır 2). Eğer minimum yığın içerisindeki kayıt sayı k'den fazla ise kayıt seçme süreci başlar. Pencere boyutu (ws) kadar kayıt minimum heap içerisinde çıkartılır ve *Window*'a eklenir. Çıkartılan her bir kayıt ve yeni oluşturulan küme arasındaki yakınlık Eşitlik (5.5)'te tanımlanan metriğe göre hesaplanır (satır 9-12). Minimum değeri veren kayıt kümeye eklenir ve *Window* içerisinden çıkartılır (satır 13-14). Son eklenen kayıt ile kümenin anonimizasyon seviyesi değiştiği için *Window* içerisinde bulunan her bir kayıt ile küme arasındaki $CombinedLoss$ metriği tekrar hesaplanmaktadır (satır 15). Bu işlem kümedeki eleman sayısı k olana kadar devam etmektedir.

Input: t_o, $buffer$, ws, k, ILW, CW, SW, $AnonyCls$
Output: $NewCluster$

```
1:  MinHeap ← ∅
2:  Window ← ∅
3:  if |buffer| ≥ k then
4:      for each tuple_i ∈ buffer do
5:          IL_i ← CalculateIL(tuple_i, t_o)
6:          MinHeap ← MinHeap ⋃ {(IL_i, tuple_i)}
7:      NewCluster ← TupleSelection(MinHeap, ws, k, ILW, CW, SW)
8:      if NewCluster is Empty then
9:          SuppressOrReuse(t_o, AnonyCls)
10:         return Empty
11:     Find a cluster(C_i) that covers t_o with minimum information loss
12:     if C_i exist AND C_i.IL < NewCluster.IL then
13:         Anonymize t_o with C_i
14:         return Empty
15:     else
16:         Anonymize NewCluster
17:         return NewCluster
18: else
19:     SuppressOrReuse(t_o, AnonyCls)
20:     return Empty
```

Şekil 5.3 Publish prosedürü

Input: $MinHeap, t_o,\ ws,\ k,\ ILW,\ CW,\ SW$
Output: $NewCluster$
1: $NewCluster \leftarrow \emptyset$
2: $NewCluster \leftarrow NewCluster \bigcup t_o$
3: $Window \leftarrow \emptyset$
4: $j \leftarrow 0$
5: $heapSize \leftarrow |MinHeap|$
6: **if** $heapSize < k$ **then**
7: return Empty
8: **while** $j < k$ **do**
9: **while** $|Window| < ws$ **do**
10: $tuple \leftarrow MinHeap.pop()$
11: $loss \leftarrow$ calculate *loss* value for the tuple using Eq. 7
12: Window $\leftarrow$ Window $\bigcup$(tuple, loss)
13: insert the tuple with minimum *loss* value to the NewCluster
14: remove the tuple from Window
15: update loss value between the cluster and tuples in Window
16: $j \leftarrow j + 1$
17: *return* NewCluster

Şekil 5.4 TupleSelection prosedürü

SuppressOrReuse prosedürü: Bu prosedür içerisinde t_o ile öncesinde anonimleştirilen kümeler arasında eşleşen bir küme olup olmadığı kontrol edilir (satır 1). Eğer böyle bir küme bulunursa kayıt bu küme ile anonimleştirilir (satır 3). Aksi durumda bütün yarı tanımlayıcılar için kayıt en üst seviyeden anonimleştirilir.

Input: $t_0,\ AnonyCls$
1: Find a cluster(C_i) in AnonyCls, that match with the tuple t_0
2: **if** C_i exist **then**
3: publish t_0 using C_i
4: **else**
5: suppress t_0

Şekil 5.5 SuppressOrReuse prosedürü

5.4 Deneysel Değerlendirme

Bu bölümde, CUDSA algoritmasının etkinliğini göstermek için gerçekleştirilen deneyler açıklanacak ve sonrasında bu deneylere ait sonuçlar verilecektir. Yapılan deneylerde sonuçlar beş farklı metriğe göre değerlendirilecektir:

1. Bilgi kaybı (IL)
2. Hedef özniteliğin entropisi (CE)
3. Bir eksi hassas özniteliğin entropisi (SE')
4. Sınıflandırma başarısı (CA)

5. Sınıflandırma modelinin doğruluğu ve hedef özniteliğin entropisi arasındaki korelasyon

Bu bölüm içerisinde gerçekleştirilen bütün deneyler Intel Core 2.2GHz CPU ve 16 GB RAM kapasiteli bir kişisel bilgisayar üzerinde çalıştırılmaktadır. Deneyler içerisinde sunulan algoritmalar Java ve Python programlama dilleri kullanarak geliştirilmiştir.

Deneyler gerçekleştirilmeden önce bazı parameterler sabitlenmiştir ve bu parameterler şu şekildedir:

- δ değeri ADULT veri kümesi için 10000, NURSERY veri kümesi için 3000 olarak belirlenmiştir. İlgili değerlere karar verilirken akan veri anonimleştirme çalışmalarında sıklıkla kullanılan değerler tercih edilmiştir. ADULT içerisinde bulunan kayıt sayısı NURSERY'e göre yaklaşık olarak 3 kat daha fazla olduğundan δ için de benzer bir oran uygulanmıştır.
- $T_{kc} = 200$, saklanan anonimleştirilmiş küme sayısı FADS tabanlı yöntemlerde genel olarak 200'e sabitlenmiştir.
- Pencere boyutu $(ws) = 100$ olarak belirlenmiştir. Pencere boyutunun yüksek olması önerilen yöntemin başarısını arttırmasına rağmen, getirdiği ekstra işlem maaliyeti nedeniyle ılımlı (*moderate*) bir değere sabitlenmiştir.

5.4.1 Veri kümeleri

Gerçekleştirilen bütün deneylerde UCI Machine Learning Repository içerisinden alınan ve veri mahremiyeti çalışmalarında sıklıkla kullanılan ADULT (Url-1) ve NURSERY (Url-2) veri kümeleri kullanılmaktadır. ADULT ve NURSERY veri kümeleri ile ilgili detaylı bilgi Bölüm 4.5.1'de verilmiştir. Bu çalışmalar kapsamında bu iki veri kümesine ait kullanılacak yarı tanımlayıcılar, hassas öznitelik ve hedef öznitelik bilgileri Çizelge 5.1'de verilmiştir. ADULT veri kümesi için *Gender* hassas veri olarak kullanılırken, *Income* hedef öznitelik olarak belirlenmiştir. NURSERY veri kümesi içerisinden *Health* ve *Class* öznitelikleri sırasıyla hassas ve hedef öznitelik olarak belirlenmiştir.

ADULT veri kümesi kategorik ve nümerik öznitelikler içerirken, NURSERY veri kümesi sadece kategorik değerlerden oluşmaktadır. Ayrıca bu iki veri kümesi statik olmalarına rağmen, çalışma kapsamında akan veri olarak simüle edilmektedir. Simülasyon esnasında kayıtların sisteme gelme sıraları orijinal veri kümesi içerisindeki kayıtların sıraları ile aynıdır.

Çizelge 5.1 Veri kümelerine ait öznitelik bilgileri.

ADULT - Öznitelikler	**Tanım Kümesi Boyutu**	**NURSERY - Öznitelikler**	**Tanım Kümesi Boyutu**
Age	100	Parents	3
Education	16	Has-nurs	5
Status	7	Form	4
Relationship	6	Children	4
Race	5	Housing	3
Workclass	8	Finance	2
Occupation	14	Social	3
Hours per week	100		
Capital loss	5000		
Nation	41		
Gender	2	***Health***	3
Income	2	***Class***	5

5.4.2 Deney sonuçları

Bu bölümde sunulacak olan deneylerde farklı parametreler kullanılarak, önerilen yöntem farklı açılardan değerlendirilmektedir. Bütün deneyler ADULT ve NURSERY veri kümeleri üzerinde gerçekleştirilmiştir.

Deneylerin daha doğru değerlendirilebilmesi için sonuçlar 4 alt başlık altında sunulacaktır: (1) $CombinedLoss$ metriğinde tanımlanmış ağırlıkların sonuçlar üzerindeki etkisi, (2) ağırlıkların değerlerine karar vermeyi kolaylaştıracak deneyler, (3) sınıflandırma başarısı, (4) hedef özniteliğin entropisi ve sınıflandırma modelinin doğruluğu arasındaki korelasyon.

5.4.2.1 Önceliklendirme ağırlıklarının sonuçları üzerindeki etkisi

Tanımlanan $CombinedLoss$ metriği ile, akan veri mahremiyeti için çok amaçlı ve bu amaçların önceliklendirilebildiği bir model önerilmiştir. Bu işlem $CombinedLoss$ metriği içerisinde bulunan ve anonimleştirme işleminden önce belirlenmesi gereken ağırlıklar (ILW, CW, SW) yardımıyla yapılmaktadır. Bu bölüm içerisinde sunulacak olan deneylerde farklı ağırlıklar kullanılarak, bu ağırlıklardan bilgi kaybı, hassas veri ve hedef özniteliğin entropi değerlerinin nasıl etkilendiği gösterilmektedir. Bu amaçla, ilk olarak ağırlıkların ikili komibnasyonları kullanılarak deneyler yürütülmektedir. Ayrıca deneylerde farklı anonimizasyon seviyelerinin ağırlıklara etkisini de değerlendirebilmek için deneyler farklı k değerleri ile tekrarlanmıştır.

ADULT ve NURSERY veri kümeleri üzerinde dört farklı k değeri ile gerçekleştirilen deneylere ait sonuçlar Şekil 5.6, Şekil 5.7, Şekil 5.8, Şekil 5.9, Şekil 5.10, Şekil 5.11 ve Şekil 5.12'de gösterilmektedir. Deneylerde ikili kombinasyonlar için berlirlenen ağırlığın dışında kalan bileşen için ağırlık 0 olarak sabitlenmiştir. Örneğin, Şekil 5.6(a)'da ILW = 0 ve CW ve SW değerlerinin toplamı her zaman 1'dir. X ekseninde CW'nin değeri artmaktadır ve SW'nin değeri X ekseni boyunca $SW = 1 - CW$ olarak hesaplanmaktadır. Bu şekilde ikili kombinasyonların etkisini daha iyi yorumlayabiliyoruz.

Deney sonuçlarına göre, tanımlanan ağırlıklar üzerinde yapılan değişiklikler ilgili hedefin sonuçlarını doğrudan etkilemektedir. Örneğin hedef öznitelik için belirlenen ağırlığın (CW) arttırılması, üretilen yarı-tanımlayıcı gruplar içerisinde hedef öznitelik altındaki değerlerin entropisinin azalmasına neden olmaktadır. Bu durum bütün k değerlerinde gözlenmektedir. Aynı durum SW' ve ILW değerleri arttığında ilgili hedefler içinde görülmektedir. Genel olarak elde edilen sonuçlar incelendiğinde belirlenen ağırlıklar ile hedefler arasında bir önceliklendirme yapılabildiği açıkca görülmektedir.

5.4.2.2 Ağırlıkların değerlerine karar verilmesi

Ağırlıkların etkilerini nasıl dengeleyeceğimizi anlamak için, her iki veri kümesi için de bu bölümde sunulan ısı haritalarını inceledik. Deneylerde k değeri sabit tutulurken, farklı ağırlıklar ile deneyler gerçekleştirilmiştir. Deney sonuçlarına ait ısı haritaları Şekil 5.14 ve Şekil 5.15'te gösterilmektedir. Sonuçlarda sunulan "Balanced" sütünu, CE, IL ve SE değerlerinin 0 ile 1 arasına normalleştirildikten sonra ortalaması alınarak hesaplanmıştır. Normalizasyon işlemi her bir hedef için kendi içlerinde gerçekleştirilmiştir.

Her iki ısı haritasında da bir ağırlığın arttırılması, ilgili performansın iyileşmesini sağlamaktadır. "Balanced" sütununda da en iyi sonuçlar ağırlıklar dengeli olarak belirlendiğinde görülmektedir. Sonuçlar, yaklaşımımızın ve algoritmamızın, üretilecek anonim akan verinin kullanıcı tercihine bağlı olarak üç hedef üzerinde ayarlanabilir olduğunu doğrulamaktadır.

5.4.2.3 Sınıflandırma deneyleri

Bu çalışmanın ana amacı snıflandırma başarısını arttırmaktır. Bu amaçla üretilen anonim akan veriler ile beslenen farklı sınıflandırma modellerinin başarıları karşılaştırılmaktadır. Sınıflandırma algoritması olarak *Decision Tree* ve *Random Forest* kullanılmaktadır. Modeller 10 katmanlı çapraz geçerlilik (*10-fold cross validation*) kullanılarak modeller üretilmiş ve değerlendirilmiştir. ADULT veri kümesi üzerinde *Salary* hedef öznitelik olarak belirlenmiştir ve bu veri kümesi bir ikili sınıflandırma (*binary classification*) örneği olacaktır. NURSERY veri kümesi çok sınıflı (*multi-class*) sınıflandırma örneğinin test edilebilmesi için önemlidir. Bu veri kümesi üzerinde *Class* hedef öznitelik olarak belirlenmiştir.

Anonim akan veride sınıflandırma doğruluğunu, CW ağırlığı için daha yüksek değerler belirleyerek artırmayı umuyoruz. Bunun nedeni, bu şekilde oluşturulacak QI-gruplarda hedef öznitelik için saflığın korunacak olmasıdır. Farklı CW değerleri ile yapılan deneylere ait sonuçlar Şekil 5.16, Şekil 5.17, Şekil 5.18 ve Şekil 5.19'de verilmiştir.

Sonuçlar incelendiğinde, CW'nin artması ile genellikle sınıflandırma doğruluğu da artma eğilimindedir ve bu durum çalışma için belirlenen motivasyonun faydasını gösterir.

5.4.2.4 Hedef özniteliğin entropisi ve sınıflandırma başarısı arasındaki korelasyon

Hedef özniteliğin entropisine yönelik farklı ağırlık konfigürasyonları ile üretilen anonim veri kümeleri kullanılarak deneyler gerçekleştirilmektedir. Konfigürasyonlar ile ilgili detaylı bilgi Çizelge 5.2'de verilmiştir.

Çizelge 5.2 Korelasyon deneylerinde kullanılan ağırlık konfigürasyonları

CW	SW	ILW
0	1.0	0
0.25	0.75	0
0.50	0.50	0
0.75	0.25	0
1.0	0	0
0	0	1.0
0.25	0	0.75
0.50	0	0.50
0.75	0	0.25
1.0	0	0

Bu bölümde, üretilen anonim akan verinin üzerinde hedef özniteliğin entropisi ve bu anonim akan veri ile eğitilen sınıflandırma modelinin başarısı arasındaki korelasyon incelenmektedir. Bu deneylere ait sonuçlar Çizelge 5.3 ve Çizelge 5.4'te verilmiştir.

Çizelge 5.3 Hedef özniteliğin entropisi ve sınıflandırma modelinin başarısı arasındaki korelasyon (ADULT veri kümesi).

k	**Decision Tree**	**Random Forest**
50	-0.97	-0.99
100	-0.87	-0.98
150	-0.88	-0.92
200	-0.85	-0.90

Çizelge 5.4 Hedef özniteliğin entropisi ve sınıflandırma modelinin başarısı arasındaki korelasyon (NURSERY veri kümesi).

k	**Decision Tree**	**Random Forest**
25	-0.58	-0.33
50	-0.87	-0.89
75	-0.75	-0.80
100	-0.82	-0.86

Özellikle ADULT veri kümesi üzerinde yapılan deneylerin sonuçlarına göre, iki metrik arasında yüksek bir korelasyon bulunmaktadır. k değerinin artması ile korelasyon üzerinde sınıflandırma algoritmasının doğruluğunun düşmesinden kaynaklı azalma görülmektedir. Bu da k değerinin artması ile örtüşen QI-grupların sayısının artmasından kaynaklanmaktadır.

5.4.2.5 Literatürdeki yöntemler ile karşılaştırma

Bu bölümde tez kapsamında önerilen CUDSA yöntemi ile popüler akan veri anonimleştirme yöntemleri, bilgi kaybı ve sınıflandırma modelinin başarısı açısından karşılaştırılacaktır. CUDSA yöntemine ait 3 farklı konfigürasyon deney sonuçlarında sunulmaktadır. Konfigürasyonlara ait detaylar şu şekildedir:

- **CUDSA1**: CW = 1, SW = 0 ve ILW = 0
- **CUDSA2**: CW = 0.75, SW = 0 ve ILW = 0.25
- **CUDSA3**: CW = 0.50, SW = 0 ve ILW = 0.50

Bu bölümde sunulacak deneylerde bilgi kaybı miktarının sınıflandırma modelinin başarısı üzerinde kritik bir role sahip olduğu için belirlenen konfigürasyonlarda SW değeri 0 olarak sabitlenmiştir. Deneylerde bilgi kaybı ve hedef özniteliğin entropisi için belirlenmiş ağırlıkların değişiminden sınıflandırma modelinin nasıl etkilendiği gösterilmektedir. Bu deneylere ait sonuçlar Şekil 5.20 ve Şekil 5.21'de sunulmuştur.

Ayrıca anonimleştirilen akan veriler ile eğitilen sınıflandırma modellerinin başarısı anonimleştirme seviyesine bağlı olarak azalmaktadır.

Karşılaştırılan sonuçlardaki istatistiksel farkı anlamak için, FADS ve bizim algoritmamız "paired t-test" kullanarak bilgi kaybı ve sınıflandırma doğruluğu açısından karşılaştırılmıştır. Yapılan teste ait sonuçlar stiksel olarak anlamlı değildir.

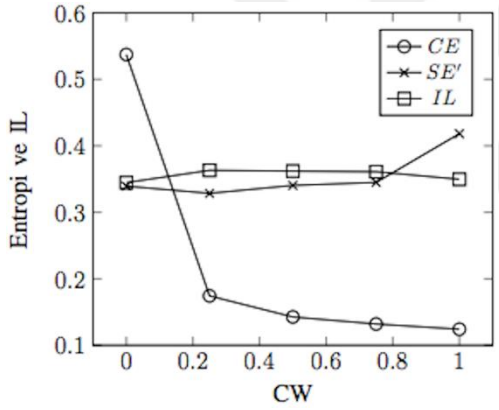
Çizelge 5.5'te verilmiştir. Sonuçlar, algoritmamızın her üç konfigürasyonda da, sınıflandırma doğruluğu açısından FADS yönteminden istatistiksel olarak önemli ölçüde daha iyi olduğunu göstermektedir. Bununla birlikte, FADS'ın bilgi kaybı sonuçları sadece CUDSA1'in sonuçlarıyla karşılaştırıldığında istatistiksel olarak anlamlıdır, yani CUDSA2 ve CUDSA3'e karşı sonuçlar istatistiksel olarak anlamlı değildir.

Çizelge 5.5 paired t-test sonuçları (anlamlılık düzeyi 0.01 olarak belirlenmiştir)

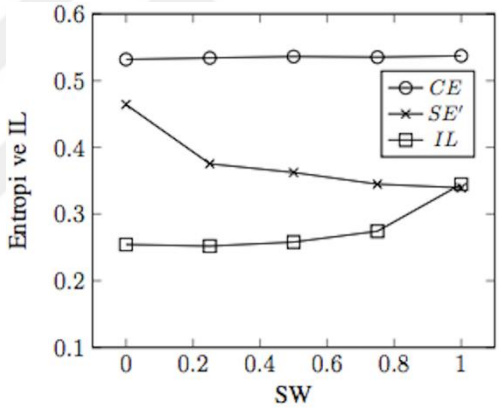
Yöntem	$p-değeri\ (IL)$	$p-değeri$ (sınıflandırma doğruluğu)
FADS-CUDSA1	0. 002426	0.000021
FADS-CUDSA2	0. 044523	0. 004816
FADS-CUDSA3	0. 152091	0. 009792

5.5 Değerlendirmeler

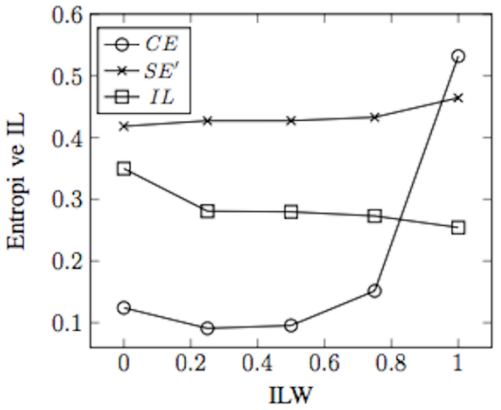
Akan veriler için anonimleştirme yöntemlerinin, statik veriler için geliştirilmiş yöntemlere kıyasla daha fazla kısıtı dikkate almaları gerekmektedir. CASTLE, FAANST ve FADS gibi akan verilerin anonimleştirilmesi için geliştirilmiş birçok yöntem bulunmaktadır. Bu çalışmalarda akan veri anonimizasyonu sırasında bilgi kaybı miktarı minimum seviyede tutulmaya çalışılmaktadır. Üretilen anonim veri ile üretilecek bir sınıflandırma modelinin başarısı dikkate alınmamaktadır. Bu çalışmada yeni bir akan veri anonimleştirme yaklaşımı sunulmaktadır. Bu yöntemde önceden belirlenmiş konfigürasyonlara göre akan verinin anonimizasyonu sağlanmaktadır. Üç başlık altında toplanabilecek olan bu konfigürasyonlar ile üretilecek anonim verinin bilgi kaybı miktarı, oluşacak QI-gruplar içinde hassas verilerin farklılığı ve üretilen anonim akan veri ile beslenmiş sınıflandırma modelinin başarısı arasında bir önceliklendirme yapılabilmektedir.



(a) CW + SW = 1 ve ILW = 0

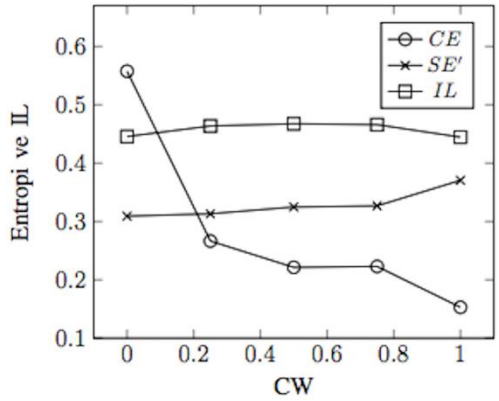


(b) SW + ILW = 1 ve CW = 0

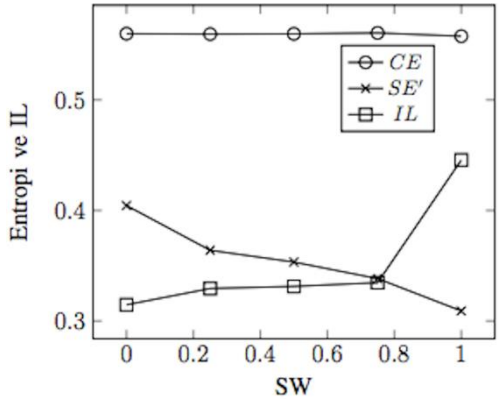


(c) CW + ILW = 1 ve SW = 0

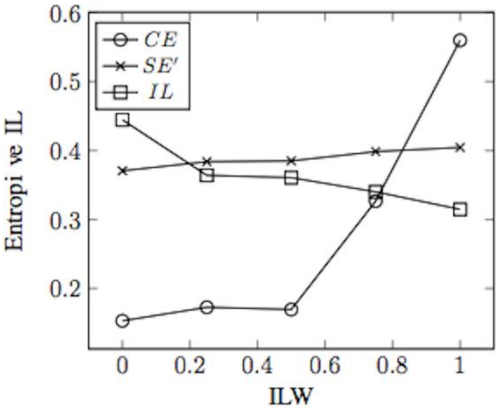
Şekil 5.6 Ağırlık değişiminin bilgi kaybı ve entropi üzerindeki etkisi (ADULT veri kümesi, k = 50)



(a) CW + SW = 1 ve ILW = 0

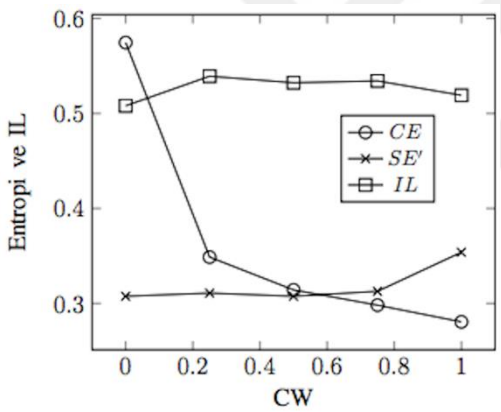


(b) SW + ILW = 1 ve CW = 0

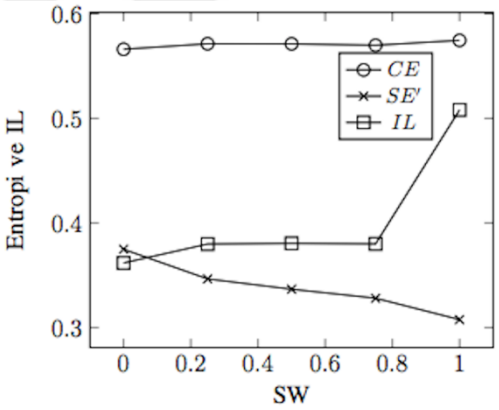


(c) CW + ILW = 1 ve SW = 0

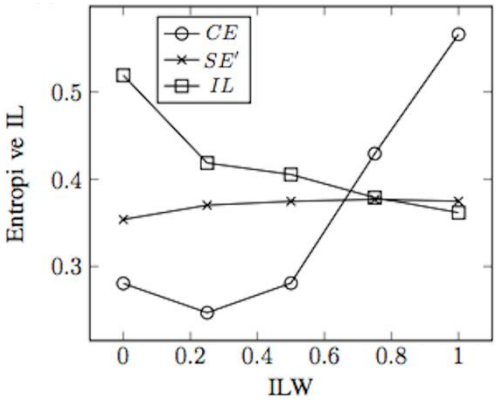
Şekil 5.7 Ağırlık değişiminin bilgi kaybı ve entropi üzerindeki etkisi (ADULT veri kümesi, k = 100)



(a) CW + SW = 1 ve ILW = 0

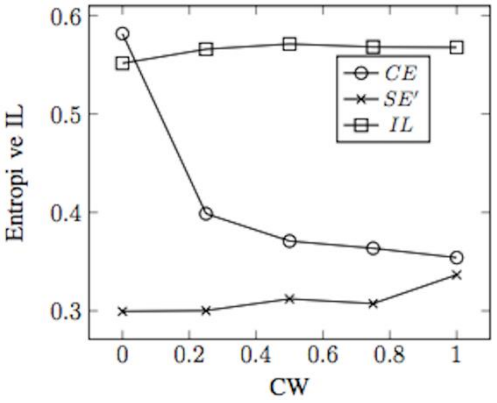


(b) SW + ILW = 1 ve CW = 0

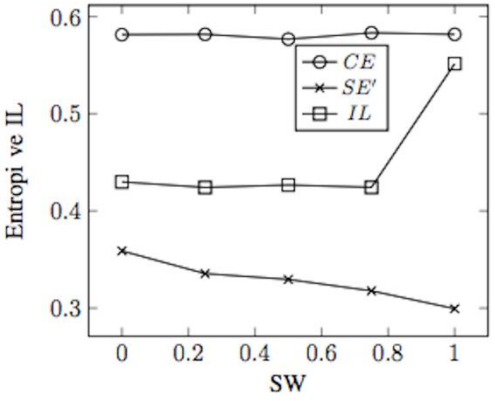


(c) CW + ILW = 1 ve SW = 0

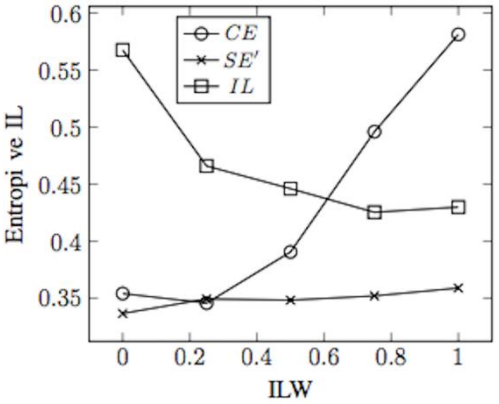
Şekil 5.8 Ağırlık değişiminin bilgi kaybı ve entropi üzerindeki etkisi (ADULT veri kümesi, k = 150)



(a) CW + SW = 1 ve ILW = 0

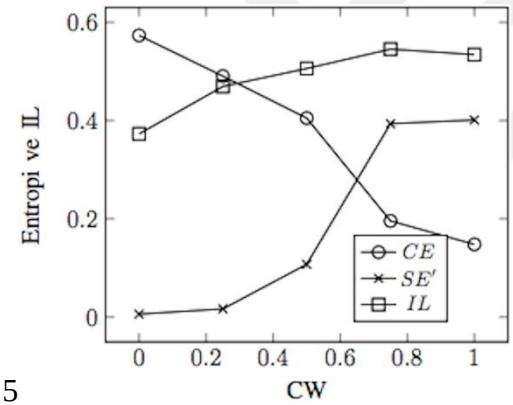


(b) SW + ILW = 1 ve CW = 0



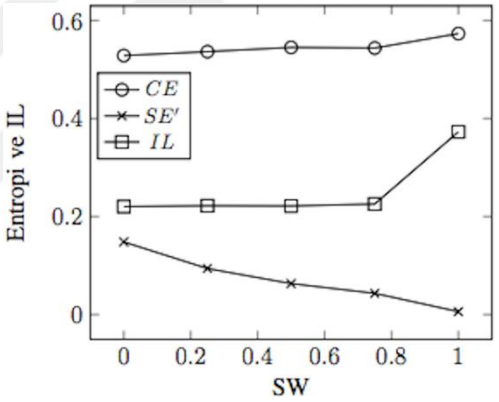
(c) CW + ILW = 1 ve SW = 0

Şekil 5.9 Ağırlık değişiminin bilgi kaybı ve entropi üzerindeki etkisi (ADULT veri kümesi, k = 200)

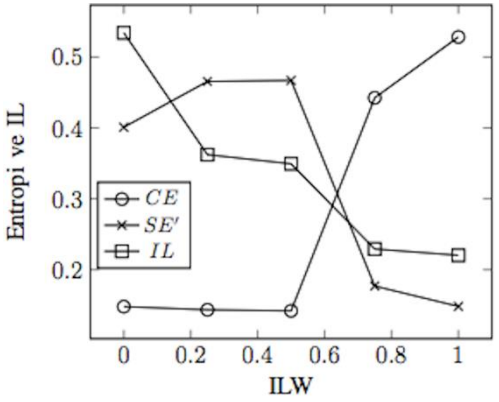


5

(a) CW + SW = 1 ve ILW = 0

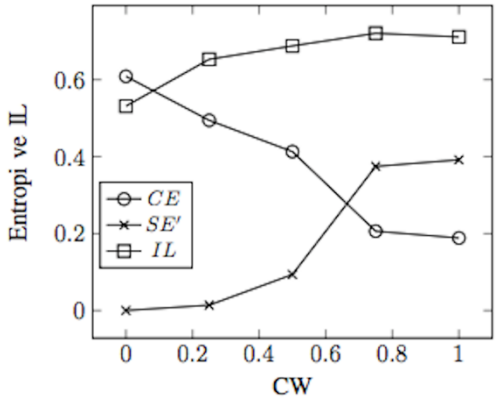


(b) SW + ILW = 1 ve CW = 0

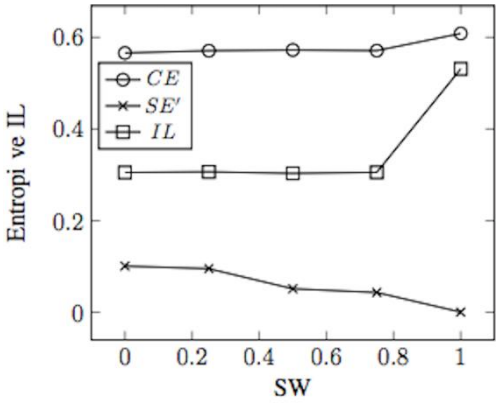


(c) CW + ILW = 1 ve SW = 0

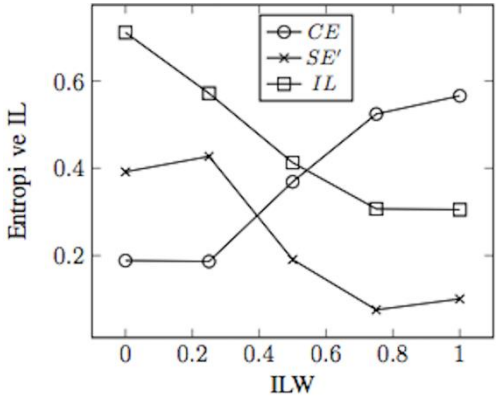
Şekil 5.10 Ağırlık değişiminin bilgi kaybı ve entropi üzerindeki etkisi (NURSERY veri kümesi, k = 25)



(a) CW + SW = 1 ve ILW = 0

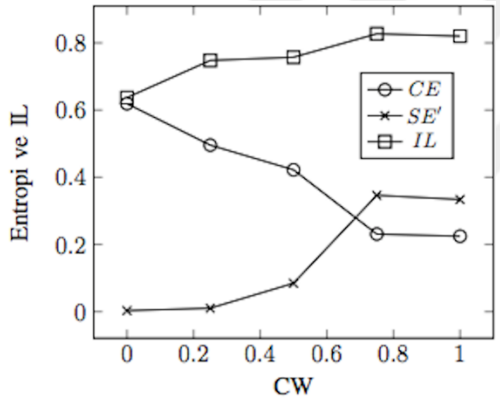


(b) SW + ILW = 1 ve CW = 0

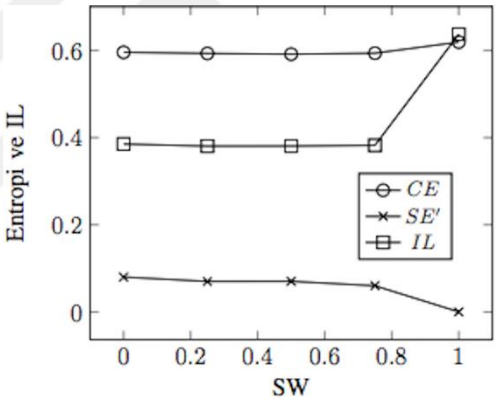


(c) CW + ILW = 1 ve SW = 0

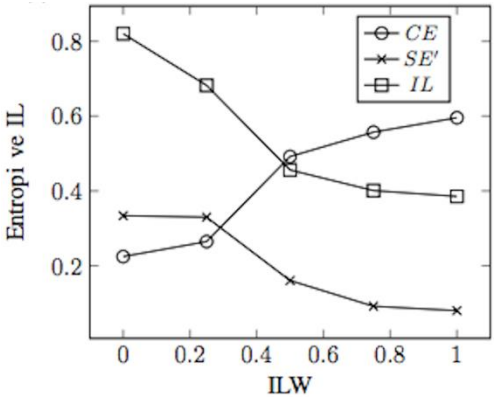
Şekil 5.11 Ağırlık değişiminin bilgi kaybı ve entropi üzerindeki etkisi (NURSERY veri kümesi, k = 50)



(a) CW + SW = 1 ve ILW = 0

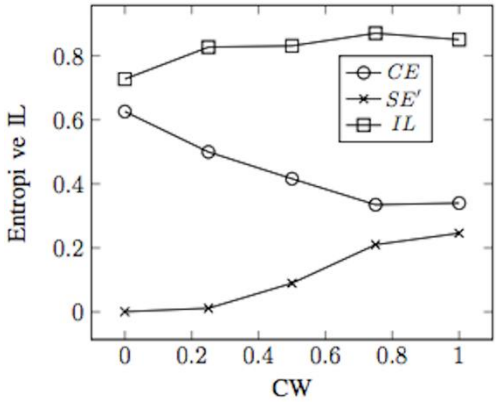


(b) SW + ILW = 1 ve CW = 0

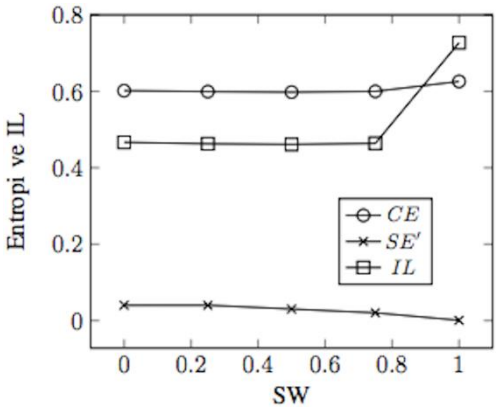


(c) CW + ILW = 1 ve SW = 0

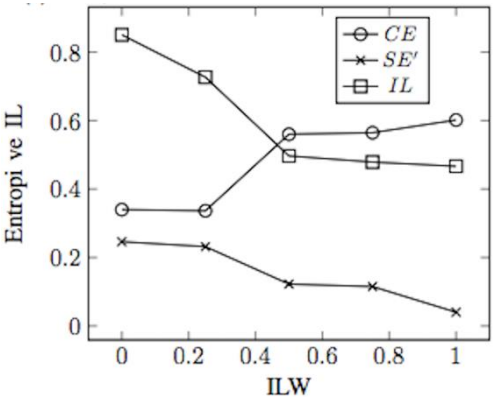
Şekil 5.12 Ağırlık değişiminin bilgi kaybı ve entropi üzerindeki etkisi (NURSERY veri kümesi, k = 75)



(a) CW + SW = 1 ve ILW = 0



(b) SW + ILW = 1 ve CW = 0



(c) CW + ILW = 1 ve SW = 0

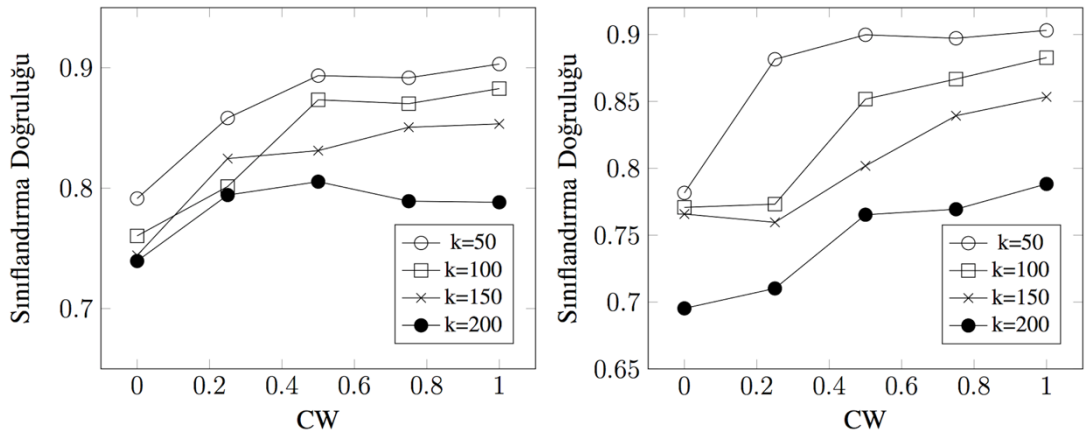
Şekil 5.13 Ağırlık değişiminin bilgi kaybı ve entropi üzerindeki etkisi (NURSERY veri kümesi, k = 100)

CW	ILW	SW	CE	IL	SE'	Balanced
0,25	0,25	0,5	0,208300817	0,361577398	0,34032053	0,256720753
0,5	0,25	0,25	0,1814389	0,359453387	0,354926475	0,281182374
0,25	0,5	0,25	0,274257813	0,340943698	0,359866003	0,333980635
0,5	0,5	0	0,169396073	0,360634335	0,384943898	0,378853981
0,75	0,25	0	0,172629042	0,364077028	0,383795824	0,384994906
0,25	0	0,75	0,266520223	0,463692111	0,313242445	0,432443164
0,5	0	0,5	0,221424812	0,46744546	0,324818319	0,44422158
0,75	0	0,25	0,222981079	0,465943166	0,326715252	0,448848788
0	0,25	0,75	0,560564018	0,334664601	0,338360006	0,479245466
1	0	0	0,15298319	0,444571905	0,370594607	0,498385054
0,25	0,75	0	0,326667883	0,340109956	0,398454976	0,50993578
0	0,5	0,5	0,559721062	0,331307323	0,353308106	0,523491948
0	0,75	0,25	0,559509087	0,329396104	0,363967247	0,556414555
0	0	1	0,557585648	0,445608734	0,309111043	0,61659118
0	1	0	0,559864398	0,314654567	0,404455074	0,666094494

Şekil 5.14 Ağırlıkların etkilerinin ısı haritası üzerindeki gösterimi (ADULT veri kümesi, k = 100)

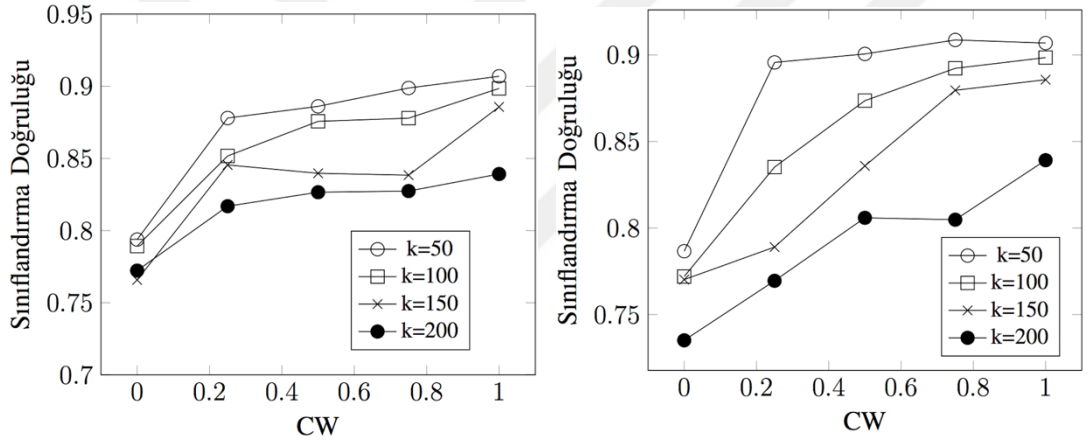
CW	ILW	SW	CE	IL	SE'	Balanced
0,25	0,5	0,25	0,5263	0,311	0,0196	0,289471399
0,25	0,25	0,5	0,5277	0,329	0,0113	0,298473041
0,5	0,25	0,25	0,4382	0,3758	0,0604	0,303508507
0,25	0,75	0	0,5243	0,3071	0,0758	0,328668889
0	0,25	0,75	0,5711	0,3055	0,0432	0,338896863
0	0,5	0,5	0,5725	0,3033	0,0513	0,344571528
0	1	0	0,566	0,3052	0,1008	0,379614733
0	0,75	0,25	0,5709	0,3065	0,0951	0,380071879
0,5	0,5	0	0,3691	0,413	0,191	0,380641357
0	0	1	0,6086	0,5313	0,0004	0,515456506
0,25	0	0,75	0,4944	0,6529	0,0142	0,533161376
0,75	0,25	0	0,1866	0,5717	0,4272	0,547727454
0,5	0	0,5	0,413	0,6877	0,0939	0,558908298
1	0	0	0,1886	0,711	0,3919	0,633008592
0,75	0	0,25	0,2063	0,7206	0,3747	0,641224677

Şekil 5.15 Ağırlıkların etkilerinin ısı haritası üzerindeki gösterimi (NURSERY veri kümesi, k = 50)



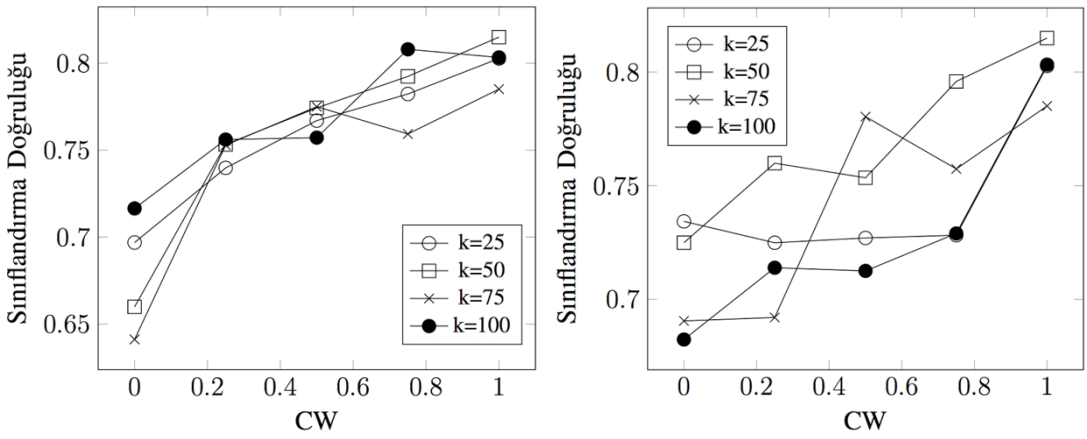
(a) CW + SW = 1 ve ILW = 0 (b) CW + ILW = 1 ve SW = 0

Şekil 5.16 Decision Tree algoritmasının, CUDSA ile anonimleştirilen ADULT veri kümesi üzerindeki doğruluğu



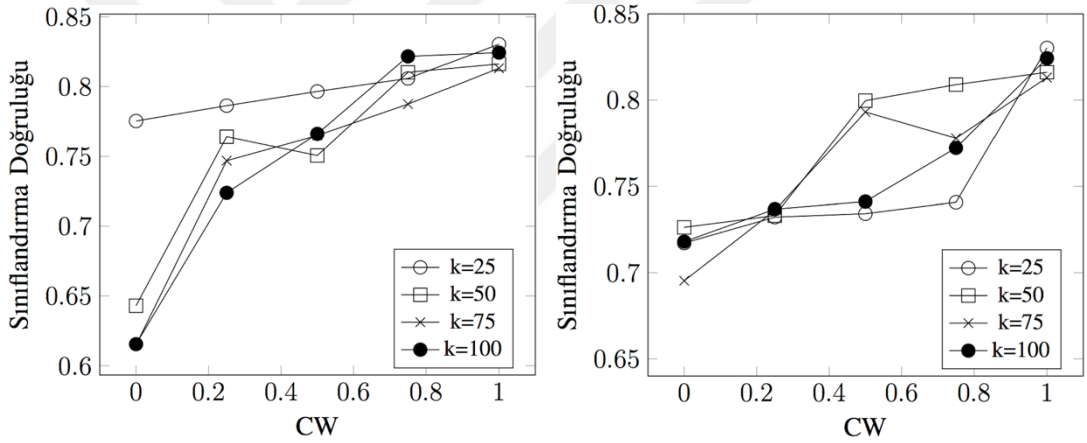
(a) CW + SW = 1 ve ILW = 0 (b) CW + ILW = 1 ve SW = 0

Şekil 5.17 Random Forest algoritmasının, CUDSA ile anonimleştirilen ADULT veri kümesi üzerindeki doğruluğu



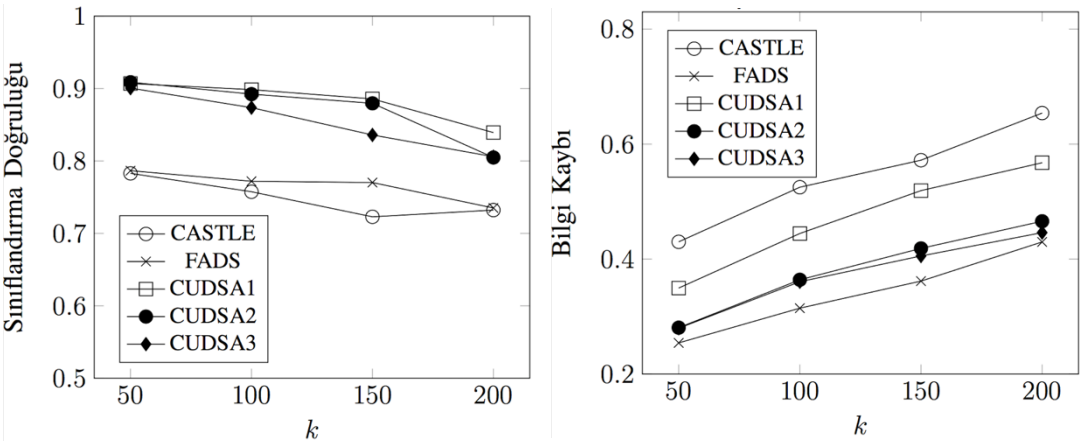
(a) CW + SW = 1 ve ILW = 0 (b) CW + ILW = 1 ve SW = 0

Şekil 5.18 Decision Tree algoritmasının, CUDSA ile anonimleştirilen NURSERY veri kümesi üzerindeki doğruluğu



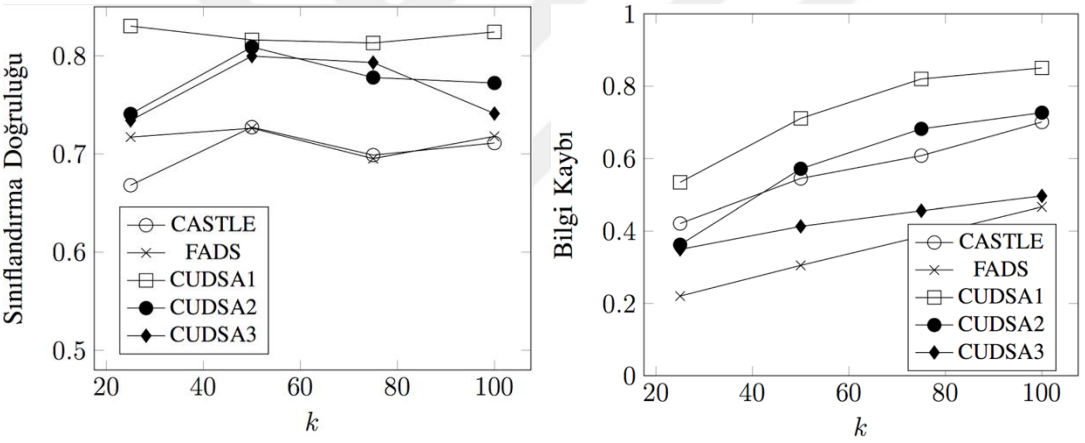
(a) CW + SW = 1 ve ILW = 0 (b) CW + ILW = 1 ve SW = 0

Şekil 5.19 Random Forest algoritmasının, CUDSA ile anonimleştirilen NURSERY veri kümesi üzerindeki doğruluğu



(a) CW + SW = 1 ve ILW = 0 (b) CW + ILW = 1 ve SW = 0

Şekil 5.20 Akan veri anonimleştirme algoritmalarının bilgi kaybı ve sınıflandırma başarısı açısından karşılaştırılması (ADULT veri kümesi)



(a) CW + SW = 1 ve ILW = 0 (b) CW + ILW = 1 ve SW = 0

Şekil 5.21 Akan veri anonimleştirme algoritmalarının bilgi kaybı ve sınıflandırma başarısı açısından karşılaştırılması (NURSERY veri kümesi)

Önerilen yöntem olan CUDSA ile literatürde iyi bilinen iki akan veri anonimleştirme yöntemi olan FADS ve CASTLE ile kıyaslanmıştır. Yöntemler ADULT ve NURSERY veri kümeleri üzerinde çalıştırılmıştır. Üretilen anonim akan verinin bilgi kaybı miktarı ve bu anonim akan veri ile beslenen sınıflandırma modellerinin doğruluğu üzerinden yöntemler karşılaştırımıştır. Sonuçlara göre CUDSA yöntemi ile anonimleştirilen veri kümeleri ile beslenen sınıflandırma modellerinin başarısı diğer yöntemlere göre istatistiksel olarak önemli derecede daha iyi sonuçlar vermektedir. Fakat, FADS yöntemi bilgi kaybı açısından CUDSA yaklaşımına göre daha iyi sonuçlar vermektedir. Gerçekleştirilen bazı deneyler içerisinde k değeri ADULT için

100 ve NURSERY için 50 olarak sabitlenmiştir. k değeri üzerinde yapılacak ufak değişiklikler bilgi kaybı açısından deney sonuçlarını büyük oranda etkileyebilir. Fakat elde edilen sonuçların deseni bakımından, büyük değişiklikler gözlenmeyecektir.

Bir akan verinin anonimleştirilme sebebi, bu verinin üçüncü partiler tarafından kullanılmasının istenmesidir. Dolayısıyla, bu yöntemin diğer akan veri anonimleştirme yöntemlerine göre daha tercih edilebilir bir yöntem olacağı öngörülmektedir.

6. SONUÇLAR ve ÖNERİLER

Tez kapsamında büyük veri ve akan veri mahremiyetinin korunması için üç yeni yöntem önerilmiştir. Bu yöntemler literatürde bulunan önemli çalışmalar ile çeşitli deneyler yapılarak karşılaştırılmıştır ve bu deney sonuçları incelendiğinde önerilen yöntemler karşılaştırıldıkları yöntemlere göre motivasyonları özelinde çoğunlukla üstünlük sağlamıştır.

Büyük verinin mahremiyetinin sağlanması için önerilen anonimleştirme çözümünde, verinin bir bilgisayar kümesi üzerine dağıtılarak TDS algoritmasının ihtiyaç duyduğu matematiksel işlemlerin dağıtık veri üzerinde yapılması ve elde edilen sonuçların ana makine üzerinde toplanıp TDS yaklaşımı ile anonimleştirme işlemi gerçekleştirilir. Bilgisayar kümesi üzerinde verinin dağıtılması ve işlenmesi Apache Spark yardımı ile gerçekleştirilmiştir. Önerilen yöntem ile anonimleştirme işlemini ölçeklenebilir bir şekilde gerçekleştirilebilmektedir.

Akan verinin mahmremiyetini korumak için dikkate alınması gereken kısıtlar toplu veri kümeleri için geliştirilen mahremiyet çalışmalarında uyulan kısıtlara göre daha fazladır. Bunun için akan verinin anonimizasyonu daha zorlu bir problem olarak gözükmektedir. En önemli kısıtlardan birisi anonimleştirilmek üzere sisteme gelen bir kaydın sistemde sınırlı bir süre sistemde tutulabilmesidir. Sistemde verinin uzun süre tutulması verinin yaşlanmasına yani önemini kaybetmesine neden olmaktadır, ayrıca sisteme gelecek kayıt miktarı ile ilgili bir bilgiye sahip olunmadığı ve sistem sınırlı bir kaynağa sahip olduğu için kayıtlar sistemde uzun süre bekletilememektedir. Bu nedenle, geliştirilen yöntemlerde kayıtlar için önceden bir gecikme kısıtı belirlenmektedir. Fakat gecikme kısıtının düşük tutulması bilgi kaybı miktarını arttırırken, yüksek tutulması verinin yaşlanmasına neden olmaktadır. Veri için gecikme ve bilgi kaybı arasında negatif korelasyon bulunmaktadır. Tez kapsamanında önerilen yöntemde sisteme gelen kayıtların ortalama gecikme sürelerini ve bilgi kaybı miktarını minimum seviyede tutmak hedeflenmiştir. Ayrıca geliştirilen yöntem üzerinde yapılan deneylerde bilgi kaybı ve ortalama gecikme arasında bir denge

noktası olabileceği gösterilmiştir. Elde edilen sonuçlar ışığında bu yöntem akan veri anonimleştirme çalışmalarında tercih edilecek bir yöntem olcaktır.

Anonimleştirilen bir veri kümesi ile beslenen öğrenme algoritmalarının başarısı gerçek veri kümesi ile beslenen yönteme göre düşmektedir. Bu nedenle statik veri kümeleri için geliştirilen birçok anonimleştirme çalışması, anonimizasyon sırasında öğrenme algoritmalarının başarısını dikkate almakta ve anonimleştirme işlemlerini bu doğrultuda yapmaktadırlar. Fakat akan veri için geliştirilen yöntemler arasında bu yönde bir çalışma literatürde bulunmamaktadır. Tez kapsamında önerilen üçüncü yöntemde akan verinin anonimizasyonu, üretilecek anonim akan veri ile eğitilecek bir sınıflandırma modelinin başarısı gözetilerek gerçekleştirilmektedir. Önerilen yöntemde bilgi kaybı, sınıflandırma algoritmalarının başarısı ve hassas verinin farklılığı arasında önceliklendirme yapılabilecek bir yaklaşım sunulmaktadır. Akan verinin önerilen yöntem ve popüler akan veri anonimleştirme yaklaşımları kullanılarak anonimizasyonu sağlanmıştır. Elde edilen anonim akan veri ile beslenen sınıflandırma modellerinin başarıları incelendiğinde tez kapsamında önerilen yöntem ile anonimleştirilen veri kümeleri kullanılarak eğitilen sınıflandırma algoritmaları istatistiksel olarak daha iyi sonuçlar vermektedir. Anonim verinin sınıflandırma algoritmalarındaki başarısının önemsendiği durumlarda tercih edilebilecek bir çözüm olacaktır.

6.1 Gelecekteki Çalışmalar için Öneriler

Tez kapsamında sunulan çözümler için çeşitli iyileştirmeler ve ilave geliştirmeler yapılabilir durumdadır. Bunlar arasında:

1. Belirli sıklıklar ile verinin sürekli gönderildiği bir simülasyon üzerinde akan veri anonimleştirme yöntemleri için testler tekrarlanabilir durumdadır.
2. Kategorik öznitelikler için hazırlanan taksonomi ağaçlarının sınıflandırma algoritmalarının başarısını nasıl etkilediği ile ilgili detaylı deneysel değerlendirme mümkündür.
3. Büyük veri için yapılan deneyler literature kazandırılması muhtemel daha büyük veri kümeleri ile tekrarlanabilir.

4. Önerilen yöntemlerle anonimleştirilen veri kümeleri ile üretilen regresyon modellerinin başarıları incelenip, sınıflama yanında regresyon algoritmalarına yönelik olarak algoritmik optimizasyonlar hedeflenmektedir.

KAYNAKLAR

Aggarwal, C. C., (2005). On k-anonymity and the curse of dimensionality, *VLDB*, 5, 901-909.

Agmon Ben-Yehuda, O., Ben-Yehuda, M., Schuster, A., Tsafrir, D., (2014), The rise of RaaS: the resource-as-a-service cloud, *Communications of the ACM*, 57 (7), 76-84.

Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J., (2002), Models and issues in data stream systems. *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (Sf. 1-16).

Bayardo, R. J., Agrawal, R., (2005). Data privacy through optimal k-anonymization, *IEEE 21st International conference on data engineering* (ICDE'05), (Sf. 217-228).

Bertino, E., Ooi, B. C., Yang, Y., Deng, R. H., (2005), Privacy and ownership preserving of outsourced medical data, *IEEE 21st International Conference on Data Engineering (ICDE'05)*, (Sf. 521-532).

Brickell, J., Shmatikov, V., (2008), The cost of privacy: destruction of data-mining utility in anonymized data publishing. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, (Sf. 70-78).

Bu, Y., Fu, A. W., Wong, R. C., Chen, L., Li, J., (2008), Privacy preserving serial data publishing by role composition, *Proceedings of the VLDB Endowment*, 1 (1), 845-856.

Bu, Y., Howe, B., Balazinska, M., Ernst, M. D., (2010), HaLoop: Efficient iterative data processing on large clusters, *Proceedings of the VLDB Endowment*, 3 (1-2), 285-296.

Cao, J., Carminati, B., Ferrari, E., Tan, K.-L., (2010), Castle: Continuously anonymizing data streams, *IEEE Transactions on Dependable and Secure Computing*, 8 (3), 337-352.

Chen, K., Sun, G., Liu, L., (2007), Towards attack-resilient geometric data perturbation, *SIAM international conference on Data mining* (Sf. 78-89).

Chen, M., Mao, S., Liu, Y., (2014)**,** Big data: A survey. Mobile networks and applications, 19 (2), 171-209.

Chester, S., Srivastava, G., (2011), Social network privacy for attribute disclosure attacks. *International Conference on Advances in Social Networks Analysis and Mining* (s. 445-449), IEEE.

Chiusano, S.A., Ruiz, E.M., Scurti, M., (2019), Data-Driven Analysis to Improve Oncological Processes in Hospital (doktora tezi).

Domingo-Ferrer, J., Gonzalez-Nicolas, U., (2010), Hybrid microdata using microaggregation, Information Sciences, 180, 2834-2844.

Domingo-Ferrer, J., Mateo-Sanz, J. M., (2002), Practical data-oriented microaggregation for statistical disclosure control, IEEE Transactions on Knowledge and data Engineering, 14, 189-201.

Dwork, C., (2008), Differential privacy: A survey of results, *International conference-on theory and applications of models of computation* (Sf. 1-19).

Fung, B. C., Wang, K., Yu, P. S., (2005), Top-down specialization for information and privacy preservation, *IEEE 21st international conference on data engineering (ICDE'05)*, (Sf. 205-216).

Gachanga, E., Kimwele, M., Nderu, L., (2019). Feature Based Data Anonymization with Slicing Method for Data Publishing, *11th International Conference on Machine Learning and Computing*, (Sf. 274-279).

Guo, K., & Zhang, Q., (2013), Fast clustering-based anonymization approaches with time constraints for data streams, *Knowledge-Based Systems*, 46, 95-108.

Hadjar, K., Jedidi, A., (2019), A New Approach for Scheduling Tasks and/or Jobs in Big Data Cluster, *MEC International Conference on Big Data and Smart City (ICBDSC)*, (Sf. 1-4).

Hashemian, H. M., (2010), State-of-the-art predictive maintenance techniques, *IEEE Transactions on Instrumentation and measurement*, 60 (1), 226-236.

Hayes, B. (2008), Cloud computing, *Communications of the ACM, (7):9–11*.

Hofgesang, P. I., Kowalczyk, W., (2005), Analysing clickstream data: From anomaly detection to visitor profiling, *Proc. of ECML/PKDD Discovery Challenge.*

Inan, A., Kantarcioglu, M., Bertino, E., (2009), Using anonymized data for classification, *IEEE 25th International Conference on Data Engineering* (Sf. 429-440).

Iyengar, V. S., (2002), Transforming data to satisfy privacy constraints, *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, (Sf. 279-288).

Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., (2017), Artificial intelligence in healthcare: past, present and future, *Stroke and vascular neurology*, 2 (4), 230-243.

Jiang, W., Clifton, C., (2006), A secure distributed framework for achieving k-anonymity, *The VLDB Journal*, 15 (4), 316-333.

Kargupta, H., Souptik Datta, Q. W., Krishnamoorthy, S., (2003), On the privacy preserving properties of random data perturbation techniques, *IEEE international conference on data mining*, (Sf. 99-106).

Kifer, D., Gehrke, J., (2006), Injecting utility into anonymized datasets, *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, (Sf. 217-228).

Kim, G.-H., Trimi, S., Chung, J.-H., (2014), Big-data applications in the government sector. *Communications of the ACM*, 57 (3), 78-85.

Laney, D., (2001), 3D data management: Controlling data volume, velocity and variety, *META group research note,* Vol. 6 (Sf. 70).

LeFevre, K., DeWitt, D., (2007), Scalable anonymization algorithms for large data sets, *University of Wisconsin-Madison Department of Computer Sciences.*

LeFevre, K., DeWitt, D. J., Ramakrishnan, R., (2005), Incognito: Efficient full-domain k-anonymity, Proceedings of the 2005 ACM SIGMOD international conference on Management of data, (Sf. 49-60).

LeFevre, K., DeWitt, D. J., Ramakrishnan, R., (2008), Workload-aware anonymization techniques for large-scale datasets, *ACM Transactions on Database Systems*, 33 (3), 1-47.

Li, J., Liu, J., Baig, M., Wong, R. C.-W., (2011), Information based data anonymization for classification utility, *Data & Knowledge Engineering***,** 70 (12), 1030-1045.

Li, J., Ooi, B. C., Wang, W., (2008), Anonymizing streaming data for privacy protection, *IEEE 24th International Conference on Data Engineering* (Sf. 1367-1369).

Li, J., Wong, R. C.-W., Fu, A. W.-C., Pei, J., (2008), Anonymization by local recoding in data with attribute hierarchical taxonomies, *IEEE Transactions on Knowledge and Data Engineering*, 20 (9), 1181-1194.

Li, N., Li, T., Venkatasubramanian, S., (2007), t-closeness: Privacy beyond k-anonymity and l-diversity, *IEEE 23rd International Conference on Data Engineering*, (Sf. 106-115).

Liu, J., Wang, K., (2010), On optimal anonymization for l+-diversity. *IEEE 26th International Conference on Data Engineering (ICDE 2010)*, (Sf. 213-224).

Liu, K., Kargupta, H., Ryan, J., (2005), Random projection-based multiplicative data perturbation for privacy preserving distributed data mining, IEEE Transactions on knowledge and Data Engineering, vol. 18, 92-106.

Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M., (2007), L-diversity: Privacy beyond k-anonymity, *ACM Transactions on Knowledge Discovery from Data (TKDD).*

Majeed, A., (2019), Attribute-centric anonymization scheme for improving user privacy and utility of publishing e-health data, *Journal of King Saud University-Computer and Information Sciences*, 31 (4), 426-435.

Marler, R. Timothy, Jasbir S. Arora., (2004), Survey of multi-objective optimization methods for engineering, *Structural and multidisciplinary optimization*, 26 (6), 369-395.

Martin, D. J., Kifer, D., Machanavajjhala, A., Gehrke, J., Halpern, J. Y., (2007), Worst-case background knowledge for privacy-preserving data publishing, *IEEE 23rd International Conference on Data Engineering*, (Sf. 126-135).

Meyerson, A., Williams, R., (2004), On the complexity of optimal k-anonymity, *ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, (Sf. 223-228).

Mohammadian, E., Noferesti, M., Jalili, R., (2014), FAST: fast anonymization of big data streams, *Proceedings of the 2014 international conference on big data science and computing*, (Sf. 1-8).

Mohammed, N., Fung, B. C., Hung, P. C., Lee, C.-K., (2010), Centralized and distributed anonymization for high-dimensional healthcare data, *ACM Transactions on Knowledge Discovery from Data*, 4 (4), 1-33.

Muralidhar, K., Parsa, R., Sarathy, R., (1999). A general additive data perturbation method for database security, *Management Science*, 45(10), 1399-1415.

Neubauer, T., Heurix, J., (2011), A methodology for the pseudonymization of medical data, *International journal of medical informatics*, 80(3), 190-204.

Riedl, B., Neubauer, T., Goluch, G., Boehm, O., Reinauer, G., Krumboeck, A., (2007), A secure architecture for the pseudonymization of medical data, *The Second International Conference on Availability, Reliability and Security (ARES'07)* (Sf. 318-324).

Sánchez, D., Sergio, M. J.-F., Jordi, S.-C., Montserrat, B., (2019), μ-ANT: semantic microaggregation-based anonymization tool, *Bioinformatics*, 36(5), (Sf. 1652-1653)

Sadiku, M. N., Musa, S. M., Momoh, O. D., (2014), Cloud computing: opportunities and challenges, *IEEE potentials*, 33 (1), 34-36.

Sakpere, A. B., Kayem, A. V., (2015). Adaptive buffer resizing for efficient anonymization of streaming data with minimal information loss. *International conference on information systems security and privacy (ICISSP)*, (Sf. 1-11).

Samarati, P., Latanya, S., (1998), Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression, SRI International.

Singh, S., Singh, N. (2012), Big data analytics, 2012 International Conference on Communication, *Information Computing Technology*, (Sf. 4).

Solé, M., Muntés-Mulero, V., Nin, J., (2012), Efficient microaggregation techniques for large numerical data volumes, *International Journal of Information Security*, 11(4), 253-267.

Sopaoglu, U., Abul, O., (2017), A top-down k-anonymization implementation for apache spark. *IEEE international conference on big data (big data)* (Sf. 4513-4521).

Sopaoglu, U., Abul, O., (2020), A utility based approach for data stream anonymization. *Journal of Intelligent Information Systems*, 1-27.

Sun, X., Sun, L., Wang, H., (2011), Extended k-anonymity models against sensitive attribute disclosure, *Computer Communications*, 34 (4), 526-535.

Sweeney, L., (2002), k-anonymity: A model for protecting privacy, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10 (5), 557-570.

Wang, G., Konolige, T., Wilson, C., Wang, X., Zheng, H., Zhao, B. Y., (2013), You are how you click: Clickstream analysis for sybil detection, *22nd USENIX Security Symposium*, (Sf. 241-256).

Wang, K., Fung, B. C., (2006), Anonymizing sequential releases, *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, (Sf. 414-423).

Wang, K., Yu, P. S., Chakraborty, S., (2004), Bottom-up generalization: A data mining solution to privacy protection, *Fourth IEEE International Conference on Data Mining (ICDM'04)*, (Sf. 249-256).

Wang, L., Zhan, J., Shi, W., Liang, Y., (2011), In cloud, can scientific communities benefit from the economies of scale, *IEEE Transactions on Parallel and Distributed Systems*, 23(2), 296-303.

Wang, P., Lu, J., Zhao, L., Yang, J., (2010), B-castle: An efficient publishing algorithm for k-anonymizing data streams, *WRI Global Congress on Intelligent Systems*, vol. 2, s. 132-136.

Wang, W., Li, J., Ai, C., Li, Y., (2007), Privacy protection on sliding window of data streams, *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, (Sf. 213-221).

Wangyal, S., Dechen, T., Tanimoto, S., Sato, H., Kanai, A., (2020), A Preliminary Study of Multi-Viewpoint Risk Assessment of IoT, *Bulletin of Networking, Computing, Systems, and Software*, 9(1), s. 40-42.

Wares, S., Isaacs, J., Elyan, E., (2019), Data stream mining: methods and challenges for handling concept drift, *SN Applied Sciences*, 1 (11), 1-19.

Wong, R. C.-W., Li, J., Fu, A. W.-C., Wang, K., (2006), (a, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing, *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (Sf. 754-759).

Xiao, X., Tao, Y., (2006), Anatomy: Simple and effective privacy preservation, *Proceedings of the 32nd international conference on Very large data bases*, (Sf. 139-150).

Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., Fu, A. W.-C., (2006), Utility-based anonymization using local recoding, *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, (s. 785-790).

Yao, C., Wang, X. S., Jajodia, S., (2005), Checking for k-anonymity violation by views, *Proceedings of the 31st international conference on Very large data bases*, (Sf. 910-921).

Ye, M., Wu, X., Hu, X., Hu, D., (2013), Anonymizing classification data using rough set theory, *Knowledge-Based Systems*, 43, 82-94.

Zakerzadeh, H., Osborn, S. L., (2013), Delay-sensitive approaches for anonymizing numerical streaming data, *International journal of information security*, 12 (5), 423-437.

Zakerzadeh, H., Osborn, S. L., (2010), Faanst: fast anonymizing algorithm for numerical streaming data, *International Workshop on Autonomous and Spontaneous Security*, (Sf. 36-50).

Zhang, Q., Koudas, N., Srivastava, D., Yu, T., (2007), Aggregate query answering on anonymized tables, *IEEE 23rd international conference on data engineering* (Sf. 116-125).

Zhang, X., Liu, C., Nepal, S., Yang, C., Dou, W., Chen, J., (2013), Combining top-down and bottom-up: scalable sub-tree anonymization over big data using MapReduce on cloud, *IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, (Sf. 501-508).

Zhang, X., Yang, C., Nepal, S., Liu, C., Dou, W., Chen, J., (2013), A MapReduce based approach of scalable multidimensional anonymization for big data privacy preservation on cloud, *International conference on cloud and green computing* (Sf. 105-112).

Zhang, X., Yang, L. T., Liu, C., Chen, J., (2013), A scalable two-phase top-down specialization approach for data anonymization using mapreduce on cloud, *IEEE Transactions on Parallel and Distributed Systems*, 25(2), 363-373.

Url-1 <https://archive.ics.uci.edu/ml/datasets/Adult>, alındığı tarih: 20.01.2020.

Url-2 <https://archive.ics.uci.edu/ml/datasets/nursery/>, alındığı tarih: 21.01.2020.

Url-3 <https://aws.amazon.com/kinesis/data-streams/>, alındığı tarih: 21.01.2020.

Url-4 <http://cassandra.apache.org/>, alındığı tarih: 20.01.2020.

Url-5 <https://cloud.google.com/dataflow>, alındığı tarih: 21.01.2020.

Url-6 <https://flink.apache.org/>, alındığı tarih: 21.01.2020.

Url-7 <https://hadoop.apache.org/>, alındığı tarih: 21.01.2020.

Url-8 <https://kafka.apache.org>, alındığı tarih: 21.01.2020.

Url-9 < https://www.kaggle.com/blastchar/telco-customer-churn/data/>, alındığı tarih: 20.01.2020.

Url-10 <https://kvkk.gov.tr/>, alındığı tarih: 21.01.2020.

Url-11 < https://www.marketsandmarkets.com/Market-Reports/big-data-market-1068.html />, alındığı tarih: 27.01.2020.

Url-12 <https://www.oracle.com/big-data/guide/what-is-big-data.html>, alındığı tarih: 21.01.2020.

Url-13 <https://openrefine.org/>, alındığı tarih: 23.01.2020.

Url-14 <https://spark.apache.org/>, alındığı tarih: 20.01.2020.

Url-15 <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>, alındığı tarih: 20.01.2020.

Url-16 <https://www.statwing.com/>, alındığı tarih: 23.01.2020.

Url-17 <http://storm.apache.org/>, alındığı tarih: 20.01.2020.

ÖZGEÇMİŞ

Ad-Soyad : **Uğur SOPAOĞLU**

Uyruğu : **T.C.**

Doğum Tarihi ve Yeri : **27/06/1989**

E-posta : **usopaoglu@gmail.com**

ÖĞRENİM DURUMU:

- **Lisans** : 2012, Çankaya Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği
- **Yüksek lisans** : 2014, Çankaya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği
- **Doktora** : 2020, TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	Yer	Görev
2012-2018	Çankaya Üniversitesi	Öğretim Görevlisi
2018 – Halen	Havelsan A.Ş.	Arge Mühendisi

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Sopaoglu, U.** and Abul, O., 2017. A top-down k-anonymization implementation for apache spark, 2017 IEEE international conference on big data (big data). IEEE.
- **Sopaoglu, U.** and Abul, O., 2019. A utility based approach for data stream anonymization. (published online, to appear) Journal of Intelligent Information Systems, DOI: https://doi.org/10.1007/s10844-019-00577-6.